TARGET TRACKING WITH CORRELATED MEASUREMENT NOISE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

YEŞİM OKŞAR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

JANUARY 2007

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Prof. Dr. Kerim DEMİRBAŞ
Supervisor

Examining Committee Members

Prof. Dr. Kemal LEBLEBİCİOĞLU (METU, EE) _____

Prof. Dr. Kerim DEMİRBAŞ (METU, EE) _____

Assoc. Prof. Dr. Fahrettin ARSLAN (Ankara Unv., STAT) _____

Assist. Prof. Dr. Çağatay CANDAN (METU, EE) _____

Şebnem SAYGINER (M.S.) (ASELSAN Inc, MST) _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Yeşim OKŞAR

Signature :

ABSTRACT

TARGET TRACKING WITH CORRELATED MEASUREMENT NOISE

OKŞAR, Yeşim
M.S., Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Kerim DEMİRBAŞ

January 2007, 234 pages

A white Gaussian noise measurement model is widely used in target tracking problem formulation. In practice, the measurement noise may not be white. This phenomenon is due to the scintillation of the target. In many radar systems, the measurement frequency is high enough so that the correlation cannot be ignored without degrading tracking performance.

In this thesis, target tracking problem with correlated measurement noise is considered. The correlated measurement noise is modeled by a first-order Markov model. The effect of correlation is thought as interference, and Optimum Decoding Based Smoothing Algorithm is applied. For linear models, the estimation performances of Optimum Decoding Based Smoothing Algorithm are compared with the performances of Alpha-Beta Filter Algorithm. For nonlinear models, the estimation performances of Optimum Decoding

Based Smoothing Algorithm are compared with the performances of Extended Kalman Filter by performing various simulations.

Keywords: Correlated Measurement Noise, Target Tracking, Estimation, Optimum Decoding Based Smoothing Algorithm, Alpha-Beta Filter, Extended Kalman Filter, Decorrelation Method.

ÖZ

İLİNTİLİ ÖLÇÜM GÜRÜLTÜSÜ ETKİSİNDE HEDEF İZLEME

OKŞAR, Yeşim
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Danışmanı: Prof. Dr. Kerim DEMİRBAŞ

Ocak 2007, 234 sayfa

Beyaz Gaussian gürültü ölçüm modeli, hedef izleme problemleri formülasyonunda yaygın olarak kullanılmaktadır. Pratikte ölçüm gürültüsü beyaz gürültü olmayabilir. Bu olay hedefin titreşimden kaynaklanmaktadır. Pek çok radar sisteminde, ölçüm frekansı yeterince yüksektir. Bu nedenle ilinti durumu kestirim performansını azaltmaksızın ihmal edilemez.

Bu tezde ilintili ölçüm gürültüsü etkisinde hedef izleme problemi incelenmiştir. İlintili ölçüm gürültüsü, birinci dereceden Markov modeli kullanılarak modellenmiştir. İlinti etkisi bozucu olarak düşünülmüş ve Optimum Kod Çözümüne Dayalı Düzeltme Algoritması uygulanmıştır. Çeşitli benzetimler yapılarak Optimum Kod Çözümüne Dayalı Düzeltme Algoritması'nın kestirim performansı, doğrusal olan modeller için Alpha-Beta Süzgeç Algoritması ile, doğrusal olmayan modeller için Genişletilmiş Kalman Süzgeci ile karşılaştırılmıştır.

Anahtar Kelimeler: İlintili Ölçüm Gürültüsü, Hedef izleme, Kestirim, Optimum Kod Çözümüne Dayalı Düzeltme Algoritması, Alfa-Beta Süzgeci, Genişletilmiş Kalman Süzgeci, İlintisizleştirme Metodu.

To My Dear Family

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisior, Prof. Dr. Kerim DEMİRBAŞ, for his guidance and support during this work.

I would like to thank to my colleagues at Aselsan Inc. for their support, guidance and valuable advice throughout my thesis work. I am grateful to Aselsan Inc. for the facilities provided for the completion of this thesis.

I would also like to thank to my parents, Yasemin and Ömer COŞKUN, my sister, Münevver COŞKUN and my brother, RIFAT COŞKUN for their encouragement they have given me not only throughout my thesis but also throughout my life.

Last but not least, I am grateful to my husband, İrfan OKŞAR, for his love, encouragement, personal sacrifice and endless patiance besides his technical support and guidance throughout this thesis work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EKF	Extended Kalman Filter
Eq	Equation
Hz	Hertz
ODSA	Optimum Decoding Based Smoothing Algorithm
Q#	Quantization Number
RMS	Root Mean Square
s	Second
var	Variance

CHAPTER 1

INTRODUCTION

The measurement errors in tracking systems are not independent in practice. For example, in radar tracking of extended targets, target scintillation (or glint) causes the range and angle measurement errors to have a finite bandwidth. The bandwidth is proportional to the size of the target, its turn rate with respect to the radar-target line-of-sight, and the transmitted frequency. For tactical applications, typical bandwidths are on the order of several Hz [3, 4].

When the measurement frequency is very much smaller than the error bandwidth, the errors on successive measurements are approximately uncorrelated, and can be treated as white noise. This is often the case in track-while-scan applications, having measurement rates of, say, 0.2 Hz, or less. However, in continuous tracking radars, the measurement frequency is usually high enough that correlation cannot be ignored. It is useful than to have some idea of the impact of the correlation on tracking accuracies [3, 4].

There are various estimation methods proposed for target tracking problem. One of these estimation methods is "*Optimum Decoding Based Smoothing Algorithm (ODSA)"* that obtains a trellis diagram for the target motion and estimates the target track both in clear environment and in presence of interference [1, 2]. An important advantage of ODSA is that it can be used for both linear and nonlinear target models.

In this thesis, target tracking problem with correlated measurement noise is investigated. The correlated measurement noise is modeled by a first-order Markov model [3, 4]. The effect of correlation is thought as interference, and

ODSA [1, 2] is applied. The algorithm and simulation of environment is implemented by *MATLAB* programming language.

The estimation performances of ODSA are compared with the performance of *"Alpha-Beta Filter Algorithm"* for linear models, which was proposed by Rogers [3, 4].

The estimation performances of ODSA are compared with the performance of "*Extended Kalman Filter (EKF)"* for nonlinear models.

In Chapter 2, ODSA that uses the Viterbi decoding algorithm for estimation problems is explained. Parameters used in the ODSA algorithm are described.

In Chapter 3, ODSA is modified so that it can be applied in the presence of correlated measurement noise. To handle the correlation effect, firstly ODSA is implemented by applying decorrelation process [3, 4] on correlated measurements and secondly, ODSA is implemented by a method proposed in this thesis, which treats correlated measurement noise as interference. The simulation results of decorrelation method and proposed method are compared.

In Chapter 4, simulation results for ODSA with correlated measurement noise are presented. The effects of the parameters of ODSA parameters on the estimation performance are discussed.

In Chapter 5, the target tracking model in the presence of correlated noise proposed by Rogers [3, 4] is discussed. Alpha-Beta Filter Algorithm is explained and implemented.

In Chapter 6, simulation results for Alpha-Beta Filter Algorithm with correlated measurement noise are presented. The effects of the parameters of Alpha-Beta Filter Algorithm on the estimation performance are discussed.

In Chapter 7, simulation results of ODSA with correlated measurement noise are compared with simulation results of Alpha-Beta Filter Algorithm with correlated measurement noise.

In Chapter 8, EKF is explained and it is modified so that it can be applied in the presence of correlated noise by using decorrelation method [3, 4].

In Chapter 9, simulation results of ODSA with correlated measurement noise are compared with simulation results of EKF with correlated measurement noise.

In Chapter 10, the conclusions are given evaluating all simulation results obtained in this thesis.

In Appendix A, possible values and corresponding probabilities of the discrete random variable approximating the Gaussian distributed continuous random variables up to 20 possible values are given. These values are used by the ODSA algorithm while obtaining the trellis diagram for the target motion model.

In Appendix B, derivation of the solution of the Riccati Equations for the covariance matrix, P, of Alpha-Beta Filter Algorithm is given.

CHAPTER 2

OPTIMUM DECODING BASED SMOOTHING ALGORITHM

In this chapter, a state estimation algorithm for discrete models with or without interference is presented. This estimation algorithm is *"Optimum Decoding Based Smoothing Algorithm (ODSA)"* which is based on *"Viterbi Decoding Algorithm"*.

2.1 Models and Assumptions [1]

"Optimum Decoding Based Smoothing Algorithm" is an estimation algorithm, which can be applied for both linear and nonlinear estimation problems modeled as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = f(k, x(k), u(k), w(k)) \\ &\textit{Measurement model,} && z(k) = g(k, x(k), v(k)) \end{aligned} \tag{2.1}$$

in clear environment. In the presence of interference, the interference parameter is added to the measurement model as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = f(k, x(k), u(k), w(k)) \\ &\textit{Measurement model,} && z(k) = g(k, x(k), I(k), v(k)) \end{aligned} \tag{2.2}$$

Parameters used in Equations (2.1) and (2.2) are defined as follows:

- *x(0)* is an *nx1* initial state Gaussian distributed random vector (which determines the considered target location at time *0*),
- *x(k)* is an *nx1* (target) state vector at time *k* (which determines the considered target location at time *k*),
- *u(k)* is a *qx1* pilot-command vector at time *k* with known statistics,
- *w(k)* is a *px1* Gaussian distributed disturbance noise vector at time *k* with zero mean and known statistics,
- *v(k)* is an *lx1* Gaussian distributed measurement noise vector at time *k* with zero mean and known statistics,
- *I(k)* is an *mx1* interference vector with known statistics,
- *z(k)* is an *rx1* measurement vector at time *k,*
- Time *k* is time $t_0 + kT_0$ where t_0 and T_0 are the initial time and the measurement interval respectively.

Furthermore, *f(k, x(k), u(k), w(k))*, *g(k, x(k), v(k))* and *g(k, x(k), I(k), v(k))* are linear or nonlinear vectors with appropriate dimensions. The random vectors *x(0)*, *w(j)*, *w(k)*, *v(l), v(m), I(n)* and *I(p)* are assumed to be independent for all *j*, *k*, *l*, *m*, *n*, *p*. The goal is to estimate the state sequence {*x(0)*, *x(1)*,…, *x(L)*} by using the measurement sequence {*z(1)*, *z(2)*,…, *z(L)*} where *L* is a chosen integer.

2.2 Quantization of States and Transition Probabilities [1]

This section describes a type of quantization for target states and some difficulties in calculating transition probabilities between quantization levels.

Let *x(k)* be a random vector whose range is in the space R^n (*n* dimensional Euclidian space). Let us divide R^n into nonoverlapping subspaces R_i^n and assign a unique value x_{qi} to each subspace R_i^n, where the subscript *q* is quantization.

Definition 2.1: A function $x_q(.) \triangleq Q\{x(.)\}$ is a quantizer for the state *x(.)* if the following hold:

1) A function $x_q(.) \triangleq Q\{x(.)\} = x_{qi}$ whenever $x(.) \in R_i^n$; and
2) x_{qi} is unique for each R_i^n

Definition 2.2: The function $x_q(.)$ is the quantized state vector at time *(.)*, and its possible values are called quantization levels of the state *x(.)*.

Definition 2.3: Subspace R_i^n is called gate R_i^n.

Definition 2.4: The value x_{qi} is called the quantization level for the gate R_i^n.

Definition 2.5: The transition probability $\pi_{jm}(k)$ is the probability that the state *x(k+1)* will lie in the gate R_m^n when the state *x(k)* is in the gate R_j^n i.e.:

$$\pi_{jm}(k) \triangleq prob\left\{x(k+1) \in R_m^n \,\middle|\, x(k) \in R_j^n\right\} \tag{2.3}$$

In Figure 1, gate R_i^n, quantized state value x_{qi} and the transition probability $\pi_{jm}(k)$ can be seen schematically.

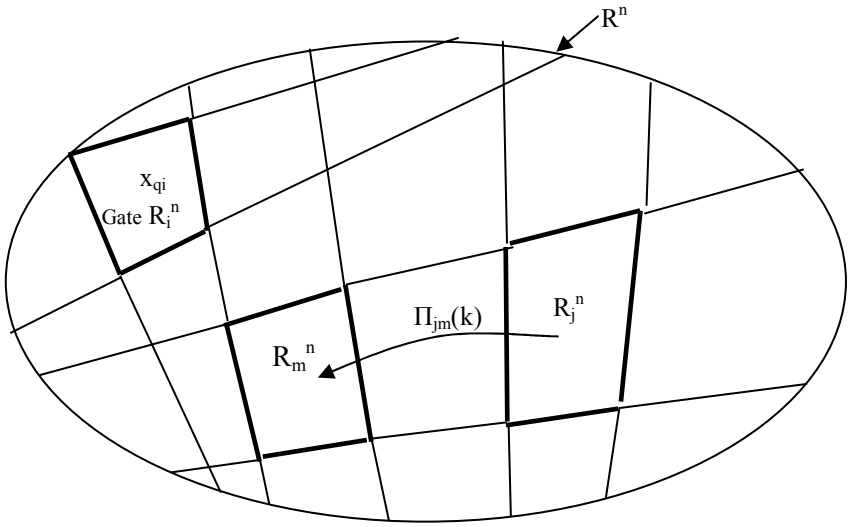

Figure-1: Quantization of states and transition probabilities

The transition probability $\pi_{jm}(k)$ is a conditional probability and can be rewritten as:

$$\pi_{jm}(k) = \frac{prob\{x(k+1) \in R_m^n, x(k) \in R_j^n\}}{prob\{x(k) \in R_j^n\}} \tag{2.4}$$

$$= \left[\int_{R_j^n} p\big(x(k)\big) dx(k) \right]^{-1} \times \left[\int_{R_j^n} \int_{R_m^n} p\big(x(k+1), x(k)\big) dx(k+1)\, dx(k) \right]$$

$$= \left[\int_{R_j^n} p\big(x(k)\big) dx(k) \right]^{-1} \times \left\{ \int_{R_j^n} \left[\int_{R_m^n} p\big(x(k+1) \mid x(k)\big) dx(k+1) \right] p\big(x(k)\big)\, dx(k) \right\}$$

(2.5)

where *p(x(k+1), x(k))* is the joint probability density function of *x(k+1)* and *x(k)*; *p(x(k))* is the probability density function of *x(k)*; and *p(x(k+1)|x(k))* is the conditional probability density function of *x(k+1)* given *x(k)*.

It is not usually easy to evaluate the transition probability $\pi_{jm}(k)$ analytically. The difficulties are due to the shapes of the gates R_i^n and R_m^n and the statistics of the disturbance-noise vector *w(.)* and the initial state vector *x(0)*.

In order to see this, consider the following linear motion example: $x(k+1) = Ax(k) + w(k)$, where *x(0)* is an *nx1* Gaussian initial state vector; *x(k)* is an *n x 1* state vector at time *k*; *w(k)* is an *nx1* Gaussian disturbance vector at time *k*; and *A* is a constant transition matrix with appropriate dimension. Moreover, the random vectors *x(0)*, *w(k)*, and *w(l)* are assumed to be statistically independent for all *k* and *l*. Hence *x(k+1)* and *x(k)* are linear transformations of the Gaussian random vectors *x(0)*, *w(1)*,…., and *w(k)*. Thus, *p(x(k))* and *p(x(k+1)|x(k))* are normal density functions. Therefore, the evaluation of $\pi_{jm}(k)$ is not analytically possible. The problem is more difficult if the motion model is nonlinear. If the transition probability $\pi_{jm}(k)$ needs to be calculated, it should be performed numerically. Even this may be difficult. In other words, the evaluation of the exact transition probabilities between gates is not practical. Therefore, section 2.3 discusses an approximate target motion model called *"finite state model"* which is obtained by approximating the disturbance noise vector *w(k)* and the initial state vector *x(0)* by discrete random vectors (see Appendix A), and by quantizing the state *x(k)* as previously described for all *k=1,2, ...* For this finite-state model, the transition probabilities can be easily calculated.

2.3 A Finite-State Model for the Target Model [1]

Gates are assumed to be generalized rectangles with origin R_0^n. The quantization levels for gates are assumed to be the center of the gates.

$$x_q(.) \triangleq Q(x(.)) = x_{qi} \qquad \text{if } x(.) \in R_i^n \tag{2.6}$$

For each *k,* the disturbance noise vector *w(k)* is approximated by a discrete random vector $w_d(k)$. This random vector can have one of the possible values $w_{d1}(k)$, $w_{d2}(k)$, ..., $w_{dm_k}(k)$ with corresponding probabilities $p_{d1}(k)$, $p_{d2}(k)$, …, $p_{dm_k}(k)$. Similarly, the initial state vector *x(0)* is approximated by a discrete random vector $x_d(0)$ whose possible values are $x_{d1}(0)$, $x_{d2}(0)$, ..., $x_{dn_0}(0)$ with corresponding probabilities $p_{d1}(k)$, $p_{d2}(k)$, …, $p_{dn_0}(k)$. The positive integers m_k and n_0 are chosen such that the random vectors *w(k)* and *x(0)* are satisfactorily approximated by the discrete random vectors $w_d(k)$ and $x_d(0)$ for the considered estimation problem respectively.

Furthermore, by replacing *w(k)* and *x(0)* with discrete random vectors $w_d(k)$ and $x_d(0)$ respectively, and then quantizing the states by Eq. (2.7), the target-motion model is reduced to a finite state model.

$$x_q(k+1)=Q(f(k, x_q(k), u(k), w_d(k)) \tag{2.7}$$

where $Q\{.\}$ is the quantizer. $x_q(k)$ is the quantized state vector at time *k* and its possible values are $x_{q1}(k)$, $x_{q2}(k)$,..., $x_{qn_k}(k)$ where n_k is the number of possible quantization levels of the state vector *x(k)*. In other words, the quantization levels of *x(0)* are assumed to be equal the possible values of the discrete random vector *x(0)*.

The transition probability $\pi_{jl}(k)$, which is defined by the conditional probability that the quantized state vector $x_q(k+1)$ will be equal to the quantization level x_{ql} for gate R_l^n, given that the quantized state vector $x_q(k)$ is equal to the quantization level x_{qj} for gate R_j^n is determined as follows:

$$\pi_{jl}(k) \triangleq prob\left\{x_q(k+1) = x_{ql} \middle| x_q(k) = x_{qj}\right\} \tag{2.8}$$

Assume that the $x_q(k)$ is equal to x_{qj} for gate R_j^n. The transitions from this quantization level to others are determined by $w_d(k)$ and the function $Q(f(k, x_q(k)= x_{qj}, u(k), w_d(k))$. If $w_d(k)$ has m_k discrete values $(w_{d1}(k), w_{d2}(k), \ldots, w_{dm_k}$, then $x_q(k+1)$ can take at most m_k various quantization levels. If the function $f(k, x_q(k)= x_{qj}, u(k), w_d(k))$ maps x_{qj} into another gate, say R_i^n for only one possible value, say $w_{di}(k)$, of the discrete random vector $w_d(k)$, then the transition probability $\pi_{ji}(k)$ from gate R_j^n to gate R_i^n is the probability that the possible value $w_{di}(k)$ of $w_d(k)$ occurs. Besides, if the function $f(k, x_q(k)= x_{qj}, u(k), w_d(k))$ maps x_{qj} into another gate, for more than one possible value, say $w_{d1}(k)$ and $w_{d2}(k)$ of $w_d(k)$, the transition probability, $\pi_{jl}(k)$, is the probability that the discrete random vector $w_d(k)$ is equal to either of the possible values $w_{d1}(k)$ or $w_{d2}(k)$, i.e., $\pi_{jl}(k) = \sum_n p_{dn}(k) = p_{d1}(k) + p_{d2}(k)$, where the summation is over all n such that $Q(f(k, x_q(k)= x_{qj}, u(k), w_{dn}(k))= x_{ql}$

Using the finite state model, the target motion can be represented by a trellis diagram.

2.4 A Trellis Diagram for the Target Motion [1]

Assuming the quantized state vector $x_q(k)$ has n_k possible values which are $x_{q1}(k), x_{q2}(k), \ldots, x_{dn_k}(0)$, the target motion can be represent by a graph. On this graph, there are some conventions, which are the followings:

1) Each possible value of $x_q(k)$ is represented on the k^{th} column by a point (sometimes called node) with the corresponding quantization level so that the k^{th} column contains the possible quantization levels of $x(k)$ (in other words, the possible gates in which the target can lie at time k) where $k=0, 1, 2, \ldots$.

2) The transition from one quantization level to another is represented by a line having a direction indicating the direction of the target motion.

Hence, the target motion from time zero to time L can be represented by a directed graph shown in Figure 2, which is called the "*trellis diagram*" for the target motion from time zero to time L.

Definition 2.6: A path in the trellis diagram is any sequence of directed lines where the final vertex of one is the initial vertex of the next.

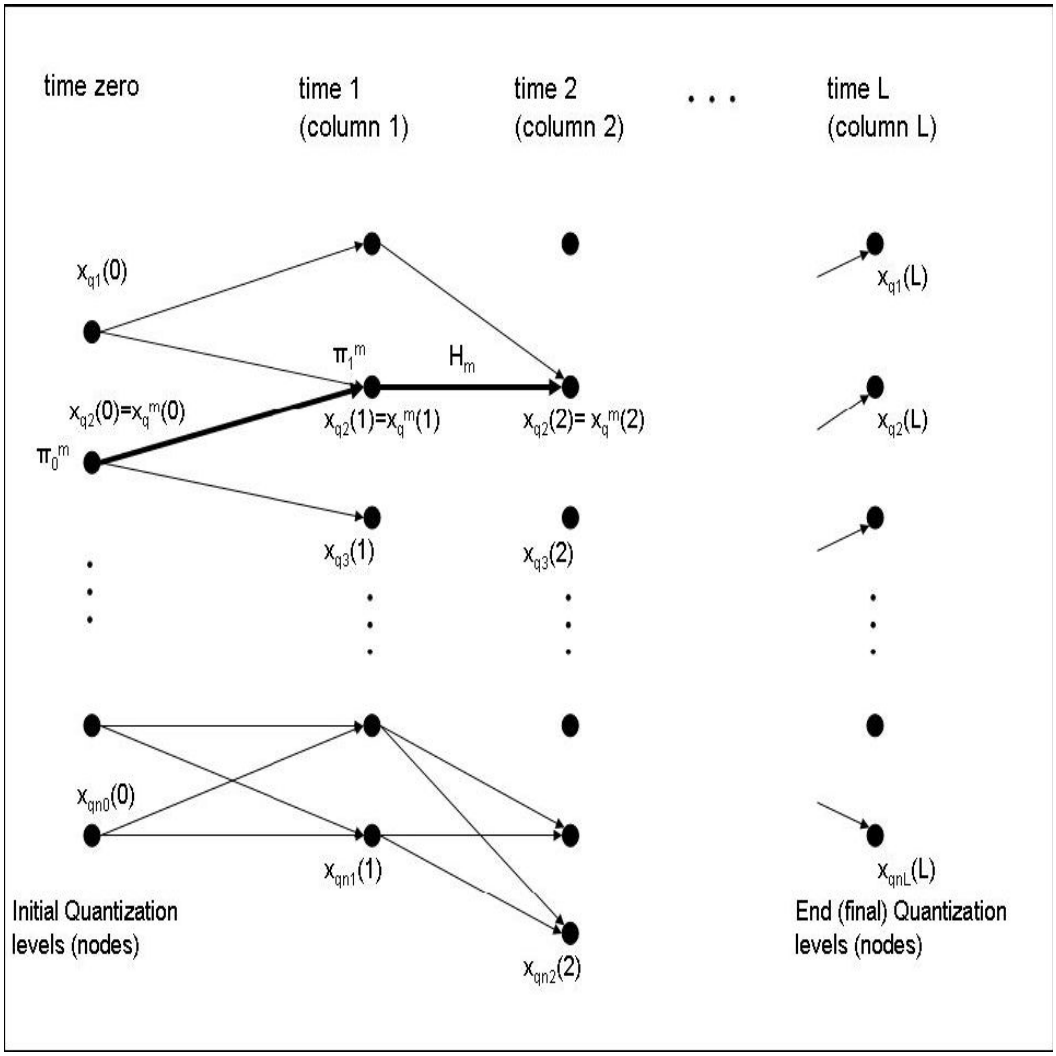

Figure-2: The trellis diagram for the target motion

2.5 Approximate Measurement Models [1]

The target motion model has been reduced to a finite-state model which uses the quantized state vector $x_q(.)$ as described in the previous sections. However, the measurement model in Equations (2.1) and (2.2) uses the state

vector $x(.)$. Thus, in the measurement model in Equations (2.1) and (2.2), by replacing the state vector $x(k)$ with the quantized state vector $x_q(.)$, the following approximate measurement models are obtained:

$$z(k) = \begin{cases} g(k, x_q(k), v(k)), & \text{in clear environments} \\ g(k, x_q(k), I(k), v(k)), & \text{in the presence of interference} \end{cases} \tag{2.9}$$

From now on, the models in the equation in (2.9) is used to refer to the measurement models.

Considering the trellis diagram in Figure 2, the state estimation process will be performed from time zero up to and including time L. Therefore, the trellis diagram is drawn from time zero to time L. Time zero refers to the initial state. The following symbols, which are used throughout further analyses, are defined as:

n_i : Number of quantization levels for the gates in which the target may lie at time i; in other words, the number of possible values of the quantized state vector $x_q(i)$, where $i = 0, 1, 2, ..., L$.

$\tilde{x}(i)$: Set of all the quantization levels for the gates in which the target may lie at time i, namely, $\tilde{x}(i) \triangleq \{x_{q1}(i), x_{q2}(i), ..., x_{qn_i}(i)\}$ where $i = 0, 1, 2, ..., L$.

M : Number of possible paths through the trellis diagram. This number is equal to or less than $\prod_{j=0}^{L} n_j$

H_m : The m^{th} path through the trellis diagram, indicated by a bold line in Figure 2.

$x_q^m(i)$: Quantization level for the gate in which the target lies at time *i* when it follows path H_m. In other words, the possible value of the quantized state vector $x_q(i)$ through which the m^{th} path passes. For example, in trellis diagram of Figure 2,

$$x_q^m(0) = x_{q2}(0);\ x_q^m(1) = x_{q2}(1);\ \ x_q^m(2) = x_{q2}(2), \ldots$$

π_0^m : Probability that the possible value of the initial state vector $x_d(0)$ from which the m^{th} path starts occurs, namely, $\pi_0^m = prob\{ x_d^m(0) = x_q^m(0) \}$. For example, in trellis diagram of Figure 2, $\pi_0^m = prob\ \{ x_q(0) = x_{q2}(0) \}$.

π_i^m : Transition probability from the $(i-1)^{th}$ gate for the m^{th} path. In other words, it is the transition probability that the target vector will be at the i^{th} quantization level (node) of path H_m at time *i* when it is at the $(i-1)^{th}$ quantization level (node) of path at time $i-1$, that is

$$\pi_i^m \triangleq prob\ \{ x_q(i) = x_q{}^m(i) \big| x_q(i-1) = x_q{}^m(i-1) \}$$

π_0^m : Maximum of the probabilities that the quantization levels at time zero occur.

π_i^m : Maximum of the transition probabilities from the quantization levels at time $i-1$ to the quantization levels at time *i* (where *i = 1, 2, ..., L*).

$\pi_0^{\min}$: Minimum of the probabilities that the quantization levels at time zero occur.

$\pi_i^{\min}$: Minimum of the transition probabilities from the quantization levels at time $i-1$ to the quantization levels at time i (where $i = 1, 2, ..., L$).

$\tilde{x}_L^m \triangleq \{x_q^m(0), x_q^m(1), ..., x_q^m(L)\}$ Sequence of the quantization levels (nodes) which the m^{th} path passes through; obviously, $x_q{}^m(i) \in \tilde{x}(i)$, where $i = 0, 1, 2, ..., L$

$z^L = \{z(1), z(2), ..., z(L)\}$ Measurement sequence from time 1 to time L.

$I^L \triangleq \{I(1), I(2), ..., I(L)\}$ Interference sequence from time 1 to time L.

Obviously, the target motion occurs along one of the possible paths through the trellis diagram. Hence, the aim is to decide upon a path through the trellis diagram which is most likely (probably) followed by the target by using the measurement sequence z^L. Because of randomness in the models, the approach must be statistical, i.e., a statistical optimization problem. Based on the measurements, the path, which was (most likely) followed by the target, will be guessed. Therefore, a criterion is needed. A suitable criterion may be the *minimum error probability criterion*, which is a special case of Bayes' criterion in detection theory. Using this criterion reduces the problem of finding the path most likely followed by the real state variable to a multiple-hypothesis-testing problem.

2.6 Minimum Error Probability Criterion [1]

In the previous section, M possible paths through the trellis diagram $H_1, H_2, ..., H_M$ were labeled. These paths are sometimes referred to as hypotheses. Hence, using the minimum error probability criterion and the

measurement sequence, true hypothesis will be decided. In other words, the path most likely followed by the target will be found. A decision rule is developed which assigns each point in the measurement space *D* to one of the hypotheses. The decision rule divides the whole measurement space *D* into *M* subspaces $D_1, D_2, ..., D_M$. If the measurements fall in the subspace D_i, H_i is decided as the true path. Subspace D_i is called the decision region for hypothesis H_i. Therefore, the decision regions must be chosen in such a way that the overall probability is minimized.

The overall error probability, sometimes called the Bayes risk *R*, is defined by

$$R \triangleq \sum_{j=1}^{M} \sum_{\substack{i=1 \\ i \neq j}}^{M} \left\{ \int_{z^L \in D_i} p(H_j) p'(z^L \mid H_j) dz^L \right\} \tag{2.10}$$

where

$$p'(z^L \mid H_j) = \begin{cases} p(z^L \mid H_j), & \text{in clear environment} \\ \int_{I^L} p(z^L \mid H_j, I^L)\, p(I^L)\, dI^L, & \text{in the presence of interference} \end{cases} \tag{2.11}$$

$p(H_j)$: Probability that the hypothesis H_j (path H_j) is true.

$p(z^L \mid H_j)$: Conditional probability of the measurement sequence z^L in clear environments given that hypothesis H_j is true.

$p(z^L \mid H_j, I^L)$: Conditional probability of the measurement sequence z^L in the presence of interference given hypothesis H_j and the interference sequence I^L.

$p(I^L)$: Joint density function of the interference sequence I^L.

In order to find the optimal decision rule, the decision regions $D_1, D_2, ..., D_M$ are varied so that the risk *R* is minimized. The optimum decision rule is:

$$\text{choose } H_i \text{ if } p(H_i)p'(z^L|H_i) > p(H_j)p'(z^L|H_j) \text{ for all } j \neq i, \qquad (2.12)$$

2.7 Optimum Decision Rule for the Target Paths [1]

Consider the motion model in equation (2.7) and the measurement model in equation (2.9). Since the disturbance noise vector *w(k)* is assumed to be independent of *w(j)* and *x(0)* for all $j \neq k$, the a priori probability of hypothesis H_i can be rewritten as:

$$p(H_i) = \prod_{k=0}^{L} \pi_k^i \qquad (2.13)$$

where $\pi_k^i = prob(x_q(k) = x_q^i(k) | x_q(k-1) = x_q^i(k-1))$, and $x_q^i(k\text{-}1)$ and $x_q^i(k)$ are the quantization levels for the gates in which the target lies at time *k-1* and *k* respectively when it follows path H_i.

Further, using the assumption that interference vector *I(k)* is independent of *I(j)* for all $j \neq k$, the joint density function of the interference sequence I^L as

$$p(I^L) = \prod_{k=1}^{L} p(I(k)) \qquad (2.14)$$

where *p(I(k))* is the probability density function of the interference vector *I(k)*.

The function $p'(z^L \mid H_i)$ in equation (2.12) can be rewritten as:

$$p'(z^L \mid H_i) = \prod_{k=1}^{L} p'(z(k) \mid x_q^i(k)) \tag{2.15}$$

where

$$p'(z(k) \mid x_q^i(k)) = \begin{cases} p(z(k) \mid x_q^i(k)) \text{ , } \textit{in clear environments} \\ \int_{I(k)} p\big(z(k) \mid x_q^i(k), I(k)\big) \times p(I(k))\, dI(k) \text{ , } \textit{in presence of interference} \end{cases} \tag{2.16}$$

and $p(z(k) \mid x_q^i(k))$ is the conditional probability of the measurement $z(k)$ in clear environments in Eq.(2.9) given that $x_q(k)=x_q^i(k)$, and $p(z(k) \mid x_q^i(k), I(k))$ is the conditional probability of the measurement $z(k)$ in the presence of interference in Eq. (2.9) given that $x_q(k)=x_q^i(k)$ and $I(k)$.

Throughout this chapter, the interference vector $I(k)$ is approximated by a discrete random vector $I_d(k)$ whose possible values are $I_{d1}(k), I_{d2}(k), \ldots, I_{dr_k}(k)$ with corresponding probabilities $p(I_{d1}(k)), p(I_{d2}(k)), \ldots, p(I_{dr_k})$, then the integral in Eq. (2.17) is reduced to a summation:

$$\int_{I(k)} p(z(k) \mid x_q^i(k), I(k)) \times p(I(k))\, dI(k) \tag{2.17}$$

$$\approx \sum_{l=1}^{r_k} p(z(k) \mid x_q^i(k), I_{dl}(k)) \times p(I_{dl}(k)) \tag{2.18}$$

where r_k is the number of possible values of the approximating discrete vector $I_d(k)$. Measurement model in the presence of interference in Eq. (2.9) becomes:

$$z(k) = g(k, x_q(k), I(k) = I_d(k), v(k))$$
$$\triangleq g(k, x_q(k), I_d(k), v(k) \tag{2.19}$$

Substituting equation (2.13) and (2.15) into the optimum decision rule of equation (2.12), the following is obtained:

Choose H_i if

$$\pi_0{}^i \prod_{k=1}^{L} \pi_k^i p'(z(k) \,|\, x_q^i(k)) > \pi_0^j \prod_{k=1}^{L} \pi_k^j p'(z(k) \,|\, x_q^j(k)) \quad \textit{for all } j \neq i. \tag{2.20}$$

Since it is more convenient to perform summations than multiplications, and the natural logarithm function is monotonically increasing, taking the natural logarithms of both sides of the inequality in equation (2.20), the following is obtained:

Choose H_i if

$$\ln(\pi_0^i) + \sum_{k=1}^{L} \left\{ \ln(\pi_k^i) + \ln(p'(z(k) \,|\, x_q^i(k))) \right\} > \ln(\pi_0^j) + \sum_{k=1}^{L} \left\{ \ln(\pi_k^j) + \ln(p'(z(k) \,|\, x_q^j(k))) \right\}$$

for all $j \neq i$, (2.21)

where in clear environments.

$$p'(z(k) \,|\, x_q^i(k)) = p(z(k) \,|\, x_q^i(k)) \tag{2.22a}$$

in the presence of interference

$$p'(z(k)|x_q^i(k)) = \begin{cases} \int p(z(k)|x_q^i(k), I(k)) p(I(k)) dI(k), & \text{using Eq.(2.9)} \\ \sum_{l=1}^{r_k} p(z(k)|x_q^i(k), I_{dl}(k)\, p(I_{dl}(k)), & \text{using Eq.(2.10)} \end{cases}$$

(2.22b)

where

$p(z(k)|x_q^i(k), I_{dl}(k)) = p(z(k)|x_q(k) = x_q^i(k), I_d(k) = I_{dl}(k))$, which is the conditional probability of $z(k)$ in Eq. (2.19) given that $x_q(k) = x_q^i(k)$ and $I_d(k) = I_{dl}(k)$. Either one of the expressions in Eqs. (2.20) and (2.21) with the convention in Section (2.6) is the *"Optimum Decision Rule"* for deciding the path most probably followed by the target.

There are some definitions, which explain the metrics to be used in this Chapter.

Definition 2.7: An initial node is a quantization level at time zero. The metric denoted by $MN(x_{qi}(0))$, of the initial node $x_{qi}(0)$ is defined by

$$\text{MN}(x_{qi}(0)) = ln\,[prob(x_q(0) = x_{qi}(0)] \tag{2.23}$$

Consequently, $MN(x_q^m(0)) = ln\,(\pi_0^m)$.

Definition 2.8: The metric, denoted by $M(x_{qj}(k-1) \rightarrow x_{qi}(k))$, of the branch which connects the quantization level $x_{qj}(k-1)$ to the quantization level $x_{qi}(k)$ is defined by:

$$M(x_{qj}(k-1) \rightarrow x_{qi}(k)) \triangleq ln[prob(x_q(k)=x_{qi}(k)|x_q(k-1)=x_{qj}(k-1))] + ln\, p'(z(k)|x_{qi}(k)) \tag{2.24}$$

Definition 2.9: The metric of a path from time zero to time i is the summation of the metric of the initial node from which the path starts and the metrics of the branches of which the path consists. For example, the metric, denoted by $M(x_q^m(i))$, of the portion between the nodes $x_q^m(0)$ and $x_q^m(i)$ of the path(hypothesis) H_m is:

$$M\left(x_q^m(i)\right) = In(\pi_0^m) + \sum_{k=1}^{i}\left[In(\pi_k^m) + In\left(p'\left(z(k) \mid x_q^m(k)\right)\right)\right] \tag{2.25}$$

Consequently, the metric, sometimes denoted by $M(H_m)$, of the path H_m (through the trellis) is:

$$M(H_m) \triangleq M\left(x_q^m(L)\right) = In\left[p(H_m)\, p'(z_L \mid H_m)\right] \tag{2.26}$$

where $x_q^m(L)$ *is* the end node of the path H_m , and $p(H_m)$ *and* $p'(z_L \mid H_m)$ are given by Eqs. (2.13), (2.15), (2.22a) and (2.22b) respectively.

Definition 2.10: The error probability of a path, say H_M, through a trellis diagram T with M possible paths $H_1, H_2, ..., H_M$ is the probability of deciding that a path which is different from H_M is the one most probably followed the path H_M .This error probability is denoted by either $P_{E_m}\left(H_1, H_2, ..., H_M\right)$ or $P_{E_m}\left(T\right)$ where subscripts E and m are the error and the m^{th} path, respectively.

Definition 2.11: The density function of the measurement sequence z^L when the state variable actually followed the path H_m is referred to as the likelihood function for the path (hypothesis) H_m .

The optimum decision rule is to choose the path with the largest metric through the trellis diagram as the decision. This can be handled by using the Viterbi decoding algorithm, which is the optimum decoding based smoothing algorithm. The algorithm which obtains a trellis diagram for the target motion model, and which finds the path most likely followed by the target by using the Viterbi decoding algorithm is referred as the **"***Optimum Decoding Based Smoothing Algorithm"*.

2.8 Optimum Decoding Based Smoothing Algorithm [1]

This algorithm, as mentioned in the previous section, finds the most probable path by comparing the metric values of the quantization values of the states from time *0* to time *L*. The implementation steps for the optimum decoding based smoothing algorithm are as below:

Preliminary Step: The target motion model is reduced to a finite state model and the trellis diagram is obtained from time *0* to time *L* until which the target will be tracked. The nodes of initial states are obtained from quantizing the initial state vector *x(0)* as explained in 2.3 and the metric of each initial node is assigned. Then, the quantized values of the disturbance noise *w(k)* are obtained in the same way as the initial state vector *x(0)*.

Step 1: For each node at time 1, using the measurement *z(1)*, the metrics of the branches connecting the initial nodes to the node at time 1 are evaluated. These metrics are added to the metrics of the initial nodes from which the branches start, and the metrics of the paths merging at the node at time 1 are found. The path with the largest metric (which is called the best path for the node at time 1) is labeled and the other paths are discarded. Finally, the largest metric to the node at time 1 (which is called the metric of the node at

time 1) is assigned. For each node, the largest metric is calculated and assigned its node.

Step k: For each node at time *k*, using the measurement *z(k)*, the metrics of the branches connecting the nodes at time *k-1* to the node at time *k* are calculated. These metrics are added to the metrics of the nodes at time *k-1* from which the branches start and the metrics of the paths merging at the node at time *k* are found. The path with the largest metric (which is called the best path for the node at time *k*) is labeled, and then the other paths are discarded. Finally, the largest metric to the node at time *k* (which is called the metric of the node at time *k*) is assigned.

At the end of time *L*, the node with the largest metric is chosen among the nodes at time *L*. The best path for this node is decided as the most probable path followed by the state transitions.

2.8.1 Reducing the Computational Time of the ODSA

In this thesis, a Matlab program is written for the ODSA algorithm, which estimates the best path with the largest metric value. The program gives opportunity to the user to modify the algorithm parameters such as the gate size, the number of quantization values of *x(0)* and *w(k)* and the maximum number of states at each time step. These parameters are directly related with the algorithm performance. However, these values also determine the computational time.

For example, smaller gate size value means higher precision, but it also means longer computation time. Moreover, increasing the number of quantization values of the initial state *x(0)* or the distribution noise *w(k)* improves the performance, but results in a slower program. For high *k* values, the number of states at each time step augments dramatically which results in a more and more complex program.

By discarding the quantization steps with lower metrics at each time step, the computational burden can be reduced without degrading the performance algorithm. This can be achieved by limiting the maximum number of states and preserving only the most probable states, which have the highest metric values.

Simulation results are given in chapter 4.

2.9 An Example of the ODSA Algorithm [1]

Figure 3 shows a target motion from time zero to time *2*. Using the ODSA, the path in the trellis diagram, which was most likely followed by the target from time zero to time *2*, will be found.

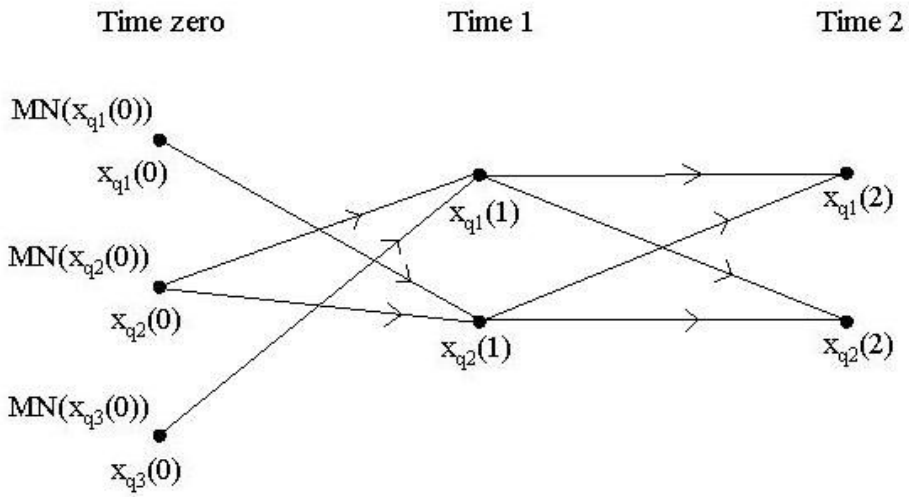

Figure-3: Trellis diagram for the target motion from time zero to time *2*

Preliminary Step: To each initial node, assign its metric, i.e., $MN(x_{qi}(0))=Prob\{x_q(0)= x_{qi}(0)\}$, where $i=1, 2, 3$. From now on, the metric of the node $x_{qi}(k)$ is represented by $MN(x_{qi}(k))$.

Step 1: Consider the node $x_{q1}(1)$. The branches $x_{q2}(0)x_{q1}(1)$ and $x_{q3}(0)x_{q1}(1)$ are the only ones connecting the nodes at time zero to the node $x_{q1}(1)$. Hence calculate the metrics of these branches, then add these metrics to the metrics of the nodes $x_{q2}(0)$ and $x_{q3}(0)$ and obtain the following:

$$A_{11} \triangleq \mathrm{M}(x_{q2}(0) \rightarrow x_{q1}(1)) + \mathrm{MN}(x_{q2}(0)) \tag{2.27}$$

$$A_{12} \triangleq \mathrm{M}(x_{q3}(0) \rightarrow x_{q1}(1)) + \mathrm{MN}(x_{q3}(0)). \tag{2.28}$$

Further, assuming that $A_{11} \geq A_{12}$, the path $x_{q2}(0)x_{q1}(1)$ is chosen as the best path for the node $x_{q1}(1)$, and A_{11} is assigned to the node $x_{q1}(1)$ as its metric, i.e., $MN(x_{q1}(1)) = A_{11}$. The path $x_{q3}(0)x_{q1}(1)$ is then discarded. Assuming that the following are similarly found for the node $x_{q2}(1)$, $x_{q1}(0)x_{q2}(1)$ is the best path for $x_{q2}(1)$, and $MN(x_{q2}(1)) = M(x_{q1}(0) \rightarrow x_{q2}(1)) + MN(x_{q1}(0))$. Hence, at the end of *0*, Figure-4 is obtained.

