SIGNAL PROCESSING ASSIGNMENT 2

SVEN-S. PORST

SSP@EARTHLING.NET

Exercise (2-4, i) – de-noising a given signal using FFT

We start off with a vector $v = (\sin \frac{4\pi j}{1024} + \eta_j)_{j=1}^{1024}$, where η is a vector of size 1024 whose entries are independent identically distributed uniform random variables in $[-0.1, 0.1]$. Performing FFT on this vector gives the vector v' whose absolute value is depicted in figure 2.

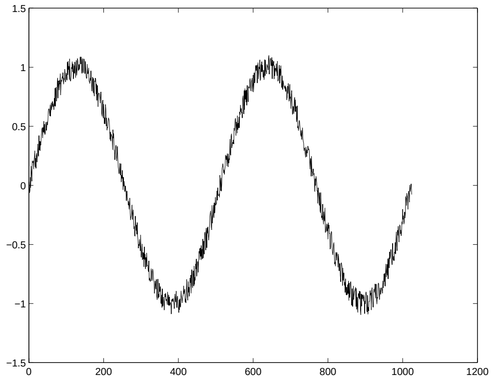

Figure 1: The original vector v given by $v = (\sin \frac{4\pi j}{1024} + \eta_j)_{j=1}^{1024}$ with η being noise.

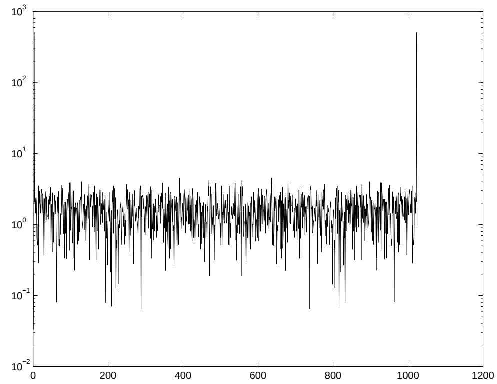

Figure 2: absolute value of v' plotted with logarithmic scale.

The maximum value of v' is approximately $\xi = 512$. The next step is to try to de-noise the signal by removing the small entries in v' caused by the noise. By looking at figure 2 one 'sees' that the noisy entries of v' are smaller than $\sqrt{10}$. Thus setting all entries of v' which are smaller than, say, $\frac{\xi}{100}$ can be expected to give good de-noising results. Indeed, this perfectly restores the original function.

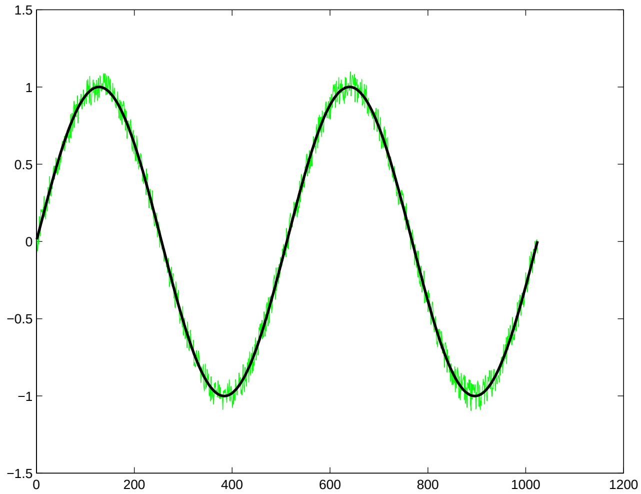

Figure 3: Graph of de-noised signal with $\alpha = \frac{\xi}{100}$ (black curve) compared to original noisy signal (grey curve).

Choosing α to be smaller than $\frac{\xi}{100}$, say, $\alpha = 3$ will only slightly de-noise the signal:

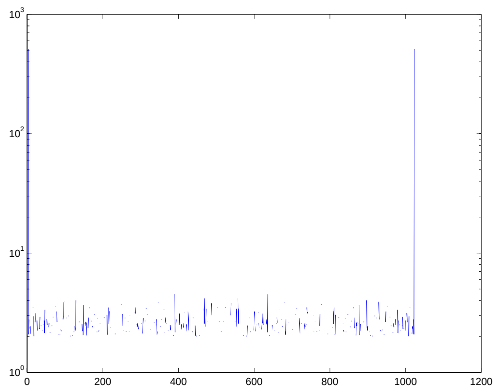

Figure 4: v' with all entries < 3 set to zero.

