Digital Communications and Laboratory
First Homework

Faccin Dario, Pegoraro Jacopo

Setup and presentation of the methods

Input signal

The random process to simulate is:

$$x(k) = e^{j(2\pi f_1 k + \phi_1)} + 0.8e^{j(2\pi f_2 k + \phi_2)} + w(k) \tag{1}$$

where $f_1 = 0.17$ and $f_2 = 0.78$ are the normalized frequencies of the exponential signals, ϕ_1 and ϕ_2 their initial phases considered as uniformly distributed in the interval $[0, 2\pi)$. The added noise $w(k)$ is a r.p. that follows a complex Gaussian distribution with zero mean and variance σ^2_{wa}. The simulation of this process has been carried out for $K = 800$ samples of a single realization. The sampling period T_c is assumed to be equal to 1.

Autocorrelation estimation

An unbiased estimate of the autocorrelation of the signal is provided by [1]:

$$\hat{\mathtt{r}}_x(n) = \frac{1}{K-n}\sum_{k=n}^{K-1} x(k)x^*(k-n), \text{ for } n = 0, 1, ..., K-1 \tag{2}$$

where K is the number of samples of the realization of $x(k)$.
A biased estimate of the autocorrelation is instead [1]:

$$\check{\mathtt{r}}_x(n) = \frac{1}{K}\sum_{k=n}^{K-1} x(k)x^*(k-n) = \Big(1 - \frac{|n|}{K}\Big)\hat{\mathtt{r}}_x(n) \tag{3}$$

The variance of the estimate gets larger and larger as n approaches K. For this reason the number of samples that provide a reliable estimate ($= L$) is much lower than the length of $x(k)$. In the following analysis we will state in each method the value of L used.

Blackman and Tukey Correlogram

This estimator uses the autocorrelation unbiased estimation $\{\hat{\mathtt{r}}_x(n)\}, n = -L, \ldots, L$.
Since the autocorrelation estimate is unbiased, also the estimator is unbiased.

$$\mathcal{P}_{BT}(f) = T_c \sum_{n=-L}^{L} \mathtt{w}(n)\hat{\mathtt{r}}_x(n)e^{-j2\pi f n T_c} \tag{4}$$

where $\mathtt{w}$ is a window of length $2L + 1$.

Periodogram

An estimate of the statistical power of $\{x(k)\}$ is given by

$$\begin{aligned} \hat{M}_x &= \frac{1}{K}\sum_{k=0}^{K-1} |x(k)|^2 \\ &= \frac{1}{KT_c}\int_{-\frac{1}{2T_c}}^{\frac{1}{2T_c}} |\tilde{\mathcal{X}}(f)|^2 df \end{aligned} \tag{5}$$

An estimator of the PSD is given by

$$\mathcal{P}_{PER}(f) = \frac{1}{KT_c}|\tilde{\mathcal{X}}(f)|^2 = T_c \sum_{n=-(K-1)}^{K-1} \check{\mathtt{r}}_x(n) e^{j2\pi f n T_c} \tag{6}$$

This method is related to the biased estimator of the autocorrelation presented in the previous section, and is therefore affected by a BIAS. This estimator also presents a very large variance due to the fact that samples of the autocorrelation up to $K-1$ are used (here $L = 800$). To compute the Fourier transform, we use `fft` function of MATLAB.

Welch Periodogram

The main idea behind the Welch periodogram is to compute periodograms over different windows of the input signal and to average them.
Given an input signal of K samples, different subsequences of consecutive D samples are extracted. Notice that two following subsequences, $x^{(s)}$ and $x^{(s+1)}$, may overlap by S samples. The number of subsequences is

$$N_s = \left\lfloor \frac{K-D}{D-S} + 1 \right\rfloor \tag{7}$$

The Welch periodogram is computed as

$$\mathcal{P}_{WE}(f) = \frac{1}{N_s} \sum_{s=0}^{N_s-1} \mathcal{P}^{(s)}_{PER}(f) \tag{8}$$

where $\mathcal{P}^{(s)}_{PER}(f)$ is the periodogram computed for $x^{(s)}$ as (6) using a window of length D that can be for example a rectangular window or an Hamming window.

