UNIVERSITY OF CALIFORNIA,
IRVINE

Channel Estimation Methods by Using Prebeamforming Technique in Massive MIMO

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical Engineering

by

Sadjad Sedighi

Dissertation Committee:
Professor Ender Ayanoglu, Chair
Professor Hamid Jafarkhani
Professor Syed Ali Jafar

2017

DEDICATION

To Maman, Baba and Sahar

TABLE OF CONTENTS

LIST OF FIGURES

Page

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Prof. Ender Ayanoglu. This work could not be done without him. The door to Prof. Ayanoglu office was always open whenever I ran into a trouble spot or had a question about my research or writing.

I also want to thank Dr. Gokhan M. Guvensen for his great help during my first year in masters and for his insightful comments and encouragement.

Finally, I must express my very profound gratitude to my parents and to my sister for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you.

Sadjad

CURRICULUM VITAE

Sadjad Sedighi

EDUCATION

Masters of Science in Electrical Engineering **2017**
University of California, Irvine *Irvine, California*

Bachelor of Science in Electrical Engineering **2015**
Sharif University of Technology *Tehran, Iran*

RESEARCH EXPERIENCE

Graduate Research Assistant **2015–2017**
University of California, Irvine *Irvine, California*

Summer Research Intern **2014**
Chinese University of Hong Kong *Hong Kong*

TEACHING EXPERIENCE

Teaching Assistant **2013–2014**
Sharif University of Technology *Tehran, Iran*

ABSTRACT OF THE DISSERTATION

Channel Estimation Methods by Using Prebeamforming Technique in Massive MIMO

By

Sadjad Sedighi

Master of Science in Electrical Engineering

University of California, Irvine, 2017

Professor Ender Ayanoglu, Chair

The number of connected wireless devices is anticipated to increase heavily in the next few years. Thus, there is a need for a new system which is able to handle billions of wireless devices. The category of massive multiple input multiple output (MIMO) systems is a great candidate for this purpose. Because of the large number of antennas in massive MIMO there is a need to reduce the dimension of the MIMO channel effectively to decrease the complexity. This could be achieved by using a particular prebeamforming technique that is introduced recently. An important aspect of wireless communication systems is the channel state information (CSI). In order to send and receive data through the channel, the transmitter and the receiver must know the CSI or at least have an estimation for it. In this thesis, different algorithms for estimating the channel vector coefficient and their performance are studied. Different approaches are used in order to find the best algorithm based on the performance of the estimating channel and the complexity of the algorithm. Also, algorithms are used to estimate the channel vector coefficient for different channel models.

Chapter 1

Introduction

1.1 Why 5G?

In the past few years there has been a lot of discussion and interest about the next generation of mobile systems (5G) and what it should be. The new generation should not just be a better version of 4G, but it should have new features which 4G does not possess. Indeed, 5G will need to be a paradigm shift that includes very high carrier frequencies with massive bandwidths, extreme base station and user device densities, and unprecedented numbers of antennas [1]. According to the annual Visual Network Index (VNI) reports released by Cisco Systems, Inc, there is quantitative evidence that the wireless data explosion is real and will continue. Driven largely by smartphones, tablets, and video streaming, the most recent (Jun. 2016) VNI report and traffic forecast [2] makes the projection that the annual global IP traffic will surpass the zettabyte (ZB) threshold in 2016, and will reach 2.3 ZB by 2020. Currently video is the main part of this data deluge (and it is anticipated that 82% of all IP traffic will be video by 2020), but new applications are expected to use a higher amount of data in the near future. Also, the number of users and devices will potentially grow exponentially

in the coming years which could reach more than tens of billions by 2020. In addition to highly visible demand for ever more network capacity, there are a number of other factors that make 5G more interesting, including the potentially disruptive move to millimeter wave (mm wave) spectrum, new market driven ways of allocating and re-allocating bandwidth, a major ongoing virtualization on the core network that might progressively spread to the edges, and the possibility of an "Internet of Things" comprised of billions of miscellaneous devices. We can comment that there will be "big three" 5G technologies: ultra-densification, mm wave, and massive MIMO. The first two are out of scope of this work and we will concentrate on only the massive MIMO part.

1.2 Massive MIMO

Stemming from research that blossomed in the late 1990s [3], [4], MIMO communication was introduced into Wi-Fi systems around 2006 and into 3G cellular shortly thereafter. Well-established by the time 4G standard LTE was developed, MIMO was quickly incorporated with two-to-four antennas per mobile device and as many as eight per base station (BS) sector, and it appeared that, because of form factors and other apparent limitations, this was expected to be the extent to which MIMO could be leveraged. Marzetta was instrumental in articulating a vision in which the number of antennas increased by more than an order of magnitude, first in a 2007 presentation [5] with the details formalized in a landmark paper [6]. The proposal was to equip BSs with a number of antennas much larger than the number of active users per time-frequency signaling resource. Given that under reasonable time-frequency selectivities, accurate channel estimation can be conducted for at most some tens of users per resource, the fact that this condition puts the number of antennas per BS into hundreds is somewhat surprising. This bold idea was initially termed "massive antenna systems" but now more popularly known as "massive MIMO."

Since the number of antennas at a BS is typically assumed to be significantly larger than the number of users, a large number of degrees of freedom are available and can be used to shape the transmitted signals in a hardware-friendly way or to null interference [7]. In order to make such a system practical, algorithms for massive MIMO systems are required to keep the complexity low.

Another advantage of massive MIMO lies in its potential energy efficiency compared to a corresponding single antenna system. Is is shown in [8] that each single-antenna user in a massive MIMO system can scale down its transmit power proportionally to the number of antennas at the BS with perfect CSI or to the square root of the number of BS antennas with imperfect CSI, to get the same performances as a corresponding single-input single output (SISO) system.

Commercial use for the massive MIMO systems is anticipated to be available in the coming years to respond to a huge increase in the number of devices connected to the Internet [9, 10]. There is a need to increase the capacity and data rate (e.g., 1000 times faster) to meet the desired system [1, 9]. One of the massive MIMO advantages is its energy efficiency. The power consumption at the BS can be scaled down by the proportion of the number of antennas in the BS (with perfect CSI). As we can see, CSI have an important role in the energy efficiency. Also, to emphasize the benefits with large gains in spectral efficiency instantaneous CSI is inevitable [11].

As mentioned in [11], CSI could be obtained by periodically inserted pilot signals. So the pilot overhead which is used by the training data is proportional to the number of BS antennas for downlink training and to the number of active users in the system for uplink training[12].

We assume instantaneous CSI at the BS can be acquired by means of uplink training in time division duplex (TDD) mode, where the uplink pilots provide the BS with downlink as well as uplink channel estimates simultaneously via leveraging channel reciprocity [11, 12, 18].

Some of limiting factors for the accurate channel acquisition especially in high mobility are

the working with the high dimensional signals, the pilot interference and the pilot overhead. Also, in the case when signal-to-noise ration (SNR) before prebeamforming is very small, the pilot overhead will be large. So for supporting longer outdoor links directional precoding/beamforming is inevitable [13, 14].

1.3 Aim and Outline

The main contributions of the thesis are summarized as follows. First a model for channel state update is proposed based on a general framework on reduced dimensional channel state information (CSI) [18]. According to the new model for channel coefficient vector update three different methods are introduced for estimating the channel. Then by simulating the channel for different cases, the minimum square error (MSE) and capacity for each method is computed. Comparing these methods from different aspects such as complexity and performance is the next step of the thesis. Also, three different algorithms are studied in two new channel models. Then by comparing the results, the best performance is identified by means of simulations.

Chapter 2

System Model

2.1 Model

We consider a massive MIMO system in which an N-antennas BS communicates with K single-antenna UTs, operating at mm wave bands in the TDD mode employing SC modulation. User are categorized into G different groups, where K_g is the number of users in group g. These users have statistically independent but identically distributed (i.i.d.) channel [15, 16, 18], and transmit training sequence with length T at the beginning of every coherent interval.

We assume a linear modulation (e.g., QAM or PSK) and transmission over a frequency selective channel for all UTs with a slow evolution in time relative to the signaling interval (symbol duration). Assuming these conditions, the baseband equivalent received signal samples, taken at symbol rate (W) after pulse matched filtering, are expressed as [18]

$$\mathbf{y}_n = \underbrace{\sum_{\{k=1,g_k\in\Omega_g\}}^{K_g}\sum_{l=0}^{L_g-1}\mathbf{h}_l^{(g_k)}x_{n-l}^{(g_k)}}_{\text{Intra-Group Signal}} + \underbrace{\sum_{\{\forall g'_k\in\Omega_{g'}|g'\neq g\}}\left(\sum_{k=1}^{K_{g'}}\sum_{l=0}^{L_{g'}-1}\mathbf{h}_l^{(g'_k)}x_{n-l}^{(g'_k)}\right)+\mathbf{n}_n}_{\boldsymbol{\eta}_n^{(g)}\text{:Inter-Group Interference + AWGN}} \tag{2.1}$$

for $n = 0, ..., T-1$, where $\{x_n^{(g_k)} : -L_g + 1 \leq n \leq T-1\}$ are the training symbols for the k^{th} user in group g and $\mathbf{h}_l^{(g_k)}$ is the $N \times 1$ multi-path channel vector, namely, the array impulse response of the serving BS stemming from the l^{th} multi-path component (MPC) of k^{th} user in group g.

Also Ω_g is the set of all UTs belonging to group g with cardinality $|\Omega_g| = K_g$, L_g is the channel memory of group g multi-path channels, and $\{g_k\}_{k=1}^{K_g}$ are UT indices which form Ω_g. We select the training symbols from a signal constellation $S \in \mathbb{C}$ and $\mathbb{E}\{|x_n^{(g_k)}|^2\}$ is set to E_s for all g_k [18].

In (2.1), $\mathbf{n}_n$ are the additive white Gaussian noise (AWGN) vectors during uplink pilot segment with spatially i.i.d. as $\mathcal{CN}(\mathbf{0}, N_0\mathbf{I}_N)$, and N_0 is the noise power.

The first term of (2.1) is the transmitted signal of the intended group g, which is labeled as the *intra-group* signal of group g users. The second term, $\boldsymbol{\eta}_n^{(g)}$, is labeled as the *inter-group interference*, includes of all the interfering signals, which stem from all inner or outer cell users belonging to different groups other than g.

Finally, the average received SNR can be defined as $SNR \triangleq \frac{E_s}{N_0}$. We assume users come in groups, either by nature or by the application of the proper *user grouping* algorithms in [16, 22], which are out of scope of this work.

In [18], the training matrix (or the convolution matrix), comprising of the transmitted pilots with the precursors for kth user in group g is defined as $\mathbf{X}_k^{(g)} = [x_{i-j}^{(g_k)}]$ and the complete training matrix that consists of the training data of all users in group g during the signaling interval T is given by

$$\mathbf{X}^{(g)} \triangleq [\mathbf{X}_1^{(g)}\ \mathbf{X}_2^{(g)}\ ...\ \mathbf{X}_{K_g}^{(g)}]. \tag{2.2}$$

At the the pre-beamforming stage, a DT-dimensional space-time vector $y^{(g)}$ can be formed based on (2.1) for all intra-cell groups by a linear transformation through a matrix $\left(\boldsymbol{\Upsilon}_S^{(g)}\right)^H \triangleq$

$\left(\mathbf{I}_T \otimes [\mathbf{S}_D^{(g)}]^H\right)$ as

$$\mathbf{y}^{(g)} = \left(\boldsymbol{\Upsilon}_S^{(g)}\right)^H \mathbf{y} = (\mathbf{X}^{(g)} \otimes [\mathbf{S}_D^{(g)}]^H)\mathbf{h}^{(g)} + \left(\boldsymbol{\Upsilon}_S^{(g)}\right)^H \boldsymbol{\xi} = \left(\boldsymbol{\Psi}^{(\mathbf{g})}\right)^H \mathbf{h}^{(g)} + \boldsymbol{\eta} \tag{2.3}$$

where $\mathbf{S}_D^{(g)}$ is an $N \times D$ statistical prebeamforming matrix that projects the N-dimensional received signal D samples $\{\mathbf{y}_n\}_{n=0}^{T-1}$ in (2.1) onto a suitable D-dimensional subspace in spatial domain and $\boldsymbol{\eta}$ is the intergroup interference after beamforming. The details can be found in [18].

We assume that the channel is block-fading and that the channel remains constant during the l-th block. The channel temporal variation across the blocks is modeled using a state-space framework as a first-order stationary Gauss-Markov process [25] with

$$\mathbf{h}_n = \sqrt{\alpha}\mathbf{h}_{n-1} + \sqrt{1-\alpha}\mathbf{b}_n \tag{2.4}$$

where $\alpha \in (0, 1]$ is the temporal fading coefficient and $\mathbf{b}_n \sim \mathcal{CN}(\mathbf{0}, \mathbf{R_h})$ is the process noise. The channel spatial correlation is given by

$$\mathbf{R_h} = \mathbb{E}\{\mathbf{h}_l \mathbf{h}_l^H\} = \alpha \mathbf{R_h} + (1-\alpha)\mathbf{R_b} \tag{2.5}$$

where $\mathbf{R_b}$ is easily calculated as $\mathbf{R_b} = E[\mathbf{b}_n \mathbf{b}_n^H] = \mathbf{R_h}$

Based on these assumptions we develop algorithms to estimate the channel with the pre-beamforming technique and then by simulating the algorithms, we can find the optimum D, which can be different for different methods. In the remaining chapters, the emphasis is on deriving efficient ways to estimate the channel and then by using different tools, we will provide the minimum beamspace dimension (D) required for the dimension reduction. This dimension could be small.

Chapter 3

A Kalman Filter Implementation of the Reduced-Rank MMSE Estimator

We will begin with a description of the standard Kalman filter which is also known as linear quadratic estimation (LQE), specified with respect to the model described in (2.4). This algorithm uses observed data and measurements over time containing noise and based on these data it tries to produce estimates of unknown variables. The Kalman filter is a recursive estimator which means that only the estimated state from the previous time step and the current measurement are needed to compute the estimate for the current state. This is a good property of the algorithm but the main drawback of this algorithm is its complexity which is caused by matrix inversion. We also assume that the beamforming block is to input our signal. For simplicity we will neglect the subscript g in the following. The assumption which users are in group g is still present. Our goal is to find a dimension reduction of the estimator. The dimension for a MIMO system with N antennas is N, where N can be very large (~ 100). We want to reduce this dimensionality to D where D is small enough to decrease the computational cost of calculations. In some cases D could be very small ($D = 1, 2$).

3.1 Algorithm

By using (2.4) one can write the Kalman filter model, using the standard derivation as follows.

Initialization

$$\hat{\mathbf{h}}_{0|-1} = 0 \quad \text{and} \quad \mathbf{P}_{0|-1} = \mathbf{R}_h \tag{3.1}$$

Prediction

$$\text{A priori State Estimate:} \quad \hat{\mathbf{h}}_{n|n-1} = \sqrt{\alpha}\hat{\mathbf{h}}_{n-1|n-1} \tag{3.2}$$

$$\text{A priori Estimate Covariance:} \quad \mathbf{P}_{n|n-1} = \alpha\mathbf{P}_{n-1|n-1} + (1-\alpha)\mathbf{R}_\eta \tag{3.3}$$

Update

$$\begin{aligned} \text{Innovation:} \quad \mathbf{z}_n &= \mathbf{y}_n - (\mathbf{X}_n^H \otimes \mathbf{S}_D^H)\hat{\mathbf{h}}_{n-1|n-1} \\ &= \mathbf{y}_n - \mathbf{\Psi}_n^H \hat{\mathbf{h}}_{n-1|n-1} \end{aligned} \tag{3.4}$$

$$\text{Innovation Covariance:} \quad \mathbf{S}_n = \mathbf{\Psi}_n^H \mathbf{P}_{n|n-1}\mathbf{\Psi}_n + \mathbf{S}_D^H \mathbf{R}_\eta \mathbf{S}_D \tag{3.5}$$

$$\text{Kalman Gain:} \quad \mathbf{K}_n = \mathbf{P}_{n|n-1}\mathbf{\Psi}_n\mathbf{S}_n^{-1} \tag{3.6}$$

$$\text{A Posteriori State Estimate:} \quad \hat{\mathbf{h}}_{n|n} = \hat{\mathbf{h}}_{n|n-1} + \mathbf{K}_n\mathbf{z}_n \tag{3.7}$$

$$\text{A Posteriori Estimate Covariance:} \quad \mathbf{P}_{n|n} = (\mathbf{I} - \mathbf{K}_n\mathbf{\Psi}_n)\mathbf{P}_{n|n-1} \tag{3.8}$$

where $\mathbf{X}_n$ is the n^{th} row of the training matrix $\mathbf{X}$

3.2 Simulation Results

Based on the proposed system model and algorithm in chapter 2 and this chapter, we provide numerical results to evaluate the performance of the Kalman filtering. In the preparation of the results reported below, the following parameter setting were used. We consider a massive MIMO system in TDD mode with one BS and K users where each user has a single receive antenna. By using the uplink training, we try to estimate the channel coefficient vector. Also, the BS is equipped with a uniform linear array (ULA) of $N = 100$ antenna elements along y-axis. For grouping the users, they were clustered into eight different groups ($G = 8$). The location of each UT is at a specific azimuth angle θ along the ring centered at the origin in the $x - y$ plane. For grouping users one can use the proper user grouping algorithms in [16, 22]. In this work, we assume that users come in groups.

In the simulations, we assume that each of the 8 groups has 3 MPCs, i.e., $L_{g_i} = 3$. We only focus on the group g, which have two users serve simultaneously, i.e. $K_g = 2$. Other 7 groups consists of 3 users, i.e., $K_{g'} = 3$, $g' \neq g$ where these users have interference with the intended group g. Also the first two MPCs of the group g stem from an azimuth angular sector $[-1°, 1°]$ for delays at $l = 0, 1$. For the last MPC at $l = 2$ in group g, the angular sector is $[5°, 7°]$ in azimuth. The angular sector for other 7 groups are given by $[-29°,-26°], [-21°,-19°]$, $[-12°,-9°]$, $[-5.5°,-3.5°]$, $[9.5°, 12.5°]$, $[15°, 17°], [24°, 27°]$ in azimuth respectively [18]. Also the noise power is set as $N_0 = 1$ so that all dB power values are relative to 1. The channel covariance matrix of each group can be calculated via same ways used in [15, 17].

For calculating the minimum square error (MSE), we used the 2-norm definition for each dimension

$$e_n = ||\mathbf{h}_n - \hat{\mathbf{h}}_n||^2 \qquad n = 1, 2, ..., T \tag{3.9}$$

In the following figures the result of the simulations are presented. First set of figures (3.1-3.5) are generated for $\text{SNR}_{in} = 30$ dB. $\text{SNR}_{in} = 3$ dB is used in the second set (3.6-3.10). And for the last set of figures (3.11-3.15), $\text{SNR}_{in} = 0$ dB is used. As can be seen through Figure 3.1-3.4, 3.6-3.9 and 3.11-3.14 the MSE increases when we increase the dimension, but at some point, usually when $n > 50$ for $\text{SNR}_{in} = 30$ dB, $n > 300$ for $\text{SNR}_{in} = 3$ and $n > 400$ for $\text{SNR}_{in} = 3$, the MSE value saturates. It is obvious that Increasing the input SNR leads to shorter saturation time. In these figures, the input SNR and the beamspace dimension is constant ($D = 8$) and in each figure, the temporal fading coefficient (α) changes. There is gap between the curves for beamspace dimension of one and other dimensions. The gap is increasing by increasing the value of α in both set of figures for different input SNRs. The important figures here are Figure 3.5, 3.10 and 3.15, which represent the MSE with respect to beamspace dimension. It is obvious that by increasing the channel coefficient from totally random ($\alpha = 0$) to a static channel which does not change with time ($\alpha = 1$) the performance of the Kalman filter should change. In the random case, as expected the performance is bad. The static channel has the best performance of estimating the channel. In these figures the final value of MSE is the mean of last 50 elements of error vector.

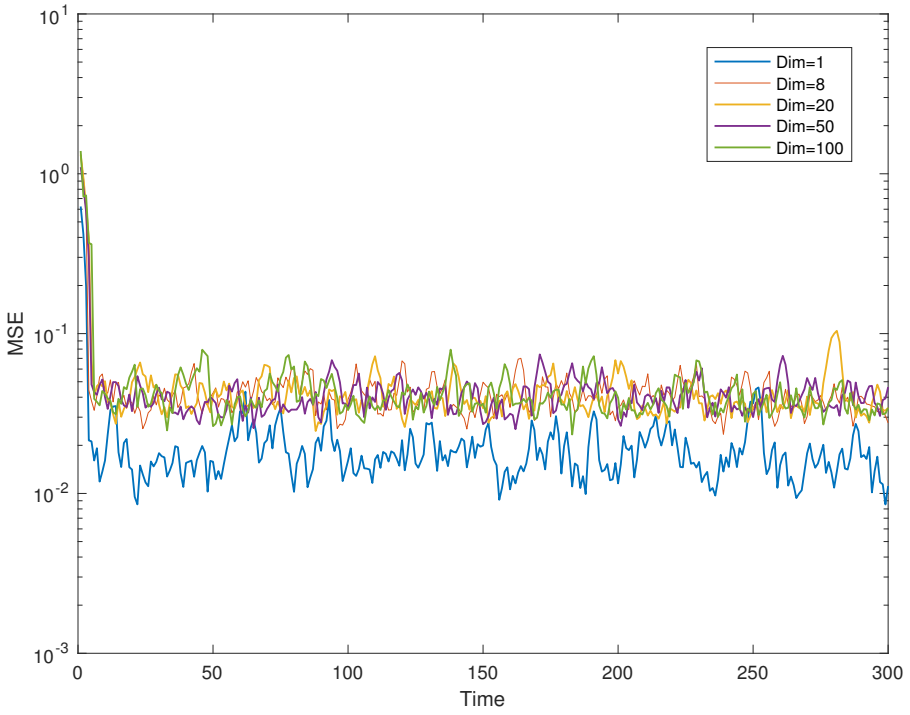

Figure 3.1: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.99$ and $\text{SNR}_{in} = 30$ dB)

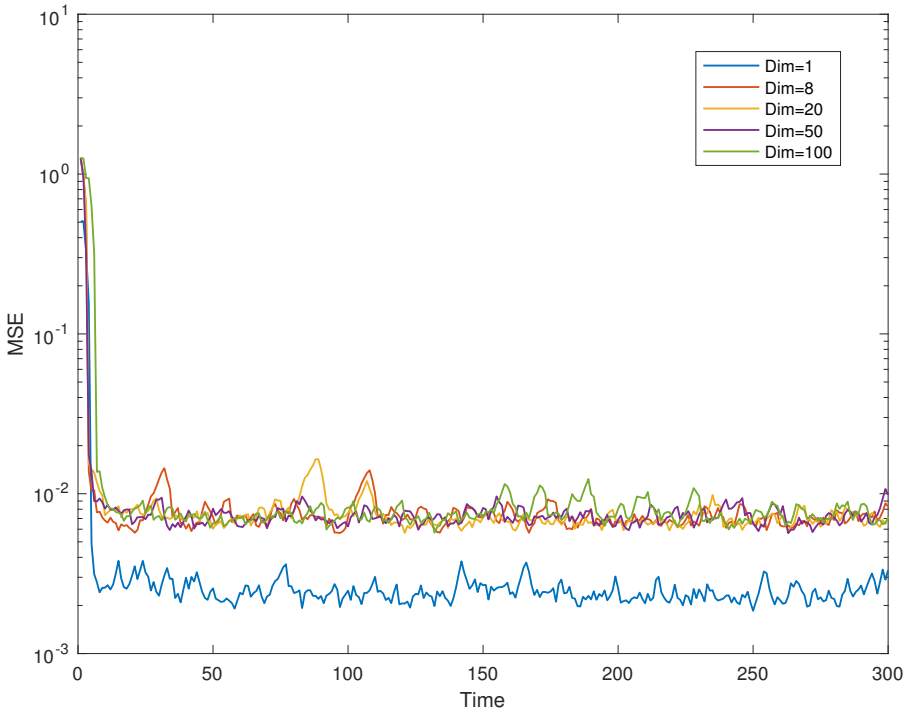

Figure 3.2: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.999$ and $\text{SNR}_{in} = 30$ dB)

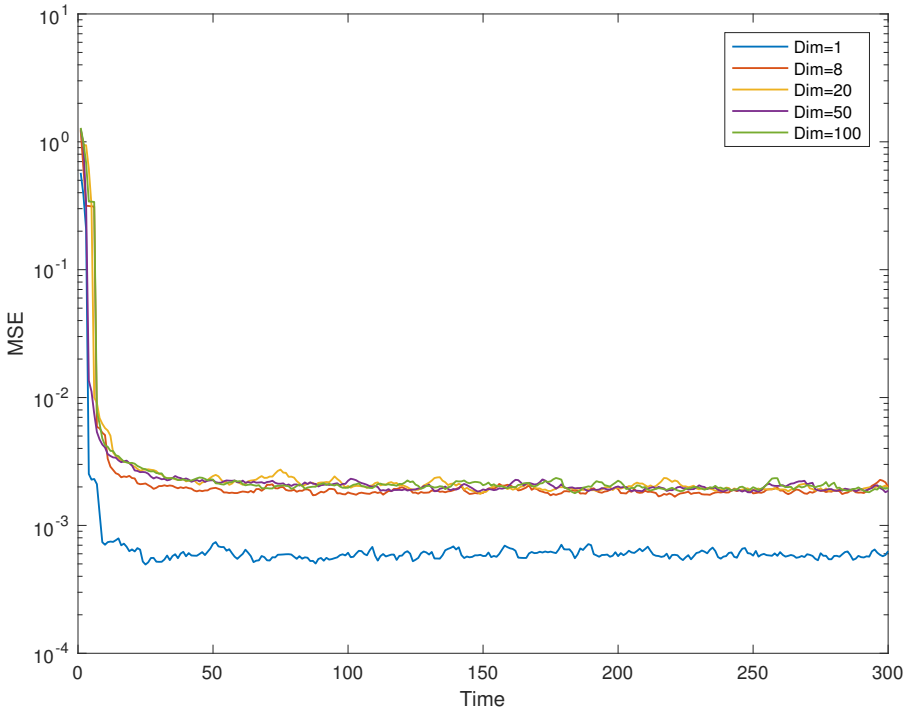

Figure 3.3: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.9999$ and $\text{SNR}_{in} = 30$ dB)

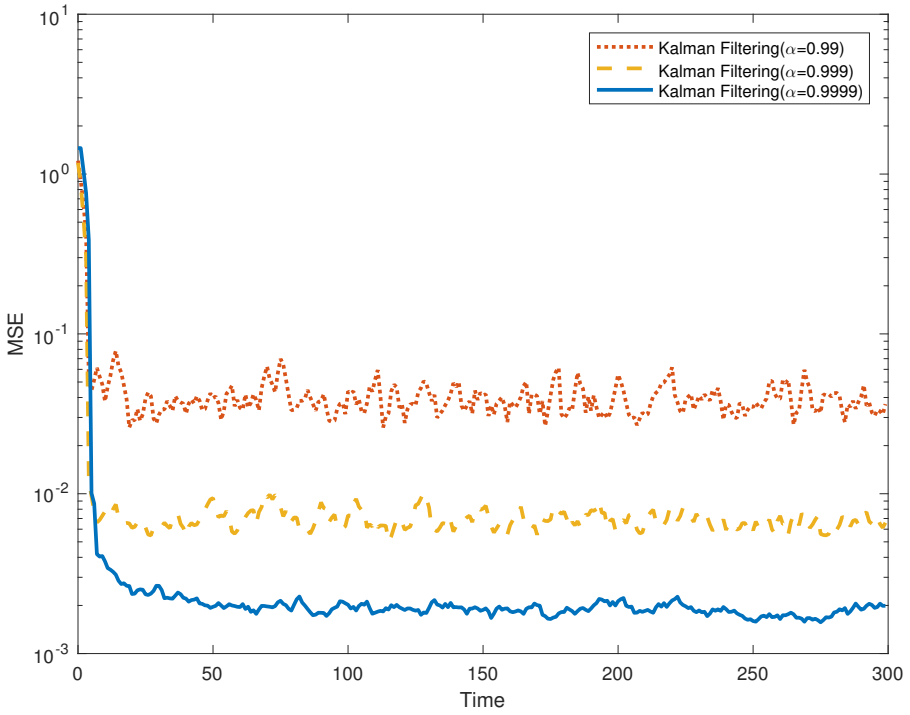

Figure 3.4: MSE for different values of fading correlation coefficient α with Kalman filtering ($Dim = 8$ and $\text{SNR}_{in} = 30$ dB)

