ENEL 671 Adpative Signal Processing Project Report

Pouyan Keshavarzian, *Student Member, IEEE,*
pkeshava@ucalgary.ca
UCID 10053710
Electrical & Computer Engineering, University of Calgary

***Abstract*—The purpose of this project is to analyze the performance of three different adaptive algorithms. These algorithms, which can be used for a variety of applications such as echo cancellation and beamforming, are implemented for the purposes of equalizing a dispersive channel. The first algorithm investigated is the Least Mean Squares (LMS). The LMS filter uses a search method to minimize the mean square error. The performance of the LMS is then compared to the Recursive Least Squares (RLS) method, which minimizes a weighted linear least squares cost funtion. Finally the Recursive Least Squares Lattice algorithm is implemented in the third project for joint process estimation. The algorithms are tested against multiple channels with varying eigenvalue spreads. Furthermore different design parameters such as step size and filter order are evaluated. The performance with regards to convergence speed and complexity is discussed. The RLSL appears to have performance advantages in terms of convergence speed at the cost of significant increase in complexity.**

I. Introduction

ADAPTIVE filtering is used for a variety of applications. When communicating across a dispersive channel, a signal is destorted by effects of multipath, noise etc REFERENCE. To recover the correct message at the receiver, an adaptive filter is implemented to equalize the channel. In general, an algorithm is used to calculate the optimum filter coefficients to minimize the error between transmit and receive. In practice, a signal transmitted along a channel contains a preamble, which is a known sequence of data that is used to esimate the parameters of the channel. The statistics are assumed to be stationary in the window of time that each preamble is sent.

The most common adaptive equalizers involve the LMS, RLS, ans RLSL algorithms. The performance of these methods varies and they each have their advantages and drawbacks. The LMS is relatively simplistic in terms of hardware/software resources. However, it's drawback is that it uses a stochastic search gradient and which limits its ability to find an optimum solution, is inefficient and not practically capable of removing all distortions. The RLS algorithm is typically an order of magnitude faster because it whitens the input data gy using the inverse correlation matrix of the data, assumed to be of zero mean REF. The RLSL algorithm uses a lattice predictor along with a stage that estimates the error of the next filter order at each time step.

The methodology used for designing of the filters in this project will be discussed. Afterwards a detailed analysis of the results for each project and a summary of their implications will be discussed. Finally a conclusions section will highlight the general results drawn from each experiment.

II. Equalization Methodology

ADAPTIVE equalization is designed to remove unwanted signals from a channel output which can be caused by multipath or other distortions. In Figure 1 the general block diagram that is simulated in these experiments is outlined. A data source that generates a random BPSK signal goes through a dispersive channel that has three paths and white noise. The adaptive filter reduces the effects of these signal distortions to generate the output. Before the specifics of this block diagram is discussed a few general principles of Wiener filter theory will be discussed. The aurocorrelation matrix is defined as

$$\boldsymbol{R} = E[\boldsymbol{u}(n)\boldsymbol{u}(n)^T]$$

and the autocovariance:

$$\boldsymbol{p} = E[\boldsymbol{u}(n)d(n)]$$

it can be shown from REF that the Wiener-Hopf or Normal Equation can be solved for the optimum tap weights:

$$\boldsymbol{R}\boldsymbol{w} = \boldsymbol{p}$$

although this is computationaly wasteful and the matrix **R** may not always be invertable therefore adaptive algorithms are used to determine the tap weight vector.

$$\boldsymbol{w} = \begin{bmatrix} w_0 & w_1 & \dots & w_{M-1} \end{bmatrix}^T$$

The output of the channel, which is the input to the adaptive filter is defined as:

$$\boldsymbol{u}(n) = \begin{bmatrix} u(n) & u(n-1) & \dots & u(n-M+1) \end{bmatrix}^T$$

$$= \boldsymbol{h}^T(n)\boldsymbol{a}(n) + \boldsymbol{v}(n)$$

where **h** is known as the channel impulse response. The impulse response for the four channels with varying distortion are defined as

Channel	$\boldsymbol{h}_1$	$\boldsymbol{h}_2$	$\boldsymbol{h}_3$
1	0.2194	1	0.2194
2	0.2798	1	0.2798
3	0.3365	1	0.3365
4	0.3887	1	0.3887