AR Model

One last method for estimating the PSD of a signal is using an AR model of order N to describe the process.
In this model, the process is assumed to be generated by

$$x(k) = -\sum_{n=1}^{N} a_n x(k-n) + w(k) \tag{9}$$

where w is supposed to be white noise with variance σ_w^2.
This equation is equivalent to the scheme in Figure 1, where w is filtered by an FIR filter with transfer function $H_{AR}(z) = A^{-1}(z)$, where $A(z) = 1 + \sum_{n=1}^{N} a_n z^{-n}$.

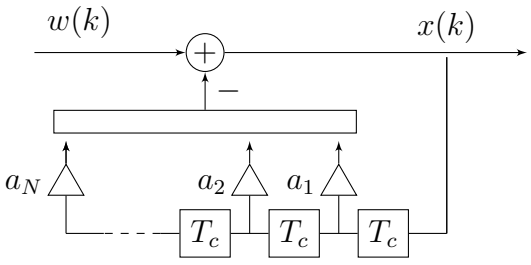

Figure 1. AR model

The PSD of x, given by the Fourier transform of the autocorrelation sequence, is:

$$\mathcal{P}_x(f) = P_z(e^{j2\pi f T_c}) = \frac{T_c \sigma_w^2}{|\mathcal{A}(f)|^2} \tag{10}$$

The coefficients $a_1, a_2, ..., a_N$ can be computed using the Yule-Walker equations. Given the autocorrelation matrix

$$\mathbf{R} = \begin{bmatrix} \mathtt{r}_x(0) & \mathtt{r}_x(-1) & \cdots & \mathtt{r}_x(-N+1) \\ \mathtt{r}_x(1) & \mathtt{r}_z(0) & \cdots & \mathtt{r}_x(-N+2) \\ \vdots & \vdots & \vdots & \vdots \\ \mathtt{r}_x(N-1) & \mathtt{r}_x(N-2) & \cdots & \mathtt{r}_x(0) \end{bmatrix} \tag{11}$$

and $\mathbf{r} = [\mathtt{r}_x(1), \mathtt{r}_x(2), \mathtt{r}_x(N)]^T$, the vector $\mathbf{a}$ of the coefficients of the AR model is given by

$$\mathbf{R}\,\mathbf{a} = -\mathbf{r} \tag{12}$$

If $\mathbf{R}$ admits inverse (matrix is not ill conditioned), we obtain

$$\mathbf{a} = -\mathbf{R}^{-1}\,\mathbf{r} \tag{13}$$

The variance σ_w^2 of the white noise $w(k)$ is then

$$\sigma_w^2 = \mathtt{r}_x(0) + \mathbf{r}^H \mathbf{a} \tag{14}$$

In this work however the autocorrelation of the process is not known because we have access to only one realization. For this reason the autocorrelation matrix will be built using the unbiased estimate $\hat{\mathtt{r}}_x(n)$ defined as in equation (2).
It's important to not confuse the added noise in the original r.p., with variance σ_{wa}^2, with the variance of the noise that is used as input for the AR model that has variance σ_w^2.

Problem 1

Here $\sigma_{wa}^2 = 2$. The parameters used for the different methods are listed below:

- **Periodogram** The estimate is computed using equation (6). In this method the samples of the autocorrelation are used up to $K-1$ so the variance of the estimator will be very high [1].
- **Correlogram** For this estimator (see equation (4)) we used an Hamming window and $L = \frac{K}{5} = 160$ to reduce the variance of the autocorrelation estimate (the unbiased one: (2)). The choice of the Hamming window is based on the outcome of the estimate and on the consideration that the two carrier frequencies are well separated so a rectangular window would perform worse. The expression for the Hamming window is:

$$w_H(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{2L}\right) \tag{15}$$

- **Welch periodogram** The estimate is based on equations (7-8). In this case we used a rectangular window of size $D = 120$ samples and overlap of size $S = 60$ samples. This decision is due to the necessity of reducing the variance of the estimator (that is done by taking more sequences) and at the same time obtain a good evaluation of the peaks (that means increasing D).