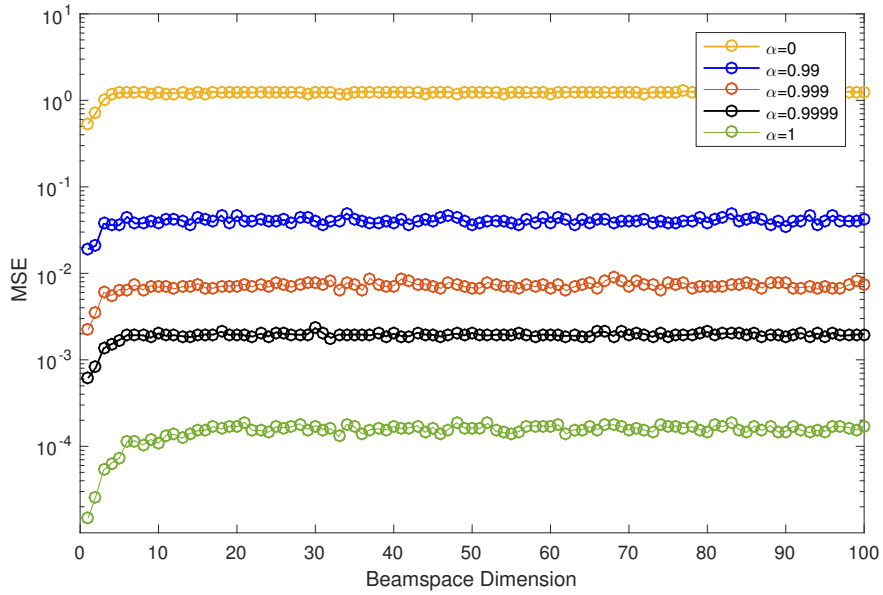

Figure 3.5: MSE for different values of fading correlation coefficient α with Kalman filtering ($\text{SNR}_{in} = 30$ dB)

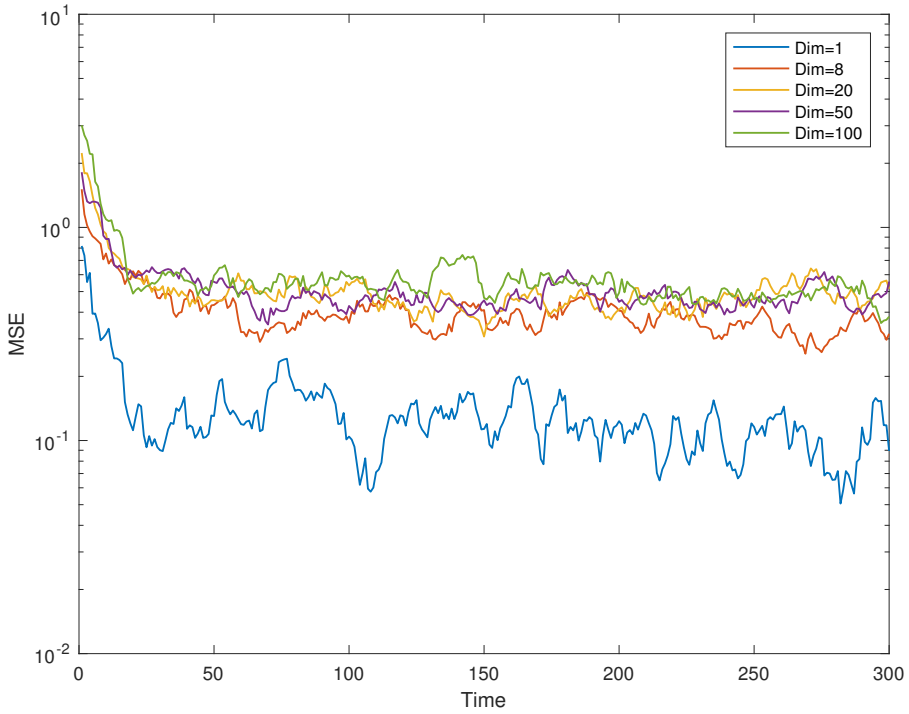

Figure 3.6: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.99$ and $\text{SNR}_{in} = 3$ dB)

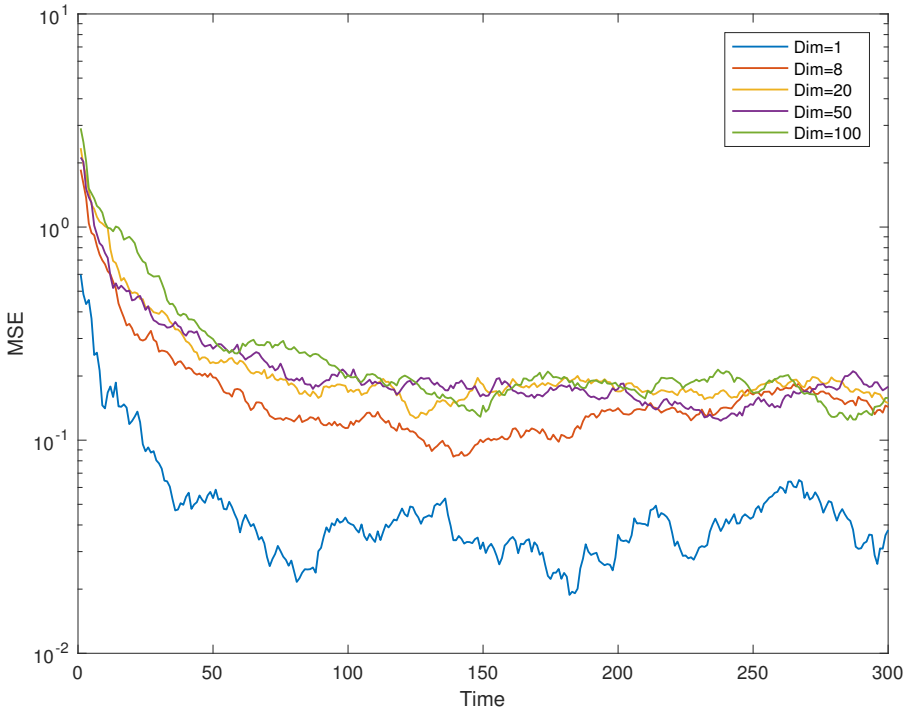

Figure 3.7: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.999$ and $\text{SNR}_{in} = 3$ dB)

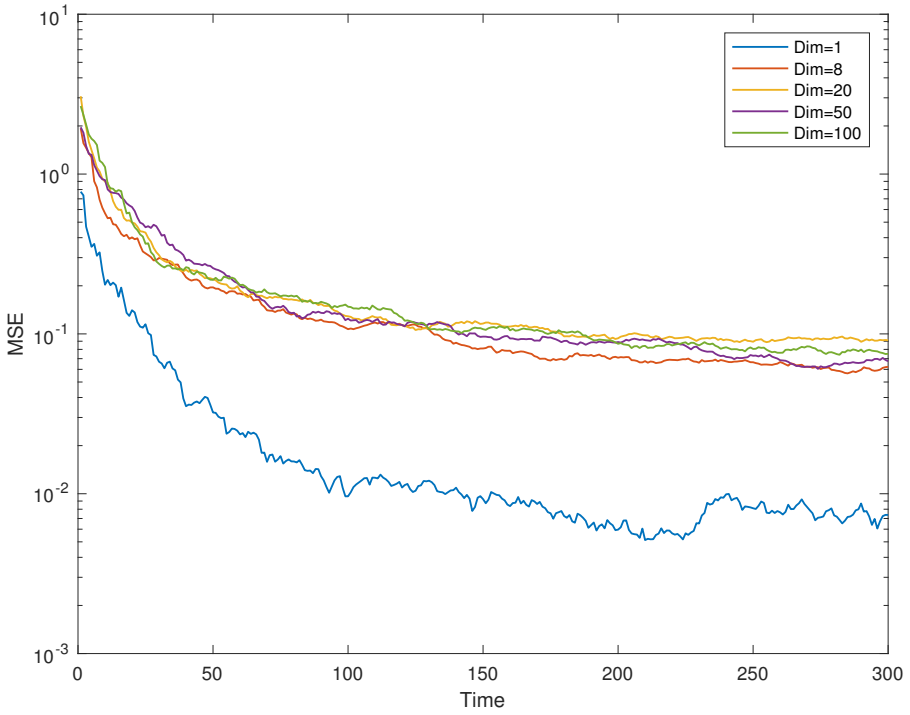

Figure 3.8: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.9999$ and $\text{SNR}_{in} = 3$ dB)

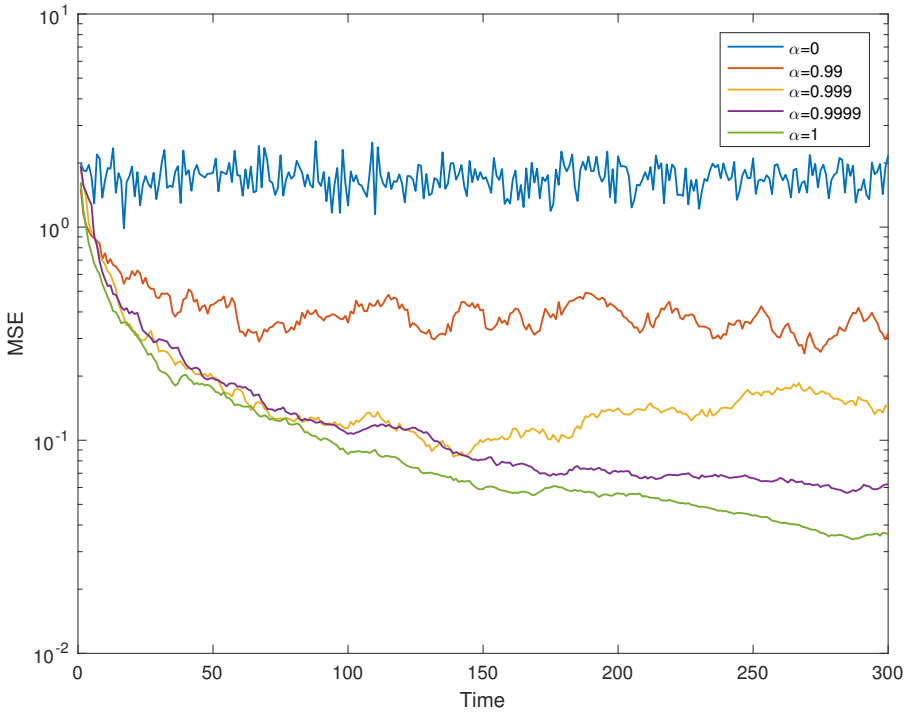

Figure 3.9: MSE for different values of fading correlation coefficient α with Kalman filtering ($Dim = 8$ and $\text{SNR}_{in} = 3$ dB)

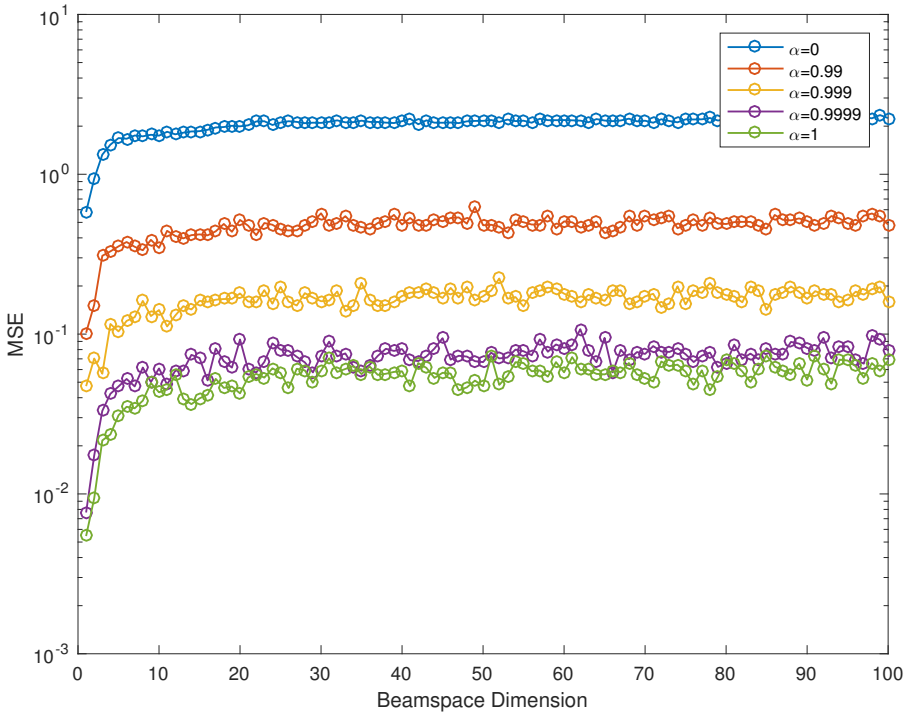

Figure 3.10: MSE for different values of fading correlation coefficient α with Kalman filtering ($\text{SNR}_{in} = 3$ dB)

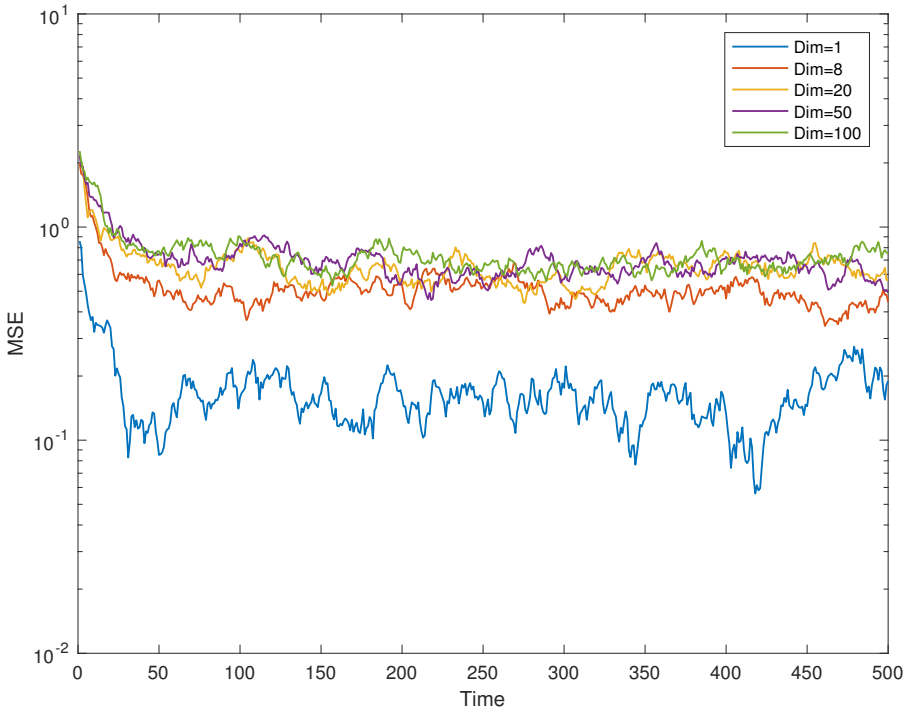

Figure 3.11: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.99$ and $\text{SNR}_{in} = 0$ dB)

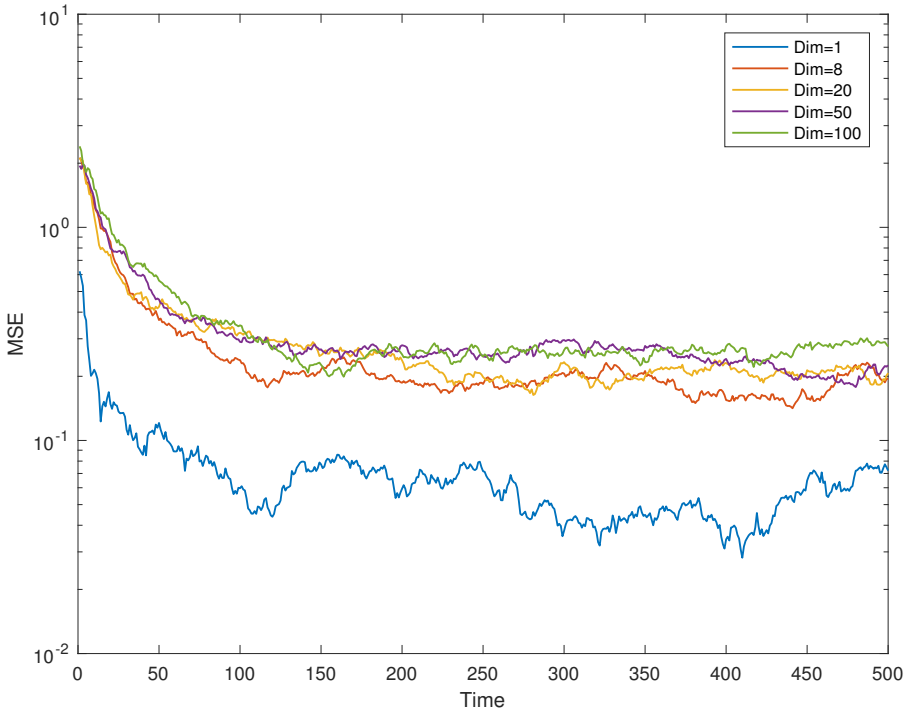

Figure 3.12: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.999$ and $\text{SNR}_{in} = 0$ dB)

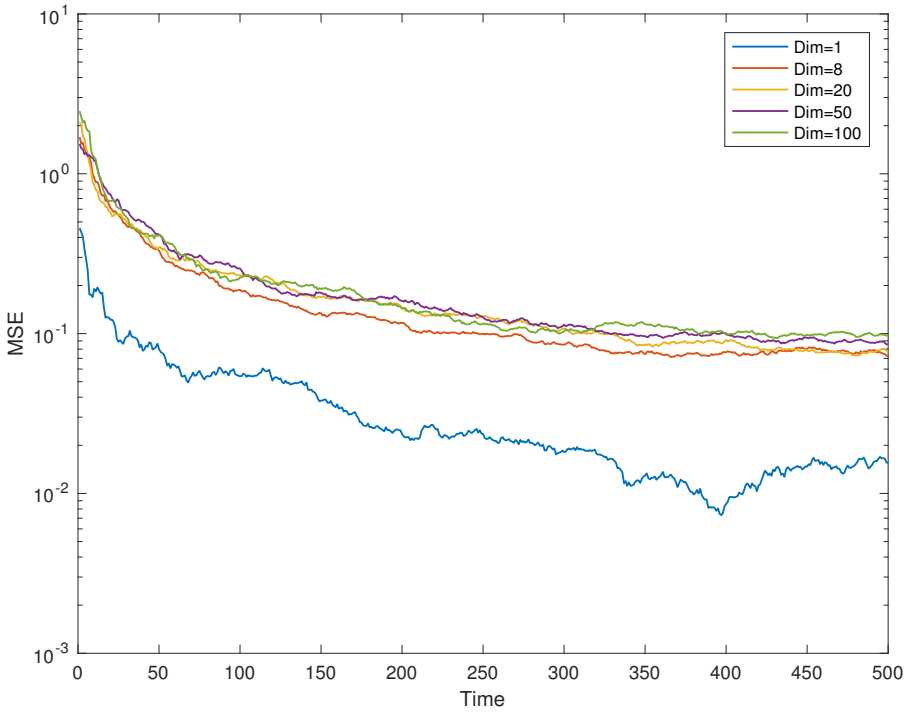

Figure 3.13: MSE for different values of beamspace dimension (Dim) with Kalman filtering ($\alpha = 0.9999$ and $\text{SNR}_{in} = 0$ dB)

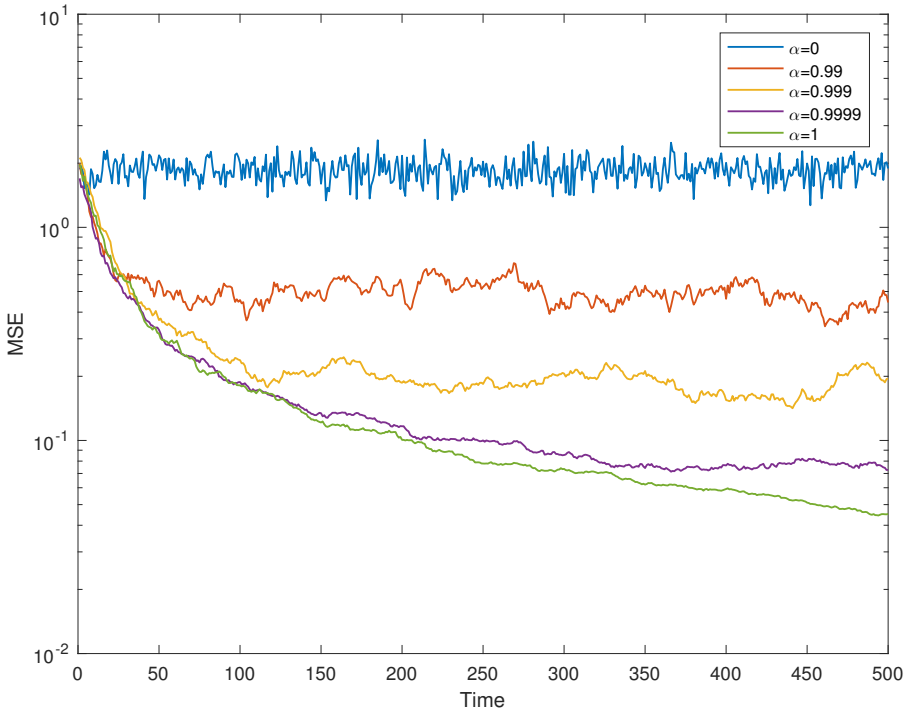

Figure 3.14: MSE for different values of fading correlation coefficient α with Kalman filtering ($Dim = 8$ and $\text{SNR}_{in} = 0$ dB)

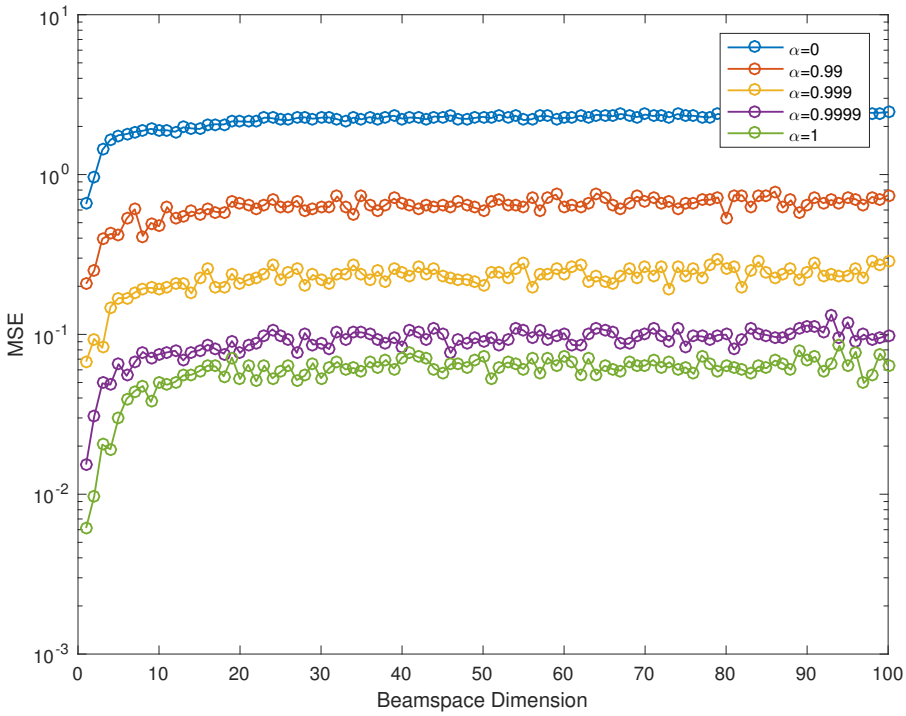

Figure 3.15: MSE for different values of fading correlation coefficient α with Kalman filtering ($\text{SNR}_{in} = 0$ dB)

Chapter 4

LMS Algorithm for Channel Estimation

We have another tool for estimating the channel. We can use the well-known Least Mean Square (LMS) algorithm. LMS is an adaptive filter which is used to find a desired filter by computing the filter coefficients that relate to producing the least mean square of the error signal (difference between the desired and the actual signal).

4.1 Deriving the Algorithm

However, the conventional LMS algorithm needs to be altered considering the subspace reduction, we define J_n as the cost function which we want to minimize over $\mathbf{h}_n$:

$$\arg\min_{\mathbf{h}_n} J_n = E\{||\mathbf{e}_n||^2\} = E\{||\mathbf{y}_n - \mathbf{\Psi}_n^H \mathbf{h}_n||^2\} \tag{4.1}$$

By using the general form of LMS algorithm which is

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} - \mu \nabla_{\mathbf{h}_n^H} J_n \tag{4.2}$$

one can easily calculate the minimum of cost function by using two above equations. In the calculation, we assume that $\hat{\mathbf{h}}_n \approx \hat{\mathbf{h}}_{n-1}$

$$\begin{aligned}
\hat{\mathbf{h}}_n &= \hat{\mathbf{h}}_{n-1} - \mu\nabla_{\mathbf{h}_n^H} J_n = \hat{\mathbf{h}}_{n-1} - \mu\nabla_{\mathbf{h}_n^H}[(\mathbf{y}_n^H - \mathbf{h}_n^H\boldsymbol{\Psi}_n)(\mathbf{y}_n - \boldsymbol{\Psi}_n^H\mathbf{h}_n)] \\
&= \hat{\mathbf{h}}_{n-1} - \mu(-\boldsymbol{\Psi}_n)(\mathbf{y}_n - \boldsymbol{\Psi}_n^H\hat{\mathbf{h}}_{n-1}) = \hat{\mathbf{h}}_{n-1} + \mu\boldsymbol{\Psi}_n(\mathbf{y}_n - \boldsymbol{\Psi}_n^H\hat{\mathbf{h}}_{n-1}) \\
&= \hat{\mathbf{h}}_{n-1} + \mu\boldsymbol{\Psi}_n\mathbf{e}_n \qquad (4.3)
\end{aligned}$$

where $\mathbf{e}_n = \mathbf{y}_n - \boldsymbol{\Psi}_n^H\hat{\mathbf{h}}_{n-1}$.

Note that in this derivation we removed the expectation operator. This is because of employing the "stochastic gradient" in deriving the LMS algorithm.

The main difference between the LMS algorithm and Kalman filter is that in Kalman filter there is matrix inversion. But in the LMS algorithm there is no need for this procedure and because of this, the LMS algorithm performs much faster than the Kalman filter.

4.2 Simulation Results

By using the setting in section 3.2, we simulated channel estimation via the LMS Algorithm and plotted the MSE with respect to time and beamspace dimension. Also, the channel fading correlation coefficient α was altered to figure out how the LMS behavior changes when it changes.

