Digital Communications and Laboratory
First Homework

Faccin Dario, Pegoraro Jacopo

Problem 1

The random process to simulate is:

$$x(k) = e^{j2\pi f_1 k + \phi_1} + 0.8e^{j2\pi f_2 k + \phi_2} + w(k) \tag{1}$$

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_w^2. The simulation of this process has been carried out for $K = 800$ samples of a single realization.

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.

Periodogram

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

$$\begin{aligned} \hat{M}_x &= \frac{1}{K} \sum_{k=0}^{K-1} |x(k)|^2 \\ &= \frac{1}{KT_c} \int_{-\frac{1}{2T_c}}^{\frac{1}{2T_c}} |\tilde{\mathcal{X}}(f)|^2 df \end{aligned} \tag{4}$$

An estimator of the PSD is given by

$$\mathcal{P}_{PER}(f) = \frac{1}{KT_c} |\tilde{\mathcal{X}}(f)|^2 = T_c \sum_{n=-(K-1)}^{K-1} \check{\mathtt{r}}_x(n) e^{j2\pi f n T_c} \tag{5}$$

This method is related to the biased estimator of the autocorrelation presented in the previous section, and is therefore affected by a BIAS. It 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

$$N_s = \left\lfloor \frac{K-D}{D-S} + 1 \right\rfloor \tag{6}$$

The Welch periodogram is computed as

$$\mathcal{P}_{WE}(f) = \frac{1}{N_s} \sum_{s=0}^{N_s-1} \mathcal{P}_{PER}^{(s)}(f) \tag{7}$$

where $\mathcal{P}_{PER}^{(s)}(f)$ is the periodogram computed for $x^{(s)}$ as (5) using an Hamming window of size $D = 70$ samples and overlap of size $S = 35$ samples.

Blackman and Tukey Correlogram

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

$$\mathcal{P}_{BT}(f) = T_c \sum_{n=-L}^{L} \mathtt{w}(n)\hat{\mathtt{r}}_x(n)e^{-j2\pi f n T_c} \tag{8}$$

where $\mathtt{w}$ is a window of length $2L+1$. For this estimator we used an Hamming window and $L = \frac{K}{5} = 160$ to reduce the variance of the autocorrelation estimate.

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 $A(z) = 1 + \sum_{n=1}^{N} a_n z^{-n}$.

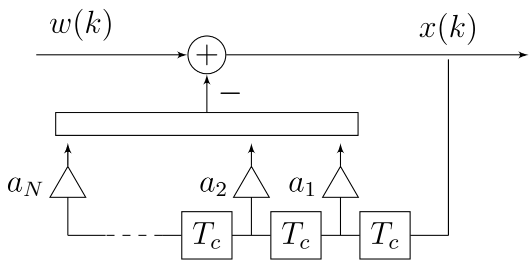


Figure 1. AR model

The z-transform of the autocorrelation sequence of x is given by

$$P_x(z) = \frac{\sigma_w^2}{A(z)A^*\left(\frac{1}{z^*}\right)} \tag{10}$$

hence the PSD of x is the Fourier transform of $P_x(z)$:

$$\mathcal{P}_x(f) = P_z(e^{j2\pi f T_c}) = \frac{T_c \sigma_w^2}{|\mathcal{A}(f)|^2} \tag{11}$$

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

$$\mathbf{R} = \begin{bmatrix} \mathtt{r}_x(0) & \mathtt{r}_x(-1) & \cdots & \mathtt{r}_x(-N+1) \\ \mathtt{r}_x(1) & \mathtt{r}_z(0) & \cdots & \mathtt{r}_x(-N+2) \\ \vdots & \vdots & \vdots & \vdots \\ \mathtt{r}_x(N-1) & \mathtt{r}_x(N-2) & \cdots & \mathtt{r}_x(0) \end{bmatrix} \tag{12}$$

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

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

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

$$\mathbf{a} = -\mathbf{R}^{-1}\,\mathbf{r} \tag{14}$$

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

$$\sigma_w^2 = \mathtt{r}_x(0) + \mathbf{r}^H\mathbf{a} \tag{15}$$

In our homework, we checked that the autocorrelation matrix $\mathbf{R}$ is indeed well conditioned, hence the solution does make sense.