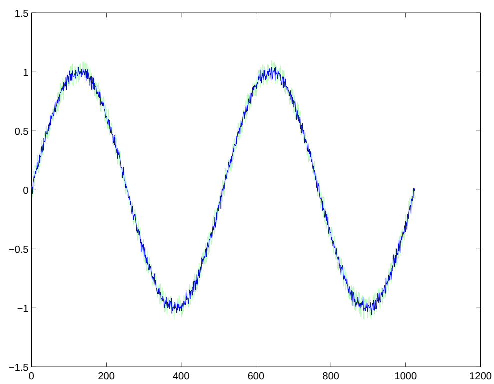

Figure 5: IFFT of v' with all entries < 3 set to zero.

Exercise (2-4,ii) – de-noising my own signal using FFT

The signal I chose is a small part of a signal recorded from the dialing tone or german telephones. This is supposed to be a sine curve at 440 Hertz. Given that, before recording the signal went through long distances of cable and through an old telephone, a certain amount of noise can be expected. Especially there should be strong noise at the frequency of 50 Hertz which is induced by the alternating current of electricity lines. The signal was recorded with a rate of 44100 samples per second, which makes the section analyzed here be a part of the sound lasting approximately 0.023 seconds for. A graph of the signal is given in figure 6.

As above, we can perform a FFT on this signal, which gives a symmetric vector because our original signal was real. The first half of this vector is shown in in figure 7. The absolute value of the maximal entry of the FFT is $\xi = 55$ and it is located at

Trying to de-noise by throwing away all entries that are smaller than three (i.e. $\alpha = \frac{3\xi}{55}$) doesn't do the job, as most of the noise remains. The following table illustrates this by giving the absolute values of those coefficients of the FFT-vector that remained:

i	2	3	4	7	8	9	10	11
$\lvert v'(i)\rvert$	5.3781	4.3935	5.2221	4.3012	5.2804	8.3439	14.998	55.7959
i	12	13	14	15	16		69	70
$\lvert v'(i)\rvert$	11.3157	4.6448	5.6726	3.9292	3.1267		5.0547	3.1678

As there are some rather large coefficients left at positions 69 and 70 this will give high frequency noise. Thus α has to be chosen slightly larger, say $\alpha = \frac{\xi}{10}$, to achieve satisfactory de-noising results which are presented in figure 8. The table of non-zero values has shrunk now to the following:

i	9	10	11	12	14
$\lvert v'(i)\rvert$	8.3439	14.998	55.7959	11.3157	5.6726

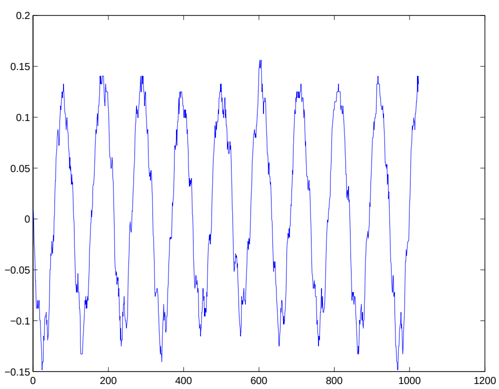

Figure 6: My own signal: Hopefully 440 Hertz with intensive noise at 50 Hertz and minor noise at other frequencies

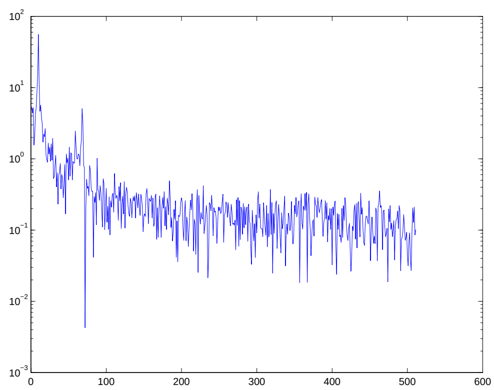

Figure 7: Absolute value of first half of FFT of my own signal plotted with logarithmic scale.