In Figure 4.1-4.3, the MSE for different values of beamspace dimension is plotted for $\text{SNR}_{in} =$ 30 dB. Figure 4.7-4.9 and 4.13-4.15 show the MSE for LMS algorithm for $\text{SNR}_{in} = 3$dB and $\text{SNR}_{in} = 3$dB, respectively. The best performance is for when the beamspace dimension is one and by increasing the dimension, the performance gets worse. Furthermore, after about 40 iterations, the algorithm converges to its final value in case $\text{SNR}_{in} = 30$dB. On the other

hand, the convergence time is around 250 iterations in Figure 4.7-4.9 and 4.13-4.15. In these figures, the time window increased, because of late convergence time of the LMS algorithm. The optimum value for the μ varies when the input SNR changes. Figure 4.6 shows the different values for MSE with respect to μ. In Figure 4.4, 4.10 and 4.16 the performance

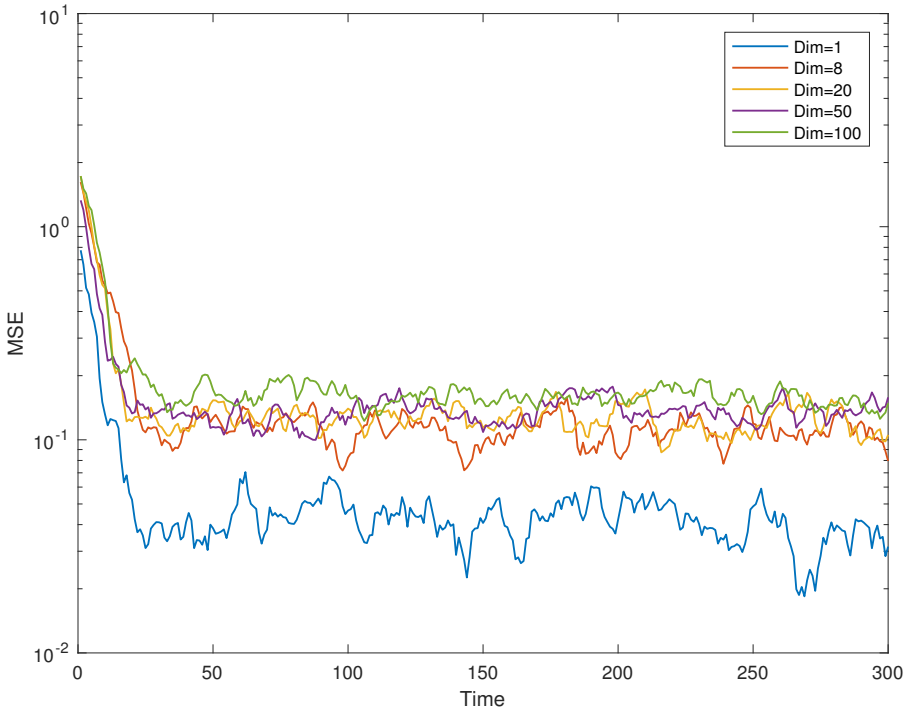

Figure 4.1: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.99, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

of the LMS algorithm gets better when the channel fading correlation coefficient α increases which is obvious because by increasing α the channel is more robust and so the algorithm can estimate the channel coefficient better.

Figure 4.5, 4.11 and 4.17 shows that the best performance for the LMS algorithm is when the channel is static which means $\alpha = 1$. In this case the channel state does not change with time and because of this, the noise coefficient in (2.4) becomes zero. This leads to the best performance of the LMS algorithm. Also, as can be seen in this figure, after some beamspace dimension, the MSE converges. This means that there is no need to increase the beamspace dimension after some point. However the best performance is when Dim=1. In this figure the final value of MSE is the mean of last 50 elements of error vector.

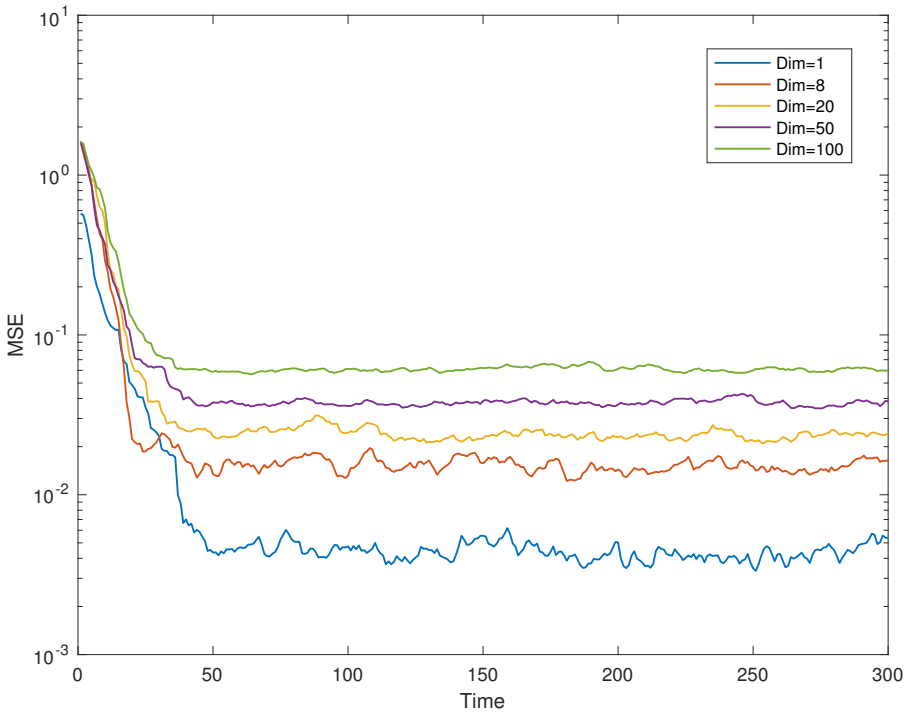

Figure 4.2: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.999, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

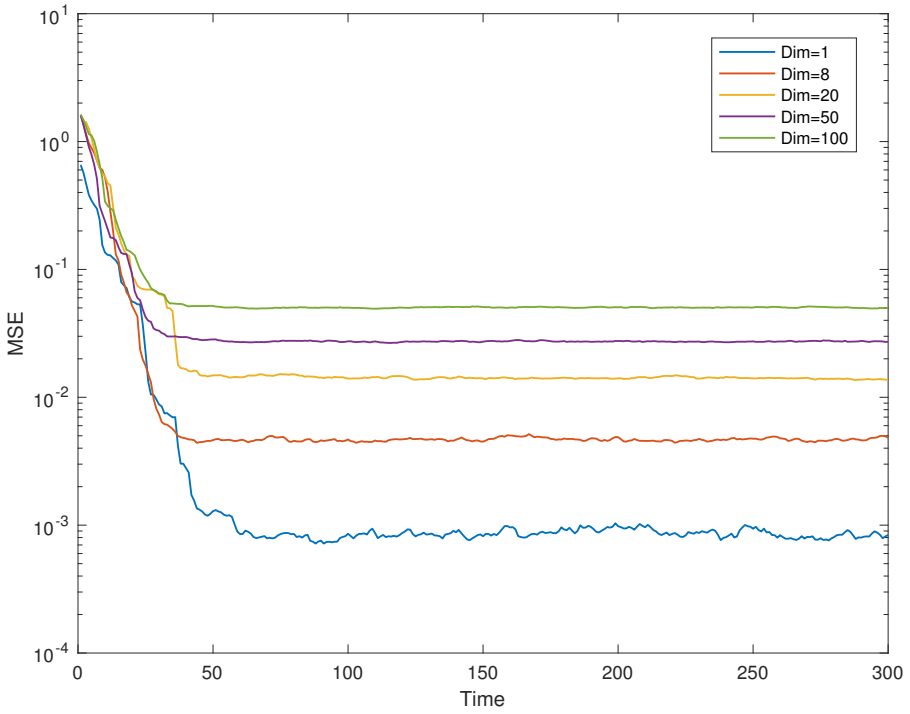

Figure 4.3: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.9999, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

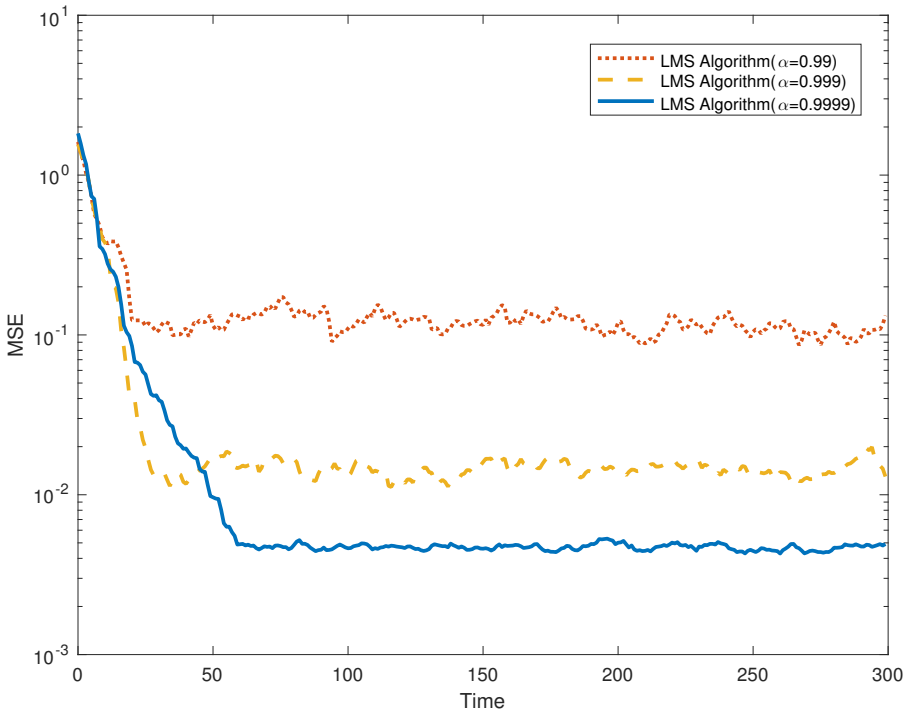

Figure 4.4: MSE for different values of fading correlation coefficient α with the LMS algorithm ($\mu = 0.1$, Dim= 8 and $\text{SNR}_{in} = 30$ dB)

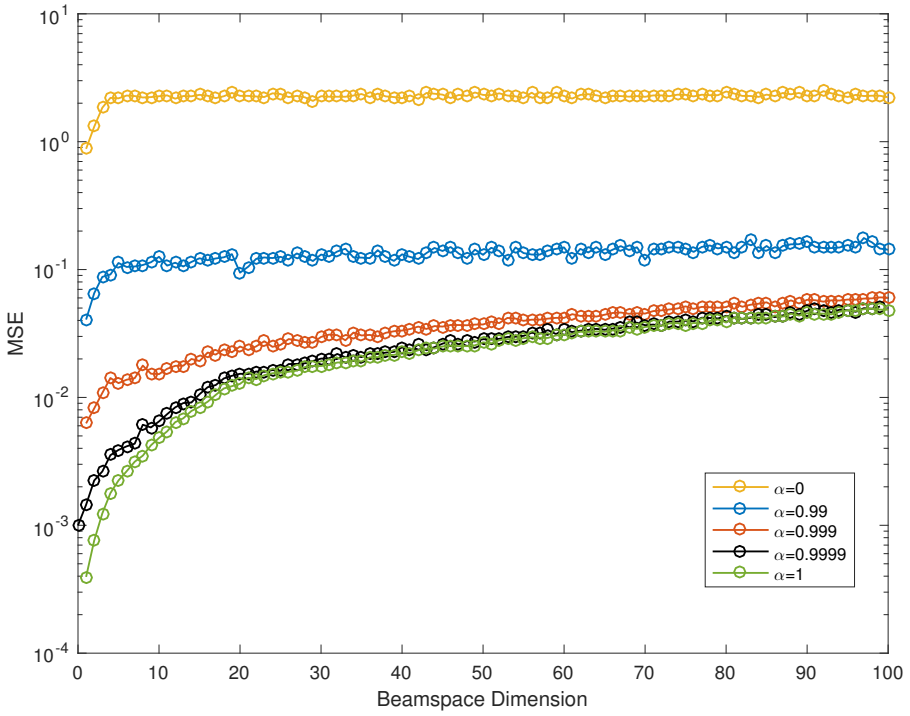

Figure 4.5: MSE for different values of fading correlation coefficient α with the LMS algorithm ($\mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

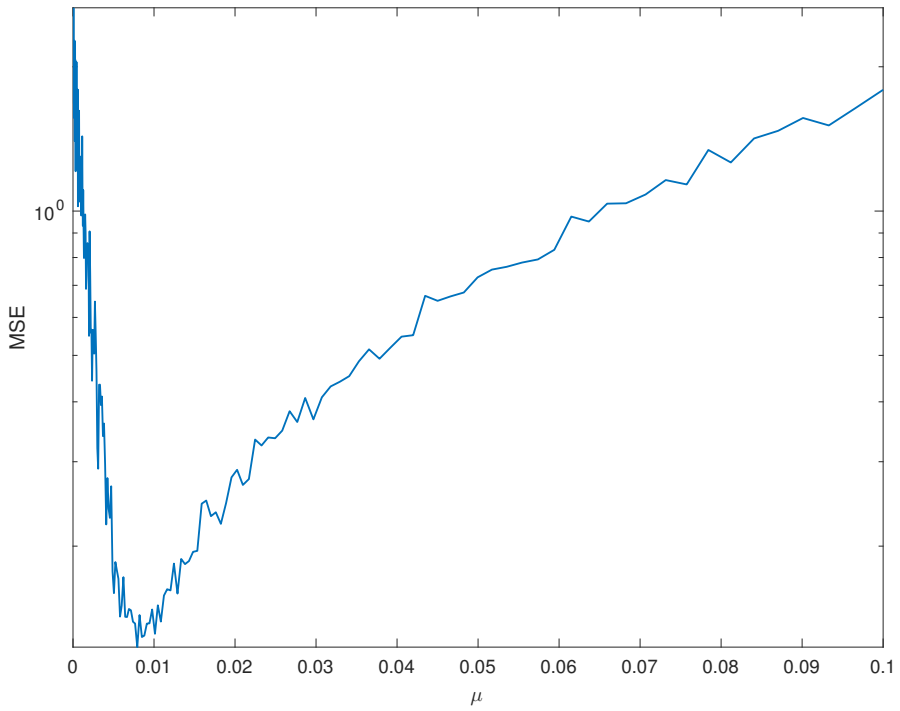

Figure 4.6: MSE for different values of μ with the LMS algorithm ($\alpha = 0.9999$ and $\text{SNR}_{in} = 3$ dB)

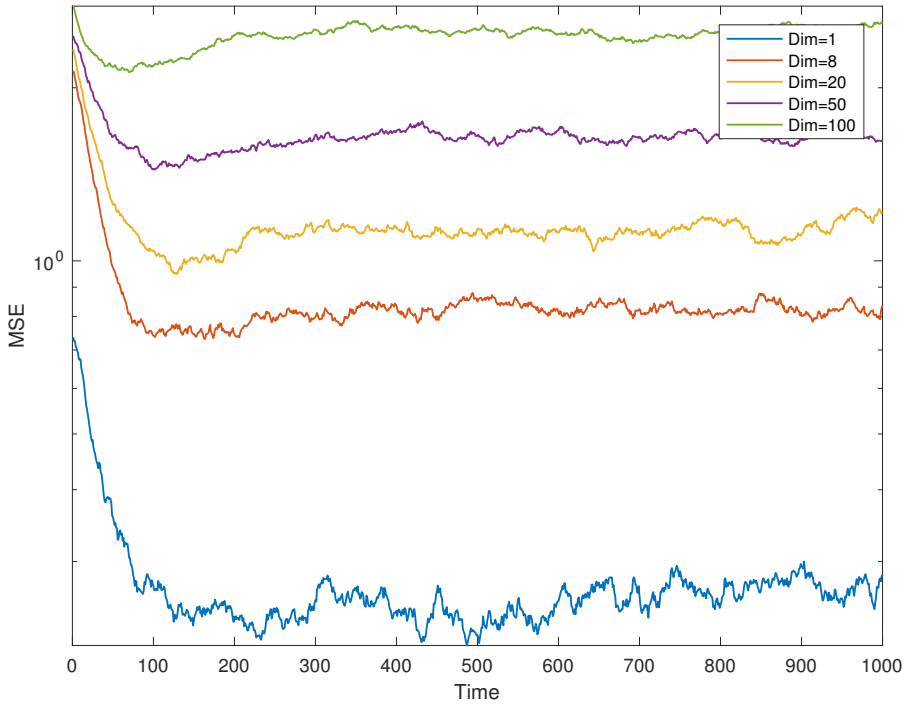

Figure 4.7: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.99, \mu = 0.01$ and $\text{SNR}_{in} = 3$ dB)

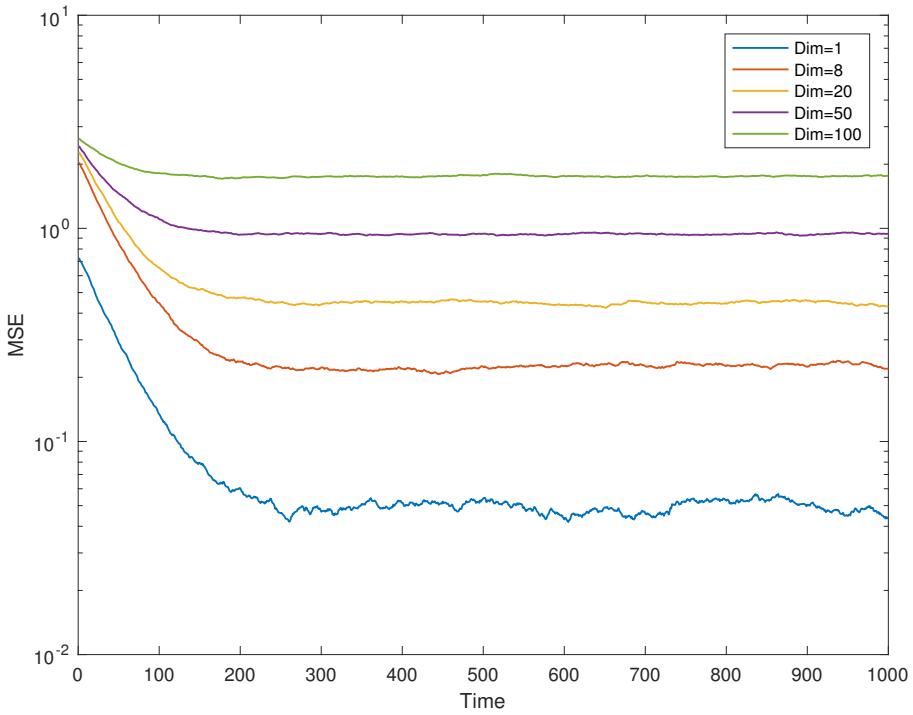

Figure 4.8: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.999, \mu = 0.01$ and $\text{SNR}_{in} = 3$ dB)

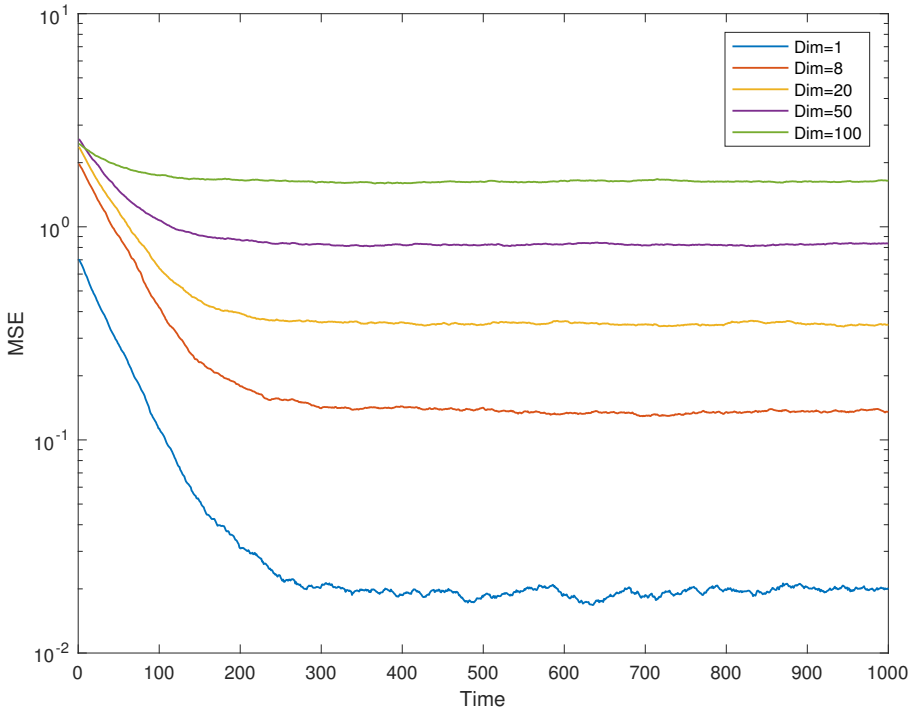

Figure 4.9: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.9999, \mu = 0.01$ and $\text{SNR}_{in} = 3$ dB)

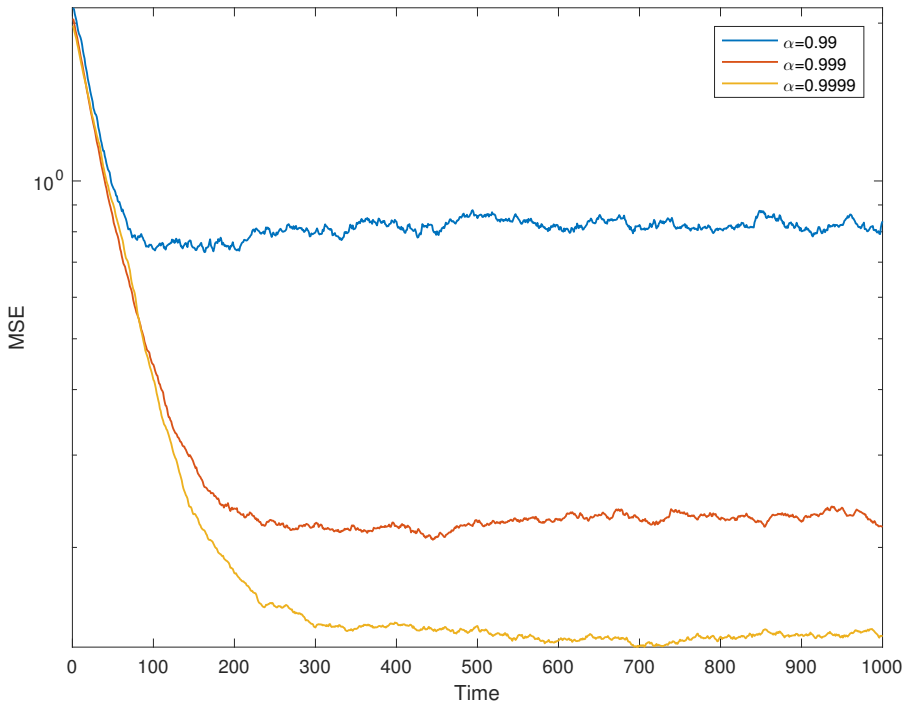

Figure 4.10: MSE for different values of fading correlation coefficient α with the LMS algorithm ($\mu = 0.01$, Dim= 8 and $\text{SNR}_{in} = 3$ dB)

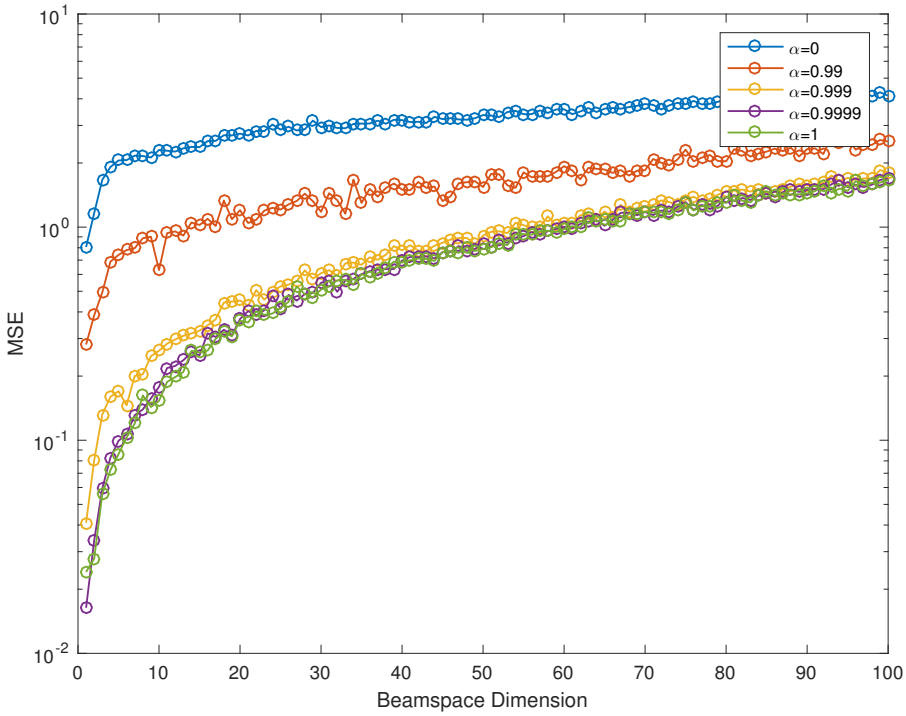

Figure 4.11: MSE for different values of fading correlation coefficient α with the LMS algorithm ($\mu = 0.1$ and $\text{SNR}_{in} = 3$ dB)

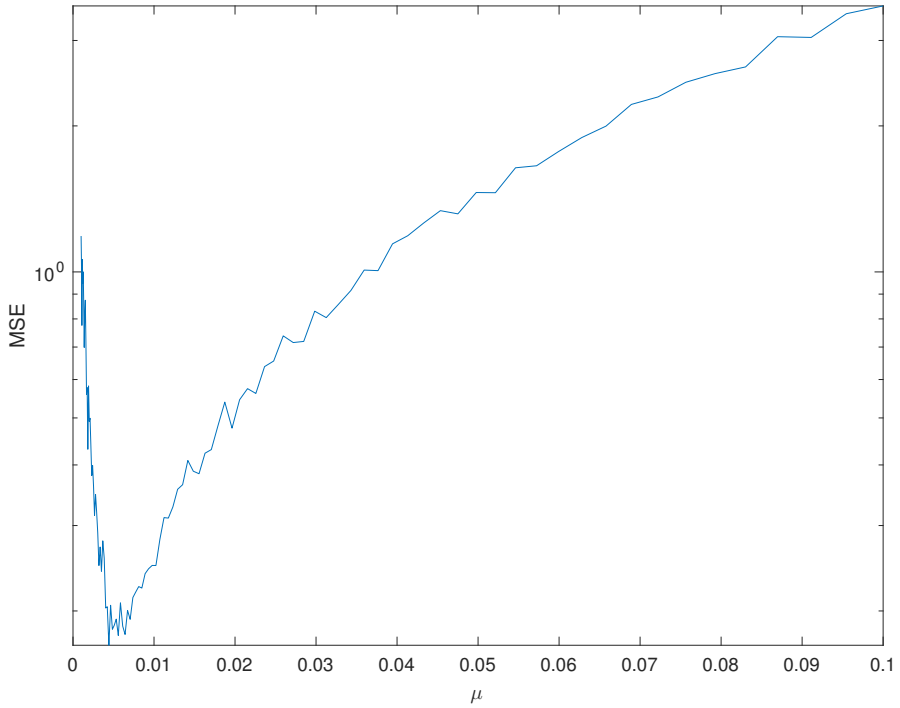

Figure 4.12: MSE for different values of μ with the LMS algorithm ($\alpha = 0.9999$, $\text{SNR}_{in} = 0$ dB and Dim=8)

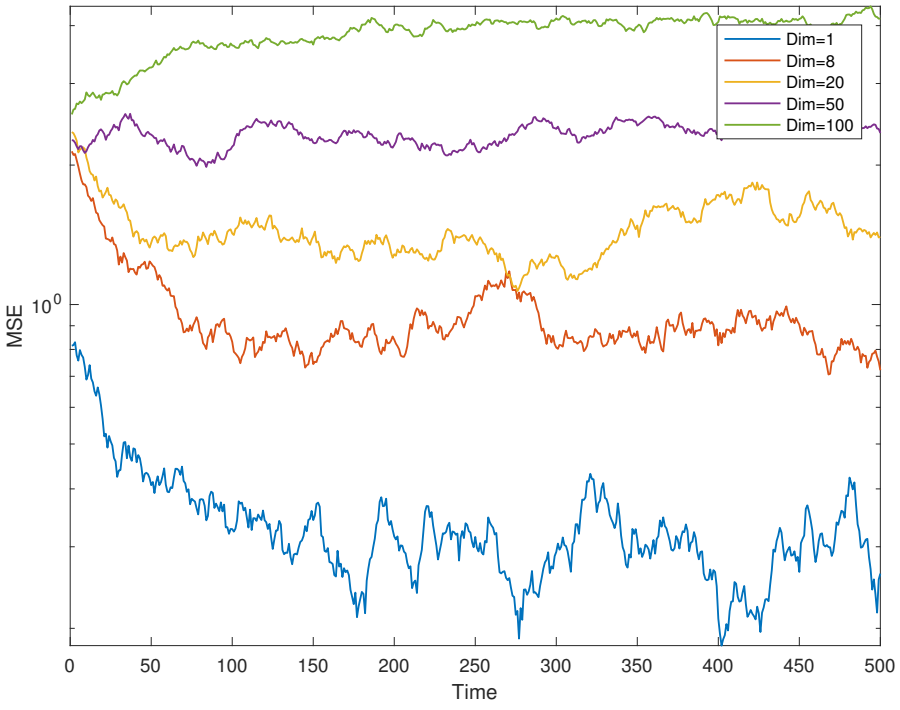

Figure 4.13: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.99$, $\mu = 0.01$ and $\text{SNR}_{in} = 0$ dB)

