# DTS Final Report

Effects of step size on LMS convergence in a Source Coding scenario

Roi Mashiah, ID：318734027

Date：20.07.2023

## Introduction

The convergence of the LMS (Least Mean Squares) method and its dependency on the step size parameter are crucial considerations in various applications. In this report, we investigate the effects of step size on LMS convergence in a source coding scenario. Section I contains a short reminder of linear prediction and presents a variable step size algorithm. Section II contains computer simulation results for clean prediction as well as a description of the different data sets used to examine the phenomenon. Section III first explains the Source Coding scheme used (ADPCM) and continues to present computer simulation results with different quantization levels (bit rates). Section IV contains the conclusions drawn from this work as well as suggestions for further work.

## Section I

Consider the stationary random process $x[n]$, with the corresponding correlation matrix $R_{xx}[k]$. To linearly predict future values of $x[n]$ we saw that one may solve the Yule-Walker Equations given by:

$$\boldsymbol{Ra} = \boldsymbol{r}$$

where $\boldsymbol{r}^{\mathrm{T}} \in \mathbb{C}^p$ is the cross- correlation vector between the inputs to the desired response, and $\boldsymbol{a} \in \mathbb{C}^p$ are the optimal weights for predicting the future value at moment n given the past p samples. The predictor is given by:

$$\hat{x}[n] = \sum_{i=1}^{p} a_i x[n-i]$$

The optimal solution for the coefficients vector is given by:

$$\boldsymbol{a}_{opt} = \boldsymbol{R}^{-1}\boldsymbol{r}.$$

In this section we introduce the LMS filter, also known as the SGD method which is a method for finding the optimal weights when the autocorrelation function is not known. Furthermore, this method may be useful even for formally nonstationary random processes to receive an optimal approaching solution.

This method is based on the topology of the error surface. We define the error between our prediction to the desired output:

$$e[n] = x[n] - \hat{x}[n]$$

We would like to minimize the error in the mean squared sense i.e., we would like to find the vector $\boldsymbol{a}$ that leads to the MMSE. Since we cannot solve for $\boldsymbol{a_{opt}}$, we use a broad search approach and exploit the fact that the error surface is quadratic. This can be seen from this form:

$$MSE \stackrel{\text{def}}{=} E\left[e^2[n]\right] = R_{xx}[0] - \boldsymbol{a^T r} - \boldsymbol{r^T a} + \boldsymbol{a^T R a}.$$

We should differentiate w.r.t the coefficients and move along the error surface towards the minimum (in the direction of the gradient). This is the method of steepest descent (gradient descent) where one calculates the gradient of the **mean error** and follows the path in a direct fashion towards the minimum:

$$\forall n \in \mathbb{Z}: \boldsymbol{a_n} = \boldsymbol{a_{n-1}} + \mu * \nabla E\left[e^2[n]\right]$$

where $\mu$ is the step size parameter and $n$ is the time index. In our case, since we don't have prior knowledge of second order statistics, we try to develop an estimation of the real gradient. To develop an estimation of the gradient at time $n$ we seek to substitute parameters regarding the correlation matrix with ones that can be calculated empirically. The expression for the gradient is given by:

$$\nabla E\left[e^2[n]\right] = -2\boldsymbol{r} + 2\boldsymbol{Ra}$$

by using the estimated cross-correlation and autocorrelation and substituting into the steepest descent method we get

$$\boldsymbol{a_n} = \boldsymbol{a_{n-1}} + \mu * \boldsymbol{x_n} * e[n]$$

Where $\boldsymbol{x_n} = \left[x[n-1]\ x[n-2], \ldots, x[n-p-1]\right]^T$ are the last p samples of the random process. This is the LMS algorithm. We skip the proof of convergence to a minimum in this work, but providing some restrictions to the step size one may show that this method converges in the mean squared sense.

As mentioned, the step size parameter may be bounded from above, but these bounds assume very optimistic assumptions on the input which leads to choosing a significantly smaller step size far from the bound. A small step size indeed solves the problem of divergence, but simultaneously leads to long convergence rates. In this work I propose a variable step size algorithm which tries to evaluate the state of the LMS process.

The proposed algorithm for varying step size is as follows:

- Every predefined block size, calculate the MSE vector based on the error vector.
  - $MSE[n] = \frac{e^2[n]}{block\ size}$
- If a previous value for the MSE exists, compare the current error to the previous.
- Decide the new step size according to the difference as follows:
  - If the difference is less than $A$ then we can increase the step size $\mu$, by $1 + G, 0 < G < 1$ to converge faster, if $\mu(1 + G) < \mu_{max}$
  - If the difference obeys $A < d < 3A$, we are probably at some area close to a minimum which implies that we should decrease the current step size.
  - If $d > 3A$, we might be diverging so reduce the step size dramatically by a factor of $G$.
  - For any other case we stay with the previous step size

We now proceed to show computer simulations.

## Section II

To demonstrate the LMS filter operation, one may show the convergence of the filter coefficients to a (approximately) minimum MSE. In this work, two types of data are examined – the first is an $AR(2)$ random process with coefficients that are time dependent, the second is recorded speech from the web[1]. The AR(2) data is chosen for control reasons – showing that the coefficients converge to the correct solution (known parameters). The second is chosen for reasons that will be realized later.

AR(2) data was simulated using an all pole filter (IIR) with random coefficients patched together to form a nonstationary sequence.

We first show the AR(2) data and compare the constant step size to the adaptive step size in two extremes – a low learning rate and a high learning rate.