TABLE I: Channel Parameters

To compare the output of the filter to the truth data source adequate delay must be added to the BPSK data. The optimal delay is given by half the filter order, that is:

$$\Delta = \frac{M}{2}$$

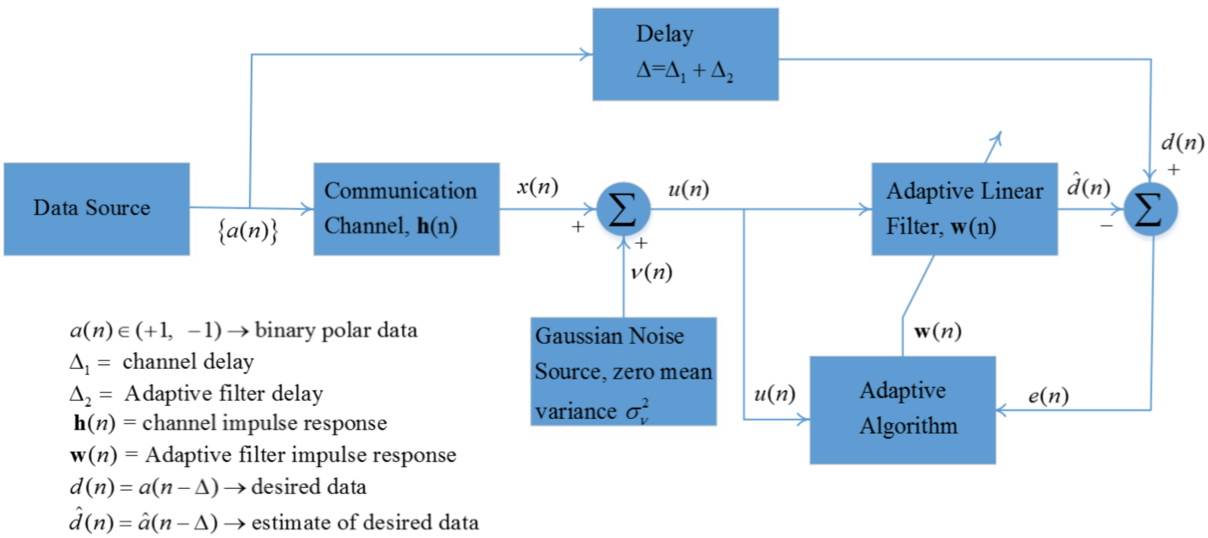

Fig. 1: Block Diagram for Equalization

III. Pre Experiment Calculations

From analysis we are given that the input autocorrelation matrix has a quintdiagonal structure of the form given by:

$$\boldsymbol{R} = \begin{bmatrix} r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) & r(2) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) & r(1) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & r(2) & r(1) & r(0) \end{bmatrix}$$

$$r(0) = h_1^2 + h_2^2 + h_3^2$$

$$r(1) = h_1 h_2 + h_2 h_3$$

$$r(2) = h_1 h_3$$

using the equations given the input autocorelation matrix and can be evaluated for each channel. The eigenvalue spreads for each channel are shown in Table II

Channel	Min Eigenvalue, λ_{min}	Max Eigenvalue, λ_{max}	Eigenvalue Spread $\chi = \lambda_{max}/\lambda_{min}$
1	0.3330	2.2086	6.0915
2	0.2127	2.3752	11.1663
3	0.1246	2.7256	21.8725
4	0.0647	3.0695	47.4274

TABLE II: Pre-Experiment Calculations

As is shown, each channel has progressively worse eigenvalue spread. The effect of this on certain algorithms will be discussed in the following sections

IV. Project 1: Least Mean Squares Algorithm

The LMS algorithm implements a search method to iteratively update the filter tap weights to find an optimal solution.