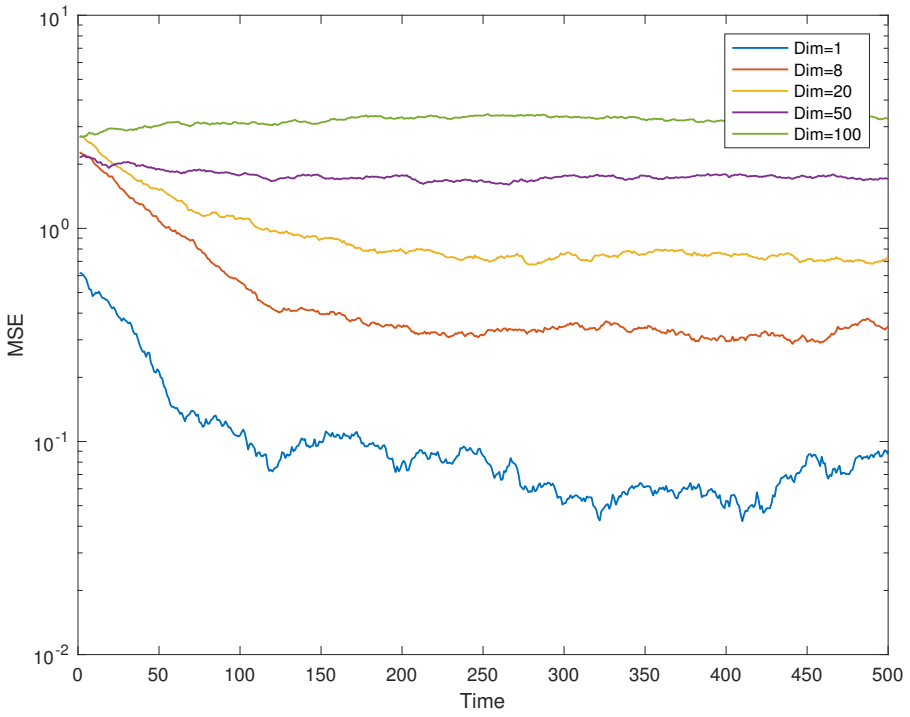

Figure 4.14: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.999, \mu = 0.01$ and $\text{SNR}_{in} = 0$ dB)

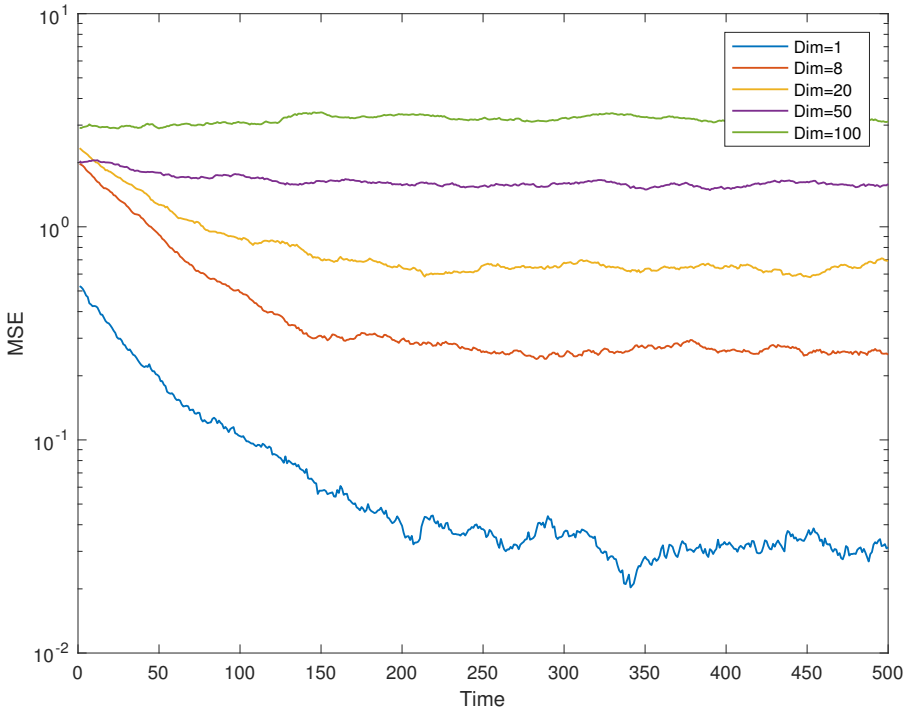

Figure 4.15: MSE for different values of beamspace dimension (Dim) with the LMS algorithm ($\alpha = 0.9999, \mu = 0.01$ and $\text{SNR}_{in} = 0$ dB)

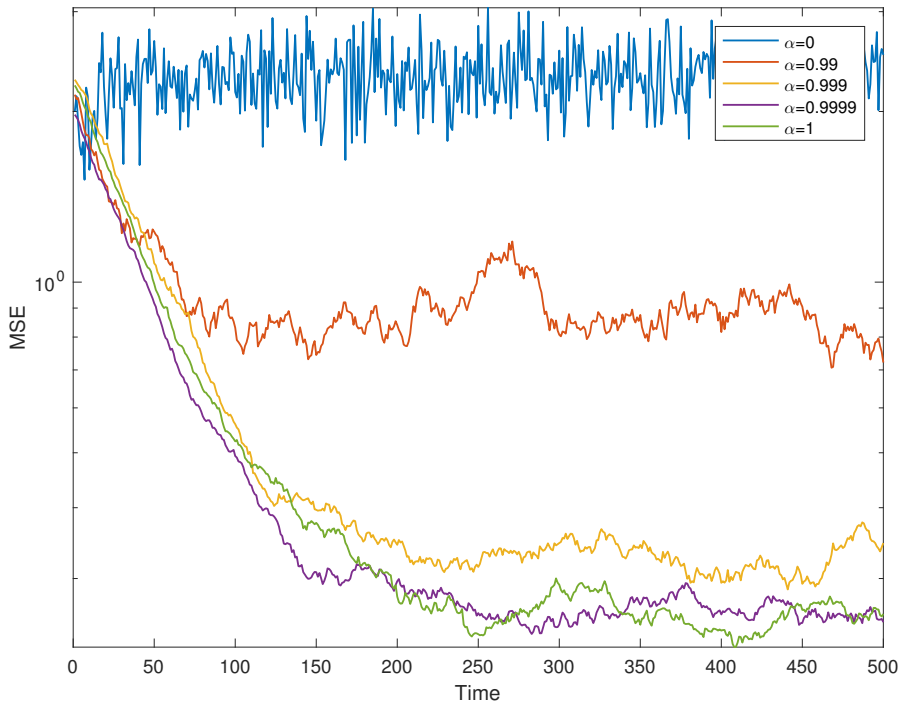

Figure 4.16: MSE for different values of fading correlation coefficient α with the LMS algorithm ($\mu = 0.01$, Dim= 8 and $\text{SNR}_{in} = 0$ dB)

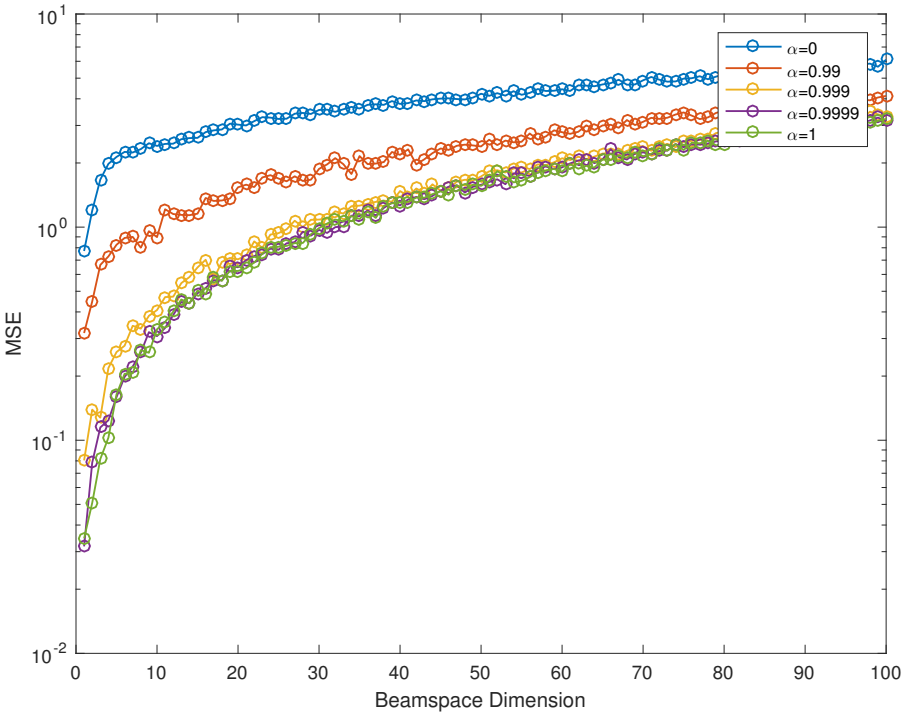

Figure 4.17: MSE for different values of fading correlation coefficient α with the LMS algorithm ($\mu = 0.1$ and $\text{SNR}_{in} = 0$ dB)

Chapter 5

RLS Algorithm for Channel Estimation

We will now consider channel estimation techniques for multiuser massive MIMO systems and employ the signal models of Chapter 2. In this section, the RLS estimator is presented. RLS algorithm recursively finds the coefficients that minimize a weighted linear least squares cost function relating to the input signals. This is in contrast to other algorithms the LMS that aim to reduce the mean square error where the mean calculation is in a statistical sense. Whereas, RLS employs time averages. The channel estimates are updated recursively upon receiving new training symbols. The channel estimation problem corresponds to solving the following least-squares (LS) optimization problem:

$$\hat{\mathbf{h}}_{n,eff} = \arg \min_{\mathbf{h}_{n,eff}} (\sum_{i=0}^{n} \lambda^{n-i} \|\mathbf{y}_i - \mathbf{B}_i^H \hat{\mathbf{h}}_{n,eff}\|^2) \tag{5.1}$$

where $\hat{\mathbf{h}}_{n,eff}$ is the $1 \times K_g L_g D$ effective channel vector coefficients at time n, or it is the reduced rank channel vector and is calculated by

$$\hat{\mathbf{h}}_{n,eff} = (\mathbf{I}_{K_g L_g} \otimes \mathbf{S}_D^H) \hat{\mathbf{h}}_n. \tag{5.2}$$

The parameter λ is a forgetting factor chosen between 0 and 1 and gives exponentially less weight to older error samples.

By using the training matrix in (2.2), $\mathbf{B}_n^H$ is defined

$$\mathbf{B}_n^H = (\mathbf{X}_n^H \otimes \mathbf{I}_D) \tag{5.3}$$

where $\mathbf{X_n}$ is the n^{th} row of training matrix $\mathbf{X}$ in (2.2) and its size is $K_g L_g \times 1$. This problem can be solved by computing the gradient terms of (5.1), equating them to a zero matrix and manipulating the terms which yields our solution.

5.1 Formulation

We rewrite (5.1) as

$$J_n = \sum_{i=0}^{n} \lambda^{n-i} \|\mathbf{e}_i\|^2$$

where

$$\mathbf{y}_i = (\mathbf{X}_i^H \otimes \mathbf{S}_D^H)\mathbf{h}_i + \boldsymbol{\eta}_i = \boldsymbol{\Psi}_i^H \mathbf{h}_i + \boldsymbol{\eta}_i$$
$$\mathbf{e}_i = \mathbf{y}_i - \mathbf{B}_i^H \hat{\mathbf{h}}_n$$

By computing the gradient we have

$$\begin{aligned}\nabla_{\hat{\mathbf{h}}_n^H} J_n &= \nabla_{\hat{\mathbf{h}}_n^H}(\sum_{i=0}^{n} \lambda^{n-i}\|\mathbf{e}_i\|^2) = \nabla_{\hat{\mathbf{h}}_n^H}(\sum_{i=0}^{n} \lambda^{n-i}\|\mathbf{y}_i - \mathbf{B}_i^H\hat{\mathbf{h}}_n\|_2) \\ &= \nabla_{\hat{\mathbf{h}}_n^H}(\sum_{i=0}^{n} \lambda^{n-i}(\mathbf{y}_i^H - \hat{\mathbf{h}}_n^H\mathbf{B}_i)(\mathbf{y}_i - \mathbf{B}_i^H\hat{\mathbf{h}}_n)) = \sum_{i=0}^{n} \lambda^{n-i}(-\mathbf{B}_i)(\mathbf{y}_i - \mathbf{B}_i^H\hat{\mathbf{h}}_n) = 0\end{aligned}$$

If we rearrange the above equation

$$\sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{B}_i^H \hat{\mathbf{h}}_n = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i \tag{5.4}$$

$$\mathbf{R}_n \hat{\mathbf{h}}_n = \mathbf{r}_n \tag{5.5}$$

where $\mathbf{R}_n = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{B}_i^H$ is the weighted covariance matrix for $\mathbf{B}_i$ and $\mathbf{r}_n = \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i$ is the equivalent estimate for the cross-covariance between $\mathbf{B}_i$ and $\mathbf{y}_i$.
Based on this expression we find the coefficients which minimize the cost function as

$$\hat{\mathbf{h}}_n = \mathbf{R}_n^{-1} \mathbf{r}_n. \tag{5.6}$$

This is a form of the well-known Wiener-Hopf equation with employing time averages.

5.2 Recursive Algorithm

The discussion resulted in a single equation to determine a coefficient vector which minimizes the cost function. In this section we want to derive a recursive solution of the form

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \Delta \hat{\mathbf{h}}_{n-1} \tag{5.7}$$

where $\Delta \hat{\mathbf{h}}_{n-1}$ is a correction factor at time $n-1$. We start the derivation of the recursive algorithm by expressing the cross covariance $\mathbf{r}_n$ in terms of $\mathbf{r}_{n-1}$

$$\begin{aligned} \mathbf{r}_n &= \sum_{i=0}^{n} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i = \sum_{i=0}^{n-1} \lambda^{n-i} \mathbf{B}_i \mathbf{y}_i + \lambda^0 \mathbf{B}_n \mathbf{y}_n \\ &= \lambda \mathbf{r}_{n-1} + \mathbf{B}_n \mathbf{y}_n \end{aligned} \tag{5.8}$$

Similarly we express $\mathbf{R}_n$ in terms of $\mathbf{R}_{n-1}$ by

$$\begin{aligned}\mathbf{R}_n &= \sum_{i=0}^{n} \lambda^{n-i}\mathbf{B}_i\mathbf{B}_i^H = \sum_{i=0}^{n-1} \lambda^{n-i}\mathbf{B}_i\mathbf{B}_i^H + \mathbf{B}_i\mathbf{B}_i^H \\ &= \lambda\mathbf{R}_{n-1} + \mathbf{B}_n\mathbf{B}_n^H \end{aligned} \tag{5.9}$$

In order to calculate the channel coefficients vector we are interested in the inverse of auto-covariance matrix. The matrix inversion lemma comes in handy for this task [26]

$$\mathbf{R}_n^{-1} = \lambda^{-1}\mathbf{R}_{n-1}^{-1} - \lambda^{-1}\mathbf{R}_{n-1}^{-1}\mathbf{B}_n(\mathbf{I}_D + \mathbf{B}_n^H\lambda^{-1}\mathbf{R}_{n-1}^{-1}\mathbf{B}_n)^{-1}\mathbf{B}_n^H\lambda^{-1}\mathbf{R}_{n-1}^{-1}.$$

To come in line with the standard literature, we define

$$\begin{aligned}\mathbf{P}_n &\triangleq \mathbf{R}_n^{-1} \\ &= \lambda^{-1}\mathbf{P}_{n-1} - \mathbf{K}_n\mathbf{B}_n^H\lambda^{-1}\mathbf{P}_{n-1} \end{aligned} \tag{5.10}$$

where the gain matrix $\mathbf{K}_n$ is

$$\begin{aligned}\mathbf{K}_n &= \lambda^{-1}\mathbf{P}_{n-1}\mathbf{B}_n(\mathbf{I}_D + \mathbf{B}_n^H\lambda^{-1}\mathbf{P}_{n-1}\mathbf{B}_n)^{-1} \\ &= \mathbf{P}_{n-1}\mathbf{B}_n(\lambda\mathbf{I}_D + \mathbf{B}_n^H\mathbf{P}_{n-1}\mathbf{B}_n)^{-1}. \end{aligned} \tag{5.11}$$

Before we move on, it is necessary to bring $\mathbf{K}_n$ into another form

$$\begin{aligned}\mathbf{K}_n(\mathbf{I}_D + \lambda^{-1}\mathbf{B}_n^H\mathbf{P}_{n-1}\mathbf{B}_n) &= \lambda^{-1}\mathbf{P}_{n-1}\mathbf{B}_n \\ \mathbf{K}_n + \lambda^{-1}\mathbf{K}_n\mathbf{B}_n^H\mathbf{P}_{n-1}\mathbf{B}_n &= \lambda^{-1}\mathbf{P}_{n-1}\mathbf{B}_n\end{aligned}$$

Subtracting the second term on the left hand side yields

$$\mathbf{K}_n = \lambda^{-1}(\mathbf{P}_{n-1} - \mathbf{K}_n\mathbf{B}_n^H\mathbf{P}_{n-1})\mathbf{B}_n. \tag{5.12}$$

With the recursive definition of $\mathbf{P}_n$ the desired form follows

$$\mathbf{K}_n = \mathbf{P}_n \mathbf{B}_n \tag{5.13}$$

Now we are ready to complete the recursion. As discussed

$$\hat{\mathbf{h}}_n = \mathbf{P}_n \mathbf{r}_n = \mathbf{P}_n(\lambda \mathbf{r}_{n-1} + \mathbf{B}_n \mathbf{y}_n) = \lambda \mathbf{P}_n \mathbf{r}_{n-1} + \mathbf{P}_n \mathbf{B}_n \mathbf{y}_n \tag{5.14}$$

The second step follows from the recursive definition of $\mathbf{r}_n$ in (5.8). Next we incorporate the recursive definition of $\mathbf{P}_n$ in (5.10) together with the alternate form of $\mathbf{K}_n$ and get

$$\begin{aligned}\hat{\mathbf{h}}_n &= \lambda(\lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{K}_n \mathbf{B}_n^H \mathbf{P}_{n-1})\mathbf{r}_{n-1} + \mathbf{K}_n \mathbf{y}_n = \mathbf{P}_{n-1}\mathbf{r}_{n-1} - \mathbf{K}_n \mathbf{B}_n^H \mathbf{P}_{n-1}\mathbf{r}_{n-1} + \mathbf{K}_n \mathbf{y}_n \\ &= \mathbf{P}_{n-1}\mathbf{r}_{n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{B}_n^H \mathbf{P}_{n-1}\mathbf{r}_{n-1}).\end{aligned}$$

With (5.12), we arrive at the update equation

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \mathbf{K}_n(\mathbf{y}_n - \mathbf{B}_n^H \hat{\mathbf{h}}_{n-1}) = \hat{\mathbf{h}}_{n-1} + \mathbf{K}_n \boldsymbol{\alpha}_n \tag{5.15}$$

where $\boldsymbol{\alpha}_n = \mathbf{y}_n - \mathbf{B}_n^H \hat{\mathbf{h}}_{n-1}$ is the a priori error. That means we found the correction factor

$$\Delta\hat{\mathbf{h}}_{n-1} = \mathbf{K}_n \boldsymbol{\alpha}_n \tag{5.16}$$

This intuitively satisfying result indicates that the correction factor is directly proportional to both the error and the gain vector, which controls how much sensitivity is desired, through the forgetting factor λ. When the channel is static over the transmission duration, it is logical to set the forgetting factor λ to one. On the other hand, when the channel is time-varying, in order to track the channel variations one needs to set λ to a value that corresponds to the coherence time of the channel.

5.3 RLS Algorithm

By using the above equations the recursive algorithm for this application can be summarized as

Initialization

$$\hat{\mathbf{h}}_{\mathbf{0}} = \mathbf{0}$$

$$\mathbf{P}_0 = \delta^{-1}\mathbf{I}_D \quad \text{where } \mathbf{I} \text{ is the identity matrix of rank } D$$

where δ is a small positive number.

Computation

For $n = 0, 1, 2, ...$

$$\mathbf{y}_n = \mathbf{\Psi}_n^H \mathbf{h}_n + \boldsymbol{\eta}_n$$

$$\mathbf{B}_n = (\mathbf{x}_n^H \otimes \mathbf{I}_D)$$

$$\boldsymbol{\alpha}_n = \mathbf{y}_n - \mathbf{B}_n^H \hat{\mathbf{h}}_{n-1}$$

$$\mathbf{S}_n = \lambda \mathbf{I}_D + \mathbf{B}_n^H \mathbf{P}_{n-1} \mathbf{B}_n$$

$$\mathbf{Q}_n = \mathbf{P}_{n-1}\mathbf{B}_n$$

$$\mathbf{K}_n = \mathbf{Q}_n \mathbf{S}_n^{-1}$$

$$\mathbf{P}_n = \lambda^{-1}\mathbf{P}_{n-1} - \lambda^{-1}\mathbf{K}_n \mathbf{Q}_n^H$$

$$\hat{\mathbf{h}}_n = \hat{\mathbf{h}}_{n-1} + \mathbf{K}_n \boldsymbol{\alpha}_n$$

By comparing the RLS algorithm and the Kalman filter, there are some similarities. Both have the same procedure and matrix inversion. But the important part is in the RLS algorithm the convergence is much faster than the Kalman filter. The reason is in the Kalman filter the size of the matrix which needs to be inverted is $K_g L_g D \times K_g L_g D$. But the dimension for the matrix inversion in RLS algorithm is $D \times D$ where D stands for the

beamspace dimension.

5.4 Simulation Results

By using the same setting as previous sections, the channel is simulated. The RLS algorithm is used to estimate the channel and plot the MSE for different values of channel fading coefficient α.

As can be seen in Figure 5.1, the RLS algorithm performance with different αs depends on the forgetting factor λ. There is a different optimum value of forgetting factor for each α. When $\alpha < 1$, the optimum forgetting factor λ_{opt} is decreasing by decreasing the α. This means, to get the best performance of RLS algorithm, changing the forgetting factor to λ_{opt} is necessary and as we will see in the next chapter, when the λ is not the optimum value, the performance of the RLS algorithm is worse than the the LMS algorithm.

From Figure 5.2-5.5, 5.7-5.10 and 5.12-5.15, the results are along the same line as the previous chapters. By increasing the beamspace dimension, the performance gets worse. Again, after some point (usually $n > 300$ for $\text{SNR}_{in} = 0$ dB, $n > 100$ for $\text{SNR}_{in} = 3$ dB and $n > 50$ for $\text{SNR}_{in} = 30$ dB) the MSE for different setups saturates. In Figure 5.6, 5.11 and 5.16, the results for different values of the channel fading coefficient α and the forgetting factor λ are plotted. The performance gets better by increasing α which is expected. Also, for getting the best result, for different values of the channel fading coefficient α, the optimum forgetting factor λ_{opt} is used. By comparing Figure these three figures, It can be seen that in case of $\text{SNR}_{in} = 30$dB, the performance is about $20 - 30$ dB better than the $\text{SNR}_{in} = 3$ dB and 0 dB.

In practice, the convergence of the RLS algorithm is much faster than the Kalman filter which was expected based on the size of the matrix inversions.

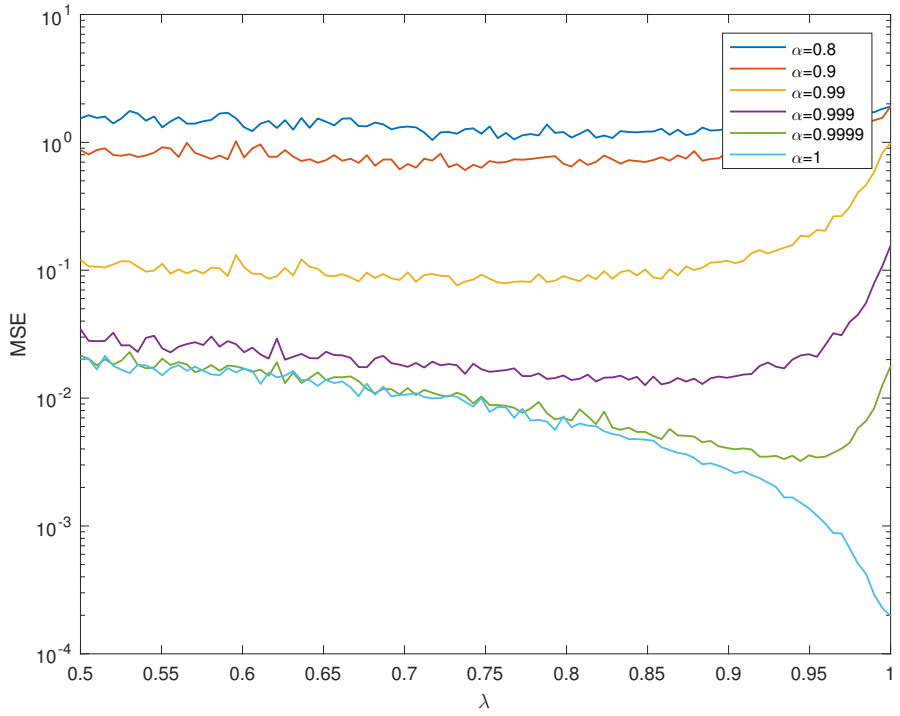

Figure 5.1: MSE for different values of λ in the RLS algorithm (Dim$=8, \delta = 0.01$ and $\text{SNR}_{in} = 30$ dB)

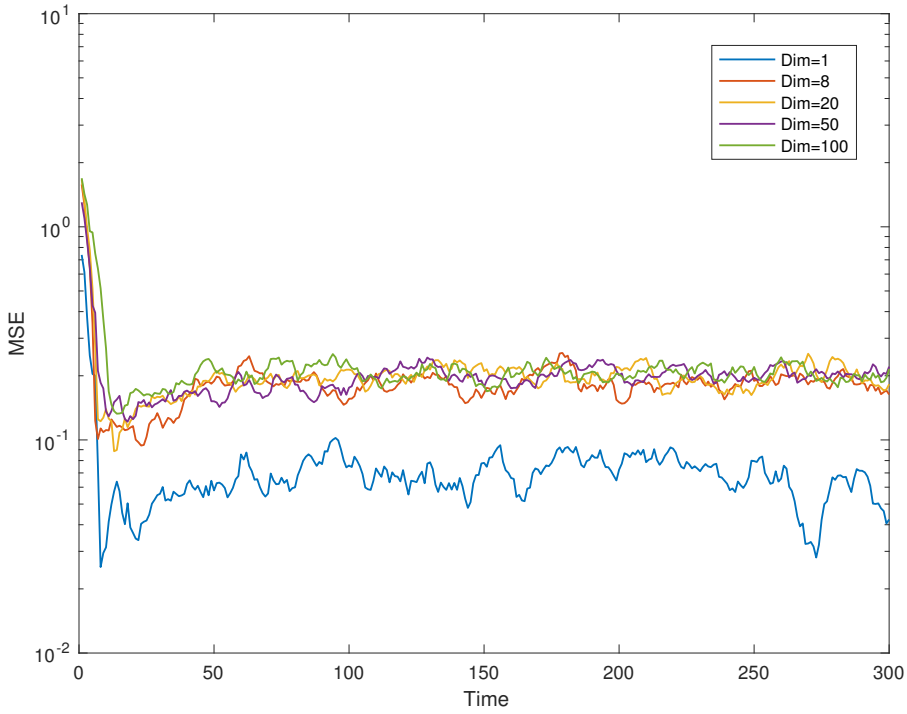

Figure 5.2: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.99, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 30$ dB)