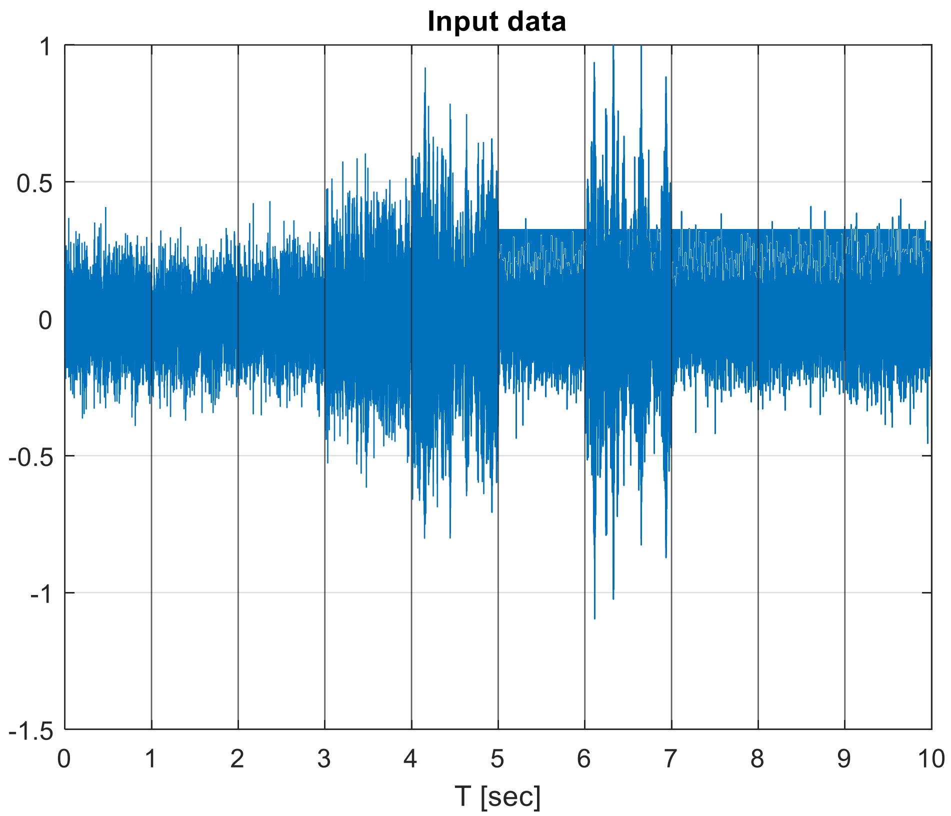



*Figure 1 Realization of AR(2) patched signal with time varying coefficients*

1- http://www.openslr.org/12

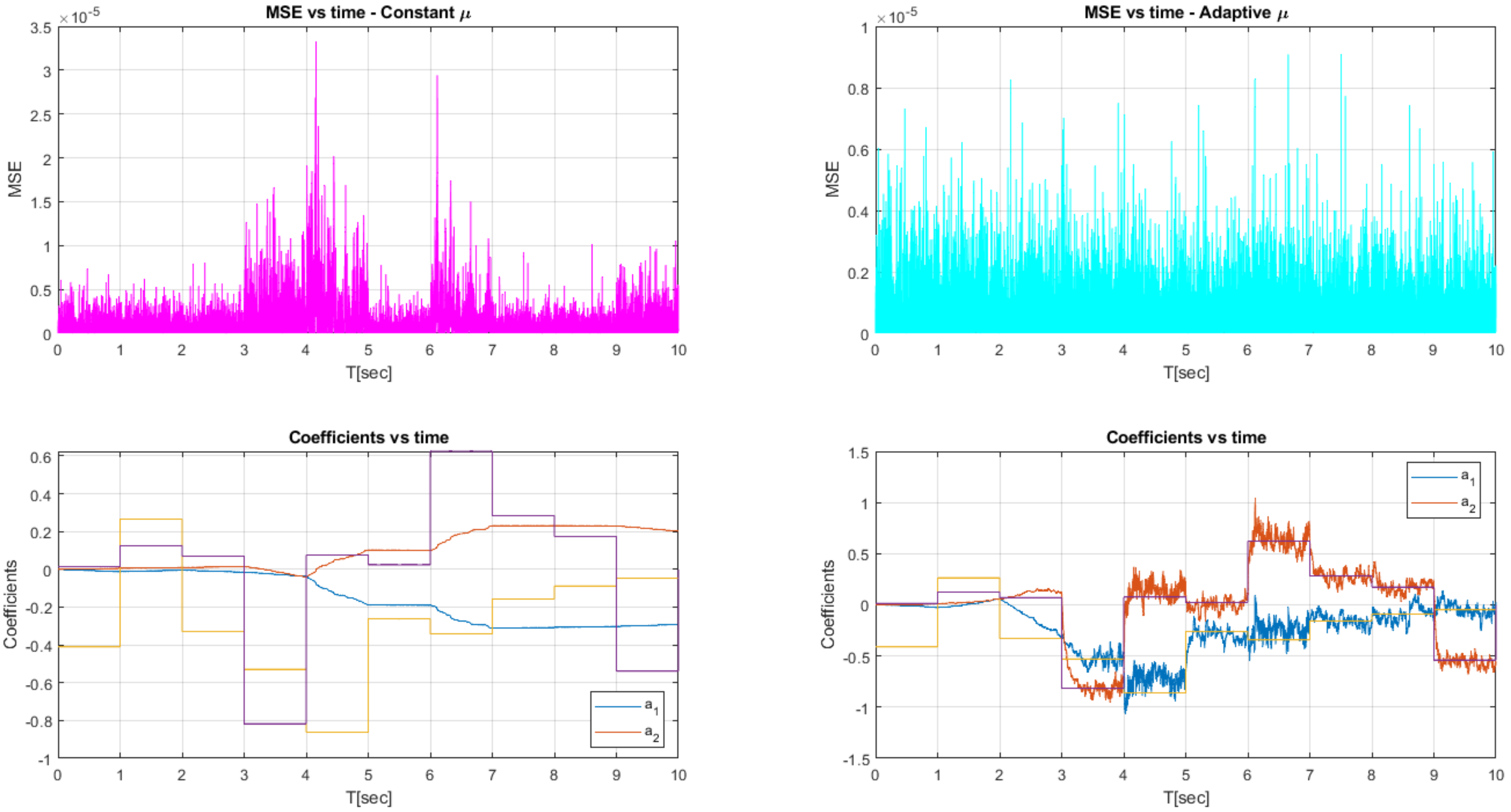



*Figure 2 Upper row - MSE vs time, left column is w/ constant LR 0.001, right column w/ adaptive LR. Bottom row - the filter coefficients over time in blue and orange, the real coefficients in yellow and purple*

In this experiment the following values were used for the adaptive run:

$$A = 1E - 3, G = 0.2, \mu_{max} = 1, Block\ Size = 0.1 * f_s$$

The choice of these values came from empirical tests. In Section IV we will address this matter.

This example demonstrates the advantage of using an adaptive step size versus a constant low step size. The adaptive step size can accommodate and follow changes in the input data while the constant low step size causes the coefficients to stall. We see that for coefficients that are far from the previous value we get a large error, while for the adaptive case the error doesn't fluctuate as much. The values for the empirical MSE are calculated and given by:

$$MSE_{constant} = \ 1.05 * 10^{-6}, MSE_{adaptive} = 5.27 * 10^{-7}$$

Indeed, we receive an order of magnitude difference between the two methods.

