EE 254 Adaptive Signal Processing

Fall 2021

Project -3

Adaptive Channel Equalization (LMS Algorithm)

Last Name: Pavthawala

First Name: Mahekkumar Mayankkumar

Student ID: 015235688

Date: 12/02/2021

Abstract

The LMS algorithm has been very often used to adapt the coefficients of a transversal filter, because it is simple and has robustness to numerical calculations. One of the application areas where the LMS adaptive filter is favourably used is that of channel equalization in digital communication systems. However, modem development of multipoint telephone network systems requires fast convergence, so-called fast start-up equalization, of the LMS adaptive filter. For this reason, many adaptive algorithms have been addressed in the last three decades. Most of them, however, sacrifice the computational complexity for updating the filter coefficients. Adaptive equalizers are necessary for reliable communication of digital data across non-ideal channels.

Contents

I. Introduction

The usual method of estimating a signal corrupted by the additive noise it to pass it through a filter that tends to suppress the noise while leaving the signal relatively unchanged i.e., *direct filtering*.

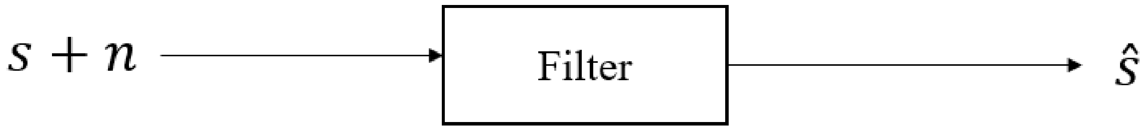

Fig. (1) Basic filter design

Filters used for direct filtering can be either Fixed or Adaptive.

1. *Fixed filter* – The design of fixed filters requires a priori knowledge of both the signal and the noise, i.e., if one knows the input signal and the added or existing noise beforehand, one can design such filter that specifically passes the frequencies contained in of the input signals and filters the frequencies occupied by the interfering noise.

2. *Adaptive filter* – Adaptive filters have ability to adjust their impulse response to the filter out the correlated signal in the input. One requires little or no a priori knowledge of the signal and noise characteristics. (If the signal is narrowband and noise broadband, which is usually the case, or vice versa, no a priori information is needed; otherwise, it requires a desired signal that is correlated in some sense to the signal to be estimated.) Moreover, the adaptive filters have the capability of adaptively tracking the signal under non-stationary conditions.

The adaptive filter is considered the inverse filter which means that this filter gives the inverse coefficient coefficients for the channel and the process for cancelling the effect of the channel. Adaptive filter also known as a convolution filter where the FIR filter convolves the filter coefficients with a sequence of input values and produces a sequence of output values equivalently numbered, so that the FIR filter is used to filter the signals using convolution. FIR impulse response (FIR) filters are finite impulse response digital filters. The FIR filter is the simplest filter to design. FIR filter also known as the convolution filter, where the FIR filter convolves the filter coefficients with a sequence of input values and produces a sequence of output values with the same number so that the FIR filter is used to filter the signals using convolution. The term taps refer to the number of filter coefficients for a FIR filter.

An adaptive equalizer is considered as a time varying filter. The adaptive equalizer is called a transversal filter. In general, adaptive filter consists of two parts an adaptive algorithm and a linear filter. The type of Linear filter is Finite Impulse Response (FIR) filter. The adaptive algorithm is the Least Mean Square (LMS) algorithm. The coefficients of a Finite Impulse Response (FIR) filter can be adjusted by the LMS algorithm to minimize the noise and Inter-symbol Interference. FIR filter is the simplest filter to design. FIR filter is digital filter that have a finite impulse response. Therefore, the least mean squares (LMS) algorithm searches for the optimum value of filter weights. This process repeated rapidly while the equalizer attempts to converge, and more than one technique (such as gradient or steepest decent algorithms) should be used to reduce the error. At the end, the adaptive algorithm freezes the filter weights until the error signal reach to acceptable level or in case of sent a new training sequence.

II. Problem

The adaptive algorithm adjusts the coefficient iteratively to minimize the magnitude of error $e(k)$. In this paper we use variable step size to improve the speed of convergence for adaptive filter. When we use LMS algorithm to create an adaptive filter, we must choose value for step size. The step size effects on stability of adaptive filter, the speed of convergence and the error of steady state. When the step size value is small the error of steady state will be small and the speed of convergence for the adaptive filter will decrease. When the step size value is large, the speed of convergence for the adaptive filter will be improved. Therefore, when the step size value is large this might make the adaptive unstable.