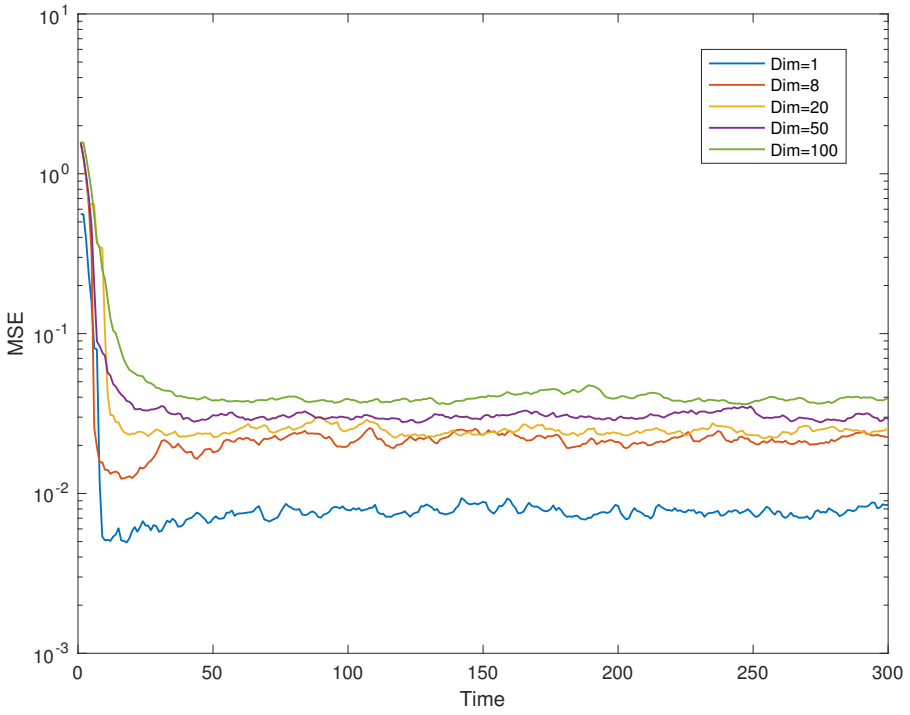

Figure 5.3: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.999, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 30$ dB)

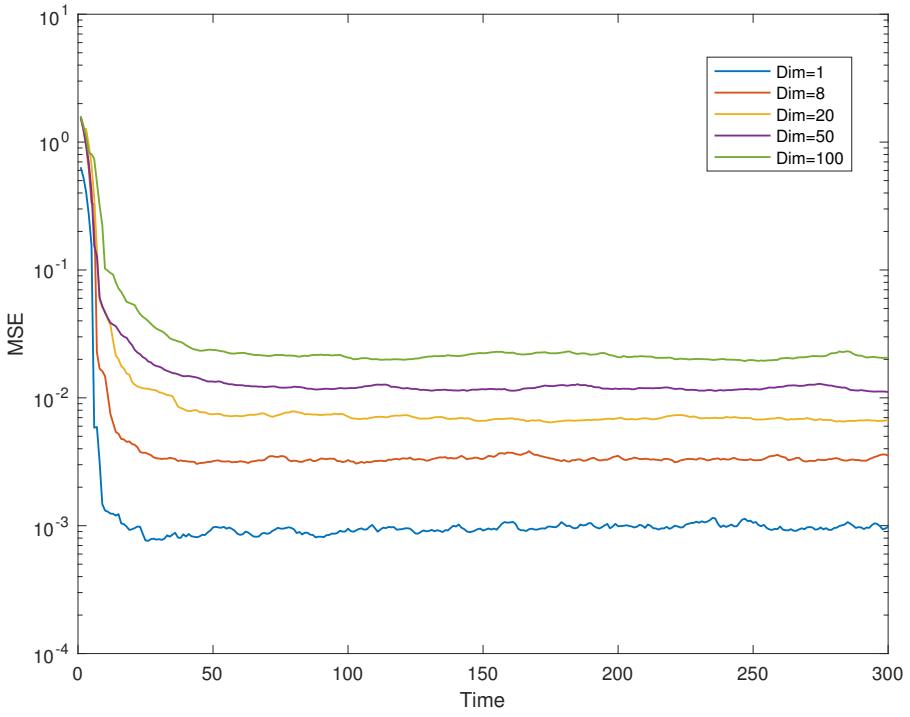

Figure 5.4: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 30$ dB)

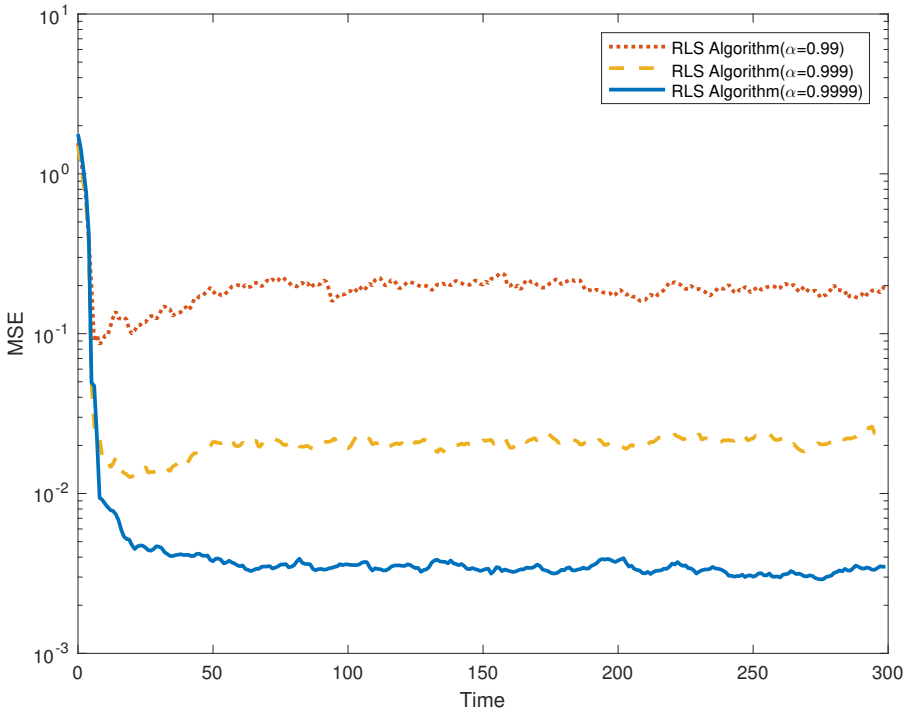

Figure 5.5: MSE for different values of fading correlation coefficient α with the RLS algorithm ($\lambda = 0.95, \delta = 0.01$, Dim= 8 and $\text{SNR}_{in} = 30$ dB)

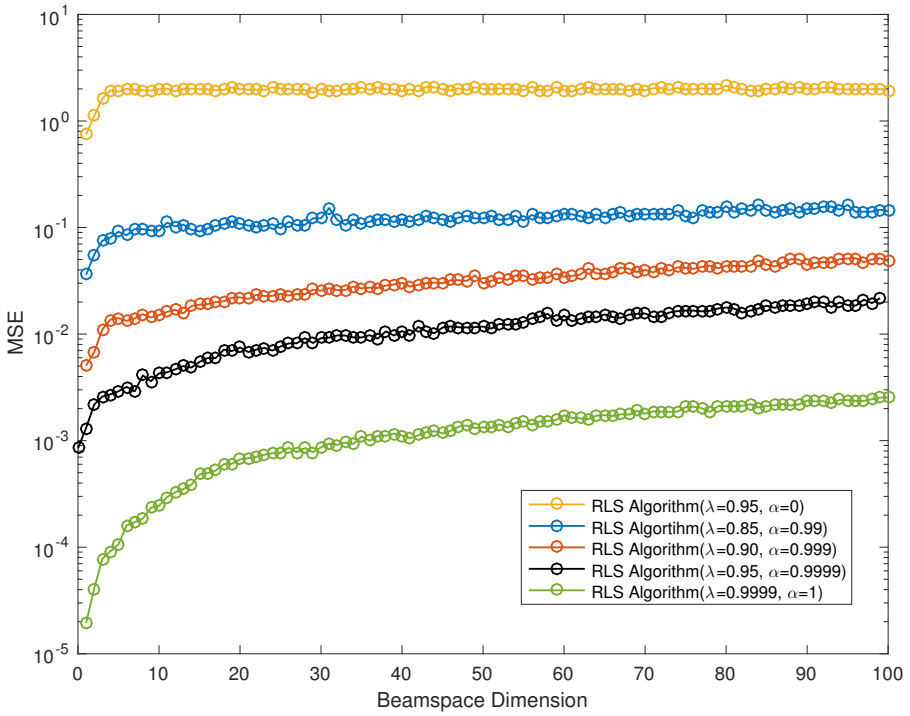

Figure 5.6: MSE for different values of fading correlation coefficient α with the RLS algorithm ($\text{SNR}_{in} = 30$ dB and $\delta = 0.01$)

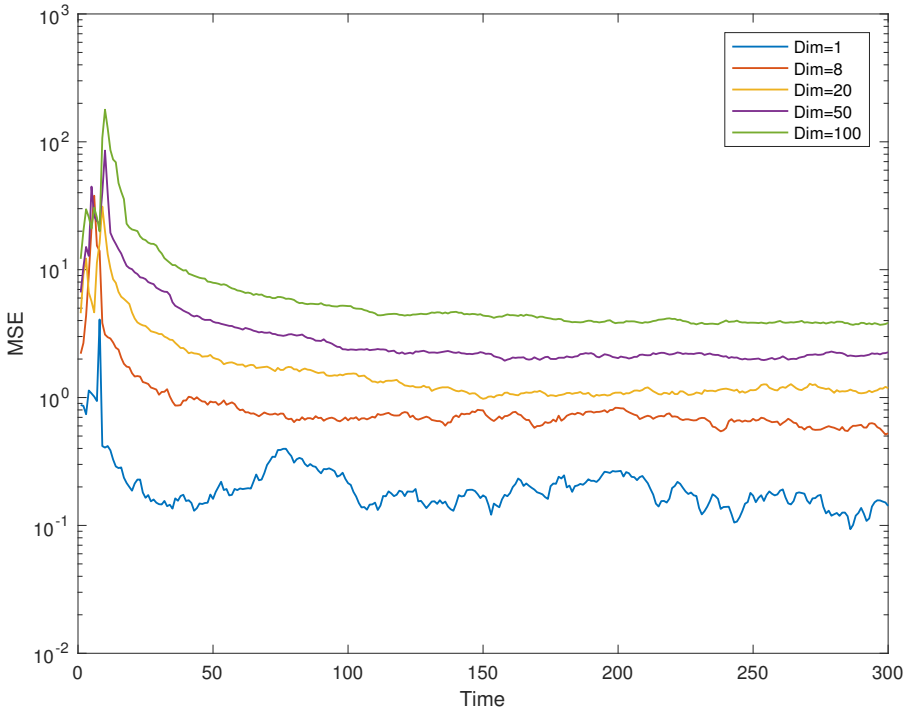

Figure 5.7: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.99, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 3$ dB)

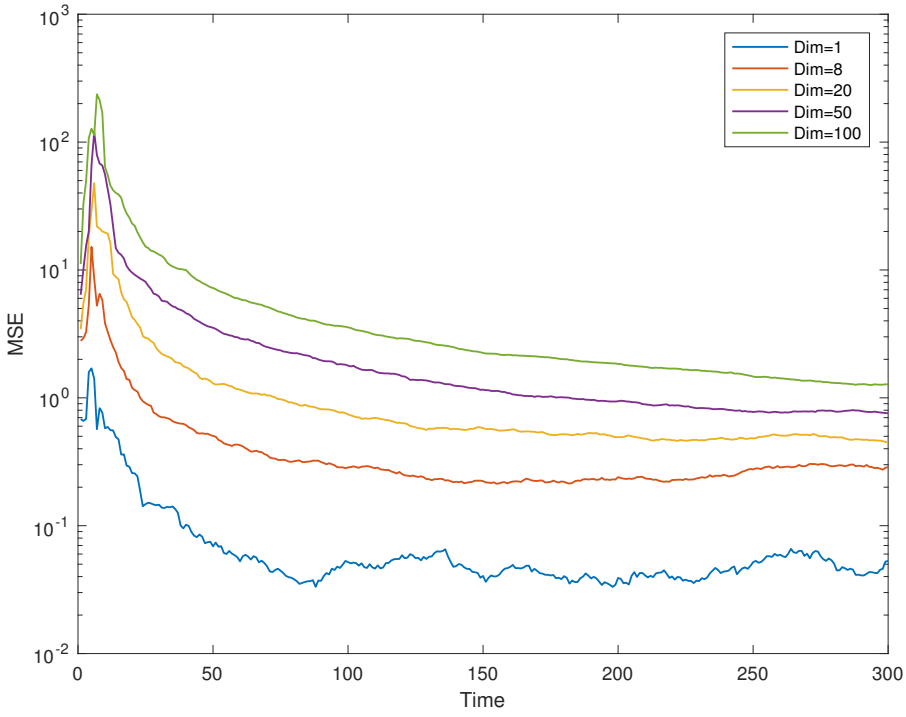

Figure 5.8: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.999, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 3$ dB)

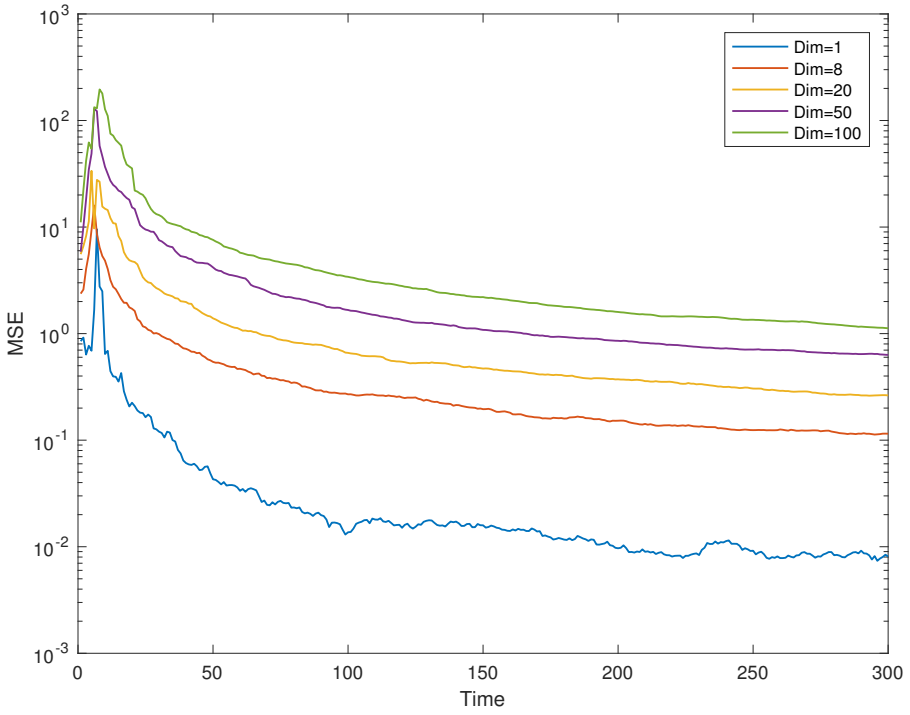

Figure 5.9: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 3$ dB)

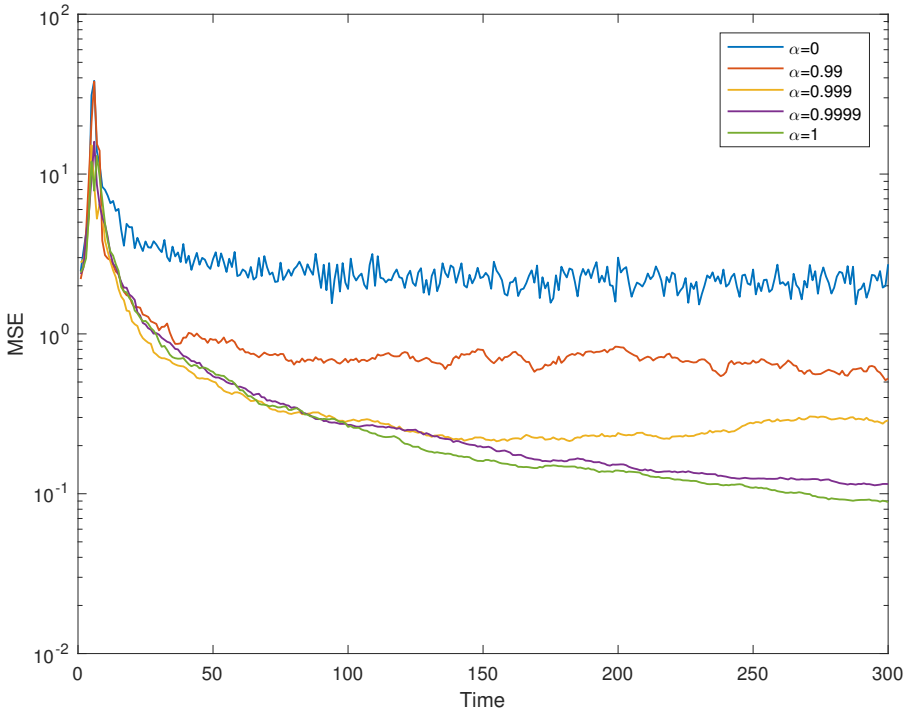

Figure 5.10: MSE for different values of fading correlation coefficient α with the RLS algorithm ($\lambda = 0.95, \delta = 0.01$, Dim= 8 and $\text{SNR}_{in} = 3$ dB)

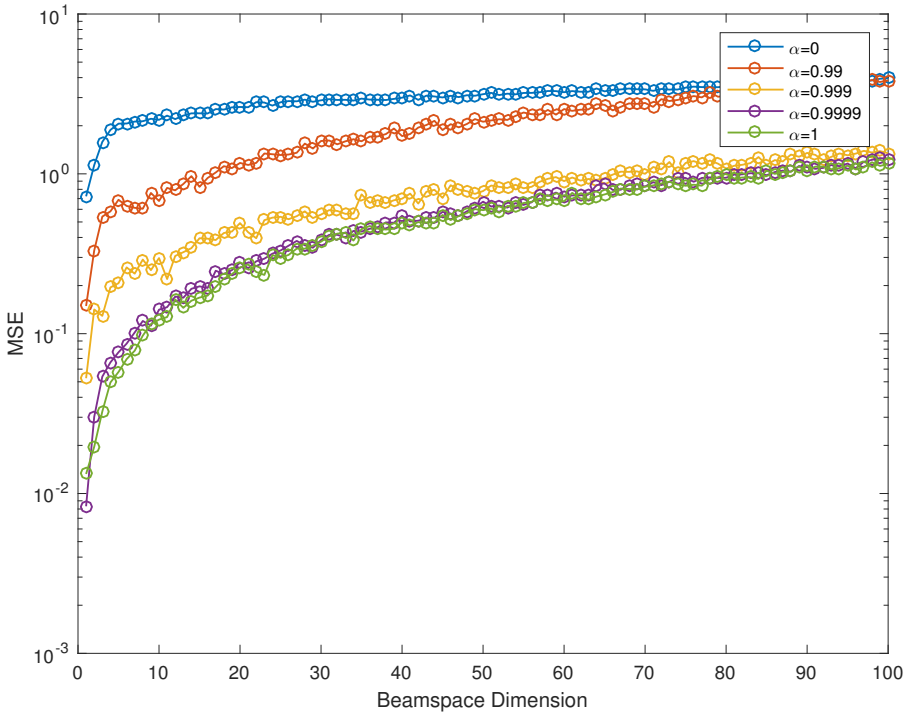

Figure 5.11: MSE for different values of fading correlation coefficient α with the RLS algorithm ($\text{SNR}_{in} = 3$ dB and $\delta = 0.01$)

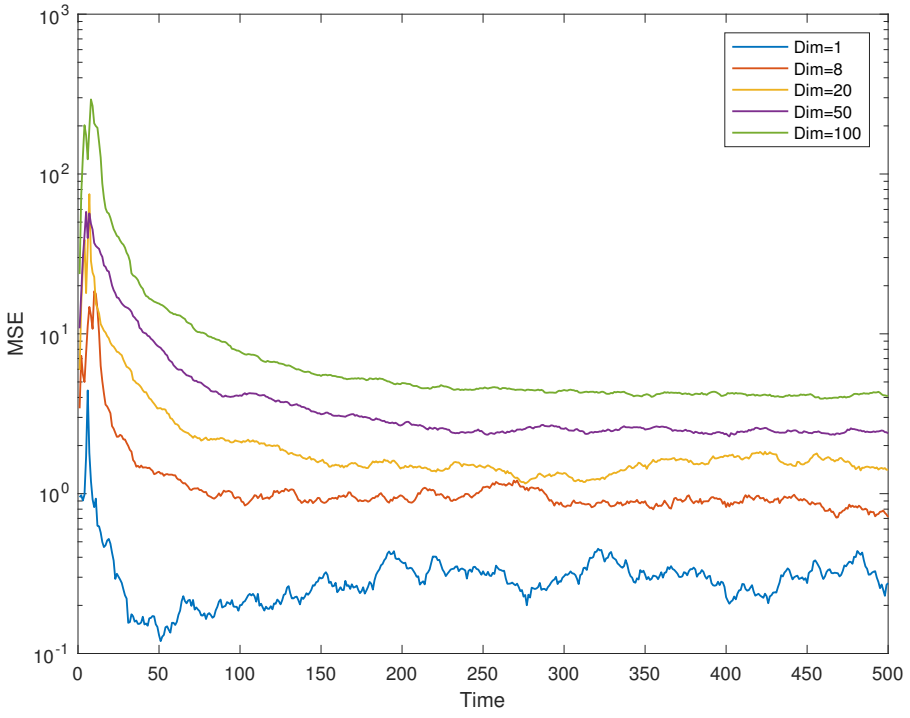

Figure 5.12: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.99, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 0$ dB)

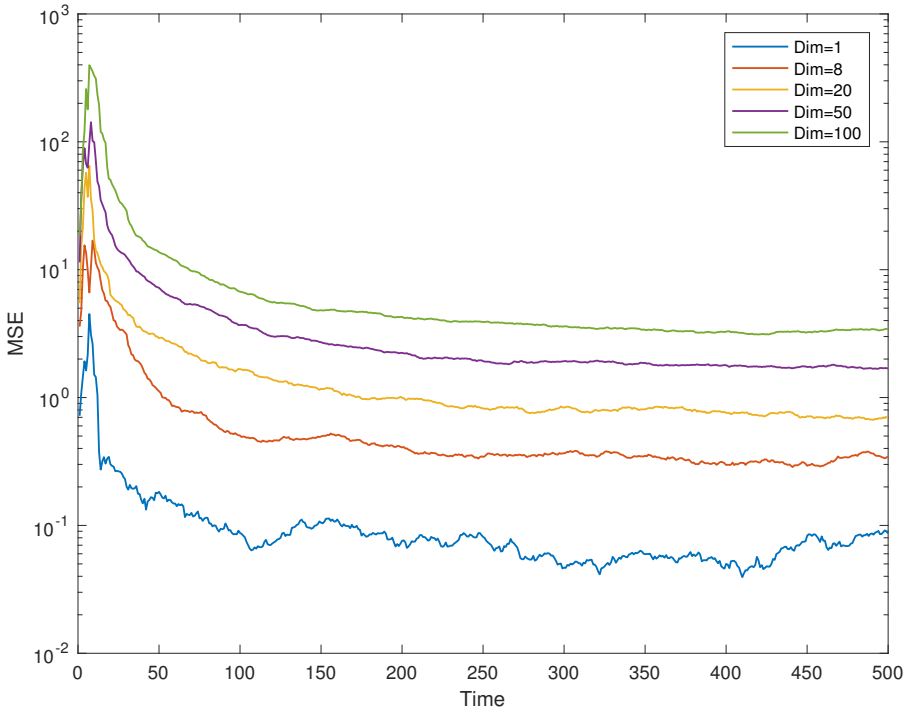

Figure 5.13: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.999, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 0$ dB)

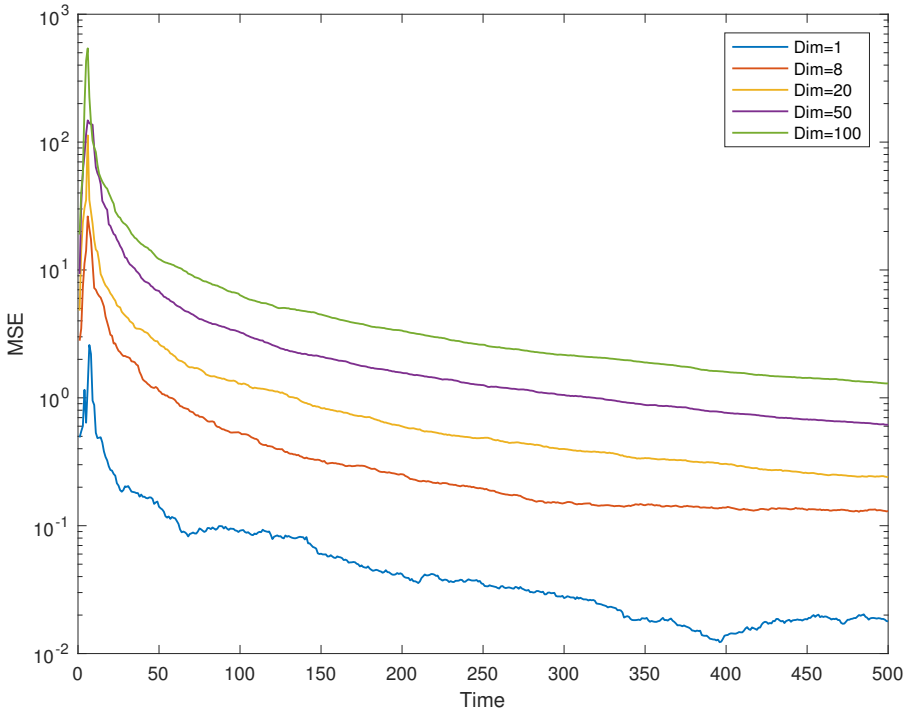

Figure 5.14: MSE for different values of beamspace dimension (Dim) with the RLS algorithm ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01$ and $\text{SNR}_{in} = 3$ dB)

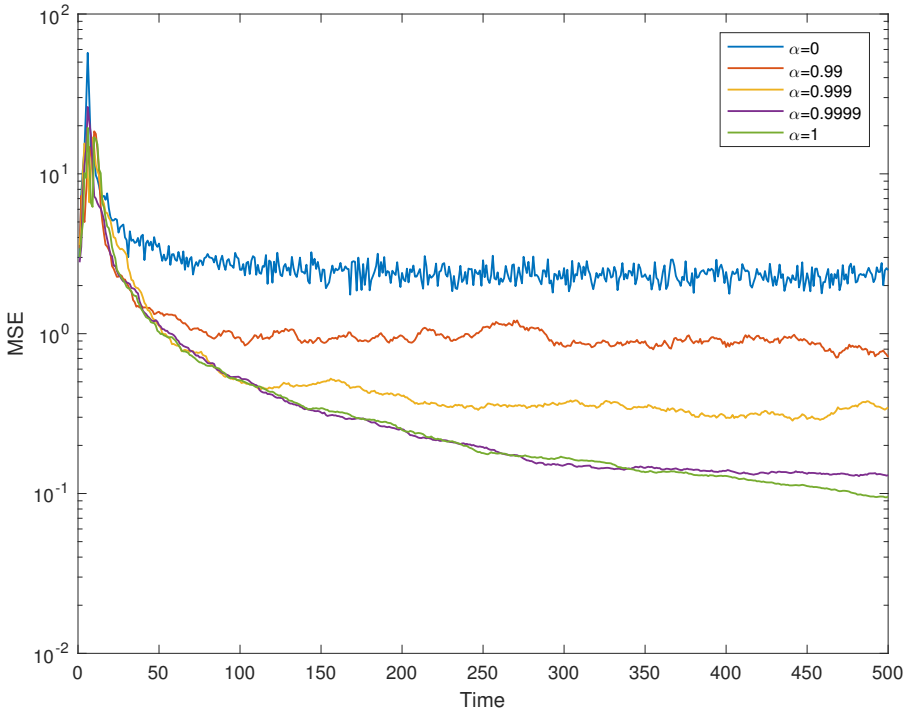

Figure 5.15: MSE for different values of fading correlation coefficient α with the RLS algorithm ($\lambda = 0.95, \delta = 0.01$, Dim= 8 and $\text{SNR}_{in} = 0$ dB)

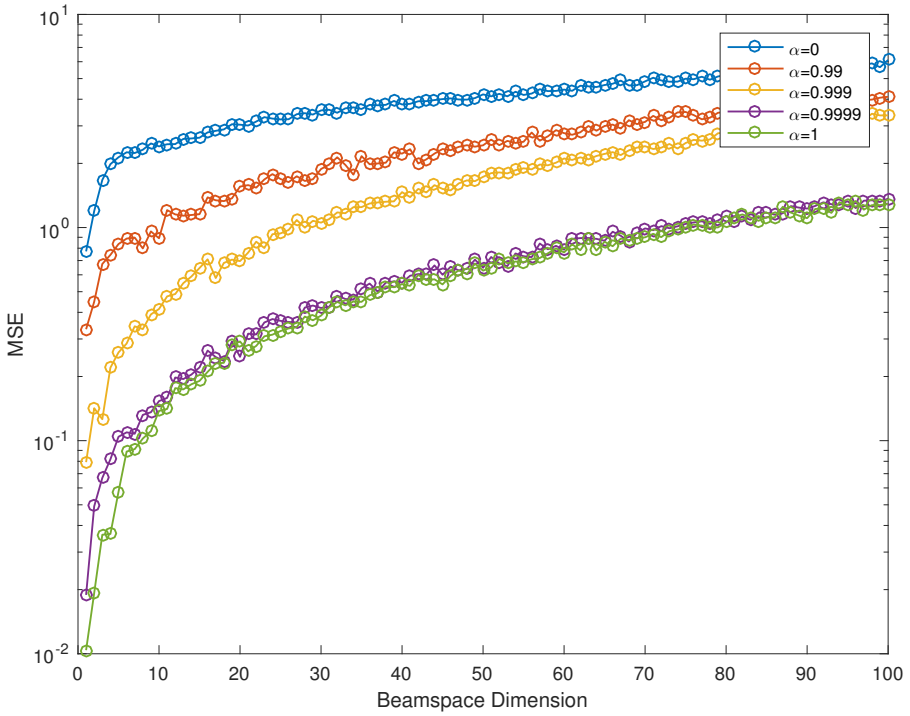

Figure 5.16: MSE for different values of fading correlation coefficient α with the RLS algorithm ($\text{SNR}_{in} = 0$ dB and $\delta = 0.01$)

Chapter 6

Comparison and Optimum Answer

6.1 Capacity as a Measure for the Optimum Dimension

So far we have used tools to estimate the channel coefficient vector $\hat{\mathbf{h}}_n$. We want to reduce the full rank channel into the efficient channel for simple calculation. For the optimum dimension, we need to define a measure. Here we are using a capacity measure which uses only the first D elements of the reduced dimension estimated channel vector to estimate the channel capacity. In other words, instead of using the whole vector $\mathbf{h}_n$ which has a large size of $K_g L_g N \times 1$, we are using a reduced dimension version of it which is the vector

$$\mathbf{H}_n = \big(\mathbf{I}_{K_g L_g} \otimes \mathbf{S}_D^H\big)\mathbf{h}_n = \mathbf{Q}\mathbf{h}_n. \tag{6.1}$$

Now, for simplicity in the definition of the capacity, instead of using the whole vector $\mathbf{H}_n$, just the first D elements of the vector is used where D stands for the beamspace dimension. We call this new vector $[\mathbf{h}'_n]_{D\times 1}$

$$\begin{aligned}\mathbf{y}_n &= \mathbf{h}'_n x_n + \boldsymbol{\eta}_n \\ &= (\mathbf{h}'_n + \hat{\mathbf{h}}'_n - \hat{\mathbf{h}}'_n)x_n + \boldsymbol{\eta}_n \\ &= \hat{\mathbf{h}}'_n x_n + \mathbf{e}_n x_n + \boldsymbol{\eta}_n \end{aligned} \tag{6.2}$$

where $\hat{\mathbf{h}}'_n$ is the first D elements of the channel estimation vector which is computed via the RLS or LMS algorithm. In the case of the Kalman filter, first we need to multiply the channel estimation vector via the dimension reducing matrix $\mathbf{Q}$ then use the first D elements of the computed vector. Also x_n is our training data.

