Signal Denoising using Wavelets in MatDeck

In this example, we use MatDeck wavelets to perform signal denoising using the thresholding approach. Wavelet thresholding denoising is very popular among engineers. Properties of the approach were investigated in a series of papers. MatDeck contains series of the function for wavelet signal processing including single level discrete wavelet decomposition discretewavet(), multilevel discrete wavelet decomposition wavedec(), discrete wavelet packet decomposition wavepacketd() and corresponding reconstruction functions. MatDeck supports over 100 different wavelet filters which can be obtained by wavefilter().

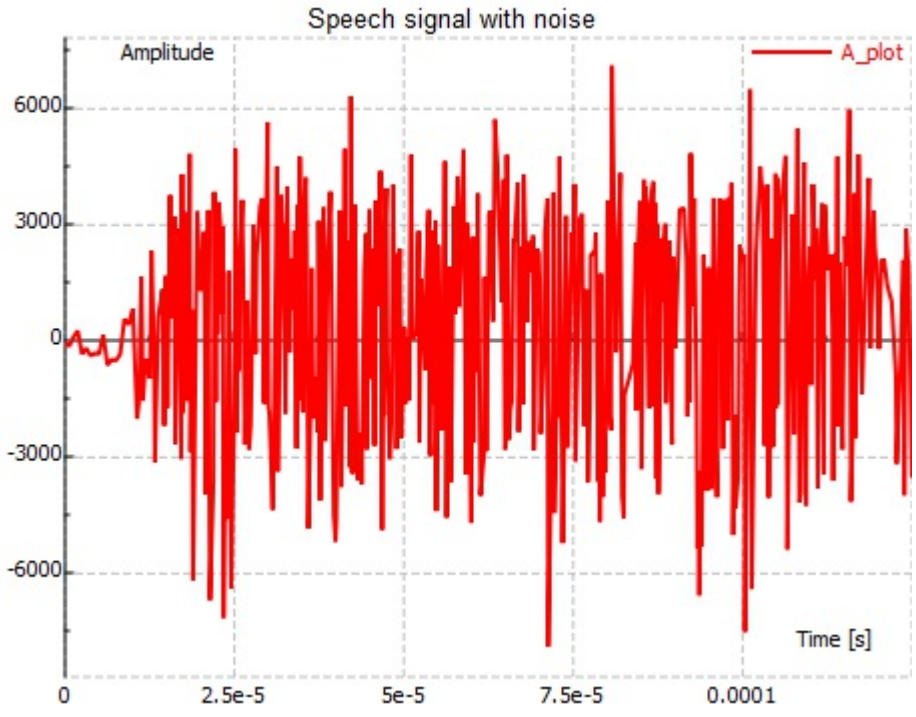
Noisy speech signal

Signal denoising is often needed for speech signals which are corrupted with additive Gaussian white noise (AWGN). In this example, we use a recorded speech signal mixed with AWGN which is read from a *.wav file.

```
filename := "nsa_wbn.wav"          File containing the noisy audio
data_f := wave read(filename)      Reading the audio file
props := wave properties(filename) Parameters of the audio format
data := subset(data_f , 1000 , 0 , 1999 , 0)   Take only the subset of the signal for illustration
audio play(props[0] , props[1] , data)
```

We plot a portion of the audio signal in the time domain. The small changes in the signal trend is actually noise.

```
t_time := xnodes([0  1] , size(data) - 2) / props[0]
A_plot := join mat cols(t_time^T, data)
```



Wavelet Packet Denoising

Next, we use the wavelet packet decomposition at the second level using Daubechies wavelet 12 known as db12. The decomposition produces four groups of the coefficients: approximation and three details of the different levels. All these coefficients are displayed in the next figure.

$W\_mat := \text{wavepacketd}(data\ ,\ 2\ ,\ "db12")$ Wavelet packet decomposition at level 2