The adaptive equalizer is a time difference filter. An adaptive equalizer is called a transversal filter. Generally, an adaptive filter consists of an adaptive algorithm and a linear filter. The linear type is a finite impulse response (FIR) filter. The adaptive algorithm is the least mean square (LMS) algorithm. The adaptive equalizers are necessary for reliable communication of digital data across nonideal channels. The basic operation of a digital communication system may be described as follows. Let $d(n)$ be a digital sequence that is to be transmitted across a channel, with $d(n)$ having values of plus or minus one. This sequence is input to a pulse generator, which produces a pulse of amplitude A at time n if $d(n) = 1$ and a pulse of amplitude -A if $d(n) = -1$. This sequence of pulses is then modulated and transmitted across a channel to a remote receiver. The receiver demodulates and samples the received waveform, which produces a discrete-time sequence $x(n)$.

First, since the channel is never ideal, it will introduce some distortion. One common type of distortion is channel dispersion that is the result of the nonlinear phase characteristics of the channel. This dispersion causes a distortion of the pulse shape, thereby causing neighbouring pulses to interfere with each other, resulting in an effect known as intersymbol interference. The second reason is that the received waveform will invariably contain noise. This noise may be introduced by the channel, or it may be the result of nonideal elements in the transmitter and receiver. Assuming a linear dispersive channel, a model for the received sequence $x(n)$ is,

$$x(n) = \sum_{k=-\infty}^{n} d(k)h(n-k) + v(n)$$

where $h(n)$ is the unit sample response of the channel and $v(n)$ is the additive noise. Given the received sequence $x(n)$, the receiver then decides as to whether a plus one or a minus one was transmitted at time n.

However, to reduce the chance of making an error, the receiver will often employ an equalizer to reduce the effects of channel distortion. Since the precise characteristics of the channel are unknown, and possibly time-varying, the equalizer is typically an adaptive filter. one of the challenges in the design of an adaptive filter is generating the desired signal d(n), which is required to compute the error $e(n)$. Without the error, a gradient descent algorithm will not work. For the channel equalizer, there are two methods that may be used to generate the error signal.

1. Training method. This method takes place during an initial training phase, which occurs when the transmitter and receiver first establish a connection. During this phase, the transmitter sends a sequence of pseudorandom digits that is known to the receiver. With knowledge of $d(n)$ the error sequence is easily determined, and the tap weights of the equalizer may be initialized.

2. Decision-directed method. Once the training period has ended and data is being exchanged between the transmitter and receiver, the receiver has no a priori knowledge of what is being sent. However, there is a clever scheme that may be used to extract the error sequence from the output of the threshold device as illustrated in Fig. 9.19b. Specifically, note that if no errors are made at the output of the threshold device and $\hat{d}(n) = d(n)$, then the error sequence may be formed by taking the difference between the equalizer output, y(n), and the output of the threshold device,

$$e(n) = y(n) - \hat{d}(n)$$

This approach is said to be decision directed since it is based on the decisions made by the receiver. Although based on the threshold device making a correct decision, this approach will work even in the presence of errors, if they are infrequent enough. Of course, once the error rate exceeds a certain level, the inaccuracies in the error signal will cause the equalizer to diverge away from the correct solution thereby causing an increase in the error rate and eventually a loss of reliable communication. Before this happens, however, the receiver may request that the training sequence be retransmitted to re-initialize the equalizer.

III. Results

After running the MATLAB code 'equalizer.m' with certain input to replicate the output, one can see the equalizer adaptations does not require the knowledge of the transmitted sequence nor carrier phase recovery and is also independent of the data symbol constellation used in the transmission system.
The channel responses are represented in two formats,

$$H_1(z) = [0.35 \quad 1 \quad -0.35]$$

and,

$$H_2(z) = [0.35 \quad 1 \quad 0.35]$$

The equalizer length, $N = 15$ and the delay, $\Delta = 9$.
The step size parameter, μ is chosen for three misadjustment values $10\%, 20\%, \& \ 30\%$.