Problem 2

Problem 3

Optimal predictor

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 we can see in Figure 3, the presence of zeros close the unit circle is a rough indicator of "discrete" frequency components.
The phase (normalized over 2π) of those zeros does determine the frequency of the discrete component.
In Table 1 the zeros of the transfer function $A(z)$ are reported, along with their magnitude and phase.

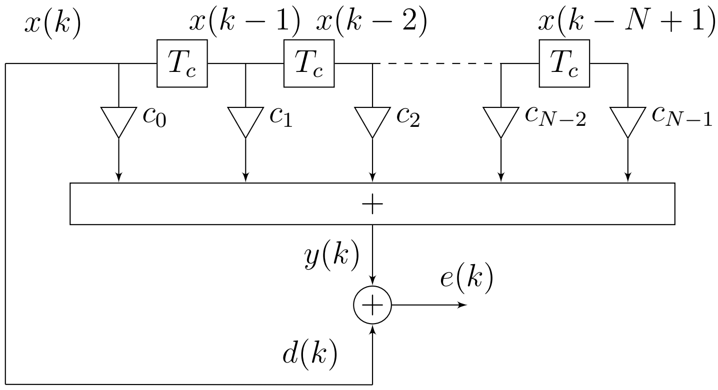


Figure 2. Wiener filter

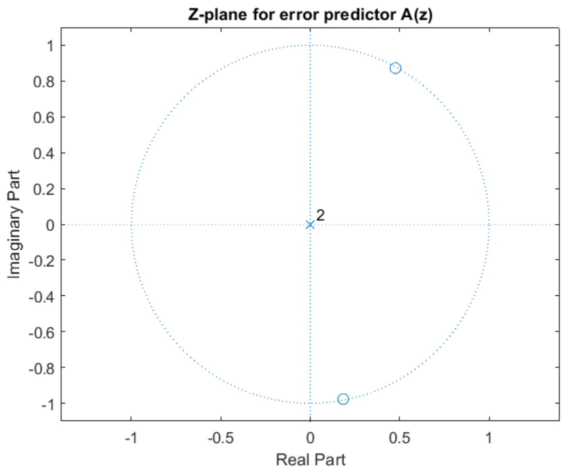


Figure 3. Z-plane for the error predictor A(z)

z	$\Re[z]$	$\Im[z]$	$\lvert z\rvert$	f
z_1	0.4766	0.8718	0.9936	0.1704
z_2	0.1845	-0.9762	0.9936	0.7797

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

$$\mathbf{R}\mathbf{c}_{opt} = \mathbf{r} \tag{16}$$

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

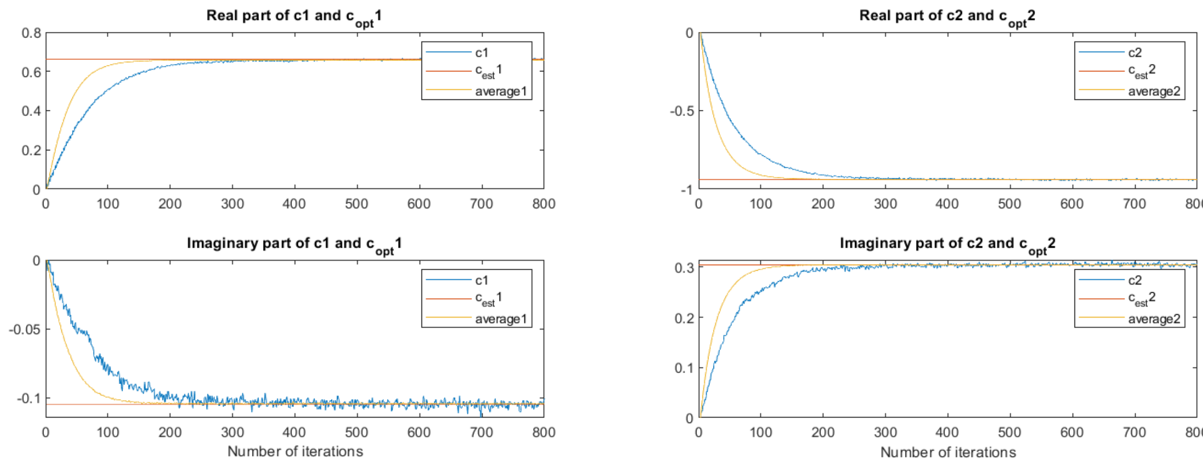
equation

$$\mathbf{c}(k+1) = \mathbf{c}(k) + \mu e(k)\mathbf{x}^*(k-1) \tag{17}$$

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 (17).