We now present the other extreme case of a large initial step size and examine the results. To show this phenomenon, we chose $\mu_{max} = 3, \mu_0 = 3, A = 1e-5$.

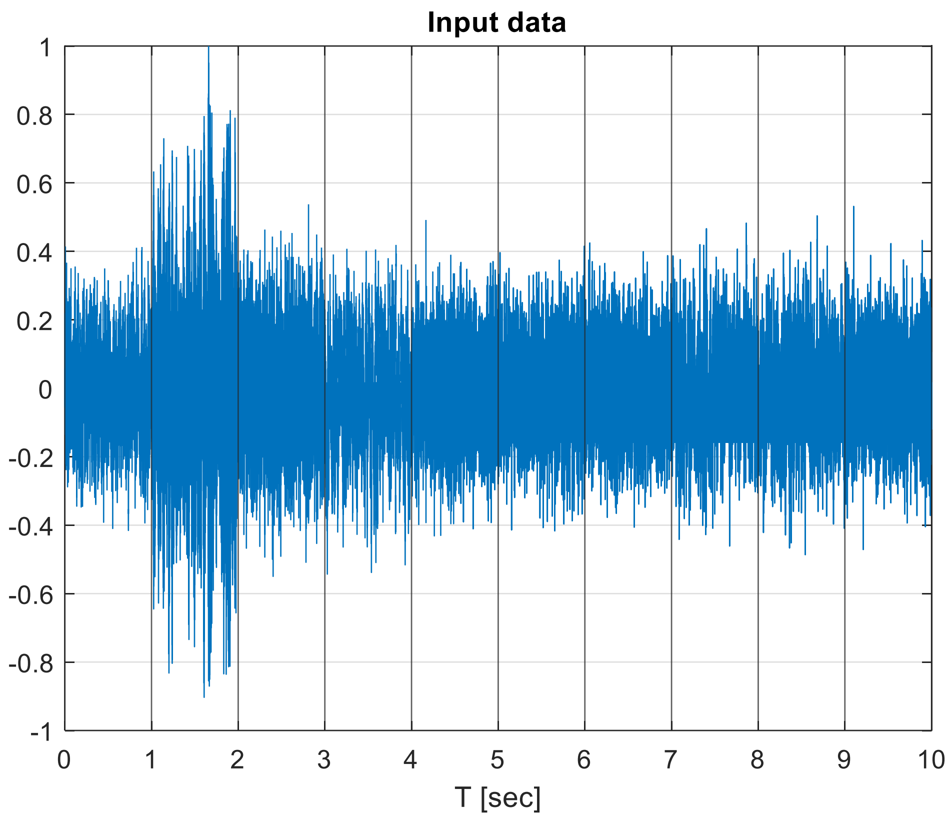
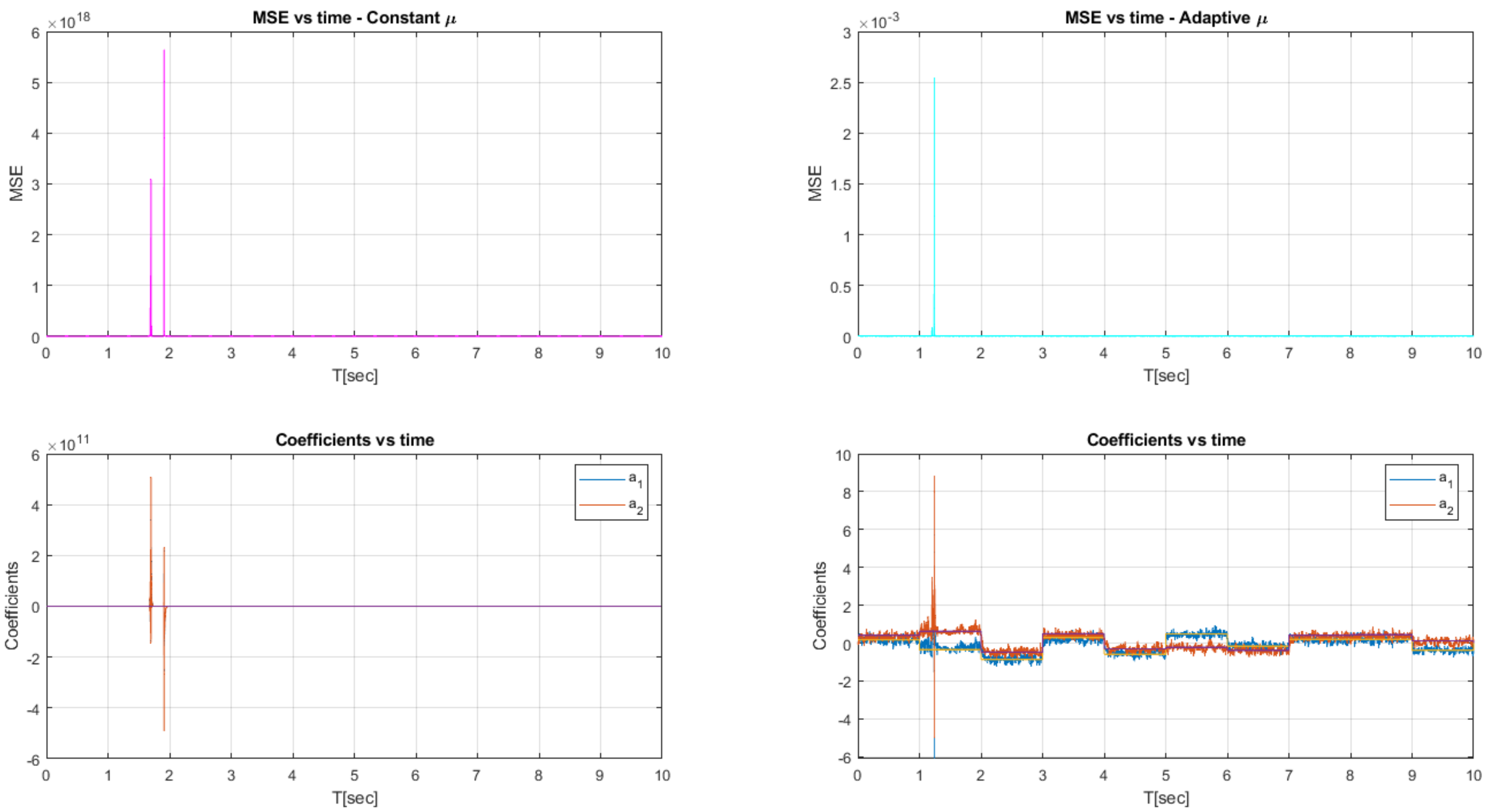