(1)

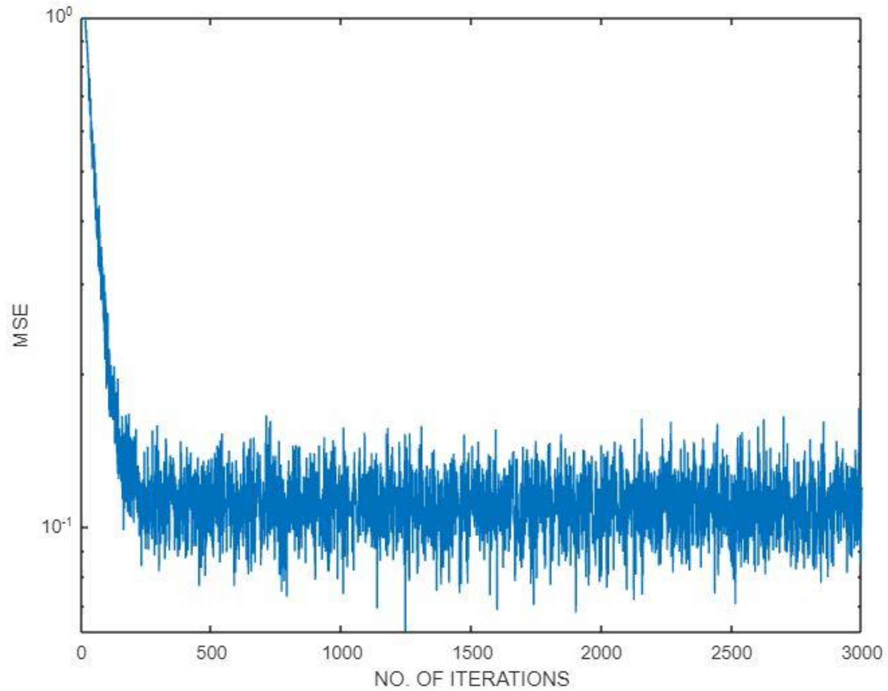

Fig. (2) The step size parameter, $\mu = 10\%$ $H_1(z)$

(2)

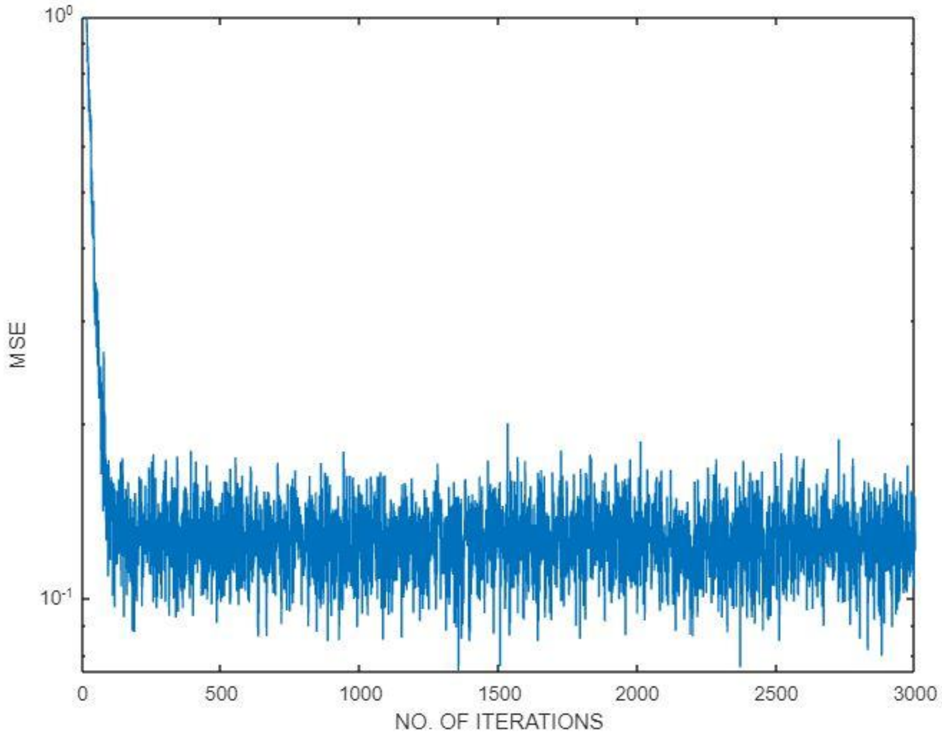

Fig. (3) The step size parameter, $\mu = 20\%$ $H_1(z)$

(3)