One can write:

$$z_n = \hat{\mathbf{h}}'^H_n \mathbf{y}_n = |\hat{\mathbf{h}}'_n|^2 x_n + \underbrace{\hat{\mathbf{h}}'^H_n(\mathbf{e}_n x_n + \boldsymbol{\eta}_n)}_{\sigma^2_\zeta} \tag{6.3}$$

By calculating the expected value of σ^2_ζ we will have:

$$\begin{aligned}\sigma^2_\zeta &= \mathbb{E}\{\hat{\mathbf{h}}'^H_n(\mathbf{e}_n x_n + \boldsymbol{\eta}_n)(\mathbf{e}^H_n x^*_n + \boldsymbol{\eta}^H_n)\hat{\mathbf{h}}'_n\} \\ &= \hat{\mathbf{h}}'^H_n \mathbb{E}\{(\mathbf{e}_n x_n + \boldsymbol{\eta}_n)(\mathbf{e}^H_n x^*_n + \boldsymbol{\eta}^H_n)\}\hat{\mathbf{h}}'_n \\ &= \hat{\mathbf{h}}'^H_n(\mathbf{R}_\mathbf{e} + \mathbf{R}_{\boldsymbol{\eta}})\hat{\mathbf{h}}'_n \end{aligned} \tag{6.4}$$

where $\mathbf{R}_\mathbf{e} = \mathbb{E}\{\mathbf{e}_n \mathbf{e}^H_n\}$ and $\mathbf{R}_{\boldsymbol{\eta}} = \mathbb{E}\{\boldsymbol{\eta}_n \boldsymbol{\eta}^H_n\}$.

One can use the above equations to write the capacity formula

$$C(D) = E_{\hat{\mathbf{h}}'_n}\{\log_2(1 + \frac{||\hat{\mathbf{h}}'_n||^2}{\sigma^2_\zeta})\} = E_{\hat{\mathbf{h}}}\{\log_2(1 + \frac{||\hat{\mathbf{h}}'_n||^2}{\hat{\mathbf{h}}'^H_n(\mathbf{R}_e + \mathbf{R}_\eta)\hat{\mathbf{h}}'_n})\} \tag{6.5}$$

In the sequel, we will use this formulation in our simulations. However, before them, we wish to introduce two other channel models.

6.2 Two-state Markov Chain Temporal Fading Coefficient

In the previous sections, we assumed that the temporal fading coefficient (α) in (2.4) is constant with respect to time. In this section, we assume that α varies with time and we try to figure out if the methods still track the channel coefficient vector. The new model for α is a two state Markov chain model where the initial state is α_0 with the probability matrix transition $\mathbf{P}_\alpha$ where

$$\mathbf{P}_\alpha = \begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} \tag{6.6}$$

We assume that our initial state is α_0 and with probability p_{00}, the current state remains the same and with probability p_{01} the state varies. The state transition diagram is in Figure 6.1.

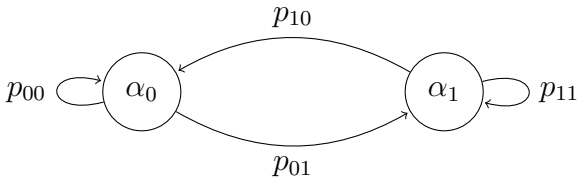

Figure 6.1: State transition diagram of the temporal fading channel which varies with $\mathbf{P}_\alpha$ transition probability matrix.

6.3 New Gauss-Markov Channel State

The channel temporal variation was defined as (2.4) in chapter 2. In this section we are going to use another definition for the channel state update. This model takes into account

$$\mathbf{h}_0 = \mathbf{R}^{\frac{1}{2}}\mathbf{b}_0 \tag{6.7}$$

$$\mathbf{h}_i = \sqrt{\alpha}\mathbf{h}_{i-1} + \sqrt{1-\alpha}\mathbf{R}^{\frac{1}{2}}\mathbf{b}_i, \qquad i \geq 1 \tag{6.8}$$

$\mathbf{R}$ is the spatial correlation matrix defined by

$$\mathbf{R} = \begin{bmatrix} 1 & q & \dots & q^{K_g \times L_g \times N-1} \\ q & 1 & & \\ \vdots & & \ddots & \\ q^{K_g \times L_g \times N-1} & & & 1 \end{bmatrix} \tag{6.9}$$

where $0 < q < 1$.

The purpose of using this model is to consider the effect of spatial correlation in the channel. This will be in addition to the temporal correlation modeled in (2.4)

6.4 Simulation Results

Different channel states in the previous sections are simulated by using the same settings in the simulations in the previous chapters. Also, by evaluating (6.5), and we also compared the results of previous chapters to each other to see which method is the better one. In this section these results are discussed.

6.4.1 Simulation Results for Capacity

For a better comparison, one can define a capacity for the actual channel

$$C_{actual}(D) = E_{\mathbf{h}'_n}\{\log_2(1 + \frac{||\mathbf{h}'_n||^2}{\mathbf{h}'^H_n(\mathbf{R}_{ns})\mathbf{h}'_n})\} \tag{6.10}$$

where $\mathbf{R}_{ns} = \mathbf{S}_D^H \mathbf{R}_n \mathbf{S}_D$ and $\mathbf{R}_n$ is the interference correlation matrix. Also the $\mathbf{h}'_n$ is the first D elements of the channel vector coefficient initialized randomly in the beginning of simulation. The channel vector coefficient is updated in each step with (2.4).

In Figure 6.2 - 6.4, the results of simulation based on the previous assumptions and (6.5) is plotted. As we can see from the figures, the Kalman filter has the maximum capacity among the algorithms which was expected. The Kalman filter has also the smallest MSE compared with the other methods. Also, by increasing the channel fading coefficient α, the capacity for the different algorithms approaches the actual channel capacity.

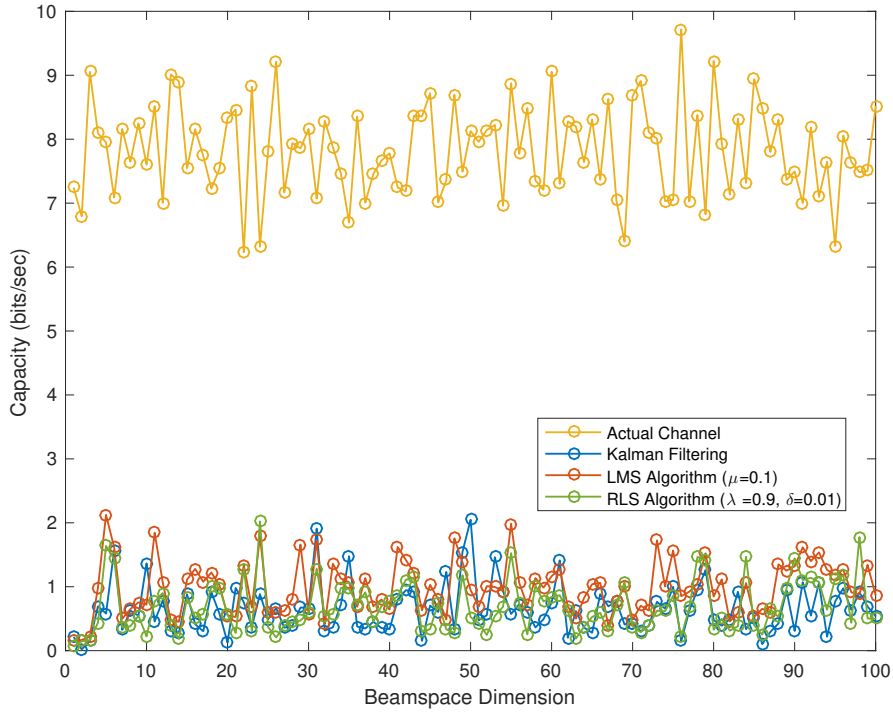

Figure 6.2: Capacity of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension and for the actual channel ($\alpha = 0.001, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

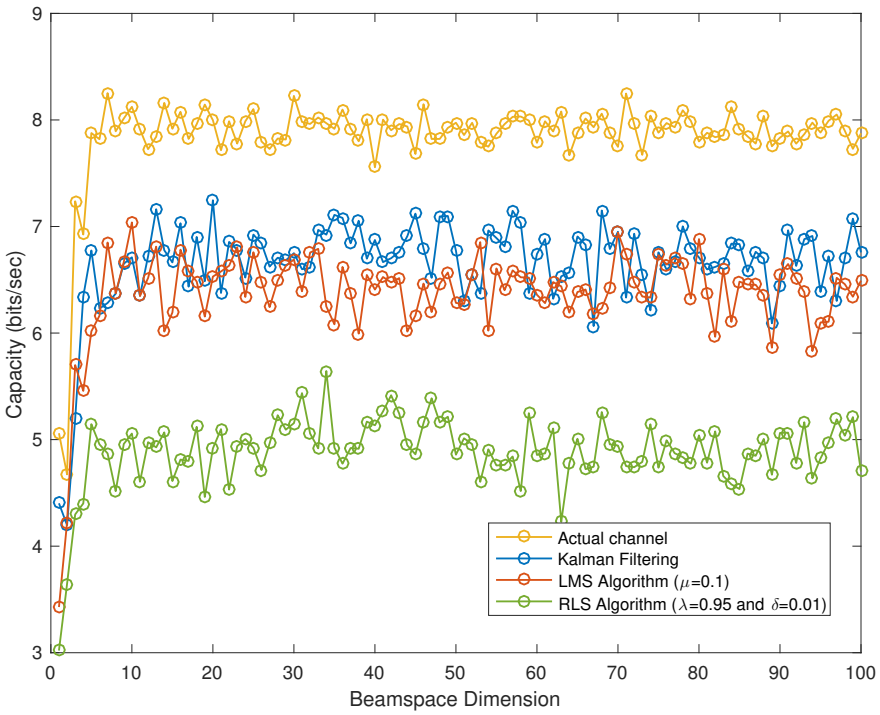

Figure 6.3: Capacity of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension and for the actual channel ($\alpha = 0.995, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

6.4.2 Simulation Results for Two-State Markov Chain Temporal Fading Coefficient

In this section, by using the model in the Section 6.2, different algorithms are used to estimate the channel under a new channel model. In Figure 6.5-6.9 the results of the simulation can be seen. The interesting thing here is that for different setups for α_0, α_1, and the transition matrix, the algorithms still can estimate the channel after some iterations. The final value of MSE in these cases is not as good as the simple channel with only one α. For example in Figure 6.7, the value for the channel fading coefficient is $\boldsymbol{\alpha} = [\alpha_0,\ \alpha_1] = [0.9999,\ 0.1]$ and the transition matrix is $\mathbf{P}_{\boldsymbol{\alpha}} = \begin{bmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{bmatrix}$. As can be seen through this figure, the three different methods are still able to estimate the channel, even when the channel fading coefficient α varies when time. The performance is worse than the constant value of α which is expected.

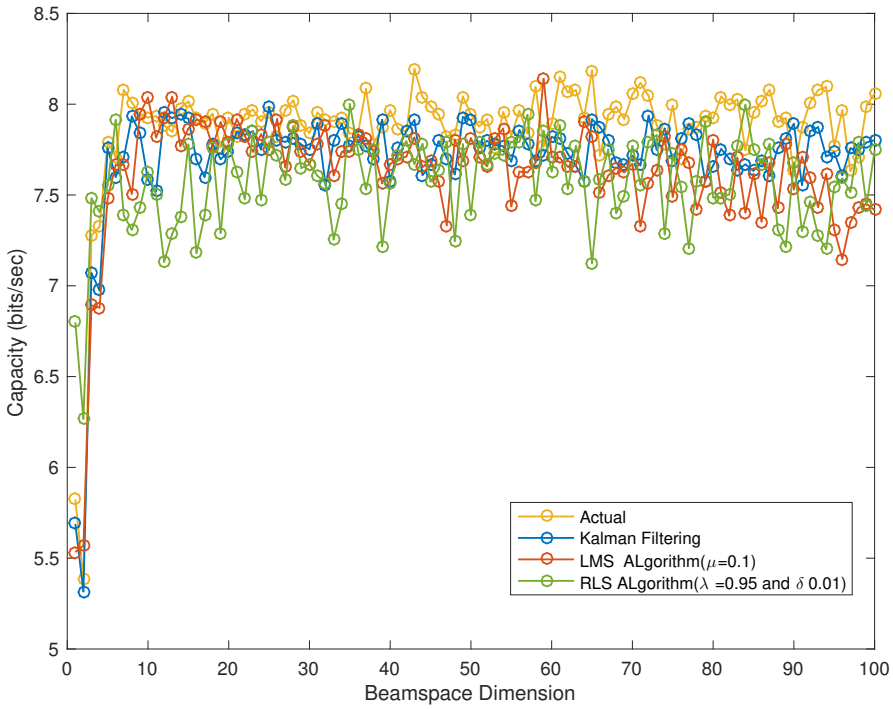

Figure 6.4: Capacity of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension and for the actual channel ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

6.4.3 Simulation Results for New Gauss-Markov Channel State

In Figure 6.10-6.18, we can see the results of the simulation based on Section 6.3 for different values of spatial correlation matrices $\mathbf{R}$. In the cases where $q = 0$, the results are the same as previous sections (in this case $\mathbf{R}$ is the identity matrix). And the results in the case ($q = 0$) are smoother. Also, by increasing the value of q, the performance of the Kalman filter gets worse. For example in the case of $q = 0.9999$, the Kalman filter has the worst performance among the algorithms.

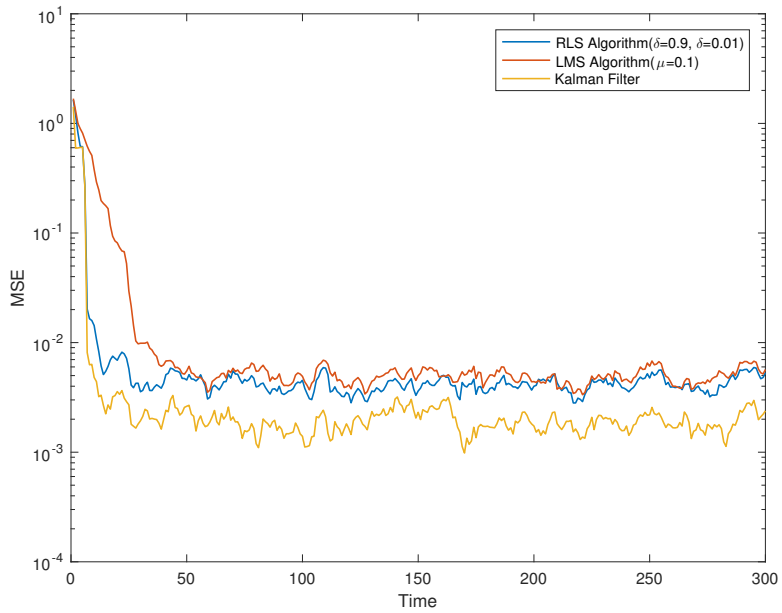

Figure 6.5: MSE for different Algorithms with Dim=8, correlation factors $[\alpha_0 = 0.9999, \alpha_1 = 0.1]$ and $p_{00} = 0.99, p_{11} = 0.1$

6.4.4 Comparing Simulation of Different Methods

In Figure 6.19, MSE vs λ is plotted for the RLS algorithm and as we can see, for different values of the channel coefficient factor α, the optimum value of λ for minimizing MSE is different. So based on this results, for different channel coefficient factor α we used the optimum λ, λ_{opt}. In Figures 6.20-6.35 the results of different methods are compared, and it is noticeable that in all simulations for different parameters, the Kalman filter performs better in estimating the channel. After the Kalman filter, RLS algorithm with λ_{opt} estimates the channel better than the LMS algorithm. As can be seen in Figure 6.23 and 6.25, the RLS algorithm performance gets better by changing the λ to the λ_{opt}, which was expected based on Figure 6.19. In figure 6.27, the performance of RLS algorithm is worse than the LMS algorithm, because of the value of λ is not the λ_{opt}. The interesting point in the Figure 6.20-6.35 is that for each algorithm, the value of MSE for the beamspace dimension one $(D = 1)$ has the minimum value among all the beamspace dimension.

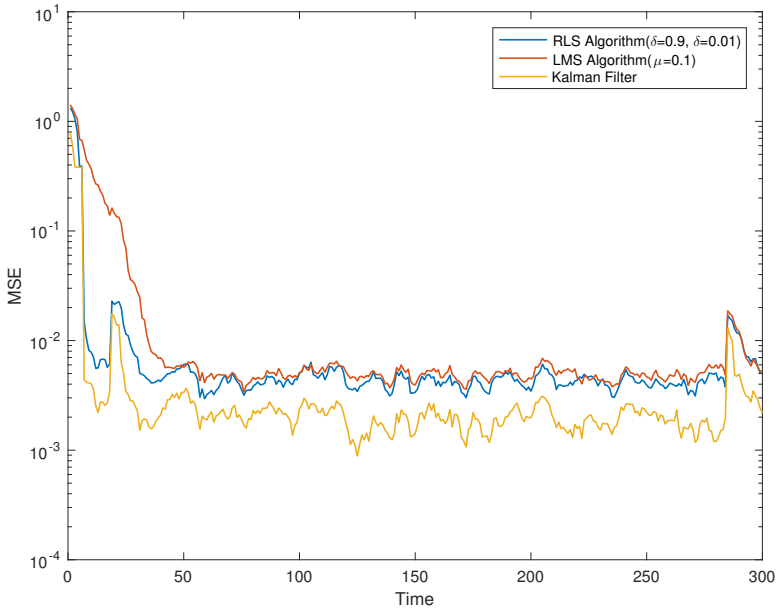

Figure 6.6: MSE for different Algorithms with Dim=8, correlation factors $\alpha = [\alpha_0 = 0.9999, \alpha_1 = 0.9]$ and $p_{00} = 0.99, p_{11} = 0.1$

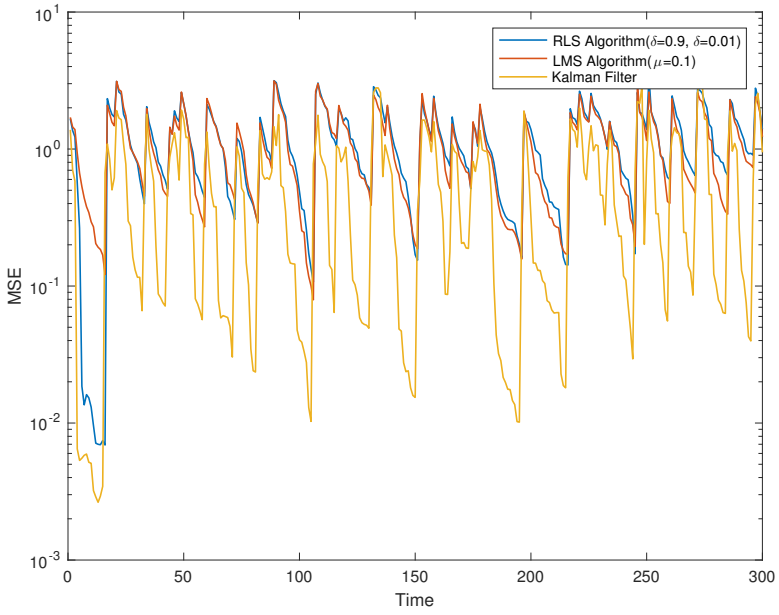

Figure 6.7: MSE for different Algorithms with Dim=8, correlation factors $\alpha = [\alpha_0 = 0.9999, \alpha_1 = 0.1]$ and $p_{00} = 0.9, p_{11} = 0.1$

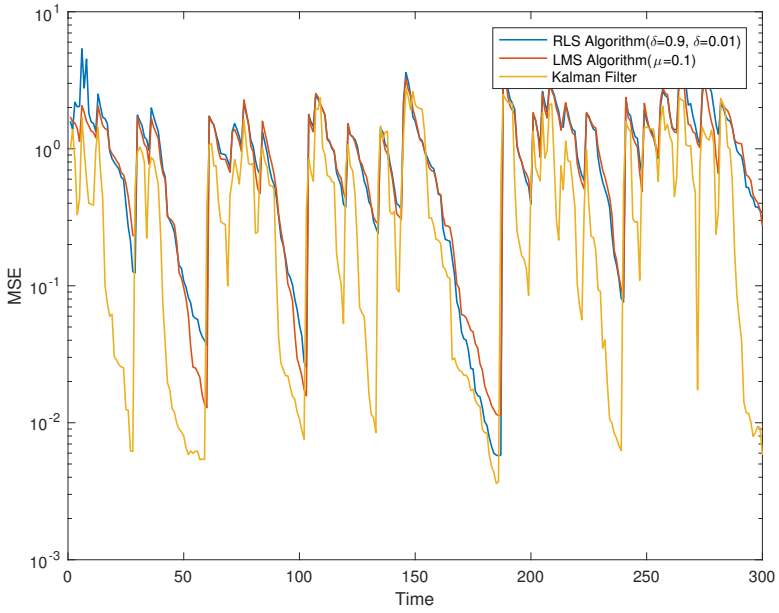

Figure 6.8: MSE for different Algorithms with Dim=8, correlation factors $\alpha = [\alpha_0 = 0.9999, \alpha_1 = 0.1]$ and $p_{00} = 0.9, p_{11} = 0.2$

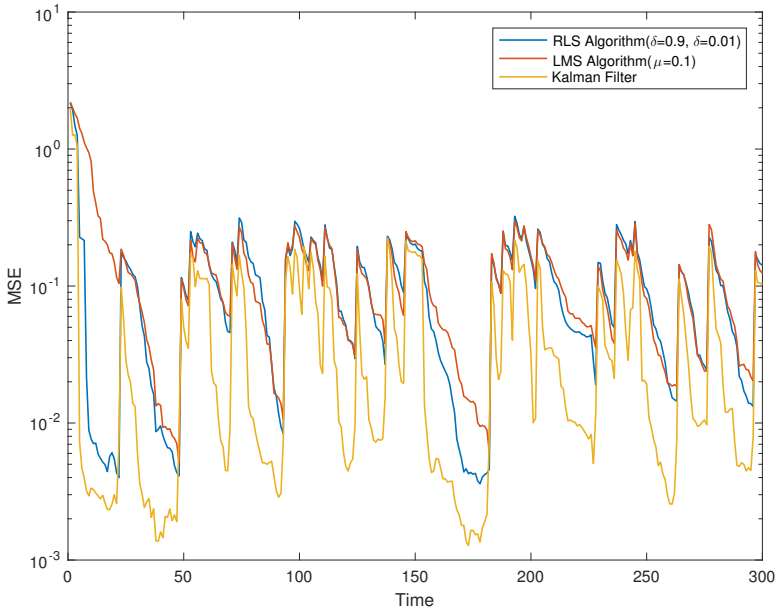

Figure 6.9: MSE for different Algorithms with Dim=8, correlation factors $\alpha = [\alpha_0 = 0.9999, \alpha_1 = 0.9]$ and $p_{00} = 0.9, p_{11} = 0.2$

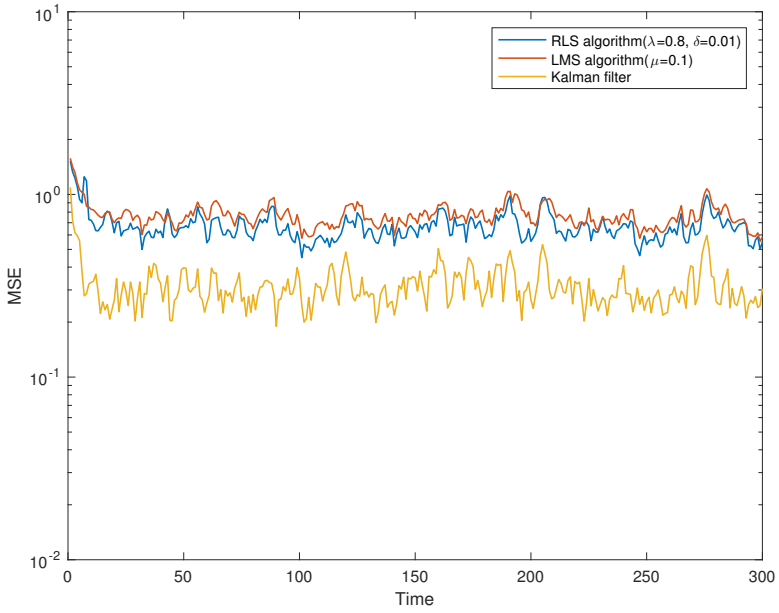

Figure 6.10: MSE for different algorithms with Dim=8, correlation factor $\alpha = 0.9$ and correlation matrix factor $q = 0$