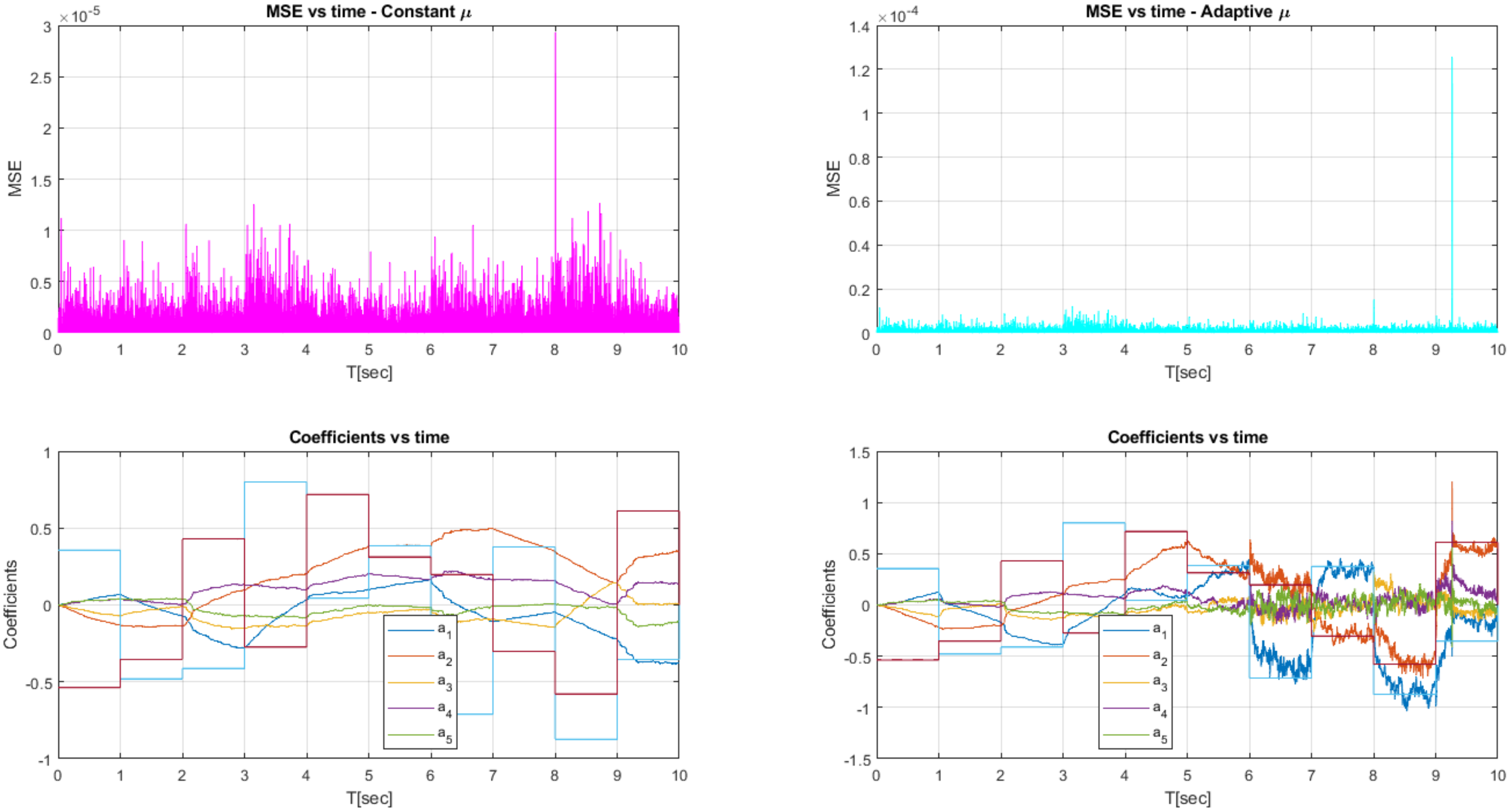

*Figure 3 Realization of AR(2) patched signal with time varying coefficients*



As we see in this case, the large LR leads to a divergence in the constant case while the adaptive step size manages to lead us in the right direction. The results for the total distortion are given by:

$$MSE_{constant} = 1.64 * 10^{15}; MSE_{adaptive} = 1 * 10^{-6}$$

In order to conclude the AR(2) dataset we show the convergence of the coefficients for a greater filter order, to justify the correctness of the LMS algorithm implementation. We set the filter order to 5 and expect that coefficients 3 to 5 approach zero:




In both cases, the coefficients are centered around zero, as we expected.

We now present the second data set – speech. For this data set we will show one example for demonstration purposes and then present a bar graph for several examples comparing the MSE.