This leaves us with a rather strong peak around 11 (in the Matlab basis, i.e. 10 in our basis, which corresponds to a wave with 10 oscillations in 0.023 seconds which is a wave of roughly 435 Hertz which is pretty close to the 440 Hertz we expected. Furthermore, by looking at the graph we can see the signal being subjected to one period of distortion which would come from the 50 Hertz signal of the alternating current. Interestingly, this isn't reflected in the coefficients that were given to the IFFT – as I would expect the second component (in the Matlab basis, i.e. the first component in our basis) to contribute this. But as we can see in the table, the second component is zero. Thus, I assume, what looks like (and presumably *is*) the 50 Hertz noise is hidden somewhere in the coefficients around 11.

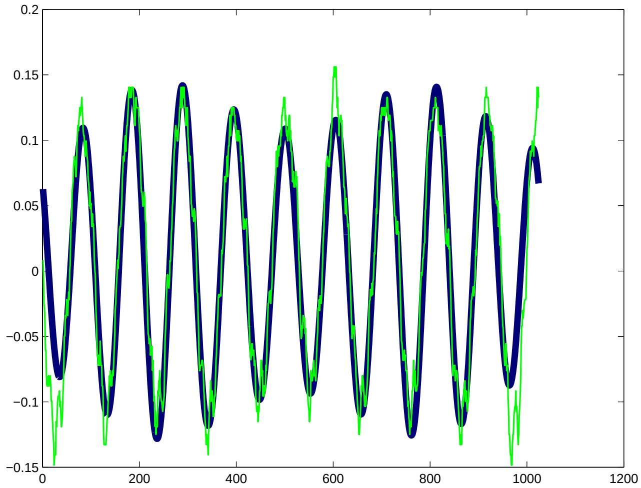

Figure 8: Graph of de-noised signal with $\alpha = \frac{\xi}{10}$ (black curve) compared to original noisy signal (grey curve).

Exercise (2-5, i) – playing with wavelets using a given signal

For this exercise I chose the signal *mishmash* that is supplied with the wavelet toolbox examples. The signal starts oscillating at a low frequency with the frequency increasing as time goes on. Furthermore there seems to be some high frequency noise in the signal (see figure 9).

Playing with different wavelet types shows us that the reconstructed signals we get using the Haar wavelet are – as we expected – not very smooth, when stripping the high-level[1] details. Thus the Daubechies wavelet seems to be more lovable concerning smoothness. Furthermore we observe that the db$x$-wavelets become smoother as x increases.

Having seen that the Daubechies wavelet is more suitable for this job – the further observations refer to experiments done with the db8-wavelet. Now, that we have fixed a wavelet it seems to be a good time to play around with the level of the analysis. As we have seen in the lecture when the idea of wavelets was introduced, the data we get at each *level* of the analysis

