UNDERSTANDING WAVELETS THROUGH ANALYZING SIGNALS.

In this homework we will try to understand how specific wavelets (and scaling functions) permit the analysis and detection of some specific signal characteristics. These will include detection of breaks in a signal, identification of sinusoidal frequency, the ability of wavelets to “not see” parts of a signal, determine long-term trends, etc. In particular, we will examine the two wavelets, \( db3 \) and \( db5 \) that are applied in two examples, “Sum of Sinusoids” and “A Frequency Breakdown”.

First do the following: Go to the Wavelet Toolbox Main Menu and invoke Wavelet Display. Since you will be examining how different wavelets (and scaling functions) behave, go to Wavelet Display and put ‘db’ and ‘3’ in the box and press Display. Look at the wavelet \( \psi \) and the associated scaling function \( \phi \). Note in particular, their support and determine approximately what might be their pseudo-frequency. Recall that for wavelets, ‘db3’ here, you can also use the commands cfreq=centfrq(’db3’,10,’plot’) and scal2frq(4,’db3’,1) to determine the corresponding (pseudo) frequencies for a given scale. Also look at the information on Daubechies Family (DB) noting the “regularity” and number of “vanishing moments” (we will cover these in class).

Now we examine the two examples. You can also find these examples in the MATLAB Help menu under Wavelet Toolbox, Examples, Wavelet Decomposition (Example 1. Sum of Sines). There you will find some analysis of these examples. In the Wavelet Toolbox Main Menu, go to Wavelet 1-D. (You can also access all these examples from the other One-Dimensional wavelet implementations).

The basic 2-channel DWT filter bank looks like so:

\[
\begin{array}{c}
\text{Analysis Filter Bank} \\
L_1 \quad \downarrow 2 \\
H_1 \\
\uparrow 2 \quad \downarrow 2 \\
L_2 \\
H_2 \\
\text{Synthesis Filter Bank} \\
x[n]
\end{array}
\]

Figure 1: The basic 2-channel DWT filter bank
The “dyadic” decomposition of a discrete-time signal using the DWT, is as follows:

1. **Sum of Sines**

![Diagram showing the dyadic decomposition of a discrete-time signal using the DWT.]

- **Figure 2**: Detail and Approximation coefficients

(1) This signal *sumsin.mat* is easily analyzed in the Fourier domain. You can get it in your workspace by *loadsumsin*. Do a DFT (fft) on the signal and confirm that that the 3 frequencies have approximate periods of 200, 20 and 2. Note that the sampling period is assumed to be 1.

(2.) We see how well wavelets work here. In the **Wavelet 1-D** window, do File → Example Analysis → Basic Signals → Sum of Sines. The example defaults to ‘db3’ and five levels. Also do View → Default Display Mode → Separate Mode since this display is probably the most useful for this analysis.

(3.) The 3 sinusoids have periods 2, 20 and 200 and hence frequencies 0.5, 0.05 and 0.005 respectively. ‘db3’ has a pseudo frequency of 0.8.

(4) For an easier determination of periods of the reconstructed signals, you can operate at the command level with the Discrete Wavelet Transform (dwt). Analysis and synthesis with the dwt, can be done as follows: For signal *sumsin* (Do a ‘load sumsin’ first):

```matlab
s = sumsin; [C, L] = wavedec(s,5,'db3');
```

To extract the level 3 approximation coefficients from C, type:

```matlab
cA3 = appcoef(C,L,'db1',3);
```

To extract the levels 3, 2, and 1 detail coefficients from C, type:

```matlab
cD3 = detcoef(C,L,3);
cD2 = detcoef(C,L,2);
cD1 = detcoef(C,L,1);
```

or

```matlab
[cD1, cD2, cD3] = detcoef(C,L,[1,2,3]);
```

To reconstruct the level 3 approximation from C, type:

```matlab
A3 = wrcoef('a',C,L,'db1',3);
```

To reconstruct the details at levels 1, 2, and 3, from C, type:

```matlab
D1 = wrcoef('d',C,L,'db3',1);
D2 = wrcoef('d',C,L,'db3',2);
D3 = wrcoef('d',C,L,'db3',3)
```
D1 primarily contains the signal with period about 2 and frequency about 0.5. (You can check this by doing a dft on D1 and see if you get a dominant frequency around 0.5 (k \approx 500 for a 1000-pt dft. (x=fft(sumsin,1000)). Recall that the sampling frequency is assumed to be 1).

Confirm that the medium frequency (.05) sine is picked up in D3 and in D4.

Note that each of the approximations \( A_1 \rightarrow A_3 \) can also be used to estimate the medium frequency. You can confirm this by doing a DFT on \( A_1 \rightarrow A_3 \).

Finally, the slow sine, is seen in A4 and A5.

2. A Frequency Breakdown

This signal is freqbrk.mat. To get the signal in your workspace, do >> load freqbrk . (Your File \rightarrow Set Path should by default include the path to the directory wavedemo). In the Wavelet 1-D GUI, as in the previous example, load Basic Signal - Frequency Breakdown. This one gives a 5-level DWT analysis using 'db5'. There are two sine waves cascaded. The slow sine with period 200 and a faster sine with period 20. Do a DFT on this signal in the workspace. Even though this signal is not stationary, does the DFT resolve the 2 frequencies?

Issues here are to see how 'db5' resolves this signal at the various levels. Note the reconstructed signal from the wavelet coefficient \( d_1 \) and approximation coefficient \( a_5 \). Let us consider identification of the “rupture” at 500 and assume that we are only interested in identifying just the point where it occurs. Try various wavelets and see how cleanly the reconstructed signal identifies the rupture.

Report

Comment on results in the Frequency Breakdown simulation..

Determine the number of coefficients, detail and approximation that you have at each level of the wavelet decomposition, using 5 levels of decomposition. Use any input signal of length greater than \( 2^5 \).