SATFD

Lab. 5

May 9, 2024

Exercise 5: Wavelet transform

Introduction

The exercise aims to introduce time-frequency analysis using the wavelet transform. As an example of a non-stationary signal, we will use speech signals and synthetic signal.
One of the fundamental issues in the analysis, recognition, and synthesis of human speech is the recognition of so-called formants - time-frequency structures corresponding to specific phonemes. Formant analysis is also carried out in the case of analyzing sounds produced by animals (bats, whales). Such analysis is also useful in engineering issues - see (Białasiewicz, Falki i aproksymacje, WNT, 2000) - where the appearance of low-frequency vibrations is interpreted as the effect of fatigue of the tested mechanisms or a design problem, which should be prevented by modifying the vibrating mechanisms. The wavelet transform, like other time-frequency analysis methods, is often used for such analysis. Here, we will familiarize ourselves with a numerically oriented approach that allows for faster calculations. It is worth mentioning that there are implementations of the discrete wavelet transform (DWT) that allow for its determination in real time, which allows for the use of its results in industrial controllers.
A linear system (quadripole) can be treated as a filter in the frequency domain. Frequency domain filtering is equivalent to convolution operation in the time domain. Multiplying the input signal (e.g., harmonic) by the transfer function gives the system's response in the frequency domain. Similarly, the system's response in the time domain can be obtained by convolving the input signal with the unit step response function, which is the inverse Fourier transform of the transfer function.
The convolution operation can be perceived as the scalar product of the investigated signal and a given base function. Thus, the output of the linear system (filter) is composed of sequentially determined scalar products, where the base function is the unit step response. The result are the coefficients of the investigated signal in the expansion of the basis of infinitely many copies of the h function, differing in the argument, which is shifted by 1 sample each time. Such an operation is only possible with a finite-length carrier of basic functions. With the appropriate construction, such functions for shifted argument values are orthogonal.
The operation described above is the essence of the wavelet transform, which is repeated first for different time locations of the "center" of the base function, and then, as described below, for different time scales.
In the wavelet transform, the signal is passed at each level through a pair of filters, which have the property of mirrored conjugate filters (quadrature mirror filters - QMF): the sum of the power of the signal at the output of the filters is equal to the power of the signal at the input. At the same time, the sum of the squares of the transfer functions is equal to 1. One of the filters is low-pass (LP), and the other is high-pass (HP). The signal at the output of the HP filter are the coefficients in the expansion of the wavelet basis (this is also the result of the wavelet transform at a given level), while the signal at the output of the LP filter are the coefficients in the expansion of the scaling function basis. Both groups of coefficients are then undergone so-called decimation - every second value is used. In this way, the analyzed frequency band is shifted down by half its value. Then the filtering operation is repeated: this time the input to both filters is the vector of coefficients that came out of the low-pass filter. In this way, at the output of the HP filter, we obtain successive coefficients in the expansion of the wavelet basis scaled in the time domain (and also frequency). After multiple executions of these operations, we obtain the so-called scalogram: a set of wavelet transform coefficients corresponding to expansions in the wavelet basis at different scales. The technique described above realizes stretching

on a dyadic grid, where the scale changes by $\frac{1}{2}$.
Therefore, the analyzed signal is each time 2x shorter, and the procedure is terminated after performing the analysis at a specified number of scales, with the number of scales being either specified as a parameter or determined by the length of the input signal - as $log_2 N$, where N is the length of the signal.
The above-described operation can be reversed, which is done by the so-called inverse wavelet transform. The difference is that instead of the LP and HP filters of the wavelet analysis, synthesis filters with different coefficients are inserted, and after changing the scale, the signal is padded with zeros, which is the opposite of decimation.
If some scales are omitted during synthesis, the output signal is subjected to filtering. The same path can be used for signal compression. This operation is used in the JPEG-2000 standard.
Filtering at multiple scales in numerical practice often uses the same filters multiple times, with the input signal changing. Formally, this operation can be presented as filtering through a set of filters.
Each wavelet has its frequency range - essentially covering a certain frequency band. Therefore, the scalogram can formally be transformed into a time-frequency diagram by assigning each scale the central frequency of the wavelet at that scale.

Realization of the task

Suggested environment: Matlab, Python.

Prepare a script that will perform the following tasks:

1. Generate the basic Coiflet1 wavelet:

```
    f = [ .038580777748, .126969125396, -.077161555496, -.607491641386,
    .745687558934, -.226584265197 ]
```

2. Analyze any non-trivial fragment of a signal of length 2^L using discrete wavelet transform, which means:

   - Create a mirror filter for the Coiflet - low-pass through the operation:

   ```
           y = -( (-1).^(1:length(f)) ).*f;
   ```

   - Perform filtering, i.e., carry out cyclic convolution (see lecture) of the signal with the wavelet filter (high-pass) and the mirror filter for the wavelet (low-pass).
   - Perform decimation of the coefficients, which constitute the result of convolution for both filters.
   - Store the coefficients of filtering with the high-pass filter as wavelet coefficients for a given scale.
   - Substitute the decimated coefficients of filtering with the low-pass filter as the signal for analysis in the next scale.
   - Perform the above operation the maximum possible number of times (for a given signal length).

3. Then, determine the central frequency of the wavelet for each scale.

4. Next, plot the scalogram of the analyzed signal, with the frequency axis scaled in Hz.

5. Please start the analysis with simple signals (harmonic signal, chirp) to clearly observe what happens in the results, and then analyze the signal from the file "rabarbar.wav".

6. Note: You can choose any other wavelet, as long as it's not db4 and of length different than 4.

7. Compare obtained results with the results obtained using CWT (continuous wavelet transformation). To do so, find proper library.

Comment on all obtained results.