[1] *high*-level means *small* indices, e.g. d_1

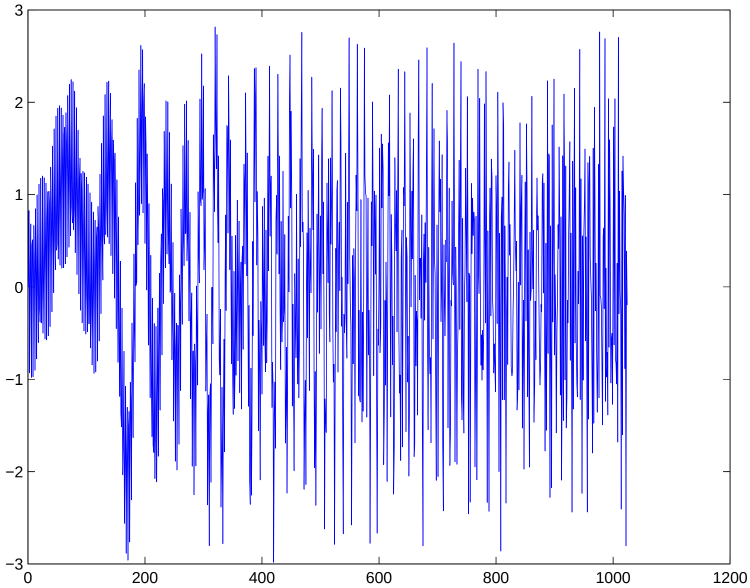

Figure 9: The *mishmash*-signal as taken from the wavelet toolbox example signals. The signal starts with low frequency oscillations but frequency increases as time goes on.

corresponds to information at a certain *scale* in the signal. Thus an analysis with a higher number of levels will allow us to distinguish information at a higher number different scales in the signal.

Apparently, the *mishmash*-signal reflects this property of wavelets in a very nice way. As the frequency of the *mishmash*-signal increases as time goes on, we need only low-detail, i.e. low-level, i.e high-indices) detail-information at the beginning of the signal but increasingly higher detail information as time goes on. Looking at the graphs in the wavelet toolbox's full-decomposition mode this phenomenon can be seen (see figures 10 and 11): In a high-level anaylysis at each detail-level there is only a short period of time in which that level of detail is non-zero, whereas in a low-level analysis these periods of time are longer.

The fact, that the analysis of the *mishmash*-signal shows that each level of detail can be localized very well, suggests, that, for this signal, windowed techniques that we've seen for the FFT in the lecture, may work very well to do compression and de-noising as this would allow us to keep just one level of detail for each of the (suitably chosen) windows.

Using the Compression and de-noising features of the wavelet-toolbox gave mixed results as it was well possible to get a nice and smooth signal by applying these techniques and playing around with the levels of threshold – however, as I don't know what the signal is supposed to look like without noise, it doesn't make too much sense to do this as having a nice and smooth signal in the end may aswell mean that we destroyed some of the interesting information contained in the signal by regarding it as noise.