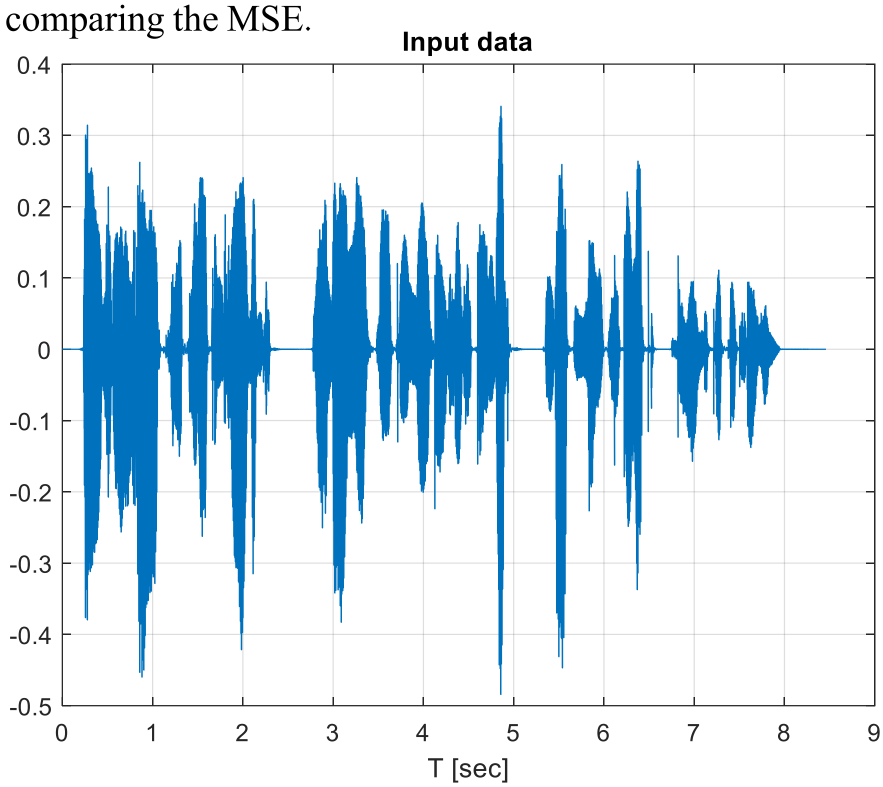



*Figure 5 Input speech data*

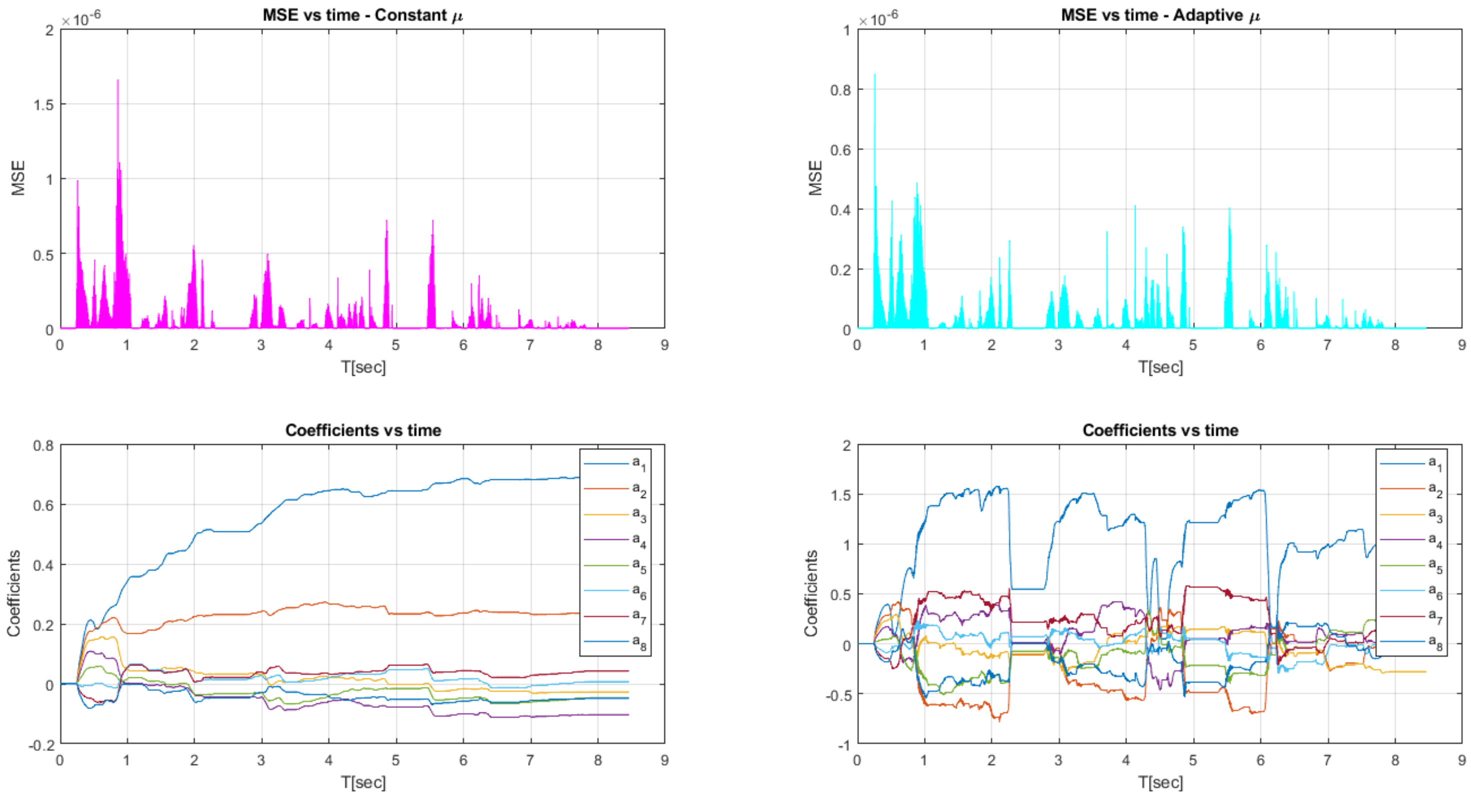
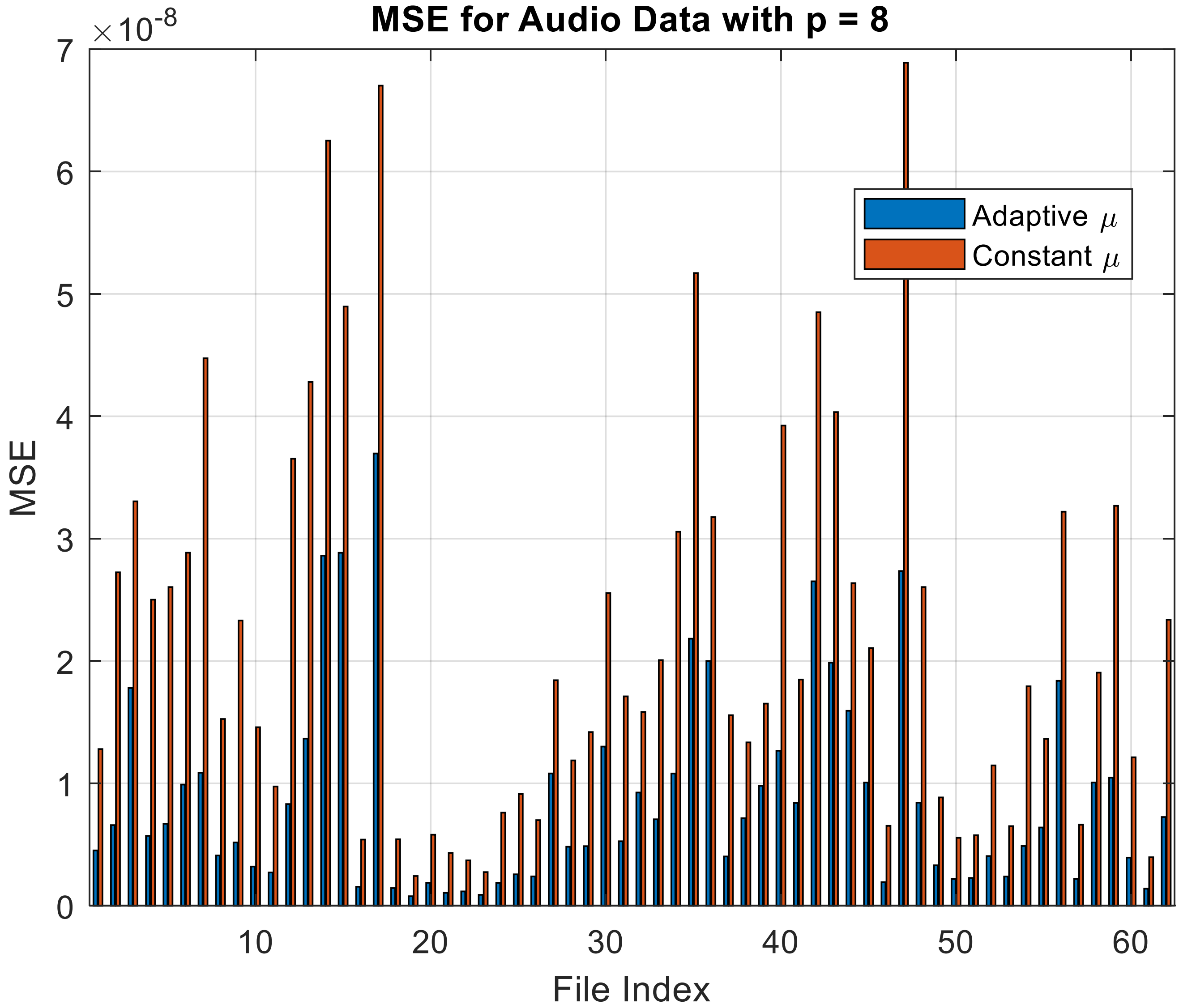



