Lab 2 report - Frame synchronization

Thomas Verelst

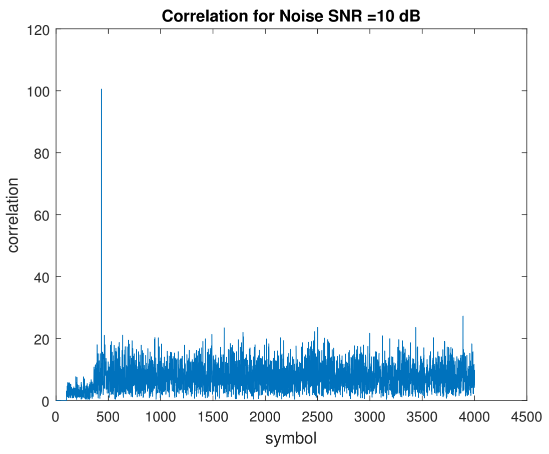
11 October 2016

Exercise 1

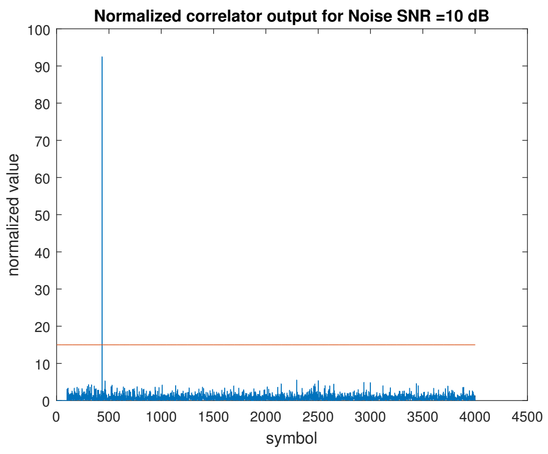
See Matlab code in file *exercise1.m*
The preamble is 11111111000010111100011010000000001000111000
10010111000000011001001001101110010000010101101101011001011

Exercise 2

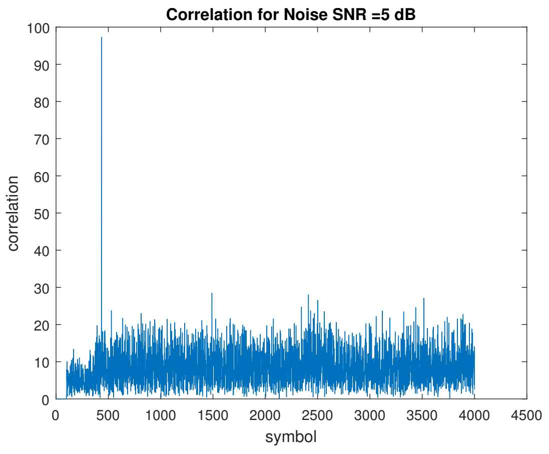
See Matlab code in file *exercise2.m* and the correlation function in *correlator.m*
To keep the plot clear the number of incombing symbols is limited to 4000.



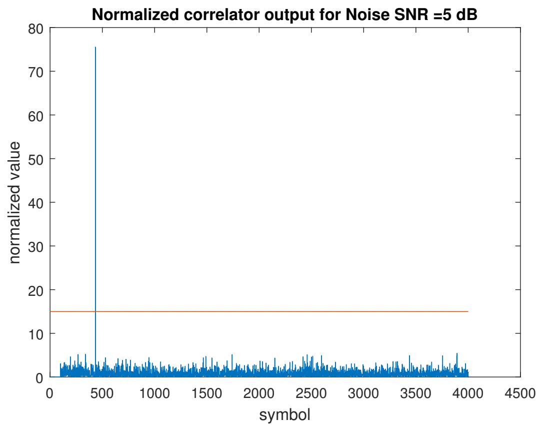
(a)



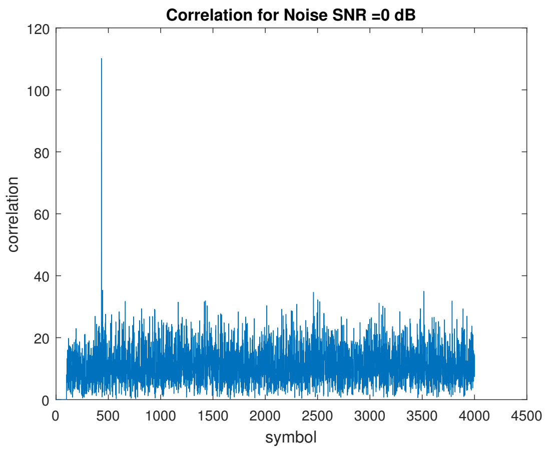
(b)



