Adaptive Equalizer Exercise

S-38.211 Signal Processing in Telecommunication I (2cr)

P. Karttunen

Helsinki University of Technology
Laboratory of Signal Processing and Computer Technology
Otakaari 5A, FIN-02015 Espoo, Finland

Phone: 09 451 2455
E-mail: Petri.Karttunen@hut.fi
Room: F302 (reception day: We 12-13)

TABLE OF CONTENTS

1. INTRODUCTION

This document gives the preliminary background information and guidelines for completing the exercise work of the course S-38.211 Signal Processing in Telecommunication I (2cr).

The exercise work is to be solved by using the MATLAB programming environment [1]. MATLAB is available for example on the computers at the Computing Center. In those machines MATLAB can be started by typing *use matlab* and then *matlab* at the UNIX prompt. The useful commands for getting more information online about the functions available at MATLAB are *lookfor <keyword>* and then *help <command>*. The prototype program code as well as this document and some other MATLAB related information can be found from the course home page at *http://keskus.hut.fi/opetus/s38211/project.*

This programming exercise is a compulsory part of passing the course. It will be rated in the scale 0-5 and it will contribute to the final grade of the course with the relative weight of 0.2. The deadline of this project work is due to **Friday 29.1.1999** at 15:30. All the documentation should be returned into the course locker (number 18) below the course information board. When returning this document remember to put your contact information, i.e., your name, study and group number and e-mail addresses in the cover page. If the document is returned after the deadline the grade for the document have to be decreased by one for each delayed day! In the case of further inquiries or problems with the exercise work you are encouraged to freely contact the assistant. The contact information you can find from the cover page of this document.

In the following sections the general system framework, the description of the prototype MATLAB program, a sample demo and the requirements for how the equalizer task should be accomplished are presented.

2. GENERAL SYSTEM FRAMEWORK

In this section, the general discrete-time transmission system is presented [2] (see the block diagram of Figure 1).

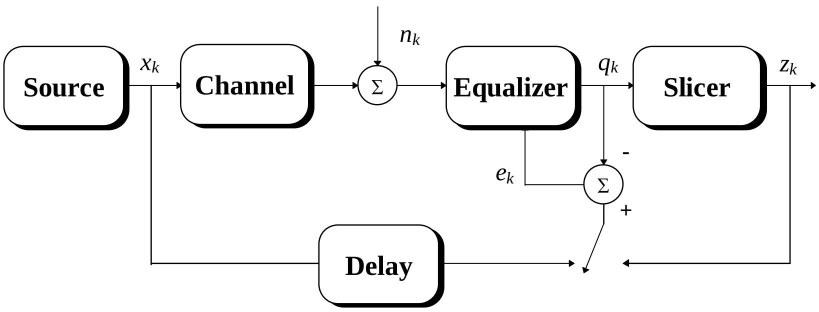

Figure 1 Adaptive equalizer in a chain of the transmission system

The *source block* produces transmitted Binary Phase Shift Keying (BPSK) symbols $x_k \in \pm 1$ (k=1, …, K) with the equal probability of 0.5. The total number of transmitted symbols is denoted as K. In the *channel block* some InterSymbol Interference (ISI) is introduced by

Finite Impulse Response (FIR) type channel model. The Z-domain transfer function of the channel $H(z)$ can be expressed as

$$H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \ldots + h_{L-1} z^{-(L-1)} \tag{1}$$

where h_l (l=0, …, L-1) are the values of the sampled impulse responses and L is the number of the channel filter coefficients. At the output of the channel, a noise sequence n_k is added. This noise is assumed to be Additive White Gaussian Noise (AWGN) with the variance σ_n^2. The sum of the channel output and the noise sequence is fed into the *equalizer block*. Finally, the equalizer output q_k is fed to the *slicer* to obtain the estimate z_k of the transmitted data symbol x_k.

The *equalizer block* performs the equalization of the channel. Different performance criteria can be utilized in the equalization [2]. In this exercise work we investigate the equalization by using the Mean Square Error (MSE) criterion. Here, the adaptive minimization of the cost function is realized by performing the weight vector search in the negative gradient direction of the cost function. The stochastic Least Mean Square (LMS) algorithm can be used for adaptively updating the filter coefficients [3]. The symbol-by-symbol update equation can be expressed as

$$\mathbf{c}_{k+1} = \mathbf{c}_k + 2\mu e_k \mathbf{x}_k \tag{2}$$

where $\mathbf{x}_k=[x_k\ x_{k-1}\ \ldots\ x_{k-(N-1)}]^T$ is the input vector, $\mathbf{c}_k$ is the weight vector, e_k is the error signal and μ is the step size. This step size μ controls the adaptation speed of the adaptive filter. The number of the adaptive filter coefficients has been denoted as N.