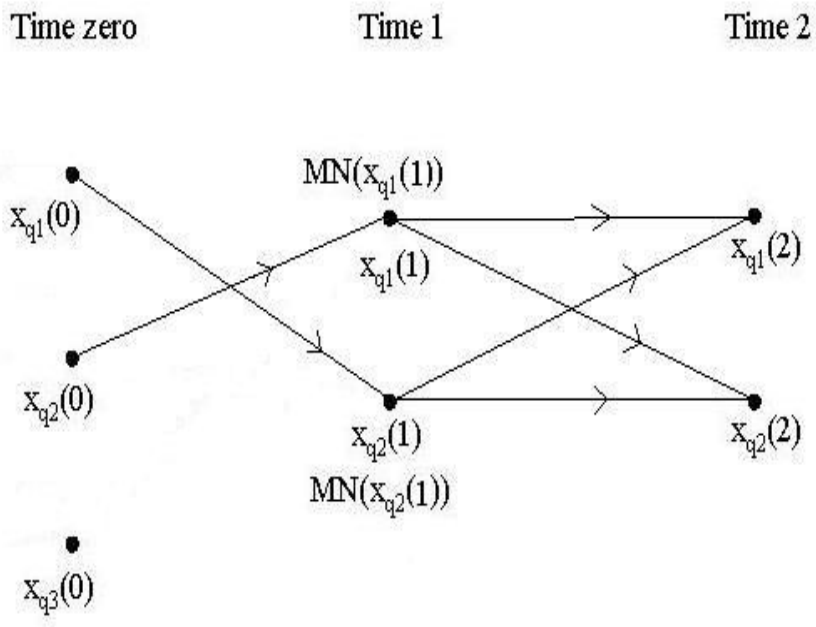

Figure-4: Trellis diagram for the target motion from time zero to time 2 at the end of first step

Step 2: Consider the node $x_{q1}(2)$. The branches $x_{q1}(1)x_{q1}(2)$ and $x_{q2}(1)x_{q1}(2)$ are those connecting the nodes at time 1 to the node $x_{q1}(2)$. Hence, calculating the metrics of these branches and adding these metrics to the metrics of the nodes $x_{q1}(1)$ and $x_{q2}(1)$, we obtain the following:

$$A_{21} \triangleq \mathrm{M}(x_{q1}(1) \rightarrow x_{q1}(2)) + \mathrm{MN}(\mathrm{xq1}(1)) \tag{2.29}$$

$$A_{22} \triangleq \mathrm{M}(x_{q2}(1) \rightarrow x_{q1}(2)) + \mathrm{MN}(x_{q2}(1)) \tag{2.30}$$

Further, assuming that $A_{22} \geq A_{21}$, the path $x_{q1}(0)x_{q2}(1)x_{q1}(2)$ is chosen as the best path for the node $x_{q1}(2)$, and A_{22} is assigned to the node $x_{q1}(2)$ as its metric, i.e., $MN(x_{q1}(2)) = A_{22}$. The path $x_{q2}(0)x_{q1}(1)x_{q1}(2)$ is then discarded. The

following are similarly found for the node $x_{q2}(2)$, then $x_{q2}(0)x_{q1}(1)x_{q2}(2)$ is the best path for $x_{q2}(2)$, and $MN(x_{q2}(2)) = M(x_{q1}(1) \rightarrow x_{q2}(2)) + MN(x_{q1}(1))$. Hence, Figure 5 is obtained at the end of 0. In addition, assuming that $MN(x_{q2}(2)) \geq MN(x_{q1}(2))$, the path $x_{q2}(0)x_{q1}(1)x_{q2}(2)$ is chosen as the path followed by the target from time zero to time *2*.

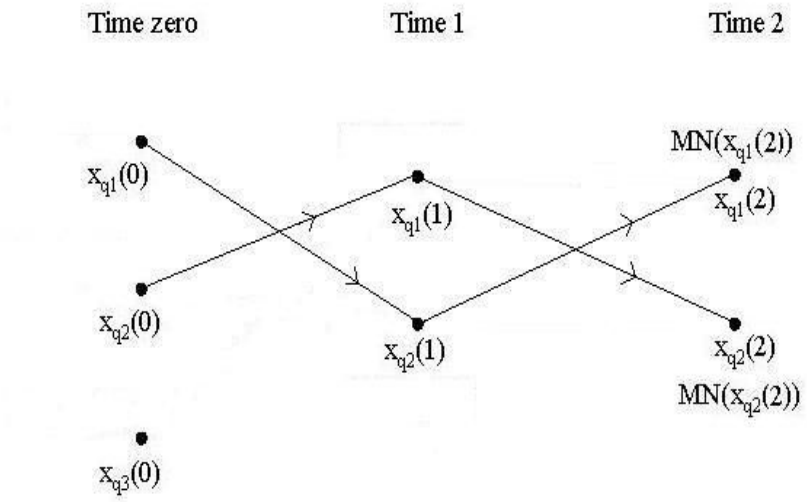

Figure-5: Trellis diagram for the target motion from time zero to time *2* at the end of second step

CHAPTER 3

ODSA WITH CORRELATED MEASURMENT NOISE

ODSA, explained in Chapter 2, can be implemented for target tracking systems which have independent Gaussian measurement noise models. In this chapter, ODSA measurement model is modified so that correlated measurement noise effect is handled. ODSA is implemented using two methods, namely *"Method of Decorrelation [3, 4]"* and *"Treating Correlation Effect as Interference",* which is proposed in this thesis. They will be used to adapt ODSA in the presence of correlated measurement noise.

3.1 Implementing ODSA Using Method of Decorrelation

3.1.1 Models and Assumptions

In Chapter 2, target motion and measurement models for ODSA (with *u(k)=0*) were defined as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = f(k, x(k), w(k)) \\ &\textit{Measurement model,} && z(k) = g(k, x(k), v(k)) \end{aligned} \tag{3.1}$$

in clear environment.

The measurement noise *v(k)* given in the measurement modelof Eq. (3.1) was defined as independent Gaussian noise in section 2.1.

In this section, the target tracking system with correlated measurement noise will be studied. The correlated measurement noise *v(k)* with zero-mean can be modelled by a first-order Markov process [3, 4]:

$$v(k) = av(k-1) + r(k) \tag{3.2}$$

where a is the correlation coefficient ($0 \le a \le 1$), and $r(k)$ is a zero-mean white Gaussian noise, with variance:

$$\sigma_r^2 = (1-a^2)\sigma_v^2 \tag{3.3}$$

where σ_v^2 is the variance of the measurement noise $v(k)$.

It can be easily proved why a is the correlation coefficient of the random variables *v(k)* and *v(k-1)* as below:

Proof for Eq.(3.3):

It is known that correlation coefficient, ρ, of the random variables x and y is defined by the ratio:

$$\rho = \frac{C_{xy}}{\sigma_x \sigma_y} \tag{3.4}$$

where $C_{xy} = E\{(x-\bar{x})(y-\bar{y})\}$ *is the covariance of the random variables x and y,* σ_x *is the standart deviation of x, and* σ_y *is the standart deviation of* y [12].

From Eq.(3.5), the correlation coefficient of random variables v(k) and v(k-1) is defined by the ratio:

$$\rho = \frac{C_{v(k)v(k-1)}}{\sigma_{v(k)}\sigma_{v(k-1)}} \tag{3.5}$$

In order to find the correlation coefficient of random variables v(k) and v(k-1), ρ , let us find firstly variance of v(k) and variance of v(k-1):

$$\sigma^2_{v(k)} = E\left[\left(v(k) - \bar{v}(k)\right)^2\right] \tag{3.6}$$

$\bar{v}(k) = E[v(k)] = 0$ *is given, then Eq.(3.7) yields:*

$$\sigma^2_{v(k)} = E\left[v^2(k)\right] \tag{3.7}$$

If v(k) given in Eq.(3.3) is inserted in Eq.(3.8):

$$\begin{aligned}\sigma^2_{v(k)} &= E\left[a^2v^2(k-1) + 2av(k-1)r(k) + r^2(k)\right] \\ &= a^2E\left[v^2(k-1)\right] + 2aE\left[v(k-1)r(k)\right] + E\left[r^2(k)\right]\end{aligned} \tag{3.8}$$

Since r(k) and v(k-1) are uncorrelated and have zero-mean values $E\{v(k-1)r(k)\} = 0$*, then the following equation is obtained:*

$$\sigma^2_{v(k)} = a^2\sigma^2_{v(k-1)} + \sigma^2_{r(k)} \tag{3.9}$$

Because v(k) is stationary $\sigma^2_{v(k)} = \sigma^2_{v(k-1)}$ [12], *then* $\sigma^2_{v(k)}$ *is found as:*

$$\sigma_{v(k)}^2 = \frac{\sigma_{r(k)}^2}{1-a^2} \tag{3.10}$$

The covariance $C_{v(k)v(k-1)}$ *of random variables v(k) and v(k-1) is:*

$$\begin{aligned} C_{v(k)v(k-1)} &= E\left[\left(v(k)-\bar{v}(k)\right)\left(v(k-1)-\bar{v}(k-1)\right)\right] \\ &= E\left[v(k)v(k-1)\right] \\ &= E\left[\left(av(k-1)+r(k)\right)v(k-1)\right] \\ &= aE\left[v^2(k-1)\right]+E\left[r(k)v(k-1)\right] \end{aligned} \tag{3.11}$$

Since r(k) and v(k-1) are uncorrelated and have zero-mean values $E\left[r(k)v(k-1)\right]=0$,

$$C_{v(k)v(k-1)} = a\,E\left[v^2(k-1)\right] = a\sigma_{v(k)}^2 \tag{3.12}$$

$$\rho = \frac{C_{v(k)v(k-1)}}{\sigma_{v(k)}\sigma_{v(k-1)}} = \frac{a\sigma_{v(k)}^2}{\sigma_{v(k)}^2} \tag{3.13}$$

$$\rho = a \tag{3.14}$$

Eq.(3.15) proves that a is the correlation coefficient random variables v(k) and v(k-1).

Proof of Eq.(3.4) is also given below to increase the comprehension:

Proof of Eq.(3.4):

$$\sigma_r^2 = E\left[\left(r(k)-\bar{r}(k)\right)\right] \tag{3.15}$$

$\bar{r}(k) = E[r(k)] = 0$ *is given, then Eq.(3.15) yields:*

$$\sigma_r^2 = E\left[r^2(k)\right] \tag{3.16}$$

If r(k) given in Eq.(3.3) is inserted in Eq.(3.17):

$$\begin{aligned}
\sigma_r^2 &= E\left[\left(v(k) - a\,v(k-1)\right)^2\right] \\
&= E\left[v^2(k) - 2\,a\,v(k)v(k-1) + a^2 v^2(k-1)\right] \\
&= E\left[v^2(k)\right] - 2\,a\,E\left[v(k)\,v(k-1)\right] + a^2\,E\left[v^2(k-1)\right] \\
&= E\left[v^2(k)\right] - 2aE\left[\left(a\,v(k-1) + r(k)\right)v(k-1)\right] + a^2 E\left[v^2(k-1)\right] \\
&= E\left[v^2(k)\right] - 2a^2 E\left[v^2(k-1)\right] - 2aE\left[r(k)v(k-1)\right] + a^2 E\left(v^2(k-1)\right)
\end{aligned} \tag{3.17}$$

Since r(k) and v(k-1) are uncorrelated and have zero-mean values, $E\left[r(k)v(k-1)\right] = 0$*, then Eq.(3.18) becomes:*

$$\sigma_r^2 = E\left[v^2(k)\right] - a^2 E\left[v^2(k-1)\right] \tag{3.18}$$

Since v(k) is a stationary process, $E\left[v^2(k)\right] = E\left[v^2(k-1)\right] = \sigma_v^2$ [12] *with* σ_v^2 *is the variance of the measurement noise v(k):*

$$\sigma_r^2 = (1 - a^2)\sigma_v^2 \tag{3.19}$$

and Eq.(3.4) results.

3.1.2 Decorrelation Process

In this section, target motion and measurement models given in Eq. (3.1) will be taken as linear models as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = x(k) + w(k) \\ &\textit{Measurement model,} && z(k) = x(k) + v(k) \end{aligned} \tag{3.20}$$

Deccorelation process [3, 4] will be described. To decorrelate the measurement noise, a new measurement $y(k)$, called *"artificial measurement"*, is generated by using the measurement given in Eq. (3.20) as below:

$$\begin{aligned} y(k) = z(k) - a\ z(k-1) &= x(k) + v(k) - a\left[x(k-1) + v(k-1)\right] \\ y(k) &= x(k) - a\,x(k-1) + \underbrace{v(k) - a\,v(k-1)}_{r(k)} \\ y(k) &= \left[x(k) - a\,x(k-1)\right] + r(k) \end{aligned} \tag{3.21}$$

for which the measurement errors are uncorrelated.

From the motion model given in Eq.(3.20), *x(k-1)* can be calculated as below:

$$x(k-1) = x(k) - w(k-1) \tag{3.22}$$

If we insert *x(k-1)* given in Eq.(3.22) into Eq.(3.21), *y(k)* yields:

$$\begin{aligned} y(k) &= x(k) - a\left[x(k) - w(k-1)\right] + r(k) \\ &= (1-a)\,x(k) + \underbrace{a\,w(k-1) + r(k)}_{\eta(k)} \end{aligned} \tag{3.23}$$

In practical applications, the first term of right-hand side in Eq.(3.23) is usually small and can be neglected without degrading much performance [7]. So we have $\eta(k) \approx r(k)$

Then, Eq.(3.23) takes the form:

$$y(k) = (1-a)\,x(k) + r(k) \tag{3.24}$$

After decorrelation process, target motion and measurement models will become as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = x(k) + w(k) \\ &\textit{Measurement model,} && y(k) = (1-a)\,x(k) + v(k) \end{aligned} \tag{3.25}$$

Having obtained the target motion and measurement models, ODSA explained in Chapter 2, can be applied easily.

Decorrelation method can be applied for linear models. In section 3.2, we will propose the method of treating correlation effect as interference; this method can also be applied for nonlinear models.

3.2 Implementing ODSA by Treating Correlation Effect as Interference

3.2.1 Models and Assumptions

In Chapter 2, target motion and measurement models for ODSA (with *u(k)=0*) were defined as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = f(k, x(k), w(k)) \\ &\textit{Measurement model,} && z(k) = g(k, x(k), v(k)) \end{aligned} \tag{3.1}$$

in clear environment. In the presence of interference, the interference parameter is added to the measurement model as below:

$$\begin{aligned} &\textit{Motion model,} && x(k+1) = f(k, x(k), w(k)) \\ &\textit{Measurement model,} && z(k) = g(k, x(k), I(k), v(k)) \end{aligned} \tag{3.2}$$

The measurement noise *v(k)* given in the measurement models of Eqs. (3.1) and (3.2) was defined as independent Gaussian noise in section 2.1.

In this section, the target tracking system with correlated measurement noise will be studied. The correlated measurement noise *v(k)* with zero-mean can be modelled by a first-order Markov process [3, 4]:

$$v(k) = av(k-1) + r(k) \tag{3.3}$$

where a is the correlation coefficient ($0 \le a \le 1$), and $r(k)$ is a zero-mean white Gaussian noise, with variance:

$$\sigma_r^2 = (1 - a^2)\sigma_v^2 \tag{3.4}$$

where σ_v^2 is the variance of the measurement noise $v(k)$.

If *v(k)* is inserted into the measurement model in clear environments defined in Eq.(3.1), the following equation is obtained:

$$\textit{measurement model,} \quad z(k) = g\big(k, x(k), av(k-1) + r(k)\big) \tag{3.21}$$

In this thesis, we propose to treat *v(k-1)* parameter as an interference parameter so that ODSA can be applied in the presence of correlated

measurement noise. Then the measurement model given by Eq. (3.21) can be written as the form:

$$z(k) = g(k, x(k), v(k-1), r(k)) \tag{3.22}$$

Eq. (3.22) resembles the measurement model in the presence of interference, which is given in Eq. (3.2). The only modifications are the following replacements: $I(k) \Rightarrow v(k\text{-}1)$ and $v(k) \Rightarrow r(k)$.

Note: *It is known from Chapter 2 that interfence vectors should be independent from each other so that ODSA can be applied in presence of interference. We will assume, v(k-1) parameter can be treated as interference parameter for weakly correlated measurement noise models, so that ODSA can be applied. We will also investigate how the proposed method behaves for higly correlated measurement noise models in the simulations given in the following chapters.*

The values that $z(k)$ take can be calculated for $k=1,2,...,L$ as below:

$$\begin{aligned}
&k = 1 \Rightarrow z(1) = g(1, x(1), \underbrace{v(0)}_{I(0)}, r(1)) \\
&k = 2 \Rightarrow z(2) = g(2, x(2), \underbrace{v(1)}_{I(1)}, r(2)) \\
&\vdots \qquad\qquad \vdots \\
&k = L \Rightarrow z(L) = g(L, x(L), \underbrace{v(L-1)}_{I(L-1)}, r(L))
\end{aligned} \tag{3.23}$$

The system in the presence of correlated measurement noise will become as below:

$$\text{Motion model,} \quad x(k+1) = f\big(k, x(k), w(k)\big) \tag{3.24}$$
$$\text{Measurement model,} \quad z(k) = g\big(k, x(k), v(k-1), r(k)\big)$$

Parameters used in equation (3.24) are defined as follows:

- *x(0)* is an *nx1* initial state Gaussian distributed random vector (which determines the considered target location at time *0*),
- *x(k)* is an *nx1* (target) state vector at time *k*,
- *w(k)* is a *px1* Gaussian distributed disturbance noise vector at time *k* with zero mean and known statistics,
- *z(k)* is an *rx1* measurement vector at time *k,*
- *v(0)* is an *mx1* initial measurement noise vector (accepted as interference vector) with zero mean (which determines the considered measurement noise at time *0*),
- *v(k)* is an *mx1* measurement noise vector (accepted as interference vector) at time *k* with zero mean and known statistics,
- *r(k)* is a *lx1* white Gaussian distributed measurement noise vector at time *k* with zero mean and known statistics

Furthermore, time *k* is time $t_0 + kT_0$ where t_0 and T_0 are the initial time and the measurement interval respectively. The random vectors *x(0)*, *w(j)*, *w(k), v(l), r(m)* are assumed to be independent for all *j*, *k, l, m*. The goal is to estimate the state sequence {*x(0)*, *x(1)*,…, *x(L)*} by using the measurement sequence {*z(1)*, *z(2)*,…, *z(L)*}, where *L* is a chosen integer.

x(0) is approximated by a discrete random vector $x_d(0)$ with n_x possible values, where n_x is the number of possible quantization levels of *x(0)*

$w(k)$ is approximated by a discrete random vector $w_d(k)$ with n_w possible values, where n_w is the number of possible quantization levels of $w(k)$

$v(0)$ is approximated by a discrete random vector $v_d(0)$ with n_v possible values, where n_v is the number of possible quantization levels of $v(0)$

$r(k)$ is approximated by a discrete random vector $r_d(k)$ with n_r possible values, where n_r is the number of possible quantization levels of $r(k)$

The number of all values that $v_d(k)$ can take are calculated for $k=1,2,...,L-1$ as below:

$$v_d(k) = a\,v_d(k-1) + r_d(k)$$

$$k = 1 \Rightarrow\ v_d(1) = a\,\underbrace{v_d(0)}_{n_v} + \underbrace{r_d(1)}_{n_r}$$

$$\Rightarrow v_d(1) \text{ will have } n_v \times n_r \text{ values.}$$

$$k = 2 \Rightarrow\ v_d(2) = a\,\underbrace{v_d(1)}_{n_v \times n_r} + \underbrace{r_d(2)}_{n_r}$$

$$\Rightarrow v_d(2) \text{ will have } n_v \times n_r \times n_r = n_v \times n_r^{\,2} \text{ values.}$$

$$k = 3 \Rightarrow v_d(3) = a\,v_d(2) + r_d(3) \Rightarrow$$

$$\Rightarrow v_d(3) \text{ will have } n_v \times n_r \times n_r \times n_r = n_v \times n_r^{\,3} \text{ values.}$$

$$\vdots \qquad \vdots$$

$$k = L-1 \Rightarrow v_d(L-1) = a\,v_d(L-2) + r_d(L-1)$$

$\Rightarrow$ $v_d(L-1)$ will have $n_v \times n_r \times n_r \times \ldots \times n_r = n_v \times n_r^{L-1}$ values. (3.25)

3.2.2 Calculation of a Metric of a Branch

From *Definition 2.9* in section 2.7**,** the metric of a path from time zero to time *i* is the summation of the metric of the initial node from which the path starts and the metrics of the branches of which the path consists. For example, the metric, denoted by $M(x_q^m(i))$, of the portion between the nodes $x_q^m(0)$ and $x_q^m(i)$ of the path(hypothesis) H_m is:

$$M\left(x_q^m(i)\right) = In(\pi_0^m) + \sum_{k=1}^{i}\left[In(\pi_k^m) + In\left(p'\left(z(k) \mid x_q^m(k)\right)\right)\right] \qquad (3.25)$$

In correlated measurement noise case, the calculation of $p'\left(z(k) \mid x_q^m(k)\right)$ changes. The rest of the metric calculation is the same as the ODSA explained in Chapter 2. So in this section, $p'\left(z(k) \mid x_q^m(k)\right)$ calculation will be discussed.

From Eq.(2.18), it is known that:

$p'\left(z(k) \mid x_q^m(k)\right) \approx \sum_{l=1}^{r_k} p\left(z(k) \mid x_q^i(k), I_{dl}(k)\right) \times p\left(I_{dl}(k)\right)$ in the presence of interference, where r_k is the number of possible values of the approximating vector $I_d(k)$. $I_d(k)$ is the approximated form of the interference vector *I(k)* whose possible values are $I_{d1}(k)$, $I_{d2}(k)$, ..., I_{dr_k} with corresponding probabilities $p(I_{d1}(k))$, $p(I_{d2}(k))$, ...,$p(I_{dr_k}(k))$.

In correlated measurement noise case, measurement model was transformed into the model given in Eq.(3.24) and since in this equation *v(k-1)*

was accepted as interference vector, $I_d(k)$, $p'\left(z(k) \mid x_q^m(k)\right)$can be calculated as below:

$$p'\left(z(k) \mid x_q^m(k)\right) = \sum_{j=1}^{n} p\left(z(k) \mid x_q^m(k), \underbrace{v(k-1)}_{v_{dj}(k-1)} \right) \times p(v_{dj}(k-1)) \qquad (3.26)$$

where n is the number of possible values of the approximating vector $v_d(k-1)$. *v(k-1) is* approximated by a discrete random vector $v_d(k-1)$ whose possible values are v_{d1}*(k-1),* v_{d2}*(k-1) ...,* v_{dn} *(k-1)* with corresponding probabilities *p(*v_{d1}*(k-1)), p(*v_{d2}*(k-1)), ...,p(*v_{dn} *(k-1)).*

A measurement model given below will be used in the following sections in this thesis:

$$z(k) = g\left(k, x_q^m(k)\right) + a\, v_d(k-1) + r_d(k) \qquad (3.27)$$

Since *z(k)* is a Gaussian-distrubuted random process, $[\, z(k) \mid x_q^m(k), v_d(k-1)\,]$ is also a Gaussian-distrubuted random process with mean $g\left(k, x_q^m(k)\right) + a v_d(k-1)$ and variance σ_r^2.So $p\left(z(k) \mid x_q^m(k), v_d(k-1)\right)$ given in Eq.(3.26) can be computed as below:

$$p\left(z(k) \mid x_q^m(k), v_d(k-1)\right) = \frac{1}{\sqrt{2\pi\sigma_r^2}} * \exp\left[-\frac{\left(z(k) - g\left(k, x_q^m(k)\right) - a v_d(k-1)\right)^2}{2\sigma_r^2} \right] \qquad (3.28)$$

By inserting $p\left(z(k) \mid x_q^m(k), v_d(k-1)\right)$ given in Eq. (3.28) into Eq.(3.26), Eq.(3.26) takes the form:

$$p\left(z(k) \mid x_q^m(k), v_d(k-1)\right)=$$

$$\sum_{j=1}^{n}\left(\frac{1}{\sqrt{2\pi\sigma_r^2}} * \exp\left[-\frac{\left(z(k)-g\left(k, x_q^m(k)\right)-a v_{dj}(k-1)\right)^2}{2\sigma_r^2}\right] \times p(v_{dj}(k-1))\right) \qquad (3.29)$$

where *n* is the total number of values that $v_d(k-1)$ can take, $v_{dj}(k-1)$ is the j[th] value of $v_d(k-1)$ and $p(v_{dj}(k-1))$ is the j[th] probability value of $p(v_d(k-1))$

For example, for $n_v=n_r=3$, $p'\left(z(k) \mid x_q^m(k)\right)$ values are calculated at for *k=1,2,...,* *L* as below:

$$\textbf{\textit{k=1}} \Rightarrow\ z(1) = g(1, x_q^m(1)) + a\, v_d(0) + r_d(1),$$

v(0) is approximated by a discrete random vector, $v_d(0)$ whose possible values are $v_{d1}(0)$, $v_{d2}(0)$, $v_{d3}(0)$ with corresponding probabilities $p(v_{d1}(0))$, $p(v_{d2}(0))$, $p(v_{d3}(0))$. Then $p\left(z(1) \mid x_q^m(1)\right)$ can be calculated as:

$$p'\left(z(1) \mid x_q^m(1)\right)= \sum_{j=1}^{n_v}\left(p\left(z(1) \mid x_q^m(1)\right), v(0)=v_{dj}(0)\right)\times p(v_{dj}(0))$$

$$p'\left(z(1)\mid x_q^m(1)\right)=\sum_{j=1}^{3}\left(\frac{1}{\sqrt{2\pi\sigma_r^2}}*\exp\left[-\frac{\left(z(1)-g\left(x_q^m(1)\right)-av_{dj}(0)\right)^2}{2\sigma_r^2}\right]*p(v_{dj}(0))\right)$$

k=2 $\Rightarrow z(2)=g(2,x_q^m(2))+a\,v_d(1)+r_d(2)$**,**

$v_d(1)=a\underbrace{v_d(0)}_{n_V}+\underbrace{r_d(1)}_{n_r}$, $v_d(1)$ will have $n_v\times n_r=3\times 3=9$ values

v(0) is approximated by a discrete random vector, $v_d(0)$ whose possible values are $v_{d1}(0)$, $v_{d2}(0)$, $v_{d3}(0)$ with corresponding probabilities $p(v_{d1}(0))$, $p(v_{d2}(0))$,$p(v_{d3}(0))$.

r(1) is approximated by a discrete random vector, $r_d(1)$ whose possible values are $r_{d1}(1)$, $r_{d2}(1)$, $r_{d3}(1)$ with corresponding probabilities $p(r_{d1}(1))$, $p(r_{d2}(1))$, $p(r_{d3}(1))$.

$v_d(1)$ values and corresponding probability values are calculated as below:

$$v_{d1}(1)=a\,v_{d1}(0)+r_{d1}(1)\Rightarrow p(v_{d1}(1))=p(v_{d1}(0))\times p(r_{d1}(1))$$

$$v_{d2}(1)=a\,v_{d1}(0)+r_{d2}(1)\Rightarrow p(v_{d2}(1))=p(v_{d1}(0))\times p(r_{d2}(1))$$

$$v_{d3}(1)=a\,v_{d1}(0)+r_{d3}(1)\Rightarrow p(v_{d3}(1))=p(v_{d1}(0))\times p(r_{d3}(1))$$

$$v_{d4}(1)=a\,v_{d2}(0)+r_{d1}(1)\Rightarrow p(v_{d4}(1))=p(v_{d2}(0))\times p(r_{d1}(1))$$

$$v_{d5}(1)=a\,v_{d2}(0)+r_{d2}(1)\Rightarrow p(v_{d5}(1))=p(v_{d2}(0))\times p(r_{d2}(1))$$

$$v_{d6}(1)=a\,v_{d2}(0)+r_{d3}(1)\Rightarrow p(v_{d6}(1))=p(v_{d2}(0))\times p(r_{d3}(1))$$

$$v_{d7}(1) = a\,v_{d3}(0) + r_{d1}(1) \Rightarrow p(v_{d7}(1)) = p(v_{d3}(0)) \times p(r_{d1}(1))$$

$$v_{d8}(1) = a\,v_{d3}(0) + r_{d2}(1) \Rightarrow p(v_{d8}(1)) = p(v_{d3}(0)) \times p(r_{d2}(1))$$

$$v_{d9}(1) = a\,v_{d3}(0) + r_{d3}(1) \Rightarrow p(v_{d9}(1)) = p(v_{d3}(0)) \times p(r_{d3}(1))$$

Then , $p'\left(z(2) \mid x_q^m(2)\right)$ can be calculated as:

$$p'\left(z(2) \mid x_q^m(2)\right) = \sum_{j=1}^{n_v \times n_r} \left(p\left(z(2) \mid x_q^m(2), v(1) = v_{dj}(1)\right) \times p(v_{dj}(1))\right)$$

$$p'\left(z(2) \mid x_q^m(2)\right) = \sum_{j=1}^{9} \left[\frac{1}{\sqrt{2\pi\sigma^2}} * \exp\left(-\frac{\left(z(2) - g\left(x_q^m(2)\right) - a\,v_{dj}(1)\right)^2}{2\sigma^2}\right) \times p(v_{dj}(1))\right]$$

k=3 $\Rightarrow z(3) = g(3, x_q^m(3)) + av_d(2) + r_d(3)$,

$v_d(2) = a\,\underbrace{v_d(1)}_{n_v \times n_r} + \underbrace{r_d(2)}_{n_r}$ $v_d(2)$ will have $n_v \times n_r \times n_r = 3 \times 3 \times 3 = 27$ values. Then:

$$p'\left(z(3) \mid x_q^m(3)\right) = \sum_{j=1}^{27} \left(p\left(z(3) \mid x_q^m(3), v(2) = v_{dj}(2)\right) \times p(v_{dj}(2))\right)$$

⋮ ⋮

k=L $\Rightarrow z(L) = g(L, x_q^m(L)) + av_d(L-1) + r_d(L)$, $v_d(L-1)$ will have $n_v \times n_r^{L-1} = 3 \times 3^{L-1}$ values, then:

$$p'\left(z(L)\mid x_q^m(L)\right)=\sum_{j=1}^{3x3^{(L-1)}}\left(p\left(z(L)\mid x_q^m(L),v(L-1)=v_{dj}(L-1)\right)\times p(v_{dj}(L-1))\right)$$

It can be seen from the equations that at each time interval the number of $v_d(k)$ states increases by the number of quantization values, n_r, so the number of states has to be limited at each time interval to reduce the execution time of code. A state limit value for $v_d(k)$ states is determined. The program chooses the states that have better metrics when the number of the states exceeds the state limit value and cancels others.

After calculating $p'\left(z(k)\mid x_q^m(k)\right)$, the natural logarithm of it is taken and it is inserted into the Eq.(3.25) to calculate the metric of a branch.

3.3 Accuracy of the Proposed Method

In Section 3.2.1 and 3.2.2, *v(k-1)* was treated as an interference parameter and ODSA was applied in the presence of interference. We will check the accuracy of the proposed method by comparing the simulation results of the proposed method by the simulation results of method of decorrelation given in section 3.1. Since method of decorrelation can be applied for linear models, we will compare only linear models. If we get approximately close results for linear models, we will assume that the proposed algorithm can also be applied for nonlinear models.

3.3.1 Comparision of the Method of Decorrelation and Proposed Method of Treating Correlation Effect as Interference

For each method, simulations are obtained after 500 executionss.

Linear models used in ODSA simulations for both methods are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise ODSA simulations, v (k), is modelled as:

$v(k)=a\ v(k-1)+r(k)$

The following parameters are used in the ODSA simulations for both methods:

total sampling time, L = 50

correlation coefficient ,a =0.1

gate size = 0.1

number of maximum states = 100

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0 , E[r(k)] = 0

The following added parameters are used in the ODSA simulations for "Proposed Method of Treating Correlation Effect as Interference" :

number of maximum v(k) states = 50

quantization numbers : Q # of v(0) = 3, Q # of r(k)=3

expected values : E[v(0)] = 0

In Figure 6, two methods are compared as the correlation coefficient changes.

In Figure 7, two methods are compared as the gate size changes.

In Figure 8, two methods are compared as the quantization number of the initial state vector changes.

In Figure 9, two methods are compared as the quantization number of the disturbance noise vector changes.

In Figure 10, two methods are compared as the initial state variance changes.

In Figure 11, two methods are compared as the disturbance noise variance changes.

In Figure 12, two methods are compared as the measurement noise variance changes.

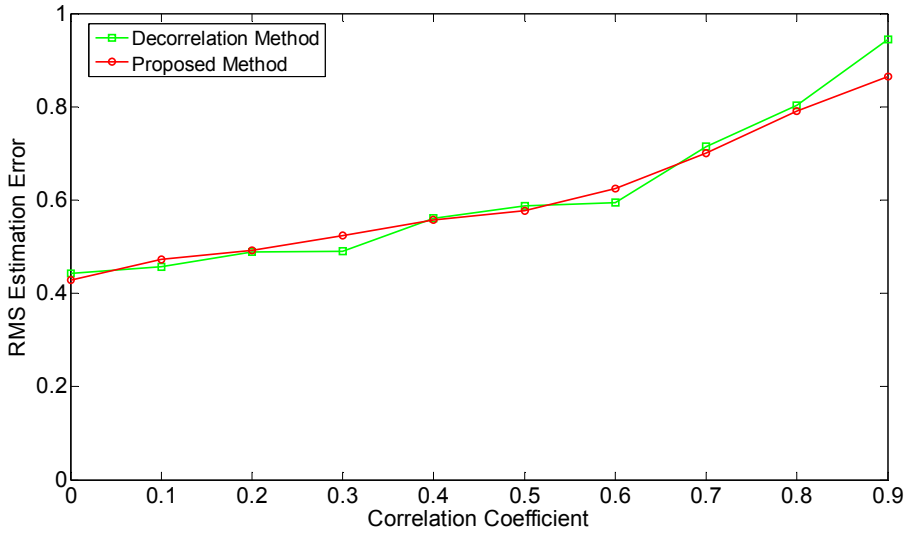

Figure-6: RMS estimation error versus correlation coefficient for ODSA using both methods

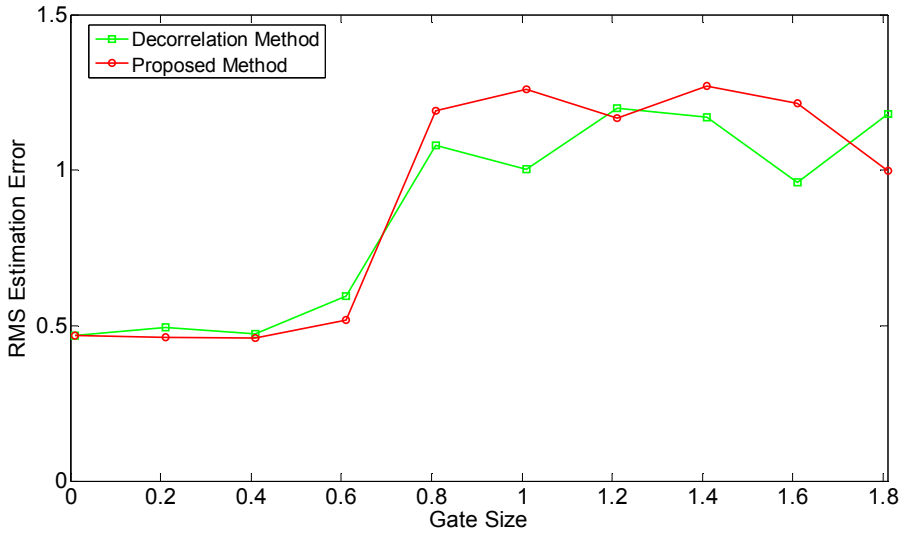

Figure-7: RMS estimation error versus gate size for ODSA using both methods

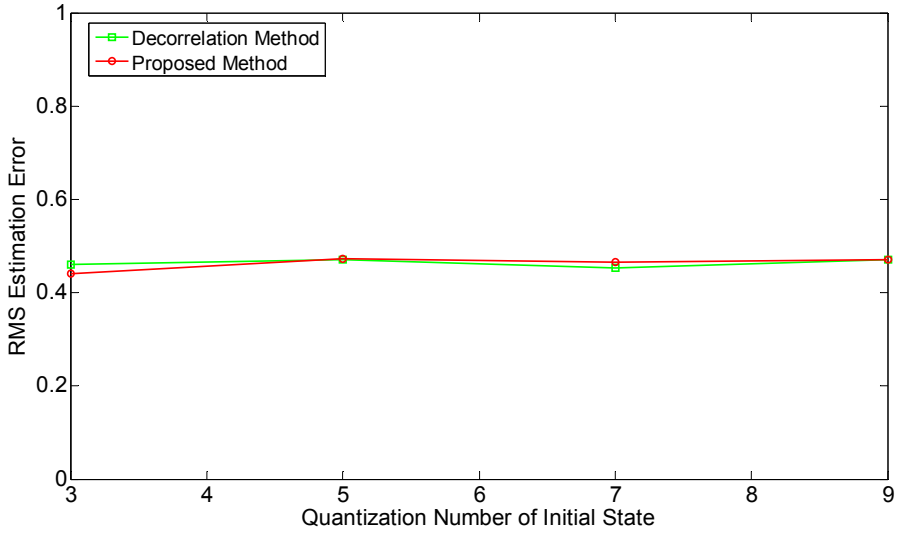

Figure-8: RMS estimation error versus quantization number of initial state vector for ODSA using both methods

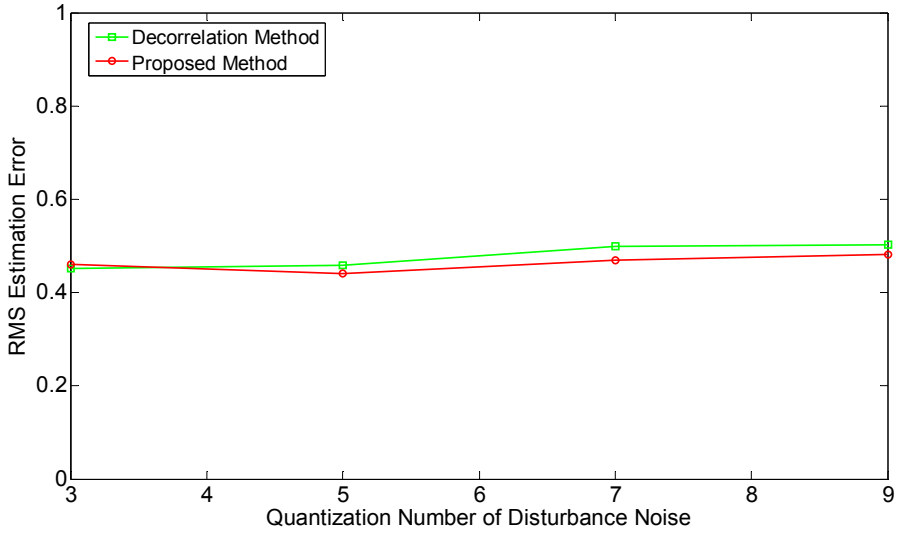

Figure-9: RMS estimation error versus quantization number of disturbance noise vector for ODSA using both methods

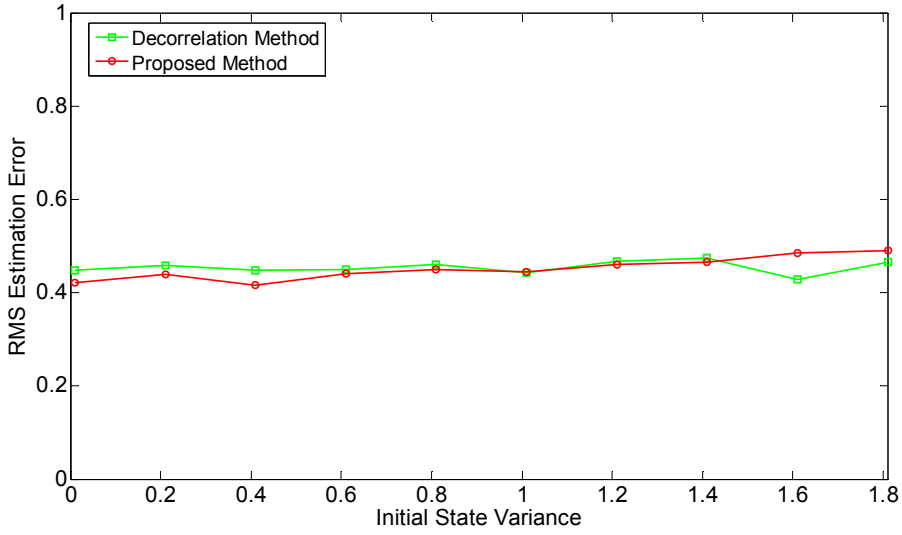

Figure-10: RMS estimation error versus initial state variance for ODSA using both methods

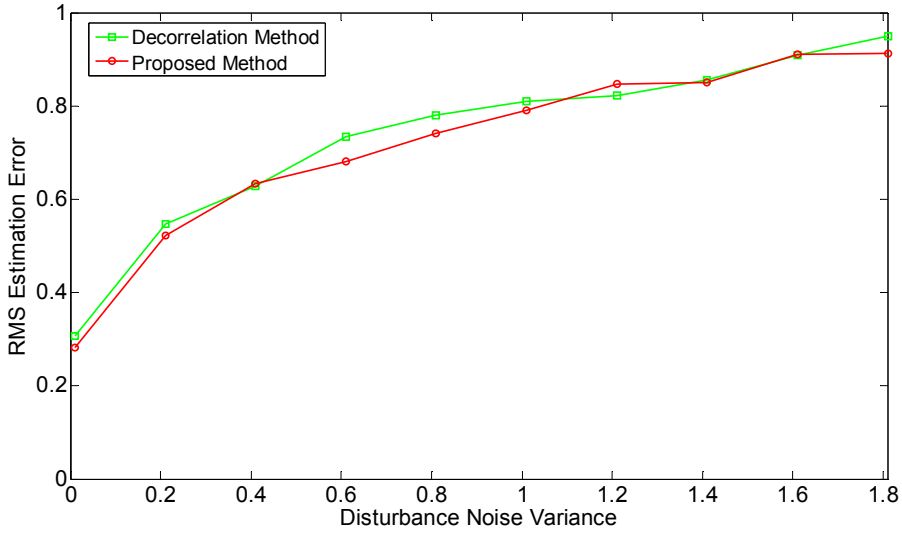

Figure-11: RMS estimation error versus disturbance noise variance for ODSA using both methods

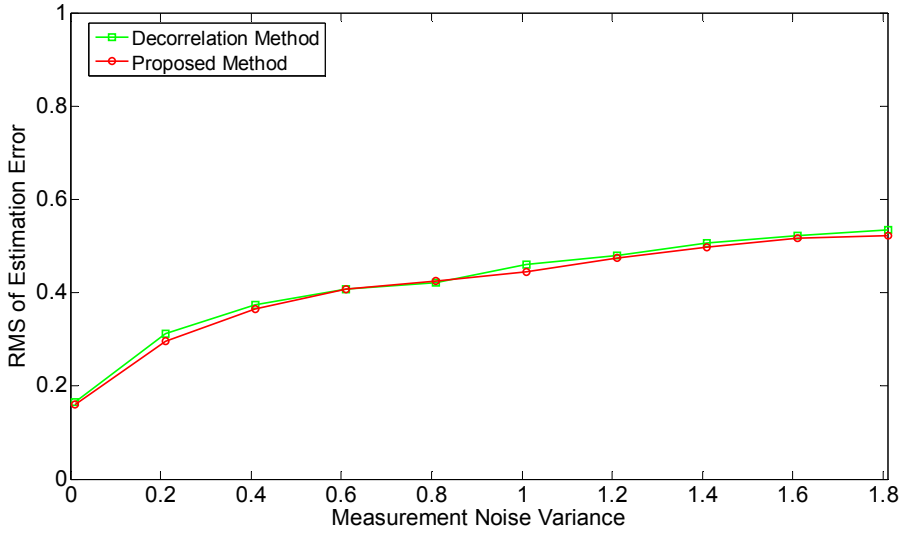

Figure-12: RMS estimation error versus measurement noise variance for ODSA using both methods

Comments:

From Figures 6-12, it can be said that as the correlation coefficient, gate size, quantization number of initial state vector, quantization number of disturbance noise vector, initial state variance, disturbance noise variance and measurement noise variance parameters change, two methods show approximately same estimation performance. Since we checked that the method we proposed can be accepted as an accurate method for linear models so we will assume that the proposed method can also be applied for nonlinear models.