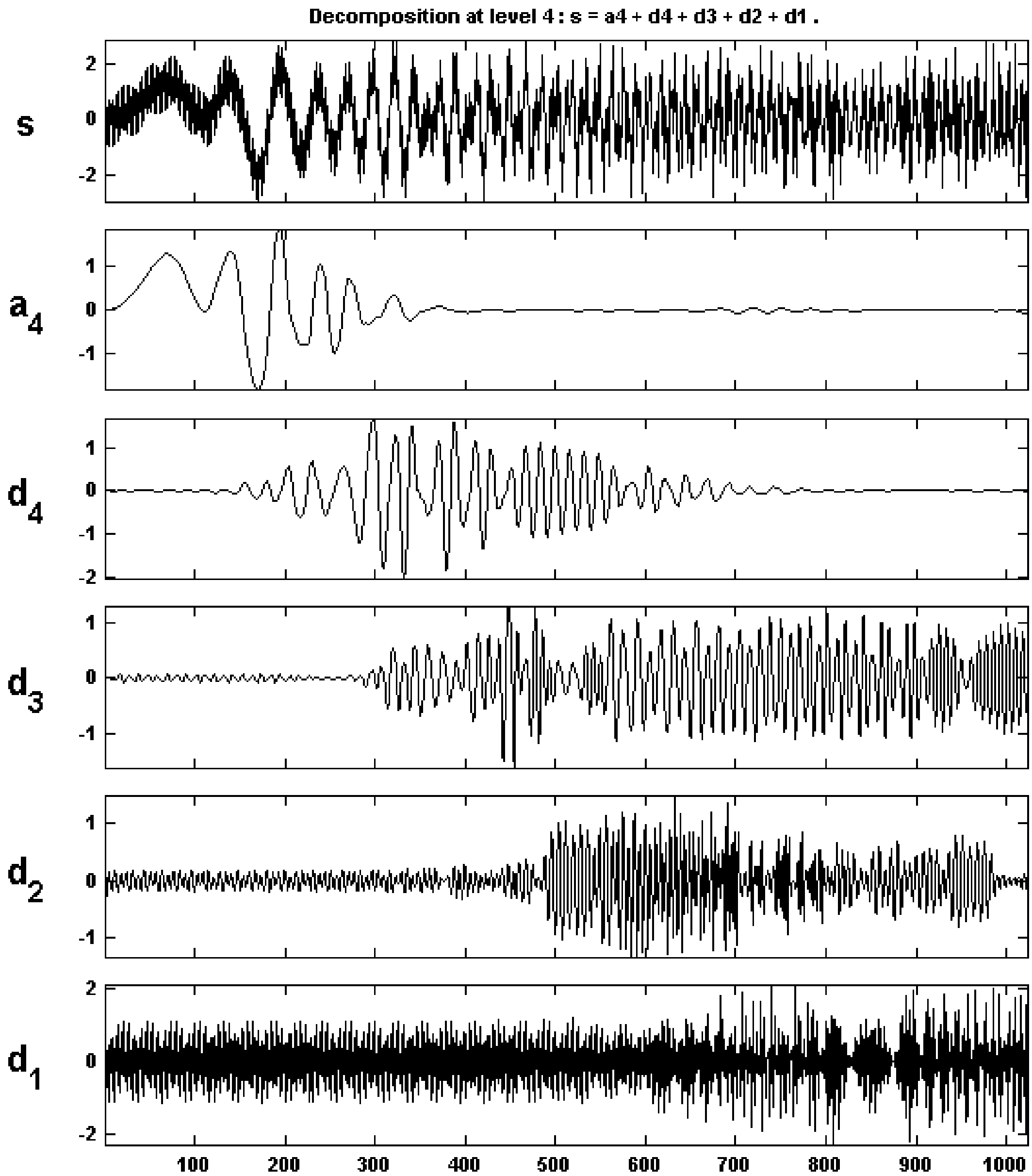

Figure 10: Full decomposition in five levels of the *mishmash*-signal using the db8-wavelet. We can see that there is only little high-detail information at the beginning of the signal as d_4, d_3, d_2 are nearly zero there.

The poor quality of this figure and figures 11 and 12 is caused by the fact that there seemed to be no better possibility to get hold of an image of the full decomposition than taking a screenshot.