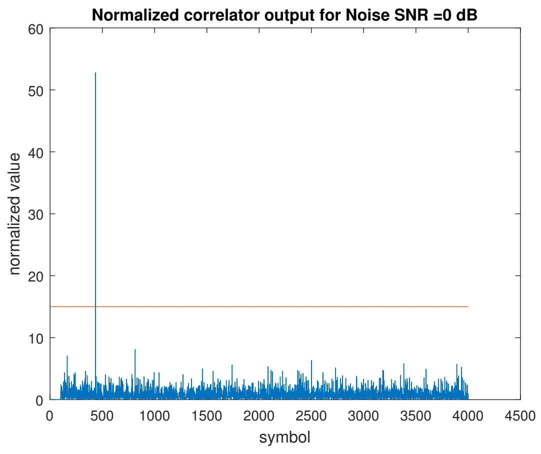
(c)



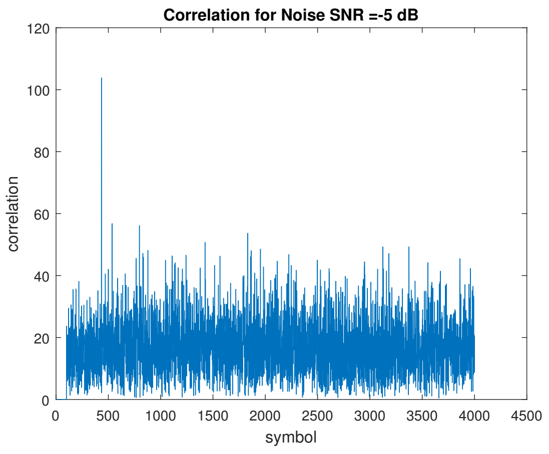
(d)



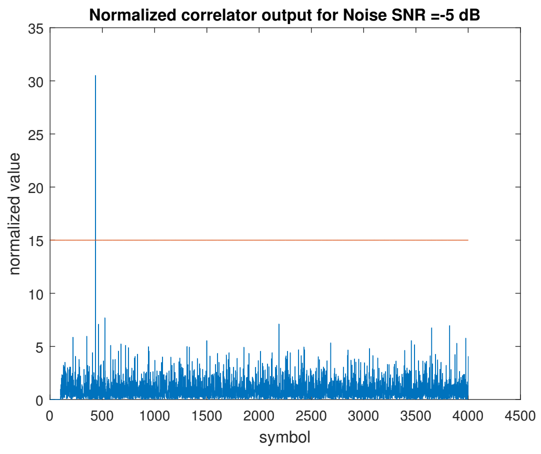
(e)



(f)



(g)



(h)

Figure 1: Correlation output

A decent treshold value is 15. As indicated in the plot, after normalizing the noise causes peak correlating values lower than 10. The average normalized correlating value is almost independant of the SNR.

Exercise 3

See Matlab code in file *exercise3.m*

The images are shown in Figure 2 and 3. When the frame synchronization fails, the bits are 'shifted' and do no represent the grey value correctly.