CHAPTER 4

SIMULATION RESULTS OF ODSA WITH CORRELATED MEASUREMENT NOISE

In this chapter, some simulations are carried out to demonstrate the effects of the parameters, which are used in the ODSA algorithm in the presence of correlated measurement noise. These parameters are:

- correlation coefficient

(denoted by "a" in the figures)

- gate size

(denoted by "gate size" in the figures)

- quantization number of initial state vector

(denoted by "Q # of x(0)" in the figures)

- quantization number of disturbance noise vector

(denoted by "Q # of w(k)" in the figures)

- quantization number of initial measurement noise vector

(denoted by "Q # of v(0)" in the figures)

- quantization number of white gaussian noise vector component of measurement noise vector

(denoted by "Q # of r(k)" in the figures)

- initial state variance

(denoted by “var[x(0)]” in the figures)

- disturbance noise variance

(denoted by “var[w(k)]” in the figures)

- initial correlated measurement variance

(denoted by “var[v(0)]” in the figures)

- limit of the maximum state number

(denoted by “state limit” in the figures)

- limit of the maximum measurement noise state number

(denoted by “v(k) state limit” in the figures)

In order to check the performance of the algorithm, the actual target path is needed. In other words, the target and the measurement vectors must be generated. Using the *“randn(.)”* command of *Matlab*, the Gaussian distributed random vectors *x(0), w(k)*, *v(0)*, and *r(k)* are generated according to the corresponding mean and variances. These values are put into the motion and measurement models and the actual values of *x(k)* and *z(k)* values are obtained.

Simulations are performed for one linear and one nonlinear model.The simulations are obtained after 500 executionss. For each execution, the state vector *x(k)* and the measurement vector *z(k)* are regenerated with the same motion and measurement equations.

As a result of performed simulations, RMS Estimation Error versus Sampling Time graphs are given in figures to show the performance of the ODSA. RMS Estimation error for a given sampling time, k, is calculated as below:

$$\text{RMS Error(k)} = \frac{\sqrt{\sum_{i=1}^{N}(X_{ik} - \widetilde{X}_{ik})^2}}{N} \quad \text{k=0,1,…, L} \tag{4.1}$$

where RMS Error(k) is the RMS Error for sampling time k, N is the total execution number, X_{ik} is the real target state at sampling time k for the i^{th} execution, $\widetilde{X}_{ik}$ is the estimated target state at sampling time k for the i^{th} execution and L is the total sampling time.

For each simulation, RMS Estimation Error versus Sampling Time graphs acquired from different parameter values are plotted on the same figure. To increase the comprehension, tables are given at the end of the figures. In these tables, for each parameter value, Average of all RMS Errors obtained from the graphs in the figures are given. Average of all RMS Errors is calculated as below:

$$\text{Average of all RMS Errors} = \frac{\sum_{k=0}^{L} RMS\ Error(k)}{L} \tag{4.2}$$

4.1 Effect of the Correlation Coefficient

In this section, effects of the correlation coefficient are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for five different values of correlation coefficient, which are [0.1 0.3 0.5 0.7 0.9], after 500 executionss.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

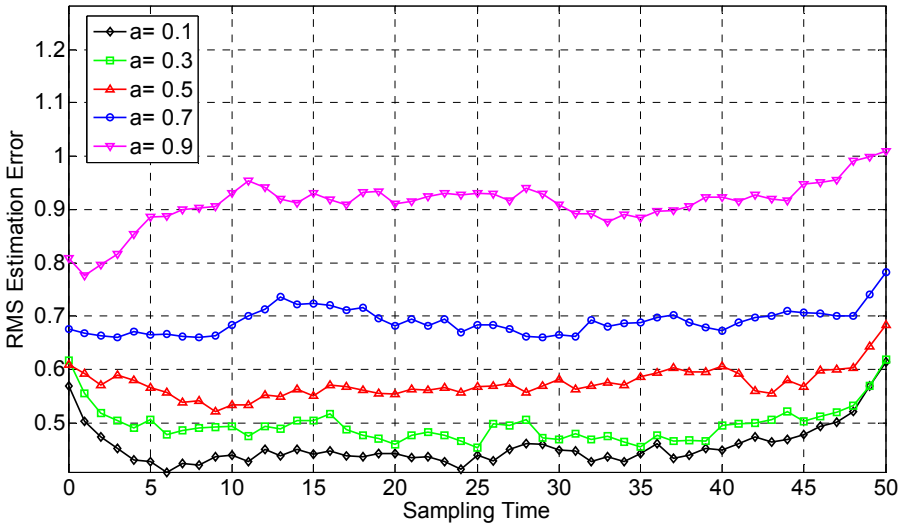

Figure-13: RMS estimation error versus sampling time for the linear model as the correlation coefficient changes

Table-1 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the correlation coefficient changes

Correlation Coefficient(a)	Average of all RMS Errors
0.1	0.4558
0.3	0.4960
0.5	0.5737
0.7	0.6902
0.9	0.9116

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k) = a\, v(k-1) + r(k)$

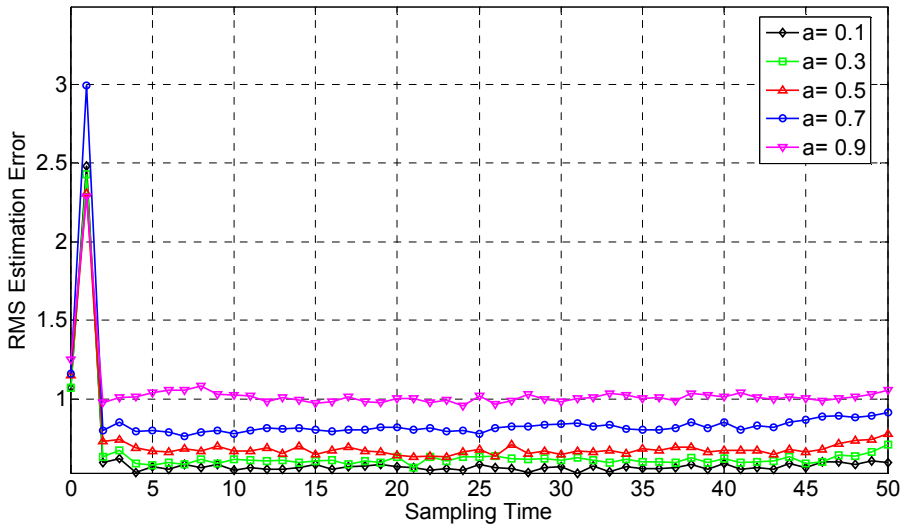

Figure-14: RMS estimation error versus sampling time for the nonlinear model as the correlation coefficient changes

Table-2 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the correlation coefficient changes

Correlation Coefficient(a)	Average of all RMS Errors
0.1	0.6168
0.3	0.6598
0.5	0.7202
0.7	0.8727
0.9	1.0394

Comment:

It can be observed from Figure 13 and Figure 14 that the correlation coefficient is directly proportional with algorithm performance for both linear and nonlinear models. Table 1 and Table 2 show that RMS estimation error increases as the correlation coefficient becomes larger.

4.2 Effect of the Gate Size

In this section, effects of the gate size are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of gate size, which are [0.1 1 2 5], after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

correlation coefficient, a= 0.1

number of maximum states = 100

number of maximum v (k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of w(k) = 3,

Q # of v(0) = 3, Q # of r(k) = 3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] =0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

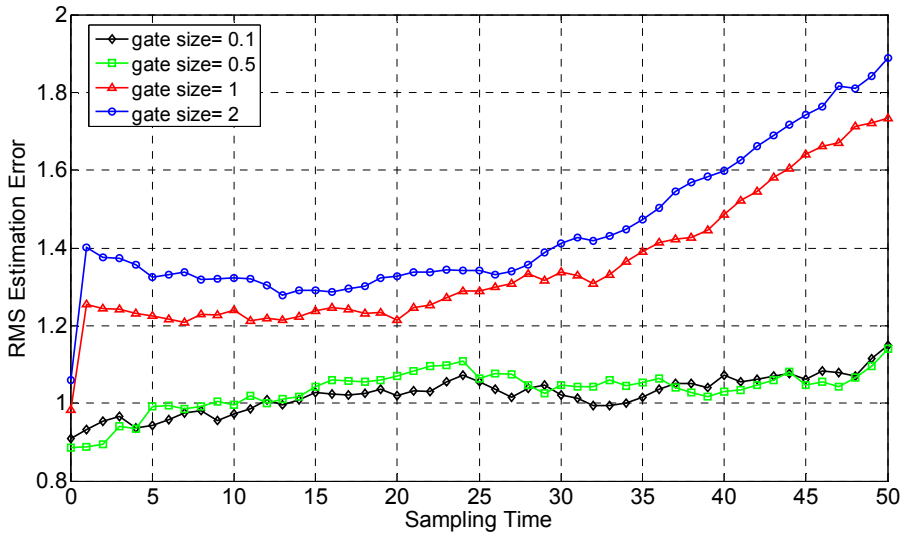

Figure-15: RMS estimation error versus sampling time for the linear model as the gate size changes

Table-3 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the gate size changes

Gate Size	Average of all RMS Errors
0.1	1.0223
1	1.0037
2	1.3495
5	1.4436

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$$v(k)=a\,v(k-1)+r(k)$$

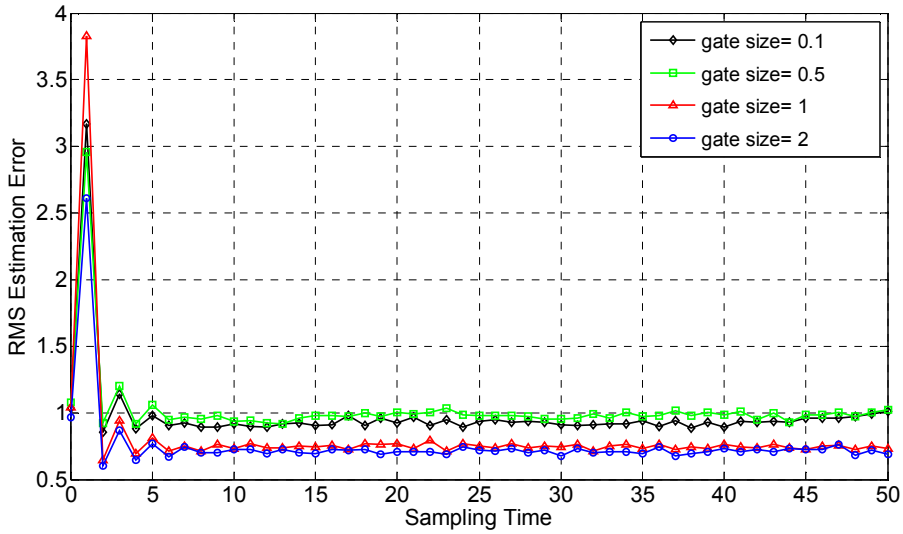

Figure-16: RMS estimation error versus sampling time for the nonlinear model as the gate size changes

Table-4 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the gate size changes

Gate Size	Average of all RMS Errors
0.1	0.9774
1	1.0217
2	0.8142
5	0.7545

Comment:

The effect of the gate size is observed more clearly in the linear model. Figure 15 shows that the gate size is directly proportional with the estimation error. Increase on the gate size causes estimation error to become larger. Figure 16 shows that increase on gate size does not affect estimation error too much in nonlinear model.

4.3 Effect of the Quantization Number of the Initial State Vector

In this section, effects of the quantization number of the initial state vector are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of the quantization number of initial state vector, which are [3 5 7 9], after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of w(k) = 3, Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)]= 0, E[v(0)] = 0 , E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, $v(k)$, is modelled as:

$v(k)=a\,v(k-1)+r(k)$

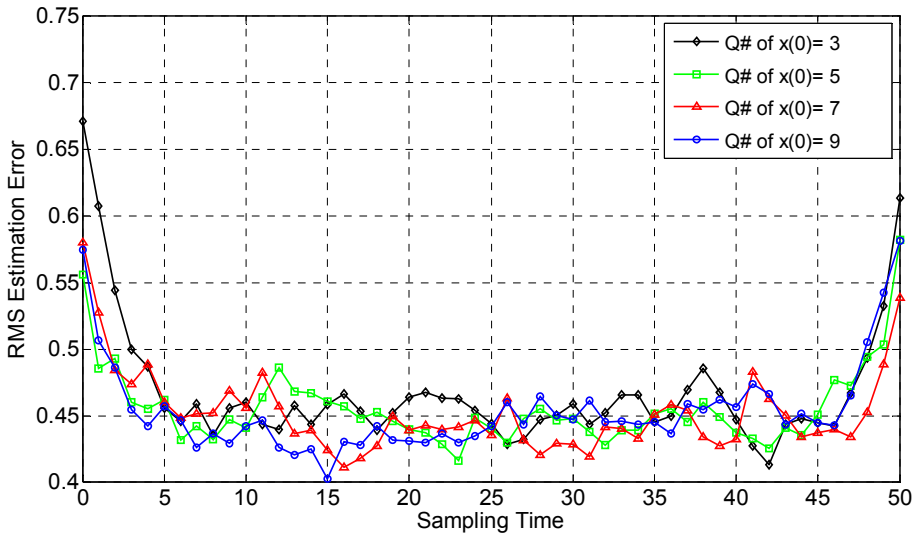

Figure-17: RMS estimation error versus sampling time for the linear model as the quantization number of x(0) changes

Table-5 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the quantization number of x(0) changes

Quantization # of x(0)	**Average of all RMS Errors**
3	0.4681
5	0.4560
7	0.4525
9	0.4541

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k) = a\, v(k-1) + r(k)$

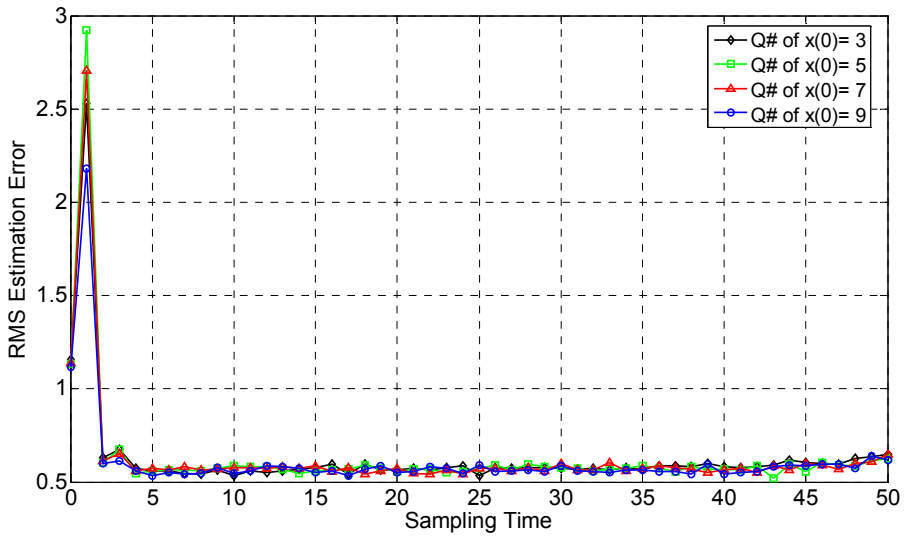

Figure-18: RMS estimation error versus sampling time for the linear model as the quantization number of x(0) changes

Table-6 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the quantization number of x(0) changes

Quantization # of x(0)	Average of all RMS Errors
3	0.6295
5	0.6323
7	0.6282
9	0.6106

Comment:

From Figure 17 and Figure 18, it can be observed that the number of the quantization levels of the initial state vector x(0) slightly affects the performance of the algorithm for both linear and nonlinear models.

4.4 Effect of the Quantization Number of the Disturbance Noise Vector

In this section, effects of the quantization number of the disturbance noise vector are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of quantization number of the disturbance noise, which are [3 5 7 9] after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

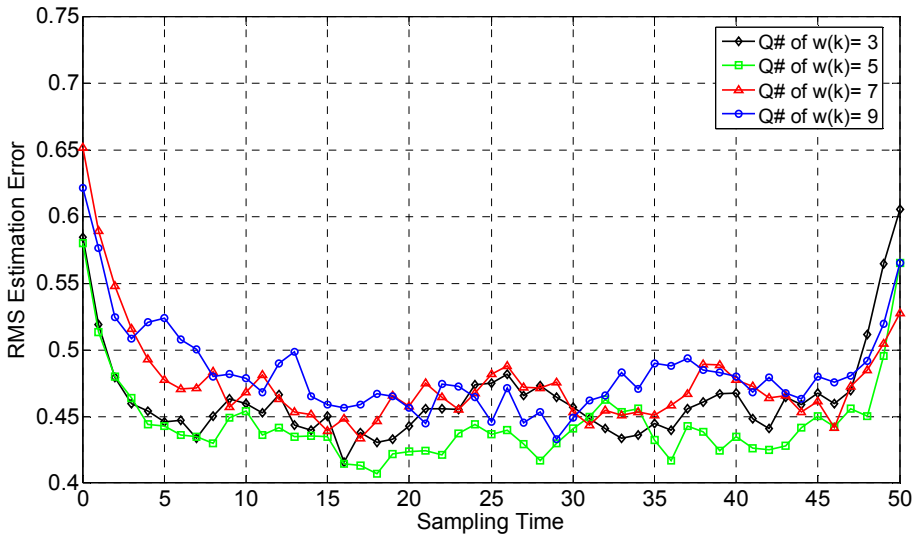

Figure-19: RMS estimation error versus sampling time for the linear model as the quantization number of w(k) changes

Table-7 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the quantization number of w(k) changes

Quantization # of w(k)	**Average of all RMS Errors**
3	0.4639
5	0.4453
7	0.4757
9	0.4835

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

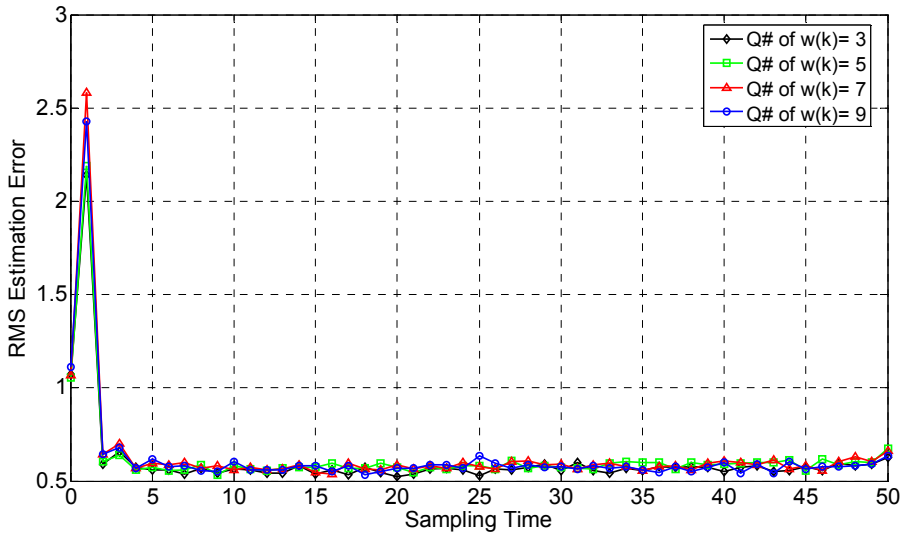

Figure-20: RMS estimation error versus sampling time for the nonlinear model as the quantization number of w(k) change

Table-8 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the quantization number of w(k) changes

Quantization # of w(k)	**Average of all RMS Errors**
3	0.6064
5	0.6241
7	0.6345
9	0.6252

Comment:

From Figure 19 and Figure 20, it can be observed that increasing the number of the quantization levels of the disturbance noise w(k) slightly affects the state estimation error for both linear and nonlinear models.

4.5 Effect of Quantization Number of Initial Measurement Noise

In this section, effects of the quantization number of the initial measurement noise vector are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of the quantization number of initial measurement noise, which are [3 5 7 9] after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a= 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3 , Q # of r(k)=3

variances : var[x(0)]= 1, var[v(0)]=1, var[w(k)]= 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, $v(k)$, is modelled as:

$v(k)=a\,v(k-1)+r(k)$

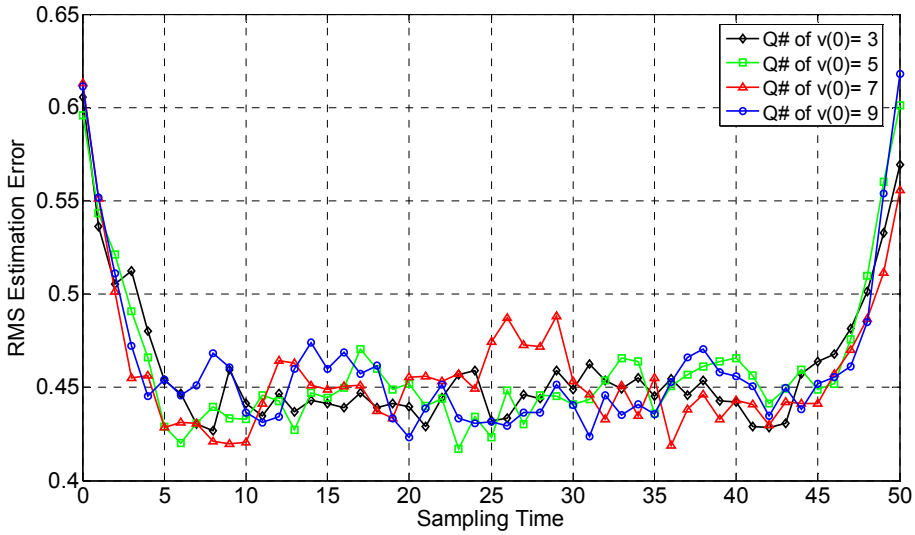

Figure-21: RMS estimation error versus sampling time for the linear model as the quantization number of v(0) changes

Table-9 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as quantization number of v(0) changes

Quantization # of v(0)	**Average of all RMS Errors**
3	0.4593
5	0.4603
7	0.4581
9	0.4603

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k) = a\, v(k-1) + r(k)$

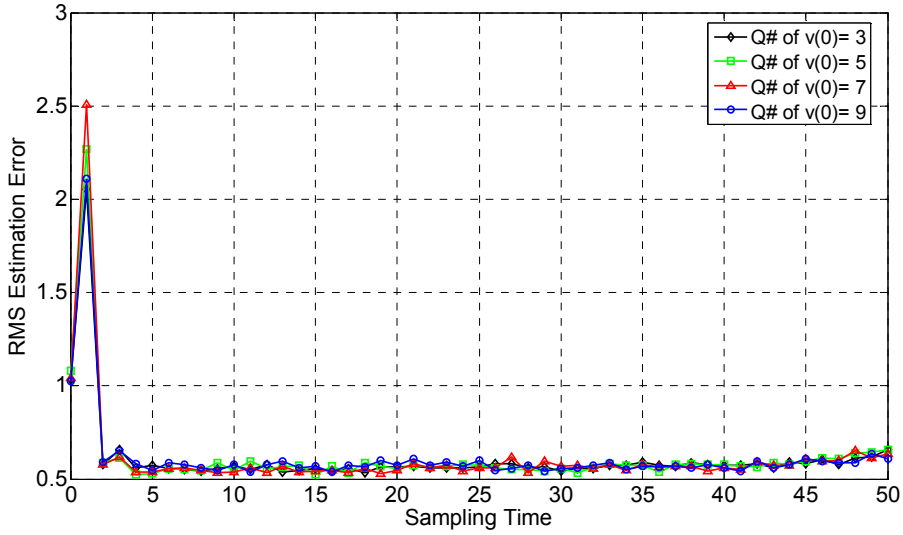

Figure-22: RMS estimation error versus sampling time for the nonlinear model as the quantization number of v(0) changes

Table-10 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as quantization number of v(0) changes

Quantization # of v(0)	Average of all RMS Errors
3	0.6096
5	0.6147
7	0.6140
9	0.6140

Comment:

From Figure 21 and Figure 23, it can be observed that the number of the quantization levels of the initial measurement noise vector v(0) slightly affects the performance of the algorithm for both linear and nonlinear models.

4.6 Effect of Quantization Number of White Gaussian Noise Vector Component of Measurement Noise Vector

In this section, effects of the quantization number of the white Gaussian noise component of measurement noise vector are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of the quantization number of the white Gaussian noise component of measurement noise vector, which are [3 5 7 9] after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of v(0)=3, Q # of w(k)=3

variances : var[x(0)]= 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

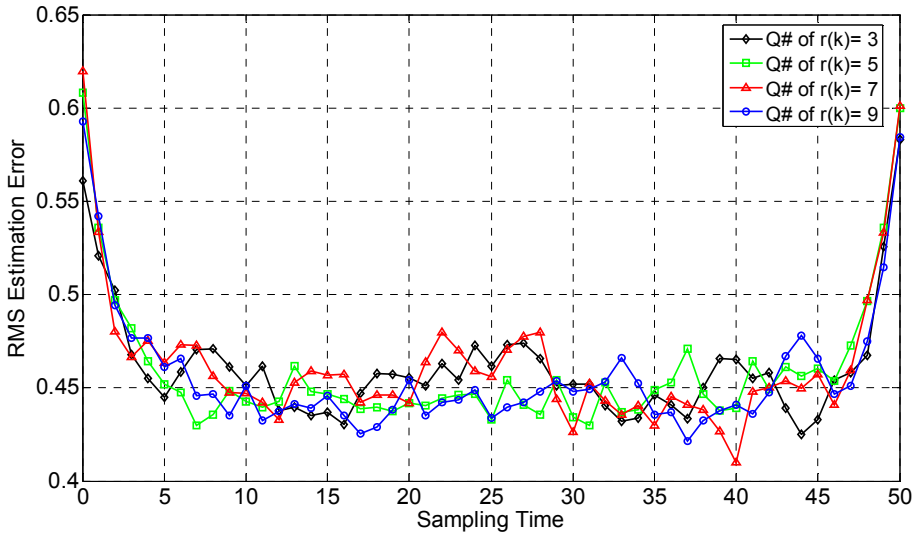

Figure-23: RMS estimation error versus sampling time for the linear model as the quantization number of r(k) changes

Table-11 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as quantization number of r(k) changes

Quantization # of r(k)	**Average of all RMS Errors**
3	0.4607
5	0.4591
7	0.4626
9	0.4567

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$$v(k)=a\,v(k-1)+r(k)$$

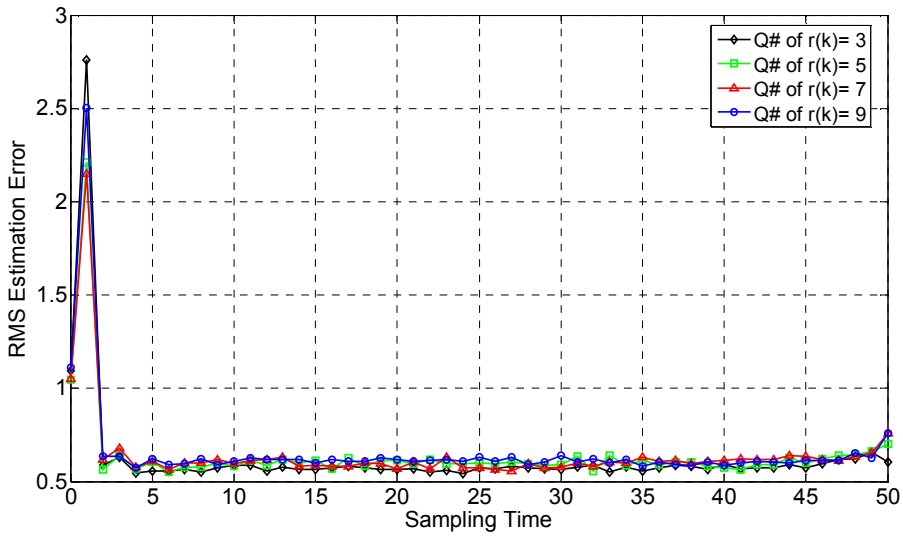

Figure-24: RMS estimation error versus sampling time for the nonlinear model as the quantization number of r(k) changes

Table-12 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as quantization number of r(k) changes

Quantization # of r(k)	**Average of all RMS Errors**
3	0.6288
5	0.6418
7	0.6443
9	0.6598

Comment:

From Figure 23 and Figure 24, it can be observed that the number of the quantization levels of r(k), slightly affects the performance of the algorithm for both linear and nonlinear models.

4.7 Effect of the Initial State Variance

In this section, effects of the initial state variance are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of initial state variance, which are [0.01 0.1 1 3], after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient , a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,
Q # of v(0) = 3, Q # of r(k)=3

variances : var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0 ,E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

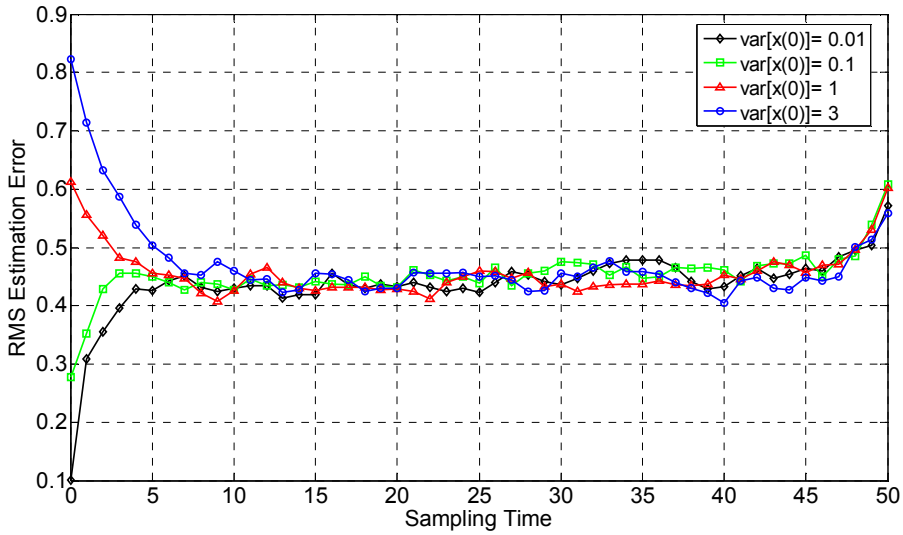

Figure-25: RMS estimation error versus sampling time for the linear model as the variance of x(0) changes

Table-13 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the variance of x(0) changes

Variance of x(0)	**Average of all RMS Errors**
0.01	0.4364
0.1	0.4512
1	0.4566
3	0.4725

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k) = a\, v(k-1) + r(k)$

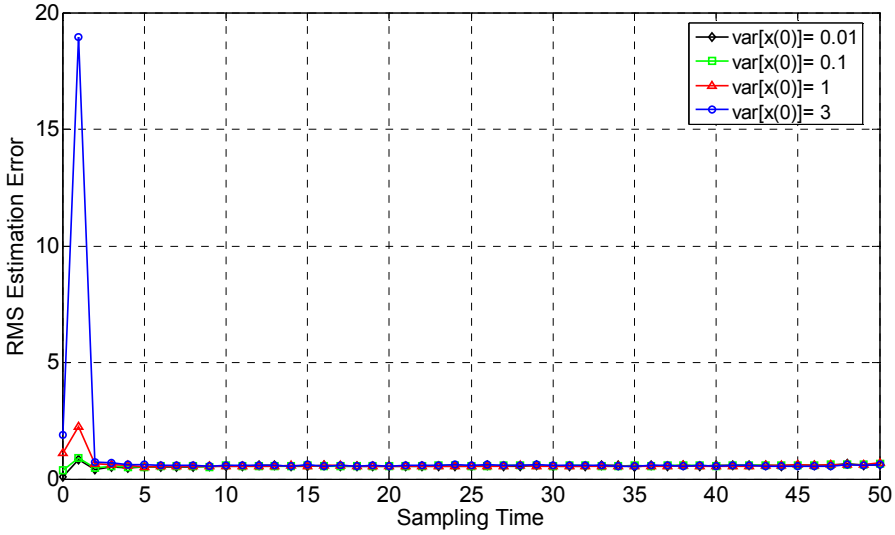

Figure-26: RMS estimation error versus sampling time for the nonlinear model as the variance of x(0) changes

Table-14 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the variance of x(0) changes

Variance of x(0)	Average of all RMS Errors
0.01	0.5535
0.1	0.5666
1	0.6110
3	0.9679

Comment:

When Figure 25 and Figure 26 are studied, it can be observed that increase on the variance of the initial state vector slightly affects the performance of the state error estimation. As shown by figures, the variance of the initial state vector has important effects on only first samples. The state estimation error begins with large error values due to the large initial state variance values, and then settles to lower values as sampling time, k, increases.

4.8 Effect of the Disturbance Noise Variance

In this section, effects of the disturbance noise vector are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of disturbance noise variance, which are [0.01 0.1 1 3], after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1

expected values : E[x(0)]= 0, E[w(k)] = 0, E[v(0)]= 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

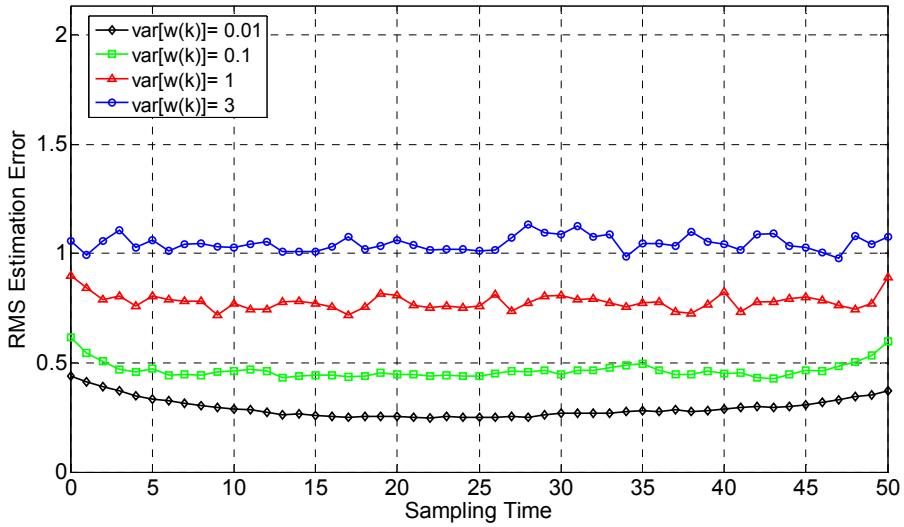

Figure-27: RMS estimation error versus sampling time for the linear model as the variance of w(k) changes

Table-15 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the variance of w(k) changes

Variance of w(k)	**Average of all RMS Errors**
0.01	0.2936
0.1	0.4655
1	0.7782
3	1.0463

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$$v(k)=a\,v(k-1)+r(k)$$

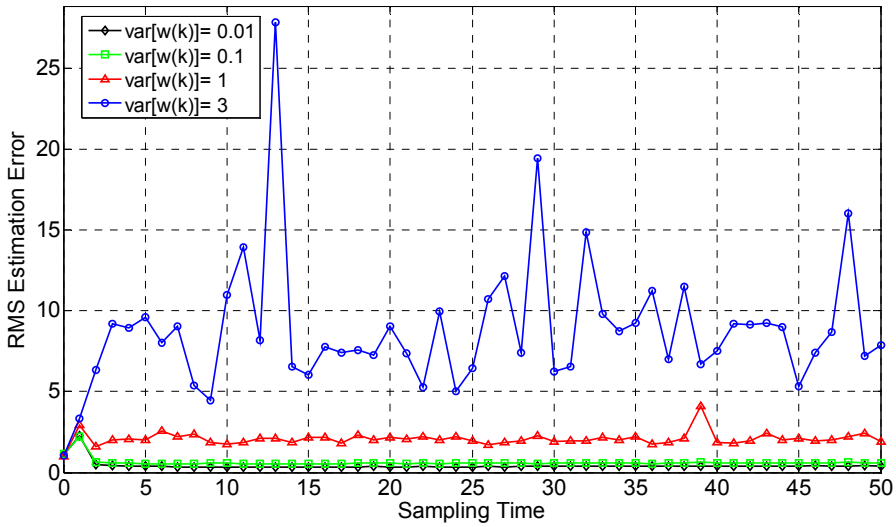

Figure-28: RMS estimation error versus sampling time for the nonlinear model as the variance of w(k) changes

Table-16 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the variance of w(k) changes

Variance of w(k)	**Average of all RMS Errors**
0.01	0.4227
0.1	0.6160
1	2.0786
3	8.8283

Comment:

It can be observed from Figure 27 and Figure 28 that little change on the disturbance noise variance directly affects performances of the state estimation error for both linear and nonlinear models. Table 15 and Table 16 show that increase on the disturbance noise variance causes the performance of the estimation to get worse.

4.9 Effect of the Initial Measurement Noise Variance

In this section, effects of the initial measurement noise variance are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of initial measurement noise variance, which are [0.01 0.1 1 10] , after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,
Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[w(k)] = 0.1, var[r(k)]= 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

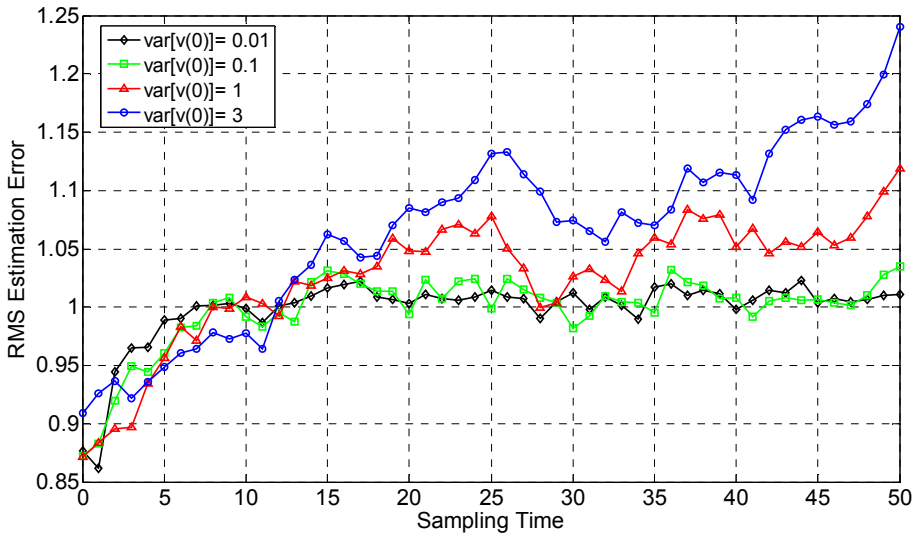

Figure-29: RMS estimation error versus sampling time for the linear model as the variance of v(0) changes

Table-17 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the variance of v(0) changes

Variance of v(0)	**Average of all RMS Errors**
0.01	0.9982
0.1	0.9977
1	1.0258
3	1.0653

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

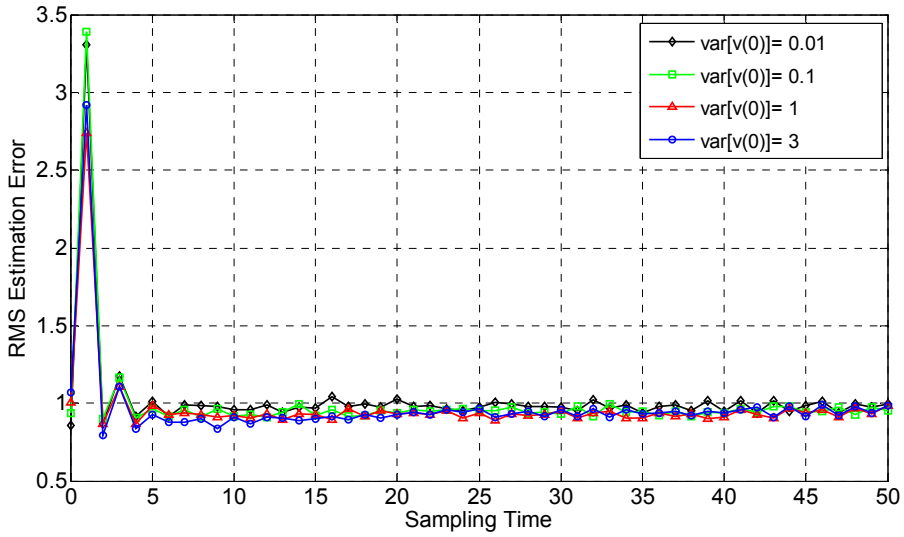

Figure-30: RMS estimation error versus sampling time for the nonlinear model as the variance of v(0) changes

Table-18 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the variance of v(0) changes

Variance of v(0)	**Average of all RMS Errors**
0.01	1.0230
0.1	0.9971
1	0.9683
3	0.9689

Comment:

The effect of the measurement noise variance is observed more clearly in the linear model. Figure 29 shows that the measurement noise variance does not

affect the performance of the algorithm very much for initial sampling time values, after a sampling time about 15, it is observed that increase on the the measurement noise variance causes estimation error to become larger. Figure 30 shows that increase on the initial measurement noise variance does not affect estimation error too much in nonlinear model.