*Figure 6 comparison between adaptive and constant LR 0.01*

We can see that for a reasonable step size the behavior of the algorithm is similar. Although, for a more conservative step size such as 0.001, we get different results. We ran the process on 62 files. The results are presented in the following bar chart:




The block size is $30ms * f_s$ and the other parameters are the same as before.

We see clearly that the total distortion is larger for the constant step size scheme compared to the adaptive step size scheme.

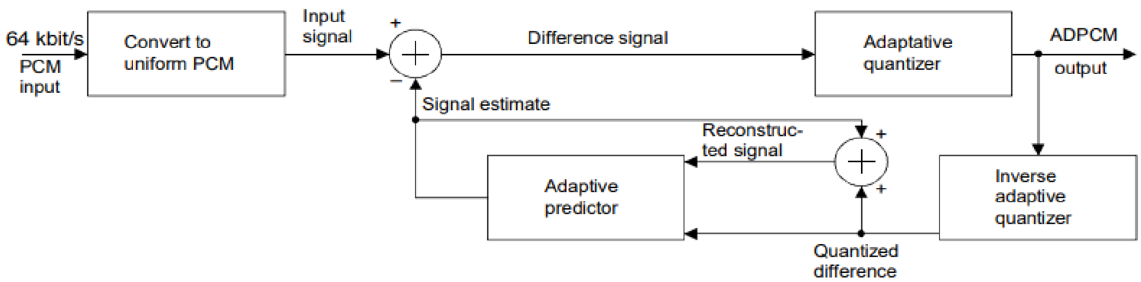
## Section III

Source coding is a mapping from (a sequence of) symbols from an information source to a sequence of alphabet symbols (usually bits) such that the source symbols can be exactly recovered from the binary bits (lossless source coding) or recovered within some distortion (lossy source coding).

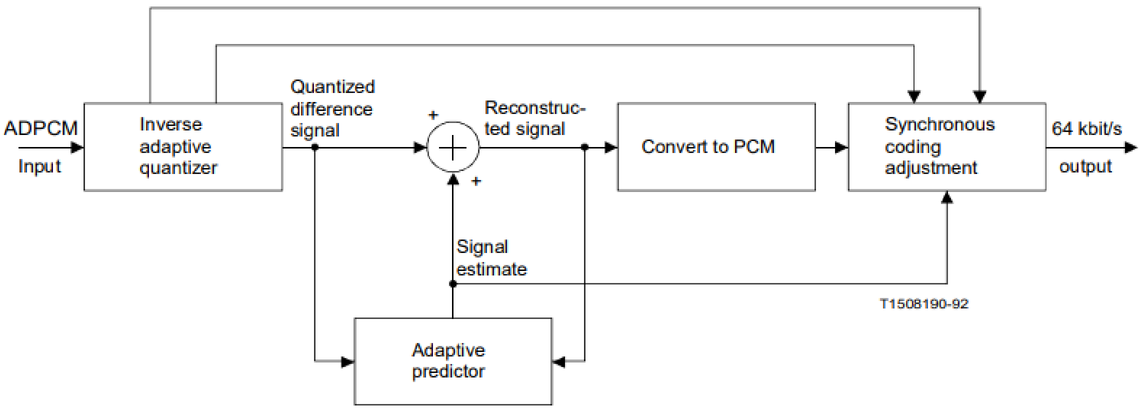
Many methods for source coding have been explored over the years and ADPCM (Adaptive Differential Pulse Code Modulation) is among them.

In general, if an information source has some non-white autocorrelation function, i.e., insights about the future may be inferred from the past, as source coding system designers, we would like to try and use the acquired observations to reduce the number of transmitted bits (reduce bit rate). This is the basic concept behind differential coding schemes such as ADPCM.