Implementation and results

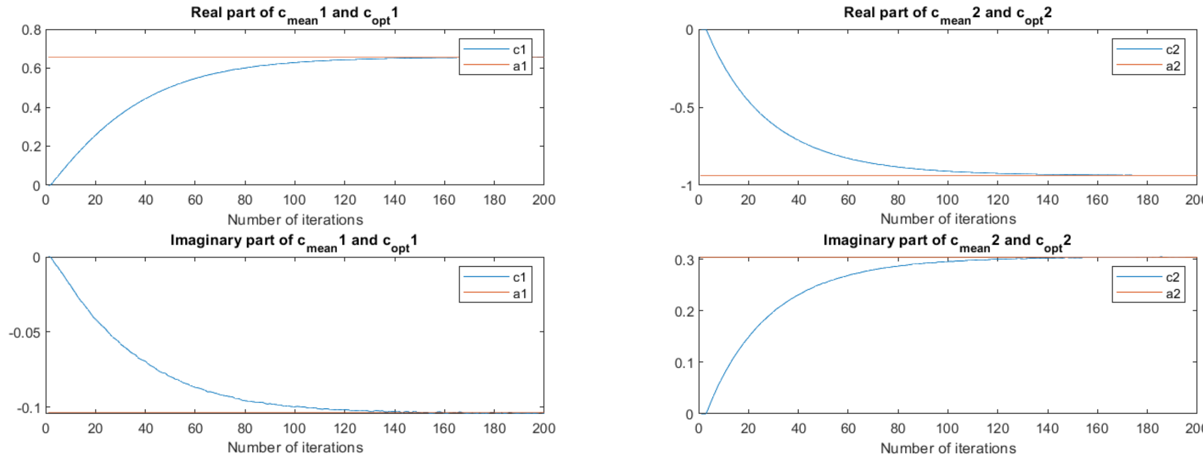
At the beginning, we decided to initialize the Wiener filter coefficients to zero and the input signal before $k = 0$ to zero: $\mathbf{c}(0) = \mathbf{0}$, $x(k) = 0, \forall\, k < 0$.
Applying the algorithm on a single realization of the process we obtained a grassy behavior of the coefficients and the error.
Using $k_{max} = 800$ and update coefficient $\tilde{\mu} = 0.05$ resulted in convergence.
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 Figure 5 it is shown the behavior of coefficients c_i, $i = 1, 2$, their mean and the mean across 300 realizations, as the number of iterations k increases.
In Figure 6 it is shown the behavior of the cost function $J(k) = E\left[|e(k)|^2\right]$ and its optimal solution J_{min} across 300 realizations, as the number of iterations k increases.



(a) Real and imaginary part of c_1

(b) Real and imaginary part of c_2

Figure 4. Coefficients for one realization



(a) Real and imaginary part of c_1 **(b)** Real and imaginary part of c_2

Figure 5. Coefficients by averaging over 300 realizations

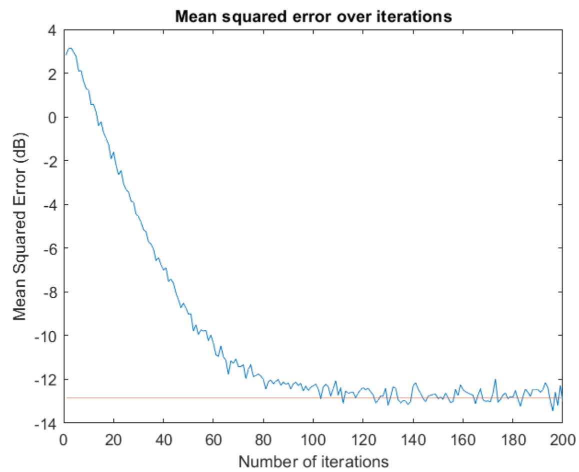


Figure 6. Mean squared error by averaging over 300 realizations

Bibliography

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