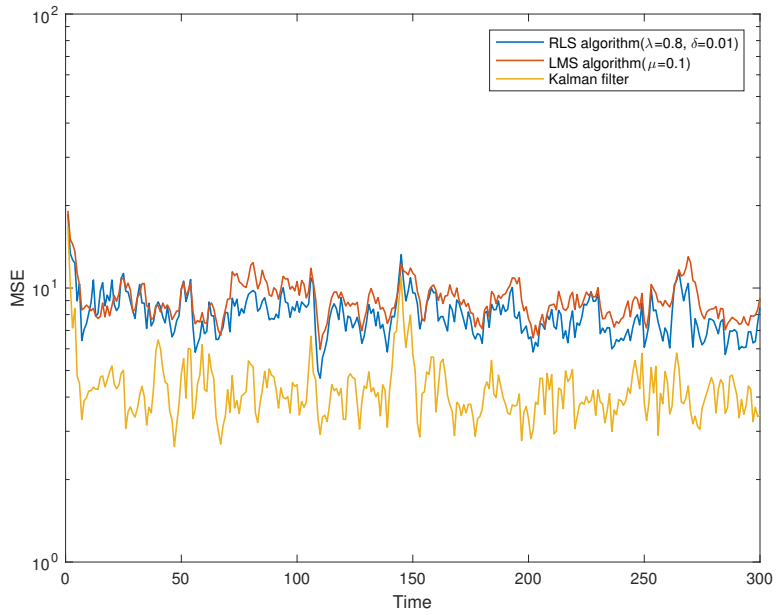

Figure 6.11: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.9$ and correlation matrix factor $q = 0.9$

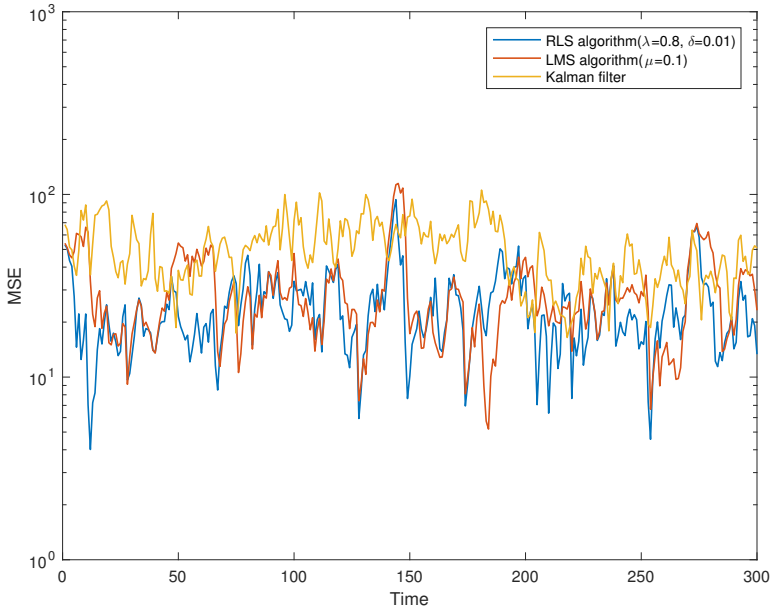

Figure 6.12: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.9$ and correlation matrix factor $q = 0.9999$

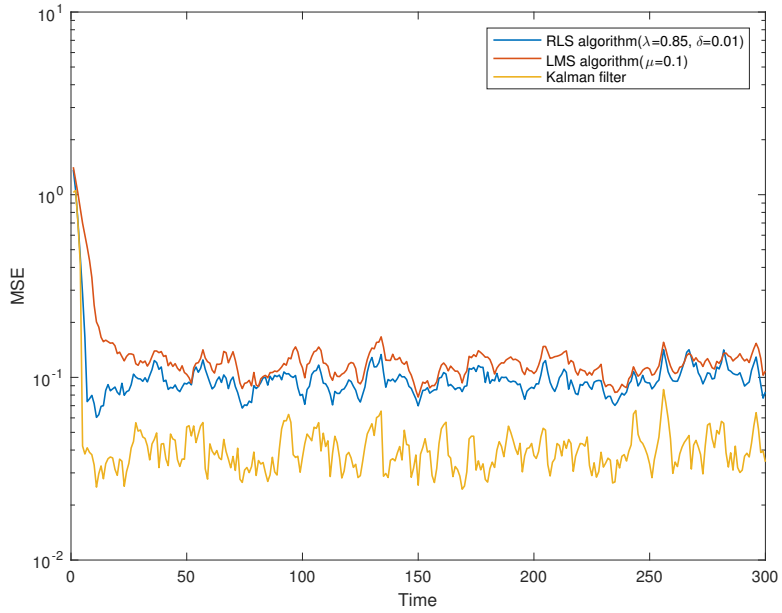

Figure 6.13: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.99$ and correlation matrix factor $q = 0$

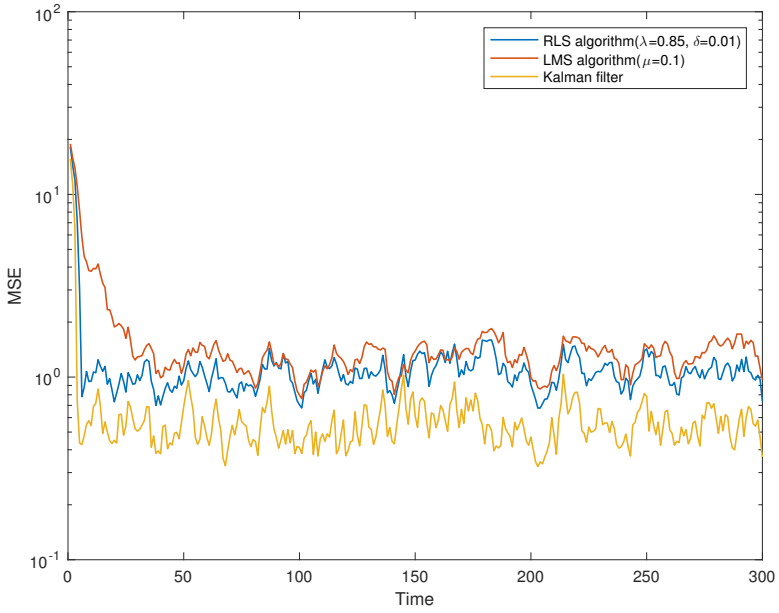

Figure 6.14: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.99$ and correlation matrix factor $q = 0.9$

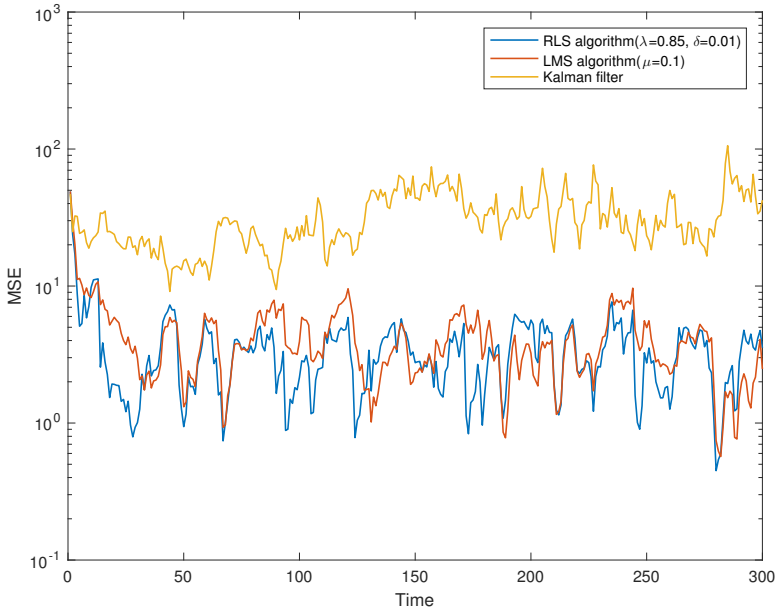

Figure 6.15: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.99$ and correlation matrix factor $q = 0.9999$

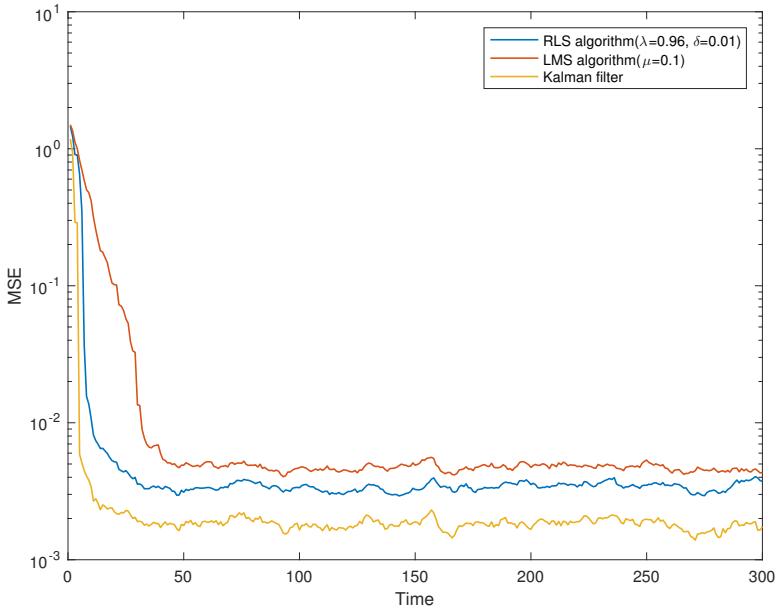

Figure 6.16: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.9999$ and correlation matrix factor $q = 0$

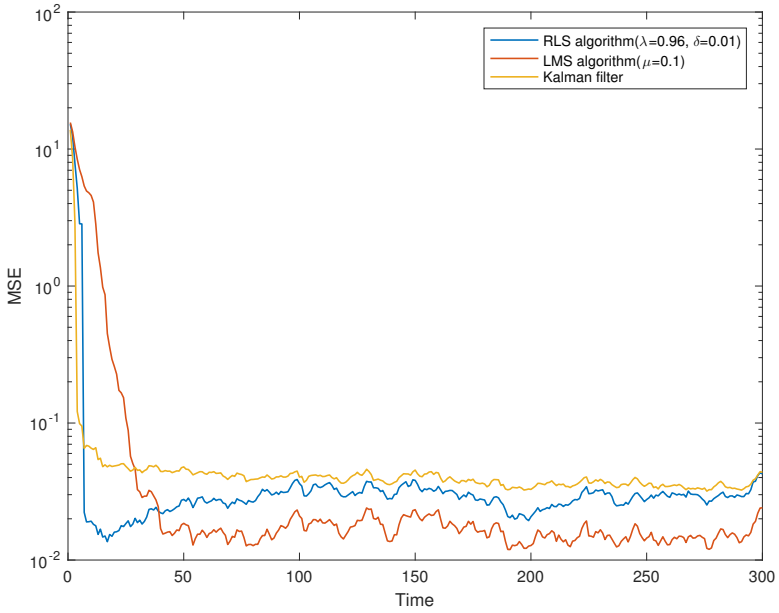

Figure 6.17: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.9999$ and correlation matrix factor $q = 0.9$

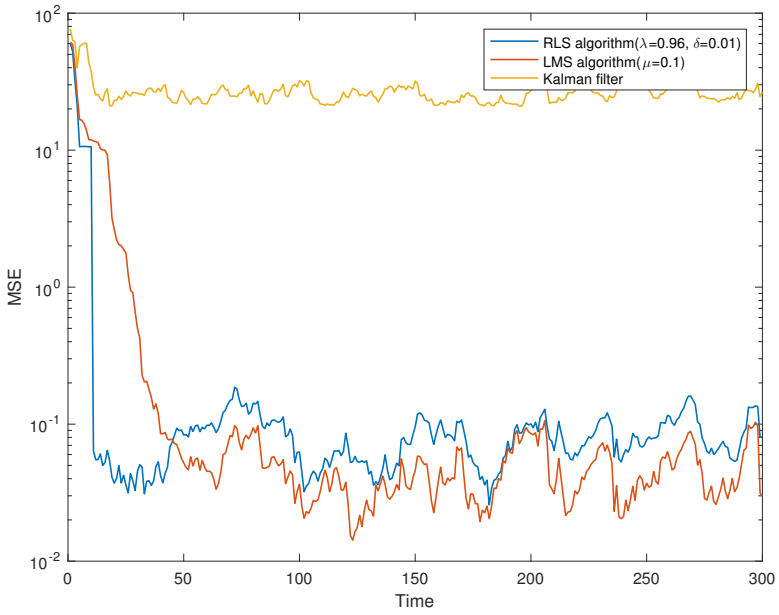

Figure 6.18: MSE for different Algorithms with Dim=8, correlation factor $\alpha = 0.9999$ and correlation matrix factor $q = 0.9999$

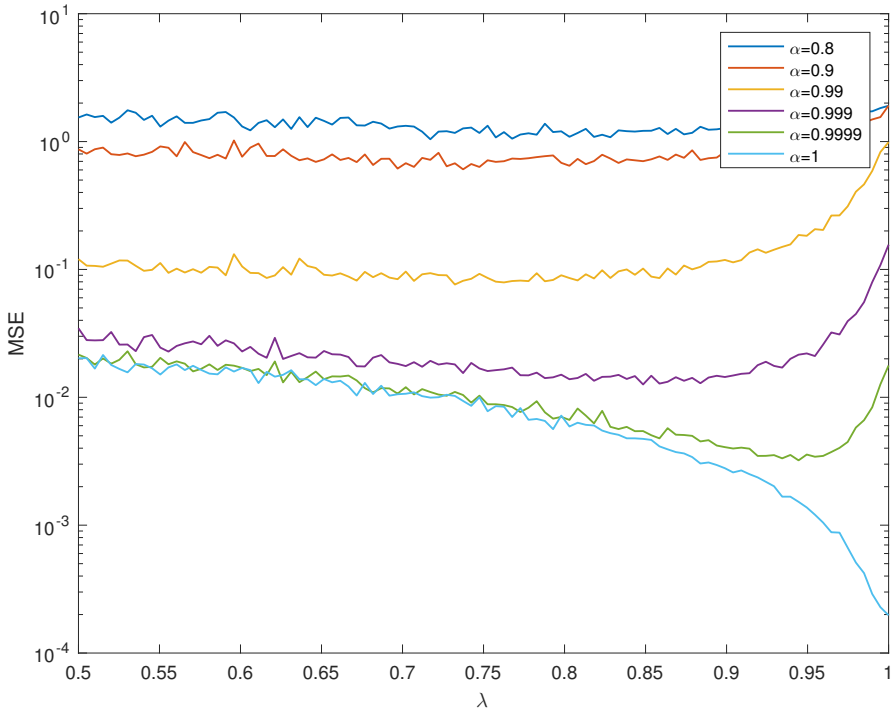

Figure 6.19: MSE for different values of λ in RLS Algorithm (Dim= $8, \delta = 0.01$ and $\text{SNR}_{in} = 30$ dB)

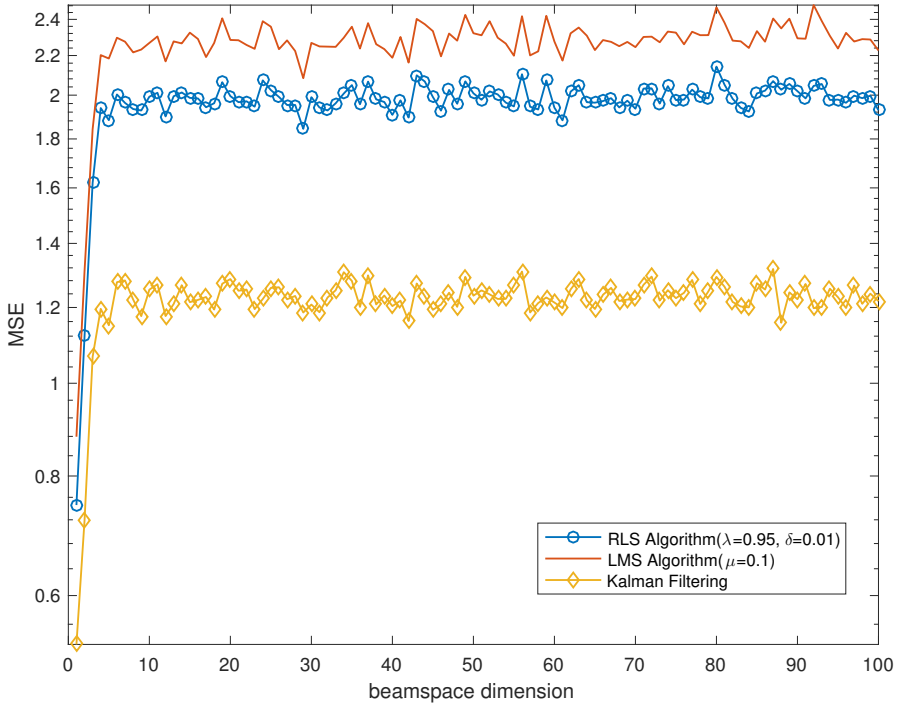

Figure 6.20: MSE of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension ($\alpha = 0, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

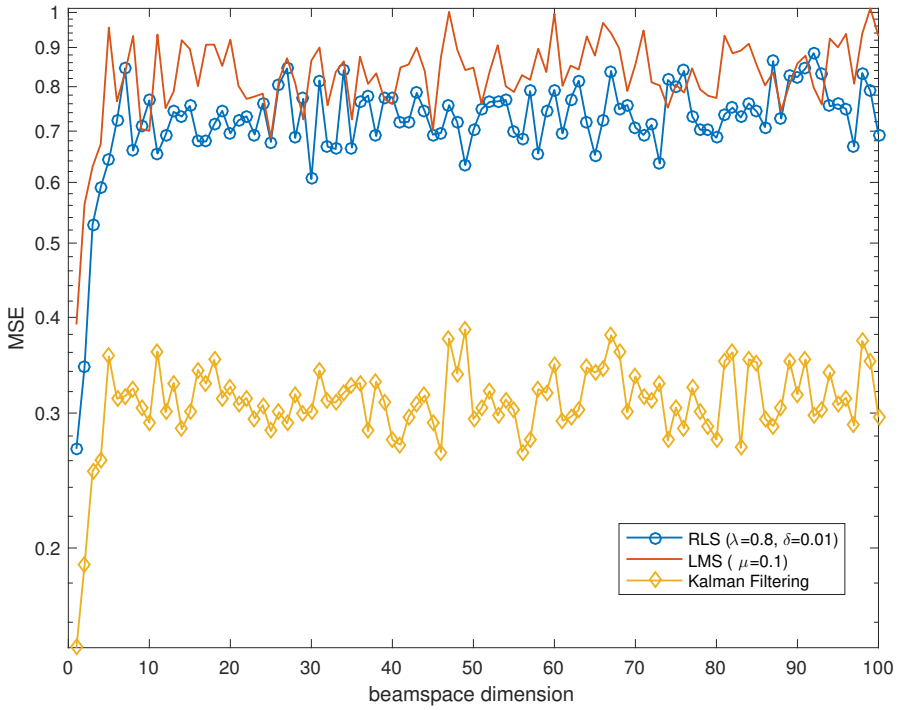

Figure 6.21: MSE of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension ($\alpha = 0.9, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

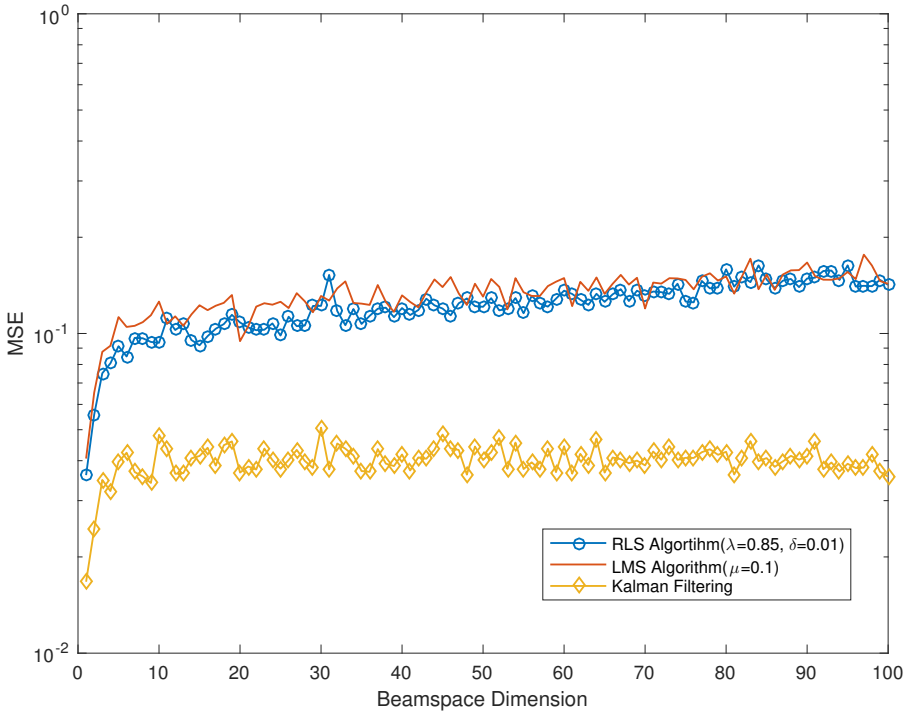

Figure 6.22: MSE of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension ($\alpha = 0.99, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

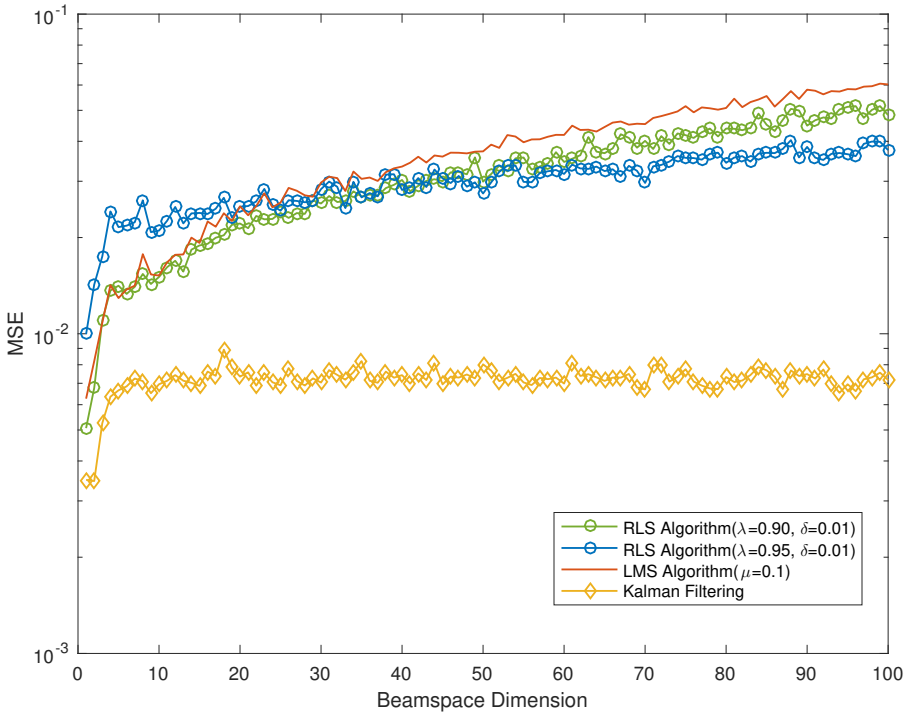

Figure 6.23: MSE of different methods of estimating the channel, RLS algorithm, LMS algorithm and Kalman filtering with respect to beamspace dimension ($\alpha = 0.999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

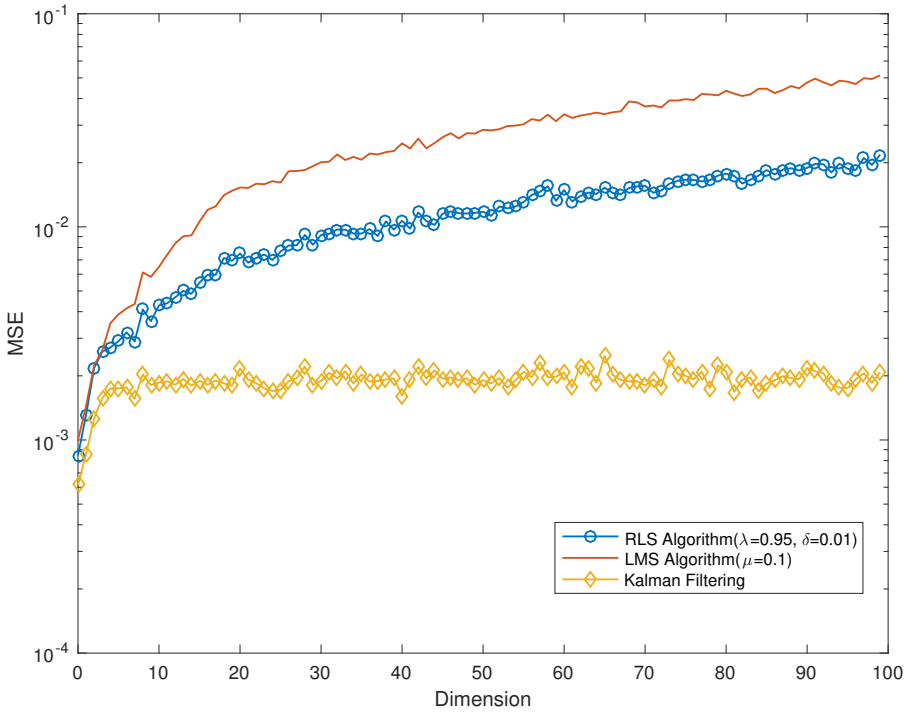

Figure 6.24: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

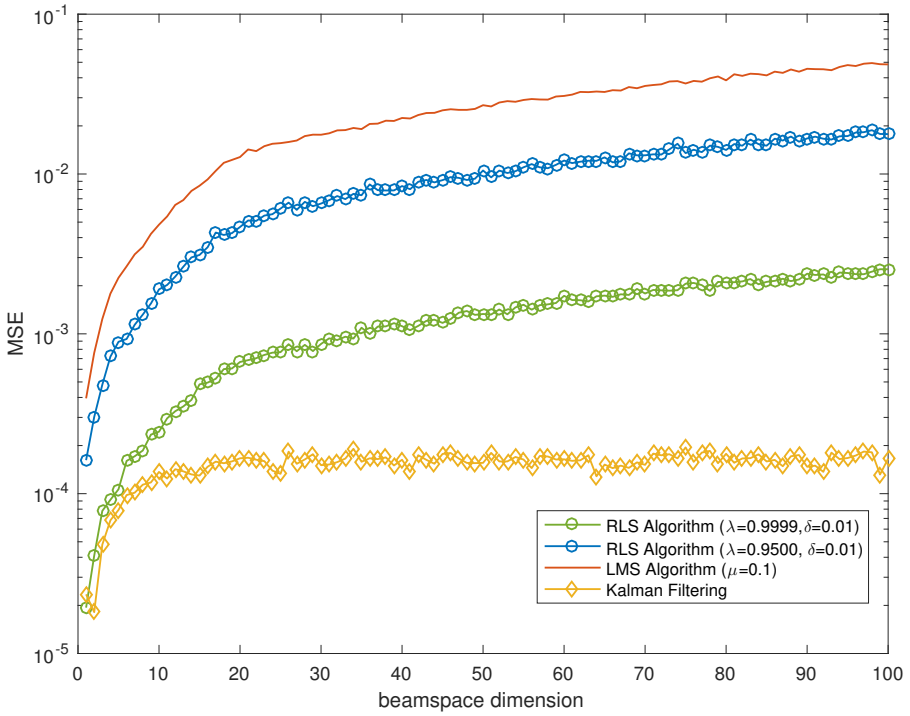

Figure 6.25: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 1, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 30$ dB)