In order to implement the adaptive equalizer, we need to generate a reference signal for adaptive algorithm. For the initial adaptation of the filter coefficients we need to know the transmitted data symbols at the receiver. This known sequence is referred to as the training sequence. During the training period the desired signal is used as a reference signal and the error signal is defined as $e_k = x_{k-D} - q_k$ (see Figure 1). After training period the equalization can be performed in the decision directed manner. This mode can be utilized if the channel can be assumed to be time-variant. Therefore, it can be assumed that the decisions in the slicer output are right most of the time and the slicer decisions can be used as reference signal. In the decision directed mode, the error signal is defined as $e_k = z_k - q_k$ (see Figure 1). The Mean Square Error (MSE) for the filter in the kth time instant is defined as

$$MSE_k = \mathrm{E}\left[\left\|e_k\right\|_2^2\right] \tag{3}$$

In the simulation system this ensemble averaging for the wide sense stationary processes can be realized by the time averaging. Therefore, I independent runs is to be averaged in the simulation program.

3. MATLAB PROGRAM DESCRIPTION

In this section, some important parts of the sample MATLAB program are briefly presented from the programming point of view. The initialization part controls the system parameters and can be modified when inspecting their effect. These values presented here are the same as for the sample demo shown in the next section.

```
%%%---------------------------------------------------------------------------------------------
%%%                                Initialization
%%%---------------------------------------------------------------------------------------------
clear all;                         % Clear all the variables.
K=500;                             % Number of symbols.
I=200;                             % Number of independent runs.
N=11;                              % Number of filter coefficients.
MSE=0*ones(1,K);                   % MSE for adaptive filter.
sigma2_n=0.001;                    % Noise power.
hn=[0.2194 1.0000 0.2194];         % Channel impulse response.
my_max=1/(N*(sum(hn.^2)+sigma2_n));% Maximum step size (Eq 5).
my=my_max/2;                       % Step size parameter.
D=7;                               % Delay factor.
```

The transmitter part generates the data and reference sequences. It should be noted that the reference signal is delayed by just inserting necessary amount of zeros in the beginning of the data sequence.

```
%%%---------------------------------------------------------------------------------------------
%%%                                Transmitter
%%%---------------------------------------------------------------------------------------------
x=sign(randn(1,K));                % Generate BPSK data.
ref_x=[0*ones(1,D),x];             % Generate reference signal.
```

The channel parts introduce ISI according to FIR impulse response model. Also the white noise is added to the distorted signal.

```
%%%---------------------------------------------------------------------------------------------
%%%                                Channel
%%%---------------------------------------------------------------------------------------------
for i=1:I,                         % Run all the indepedent runs.
 ['Iteration:',mat2str(i)]         % Iteration index.
 channel_data=filter(hn,1,x);      % Distort the signal.
 n=sqrt(sigma2_n)*randn(1,K);      % Generate white noise sequence.
 channel_output=channel_data+n;    % AWGN channel.
```

This is the part where you can introduce your equalizer. The weight vector **c** can be initialized with zeros in the beginning of the LMS adaptation. After running through all the independent iterations the results can be analyzed.

```
%%%---------------------------------------------------------------------------------------------
%%%                                Equalizer
%%%---------------------------------------------------------------------------------------------
[c,e,q,z]=lms(channel_output,ref_x,my,N);% Linear LMS equalizer.
MSE=MSE+abs(e).^2;                 % Mean Square Error (MSE).
end
```

4. SAMPLE DEMO

In this section we introduce the sample demo for adaptive equalization. This demo example makes use of the similar channel model for equalization as described by Haykin [4] in his book in pages 412 - 421. The impulse response of the raised cosine type channel h_l (l=0, …, 2) is expressed as

$$h_l = \frac{1}{2}\left[1+\cos\left(\frac{2\pi}{W}(l-1)\right)\right] \tag{4}$$

where the parameter W controls amplitude distortion (W=2.9). The parameters of the sample system are described in Table 1 below. The value of the step size μ can be determined in the following way [4][5]. The step size μ should be selected smaller than the maximum step size μ_{max} in order to avoid the divergence problem. However, too small values results in slower convergence. The maximum value can be calculated from

$$\mu_{\max} = \frac{1}{\mathrm{Trace}[\mathbf{R}]} = \frac{1}{N(\sum_{l=0}^{L-1} h_l^2 + \sigma_{\mathrm{n}}^2)} \tag{5}$$

For the demo system, the step size $\mu=\mu_{max}/3$ was selected in order to get fast enough convergence.