4.10 Effect of the Limit of the Maksimum State Number

In this section, effects of the limit of the maksimum state number are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of limit of the maksimum state number, which are [10000 1000 100 50] , after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

number of maximum v(k) states = 50

quantization numbers : Q # of x(0) = 5, Q # of v(0) =3,

Q # of w(k) =3 , Q # of r(k) =3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

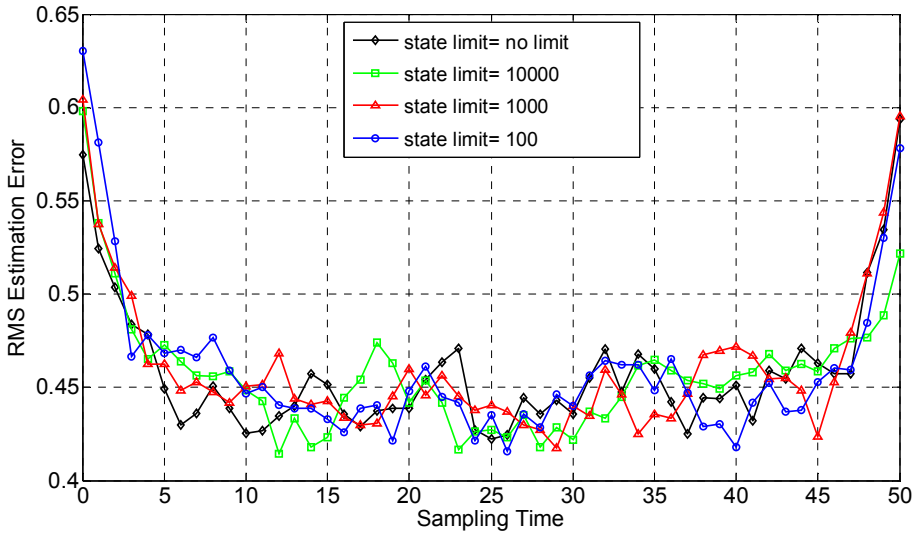

Figure-31: RMS estimation error versus sampling time for the linear model as the value of the state limit changes

Table-19 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the value of the state limit changes

State Limit	Average of all RMS Errors
no limit	0.4577
10000	0.4569
1000	0.4601
100	0.4595

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k) = a\, v(k-1) + r(k)$

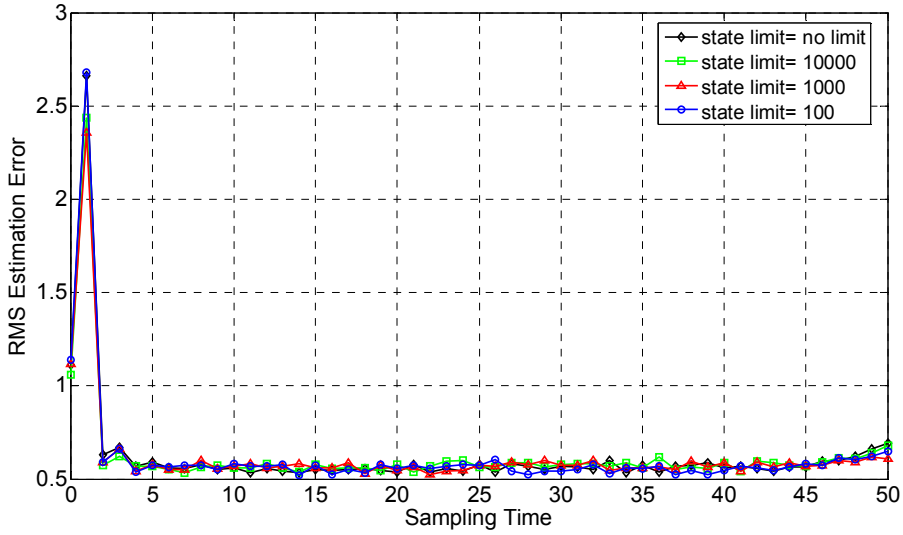

Figure-32: RMS estimation error versus sampling time for the nonlinear model as the value of the state limit changes

Table-20 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the value of the state limit changes

State Limit	Average of all RMS Errors
no limit	0.6216
10000	0.6222
1000	0.6175
100	0.6184

Comment:

It can be observed from the figures that limiting the number of states does not affect the algorithm performance in linear (Figure 31) and nonlinear (Figure 32) models too much. In the simulations, the program chooses the paths that have better metrics when the number of the states exceeds the state limit and cancels others. Since the program computation time is directly related with the maximum state number, the computation time can be reduced significantly by decreasing the state number without any loss in the performance.

4.11 Effect of the Limit of the Maximum Measurement Noise State Number

In this section, effects of the limit the of maximum measurement noise state number are investigated. There are two models, which are linear and nonlinear. For each model, simulations are obtained for four different values of the limit of maximum measurement noise state number, which are [1000 500 100 50] after 500 executions.

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time, L = 50

gate size = 0.1

correlation coefficient, a = 0.1

number of maximum states = 100

quantization numbers : Q # of x(0) = 5, Q # of v(0) =3,

Q # of w(k) =3, Q # of r(k) =3

variances : var[x(0)]= 1, var[v(0)]=1, var[w(k)] = 1

expected values : E[x(0)]= 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$v(k)=a\, v(k-1)+r(k)$

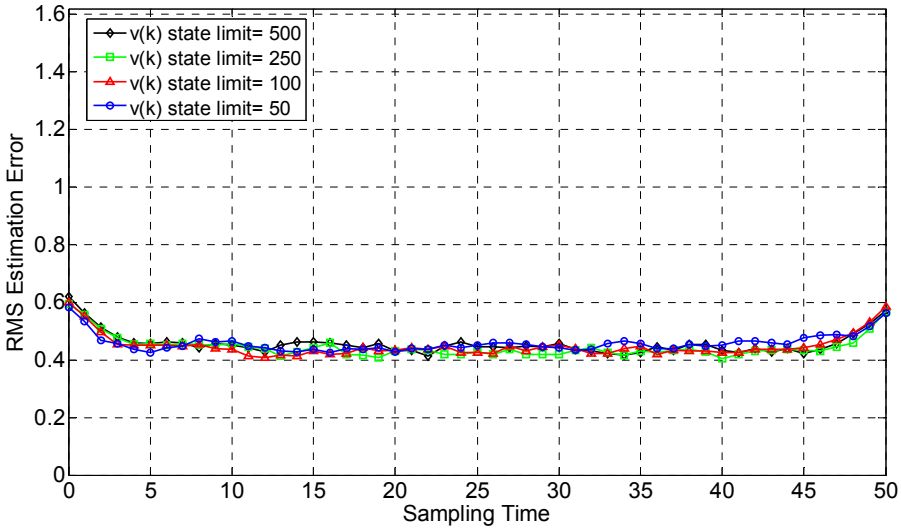

Figure-33: RMS estimation error versus sampling time for the linear model as the value of the state limit of v(k) changes

Table-21 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the value of the state limit of v(k) changes

v(k) State Limit	**Average of all RMS Errors**
500	0.4557
250	0.4445
100	0.4476
50	0.4575

Nonlinear models used in this simulation are:

Motion model : $x(k+1) = \exp(-x(k)) + w(k)$

Measurement model : $z(k) = \cos(x(k)) + v(k)$

where correlated measurement noise, v (k), is modelled as:

$$v(k)=a\,v(k-1)+r(k)$$

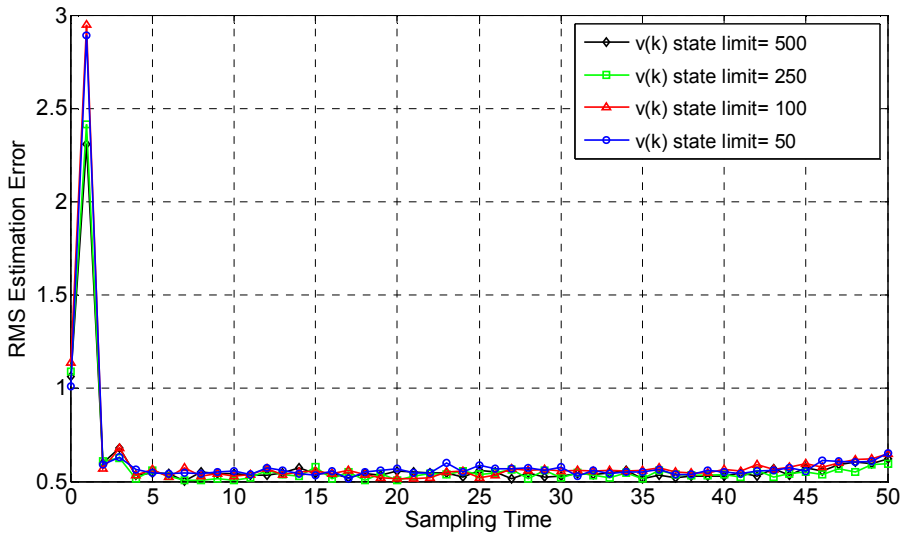

Figure-34: RMS estimation error versus sampling time for the nonlinear model as the value of the state limit of v(k) changes

Table-22 : Average values of all RMS estimation errors from k=0 to k=L for the nonlinear model as the value of the value of the state limit of v(0) changes

v(k) State Limit	**Average of all RMS Errors**
500	0.5937
250	0.5887
100	0.6177
50	0.6176

Comment:

It can be observed from the figures that limiting the number of states of v(k) does not affect the algorithm performance in linear (Figure 33) and nonlinear (Figure 34) models too much. In the simulations, the program chooses the v(k) states that have better metrics when the number of the states exceeds the v(k) state limit and cancels others. Since the program computation time is directly related with the maximum v(k) state number, the computation time can be reduced significantly by decreasing the maximum state number of v(k) without any loss in the performance.

CHAPTER 5

ALPHA-BETA FILTER WITH CORRELATED MEASUREMENT NOISE

In this chapter, target tracking system in the presence of correlated measurement noise by using Alpha-Beta Filter will be studied. Rogers [3, 4] described the correlated noise as a first-order Markov process in non-maneuvering case.

5.1 Correlated Measurement Noise Model

The tracking system produces a measurement of target position of the form:

$$z(k) = x(k) + v(k) \tag{5.1}$$

where $x(k)$ is the true target position, and $v(k)$ is the correlated measurement error at time $t_k = kT$. The measurement interval T is presumed constant. The variable $v(k)$ is modeled by the first-order Markov process:

$$v(k) = av(k-1) + r(k) \tag{5.2}$$

where a is the correlation coefficient ($0 \leq a \leq 1$), and $r(k)$ is zero-mean white noise, with variance:

$$R = \sigma_r^2 = (1 - a^2)\,\sigma_v^2 \tag{5.3}$$

where σ_v^2 is variance of the measurement noise $v(k)$.

At high measurement frequencies, the discrete measurement sequence can be approximated by a continuous time measurement process of equal spectral density [3,4,5]. So the sequence $v(k)$ can be viewed as a discrete sampling of the continuous Markov process $v(t)$ satisfying:

$$dv/dt = -\frac{v(t)}{\tau} + r(t)$$

$$a = \exp(-\frac{T}{\tau})$$

$$R(t) = E\left[r(t')r(t'+t) = \frac{2\sigma^2}{\tau}\delta(t) \right] \tag{5.4}$$

where τ is the correlation time.

5.2 Decorrelation Process

In this section, the deccorelation process [3] will be described. To decorrelate the measurement noise, a new measurement $y(k)$, called *"artificial measurement"*, is generated by using the measurement given in Eq. (5.1) as below:

$$y(k) = z(k) - a\ z(k-1) = x(k) + v(k) - a\left[x(k-1) + v(k-1)\right]$$

$$y(k) = x(k) - a\,x(k-1) + \underbrace{v(k) - a\,v(k-1)}_{r(k)}$$

$$y(k) = \left[x(k) - a\,x(k-1)\right] + r(k) \tag{5.5}$$

for which the measurement errors are uncorrelated.

5.3 Alpha-Beta Filter

The general form of the Kalman Filter is described by the following equations:

$$\begin{aligned} &\textit{Motion model,} && s(k+1) = A\,s(k) + \Gamma w(k) \\ &\textit{Measurement model,} && z(k) = M\,s(k) + v(k) \end{aligned} \tag{5.6}$$

where

- *s(k)* is the state vector of the target in track, at the k^{th} instant
- *w(k)* is the disturbance noise vector with zero-mean
- A is the state transition matrix
- Γ is the excitation or maneuver matrix
- *z(k)* is the measurement vector of target in track.
- M is the measurement or selection matrix

Alpha-Beta Filter is derived from Kalman Filter. It is a sub-optimum algorithm which has constant gain, good track-following ability, ease of adaptation to changes in the tracking conditions, and low computational cost [9].

The standart equations for the Alpha-Beta Filter are obtained by substituting:

$$s(k) = \begin{bmatrix} x(k) \\ \dot{x}(k) \end{bmatrix}; A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}; M = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

in motion and measurment models of Kalman Filter given in Eq. (5.6). This is a constant-velocity model for target motion, with only position measurements available. It is assumed that each position coordinate *x* in the target state vector is decoupled from others and can be treated separately.

In this section, Alpha-Beta Filter will be implemented to track the target in correlated measurement noise case.

Let the target state vector be denoted by

$s(k) = \begin{bmatrix} x(k) & \dot{x}(k) \end{bmatrix}^t$, where " t" denoes the matrix taransposition,

$x(k)$ and $\dot{x}(k)$ represent the true target position and velocity at time kT.

The target motion model is:

$$s(k) = As(k-1) + \Gamma w(k)$$

where $A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$, $\Gamma = \begin{bmatrix} T & 1 \end{bmatrix}^t$, $w(k)$ is a Gaussian distributed disturbance noise vector at time k with zero mean and with variance (qT). The units of (q) are [meter/sec^2]^2 per Hz.

The measurement model equation may be written as:

$$z(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} s(k) + v(k)$$

or, in terms of artificial measurement model defined in section 5.2, $y(k)$, as:

$$y(k) = H\, s(k) + v(k) \tag{5.7}$$

where $H = \begin{bmatrix} (1-a) & aT \end{bmatrix}$.

Proof of Eq.(5.7):

$$y(k) = z(k) - a\,z(k-1) = \begin{bmatrix}1 & 0\end{bmatrix}s(k) + v(k) - a\left(\begin{bmatrix}1 & 0\end{bmatrix}s(k-1) + v(k-1)\right)$$

$$y(k) = \begin{bmatrix}1 & 0\end{bmatrix}s(k) - a\begin{bmatrix}1 & 0\end{bmatrix}s(k-1) + \underbrace{v(k) - a\,v(k-1)}_{r(k)} \tag{5.9}$$

From Eq.(5.6), $s(k-1) = A^{-1}s(k) - A^{-1}\Gamma w(k)$, *with disturbance noise, if* $s(k-1)$ *is inserted in Eq.(5.9), the equation below is obtained*

$$y(k) = \begin{bmatrix}1 & 0\end{bmatrix}s(k) - a\begin{bmatrix}1 & 0\end{bmatrix}A^{-1}s(k) + \underbrace{a\begin{bmatrix}1 & 0\end{bmatrix}A^{-1}\Gamma w(k) + r(k)}_{\eta(k)}$$

$$\eta(k) = a\begin{bmatrix}1 & 0\end{bmatrix}A^{-1}\Gamma w(k) + r(k) \tag{5.10}$$

In practical applications, the first term of right-hand side in Eq.(5.10) is usually small and can be neglected without degrading much performance [7]. So we have $\eta(k) \approx r(k)$.

$$y(k) = \underbrace{\left(\begin{bmatrix}1 & 0\end{bmatrix} - a\begin{bmatrix}1 & 0\end{bmatrix}A^{-1}\right)}_{H}s(k) + r(k) \tag{5.11}$$

$$y(k) = Hs(k) + r(k)$$

$$H = \begin{bmatrix}1 & 0\end{bmatrix}(1 - aA^{-1})$$

$$A^{-1} = \begin{bmatrix}1 & -T\\ 0 & 1\end{bmatrix}$$

$$H = \begin{bmatrix}1 & 0\end{bmatrix}\left(1 - a\begin{bmatrix}1 & -T\\ 0 & 1\end{bmatrix}\right) = \begin{bmatrix}1 & 0\end{bmatrix} - \begin{bmatrix}a & 0\end{bmatrix}\begin{bmatrix}1 & -T\\ 0 & 1\end{bmatrix} = \begin{bmatrix}1 & 0\end{bmatrix} - \begin{bmatrix}a & -aT\end{bmatrix}$$

$H = \begin{bmatrix}(1-a) & aT\end{bmatrix}$, *and results Eq.(5.7)*

The use of $y(k)$ brings the model into the standard form of Kalman filter theory, in which the measurement noise presumed white.

The optimal steady-state filter has the form:

$$\hat{s}(k) = A\hat{s}(k-1) + G\left[y(k) - H\,A\hat{s}(k-1)\right] \tag{5.12}$$

where the gain, *G* is:

$$G = PH^{t}R^{-1} \tag{5.13}$$

and *P*, the steady-state covariance matrix of the smoothed state estimate obeys the algebraic Riccati equation:

$$P = (I - PH^{t}R^{-1}H)(APA^{t} + qT\Gamma\Gamma^{t}) \tag{5.14}$$

An explicit solution for P, which is derived in the Appendix B, is as follows:

$$\begin{aligned} P(1,1) &= r''\left[\alpha - 2\gamma\beta + \gamma^{2}c\right] \\ P(1,2) &= \left(\frac{r''}{T}\right)\left[\beta - \gamma c\right] \\ P(1,2) &= P(2,1) \\ P(2,2) &= \left(\frac{r''}{T^{2}}\right)c \end{aligned} \tag{5.15}$$

where

$$r'' = r\frac{(1+a)}{(1-a)}$$

$$r = \sigma_{v}^{2}$$

$$\gamma = \frac{a}{1-a}$$

$$\lambda^2 = \frac{q T^3}{r^{"}}$$

$$d = \frac{\lambda}{2} + \frac{1}{2}\sqrt{16 + [\lambda(1+2\lambda)]^2}$$

$$x = \frac{d}{2} - \sqrt{\left(\frac{d}{2}\right)^2 - 1}$$

and,

$$\alpha = 1 - x^2$$

$$\beta = \lambda x$$

$$c = \frac{\alpha\beta}{1-\beta} - \lambda^2(1+\gamma) \qquad (5.16)$$

Note that the parameter λ plays the role of the dimensionless *"target maneuvering index"*.

CHAPTER 6

SIMULATION RESULTS OF ALPHA_BETA FILTER

In this chapter, some simulations are carried out to demonstrate the effects of the parameters, which are used in the Alpha-Beta Filter algorithm in the presence of correlated measurement noise. These parameters are:

- correlation coefficient

(denoted by "a" in the figures)

- measurement interval

(denoted by "T" in the figures)

- disturbance noise nariance

(denoted by "var[w(k)]" in the figures)

- measurement noise variance

(denoted by "var[v(k)]" in the figures)

- initial state variance

(denoted by "var[s(0)]" in the figures)

Simulations are performed for only linear model because Alpha-Beta Filter Algorithm can be applied only for linear models. The simulations are obtained after 500 Monte-Carlo runs. For each run, the state vector *s(k)* and the measurement vector *z(k)* are regenerated with the same motion and measurement equations.

As a result of performed simulations, RMS Estimation Error versus Sampling Time graphs are given in figures to show the performance of the Alpha-Beta Filter Algorithm. RMS Estimation error for a given sampling time, k, is calculated as below:

$$\text{RMS Error(k)} = \frac{\sqrt{\sum_{i=1}^{N}(X_{ik} - \tilde{X}_{ik})^2}}{N} \quad \text{k=0,1,…, L} \tag{6.1}$$

where RMS Error(k) is the RMS Error for sampling time k, N is the total execution number, X_{ik} is the real target state at sampling time k for the i^{th} execution, $\tilde{X}_{ik}$ is the estimated target state at sampling time k for the i^{th} execution and L is the total sampling time.

For each simulation, RMS Estimation Error versus Sampling Time graphs acquired from different parameter values are plotted on the same figure. To increase the comprehension, tables are given at the end of the figures. In these tables, for each parameter value, Average of all RMS Errors obtained from the graphs in the figures are given. Average of all RMS Errors is calculated as below:

$$\text{Average of all RMS Errors} = \frac{\sum_{k=0}^{L} RMS\ Error(k)}{L} \tag{6.2}$$

6.1 Effect of the Correlation Coefficient

6.1.1 Simulation 1

In this section, effects of the correlation coefficient are investigated. Simulations are obtained for five different values of correlation coefficient, which are [0.1 0.3 0.5 0.7 0.9] , after 500 executions.

Linear models used in this simulation are:

Motion model : $s(k+1) = s(k) + \Gamma w(k)$

Measurement model : $z(k) = Ms(k) + v(k)$

where correlated measurement noise, v (k), is modelled as: $v(k)=a\, v(k-1)+r(k)$,

$\Gamma=[T\ 1]^t$, $M=[1\ 0]$

The following parameters are used in the simulations:

total sampling time = 50

measurement interval, T = 0.1

variances : var[s(0)] = 1, var[v(k)]=1, var[w(k)]= 0.1

expected values : E[s(0)] = 0, E[w(k)] = 0, E[v(k)] = 0 , E[r(k)] = 0

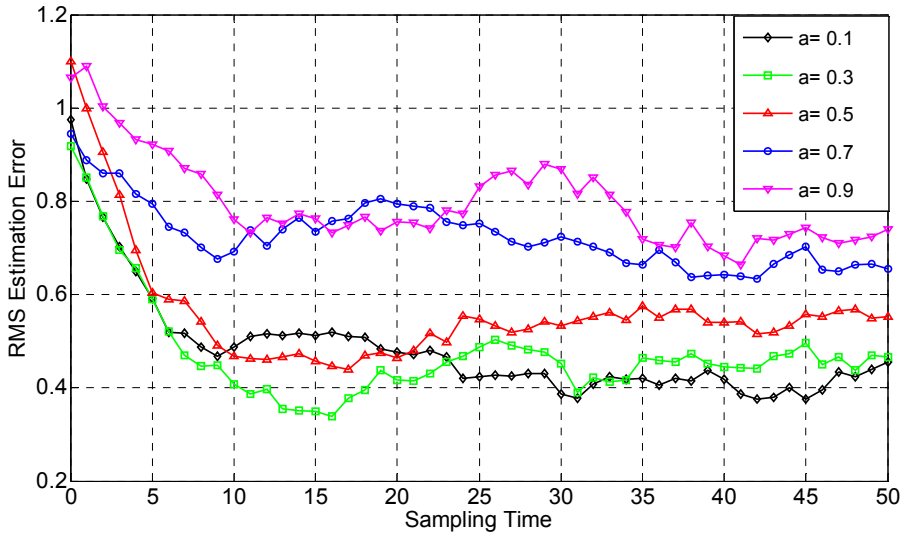

Figure-35: RMS estimation error versus sampling time as the correlation coefficient changes

Table-23 : Average values of all RMS estimation errors from k=0 to k=L as the correlation coefficient changes

Correlation Coefficient (a)	Average of all RMS Errors
0.1	0.4832
0.3	0.4752
0.5	0.5618
0.7	0.7266
0.9	0.7963

Comment:

It can be observed from the Figure 35 that the correlation coefficient is directly proportional with algorithm performance. Table 23 shows that RMS estimation error increases as the correlation coefficient becomes larger.

6.1.2 Simulation 2

Parameter values used in this simulation are same as the parameter values used in simulation 1, except measurment interval, T=1 is taken.

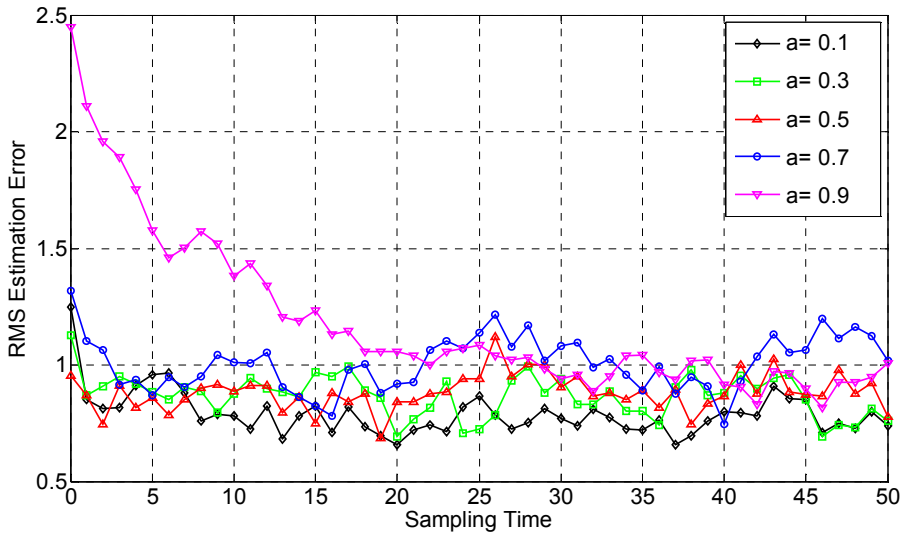

Figure-36: RMS estimation error versus sampling time as the correlation coefficient changes

Table-24 : Average values of all RMS estimation errors from k=0 to k=L as the correlation coefficient changes

Correlation Coefficient (a)	Average of all RMS Errors
0.1	0.7898
0.3	0.8676
0.5	0.8818
0.7	1.0076
0.9	1.1805

Comment:

It can be observed from the Figure 36 that the correlation coefficient is directly proportional with algorithm performance. Table 24 shows that RMS estimation error increases as the correlation coefficient becomes larger.When compared with simulation 1 given in section 6.1.1, average of all RMS errors are larger since the measurement interval, T, is increased from 0.1 to 1.

6.2 Effect of the Measurement Interval (T)

In this section, effects of the measurement interval are investigated. Simulations are obtained for four different values of measurement interval, which are [0.1 0.5 1 2] after 500 executions.

The following parameters are used in the simulations:

total sampling time = 50

correlation coefficient, a= 0.1

variances : var[s(0)]= 1, var[v(k)]=1, var[w(k)] = 0.1

expected values : E[s(0)] = 0, E[w(k)]= 0, E[v(k)] = 0, E[r(k)] = 0

Linear models used in this simulation are:

Motion model : $s(k+1) = s(k) + \Gamma w(k)$

Measurement model : $z(k) = Ms(k) + v(k)$

where correlated measurement noise, v (k), is modelled as: $v(k)=a\, v(k-1)+r(k)$, $\Gamma=[T\ 1]^t$, $M=[1\ 0]$

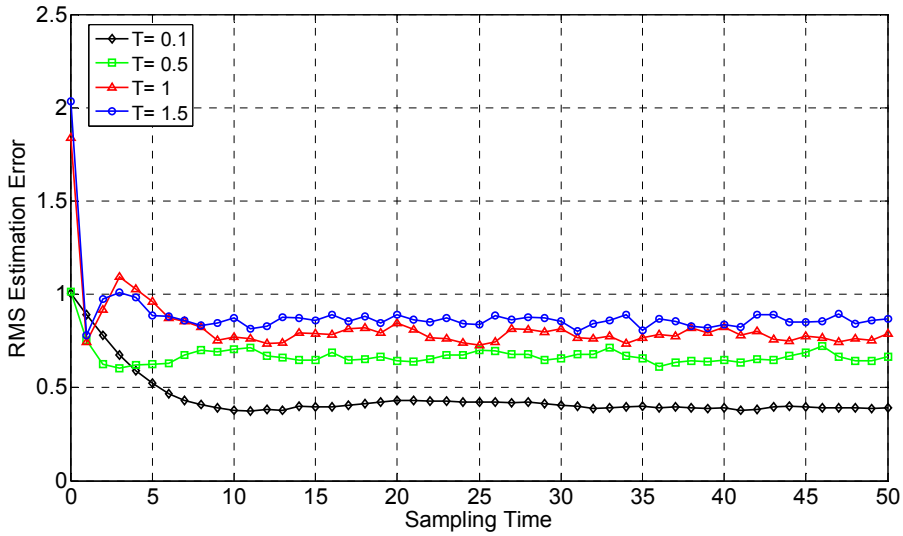

Figure-37: RMS estimation error versus sampling time as the measurement interval changes

Table-25 : Average values of all RMS estimation errors from k=0 to k=L as the measurement interval changes

Measurement Interval,T	Average of all RMS Errors
0.1	0.4423
0.5	0.6698
1	0.8192
2	0.8868

Comment:

It can be observed from the Figure 37 that the measurement interval is directly proportional with algorithm performance. Table 25 shows that RMS estimation error increases as the measurement interval becomes larger.

6.3 Effect of the Disturbance Noise Variance

In this section, effects of the disturbance variance are investigated. Simulations are obtained for four different values of the disturbance variance, which are [0.01 0.1 1 3] after 500 executions.

Linear models used in this simulation are:

Motion model : $s(k+1) = s(k) + \Gamma w(k)$

Measurement model : $z(k) = Ms(k) + v(k)$

where correlated measurement noise, v (k), is modelled as: $v(k)=a\, v(k-1)+r(k)$,

$\Gamma=[T\ 1]^t$, $M=[1\ 0]$

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time = 50

measurement interval, T =0.1

correlation coefficient, a = 0.1

variances : var[s(0)] = 1, var[v(k)]=1

expected values : E[s(0)] = 0, E[w(k)] = 0, E[v(k)] = 0, E[r(k)] = 0

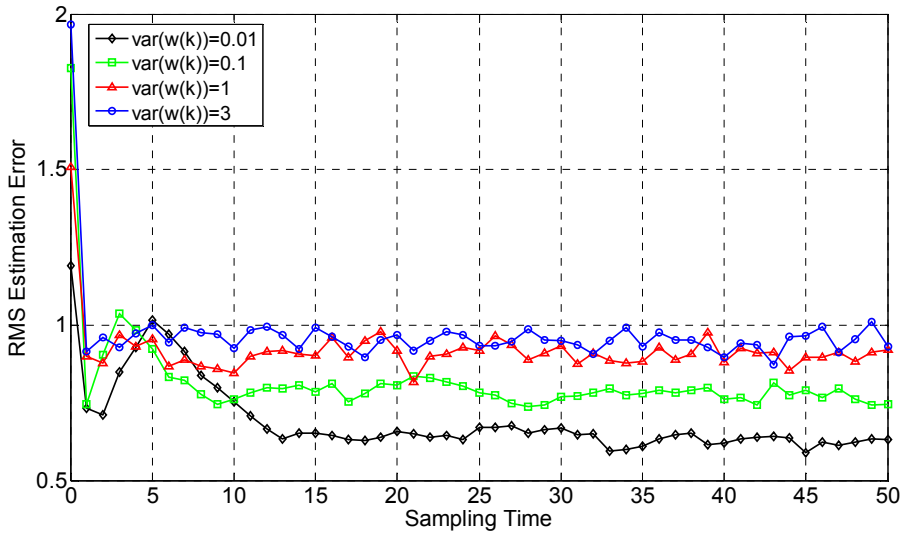

Figure-38: RMS estimation error versus sampling time as the disturbance noise variance changes

Table-26 : Average values of all RMS estimation errors from k=0 to k=L as the disturbance noise variance changes

Variance of w(k)	Average of all RMS Errors
0.01	0.6924
0.1	0.8165
1	0.9184
3	0.9713

Comment:

It can be observed from the Figure 38 that disturbance noise variance is directly proportional with algorithm performance. Table 26 shows that RMS estimation error increases as the disturbance noise variance becomes larger.

6.4 Effect of the Measurement Noise Variance

In this section, effects of the measurement noise variance are investigated. Simulations are obtained for four different values of measurement noise variance, which are [0.01 0.1 1 3] after 500 executions.

Linear models used in this simulation are:

Motion model : $s(k+1) = s(k) + \Gamma w(k)$

Measurement model : $z(k) = Ms(k) + v(k)$

where correlated measurement noise, v (k), is modelled as: $v(k)=a\, v(k-1)+r(k)$,

$\Gamma=[T\ 1]^t$, $M=[1\ 0]$

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time = 50

measurement interval, T = 0.1

correlation coefficient, a = 0.1

variances : var[s(0)] = 1, var[w(k)]=0

expected values : E[s(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)] = 0

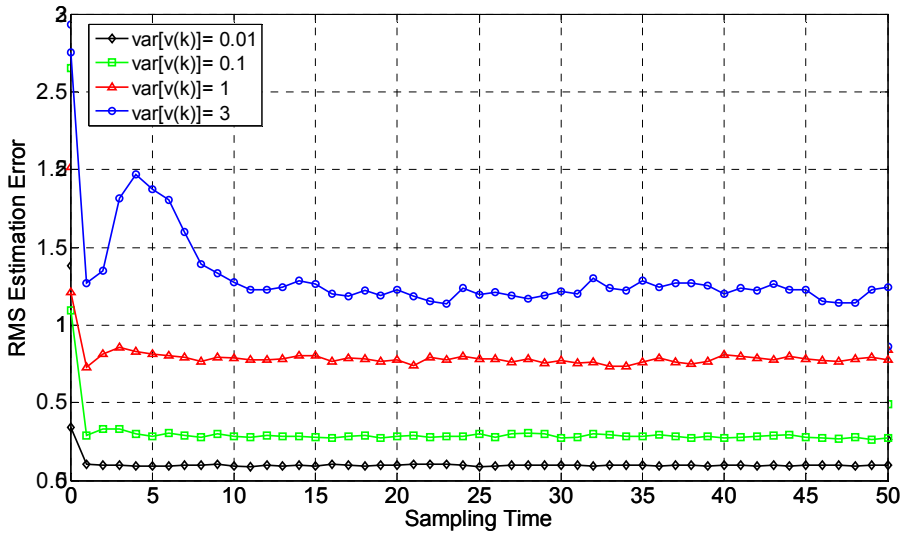

Figure-39: RMS estimation error versus sampling time for the linear model as the measurement noise variance changes

Table-27 : Average values of all RMS estimation errors from k=0 to k=L for the linear model as the measurement noise variance changes

Variance of v(k)	Average of all RMS Errors
0.01	0.1018
0.1	0.3024
1	0.7880
3	1.3167

Comment:

It can be observed from the Figure 39 that measurement noise variance is directly proportional with algorithm performance. Table 27 shows that RMS estimation error increases as the measurement noise variance becomes larger.

6.5 Effect of the Initial State Variance

In this section, effects of the initial state variance are investigated. Simulations are obtained for four different values of initial state variance, which are [0.01 0.1 1 1.2] after 500 executions.

Linear models used in this simulation are:

Motion model : $s(k+1) = s(k) + \Gamma w(k)$

Measurement model : $z(k) = Ms(k) + v(k)$

where correlated measurement noise, v (k), is modelled as: $v(k)=a\, v(k-1)+r(k)$,

$\Gamma=[T\ 1]^t$, $M=[1\ 0]$

The following parameters are used in the simulations for both linear and nonlinear models:

total sampling time = 50

measurement interval, T = 0.1

correlation coefficient, a = 0.1

variances : var[w(k])=0, var[v(k)]=1

expected values : E[s(0)] = 0, E[w(k)] = 0, E[v(k)] = 0, E[r(k)] = 0

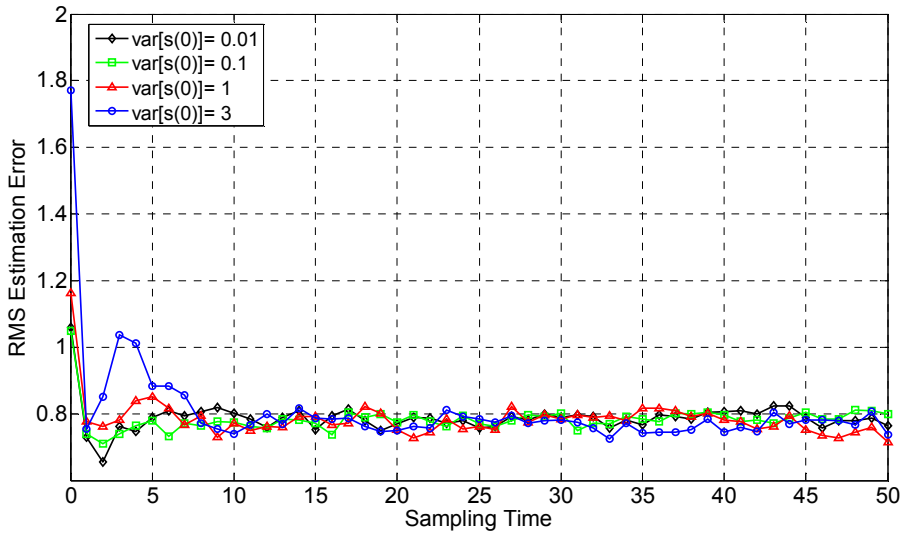

Figure-40: RMS estimation error versus sampling time as the initial state variance changes

Table-28 : Average values of all RMS estimation errors from k=0 to k=L as the initial state variance changes

Variance of s(0)	Average of all RMS Errors
0.01	0.7876
0.1	0.7833
1	0.7847
3	0.8074

Comment:

It can be observed from the Figure 40 that initial state variance slightly affects the performance of the algorithm.

CHAPTER 7

COMPARISION OF ODSA WITH ALPHA-BETA FILTER ALGORITHM

We proposed a method to handle the correlated noise problem in target tracking by using ODSA in Chapter 3. Rogers proposed decorrelation method to handle the the correlated noise problem in target tracking by using Alpha-Beta Filter algorithm [3,4], which was explained in Chapter 5. In this chapter, ODSA with correlated measurement noise and Alpha-Beta Filter Algorithm with correlated measurement noise will be compared.

ODSA as described in Chapter 3 supports both linear and nonlinear motion and measurement equations, whereas the Alpha-Beta Filter Algorithm described in Chapter 5 supports only linear equations. In order to compare these algorithms, linear motion and measurement equations will be used for both algorithms.

Linear models used in ODSA simulations are:

Motion model : $x(k+1) = x(k) + w(k)$

Measurement model : $z(k) = x(k) + v(k)$

where correlated measurement noise, $v(k)$, is modelled as:

$v(k)=a\, v(k-1)+r(k)$

Linear models used in Alpha-Beta Filter simulations are:

Motion model : $s(k+1) = s(k) + \Gamma w(k)$

Measurement model : z (k) = Ms(k) + v (k)

where correlated measurement noise, v (k), is modelled as: v(k)=a v(k-1)+r(k) , $\Gamma=[T\ 1]^t$, M=[1 0].

As a result of performed simulations, RMS Estimation Error versus Sampling Time graphs are given in figures to compare the performance of both ODSA and Alpha-Beta Filter Algorithm for the same parameter values. RMS Estimation error for a given sampling time, k, is calculated as below:

$$\text{RMS Error}(k) = \frac{\sqrt{\sum_{i=1}^{N}(X_{ik} - \widetilde{X}_{ik})^2}}{N} \quad k=0,1,\ldots, L \qquad (7.1)$$

where RMS Error(k) is the RMS Error for sampling time k, N is the total execution number, X_{ik} is the real target state at sampling time k for the i^{th} execution, $\widetilde{X}_{ik}$ is the estimated target state at sampling time k for the i^{th} execution and L is the total sampling time.

For each simulation, RMS Estimation Error versus Sampling Time graphs acquired from ODSA and Alpha-Beta Filter Algorithm are plotted on the same figure. To increase the comprehension, tables are given at the end of the figures. In these tables, for each parameter value, Average of all RMS Errors obtained from the graphs in the figures are given. Average of all RMS Errors is calculated as below:

$$\text{Average of all RMS Errors} = \frac{\sum_{k=0}^{L} RMS\ Error(k)}{L} \qquad (7.2)$$

7.1 Comparision as the Correlation Coefficient Changes

In this section, effects of the correlation coefficient on both ODSA and Alpha-Beta Filter Algorithm are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)]= 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)] = 0

Parameters used in Alpha-Beta Filter Algorithm are:

total sampling time, L = 50

measurement interval = 0.1

variances : var[s(0)] = 1, var[v(k)]=1, var[w(k)]= 0.1

expected values : E[s(0)] = 0, E[w(k)] = 0, E [v(k)]= 0 , E[r(k)] = 0,

7.1.1 Simulation 1

In this simulation, results are obtained for five different values of correlation coefficient, which are [0.01 0.03 0.05 0.07 0.09], after 500 executions for each algorithm.

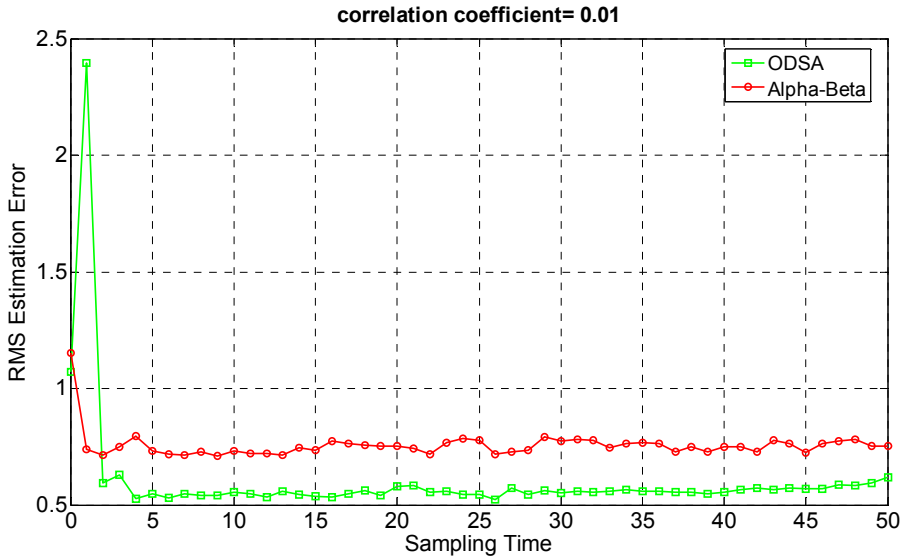

Figure-41: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.01

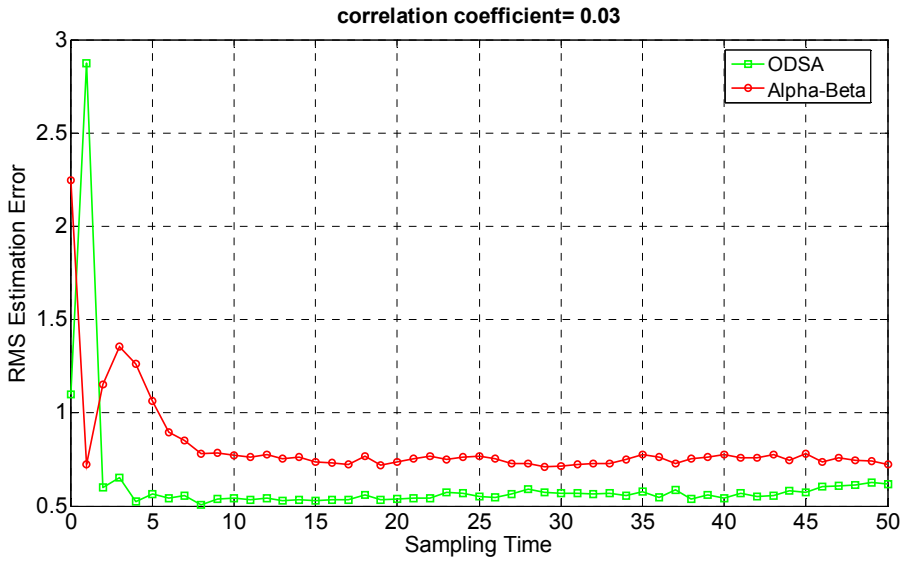

Figure-42: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.03

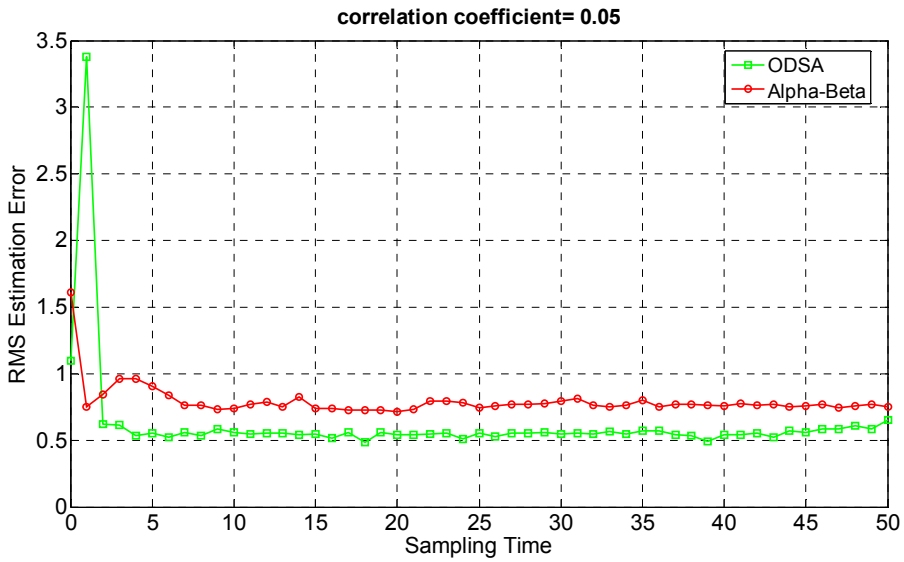

Figure-43: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.05

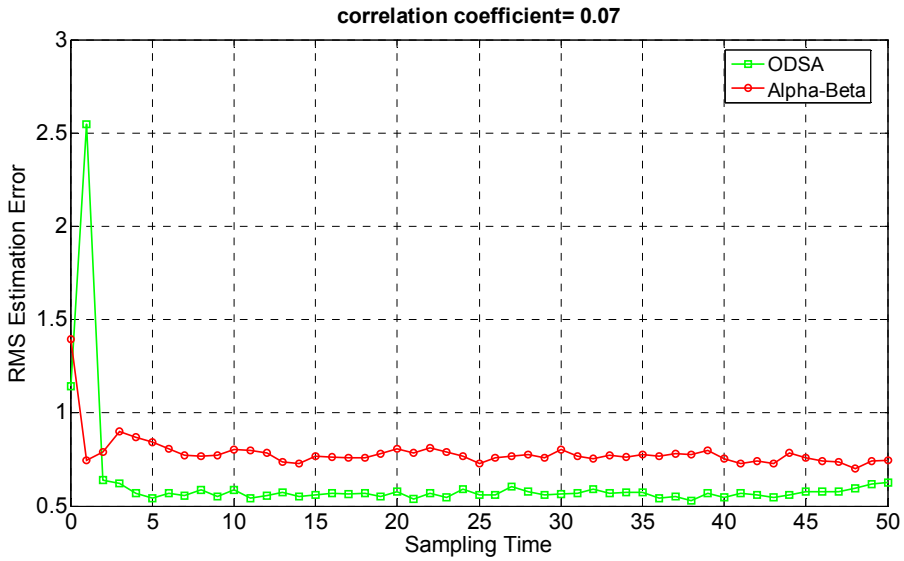

Figure-44: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.07

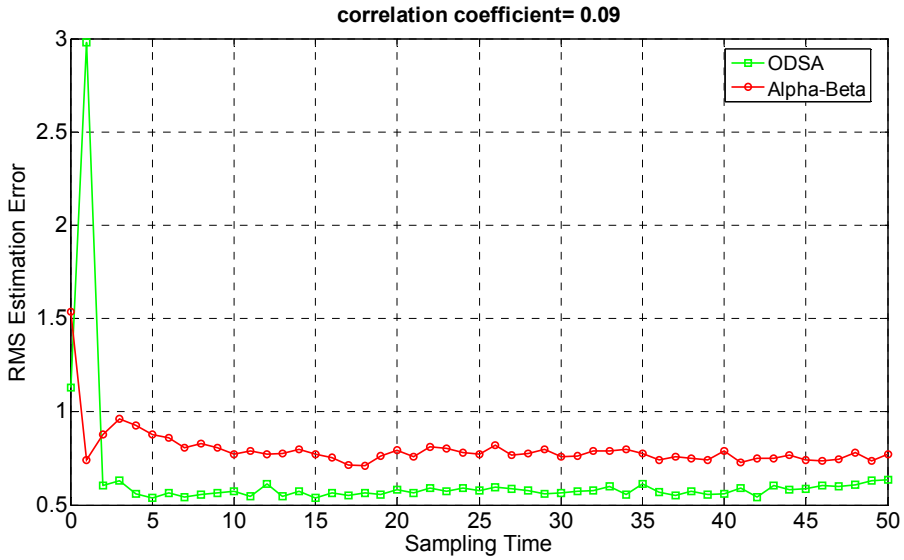

Figure-45: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm as when the correlation coefficient equals 0.09

Table-29 : Average values of all RMS estimation errors from k=0 to k=L for both ODSA and Alpha-Beta Filter Algorithm as the correlation coefficient changes

Correlation Coefficient, a	0.01	0.03	0.05	0.07	0.09
ODSA Average of all RMS Errors	0.6051	0.6168	0.6206	0.6197	0.6344
Alpha-Beta Average of All RMS Errors	0.7561	0.8194	0.7927	0.7842	0.7964

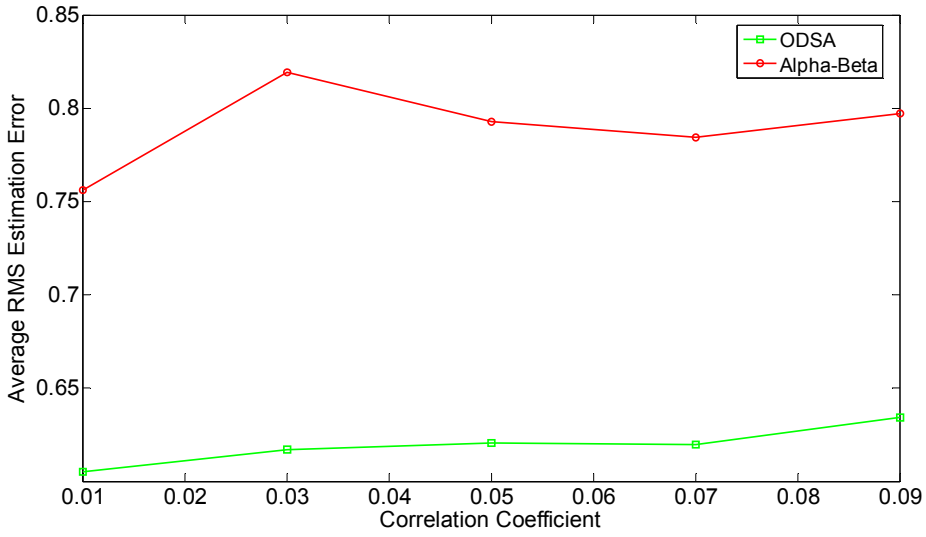

Figure-46: Estimation performance comparison of ODSA and Alpha-Beta Filter Algorithm as the correlation coefficient changes

Comment:

Figure 46 and Table 29 show that small correlation coefficient variations do not affect very much the performance of the both algorithms. It can be observed from

Figures 41, 42, 43, 44 and 45 that ODSA shows a better estimation performance than Alpha-Beta Filter Algorithm.