The mean squared error is defined by:

$$\boldsymbol{J}(n) = \sigma_d^2 - 2\boldsymbol{w}^T\boldsymbol{p} + \boldsymbol{w}^T\boldsymbol{R}\boldsymbol{w}$$

where

$$\sigma_d^2$$

is the deterministic variance and the tap weight vector is

$$\boldsymbol{w}(n) = \boldsymbol{w}(n-1) + \mu\boldsymbol{u}(n)e(n)$$

The filter output which is the tap-weight vector multipled by the tap input vector

$$\hat{d}(n) = y(n) = \boldsymbol{w}^T(n)\boldsymbol{u}(n) = \boldsymbol{u}^T(n)\boldsymbol{w}(n)$$

Intuitively the error of the filter is simply the is the estimate subtracted from the desired signal.

$$e(n) = d(n) - \hat{d}(n) = d(n) - \boldsymbol{w}^T\boldsymbol{u}(n)$$

Which in the context of this project is the randomly generated BPSK signal.

A. *Effect of Eigenvalue Spread*

The LMS algorithm was run with total K = 500 iterations and the error for each value of the tap input vector n = 1,2,...N was averaged. The error was then used to calculate the mean square error using the formula:

$$MSE = \frac{1}{K}\sum_{k=1}^{K} e_k^2(n); \qquad n = 1, 2, ...N$$

This is known as the learning curve of the algorithm. The learning curve is used to judge the performance of each experiment.The results for the eigenvalue spread experiment are shown in Figure 2

