SGN-21006 Advanced Signal Processing

Exercise 3: Optimal Wiener filters

Assistant: Petri Helin, petri.helin@tut.fi

Introduction

In this exercise, we consider an adaptive filtering application known as *channel equalization*. The idea is that when an (audio) signal is passed through a communication channel (such as a telephone line), it becomes distroted by the (unknown) impulse response of the channel. At the receiver end, it is desirable to receive a signal identical to the original signal. In order to achieve this, a channel equalizer is applied to the channel output. Its purpose is to invert the model of the channel. A block diagram of the situation is depicted in Fig. 1.

The channel equalizer is designed using the output of the channel $u(t)$ and a reference signal $d(t)$ that equals the original sent signal. In practice, it can be agreed between the transmitter and receiver that a certain signal that is known to both ends is being sent during a training phase. In a nonstationary environment, we would use an adaptive algorithm to design the filter. Adaptive algorithms will be covered later on the course. Here, we design a fixed optimal Wiener filter for the task. Your task is to estimate the statistics of the signals, namely the autocorrelation of $u(t)$ and and the cross-correlation between $d(t)$ and $u(t)$ and to utilize the Wiener-Hopf equations to find an estimate for the tap weight of the FIR Wiener filter.

Before starting the tasks given on the next page, you should download and unzip necessary Matlab files and audio signals from http://www.cs.tut.fi/~helinp/advsp/ex3.zip.

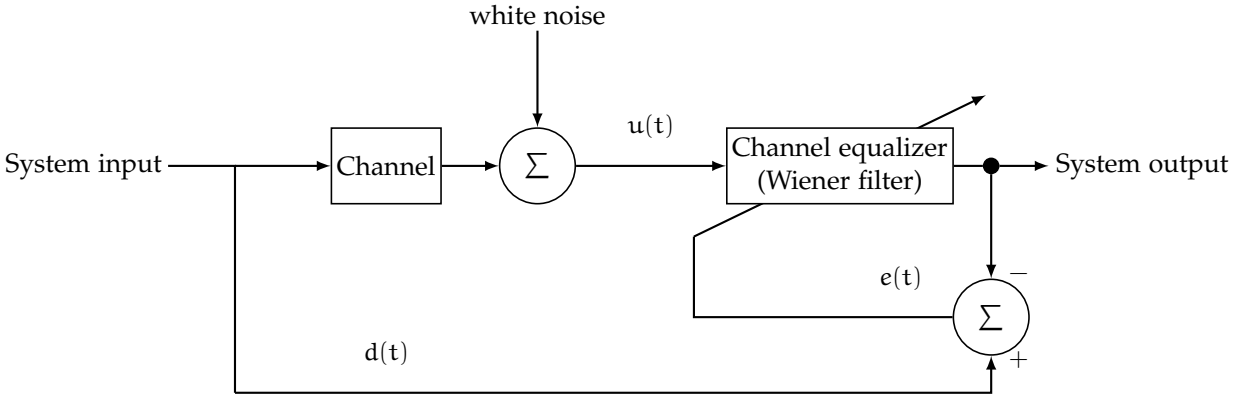

Figure 1: Channel equalization.

SGN-21006 Advanced Signal Processing
Exercise 3: Optimal Wiener filters

Tasks

Name:

Student number:

1. Open the Matlab file `wiener_filtering.m`. The code loads an audio sample that is treated as the input signal $d(t)$. The signal is passed through the channel using the function `simulate_channel.m`. Check the code and learn how to use the function.

a.) In order to compute the wiener filter, we need to estimate autocorrelation of the signal $u(t)$. Use the matlab function `xcorr` to compute a biased and an unbiased autocorrelation estimate. Implement also the unbiased version yourself (see e.g. page 5 of lecture 3). Plot the autocorrelation sequences for maximum lag value of 300.

b.) Try to take a shorter segment of the signal (equal to the maximum lag value for example) and see how the autocorrelation estimate changes. Explain the difference between the biased and the unbiased estimator. In what situations the biased estimator is preferable? Explain why one would be better than the other.

2. a.) Use `xcorr` to compute the cross-correlation between $d(t)$ and $u(t)$. Form the cross-correlation vector $\mathbf{p}$ and the matrix $\mathbf{R}$ (see e.g. page 10 of lecture 3). The structure of $\mathbf{R}$ is called Toeplitz. Use the accordingly named Matlab function.

b.) Solve the Wiener-Hopf equations $\mathbf{p} = \mathbf{R}\boldsymbol{w}$ for $\boldsymbol{w}$. Compare the result to the true $\boldsymbol{w}$ that is given as the output of `simulate_channel`.

c.) Use `filter` to filter $u(t)$ with the designed $\boldsymbol{w}$ and compute the error signal $e(t)$ and the mean square error. Check the estimation accuracy for different model orders (maximum lag value). What seems to be the optimal order after which the accuracy does not improve significantly? Why? Try increasing the amount of white noise in the channel using the input parameter of `simulate_channel`. What effect does it have?

d.) Check that the principle of orthogonality holds. (See page 7 of lecture 3). Comment the results. Keep in mind that an audio signal is presented with 16 bit integer values so the input data range is $[-2^{15}, 2^{15}]$.

3. There is another channel equalizer example at page 16 of lecture 3. The system input signal is generated there using a "useful signal generator". Implement the system in Matlab and solve the Wiener-Hopf equations numerically by estimating the autocorrelations and cross correlations. Check that you get the same result as in the slides.