7.1.2 Simulation 2

Parameter values used in this simulation are same as the parameter values used in simulation 1 given in section 7.1.1, except correlation coefficient, a, varies as [0.1 0.3 0.5 0.7 0.9].

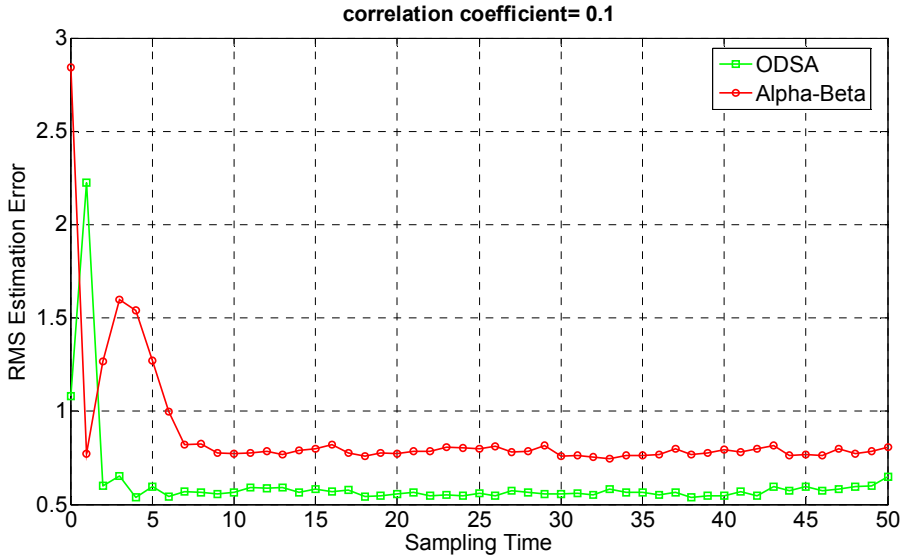

Figure-47: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.1

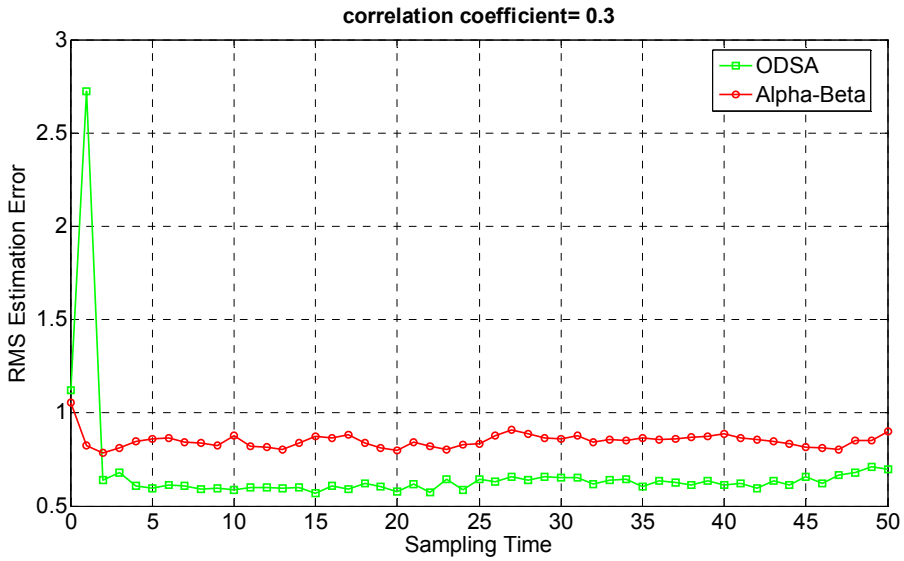

Figure-48: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.3

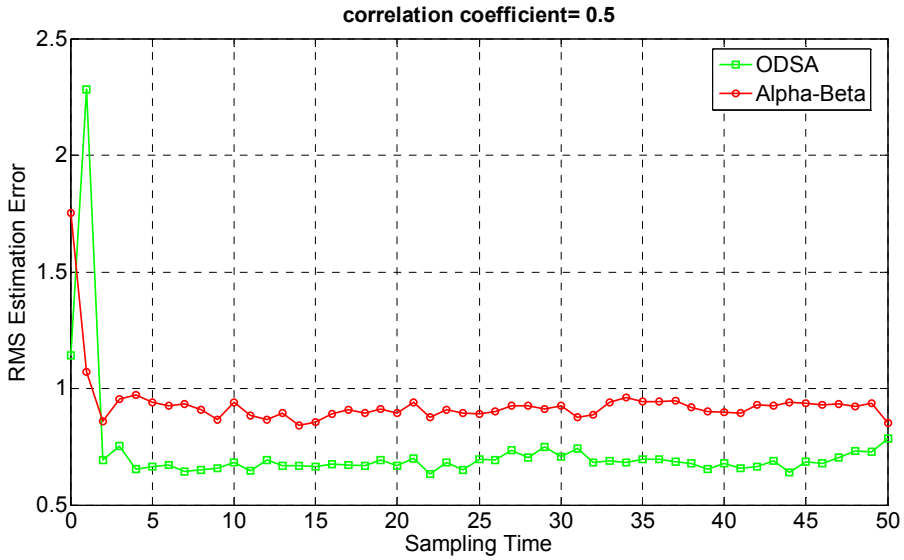

Figure-49: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.5

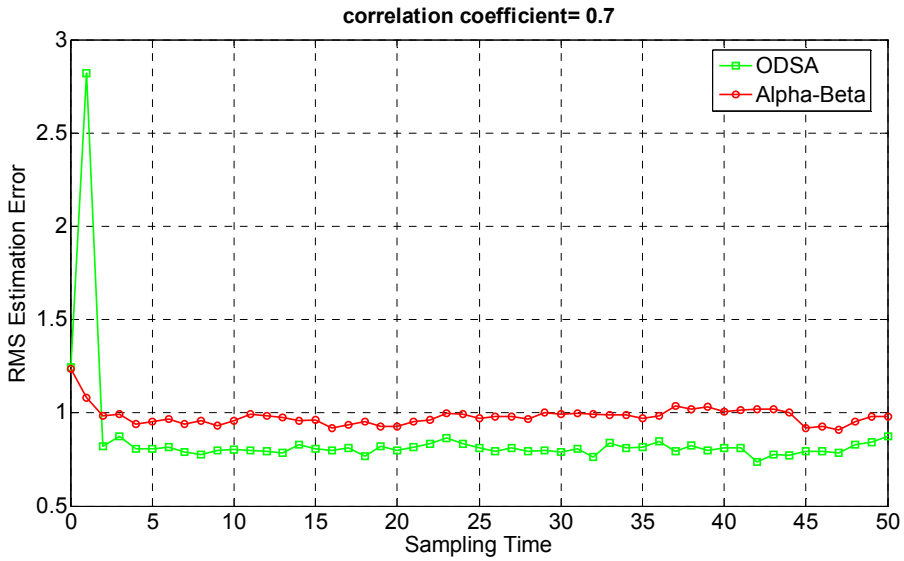

Figure-50: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.7

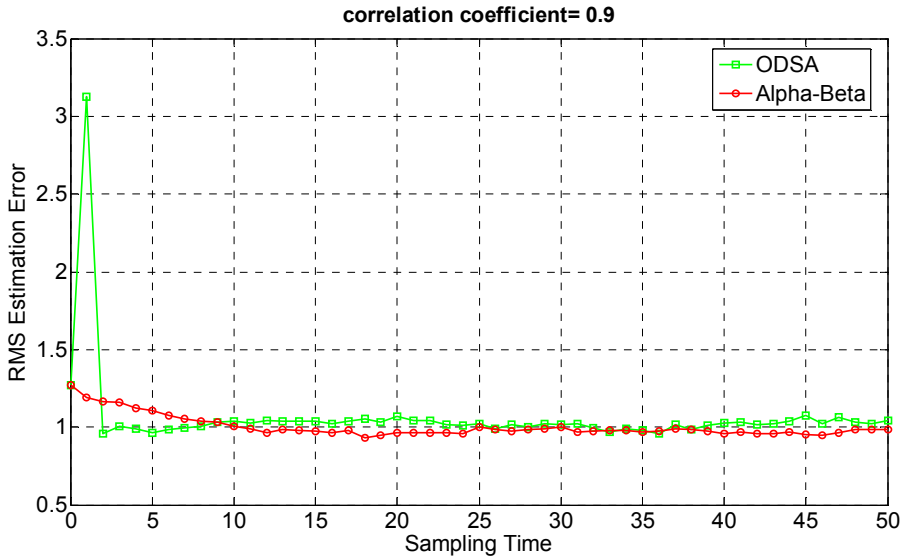

Figure-51: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the correlation coefficient equals 0.9

Table-30 : Average values of all RMS estimation errors from k=0 to k=L for both ODSA and Alpha-Beta Filter Algorithm as the correlation coefficient changes

Correlation Coefficient, a	0.1	0.3	0.5	0.7	0.9
ODSA Average of all RMS Errors	0.6114	0.6743	0.7257	0.8552	1.0642
Alpha-Beta Average of All RMS Errors	0.8778	0.8504	0.9309	0.9798	1.0023

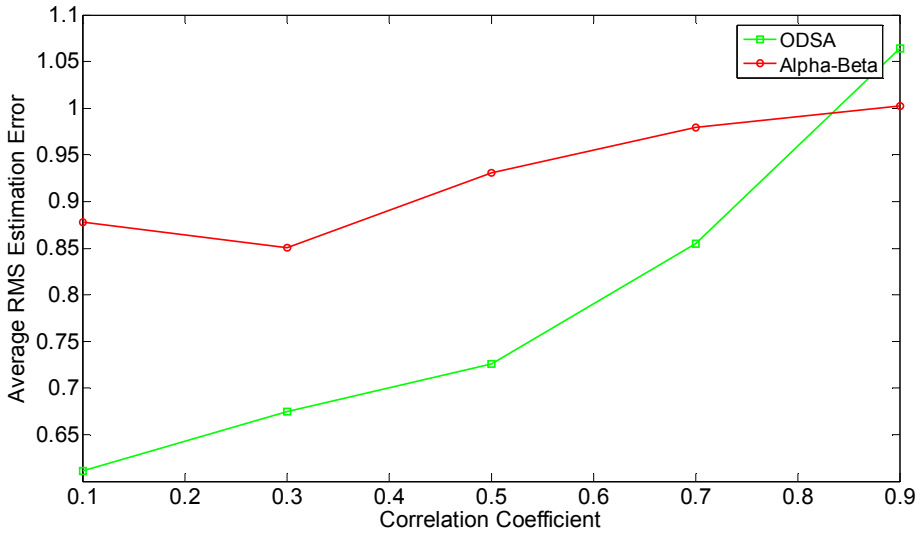

Figure-52: Estimation performance comparison of ODSA and Alpha-Beta Filter Algorithm as the correlation coefficient changes

Comment:

Figure 52 and Table 30 show that increasing the correlation coefficient causes the performances of the both algorithms to decrease. It can be observed from Figures 47, 48, 49, 50 and 51 that ODSA shows a better estimation performance than

Alpha-Beta Filter Algorithm. For only Figure 44, Alpha-Beta Filter Algorithm shows a better estimation performance than ODSA.

7.2 Comparision as the Disturbance Noise Variance Changes

In this section, effects of the disturbance noise variance on both ODSA and Alpha-Beta Filter Algorithm are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Parameters used in Alpha-Beta Filter Algorithm are:

total sampling time, L = 50

measurement interval = 1

variances : var[x(0)]= 1, var[v(k)]=1, var[w(k)]= 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0, E[r(k)] = 0

In this simulation, results are obtained for four different values of disturbance noise variance, which are [0.01 0.1 0.5 1] after 500 executions for each algorithm.

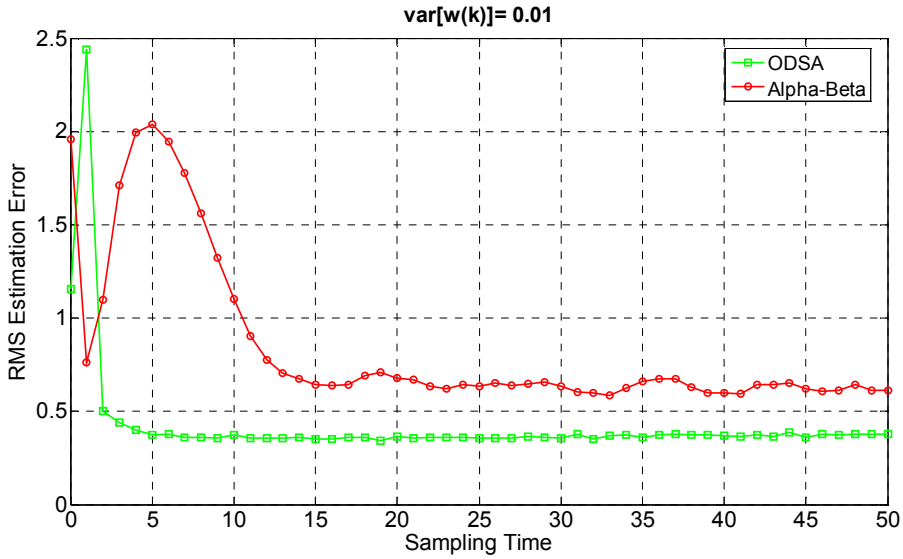

Figure-53: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of w(k) equals 0.01

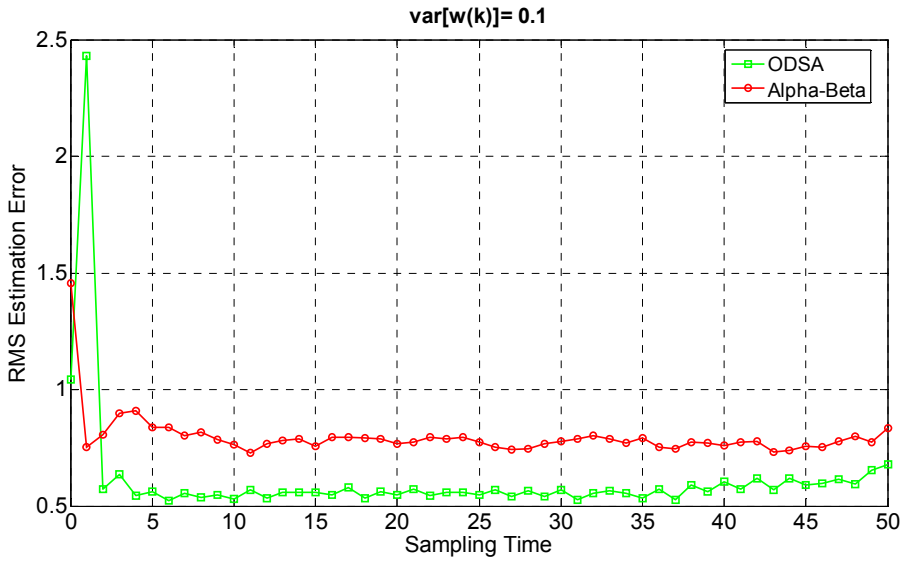

Figure-54: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of w(k) equals 0.1

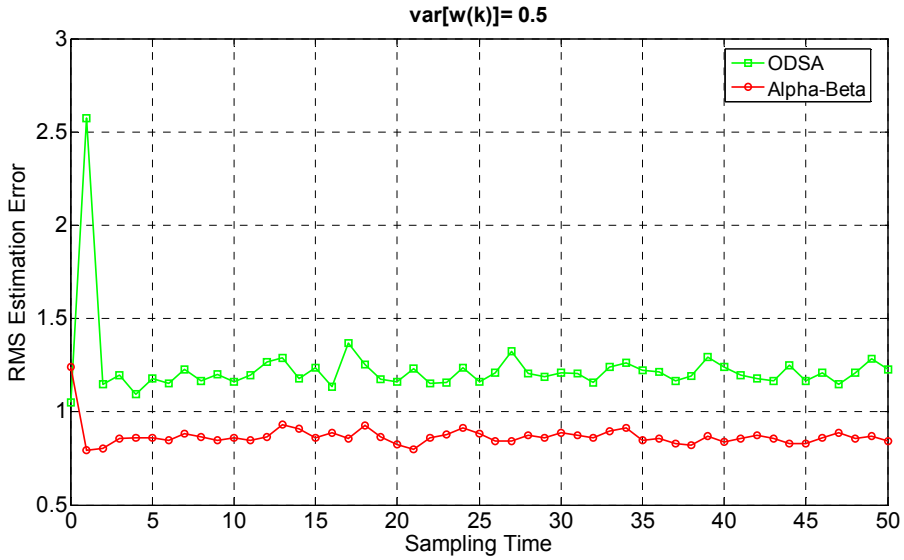

Figure-55: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of w(k) equals 0.5

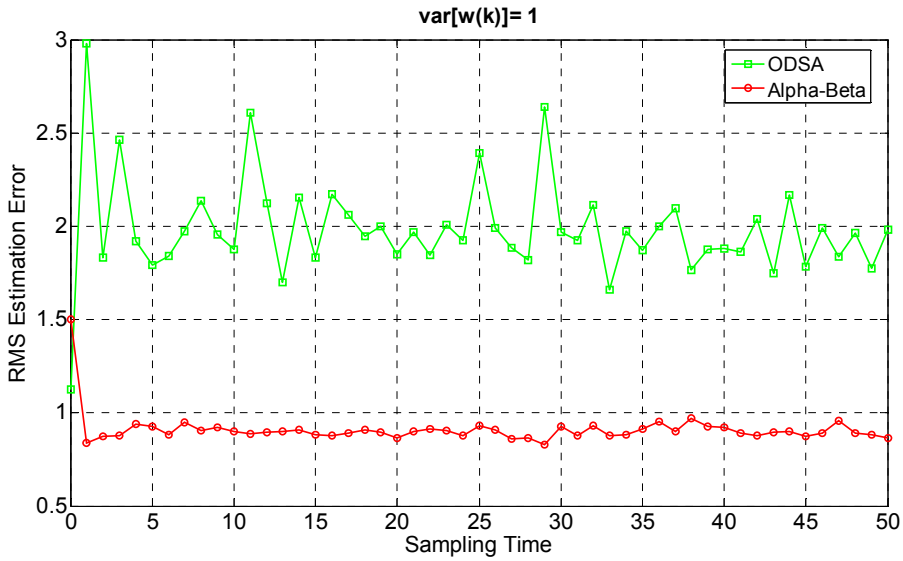

Figure-56: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of w(k) equals 1

Table-31 : Average values of all RMS estimation errors from k=0 to k=L for both ODSA and Alpha-Beta Filter Algorithm as the variance of w(k) changes

Variance of w(k)	0.01	0.1	0.5	1
ODSA Average of all RMS Errors	0.4257	0.6134	1.2293	1.9821
Alpha-Beta Average of All RMS Errors	0.8476	0.7957	0.8680	0.9101

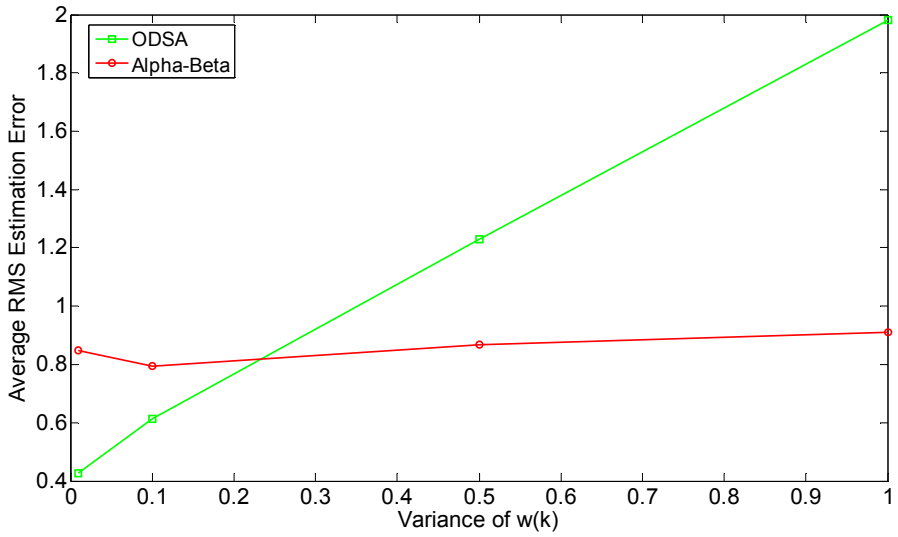

Figure-57: Estimation performance comparison of ODSA and Alpha-Beta Filter Algorithm as the variance of w(k) changes

Comment:

As seen from Figure 57, Alpha-Beta Filter RMS estimation error is not affected very much by increasing disturbance noise variance, whereas ODSA shows a linear increase in RMS estimation error. It can be observed from Figures 53 and 54 that ODSA shows a better estimation performance than Alpha-Beta Filter Algorithm. From Figures 55 and 56, after the disturbance noise variance value of 0.1, Alpha-Beta Filter Algorithm begins to show a better estimation performance than ODSA.

7.3 Comparision as the Measurement Noise Variance Changes

In this section, effects of the measurement noise variance on both ODSA and Alpha-Beta Filter Algorithm are investigated.

Parameters used in ODSA are:

total number of samples, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0 ,E[v(0)] = 0 , E[r(k)] = 0

Parameters used in Alpha-Beta Filter Algorithm are:

total number of samples, L = 50

measurement interval = 1

variances :var[x(0)] = 1, var[v(k)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(k)] = 0 , E[r(k)] = 0

In this simulation, results are obtained for four different values of correlated measurement noise variance, which are [0.01 0.1 1 3] after 500 executions for each algorithm.

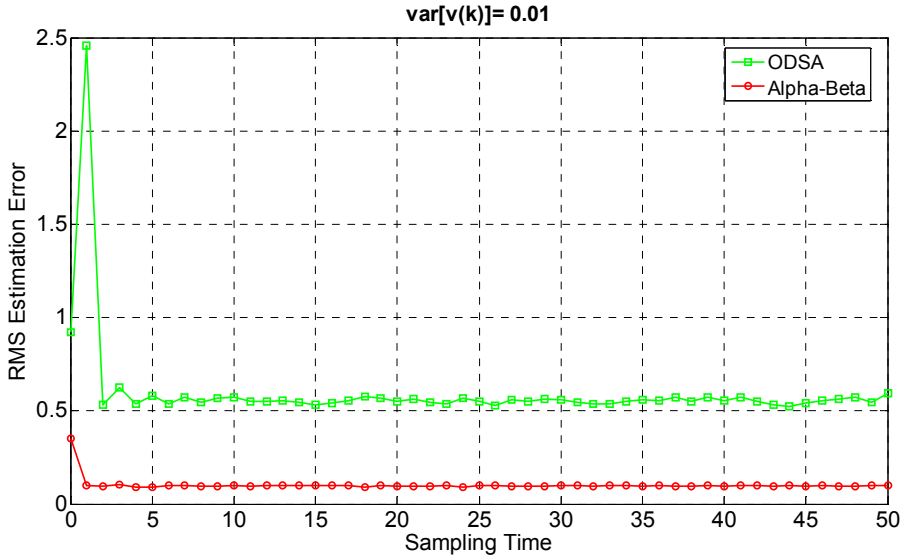

Figure-58: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of v(k) equals 0.01

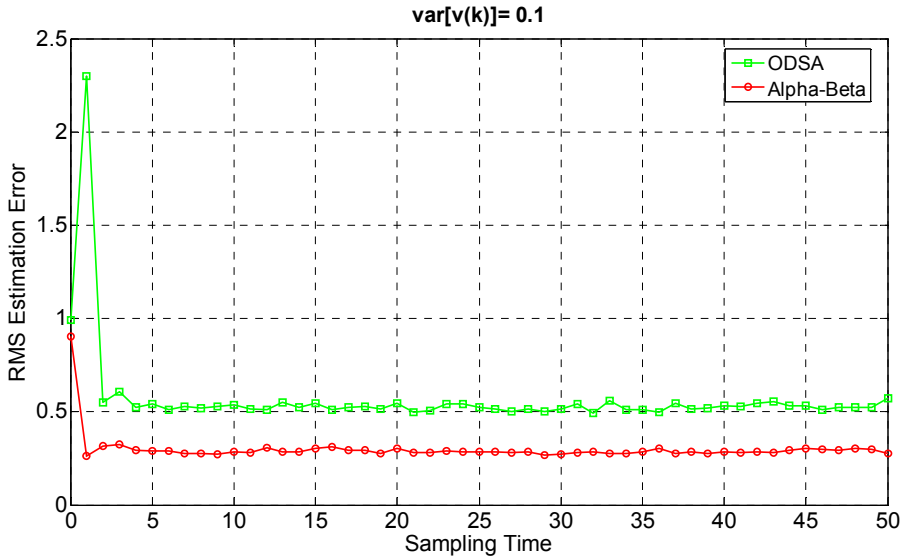

Figure-59: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of v(k) equals 0.1

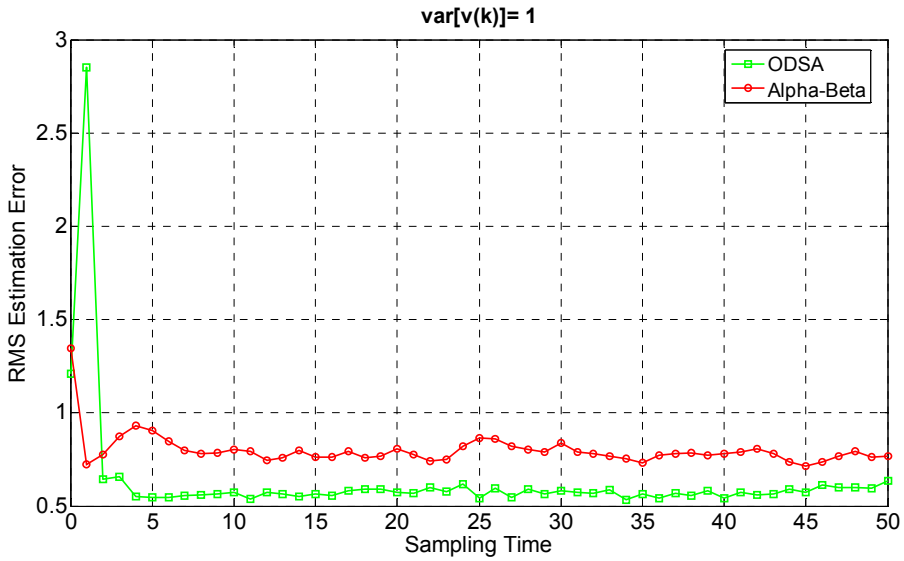

Figure-60: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of v(k) equals 1

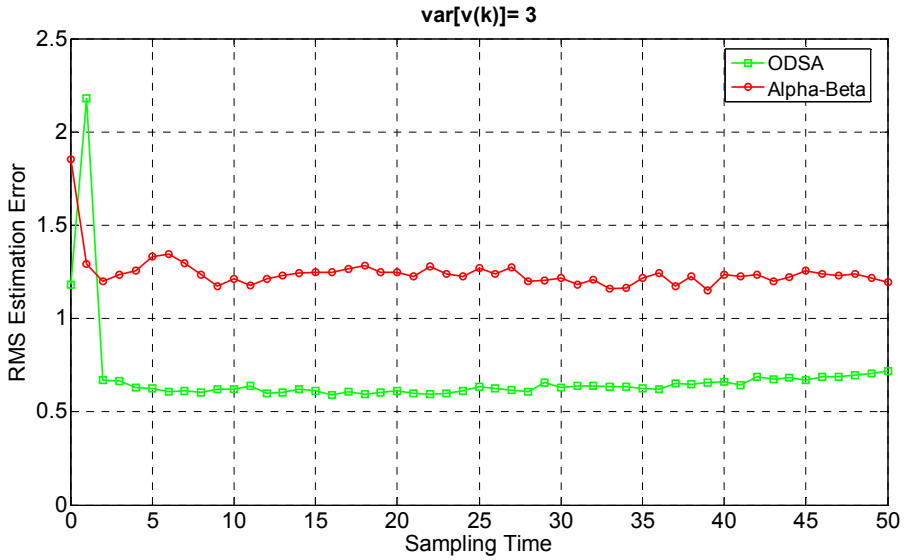

Figure-61: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of vk) equals 3

Table-32 : Average values of all RMS estimation errors from k=0 to k=L for both ODSA and Alpha-Beta Filter Algorithm as the variance of v(k) changes

Variance of v(k)	0.01	0.1	1	3
ODSA Average of all RMS Errors	0.5990	0.5709	0.6319	0.6768
Alpha-Beta Average of All RMS Errors	0.1020	0.2995	0.7988	1.2436

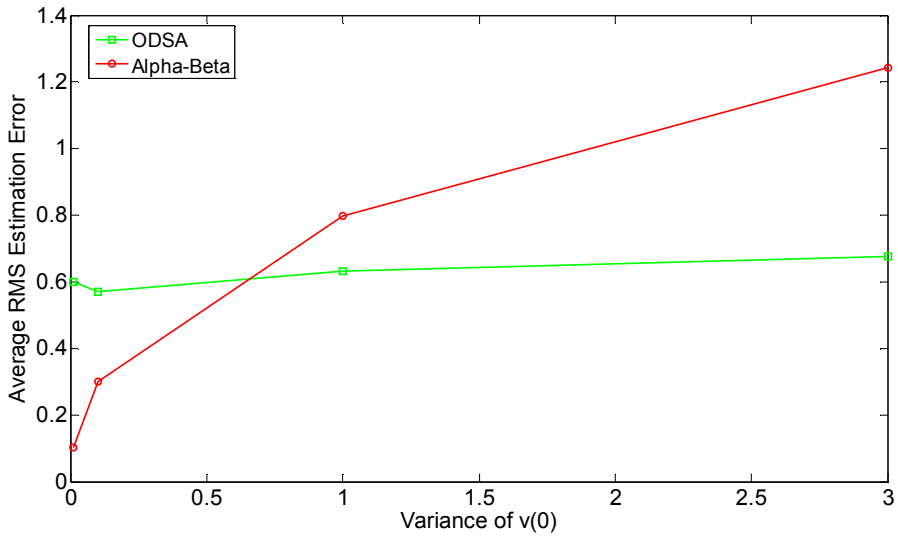

Figure-62: Estimation performance comparison of ODSA and Alpha-Beta Filter Algorithm as the variance of v(k) changes

Comment:

From Figure 62 and Table 32, it can be said that increasing the variance of v(k) causes the performance of both algorithms to get worse, but Figure 55 shows that ODSA shows smaller error increase whereas Alpha-Beta Filter Algorithm shows a dramatic increase on the estimation error. It can be observed from Figures 58 and 59 that Alpha-Beta Filter Algorithm shows a better estimation performance

than ODSA. From Figures 60 and 61, after the disturbance noise variance value of 1, ODSA begins to show a better estimation performance than Alpha-Beta Filter Algorithm.

7.4 Comparision as the Initial State Variance Changes

In this section, effects of the initial state variance on both ODSA and Alpha-Beta Filter Algorithm are investigated.

Parameters used in ODSA are:

total number of samples, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of max states = 100

number of max v(0) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)]= 0, E[w(k)]= 0, ,E[v(0)] = 0, E[r(k)] = 0

Parameters used in Alpha-Beta Filter Algorithm are:

total number of samples, L = 50

measurement interval = 1

variances : var[x(0)] = 1, var[v(k)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)]= 0, E[v(k)] = 0, E[r(k)] = 0

In this simulation, results are obtained for four different values of initial state variance, which are [0.01 0.1 1 3] , after 500 executions for each algorithm.

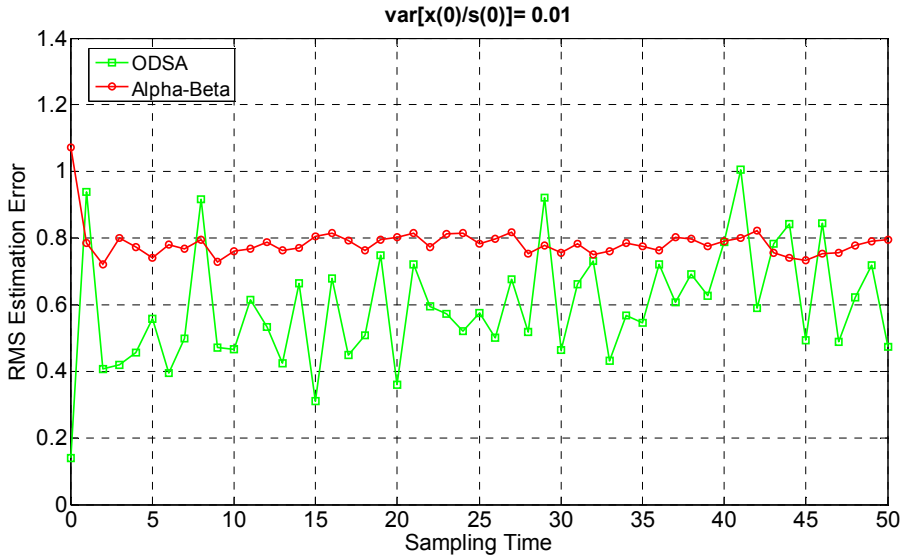

Figure-63: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of x(0)/s(0) equals 0.01

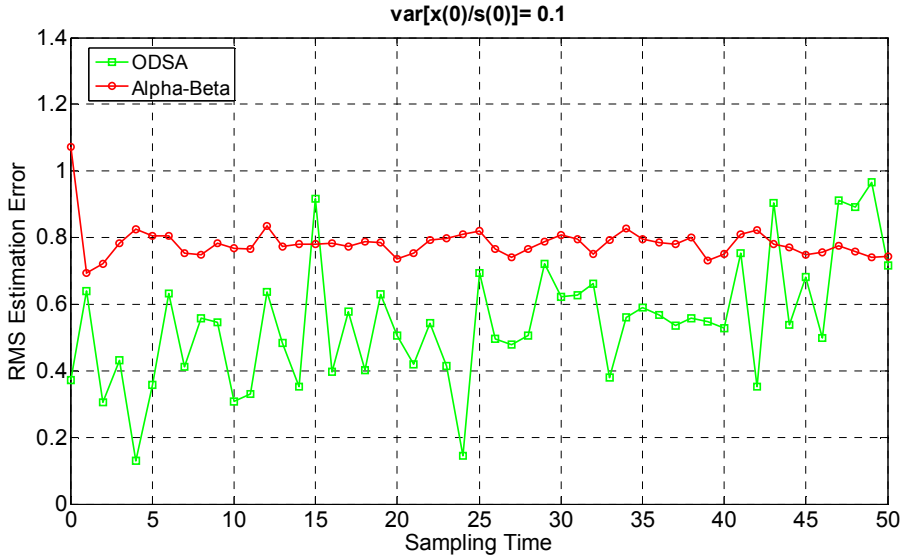

Figure-64: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of x(0)/s(0) equals 0.1

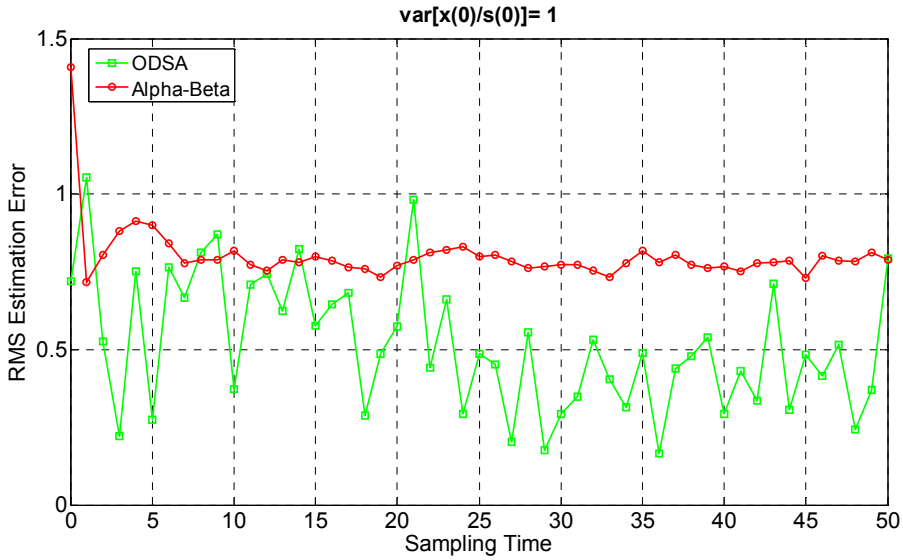

Figure-65: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of x(0)/s(0) equals 1

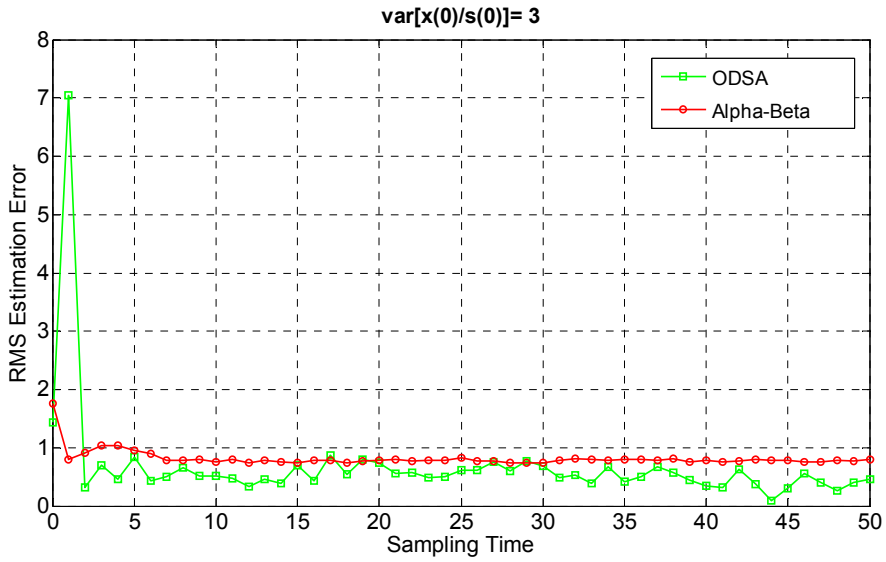

Figure-66: RMS estimation error versus sampling time for both ODSA and Alpha-Beta Filter Algorithm when the variance of x(0)/s(0) equals 0.5

Table-33 : Average values of all RMS estimation errors from k=0 to k=L for both ODSA and Alpha-Beta Filter Algorithm as variance of x(0)/s(0) changes

Variance of x(0)/s(0)	0.01	0.1	1	3
ODSA Average of all RMS Errors	0.5931	0.5440	0.5163	0.6700
Alpha-Beta Average of All RMS Errors	0.7841	0.7826	0.8005	0.8163

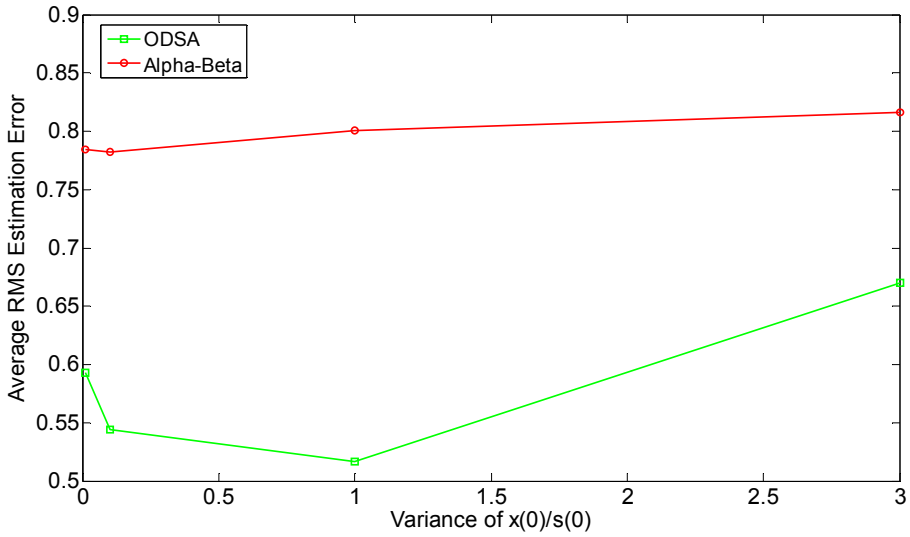

Figure-67: Estimation performance comparison of ODSA and Alpha-Beta Filter Algorithm as the variance of x(0)/s(0) changes

Comment:

From Table 33, it can be said that increasing the variance of x(0)/s(0) does not affect the performance of both algorithms very much but Figure 67 shows that ODSA estimation errors are smaller than the Alpha-Beta Filter Algorithm estimation errors. It can be observed from Figures 63, 64, 65 and 66 that ODSA shows a better estimation performance than Alpha-Beta Filter Algorithm.

