521324S SSP II — Matlab Simulation Exercise Lab 1

Lucas Ribeiro, lucas.ribeiro@oulu.fi
Centre for Wireless Communications, University of Oulu

Optimal Wiener Filter (6 pts)

Deadline: January 24th, 2024

You are given a communications system given in Figure 1. A data signal $d(n)$, with zero-mean white noise of variance $\frac{11}{18}$, i.e., $d(n) \sim \mathcal{N}(0, \frac{11}{18})$, is passed through a communication channel described by third-order AR process. The output signal is

$$x(n) = \frac{1}{9}x(n-1) + \frac{1}{9}x(n-2) + \frac{4}{9}x(n-3) + d(n)$$

Signal $x(n)$ is corrupted by additive Gaussian noise $v(n)$ so that the output is $u(n) = x(n) + v(n)$. The variance of $v(n)$ is σ^2 and mean of $E[v(n)] = 0$, i.e., $v(n) \sim \mathcal{N}(0, \sigma^2)$. An L-tap Wiener filter is applied to $u(n)$ to estimate the desired signal $d(n)$.

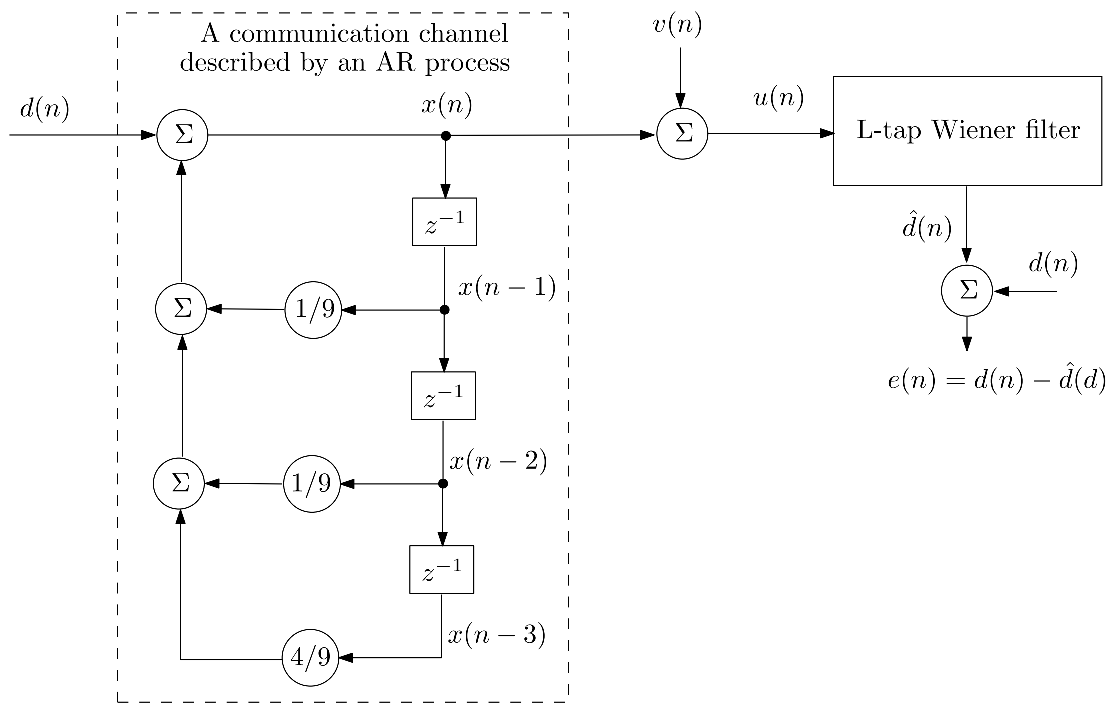

Fig. 1: Communications system based on an AR process.

Tasks: You are given an incomplete Matlab script. Complete the Matlab routine to simulate the system:

1) Generate the signals $d(n)$, $x(n)$, $v(n)$, and $u(n)$. Plot figures of realizations of $d(n)$, $x(n)$, and $u(n)$.
2) Calculate the tap weights of the optimal Wiener filter for this system using theoretical equations, with $\sigma^2 = 0.1$ and $L = 3$. Calculate also the corresponding mean square error (MSE).
3) With the tap weights obtained from (2), write a Matlab code to perform the Wiener filter on $u(n)$ to estimate the desired response of $d(n)$. The estimate is denoted as $\hat{d}(n)$. Plot figures of the realization

of signal $\hat{d}(n)$ and compare it to that of $d(n)$. Compute the average MSE and compare with the value obtained by the theoretical equations.

4) Repeat the simulation to the case with different values of σ^2, i.e., $\sigma^2 = 0.01$ and $\sigma^2 = 0.5$. Compare with the results obtained above.
5) Repeat the simulation for a filter with $L = 4$ taps. Compare the results.

Hint:

- For filtering operation in general, use Matlab function: *filter()*. Type <*doc filter*> to see how the function works.
- For the Matlab simulation program development and debugging and to observe the signals, run the simulation with small numbers of samples, e.g., $N = 10$. To calculate the average MSE, you should run the simulation with large samples of the signals, e.g., $N = 10^5$.
- If you want, you can write the code to calculate the autocorrelation matrix, the cross-correlation vector, and the Wiener filter tap coefficient from $u(n)$, instead of doing the calculation manually.