The block diagram of ADPCM codec is presented:




*a) Encoder*




*b) Decoder*

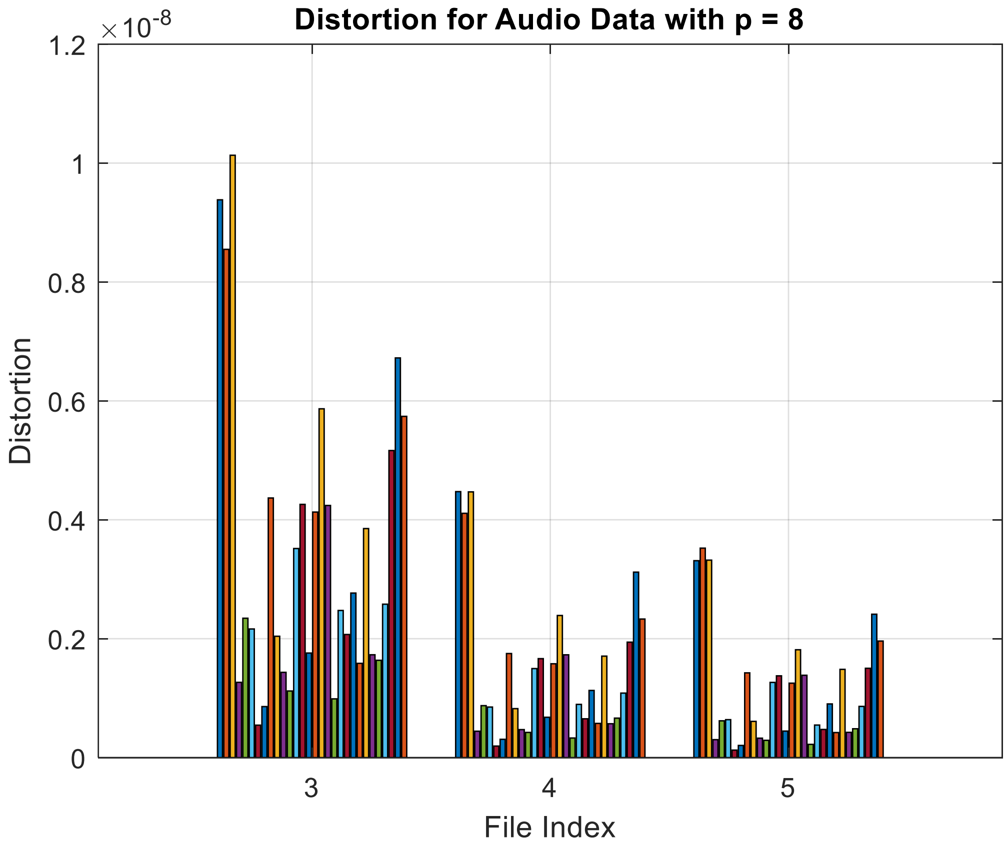
We will focus on the adaptive filter in the ADPCM scheme and demonstrate how the LMS filter operates on the quantized error as earlier we used the real error.

In this work, to focus on the main theme of convergence of the LMS filter coefficients as a function of the learning rate, we implement the ADPCM codec without an adaptive quantizer. Instead, we use one of the speech examples to generate an optimal partition for the error signal (assuming it represents the speech data) using Lloyds Algorithm and use it in the encoder and decoder.

We use the adapting step size scheme for the simulation with the following parameters:

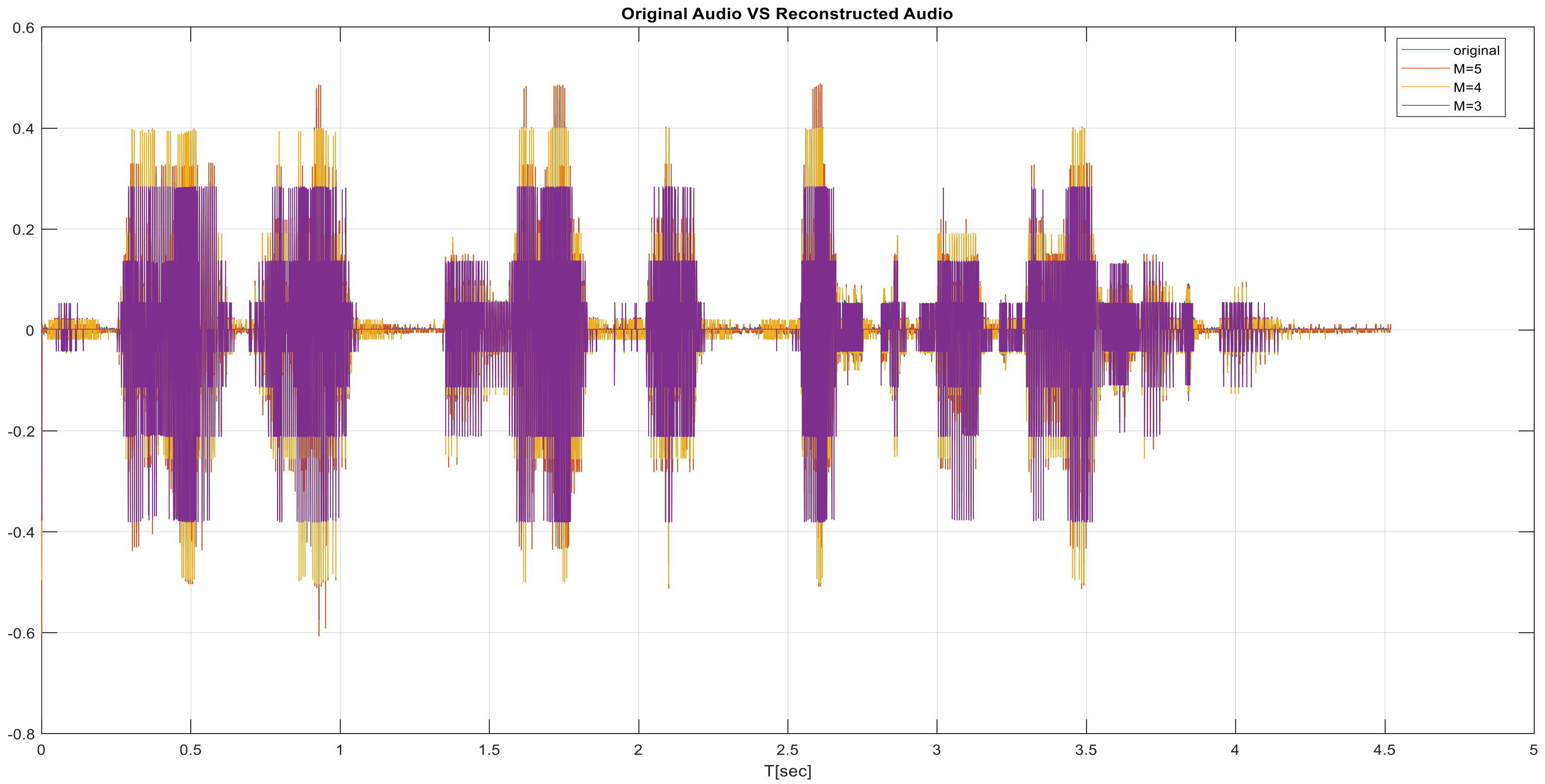
$$\mu_0 = 0.01, p = 8, G = 0.1, \mu_{max} = 1, A = 1 * 10^{-12}, Block\ Size\ 3ms * f_s$$

We run the simulation of the system for 3, 4, 5 bits for the quantized error on 30 speech examples and show the results in a bar graph as before.