7.5 Comparision as the Sampling Number Changes

In this section, effects of the sampling number on both ODSA and Alpha-Beta Filter Algorithm are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a= 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Parameters used in Alpha-Beta Filter Algorithm are:

total sampling time, L = 50

measurement interval = 1

variances :var[x(0)] = 1, var[v(k)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)]= 0, E[v(k)] = 0

In this simulation, results are obtained for five different values of sampling time, which are [10 20 30 40 50] after 500 executions for each algorithm.

Table-34 : Average values of all RMS estimation errors from k=0 to k=L for the linear model of both ODSA and Alpha-Beta Filter Algorithm as the sampling number changes

Sampling Time	10	20	30	40	50
ODSA Average of all RMS Errors	0.7907	0.7092	0.6625	0.6343	0.6100
Alpha-Beta Average of All RMS Errors	1.0403	1.2935	0.6490	0.6577	0.9639

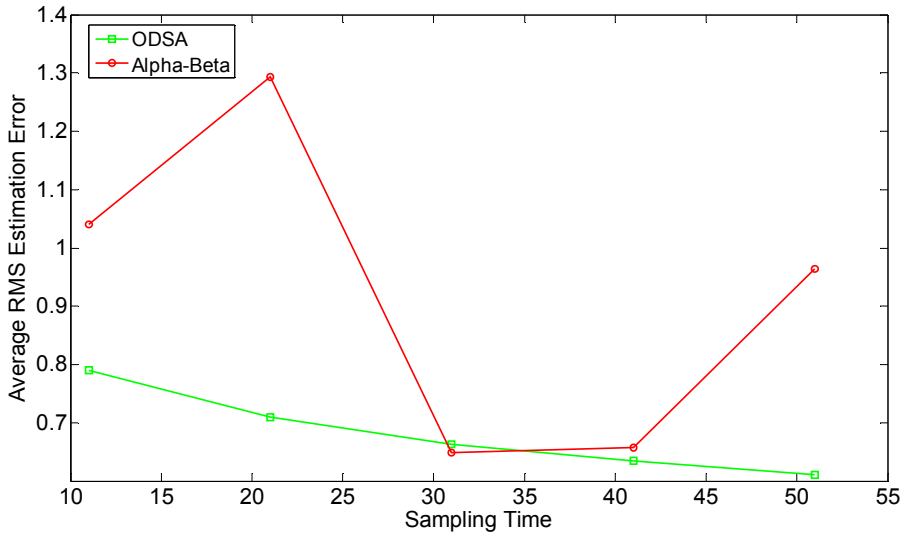

Figure-68: Estimation performance comparison of ODSA and Alpha-Beta Filter Algorithm as the sampling number changes

Comment:

Table 34 and Figure 68 show that increase on sampling time causes the performance of ODSA to increase. On the other hand, Alpha-Beta Filter Algorithm shows an unstable behaviour, firstly its performance begins to decrease, then to increase and after a sampling time value of 40, its performance begins to decrease again. It can be said that ODSA shows a better estimation performance than Alpha-Beta Filter Algorithm in this simulation.

7.6 Run-Time Comparision of ODSA And Alpha-Beta Filter Algorithm

In this section, run time comparision of ODSA and Alpha-Beta Filter Algorithm will be done as the sampling number changes. To get approximately the same estimation error performance for both algorithms, the parameters below are used:

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a= 0.1

gate size = 1

number of maximum states = 50

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 3, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)] = 0

Parameters used in Alpha-Beta Filter Algorithm are:

total sampling time, L = 50

measurement interval = 1

variances :var[x(0)] = 1, var[v(k)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)]= 0, E[v(k)] = 0, E[r(k)] = 0

In this simulation, results are obtained for five different values of sampling time, which are [10 20 30 40 50].

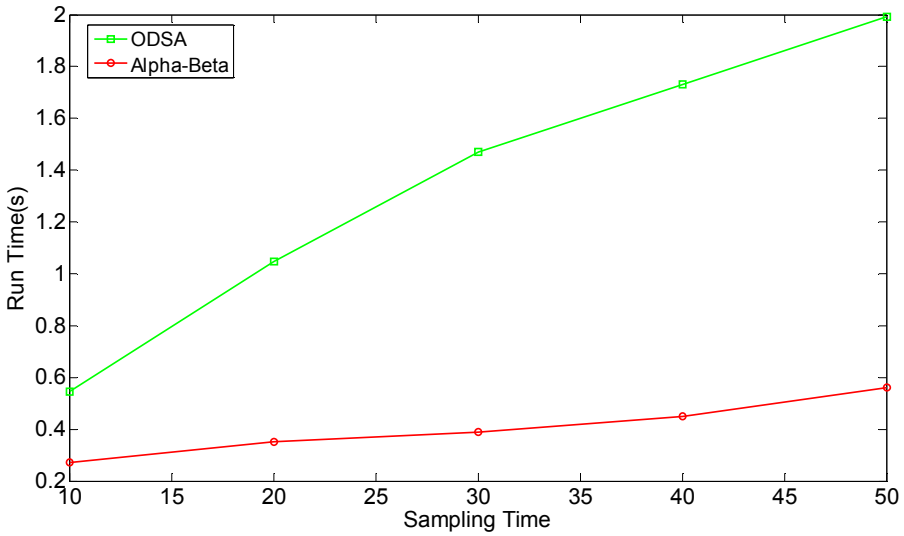

Figure-69: Run time comparison of ODSA and Alpha-Beta Filter Algorithm filter Algorithm as the sampling number changes

Table-35 : Run Time of both ODSA and Alpha-Beta Filter Algorithm as the sampling number changes

Sampling Time	10	20	30	40	50
ODSA Run Time(s)	0.5460	1.0470	1.4690	1.7320	1.9910
Alpha-Beta Run Time(s)	0.2700	0.3500	0.3900	0.4500	0.5600

Comment: From Figure 69 and Table 35, it can be observed that Alpha-Beta Filter Algorithm is faster than ODSA. For both algorithms, run time consumption increases as the sampling time increases, but Alpha-Beta Filter Algorithm shows smaller linear increase, whereas ODSA shows a larger linear increase. This is not surprising, since Alpha-Beta Filter has far less computational work than ODSA.

CHAPTER 8

EXTENDED KALMAN FILTER (EKF) WITH CORRELATED MEASUREMENT NOISE

The Kalman filter addresses the general problem of estimating the state of a discrete-time controlled process that is governed by a *linear* stochastic difference equation. For nonlinear stochastic motion and/or measurement equations, *extended Kalman filter (EKF)* is developed which linearizes the equations about the current mean and covariance [20].

8.1 Models and Assumptions [20]

The non-linear stochastic motion equation with a state vector $x \in R^n$ is

$$x(k) = f(x(k-1), u(k), w(k-1)) \tag{8.1}$$

with a measurement $z \in R^m$

$$z(k) = h(x(k), v(k)) \tag{8.2}$$

where the random variables $w(k)$ and $v(k)$ represent the disturbance and measurement noise respectively. The *non-linear* function f in the motion equation relates the state at the previous time step $k-1$ to the state at the current time step k. It includes parameters of any driving function $u(k)$ and the zero-mean process

noise *w(k)*. The *non-linear* function *h* in the measurement equation relates the state *x(k)* to the measurement *z(k)*.

The set of EKF equations is shown below:

EKF time update equations:

$$\hat{x}(k|k-1) = f(\hat{x}(k-1), u(k), 0) \tag{8.3}$$

$$P(k|k-1) = A(k)P(k-1)A_k^{\ T} + W(k)Q(k-1)W^T(k) \tag{8.4}$$

EKF measurement update equations:

$$K(k) = P(k|k-1)H^T(k)\left(H(k)P(k|k-1)H^T(k) + V(k)R(k)V^T(k)\right)^{-1} \tag{8.5}$$

$$\hat{x}(k) = \hat{x}(k|k-1) + K(k)\left(z(k) - h(\hat{x}(k|k-1), 0)\right) \tag{8.6}$$

$$P(k) = \left(I - K(k)H(k)\right)P(k|k-1) \tag{8.7}$$

where

- *Q(k)* is the covariance matrix of the disturbance noise *w(k)*

$$Q(k) = E\left\{\left[w(k) - E(w(k))\right]\left[w(k) - E(w(k))\right]^T\right\} \tag{8.8}$$

- *R(k)* is the covariance matrix of the measurement noise *v(k)*

$$R(k) = E\left\{\left[v(k) - E(v(k))\right]\left[v(k) - E(v(k))\right]^T\right\} \tag{8.9}$$

- *P(k)* is the state estimation error covariance

$$P(k) = E\left\{\left[\hat{x}(k) - x(k)\right]\left[\hat{x}(k) - x(k)\right]^T\right\} \tag{8.10}$$

- A is the Jacobian matrix of partial derivatives of f with respect to x:

$$A_{ij} = \frac{\partial f_i}{\partial x_j}\left(\hat{x}(k-1), u(k), 0\right) \tag{8.11}$$

- W is the Jacobian matrix of partial derivatives of f with respect to w:

$$W_{ij} = \frac{\partial f_i}{\partial w_j}\left(\hat{x}(k-1), u(k), 0\right) \tag{8.12}$$

- H is the Jacobian matrix of partial derivatives of h with respect to x:

$$H_{ij} = \frac{\partial h_i}{\partial x_j}\left(\hat{x}(k|k-1), 0\right) \tag{8.13}$$

- V is the Jacobian matrix of partial derivatives of h with respect to v:

$$V_{ij} = \frac{\partial h_i}{\partial v_j}\left(\hat{x}(k|k-1), 0\right) \tag{8.14}$$

Figure 70 shows the complete operation of EKF.

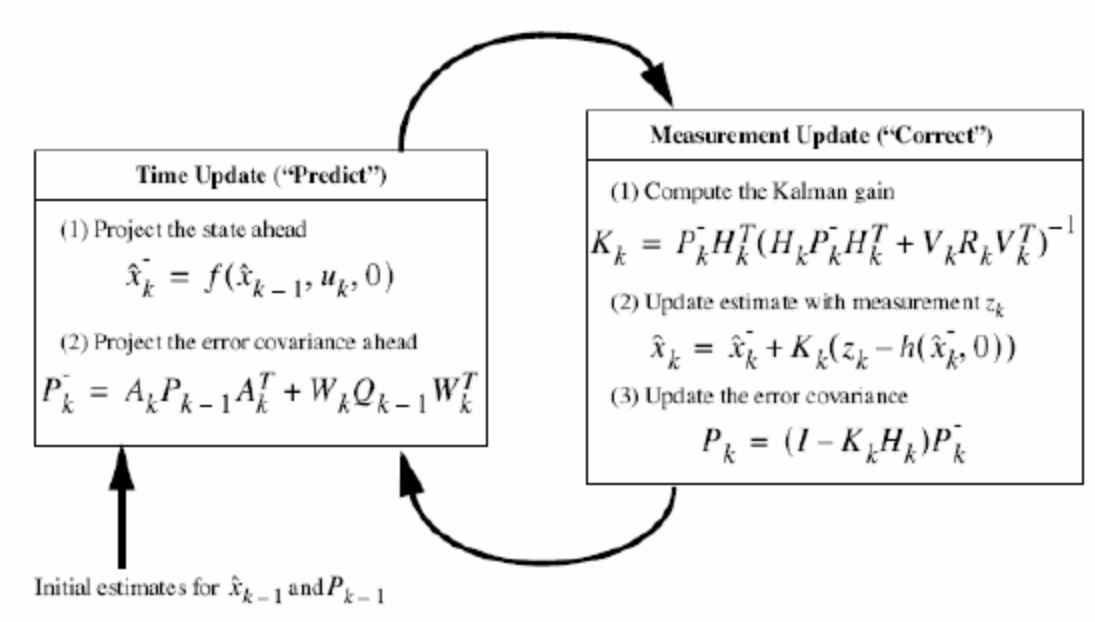

Figure-70: A complete picture of the operation of the EKF

Note that in Figure 70, $\hat{x}_k^- = \hat{x}(k|k-1)$ and $P_k^- = P(k|k-1)$.

8.2 Application of EKF for Correlated Measurement Noise Model

Let us take nonlinear target motion and measurement models as below:

$$\textit{Motion model} \quad : \quad x(k+1) = f(x(k)) + w(k) \tag{8.15}$$

$$\textit{Measurement model} \quad : \quad z(k) = h(x(k)) + v(k)$$

where f and h are nonlinear functions of $x(k)$. In measurement model, correlated measurement noise, $v(k)$, is modelled as:

$$v(k) = a\, v(k-1) + r(k) \tag{8.16}$$

where a is the correlation coefficient ($0 \le a \le 1$), and $r(k)$ is a zero-mean white Gaussian noise, with variance: $\sigma_r^2 = (1-a^2)\sigma_v^2$, where σ_v^2 is the variance of the measurement noise $v(k)$.

As mentioned in previous sections, EKF linearizes the equations about the current mean and covariance. If we think the linearized equations at each sampling time as below:

$$\begin{aligned} &\textit{Motion model} \quad : \quad x(k+1) = A\,x(k) + w(k) \\ &\textit{Measurement model} \quad : \quad z(k) = H\,x(k) + v(k) \end{aligned} \tag{8.17}$$

We can easily apply decorrelation method [3] to handle the correlated measurement noise effect at each sampling time.

To decorrelate the measurement noise, a new measurement $y(k)$, called *"artificial measurement"*, is generated by using the measurement given in the measurement model of Eq. (8.17) as below:

$$\begin{aligned} y(k) &= z(k) - a\,z(k-1) = H\,x(k) + v(k) - a\left(H\;x(k-1) + v(k-1)\right) \\ y(k) &= H\,x(k) - aH\,x(k-1) + \underbrace{v(k) - a\,v(k-1)}_{r(k)} \end{aligned} \tag{8.18}$$

From the motion model of Eq.(8.17), $x(k-1) = A^{-1}x(k) - A^{-1}w(k-1)$, if $x(k-1)$ is inserted in Eq.(8.18), the equation below is obtained:

$$\begin{aligned} y(k) &= H\,x(k) - a\,HA^{-1}x(k) + \underbrace{a\,H\,A^{-1}w(k-1) + r(k)}_{\eta(k)} \\ \eta(k) &= a\,HA^{-1}w(k-1) + r(k) \end{aligned} \tag{8.19}$$

In practical applications, the first term of right-hand side in Eq.(8.19) is usually small and can be neglected without degrading much performance [7]. So we have:

$$\eta(k) \approx r(k)$$
$$y(k) = \left(H - a\,H\,A^{-1}\right)x(k) + r(k) \tag{8.20}$$
$$y(k) = (1 - a\,A^{-1})\,H\,x(k) + r(k)$$

Motion and measurement models at each sampling time after decorrelation will be:

$$\textit{Motion model} \quad : \quad x(k+1) = A\,x(k) + w(k) \tag{8.21}$$
$$\textit{Measurement model} \quad : \quad z(k) = (1 - a\,A^{-1})\,H\,x(k) + v(k)$$

We obtained the motion and measurement models given in Eq.(8.21), so we can apply EKF. Simulation results of comparision of ODSA and EKF are given in Chapter 9.

CHAPTER 9

COMPARISION OF ODSA WITH EKF

In this chapter, ODSA with correlated measurement noise (by using the proposed method of treating correlation effect as interference explained in section 3.2) and EKF with correlated measurement noise (by using the decorrelation method explained in section 8.2) will be compared.

In order to compare these algorithms, two nonlinear system models will be used for both algorithms.

As a result of performed simulations, RMS Estimation Error versus Sampling Time graphs are given in figures to compare the performance of both ODSA and EKF for the same parameter values. RMS Estimation error for a given sampling time, k, is calculated as below:

$$\text{RMS Error}(k) = \frac{\sqrt{\sum_{i=1}^{N} (X_{ik} - \widetilde{X}_{ik})^2}}{N} \quad k=0,1,\ldots, L \qquad (9.1)$$

where RMS Error(k) is the RMS Error for sampling time k, N is the total execution number, X_{ik} is the real target state at sampling time k for the i^{th} execution, $\widetilde{X}_{ik}$ is the estimated target state at sampling time k for the i^{th} execution and L is the total sampling time.

For each simulation, RMS Estimation Error versus Sampling Time graphs acquired from ODSA and EKF Algorithm are plotted on the same figure. To increase the comprehension, tables are given at the end of the figures. In these tables, for each parameter value, Average of all RMS Errors obtained from the graphs in the figures are given. Average of all RMS Errors is calculated as below:

$$\text{Average of all RMS Errors} = \frac{\sum_{k=0}^{L} RMS\ Error(k)}{L} \tag{9.2}$$

9.1 Simulations for Nonlinear System Model 1

In this section, nonlinear models given below will be used in the simulations for both ODSA and EKF:

Motion model : x (k+1) = exp (- x (k)) + w (k) (9.3)

Measurement model : z (k) = cos(x (k)) + v (k)

where correlated measurement noise, v (k), is modelled as:

v(k)=a v (k-1)+r (k).

To use the EKF for the nonlinear models given in Eq. (9.3), Jacobian matrices: *A, W, H* and *V* are calculated as below:

- *A* , Jacobian matrix of partial derivatives of *f* with respect to *x*:

$$A_{ij} = \frac{\partial f_i}{\partial x_j}\left(\hat{x}(k-1), u(k), 0\right) = -\exp(-x(k)) \tag{9.4}$$

- W is the Jacobian matrix of partial derivatives of f with respect to w:

$$W_{ij} = \frac{\partial f_i}{\partial w_j}\left(\hat{x}(k-1), u(k), 0\right) = 1 \tag{9.5}$$

- H is the Jacobian matrix of partial derivatives of h with respect to x:

$$H_{ij} = \frac{\partial h_i}{\partial x_j}\left(\hat{x}(k|k-1), 0\right) = -\sin(x(k)) \tag{9.6}$$

- V is the Jacobian matrix of partial derivatives of h with respect to v:

$$V_{ij} = \frac{\partial h_i}{\partial v_j}\left(\hat{x}(k|k-1), 0\right) = 1 \tag{9.7}$$

9.1.1 Comparision as the Correlation Coefficient Changes

In this section, effects of the correlation coefficient on both ODSA and EKF are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1

expected values : E[x(0)]= 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)]= 0

Parameters used in EKF are:

total sampling time, L = 50

variances : var[x(0)] = 1, var[v(k)]=1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]= 0

9.1.1.1 Simulation 1

In this simulation, results are obtained for five different values of correlation coefficient, which are [0.01 0.03 0.05 0.07 0.09], after 500 executionss for each algorithm. Disturbance noise variance, **var[w(k)], is taken as 0.1**

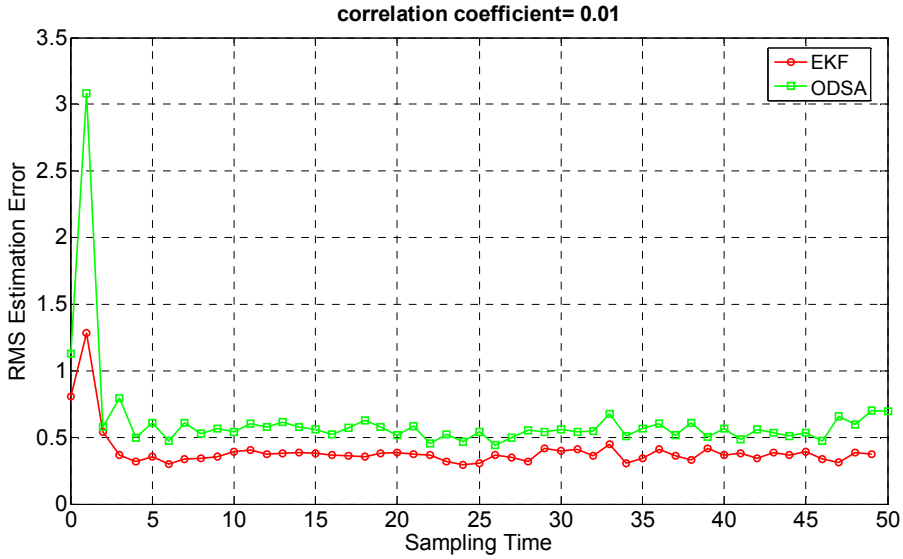

Figure-71: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.01 and disturbance noise variance equals 0.1

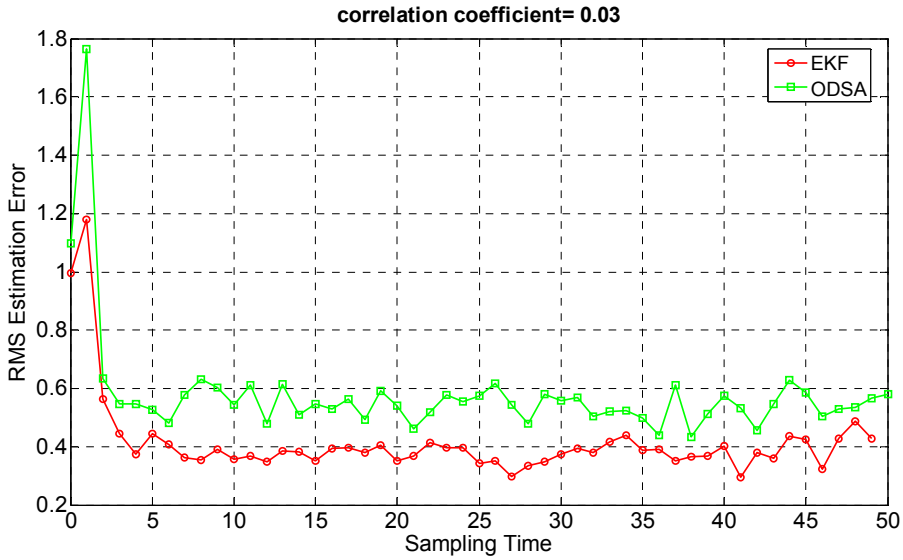

Figure-72: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.03 and disturbance noise variance equals 0.1

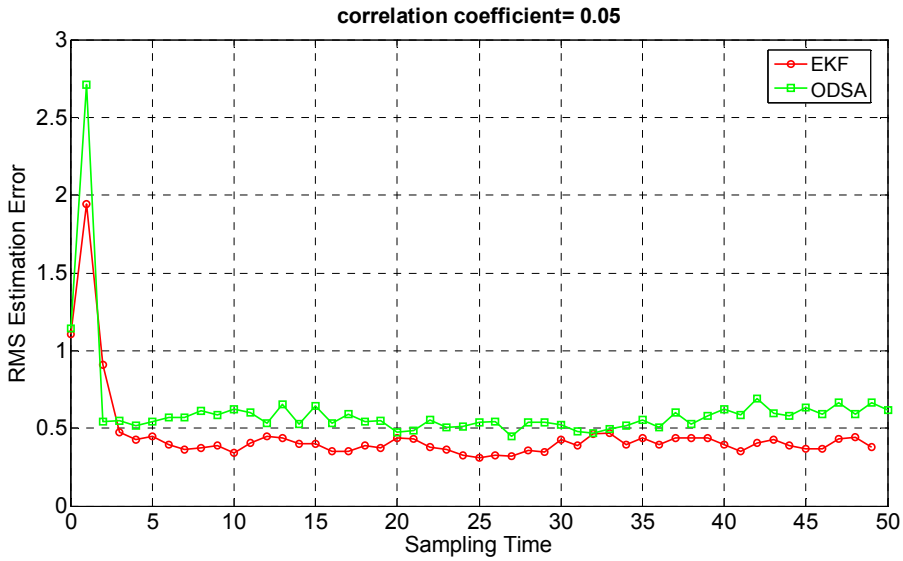

Figure-73: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.05 and disturbance noise variance equals 0.1

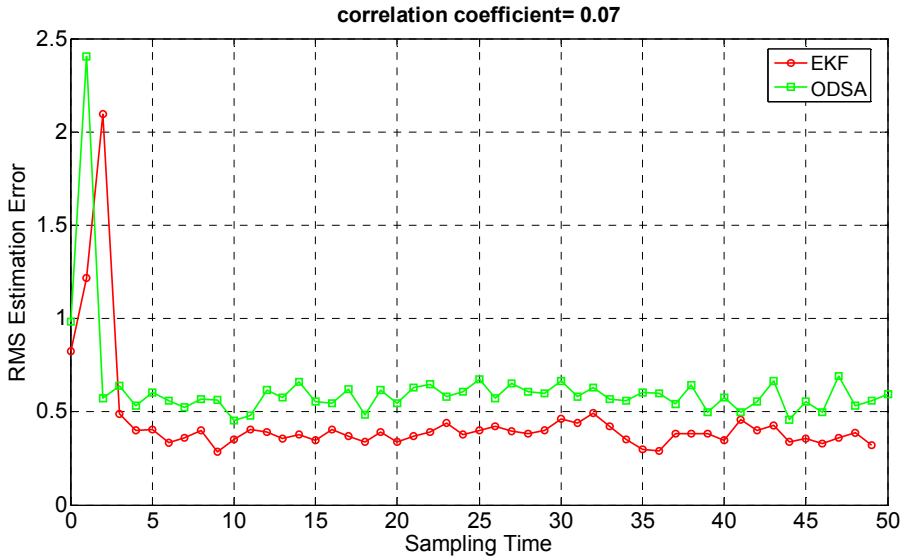

Figure-74: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.07 and disturbance noise variance equals 0.1

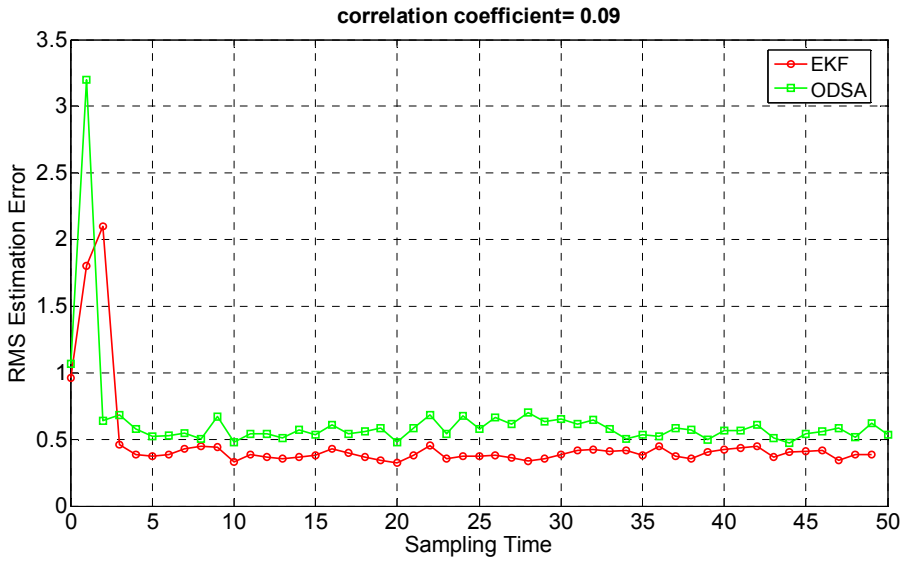

Figure-75: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.09 and disturbance noise variance equals 0.1

Table-36 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the correlalation coefficient changes and when the disturbance noise variance equals 0.1

Correlation Coefficient, a	0.01	0.03	0.05	0.07	0.09
EKF Average of all RMS Errors	0.3946	0.4143	0.4514	0.4410	0.4653
ODSA Average of All RMS Errors	0.6214	0.5793	0.6153	0.6224	0.6346

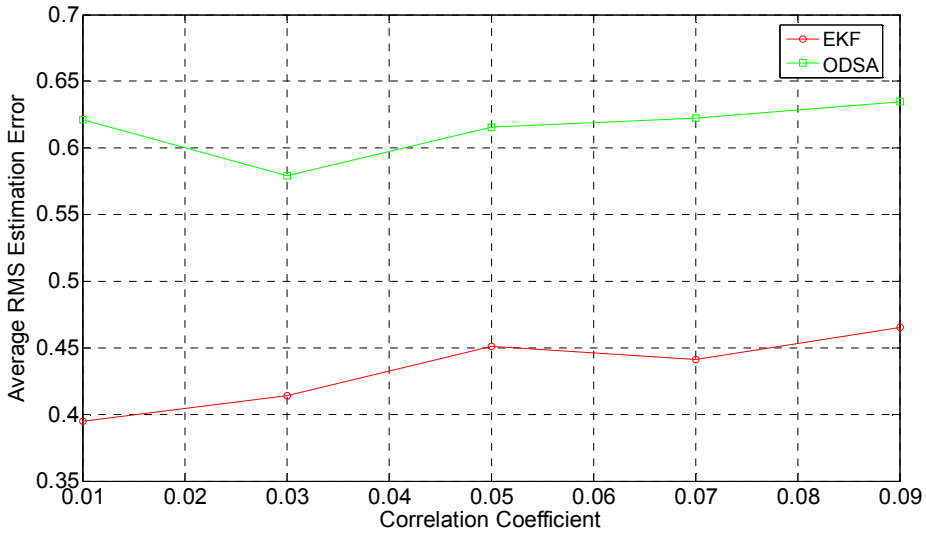

Figure-76: Estimation performance comparison of ODSA and EKF as the correlation coefficient changes and when the disturbance noise variance equals 0.1

Comment:

From Figures 71-75, Figure 76 and Table 36, it can observed that EKF shows a better estimation performance than ODSA in these simulations.

9.1.1.2 Simulation 2

Parameter values used in this simulation are same as the parameter values used in Simulation 1 given in section 9.1.1.1, except disturbance noise variance, **var[w(k)] is taken as 5**

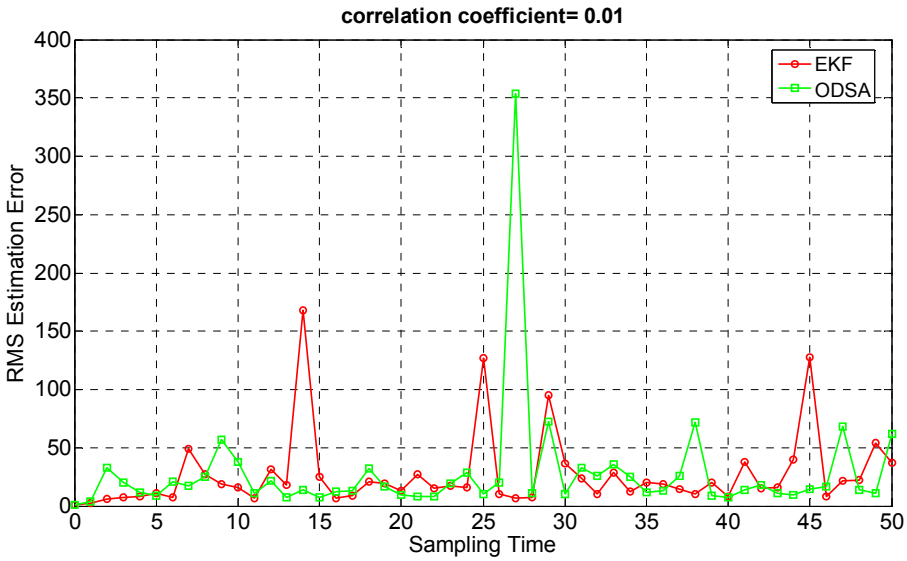

Figure-77: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.01 and disturbance noise variance equals 5

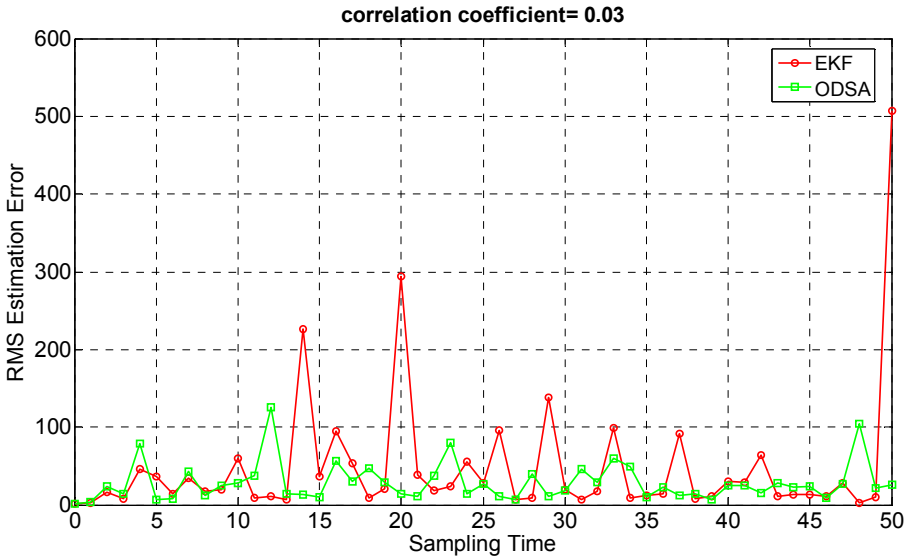

Figure-78: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.03 and disturbance noise variance equals 5

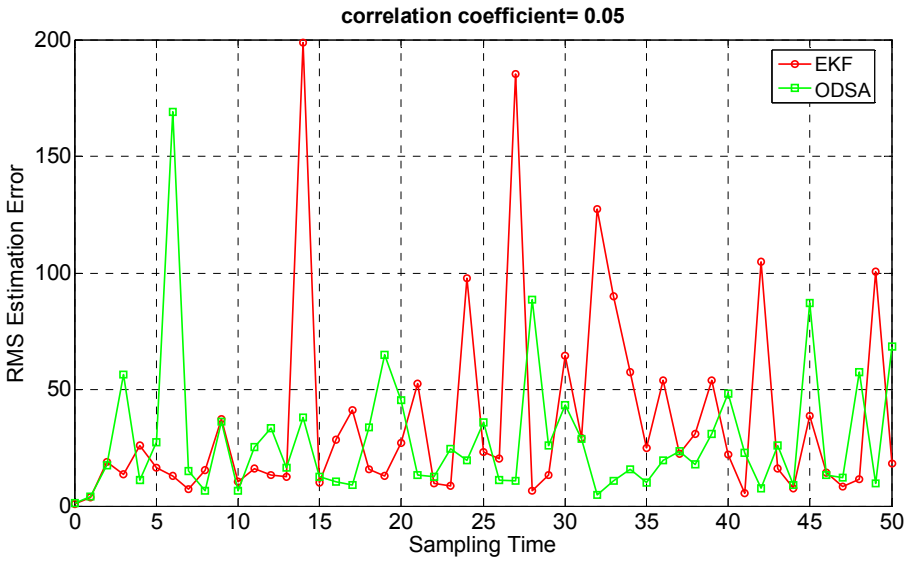

Figure-79: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.05 and disturbance noise variance equals 5

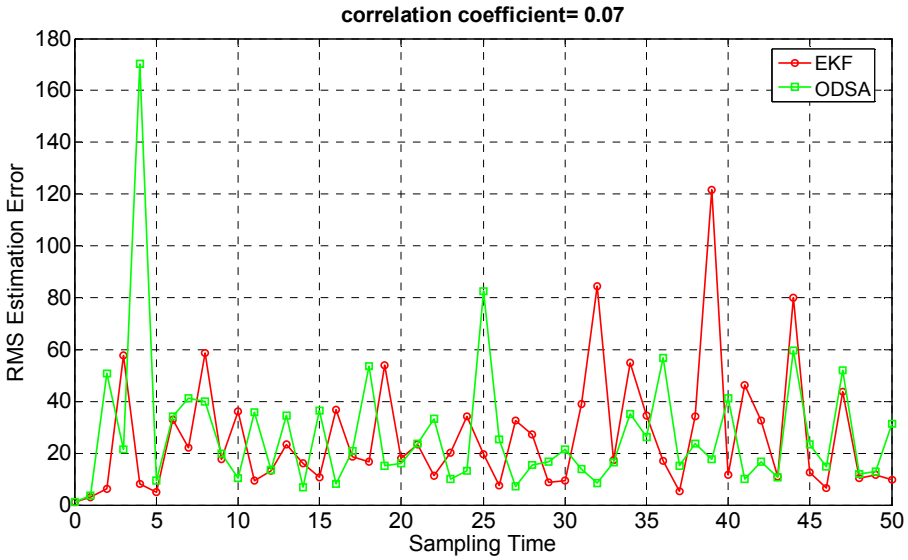

Figure-80: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.07 and disturbance noise variance equals 5

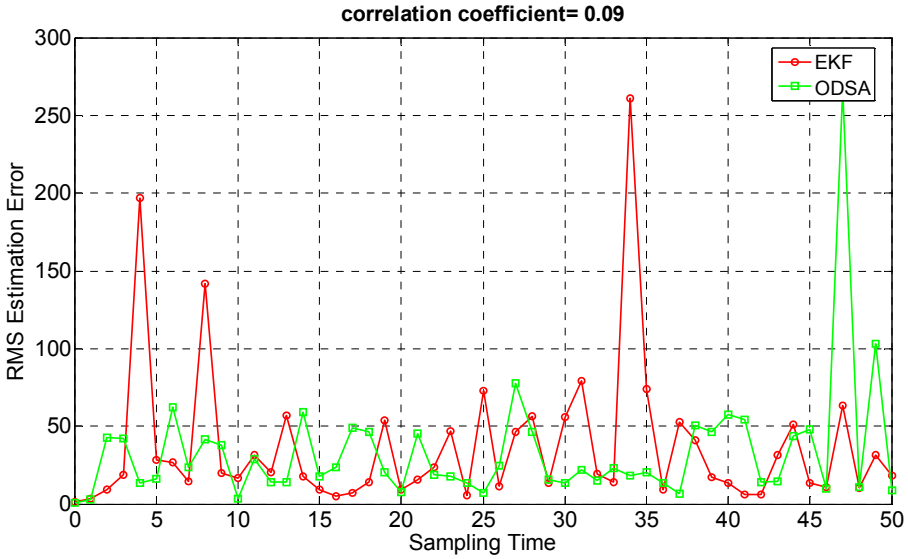

Figure-81: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.09 and disturbance noise variance equals 5

Table-37 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the correlation coefficient changes and when the disturbance noise variance equals 5

Correlation Coefficient, a	0.01	0.03	0.05	0.07	0.09
EKF Average of all RMS Errors	27.0235	47.8931	36.4352	26.3299	36.7627
ODSA Average of All RMS Errors	27.8222	28.5176	28.4242	27.2222	33.2657

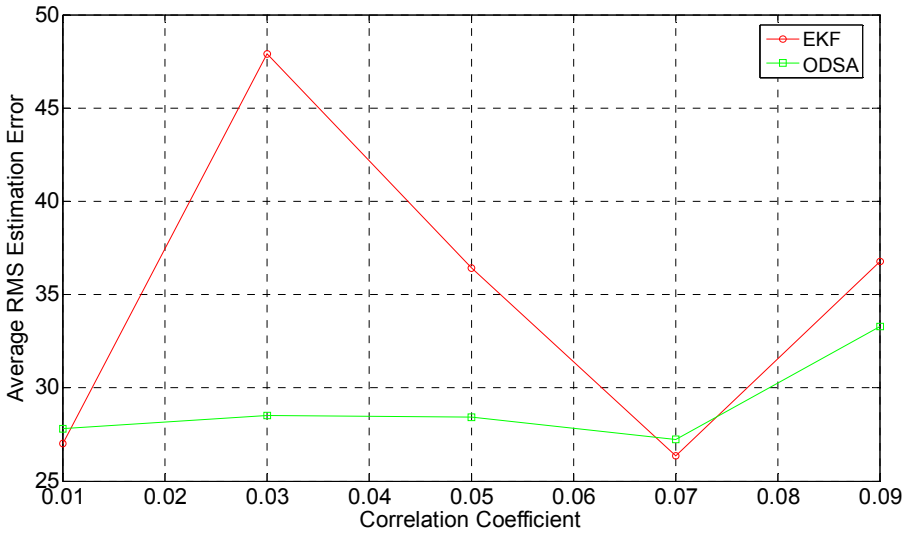

Figure-82: Estimation performance comparison of EKF and ODSA as the correlation coefficient changes and when the disturbance noise variance equals 5

Comment:

Figures 77-81, Figure 82 and Table 37 show that ODSA has a better estimation performance than EKF when the disturbance noise variance is increased to 5. This is because decorrelation method can be applied for small disturbance noise variance values for EKF. In fact, both algorithms are not applicable in this case since RMS estimation errors of the both algorithms are very large.