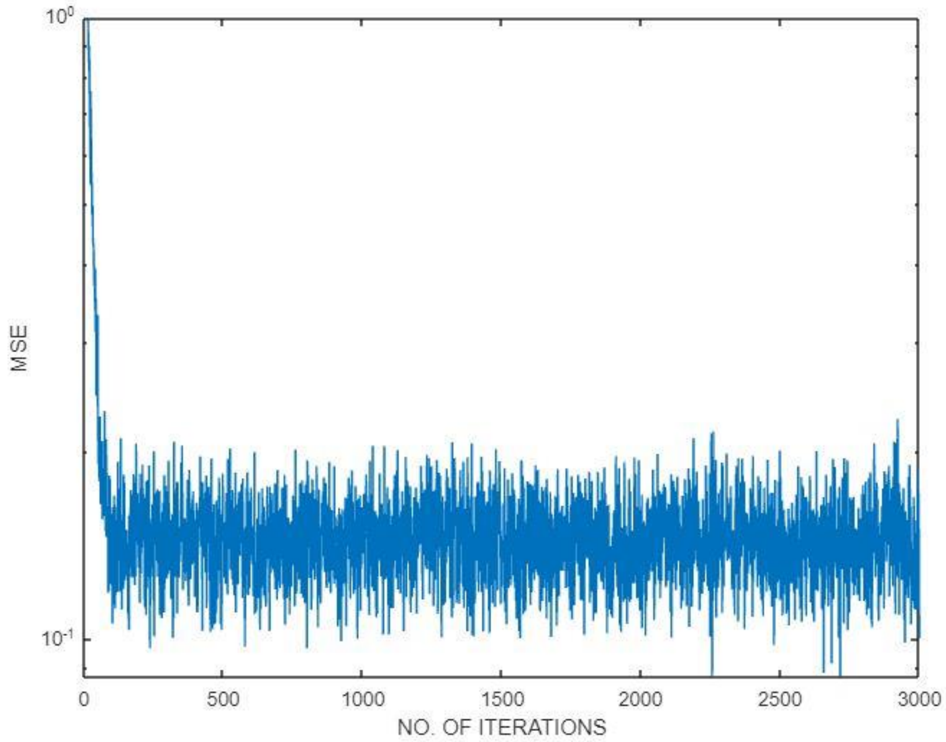

Fig. (4) The step size parameter, $\mu = 30\%$ for $H_1(z)$

For the channel response $H_2(z)$,

(1)