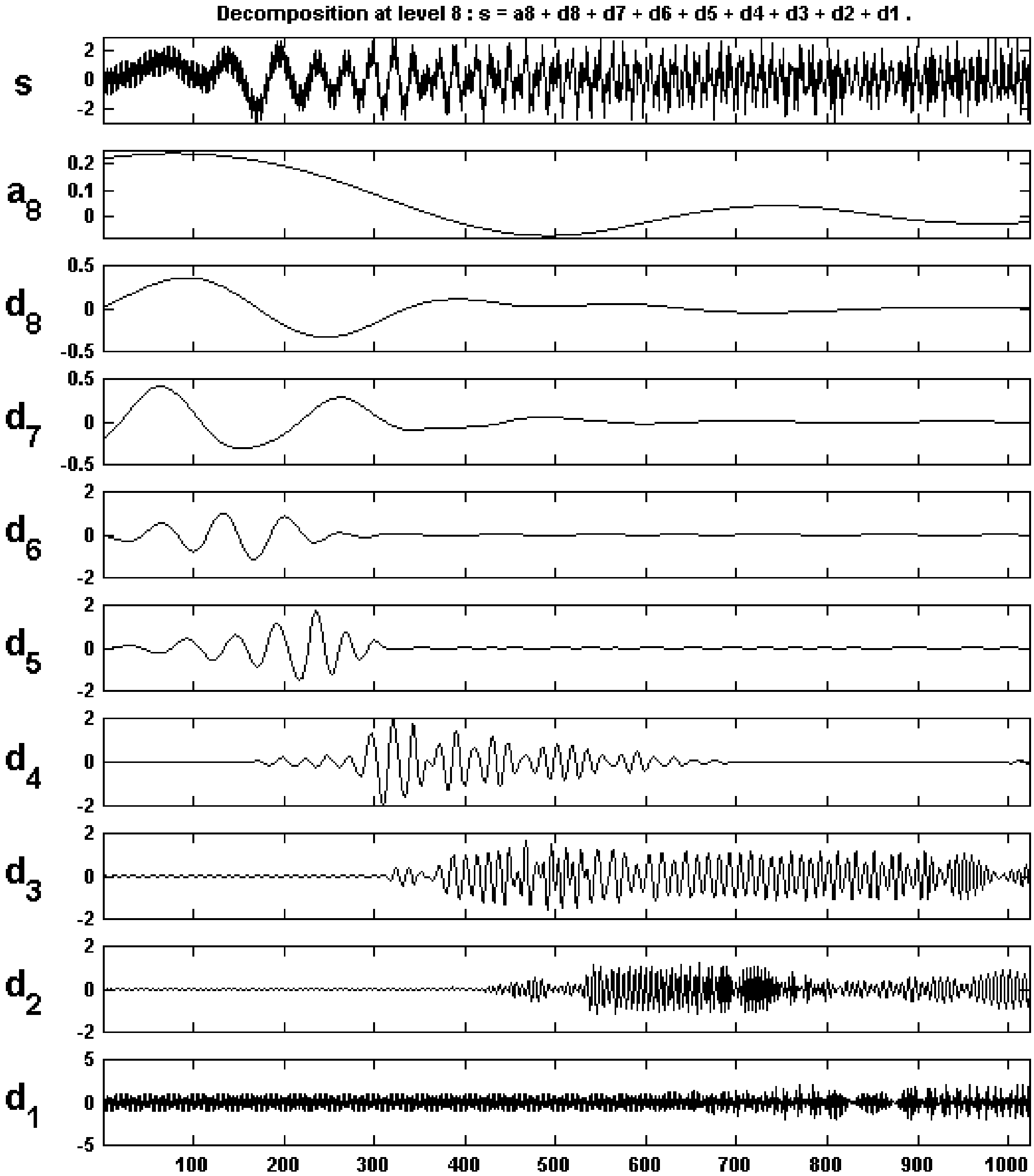

Figure 11: Full decomposition in eight levels of the *mishmash*-signal using the db8-wavelet. Even more than in figure 10 we can see that there is only little high-detail information towards the beginning of the signal and that there is only litte low-detail information towards its end.

Exercise (2-5, ii) – playing with wavelets using my own signal

When using my own signal, the observations made when playing around with the different wavelets were mainly the same as in part (i) of this exercise. In particular, de-noising and compression results were good when thresholding away nearly all of the coefficients except for a few low-detail ones.

In addition to these observations, my own signal featured a significant bonus over the *mish-mash*-signal I looked at in part (i): When doing analysis with the db8-wavelet (other sufficiently smooth db-wavelets will do here as well) at level 4, the coarsest scale happens to coincide very well with a de-noised version of my signal. This is shown in figure 12. Thus to do compression and de-noising in a very efficient way, all we have to do is to dump *all* of the detail-coefficients, as the de-noised signal is completely contained in the remaining average (a_4) coefficients. When doing compression this gives us spectacularly good values: We can retain 97.5% of the energy while having 92.8% zeroes.

However, I believe that these good results in compression and de-noising cannot be achieved for any signal. My signal seems to be a lucky choice for treatment with that special wavelet at that special level. Given a suitable choice of different wavelets and a fair bit of patience this result suggests that when doing de-noising and compression with wavelets you might want to choose the wavelet and the level of analysis you use carefully to suit your signal. This – possibly together with windowed techniques – may significantly improve the levels of de-noising and compression that can be achieved. Please be assured that these last remarks are purely speculative and lack any mathematical rigour or background.