9.1.1.3 Simulation 3

Parameter values used in this simulation are same as the parameter values used in simulation 1 given in section 9.1.1.1, except correlation coefficient, a, varies as [0.1 0.3 0.5 0.7 0.9] and disturbance noise variance, **var[w(k)], is again taken as 0.1**

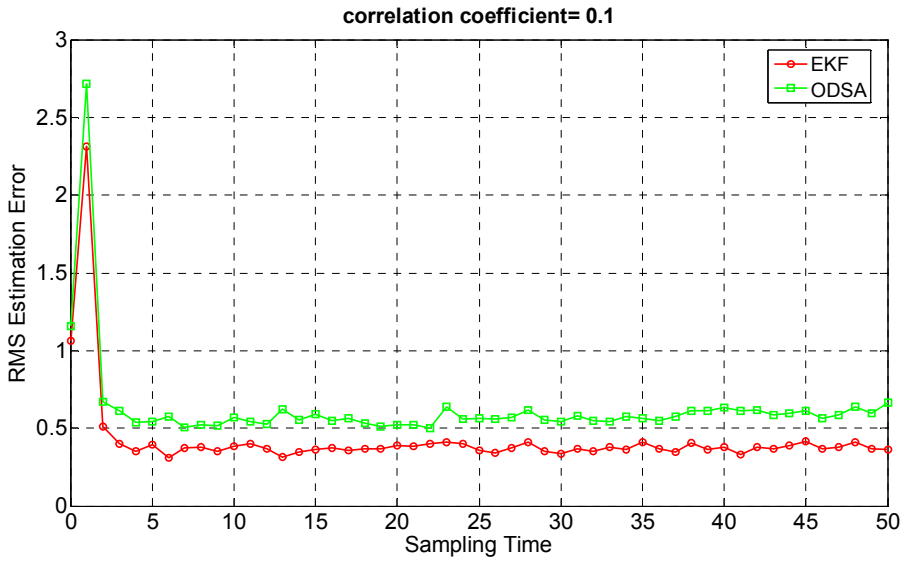

Figure-83: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.1 and disturbance noise variance equals 0.1

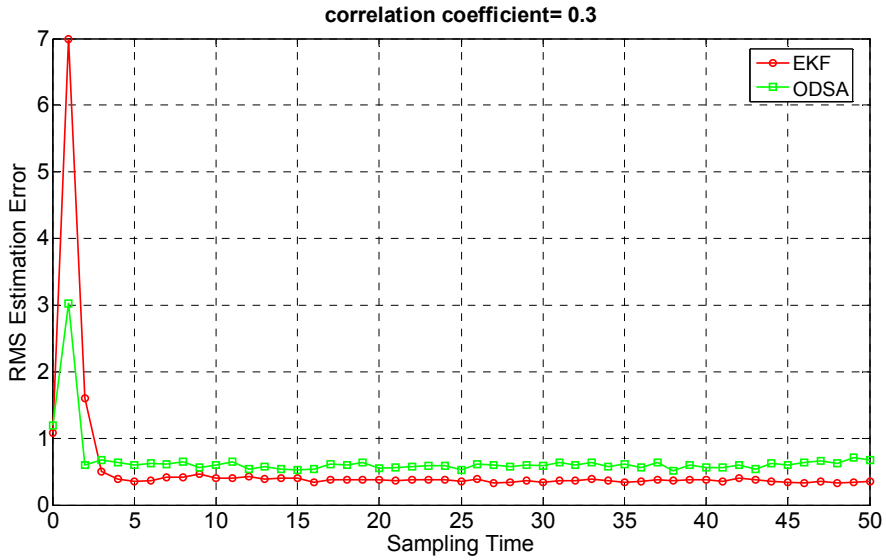

Figure-84: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.3 and disturbance noise variance equals 0.1

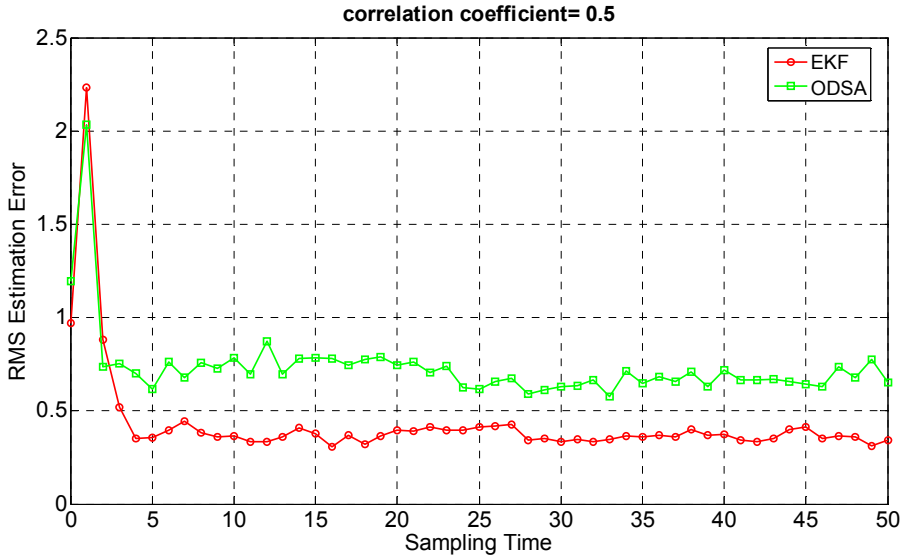

Figure-85: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.5 and disturbance noise variance equals 0.1

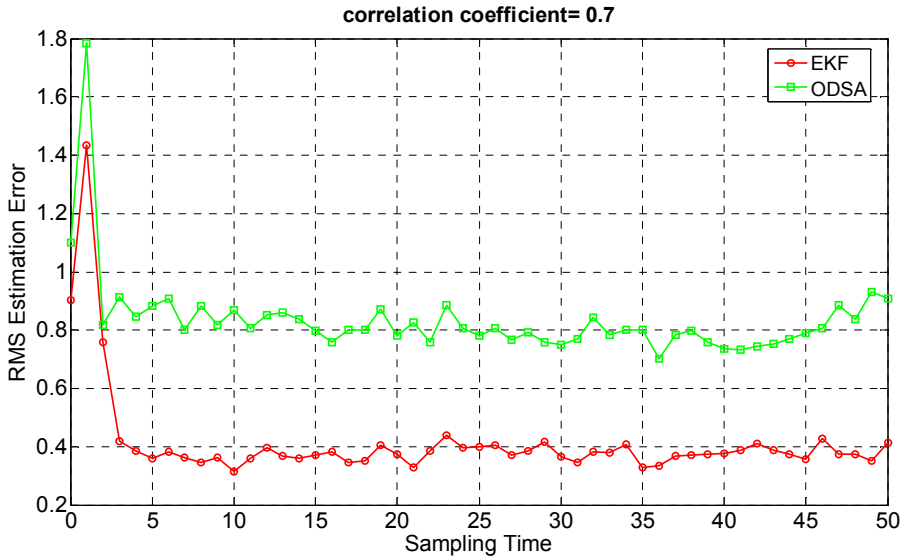

Figure-86: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.7 and disturbance noise variance equals 0.1

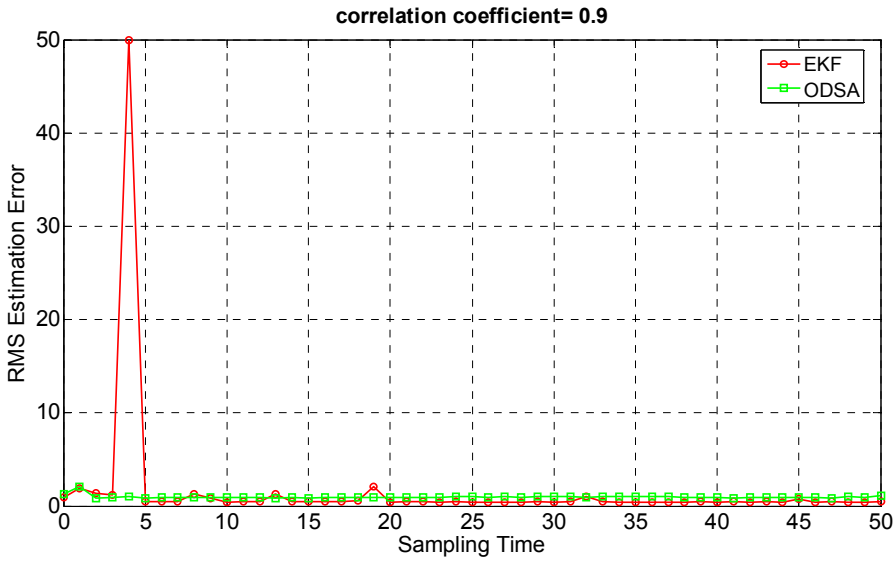

Figure-87: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.9 and disturbance noise variance equals 0.1

Table-38 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the correlation coefficient changes and when the disturbance noise variance equals 0.1

Correlation Coefficient, a	0.1	0.3	0.5	0.7	0.9
EKF Average of all RMS Errors	0.4269	0.5429	0.4292	0.4147	1.5721
ODSA Average of All RMS Errors	0.6265	0.6583	0.7331	0.8364	0.9650

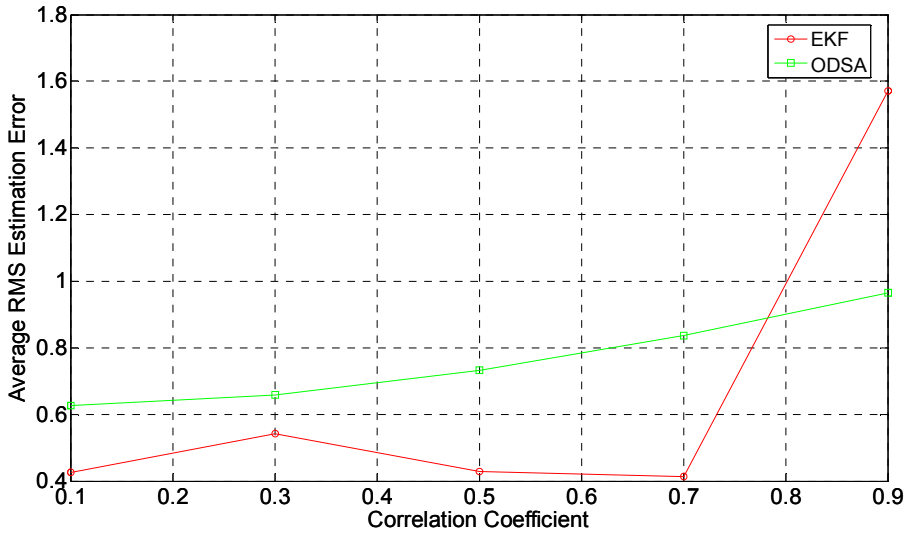

Figure-88: Estimation performance comparison of EKF and ODSA as the correlation coefficient changes and when the disturbance noise variance equals 0.1

Comment:

Figure 88 shows that EKF shows a better estimation performance than ODSA for the correlation coefficient values of 0.1, 0.3, 0.5 and 0.7 while ODSA shows a better performance than EKF when the correlation coefficient equals 0.9.

9.1.1.4 Simulation 4

Parameter values used in this simulation are same as the parameter values used in Simulation 1 given in section 9.1.1.1, except correlation coefficient, a, varies as [0.1 0.3 0.5 0.7 0.9] and disturbance noise variance, **var[w(k)], is taken as 1**

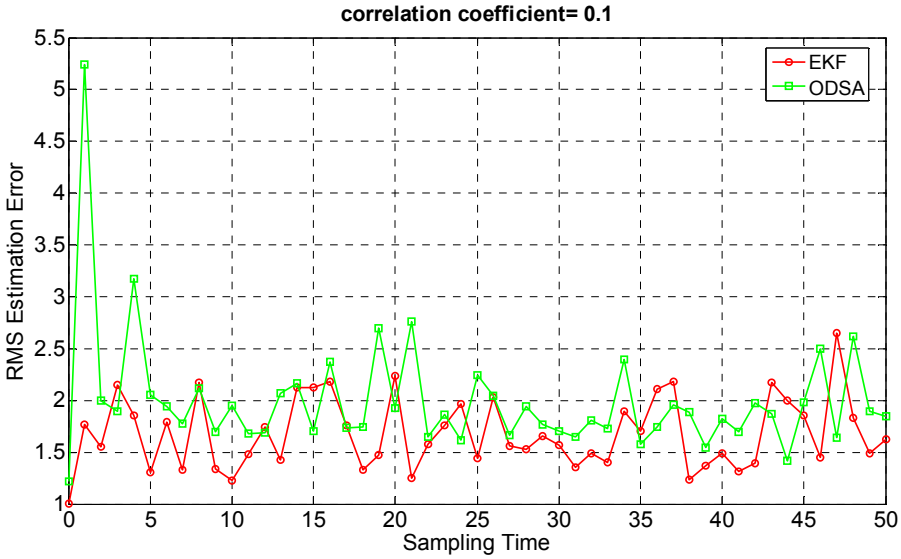

Figure-89: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.1 and disturbance noise variance equals 1

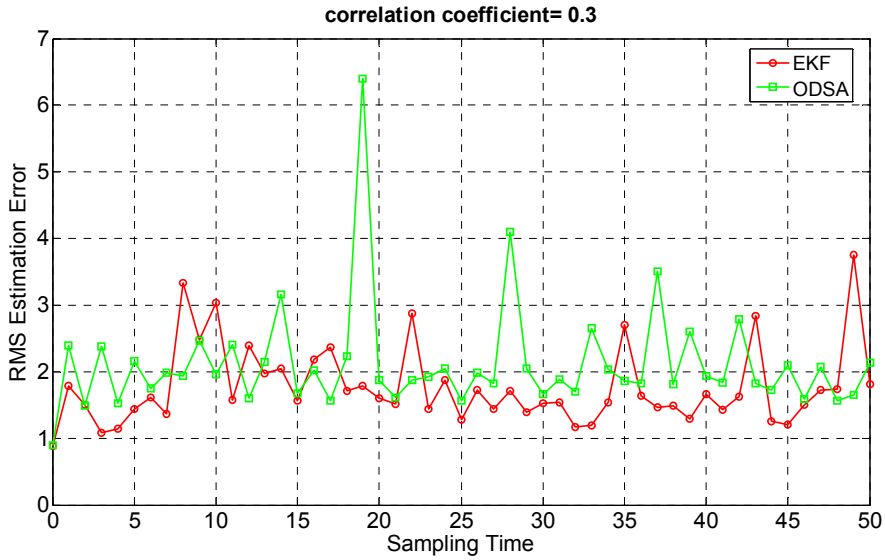

Figure-90: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.3 and disturbance noise variance equals 1

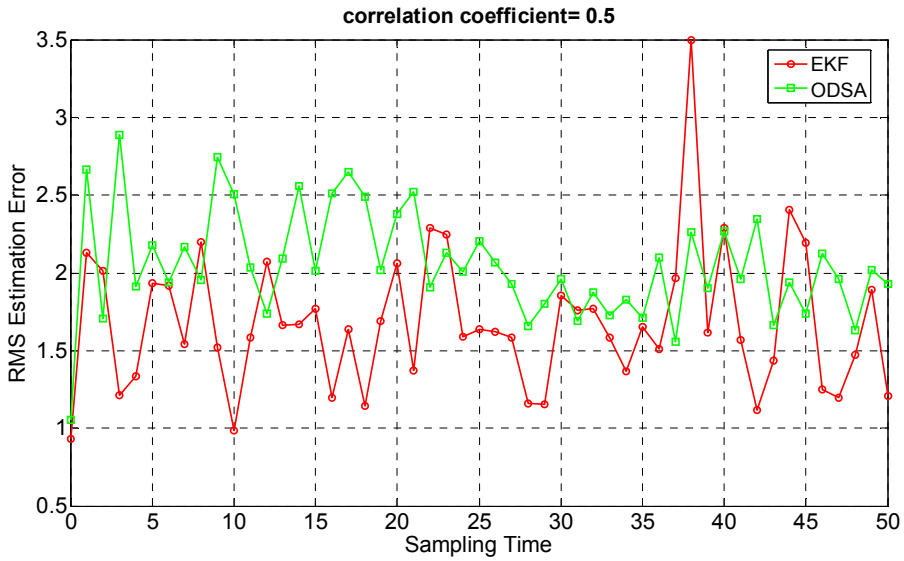

Figure-91: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.5 and disturbance noise variance equals 1

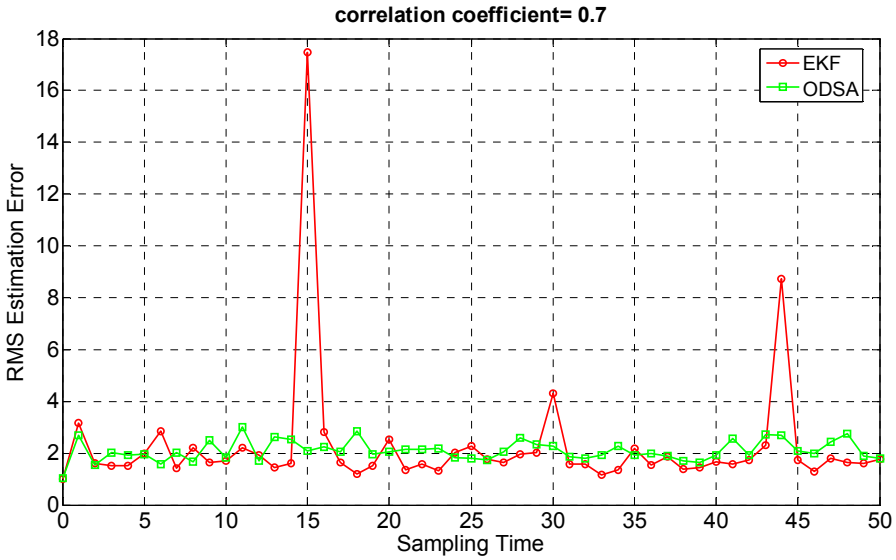

Figure-92: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.7 and disturbance noise variance equals 1

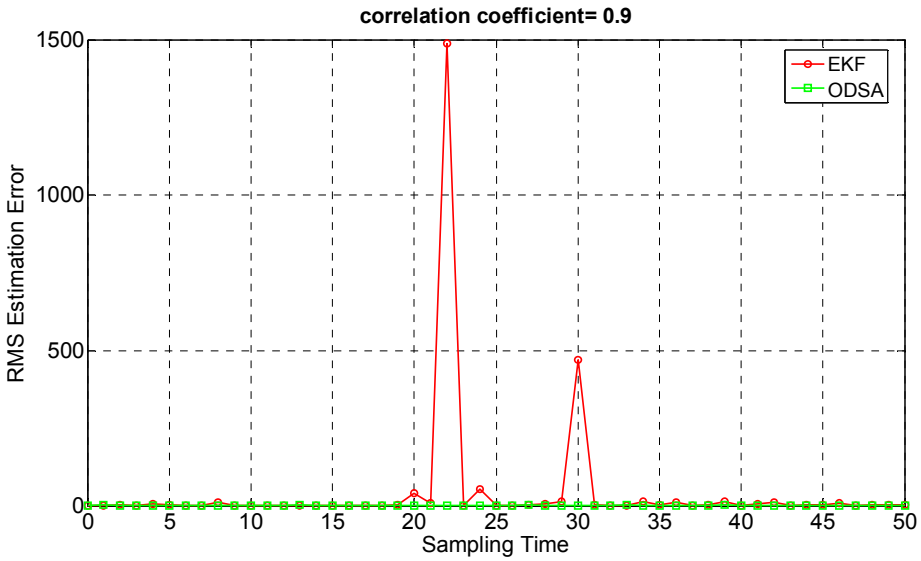

Figure-93: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.9 and disturbance noise variance equals 1

Table-39 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the correlation coefficient changes and disturbance variance value equals 1

Correlation Coefficient, a	0.1	0.3	0.5	0.7	0.9
EKF Average of all RMS Errors	1.6807	1.7685	1.6757	2.2559	44.3652
ODSA Average of All RMS Errors	1.9930	2.1129	2.0513	2.0901	2.2063

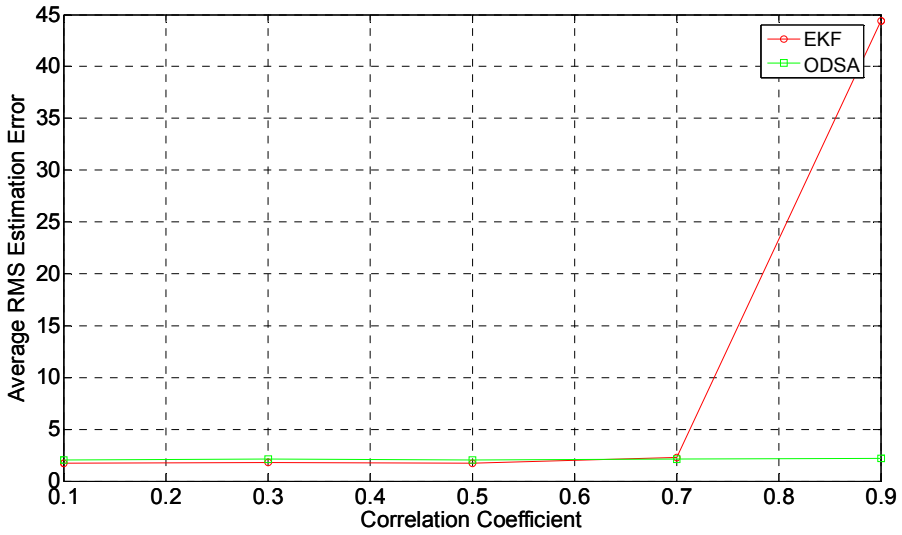

Figure-94: Estimation performance comparision of ODSA and EKF as the correlation coefficient changes and disturbance noise variance equals 1

Comment:

Figure 94 and Table 39 show that ODSA shows a better estimation performance than EKF. For correlation coefficient values of 0.1, 0.3, 0.5 and 0.7 the performance difference is not very much but after the correlation coefficient value of 0.9, EKF estimation error increases dramatically.

9.1.2 Comparision as the Disturbance Noise Variance Changes

In this section, effects of the disturbance noise variance on both ODSA and EKF are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)]=0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[x(0)] = 1, var[v(k)]=1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for five different values of disturbance noise variance, which are [0.1 1 3 5 7] after 500 executions for each algorithm.

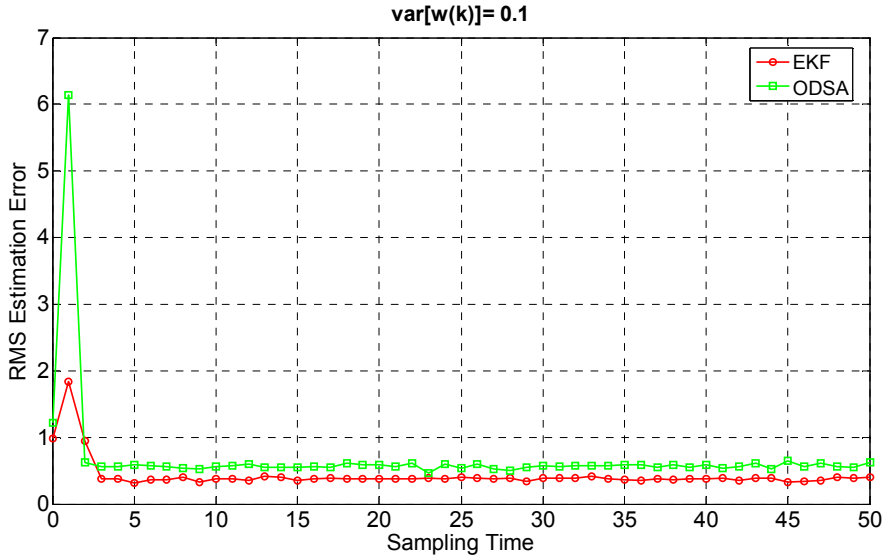

Figure-95: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 0.1

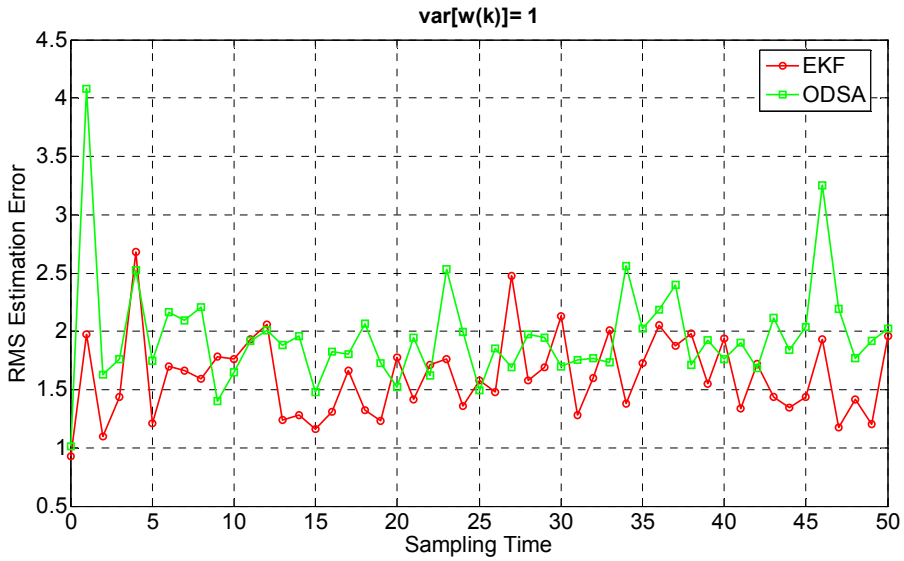

Figure-96: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 1

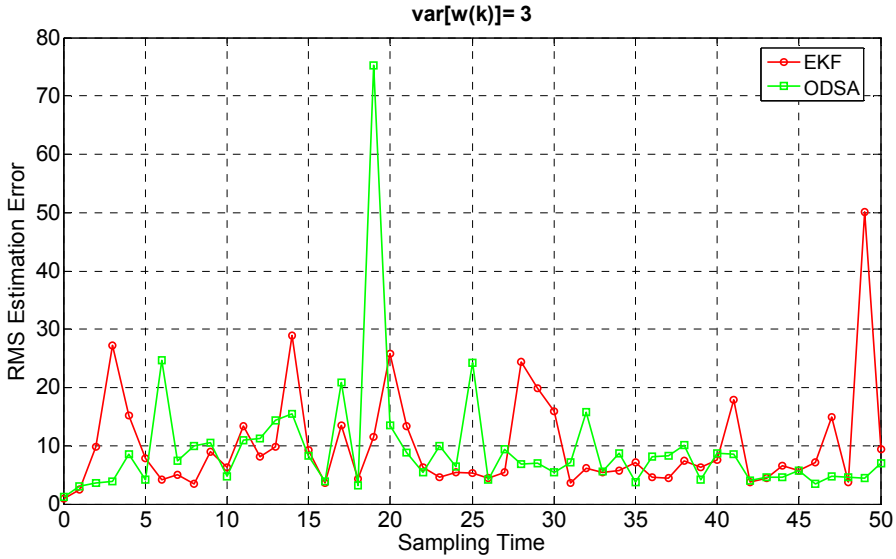

Figure-97: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 3

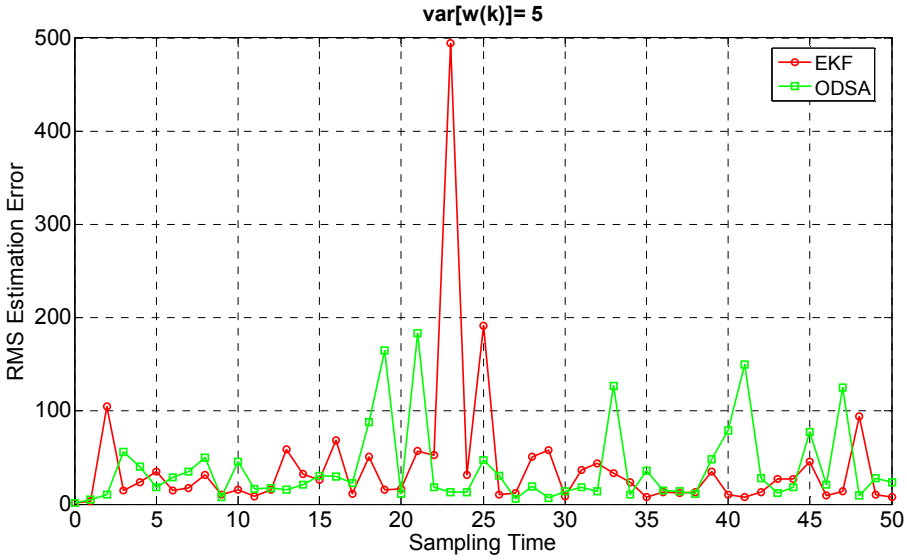

Figure-98: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 5

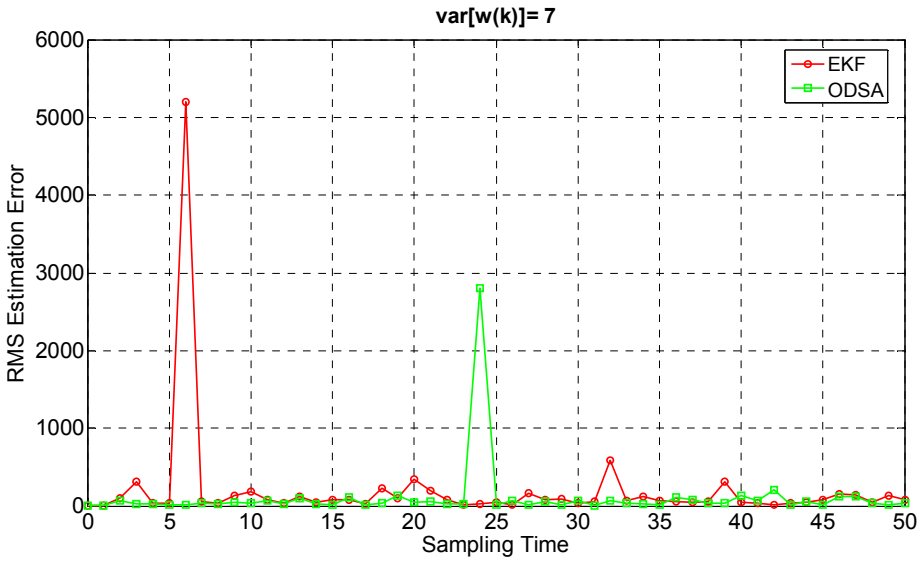

Figure-99: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 7

Table-40 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the variance of w(k) changes

Variance of w(k)	0.1	1	3	5	7
EKF Average of all RMS Errors	0.4272	1.6141	9.9372	39.6615	199.6869
ODSA Average of All RMS Errors	0.6909	1.9561	9.3778	37.7450	103.9703

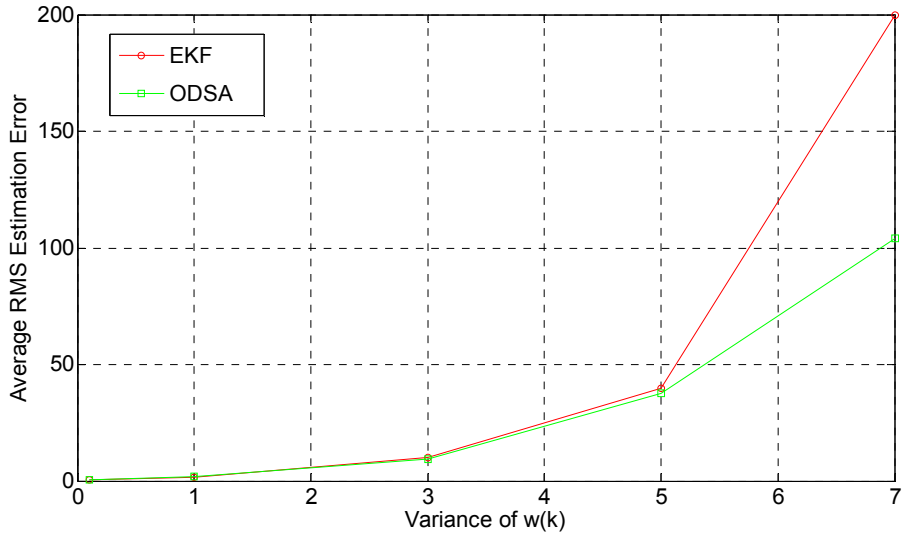

Figure-100: Estimation performance comparision of EKF and ODSA as the variance of w(k) changes

Comment:

It can be seen from Figure 100 and Table 40 that ODSA and EKF shows nearly close performance for disturbance variance values of 0.1, 1, 3 and 5, but for disturbance noise variance value of 7, EKF estimation error increases dramatically. This is because decorrelation method can be applied for small disturbance noise variance values for EKF. In fact, both algorithms are not applicable for high disturbance values of 5 and 7, since RMS estimation errors of the both algorithms are very large for these variance values.

9.1.3 Comparision as the Measurement Noise Variance Changes

In this section, effects of the measurement noise variance on both EKF and ODSA are investigated.

Parameters used in ODSA are:

total number of samples, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)] = 0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[x(0)] = 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

9.1.3.1 Simulation 1

In this simulation, results are obtained for four different values of correlated measurement noise variance, which are [0.1 1 3 5] after 500 executions for each algorithm. Disturbance noise variance, **var[w(k)], is taken as 0.1**

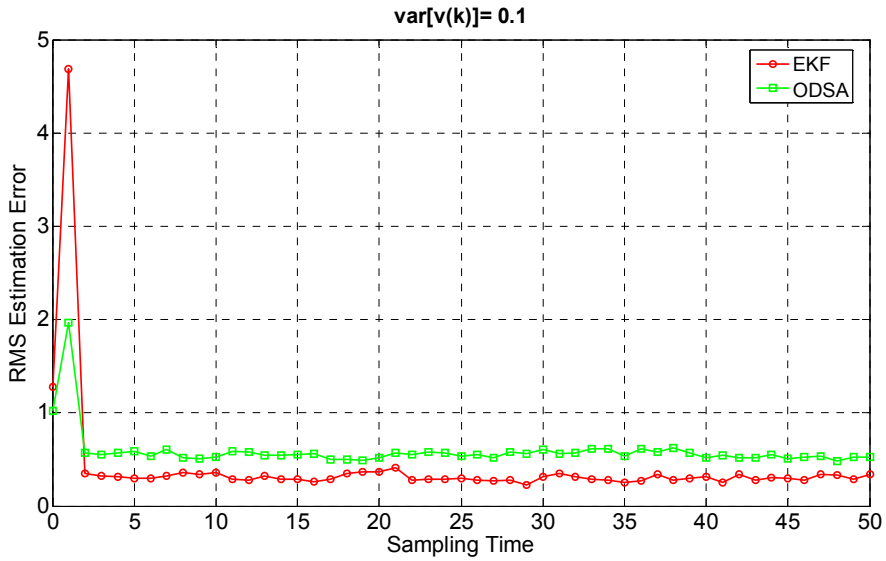

Figure-101: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 0.1 and disturbance noise variance equals 0.1

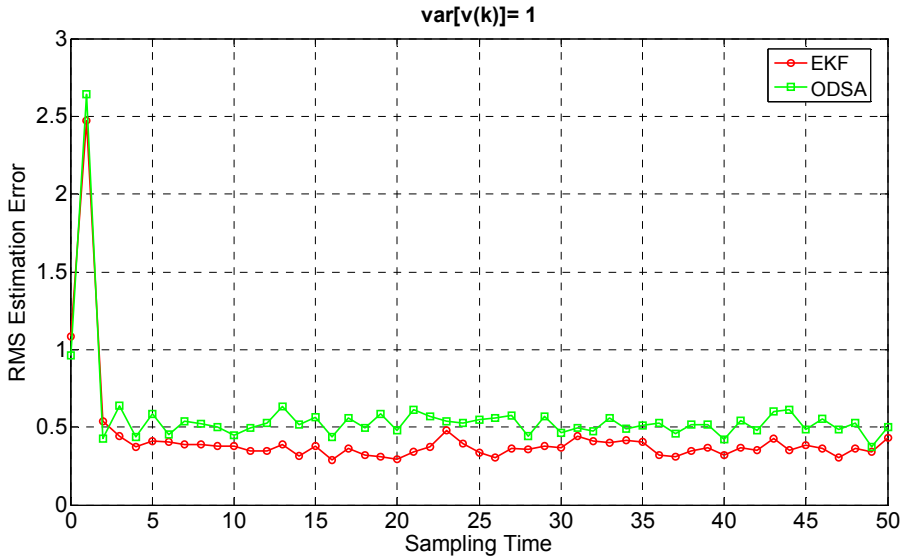

Figure-102: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 1 and disturbance noise variance equals 0.1

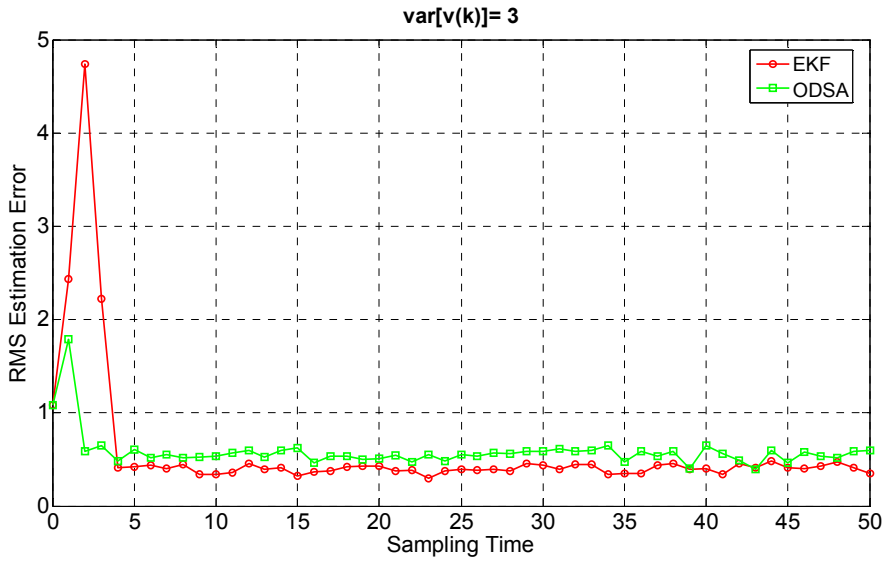

Figure-103: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 3 and disturbance noise variance equals 0.1

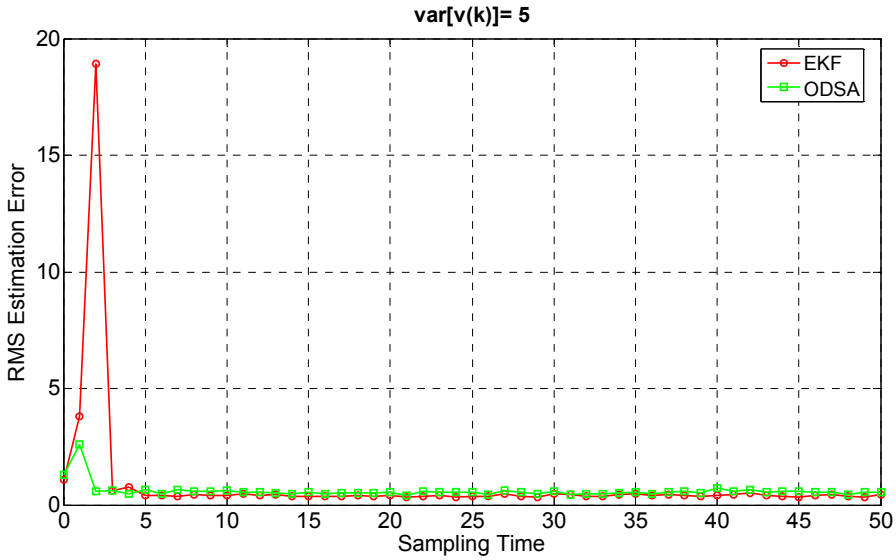

Figure-104: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 5 and disturbance noise variance equals 0.1

Table-41 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the variance of v(k) changes and when the disturbance noise variance equals 0.1

Variance of v(k)	0.1	1	3	5
EKF Average of all RMS Errors	0.4105	0.5741	0.8537	0.4196
ODSA Average of All RMS Errors	0.5875	0.5684	0.5832	0.5972

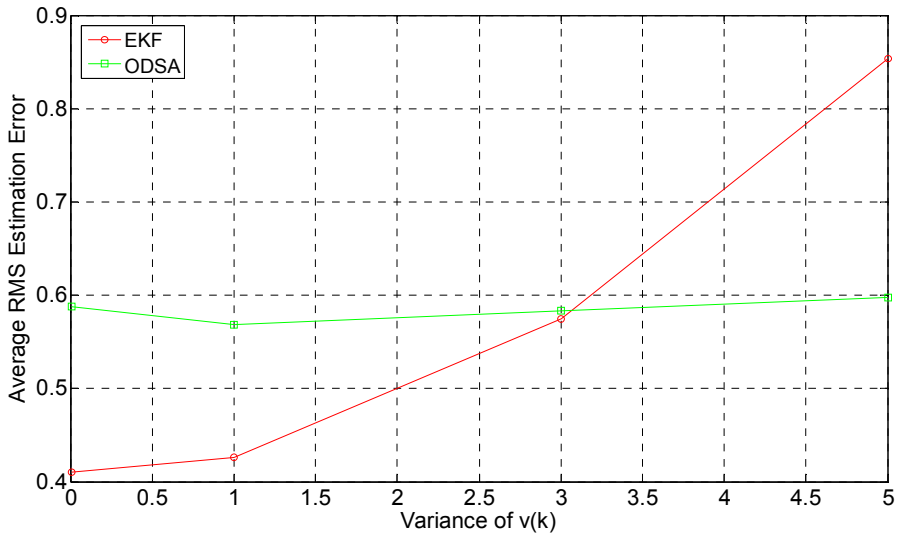

Figure-105: Estimation performance comparison of EKF and ODSA as the variance of v(k) changes and when the disturbance noise variance equals 0.1

Comment:

It can be observed form Figure 105 and Table 41 that EKF shows a better estimation performance than ODSA for small measurement noise variance values of 0.1 and 1 while ODSA shows a better estimation performance than EKF for high measurement noise variance values. It can also be observed that ODSA

estimation performance do not change very much with increasing measurement noise variance while EKF has a dramatic increase in estimation error with increasing measurement noise variance.