- **AR model** The AR model (see (9-13-14)) with this value of the variance of the error requires a high order to estimate correctly the PSD. To correctly choose the order, we can plot the value of the variance of the input white process σ_w^2 versus N and choose the value that represents a good trade-off between complexity and low variance. This is indeed a good method because if we recall the relation between the AR model of a process and the optimal linear predictor for the same process we can interpret σ_w^2 as the error made by the predictor, J_{min}. However in this case J_{min} shows a decreasing behavior without flattening even at high values of N so the decision is not easily determined. We chose $N = 25$ because, without being too high, it provided a good representation of the expected peaks in the PSD estimate.

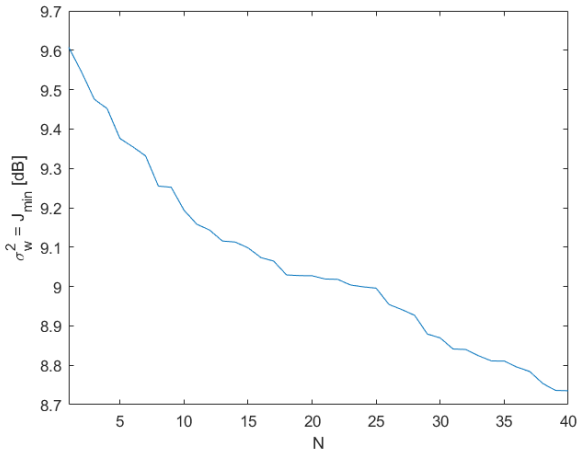

Figure 2. J_{min} over N, $\sigma_{wa}^2 = 2$

In Figure 3 it is shown the power spectral density estimation along with the analytical PSD.

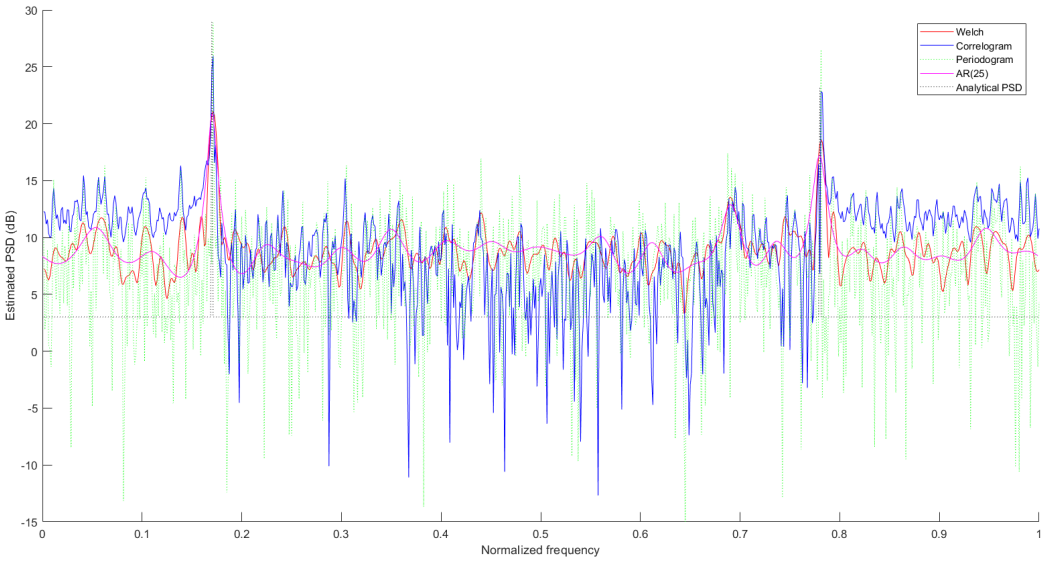

Figure 3. Estimated PSD, $\sigma_{wa}^2 = 2$

Problem 2

Here $\sigma_{wa}^2 = 0.1$

- **Periodogram** Equation (6). In this method the samples of the autocorrelation are used up to $K-1$ so the variance of the estimator will be very high [1].
- **Correlogram** For this estimator (see equation (4)) we used $L = \frac{K}{5} = 160$ to reduce the variance of the autocorrelation estimate (unbiased (2)). This time the window used was a Blackman-Harris window, because it led to less oscillations than the Hamming window:

$$w_{BH}(n) = 0.35875 - 0.48829\cos\left(\frac{2\pi n}{2L}\right) + 0.14128\cos\left(\frac{4\pi n}{2L}\right) - 0.01168\cos\left(\frac{6\pi n}{2L}\right) \quad (16)$$

- **Welch periodogram** Equation (7-8). In this case we used a Hamming window of size $D = 160$ samples and overlap of size $S = 80$ samples. This decision is due to the necessity of reducing the variance of the estimator (that is done by taking more sequences) and at the same time obtain a good evaluation of the peaks (that means increasing D). This time an Hamming window was chosen because a rectangular window would lead to much higher side peaks.
- **AR model** To choose the parameter N we exploited the relation between the AR model and the linear predictor as in exercise 1, plotting $\sigma_w^2 = J_{min}$ versus N.

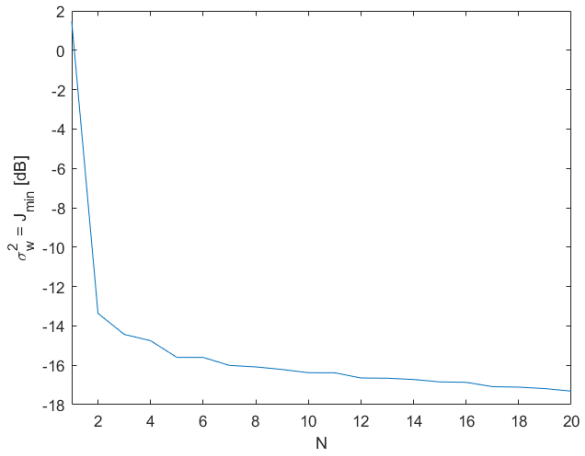

Figure 4. J_{min} over N, $\sigma_{wa}^2 = 0.1$

From Figure 4, it can be seen that a suitable choice is $N = 2$, because it corresponds to the knee of the curve. An higher order could increase the precision of the predictor, for example $N = 5$ would lead to an improvement of about 2 dB. However, the major improvement is gained by going from $N = 1$ to $N = 2$, and not increasing the complexity too much is a key aspect when we have to run an algorithm to estimate the coefficients of the predictor as in exercise 4, because a high order could cause convergence problems. For this reason we choose $N = 2$.
Using this value the autocorrelation matrix $\mathbf{R}$ is indeed well conditioned, hence the solution is acceptable.

In Figure 5 is shown the power spectral density estimation along with the analytical PSD.