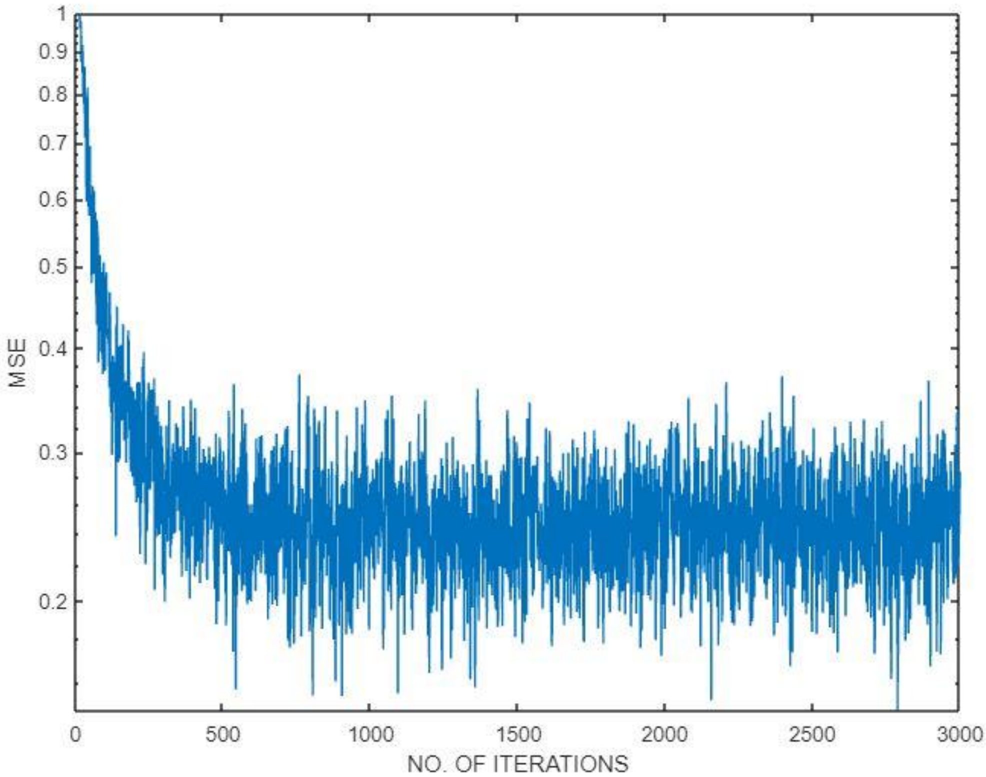

Fig. (5) The step size parameter, $\mu = 10\%$ for $H_2(z)$

(2)

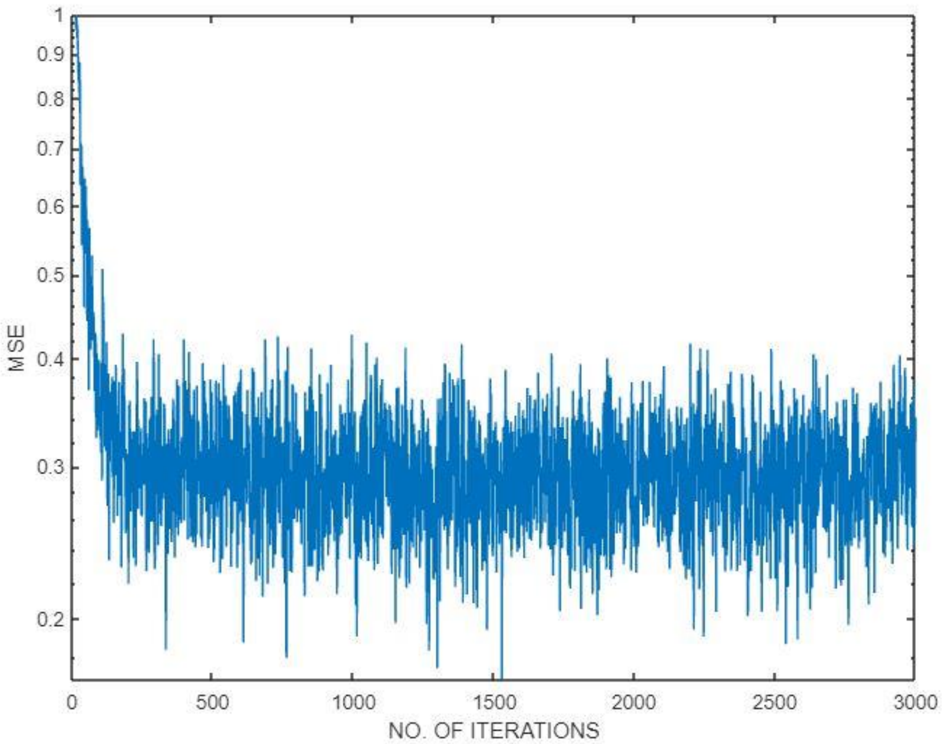

Fig. (6) The step size parameter, $\mu = 20\%$ for $H_2(z)$

(3)