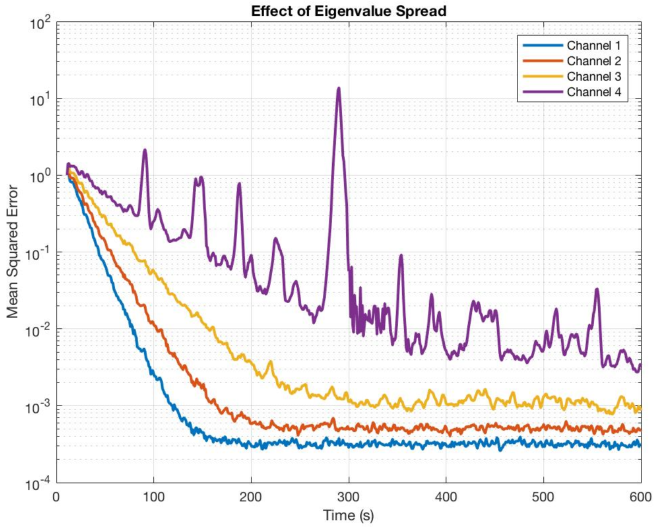

Fig. 2: Learning Curves for Different Channels

These results show that the LMS is very suceptible to eigenvalue spread

B. Effect of Filter Order

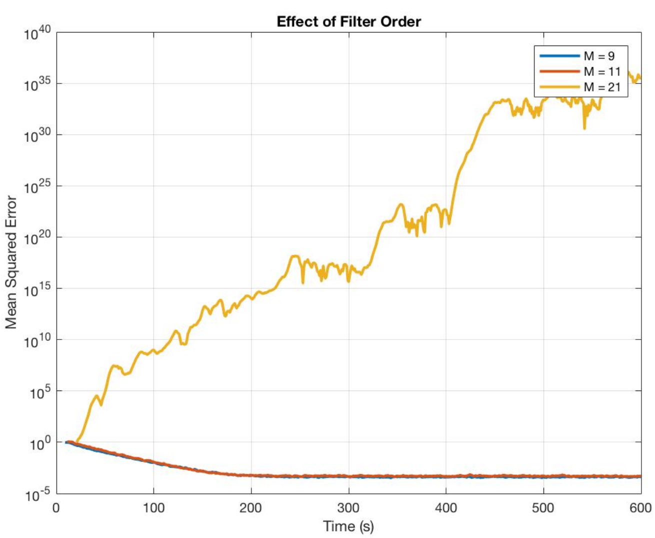

Fig. 3: Learning Curves with Varying Filter Order

C. Analysis of Step Size Parameter

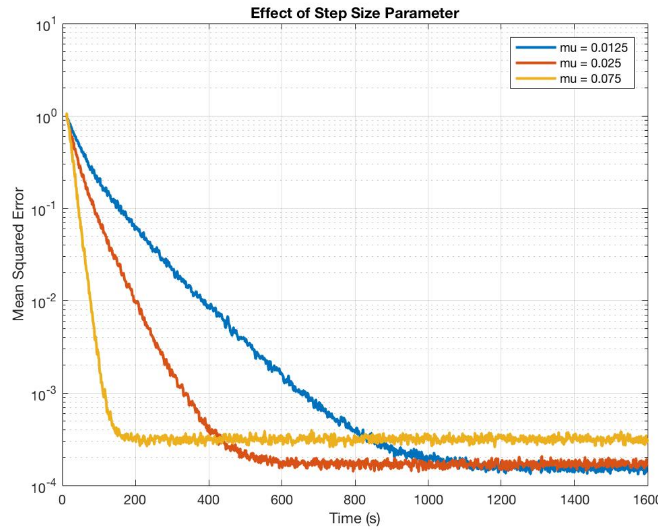

Fig. 4: Learning Curves of Varying Step Size Parameters

D. Comparison of Normalized LMS and Standard LMS

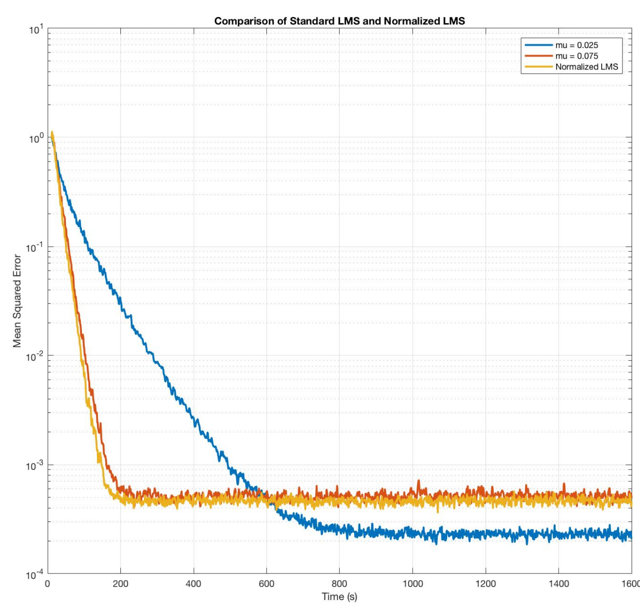

Fig. 5: Learning Curves of Standard and Normalized LMS Algorithms

V. PROJECT 2: RECURSIVE LEAST SQUARES ALGORITHM

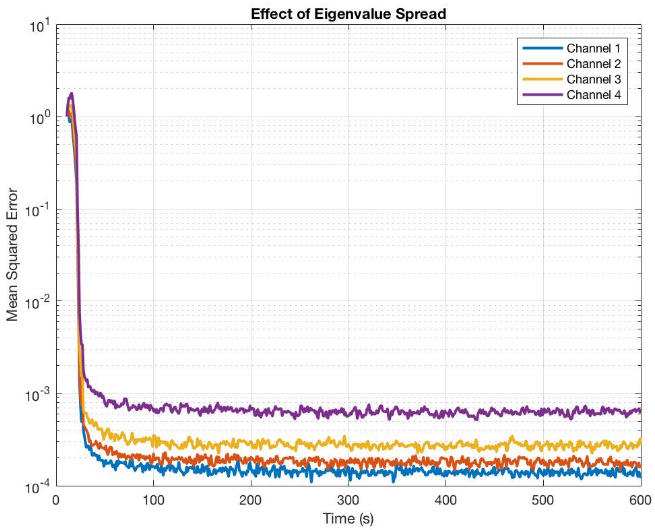

Fig. 6: Effect of Eigenvalue Spread

1) Effect of Eigenvalue Spread:

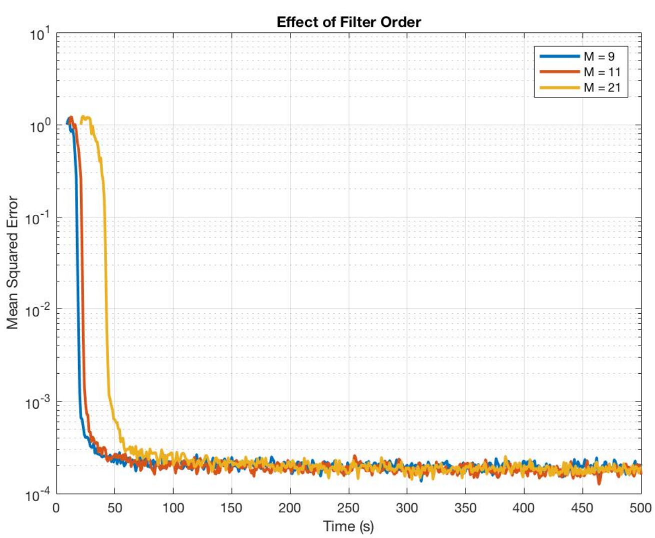

Fig. 7: Effect of Eigenvalue Spread

2) Effect of Filter Order:

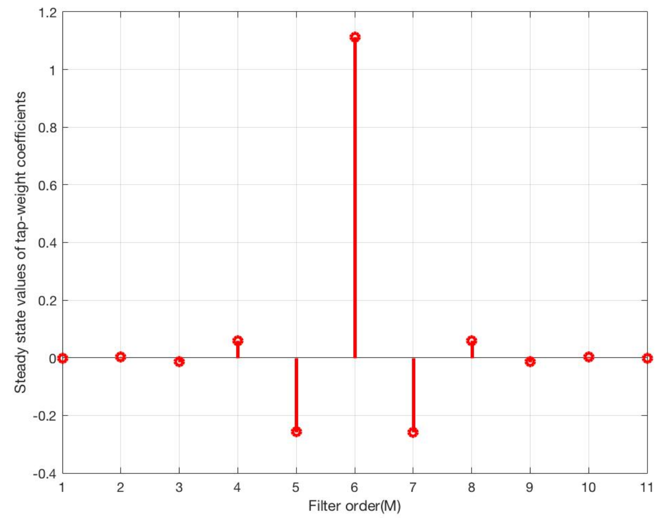

Fig. 8: Effect of Filter Order

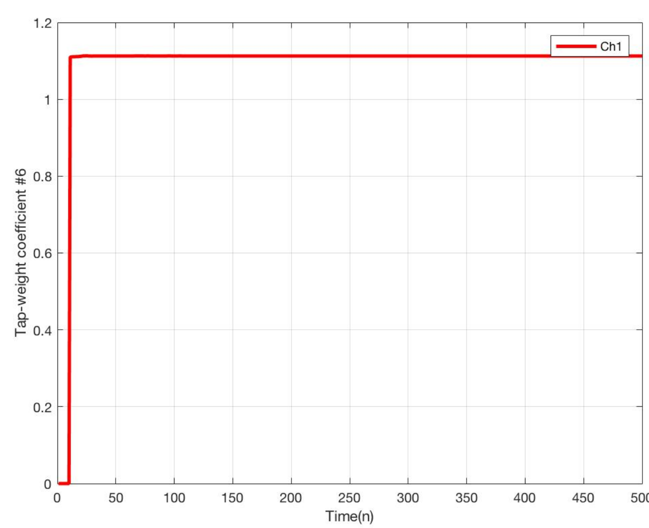

Fig. 9: Stem Plot of Steady-State Tap-Weights

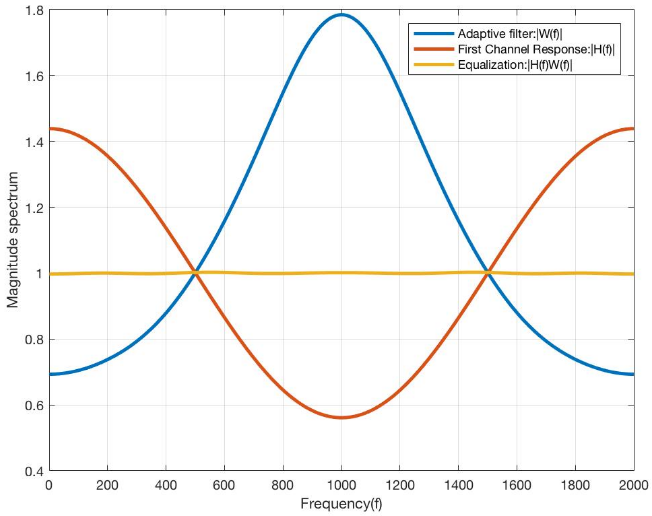

Fig. 10: Center Tap Weight, w_6

3) *Tap-Weight Analysis:*

VI. Conclusion

The conclusion goes here.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgment

The authors would like to thank...

References

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.