$W\_signal := W\_mat[0]$ aproximation and detail coefficients

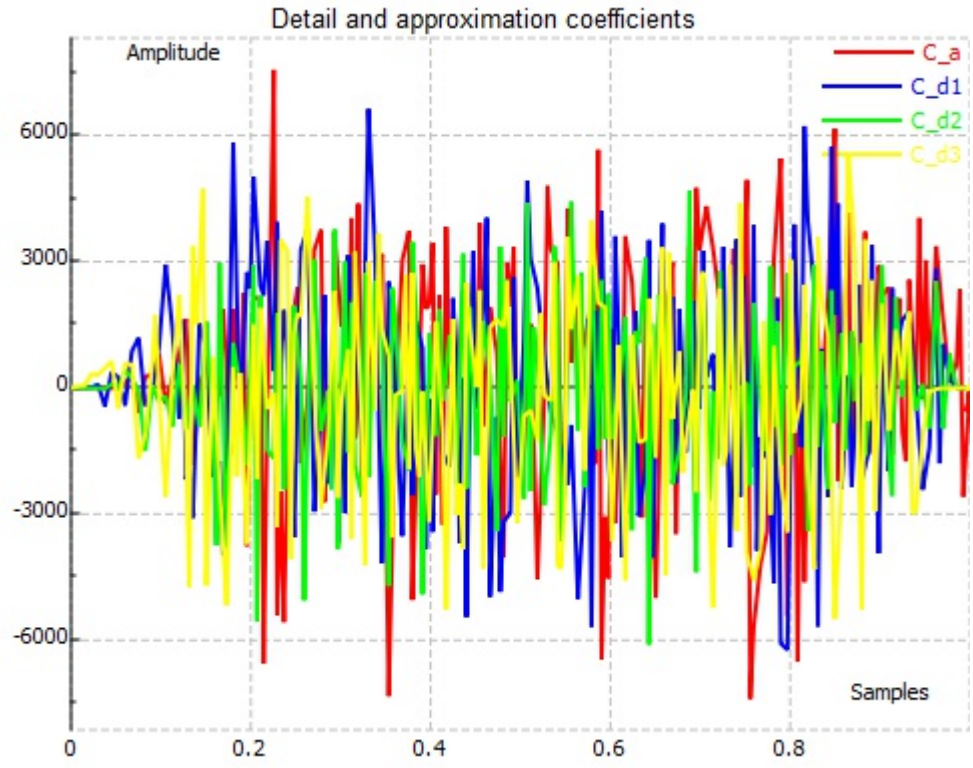
$$N\_sampl := \text{xnodes}\left(\begin{bmatrix} 0 & 1 \end{bmatrix}\ ,\ \text{size}\left(\text{col2vec}(W\_signal\ ,\ 0)\right) - 2\right)^{T}$$

$$C\_a := \text{subset}\left(\text{join mat cols}\left(N\_sampl\ ,\ \text{col2vec}(W\_signal\ ,\ 0)\right)\ ,\ 0\ ,\ 0\ ,\ \text{rows}(N\_sampl) - 1\ ,\ 1\right)$$

$$C\_d1 := \text{subset}\left(\text{join mat cols}\left(N\_sampl\ ,\ \text{col2vec}(W\_signal\ ,\ 1)\right)\ ,\ 0\ ,\ 0\ ,\ \text{rows}(N\_sampl) - 1\ ,\ 1\right)$$

$$C\_d2 := \text{subset}\left(\text{join mat cols}\left(N\_sampl\ ,\ \text{col2vec}(W\_signal\ ,\ 2)\right)\ ,\ 0\ ,\ 0\ ,\ \text{rows}(N\_sampl) - 1\ ,\ 1\right)$$

$$C\_d3 := \text{subset}\left(\text{join mat cols}\left(N\_sampl\ ,\ \text{col2vec}(W\_signal\ ,\ 3)\right)\ ,\ 0\ ,\ 0\ ,\ \text{rows}(N\_sampl) - 1\ ,\ 1\right)$$



Signal reconstruction

After thresholding has finished, the signal is reconstructed using a wavelet packet reconstruction with the same Daubechies wavelet 12. We can inspect the noise reduction by hearing it.

```
W_matr := [ W_signal ]
          [ W_mat[1] ]      Prepare the data structure for reconstruction

data_r := wavepacketr(W_matr , "db12")      Wavelet packet reconstruction

audio play(props[0] , props[1] , data_r)
```

Hard thresholding

Detail coefficients of the highest precision, denoted by C_d3, are presumed to contain mostly AWGN. In order to remove the noise, we compare the values to the threshold. All samples whose absolute value is smaller than the predefined threshold will be set to zero. The method is known as hard thresholding, and is implemented within the script function below.

```
th := 1000      Threshold value
W_signalr := hthreshold(W_signal , 10000 , 3)      Hard thresholding

hthreshold(mat_in , th_value , ind)
{
  N_col := cols(mat_in)
  N_row := rows(mat_in)
  Temp := mat_in
  for(i := 0 , i < N_row , i += 1 )
  {
    if( abs(value at(Temp , i , ind)) < th_value )
    {
      Temp[i · N_col + ind ] = 0
    }
  }
  return(Temp )
}
```