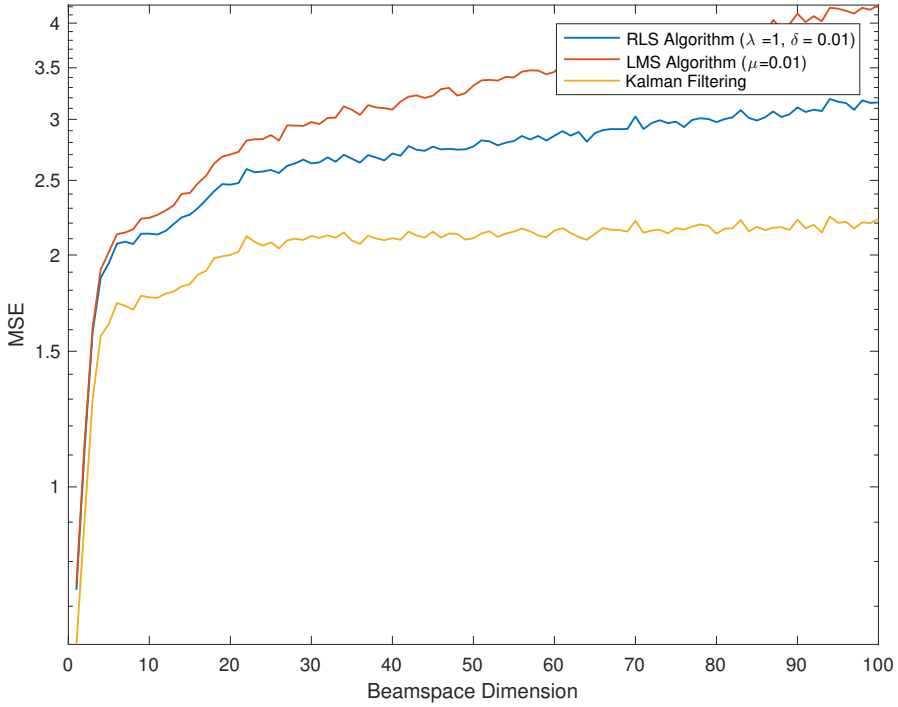

Figure 6.26: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 3$ dB)

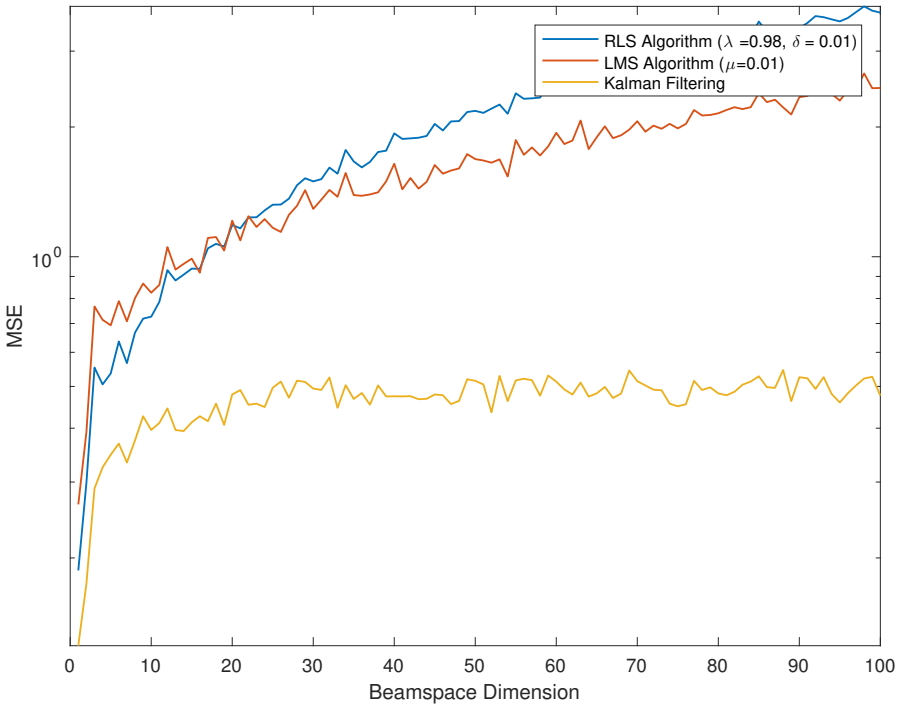

Figure 6.27: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.99, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 3$ dB)

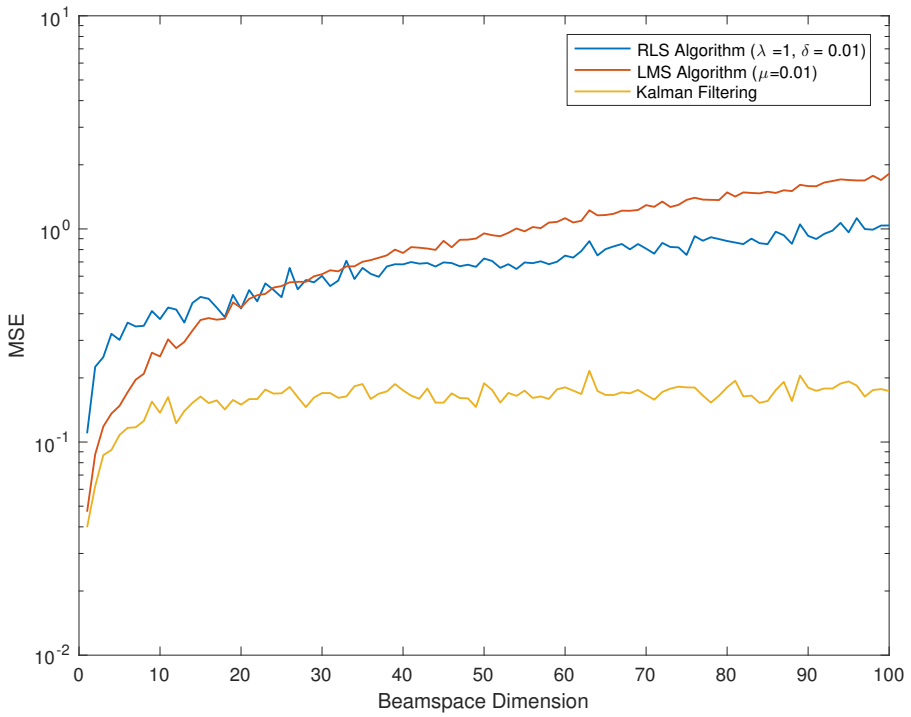

Figure 6.28: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 3$ dB)

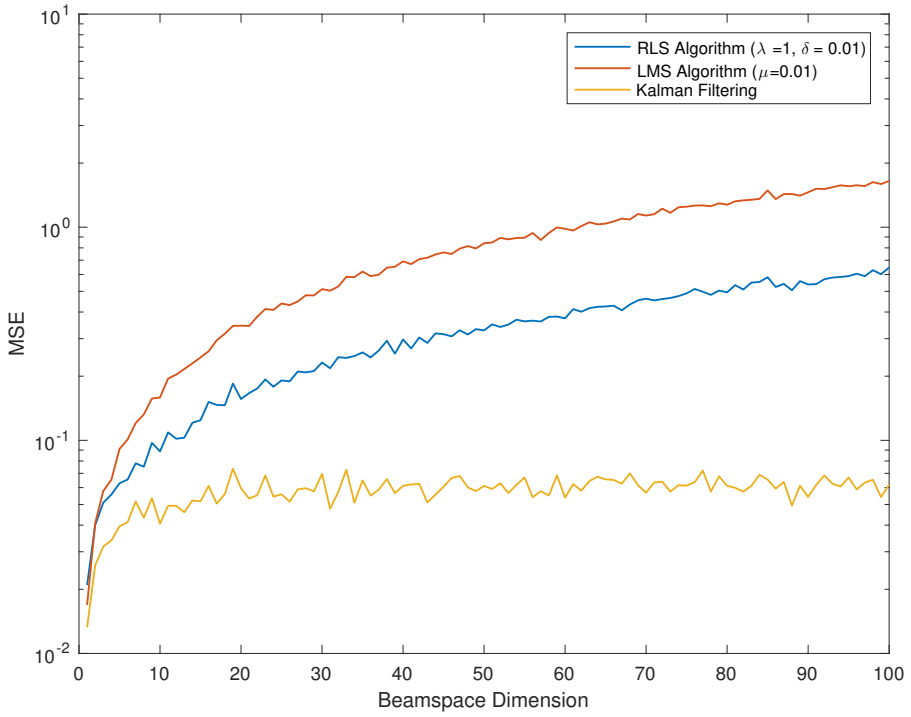

Figure 6.29: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 3$ dB)

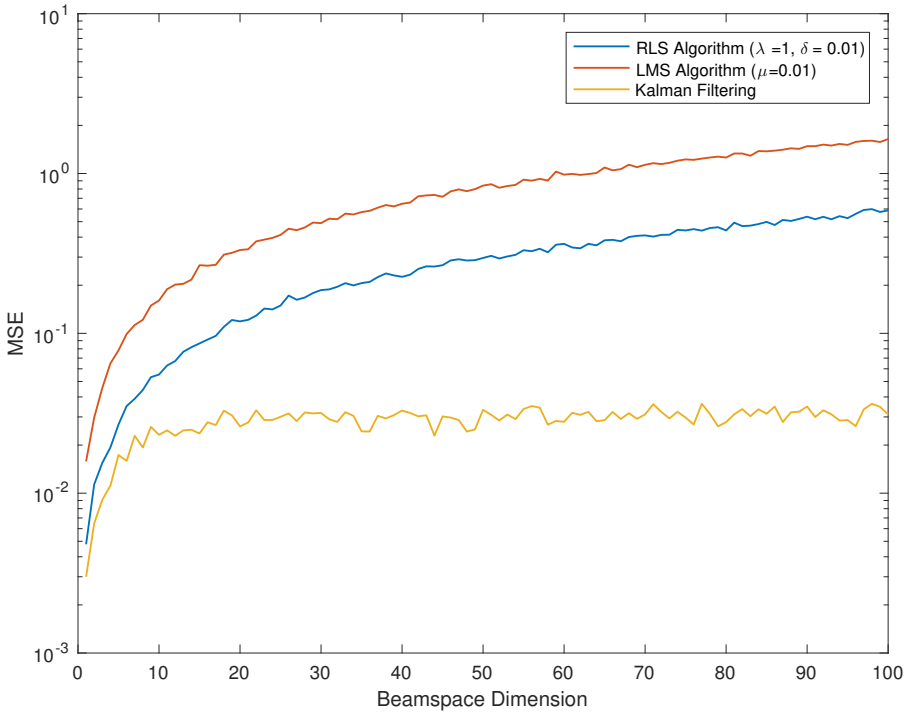

Figure 6.30: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 1, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 3$ dB)

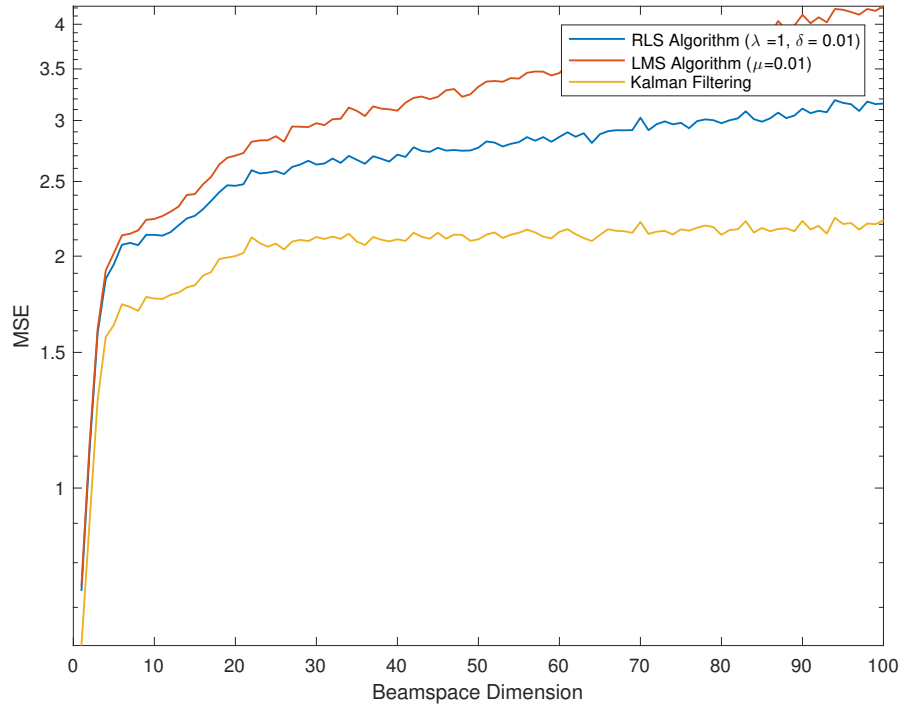

Figure 6.31: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 0$ dB)

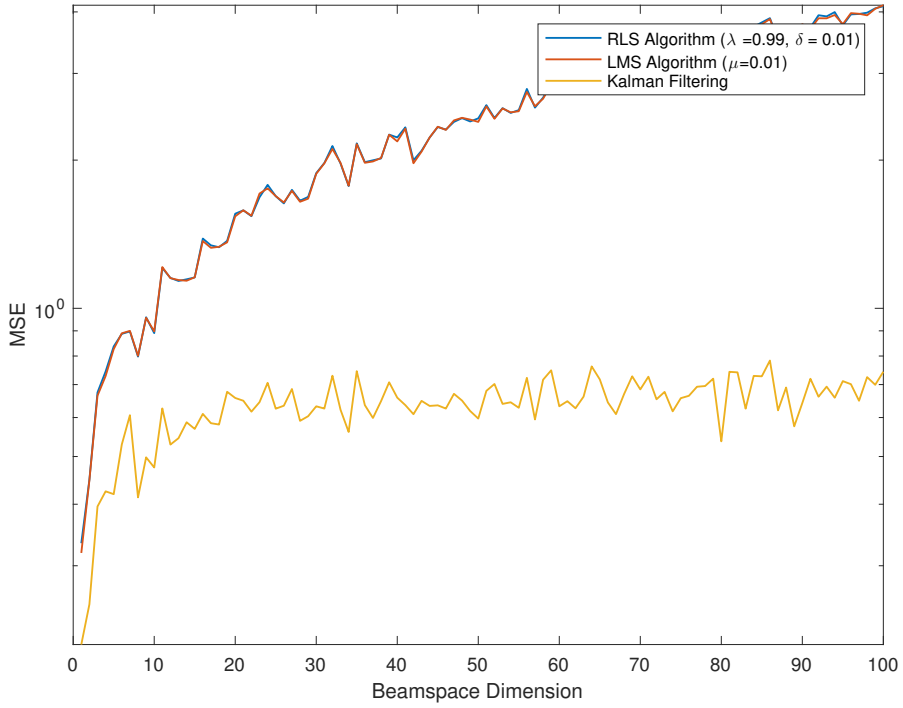

Figure 6.32: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.99, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 0$ dB)

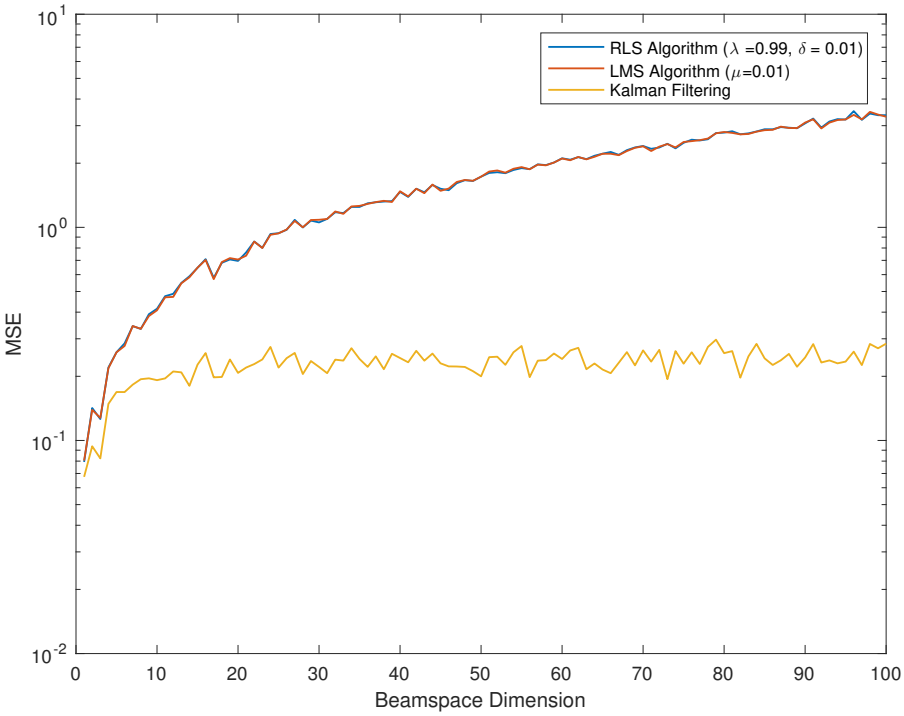

Figure 6.33: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 0$ dB)

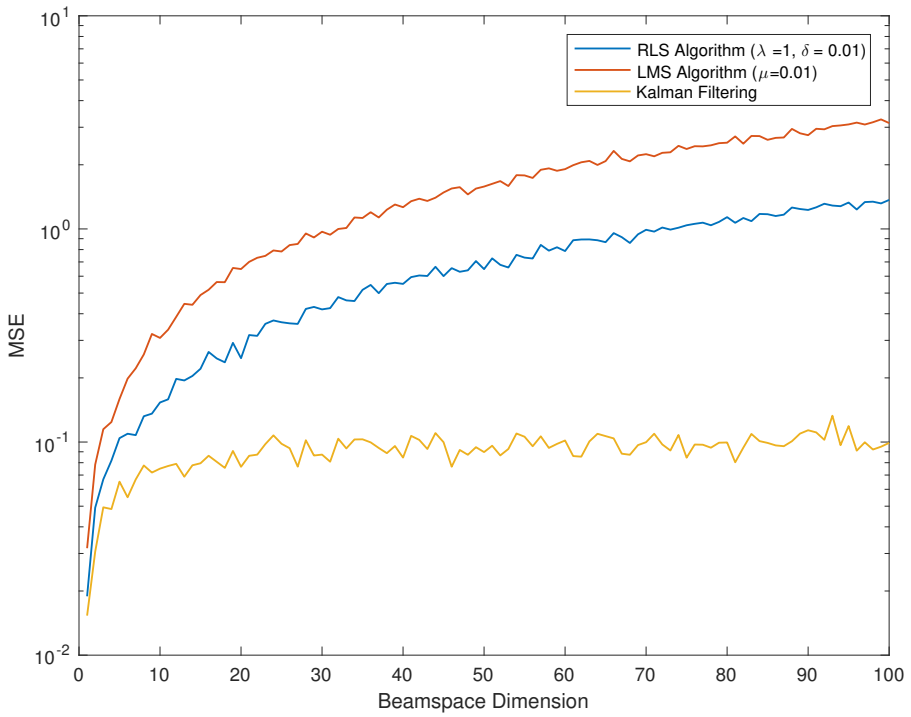

Figure 6.34: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 0.9999, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 0$ dB)

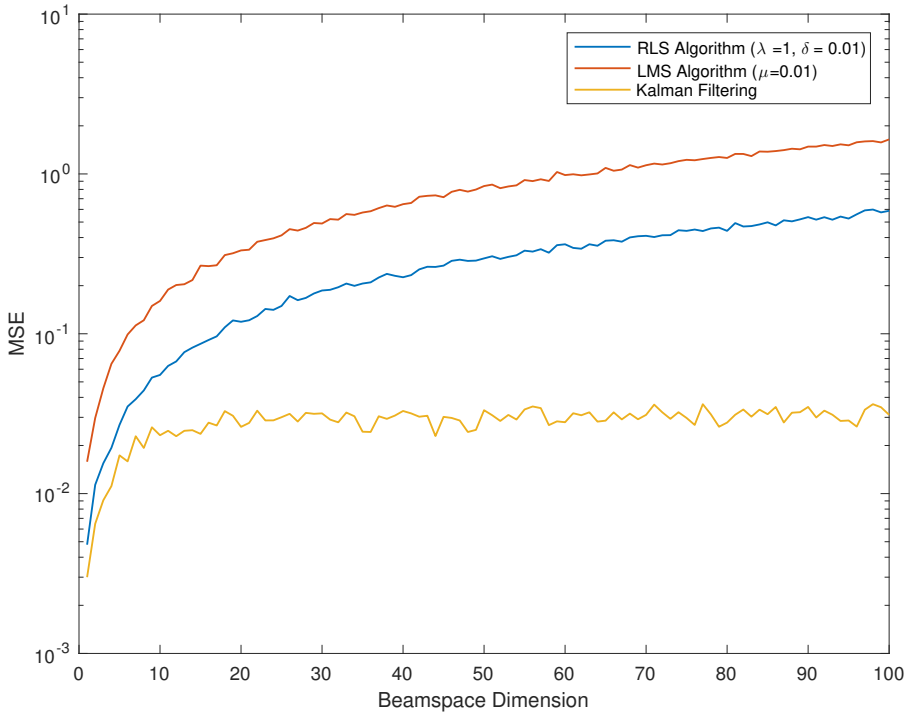

Figure 6.35: MSE of different methods of estimating the channel, RLS Algorithm, LMS Algorithm and Kalman Filtering wrt beamspace dimension ($\alpha = 1, \lambda = 0.95, \delta = 0.01, \mu = 0.1$ and $\text{SNR}_{in} = 0$ dB)

Chapter 7

Conclusion

In this thesis, estimating the channel vector coefficient with different methods by using prebeamforming technique were investigated in massive MIMO systems. This work is based on the idea of dividing users in different groups. There is a model for the system. Based on the model we came up with different algorithms to estimate the channel coefficients. The first algorithm was Kalman filter. Kalman filter is commonly used in different areas such as guidance navigation, control and signal processing. By formulating the filter and using it on our model, the channel vector is estimated. The Kalman filter employs the inversion of a large matrix. Consequently, it is computationally the most demanding of three techniques. On the other hand, its performance in terms of MSE and capacity is the best. The next algorithm is the LMS algorithm which avoids matrix inversion. The trade-off for computational simplicity is in convergence time and performance. The last algorithm which was studied is the RLS algorithm. The algorithm is a recursive method to estimate the channel. The results in this case were dependent on a forgetting factor. By deriving the optimum value for the forgetting factor we could have better performance than LMS Algorithm but worse than Kalman filter was achieved. The complexity of this algorithm is also between those of the Kalman filter and the LMS algorithm.

Also, in chapter 6 we have shown that the Kalman filtering has the maximum capacity values among the algorithms. Comparing these three methods based on MSE and capacity leads to the best performance among them which is Kalman filter. Kalman filter behavior was pretty good comparing the other algorithms.

Besides these works, we have also studied the effect of changing the channel coefficient factor in each time and our model was able to estimate the channel very well, whether the channel coefficient factor is constant or time dependent(in a Markov chain model). This shows that the algorithms are able to estimate the channel in a fast way. Also, the results of simulations for a new Gauss-Markov channel state were studied. In this case, when the channel is more uncorrelated (correlation factor goes to zero), the performance of the algorithms increases. On the other hand, when the correlation factor is near one, the performance of algorithms decreases. Even the performance of the Kalman filter becomes the worst one.

Bibliography

[1] J. G. Andrews, W. Choi S. Buzzi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang. What will 5g be? *IEEE J. Sel. Areas Commun.*, 32:1065–1082, 2014.

[2] Cisco. Visual networking index. *white paper at Cisco.com*, Jun. 2016.

[3] G. Foschini and M. J. Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Pers. Commun.*, 6(3):311–335, Mar. 1998.

[4] E. Telatar. Capacity of multi-antenna gaussian channels. *European Trans. Telecommun.*, 10(6):585–595, Nov./Dec. 1999.

[5] T. L. Marzetta. The case for many (greater than 16) antennas as the base station. *Proc. ITA, San Diego, CA, USA*, Jan. 2007.

[6] T. Marzetta. Noncooperative cellular wireless with unlimited numbers of base station antennas. *IEEE Trans. Wireless Commun.*, 9(11):3590–3600, Sep. 2010.

[7] E. G. Larsson. Very large mimo systems: Opportunities and challenges. 2012.

[8] H. Q. Ngo, M. Matthaiou, T. Q. Duong, and E. G. Larsson. Uplink performance analysis of multiuser mu-simo systems with zf receivers. *IEEE Trans. Veh. Technol.*, 62(9):4471–4483, 2010.

[9] A. Ghosh, T. A. Thomas, M. C. Cudak, R. Ratasuk, P. Moorut, F. W. Vook, T. S. Rappaport, G. R. MacCartney, S. Sun, and S. Nie. Millimeter-wave enhanced local area systems: A high-data-rate approach for future wireless networks. *IEEE J. Sel. Areas Commun.*, 32:1152–1163, Jun. 2014.

[10] A. L. Swindlehurst, E. Ayanoglu, P. Heydari, and F. Capolino. Millimeter-wave massive mimo: The next wireless revolution? *IEEE Commun. Mag.*, 52:56–62, Sep. 2014.

[11] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang. An overview of massive mimo: Benefits and challenges. *IEEE J. Sel. Areas Commun.*, 8:742–758, Oct. 2014.

[12] J. Jose, A. Ashikhmin, T. L. Marzetta, and S. Vishwanath. Pilot contamination and precoding in multi-cell tdd systems. *IEEE Trans. Wireless Commun.*, 10:2640–2651, Aug. 2014.

[13] A. Alkhateeb J.Mo N.Gonzalez-Prelcic and R. W. Heath. Mimo precoding and combining solutions for millimeter-wave systems. *IEEE Commun. Mag.*, 52:122–131, Dec. 2014.

[14] A. Alkhateeb, O .E. Ayach, G. Leus, and R. W. Heath. Channel estimation and hybrid precoding for millimeter wave cellular systems. *IEEE J. Sel. Topics Signal Process.*, 8:831–864, Oct. 2014.

[15] A. Adhikary, J. Nam, J. Y. Ahn, and G. Caire. Joint spatial division and multiplexing: The large-scale array regime. *IEEE Trans. Inf. Theory*, 59:6441–6463, Oct. 2013.

[16] J. Nam, A. Adhikary, J. Y. Ahn, and G. Caire. Joint spatial division and multiplexing: Opportunistic beamforming, user grouping and simplified downlink scheduling. *IEEE J. Sel. Topics Signal Process.*, 8:876–890, Oct. 2014.

[17] D.Kim, G.Lee, and Y.Sung. Two-stage beamformer design for massive mimo downlink by trace quotient formulation. *IEEE Trans. Commun.*, 63:2200–2211, Jun. 2015.

[18] Gokhan M. Guvensen and Ender Ayanoglu. Beamspace aware adaptive channel estimation for single-carrier time-varying massive mimo channels. *Proc. of IEEE International Conference on Communications*, pages 1–7, May. 2017.

[19] O. E. Ayach, S. Rajagopal, Z. Pi S. Abu-Surra, and R. W. Heath. Spatially sparse precoding in millimeter wave mimo systems. *IEEE Trans. Wireless Commun.*, 13:1499–1513, Mar. 2014.

[20] A. Liu and V. Lau. Phase only rf precoding for massive mimo systems with limited rf chains. *IEEE Trans. Signal Process.*, 62:4505–4515, Sep. 2014.

[21] S. Noh, M. D. Zoltowski, and D. J. Love. Training sequence design for feedback assisted hybrid beamforming in massive mimo systems. *IEEE Trans. Commun.*, 64:187–200, Jan. 2016.

[22] A. Adhikary, E. A. Safadi, M. K. Samimi, R. Wang, G. Caire, T. S. Rappaport, and A.F.Molisch. Joint spatial division and multiplexing for mm-wave channels. *IEEE J. Sel. Areas Commun.*, 32:1239–1255, Jun. 2014.

[23] L. You, X.Gao, A. L. Swindlehurst, and W. Zhong. Channel acquisition for massive mimo-ofdm with adjustable phase shift pilots. *IEEE Trans. Signal Process.*, 64:1461–1476, Mar. 2016.

[24] T. S. Rappaport, S. Sun, R. Mayzus, Y. Azar H. Zhao, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez. Millimeter wave mobile communications for 5g cellular: It will work! *IEEE Access*, 1:335–349, May. 2013.

[25] B. M. Sadler L. Tong and M. Dong. Pilot-assisted wireless transmissions: General model, design criteria, and signal processing. *IEEE Signal Process. Mag.*, 21(6):12–25, Nov. 2004.

[26] S. Haykin. *Adaptive Filter Theory, 4th ed.* Prentice- Hall, Englewood Cliffs, NJ, 2002.