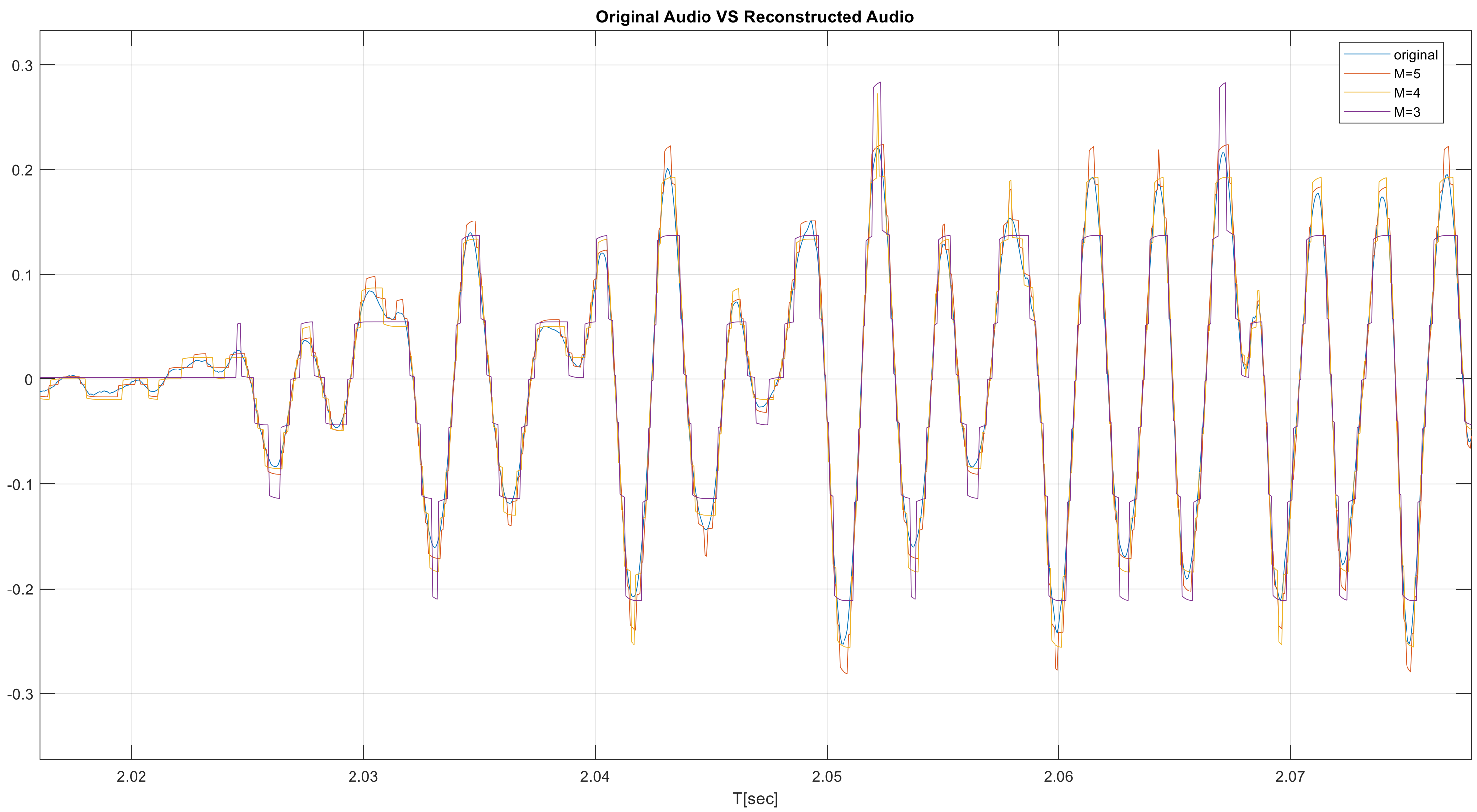



As expected, the distortion reduces as the bit rate increases. This shouldn't surprise us as we know from Shannon's Distortion rate theorem that the rate as a function of the allowed distortion is related logarithmically – the more distortion we allow the less bits we need to transmit to recover at the receiver with said distortion.

To further demonstrate, we take one file and plot the difference between the original audio and the reconstructed audio for each of the bit rates.




We can clearly see that the case of M=3 produces the worst reconstruction while the M=5 is the best. This may be clarified by zooming in on one of the sections:



## Section IV

In this report, the effects of step size on the convergence of the LMS (Least Mean Squares) method in a source coding scenario were examined. The LMS filter, a method for finding optimal weights in the absence of known autocorrelation, was introduced. A variable step size algorithm was proposed to evaluate the state of the LMS process and adjust the step size accordingly.

Computer simulations were conducted using two types of data: an AR(2) random process with time-varying coefficients and recorded speech. The results demonstrated the advantages of using an adaptive step size compared to a constant low step size. The adaptive step size showed better accommodation to changes in input data and faster convergence, while the constant step size resulted in stalled coefficients.

Furthermore, the simulations explored the convergence of the LMS algorithm for different filter orders, validating the correctness of the implementation. The second data set, speech, was used to demonstrate the performance of the LMS filter in a source coding scenario. The adaptive step size was applied to the quantized error, resulting in lower distortion compared to a constant step size.

The supposed algorithm for adaptation of the step size performed relatively well in some scenarios, although this comes with a price of processing delay and complexity. It is acknowledged that the supposed adaptation algorithm is not robust nor optimal in any sense, but it may be improved by optimizing the delta factor $A$, the gain factor $G$, as well as the upper bound for the step size $\mu_{max}$ for certain data types to ensure optimality in convergence rate. This may be done using machine learning techniques with large data sets that may improve the results.

Overall, the study emphasized the importance of selecting an appropriate step size to achieve efficient convergence in LMS-based systems.