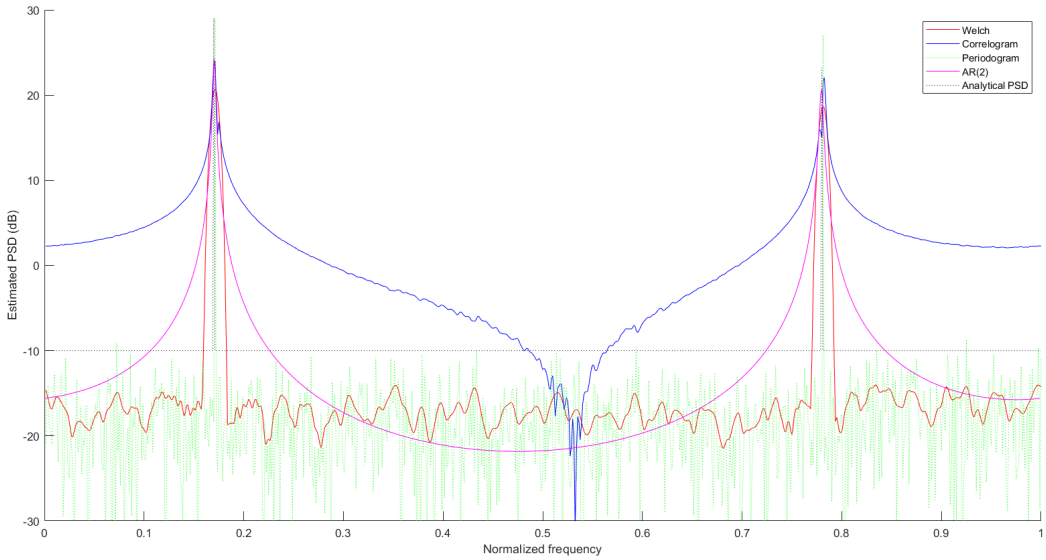

Figure 5. Estimated PSD, $\sigma^2_{wa} = 0.1$

Problem 3

The coefficients for the optimal error predictor are given by $\mathbf{c} = -\mathbf{a}$, where the vector $\mathbf{a}$ is given by the coefficients of the AR model. As already introduced in the previous section, exploiting the relation between the coefficients of the predictor and of the AR model one can decide the best value for the order N of both models just by plotting the error of the predictor J_{min} versus N (see again figure 4.)

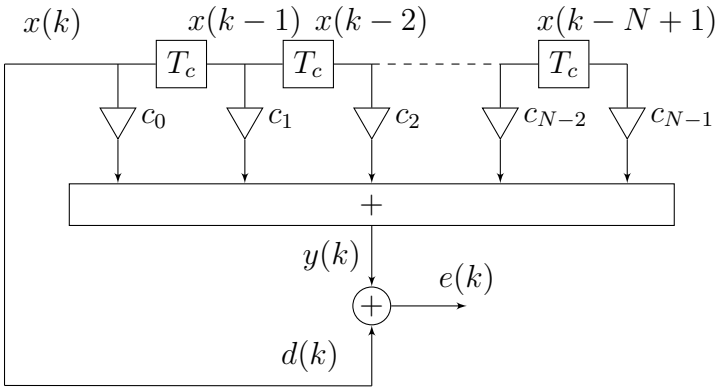

Figure 6. Wiener filter

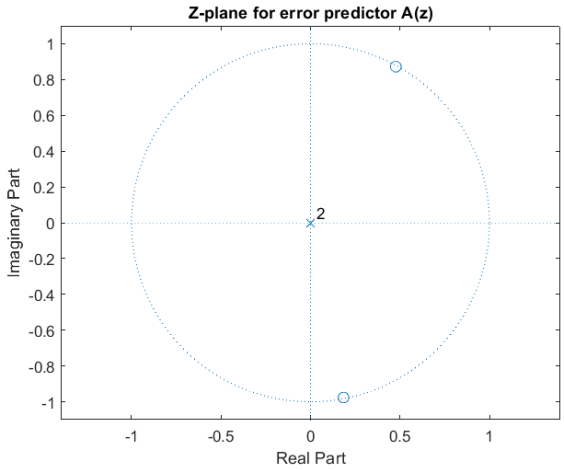

Figure 7. Z-plane for the error predictor A(z)

Knowing the coefficients of the best predictor we can write the extended expression of its transfer function $A(z)$. The presence of "discrete" components in the spectrum can be detected by looking at the zeros of $A(z)$ because they coincide with the poles of the transfer function of the AR filter: poles near the unit circle result in peaks in the frequency response $1/A(f)$ and so also peaks in the PSD estimate obtained with the AR model (see equation (11)). In our specific case we can see in figure 7 that we have the two zeros of $A(z)$ near the unit circle and by computing their frequencies we can infer the frequencies of the discrete components in the spectrum of the signal $x(k)$. To obtain the correct numerical value we need to normalize the frequencies over 2π. In Table 1 the zeros of the transfer function $A(z)$ are reported, along with their magnitude and phase, proving that up to a certain level of precision we can compute f_1 and f_2 from the transfer function of the predictor without using the Fourier transform of the signal.