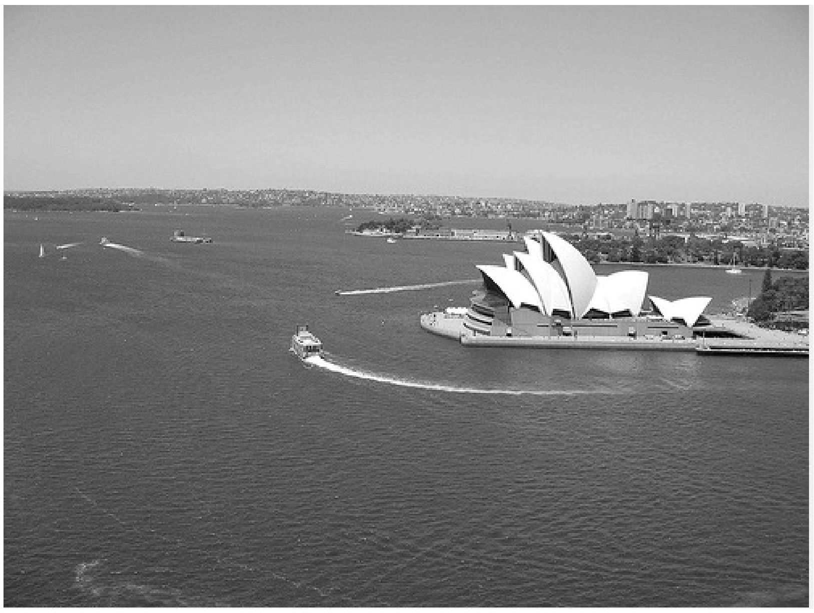


Figure 2: Correct image with threshold of 15

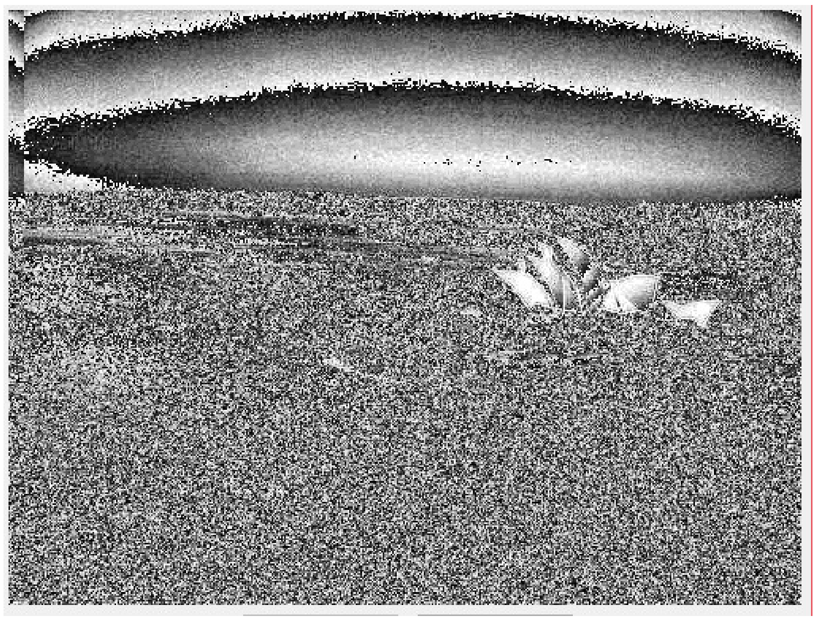
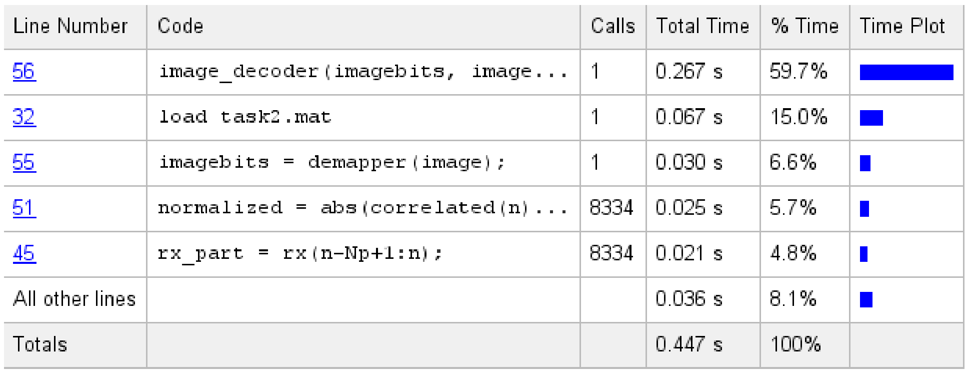


Figure 3: Bad image with threshold of 3

Exercise 4

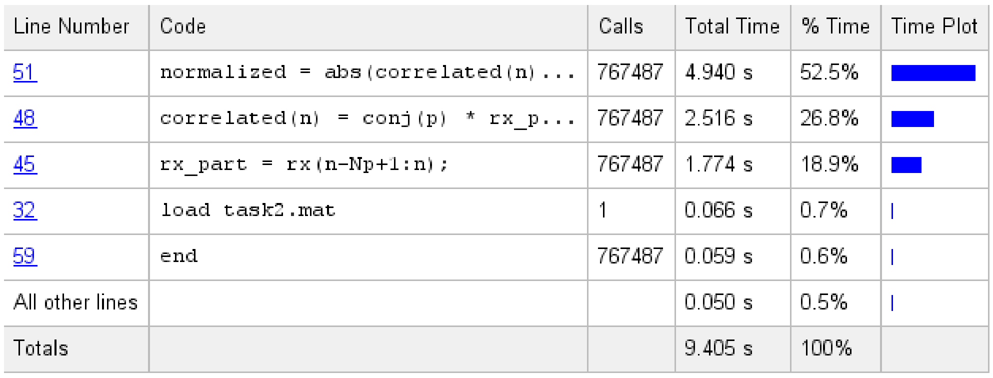
When running the file with a normal treshold (value 15), the processing stops when the preamble is detected. In this case the calculations done are very limited.

We can also loop over all values (for example, when choosing a very high treshold, so the preamble is never detected). In that case the normalizing takes the longest time, followed by the calculation of the correlation.
The results are in the *profile_results* directory.



Line Number	Code	Calls	Total Time	% Time	Time Plot
56	`image_decoder(imagebits, image...`	1	0.267 s	59.7%	▬
32	`load task2.mat`	1	0.067 s	15.0%	▬
55	`imagebits = demapper(image);`	1	0.030 s	6.6%	▬
51	`normalized = abs(correlated(n)...`	8334	0.025 s	5.7%	▬
45	`rx_part = rx(n-Np+1:n);`	8334	0.021 s	4.8%	▬
All other lines			0.036 s	8.1%	▬
Totals			0.447 s	100%	

Figure 4: Time analysis with treshold = 15, so processing stops when preamble is found



Line Number	Code	Calls	Total Time	% Time	Time Plot
51	`normalized = abs(correlated(n)...`	767487	4.940 s	52.5%	▬
48	`correlated(n) = conj(p) * rx_p...`	767487	2.516 s	26.8%	▬
45	`rx_part = rx(n-Np+1:n);`	767487	1.774 s	18.9%	▬
32	`load task2.mat`	1	0.066 s	0.7%	▬
59	`end`	767487	0.059 s	0.6%	▬
All other lines			0.050 s	0.5%	▬
Totals			9.405 s	100%	

Figure 5: Time analysis with high treshold, so all values are processed