Table 1 System parameters for the sample demo

Parameter	Value
Number of independent runs I	200
Number of transmitted symbols K	500
Number of filter coefficients N	11
Noise power σ_n^2	0.001
Step size μ ($\mu_{max}/3$)	0.0276
Channel coefficients [h_0 h_1 h_2]	[0.2194 1.0000 0.2194]

Figure 2 shows MSE for adaptive filter. This plot confirms that the equalizer actually converges close to the noise level. Figure 3 shows the impulse responses for the channel, the weight coefficients of the adaptive filter after the convergence, and the convoluted impulse response of the channel and equalizer. The impulse response of the channel is divided into the causal and non-causal parts and is symmetric about l=1. The equalizer response is also symmetric about n=8. As a rule of thumb, the delay factor D should be selected near the value $D=(N+L)/2=7$. Figure 4 shows the respective frequency responses. Flat frequency response confirms that the equalizer has enabled to equalize the channel. Figure 5 shows the discrete time signal representations for the data, for the data after the channel, for the data after the equalizer and for the slicer output data. The effect of the equalizer on the transmitted data can be concluded from these subplots.

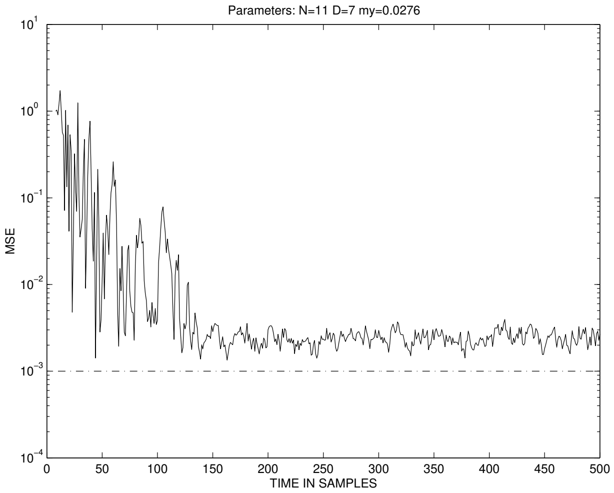

Figure 2 MSE as a function of time instant k

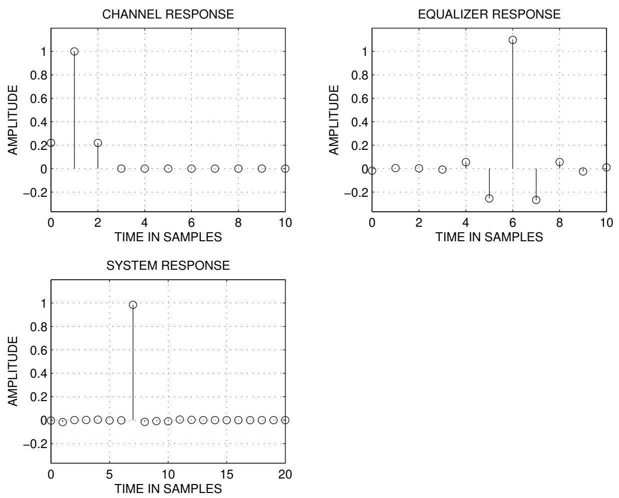

Figure 3 Impulse responses of the channel, the equalizer, and the channel plus equalizer

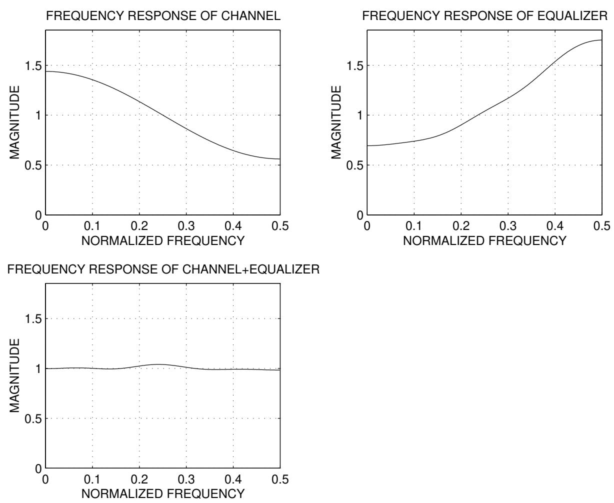

Figure 4 Frequency responses of the channel, the equalizer and the channel plus equalizer