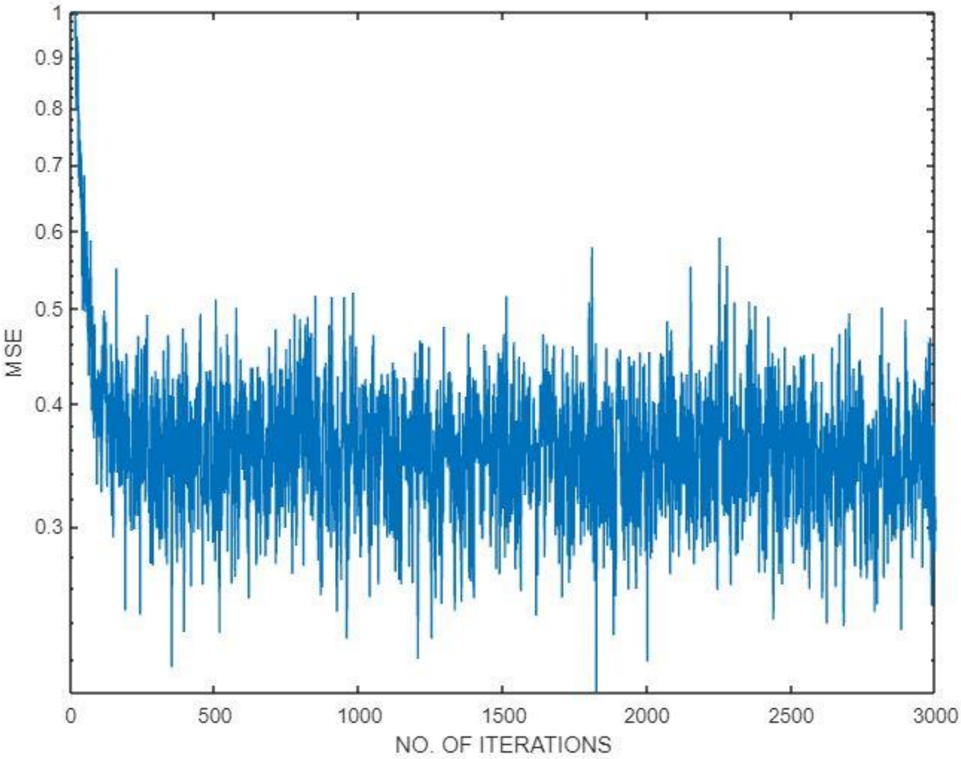

Fig. (7) The step size parameter, $\mu = 30\%$ for $H_2(z)$

From these observations, one can say that when misadjustment rate increases, the output MSE vs. No of iterations curves reach near to a straight line, eventually.

When the parameters are changed, and the no. of iterations are decreased, it is more visible.

(1)

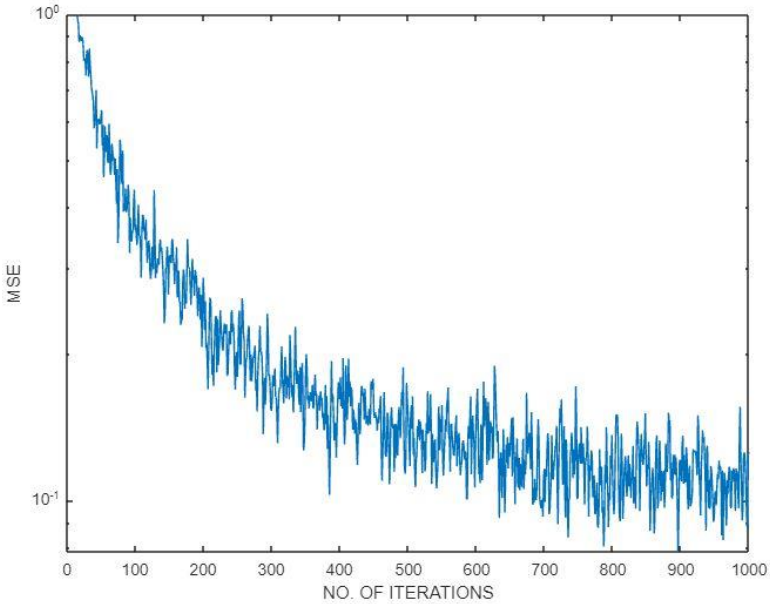

Fig. (8) The step size parameter, $\mu = 10\%$ for $H_2(z)$ for 1000 iterations

(2)