z	$\Re[z]$	$\Im[z]$	$\lvert z\rvert$	f
z_1	0.4725	0.8718	0.9935	**0.1704**
z_2	0.1826	-0.9729	0.9899	**0.7795**

Table 1. Zeros of A(z)

Problem 4

Least Mean-Square algorithm

Given a signal $x(k)$, wide sense stationary, with zero mean, let $\mathbf{x}^T(k-1)$ be the vector $[x(k-1), x(k-2), \ldots, x(k-N)]$. The one-step predictor of order N tries to estimate the value of $x(k)$ given $\mathbf{x}^T(k-1)$.
This problem can be solved considering $\mathbf{x}^T(k-1)$ the input of a Wiener filter of order N and $x(k)$ the reference signal. Then the Wiener-Hopf equation computes the optimal coefficients of the filter with

$$\mathbf{R}\mathbf{c}_{opt} = \mathbf{r} \tag{17}$$

The LMS algorithm is a version of the steepest descent algorithm which provides an iterative method to approximate the optimal Wiener-Hopf solution, without knowing the autocorrelation matrix $\mathbf{R}$ and the vector $\mathbf{r}$.

The LMS algorithm updates the coefficients of the Wiener filter at each iteration k with the equation

$$\mathbf{c}(k+1) = \mathbf{c}(k) + \mu e(k)\mathbf{x}^*(k-1) \tag{18}$$

where $e(k)$ is the estimation error between the reference signal $x(k)$ and the filter prediction $y(k)$.
Besides the filter order N, LMS relies on the update coefficient: $\mu = \frac{\tilde{\mu}}{N r_x(0)}$.
For convergence, it must hold $0 < \tilde{\mu} < 2$.
At each step, the algorithm computes $y(k) = \mathbf{x}^T(k-1)\mathbf{c}(k)$ and $e(k)$, then it updates the coefficients using Equation (18).

Implementation and results

At the beginning, we decided to initialize the Wiener filter coefficients to zero and the input signal before $k = 0$ to zero: $\mathbf{c}(0) = \mathbf{0}$, $x(k) = 0, \forall\, k < 0$.
Applying the algorithm on a single realization of the process we obtained a grassy behavior of the coefficients and the error.
Using $k_{max} = 800$ iterations and update coefficient $\tilde{\mu} = 0.05$ resulted in convergence before reaching k_{max}.
Then we applied the LMS over 300 different realizations of the signal and took the average value for coefficients and error for each iteration. The behavior is much smoother than in the previous analysis, especially for $|e(k)|^2$.
In Table 2 are reported the coefficients of the predictor at convergence, their optimal value, computed using (12) and their percentage difference.

Coefficient	$\Re[c_{predictor}]$	$\Re[c_{opt}]$	Difference (%)	$\Im[c_{predictor}]$	$\Im[c_{opt}]$	Difference (%)
c_1	0.6639	0.6590	0.07	-0.0969	-0.1011	4.33
c_2	-0.9388	-0.9351	0.04	0.3057	0.3044	0.04

Table 2. Prediction and optimal coefficients

In Figures 8 and 9 it is shown the behavior of the coefficients c_i, $i = 1, 2$, their mean across 300 realizations (as k increases) and the optimal solution by the Wiener-Hopf equation. We can see that the convergence is very good in all cases, only the imaginary part of coefficient c_1 is a bit worse than the others.
In Figure 10 it is shown the behavior of the cost function $J(k) = E\left[|e(k)|^2\right]$ across 300 realizations, the values of $|e(k)|^2$ obtained with the LMS algorithm and the solution of the direct method J_{min}, as the number of iterations k increases. We can notice that, according to the theoretical results, the mean squared error $J(k)$ presents an *excess noise* $J_{ex}(\infty)$ that makes it not perfectly converging to the analytical value J_{min} (the difference is small but visible in Figure 10). To reduce this excess value we could choose a smaller value of $\tilde{\mu}$, but this would result in a slower convergence of the coefficients that we want to avoid having at our disposal only a limited number of samples of the signal.