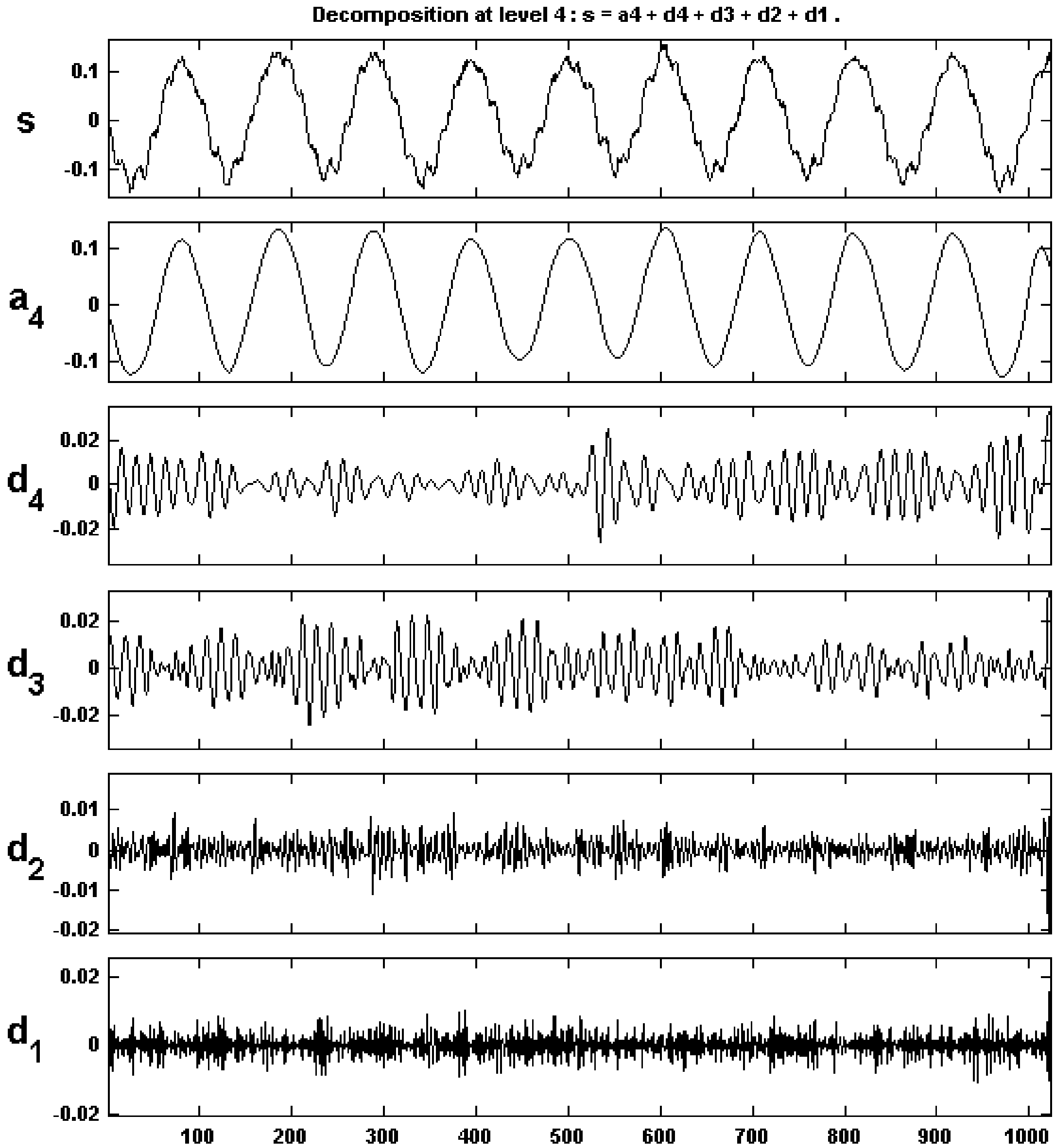

Figure 12: Full decomposition in four levels of my own signal signal using the db8-wavelet. Amzingly the averaged signal coincides very well with what we would think of as a de-noised version of the signal.