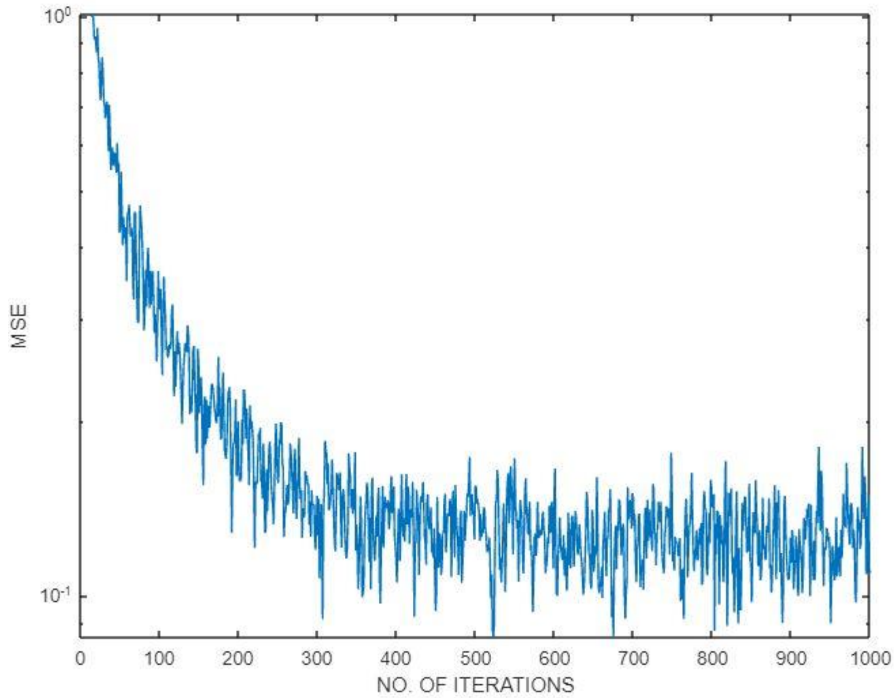

Fig. (9) The step size parameter, $\mu = 20\%$ for $H_2(z)$ for 1000 iterations

(3)

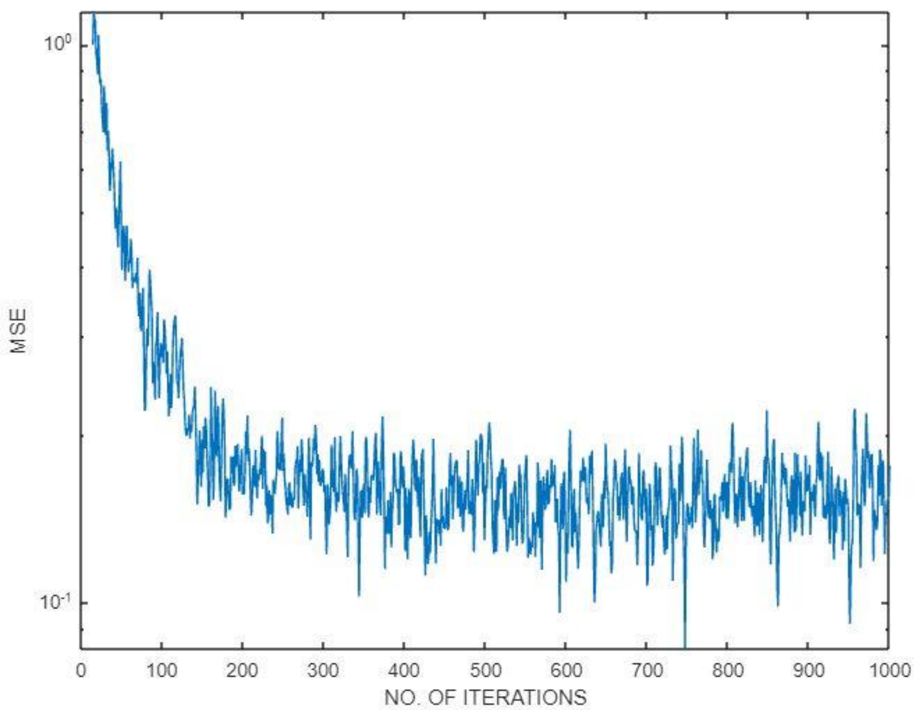

Fig. (10) The step size parameter, $\mu = 30\%$ for $H_2(z)$ for 1000 iterations

IV. Conclusion

The Least Mean Square (LMS) algorithm requires fewer computational because it is simple and no need for matrix inversion. LMS algorithm is applied to equalize the effect of the channel. In this paper we take different value for step size in adaptive equalizer. The value for the step size must be used when the Least Mean Square (LMS) algorithm used to create an adaptive filter. From the simulation result we conclude that when the step size value is small the system will be slow, but the steady state mean square error will be small. Otherwise, when the step size value is large the system will be fast, but the steady state mean square error will be large.

V. References

[1] Wang Junfeng, Zhang Bo, "Design of Adaptive Equalizer Based on Variable Step LMS Algorithm" ,ISCSCT' 10 , Jiaozuo, P. R. China, 14-15, August 2010.

[2] M. H. Hayes, "Adaptive filtering," in Statistical digital signal processing and modeling, John Wiley & Sons, 1996.

[3] MathWorks, “Compare RLS and LMS Adaptive Filter Algorithms,” [Online]. Available: https://www.mathworks.com/help/dsp/ug/compare-rls-and-lmsadaptive-filter-algorithms.html [Accessed: 28 11 2021].