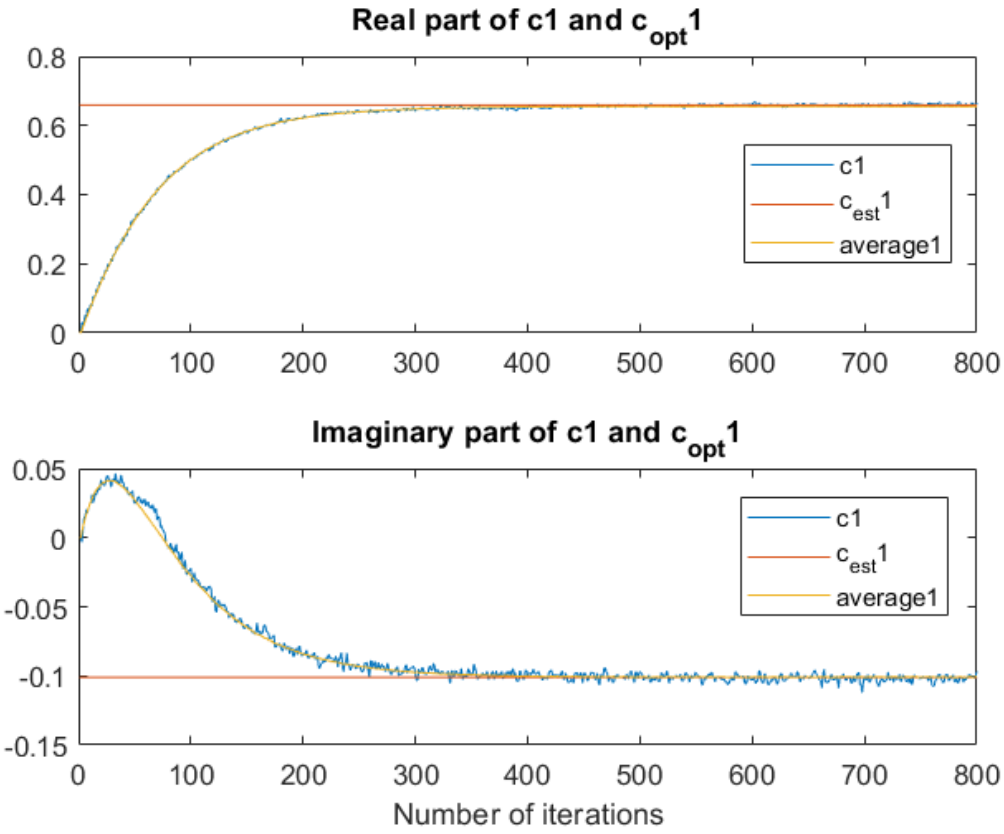

Figure 8. Real and imaginary part of c_1

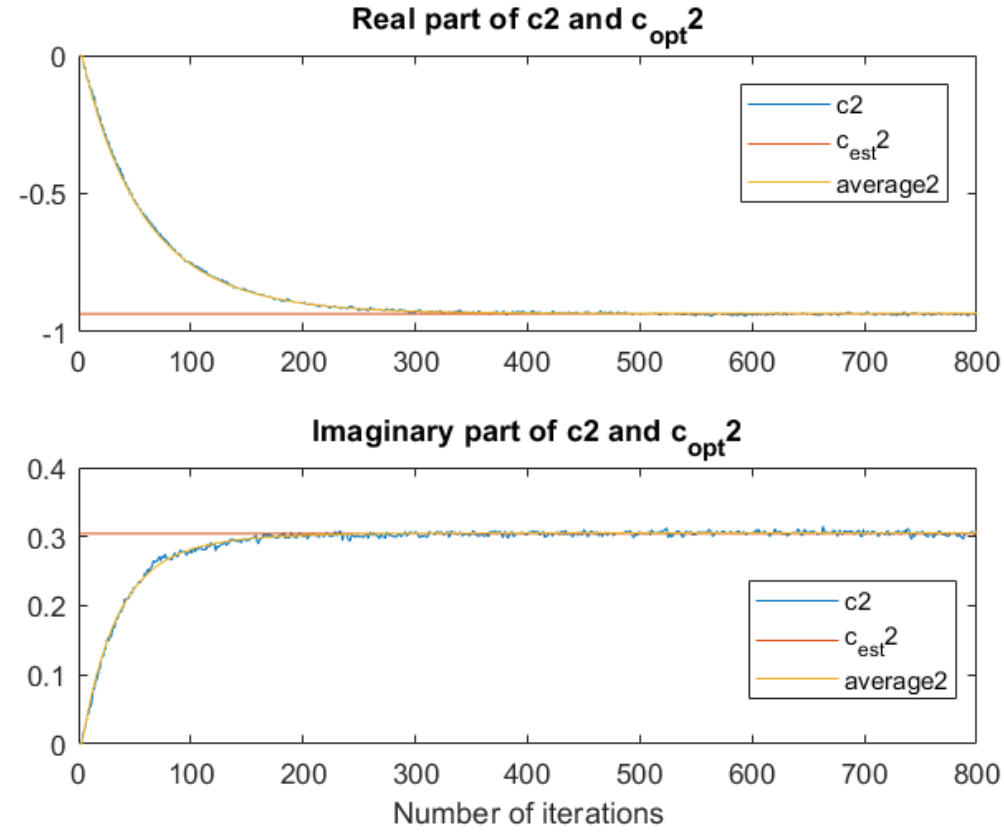

Figure 9. Real and imaginary part of c_2

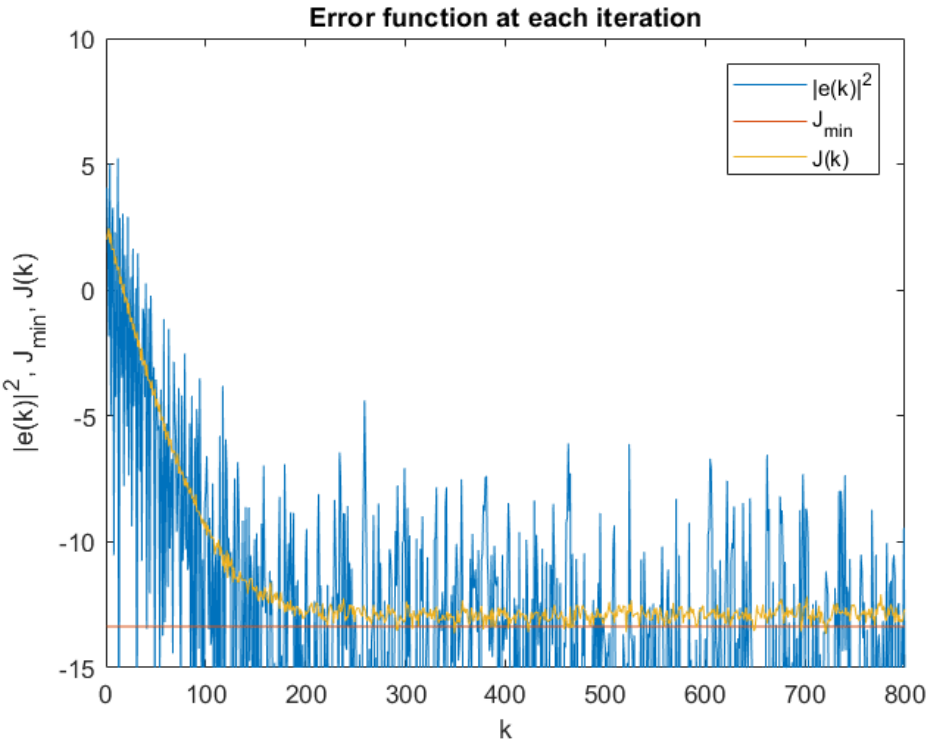

Figure 10. Mean squared error obtained with LMS (blue), by averaging over 300 realizations (yellow) and with the direct method (red).

Bibliography

[1] Nevio Benvenuto, Giovanni Cherubini, *Algorithms for Communication Systems and their Applications*. Wiley, 2002.