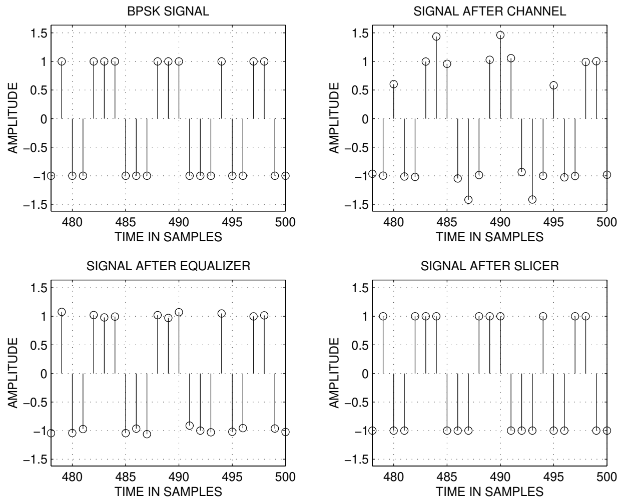

Figure 5 Discrete time signal for transmitted data, the data after the channel, the data after equalizer and the data at the slicer output

5. EXERCISE REQUIREMENTS

In this section the exercise tasks for each group are specified. It is recommended that the prototype MATLAB program will be used. You are not strictly restricted to this one so that you can freely modify it. Use the channel coefficients h_l (l=0, …, L-1) according to your group number as specified inTable 2 below.

Table 2 Coefficients of the channel model

Group #	h_0	h_1	h_2	h_3	h_4	h_5	h_6
1	0.2500	1.0000	0.2500	-	-	-	-
2	0.2798	1.0000	0.2798	-	-	-	-
3	0.3887	0.1883	1.0000	0.1883	0.3887	-	-
4	0.3365	1.0000	0.3365	-	-	-	-
5	0.3632	1.0000	0.3632	-	-	-	-
6	0.2500	0.2500	1.0000	0.2500	0.2500	-	-
7	0.3087	1.0000	0.3087	-	-	-	-
8	0.3149	0.2194	1.0000	0.2194	0.3149	-	-
9	0.5000	0.7500	0.0670	1.0000	0.0670	0.7500	0.5000
10	0.4709	0.1569	1.0000	0.1569	0.4709	-	-
11	0.6545	0.6545	0.0955	1.0000	0.0955	0.6545	0.6545
12	0.7840	0.5603	0.1257	1.0000	0.1257	0.5603	0.7840

Exercise specifications:

1. Implement a linear LMS based equalizer block by adding the necessary parts into the given MATLAB program.

2. Use N=11 filter coefficients and fix the noise variance to the small value of σ_n^2=0.001 and use high enough number of the symbols K. What is the largest step size μ_{max} which still guarantees the convergence of filter coefficients in your problem? Find the optimal

delay factor *D* for your system (function *grpdelay* might also be useful). Has the delay factor any influence on the results and how? Experiment with the step size μ. How the convergence is affected by the selection of the step size? How does the change in the number of filter coefficients *N* affect the results? How does the increase in the noise level affects the performance? Give MSE plots in each case. It is recommended when studying the effect of a parameter that the rest of the parameters are kept fixed. The MSE curves can be combined into a single plot for ease of comparison in each parameter study case.

3. Choose a step size smaller than μ_{max} (for example $\mu_{max}/5$) and use a large number of symbols to make sure that the equalizer has converged. Plot the impulse response of the channel, equalizer and convoluted response of the channel and equalizer by using, for example, 2*N* data points. Comment the results of each plot. Plot the frequency responses of the channel, equalizer and the channel plus equalizer. Comment the results in each plot. What kind of results should we get in the ideal case and explain why the equalizer can not always achieve these aims? Plot, for example, the last 2*N* symbols when the equalizer has converged for the transmitted data, the data after the channel, data after the equalizer and the data at the slicer output. What is the effect of the equalizer on the received symbols? How the effect of the delay factor *D* can be seen on these plots?

4. Now do some experiments by using the decision directed mode. How does the MSE behave when switched to the decision directed mode at different times before convergence? Give some illustrative MSE plots.

6. DOCUMENTATION

The project work should be documented. Explain in sufficient detail what you have done, how and why, so that the document shows that you have understood the problem. The document should also include the answers to the questions and the necessary simulation plots. Also, add the listing of your MATLAB program(s) for LMS equalizer with comments as part of your document in the Appendix. When documenting, you can use the format of this document for both the text part and plots. In order to import the current plot in the encapsulated postscript form from the MATLAB environment to the Word document the command *print -deps -tiff picture.eps* is recommended. The additional comments how the exercise work could be improved are welcome. Good luck!

REFERENCES

[1] MATHWORKS Inc, *MATLAB High-performance Numeric Computation and Visualization Software, User's Guide*, South Natick, MA, USA, 1994.

[2] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, Kluwer Academic Publishers, second edition, Boston, 1994.

[3] S. U. H. Qureshi, "Adaptive Equalization", *Proceedings of the IEEE*, Vol. 73, No. 9, Sep. 1995, pp. 1349-1287.

[4] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, third edition, New Jersey, 1996.

[5] P. S. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation.* Kluwer Academic Publishers, Boston, MA, USA, 1997.