9.1.3.2 Simulation 2

Parameter values used in this simulation are same as the parameter values used in Simulation 1 given in section 9.1.3.1, except correlation coefficient, **var[w(k)] is taken as 1**

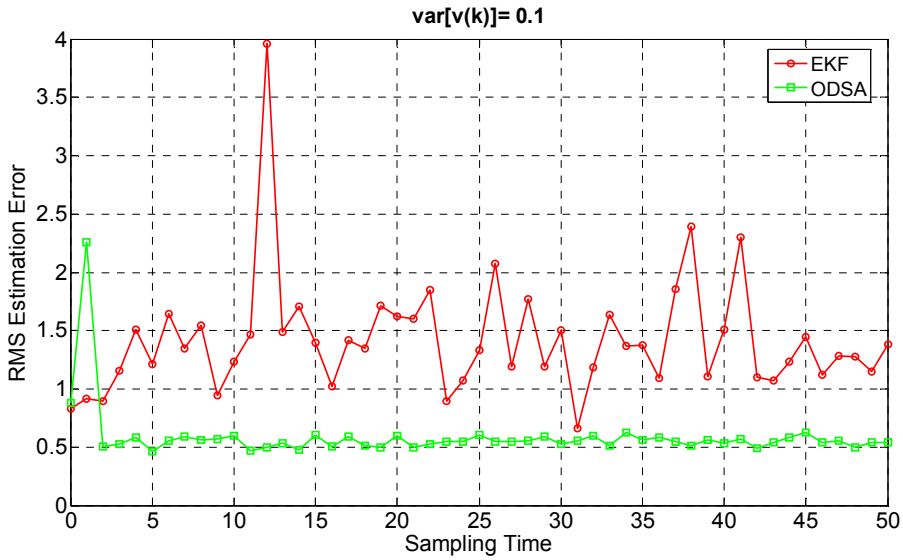

Figure-106: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 0.1 and disturbance noise variance equals 1

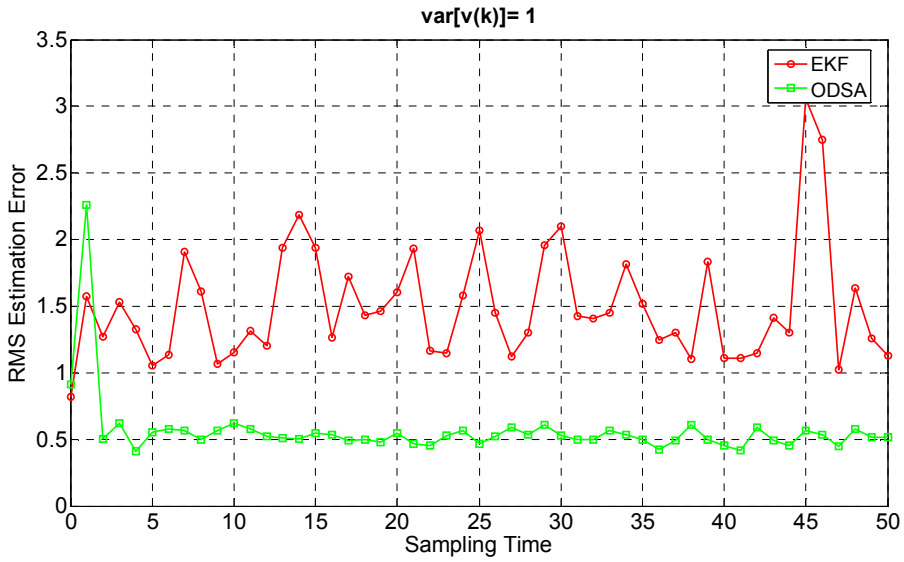

Figure-107: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 1 and disturbance noise variance equals 1

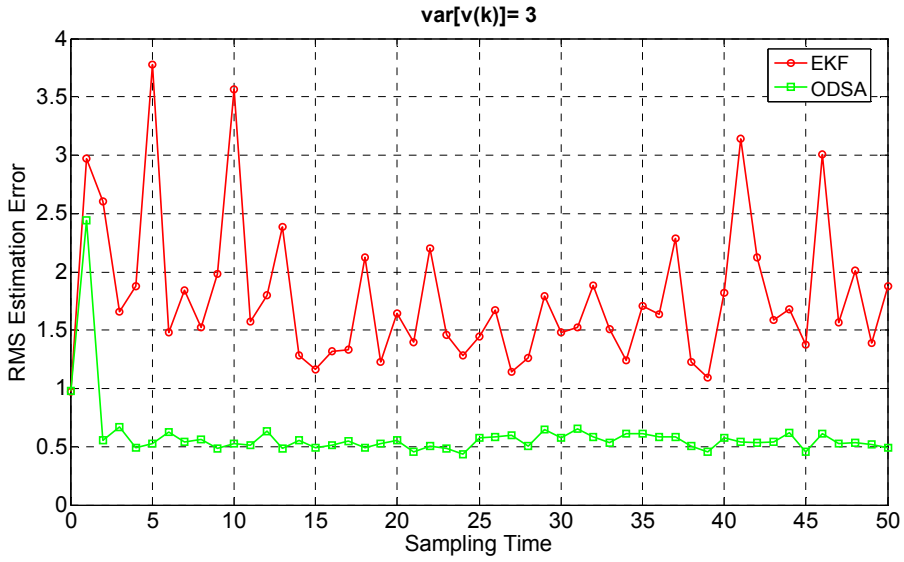

Figure-108: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 3 and disturbance noise variance equals 1

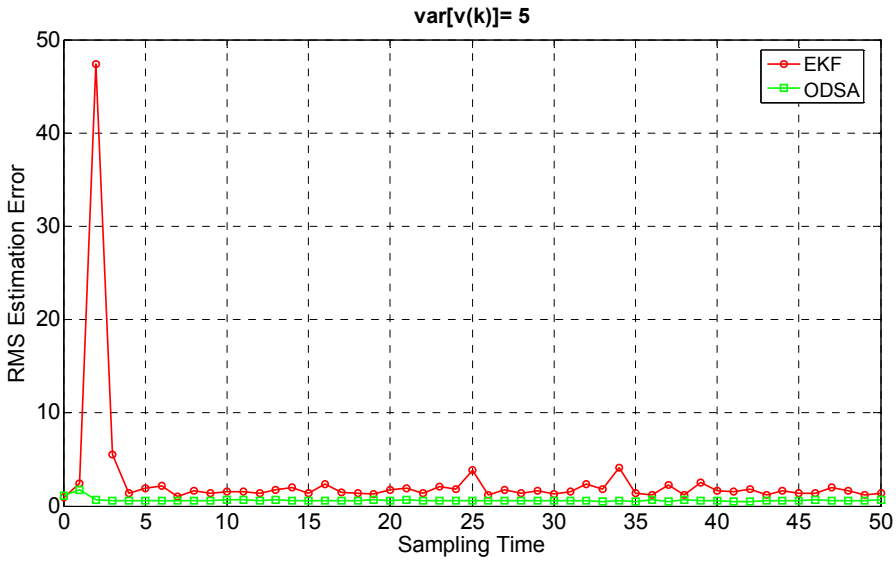

Figure-109: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 5 and disturbance noise variance equals 1

Table-42 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the variance of v(k) changes and when the disturbance noise variance equals 1

Variance of v(k)	0.1	1	3	5
EKF Average of all RMS Errors	1.4190	1.4975	1.7826	2.6814
ODSA Average of All RMS Errors	0.5872	0.5638	0.5906	0.6155

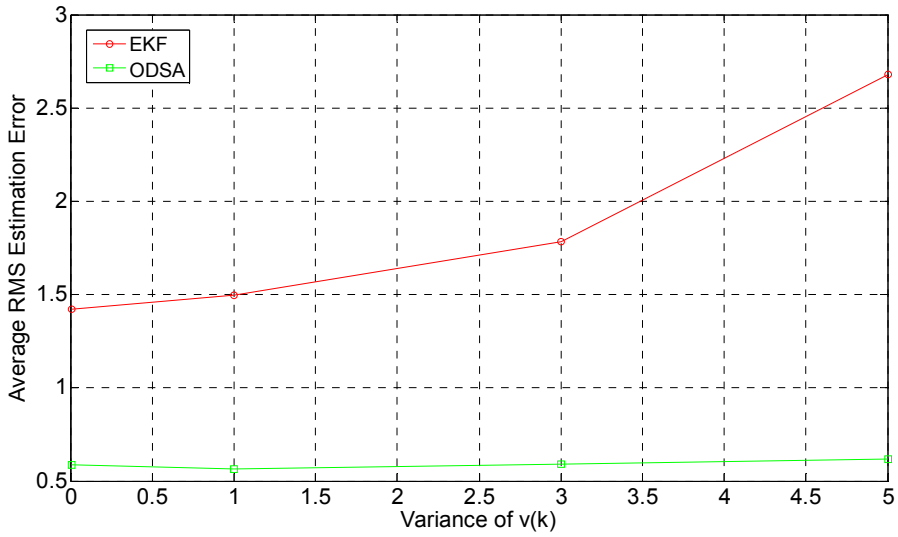

Figure-110: Estimation performance comparision of EKF and ODSA as the variance of v(k) changes and when the disturbance noise variance equals 1

Comment:

When Figure 110 and Table 42 are studied, it can be said that ODSA shows a better and more stable estimation performance than EKF.

9.1.4 Comparision as the Initial State Variance Changes

In this section, effects of the initial state variance on both EKF and ODSA are investigated.

Parameters used in ODSA are:

total number of samples, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of max states = 100

number of max v(0) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)]= 0, E[w(k)]= 0, ,E[v(0)] = 0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[w(k)] = 0.1, var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for four different values of initial state variance, which are [0.01 0.1 1 3] , after 500 executions for each algorithm.

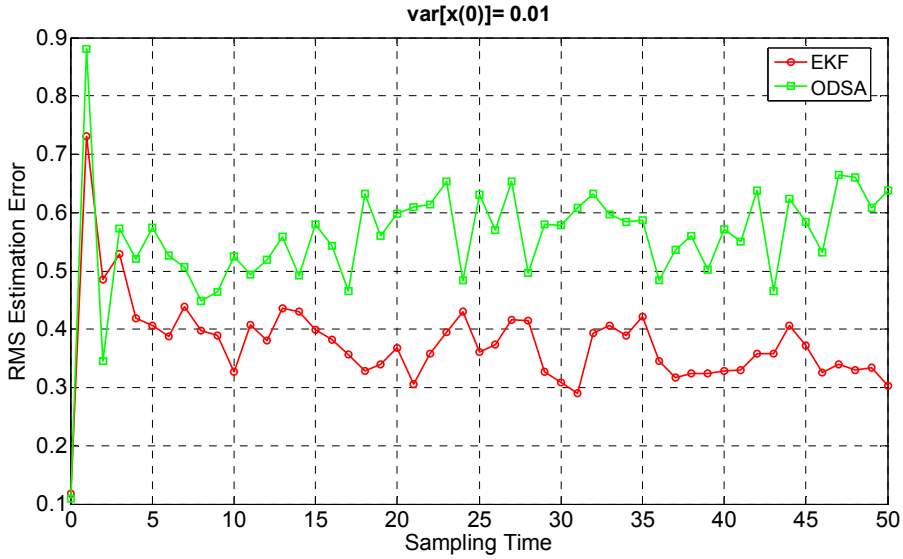

Figure-111: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 0.01

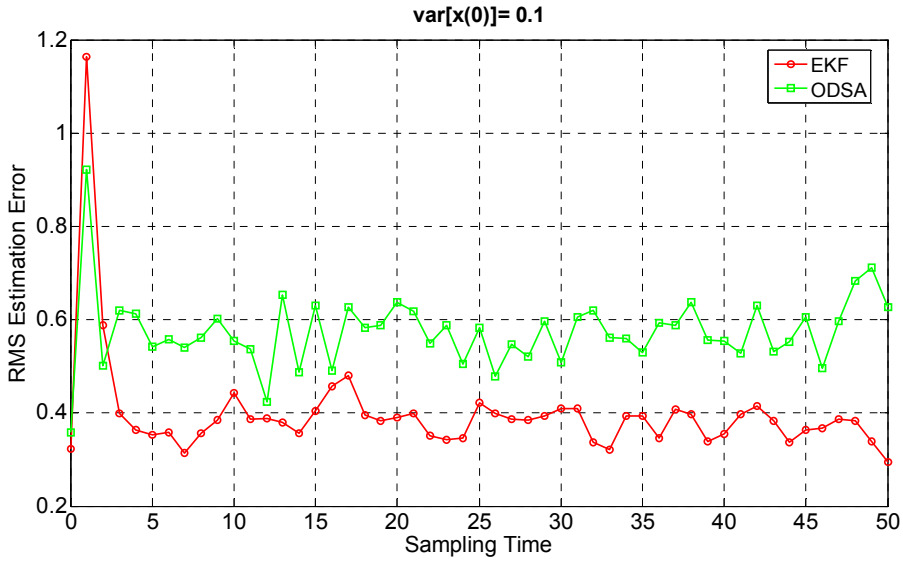

Figure-112: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 0.1

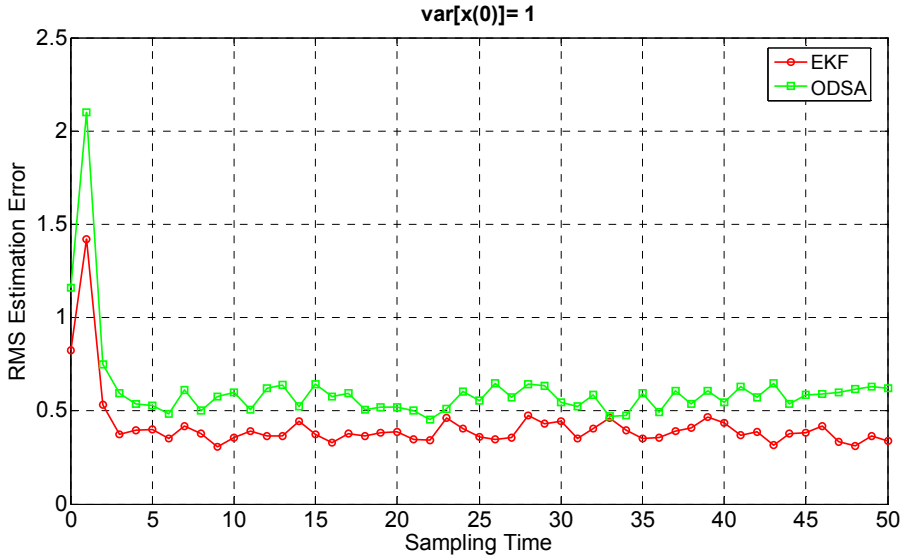

Figure-113: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 1

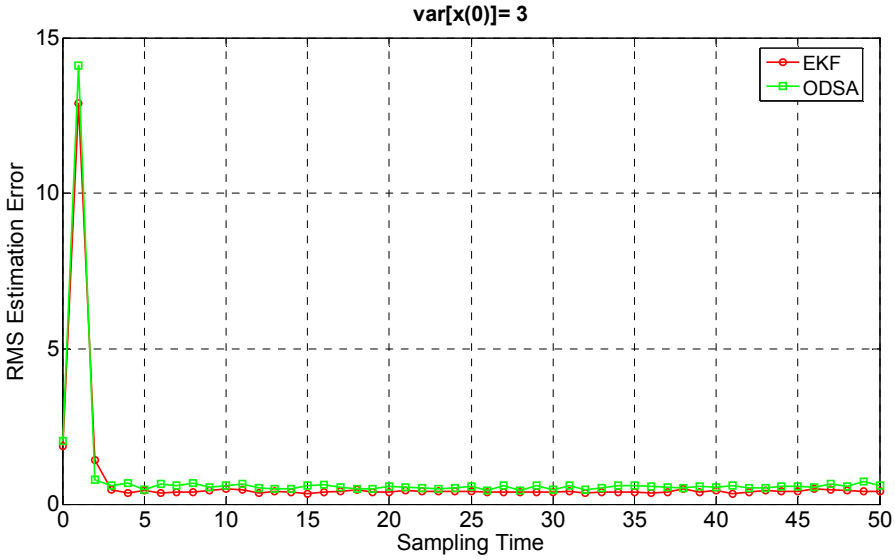

Figure-114: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 3

Table-43 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the variance of x(0) changes

Variance of x(0)	0.01	0.1	1	3
EKF Average of all RMS Errors	0.3752	0.3973	0.4140	0.6990
ODSA Average of All RMS Errors	0.5570	0.5743	0.6119	0.8574

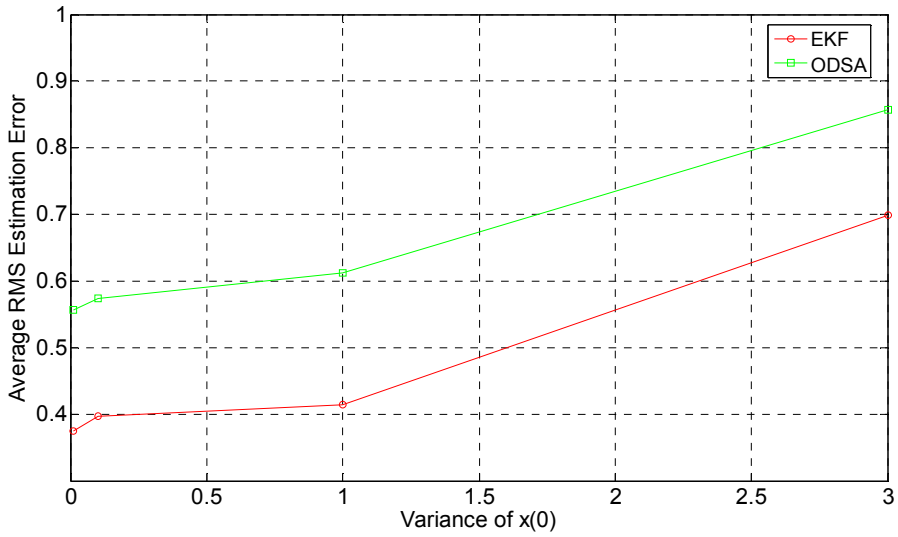

Figure-115: Estimation performance comparison of EKF and ODSA as the variance of x(0) changes

Comment:

Figure 115 and Table 43 show that EKF shows a better estimation performance than ODSA.

9.1.5 Comparision as the Sampling Number Changes

In this section, effects of the sampling number on both EKF and ODSA are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a= 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)]=0

Parameters used in EKF are:

correlation coefficient=0.1

variances : var[x(0)] = 1, var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

9.1.5.1 Simulation 1

In this simulation, results are obtained for ten different values of sampling time, which are [10 20 30 40 50 60 70 80 90 100] after 500 executions for each algorithm. Disturbance noise variance, **var[w(k)], is taken as 0.1**

Table-44 : Average values of all RMS estimation errors from k=0 to k=L for the linear model of both EKF and ODSA as the sampling number changes

Sampling Time	10	20	30	40	50
EKF Average of all RMS Errors	0.8947	0.6476	0.5362	0.4866	0.4539
ODSA Average of All RMS Errors	0.7097	0.7126	0.6479	0.6361	0.6143
Sampling Time	60	70	80	90	100
EKF Average of all RMS Errors	0.4949	0.4058	0.4446	0.4282	0.4445
ODSA Average of All RMS Errors	0.6083	0.6133	0.6049	0.6014	0.5783

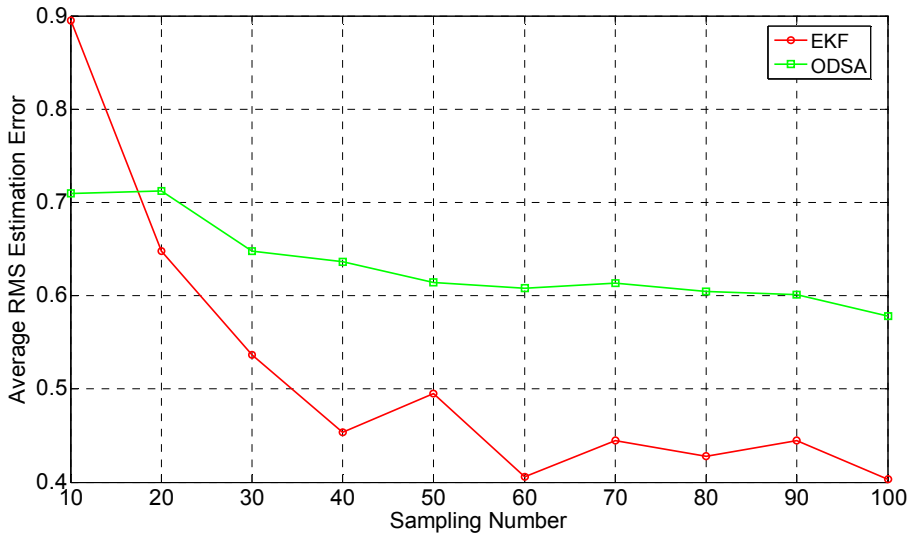

Figure-116: Estimation performance comparison of EKF and ODSA as the sampling number changes

Comment:

Figure 116 shows that EKF shows a better estimation performance than ODSA.

9.1.5.2 Simulation 2

Parameter values used in this simulation are same as the parameter values used in Simulation 1 given in section 9.1.5.1, except correlation coefficient, **var[w(k)] is taken as 2**

Table-45 : Average values of all RMS estimation errors from k=0 to k=L for the linear model of both EKF and ODSA as the sampling number changes

Sampling Time	10	20	30	40	50
EKF Average of all RMS Errors	4.1977	2.8047	5.7412	3.4738	4.0088
ODSA Average of All RMS Errors	4.0700	3.1142	3.9195	4.0860	4.7830
Sampling Time	60	70	80	90	100
EKF Average of all RMS Errors	4.3253	3.6128	4.3689	3.9229	3.7966
ODSA Average of All RMS Errors	4.3757	3.9557	3.8071	4.2917	4.2682

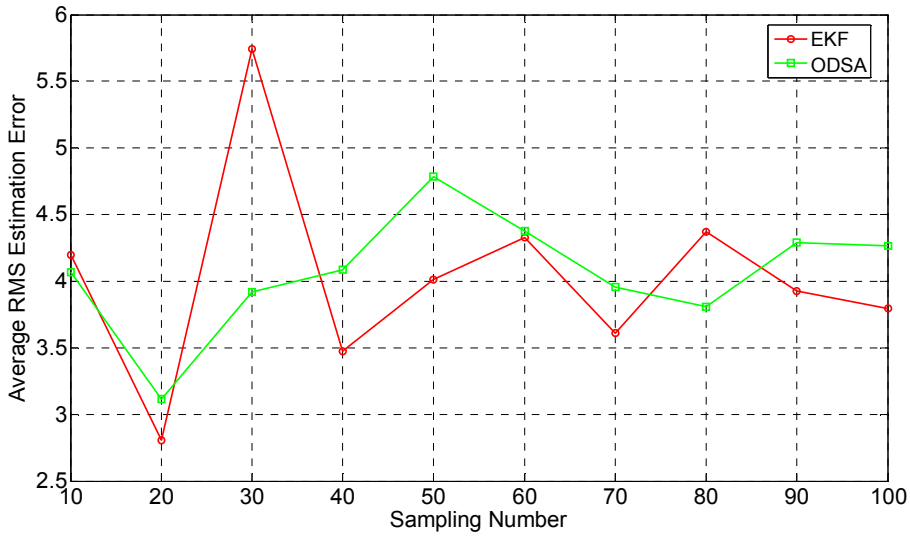

Figure-117: Estimation performance comparison of EKF and ODSA as the sampling number changes

Comment:

From Figure 117, it can be said that ODSA and EKF show nearly close estimation performances as the sampling number changes when the disturbance noise variance is increased to the value of 2.

9.1.6 Run-Time Comparision of ODSA And EKF

In this section, run time comparision of EKF and ODSA will be done as the sampling number changes. To get approximately the same estimation error performance for both algorithms, the parameters below are used:

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a= 0.1

gate size = 1

number of maximum states = 50

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 3, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)]=0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[x(0)] = 1 var[w(k)] = 0.1, var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for ten different values of sampling time, which are [10 20 30 40 50 60 70 80 90 100].

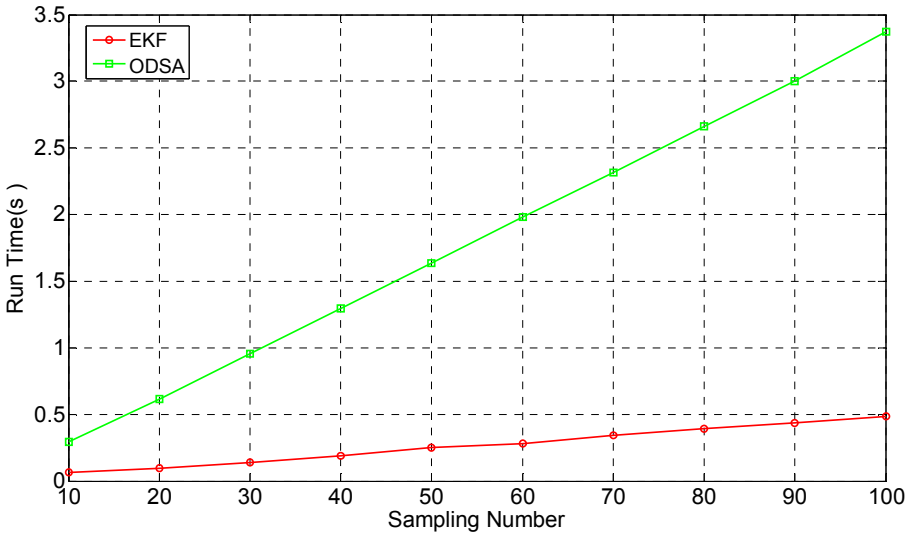

Figure-118: Run time comparison of EKF and ODSA as the sampling number changes

Table-46 : Run Time of both EKF and ODSA as the sampling number changes

Sampling Time	10	20	30	40	50
EKF Run Time(s)	0.0630	0.0940	0.1410	0.1870	0.2500
ODSA Run Time(s)	0.2968	0.6125	0.9531	1.2960	1.6344
Sampling Time	60	70	80	90	100
EKF Run Time(s)	0.2820	0.3434	0.3910	0.4370	0.4850
ODSA Run Time(s)	1.9797	2.3156	2.6594	3.0047	3.3734

Comment:

From Figure 94 and Table 42, it can be observed that EKF is faster than ODSA. For both algorithms, run time consumption increases as the sampling time increases, but EKF shows smaller linear increase, whereas ODSA shows a larger

linear increase. This is not surprising, since EKF has far less computational work than ODSA.

9.2 Simulations for Nonlinear System Model 2

In this section, nonlinear models given below will be used in the simulations for both ODSA and EKF:

Motion model : x (k+1) = sin(-x(k)+exp (- x (k)) + w (k) (9.8)

Measurement model : z (k) = cos(x (k)) + v (k)

where correlated measurement noise, v (k), is modelled as:

v (k)=a v (k-1)+r (k).

To use the EKF for the nonlinear models given in Eq. (8.15), Jacobian matrices: *A, W, H* and *V* are calculated as below:

- *A* , Jacobian matrix of partial derivatives of *f* with respect to *x*:

$$A_{ij} = \frac{\partial f_i}{\partial x_j}\left(\hat{x}(k-1), u(k), 0\right) = -\cos(x(k)) - \exp(-x(k)) \qquad (9.9)$$

- *W* is the Jacobian matrix of partial derivatives of *f* with respect to *w*:

$$W_{ij} = \frac{\partial f_i}{\partial w_j}\left(\hat{x}(k-1), u(k), 0\right) = 1 \tag{9.10}$$

- H is the Jacobian matrix of partial derivatives of h with respect to x:

$$H_{ij} = \frac{\partial h_i}{\partial x_j}\left(\hat{x}(k|k-1), 0\right) = -\sin(x(k)) \tag{9.11}$$

- V is the Jacobian matrix of partial derivatives of h with respect to v:

$$V_{ij} = \frac{\partial h_i}{\partial v_j}\left(\hat{x}(k|k-1), 0\right) = 1 \tag{9.12}$$

9.2.1 Comparision as the Correlation Coefficient Changes

In this section, effects of the correlation coefficient on both ODSA and EKF are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)]= 0, E[w(k)] = 0, E[v(0)] = 0, E[r(k)]= 0

Parameters used in EKF are:

total sampling time, L = 50

variances : var[x(0)] = 1, var[v(k)]=1, var[w(k)]= 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]= 0

9.2.1.1 Simulation 1

In this simulation, results are obtained for five different values of correlation coefficient, which are [0.01 0.03 0.05 0.07 0.09], after 500 executions for each algorithm.

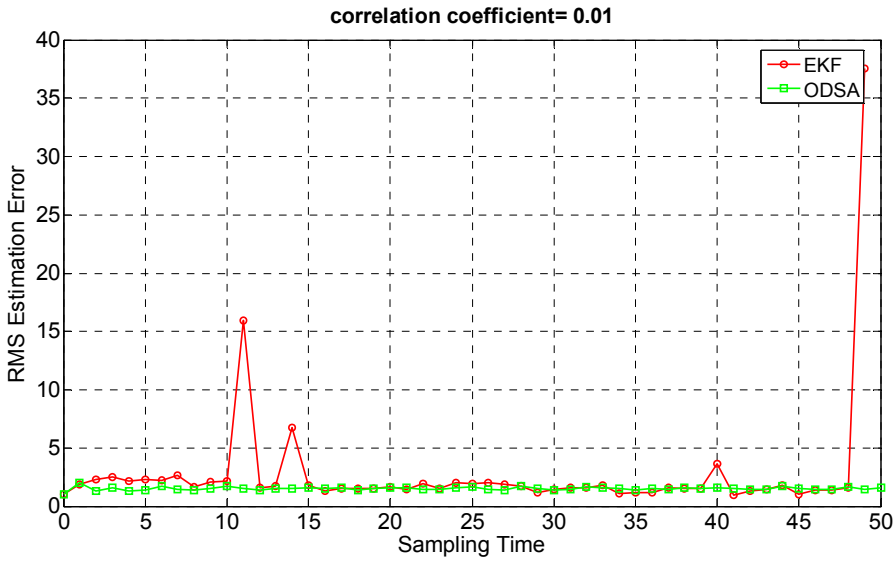

Figure-119: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.01

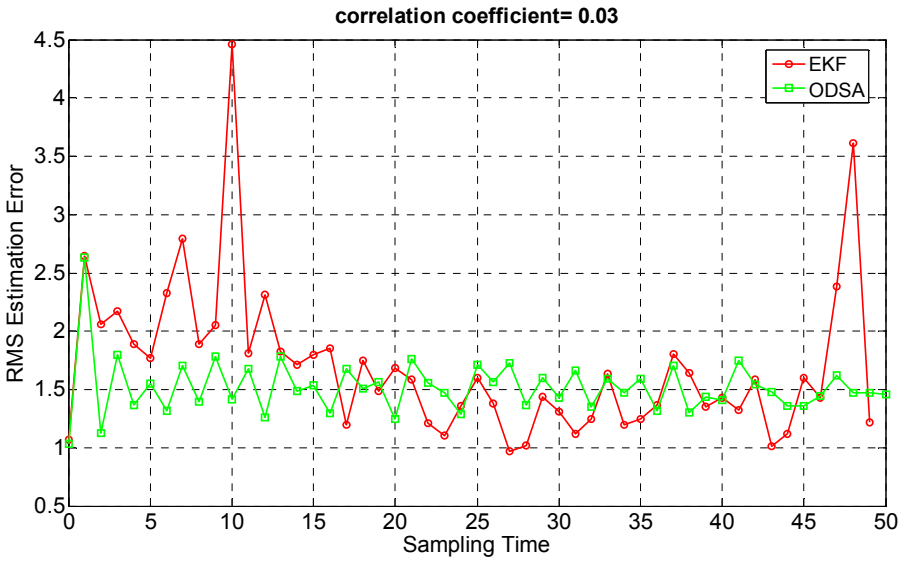

Figure-120: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.03

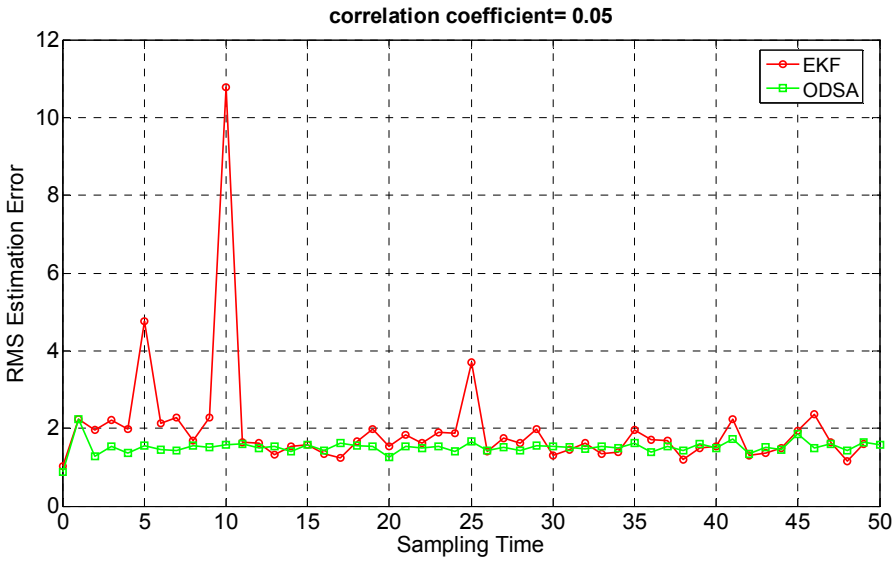

Figure-121: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.05

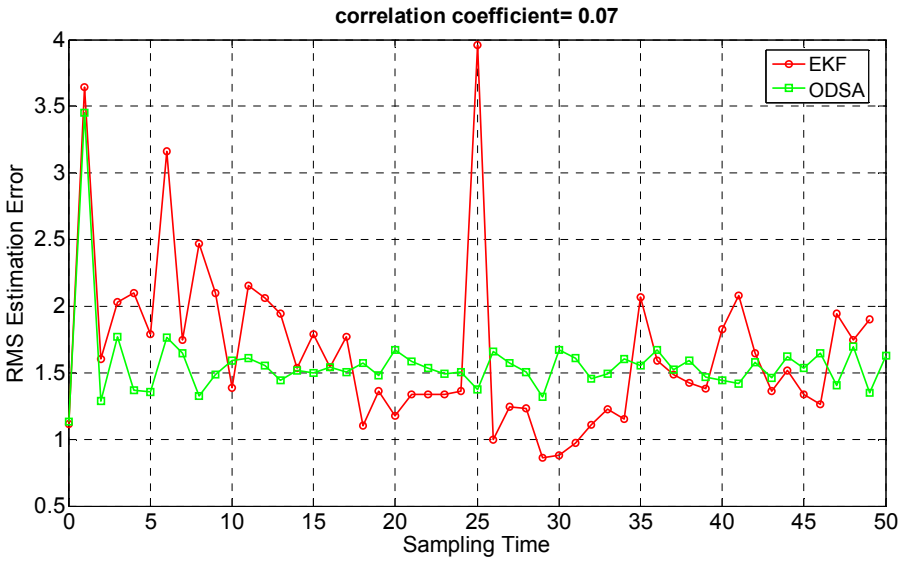

Figure-122: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.07

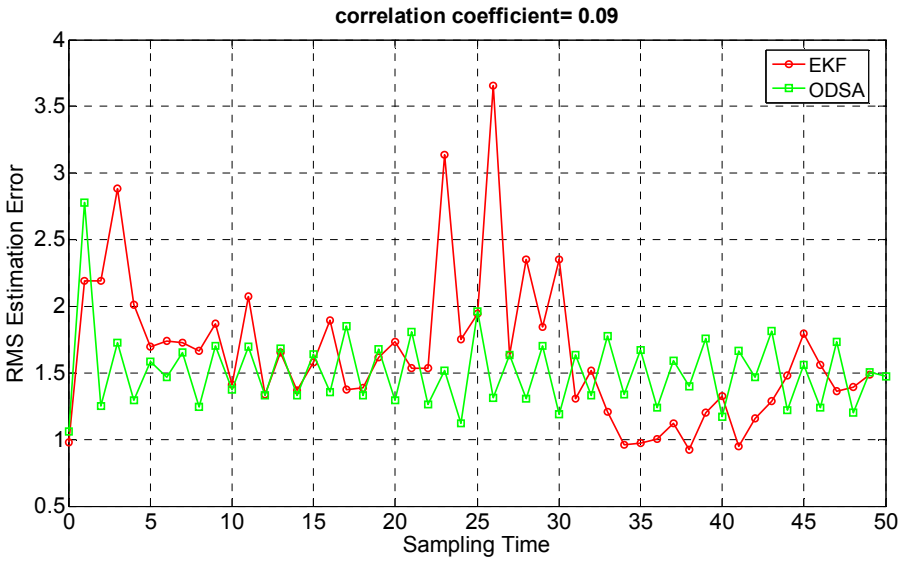

Figure-123: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.09

Table-47 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the correlelation coefficient changes

Correlation Coefficient, a	0.01	0.03	0.05	0.07	0.09
EKF Average of all RMS Errors	2.8102	1.6968	1.9657	1.6635	1.6432
ODSA Average of All RMS Errors	1.5166	1.5170	1.5153	1.5597	1.5088

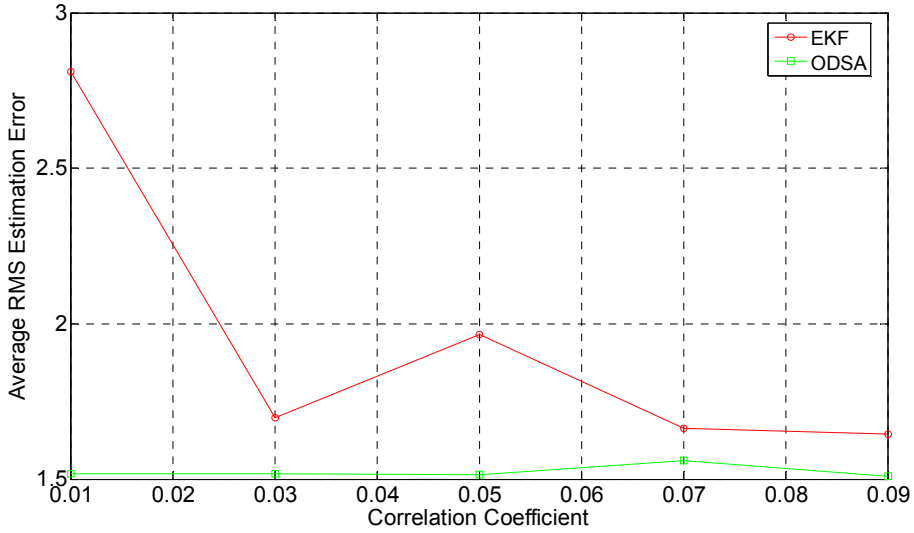

Figure-124: Estimation performance comparison of EKF and ODSA as the correlation coefficient changes

Comment:

Figure 124 and Table 47 show that ODSA shows a better estimation performance than EKF.

9.2.1.2 Simulation 2

Parameter values used in this simulation are same as the parameter values used in Simulation 1 given in section 9.2.1.1, except correlation coefficient, a, varies as [0.1 0.3 0.5 0.7].

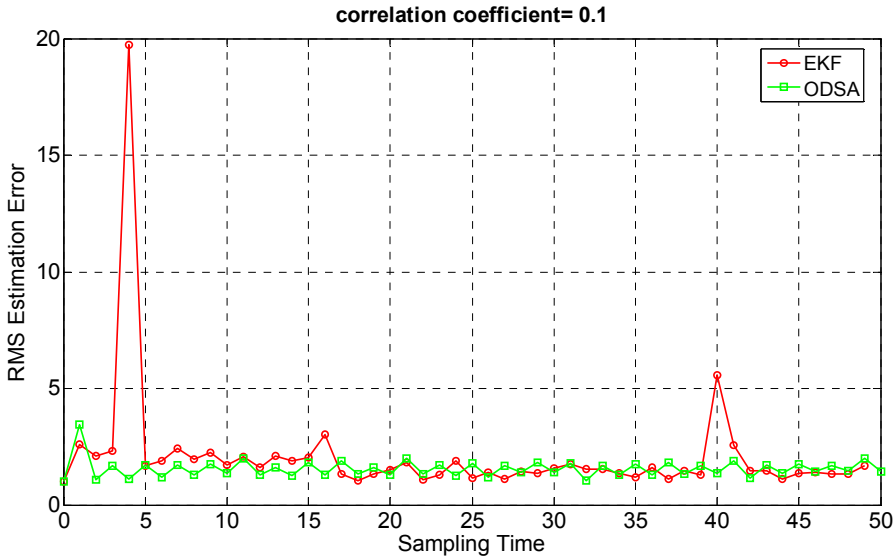

Figure-125: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.1

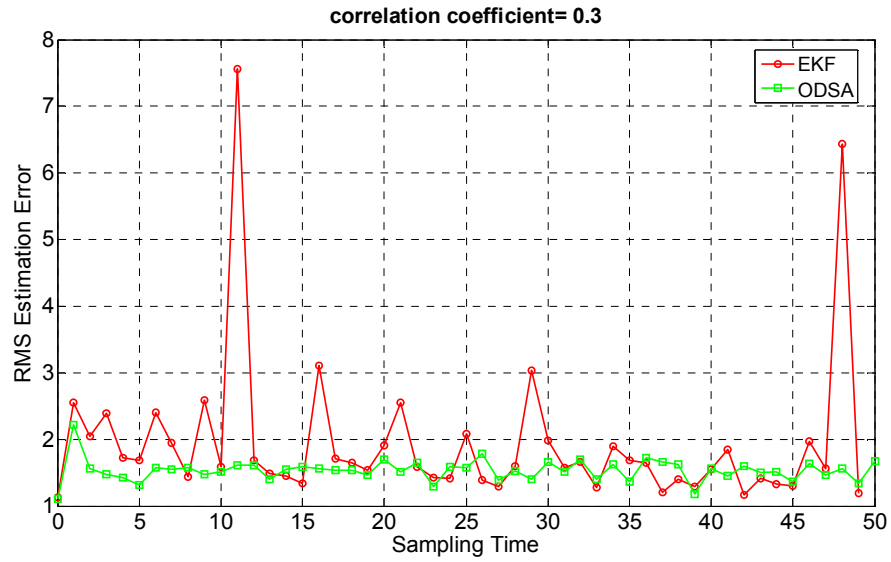

Figure-126: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.3

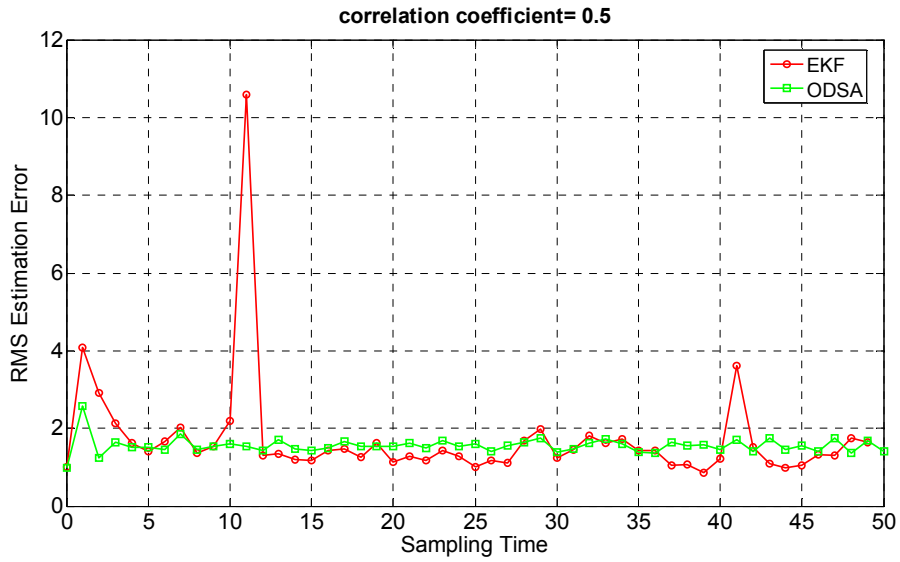

Figure-127: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.5

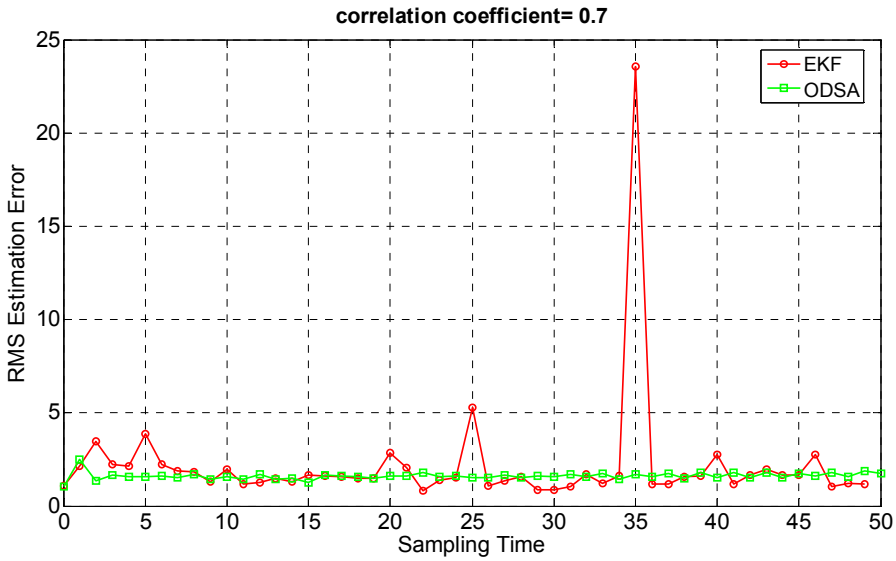

Figure-128: RMS estimation error versus sampling time for both EKF and ODSA when the correlation coefficient equals 0.7

Table-48 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the correlation coefficient changes

Correlation Coefficient, a	0.1	0.3	0.5	0.7
EKF Average of all RMS Errors	2.0764	1.9348	1.7174	2.1698
ODSA Average of All RMS Errors	1.5520	1.5341	1.5572	1.6067

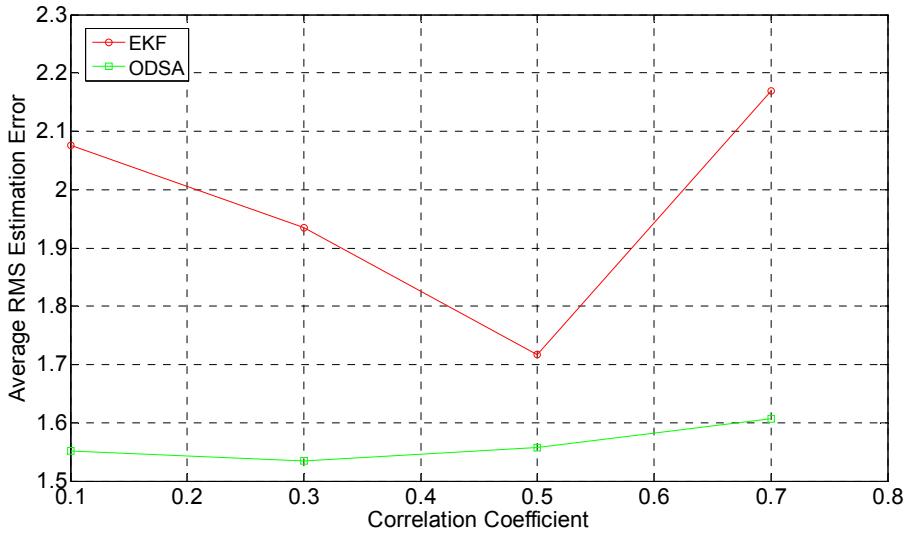

Figure-129: Estimation performance comparison of EKF and ODSA as the correlation coefficient changes

Comment:

Figure 129 and Table 48 show that ODSA shows a better estimation performance than EKF.

9.2.2 Comparision as the Disturbance Noise Variance Changes

In this section, effects of the disturbance noise variance on both ODSA and EKF are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[x(0)] = 1, var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for four different values of disturbance noise variance, which are [0.01 0.05 0.1 0.5] after 500 executions for each algorithm.

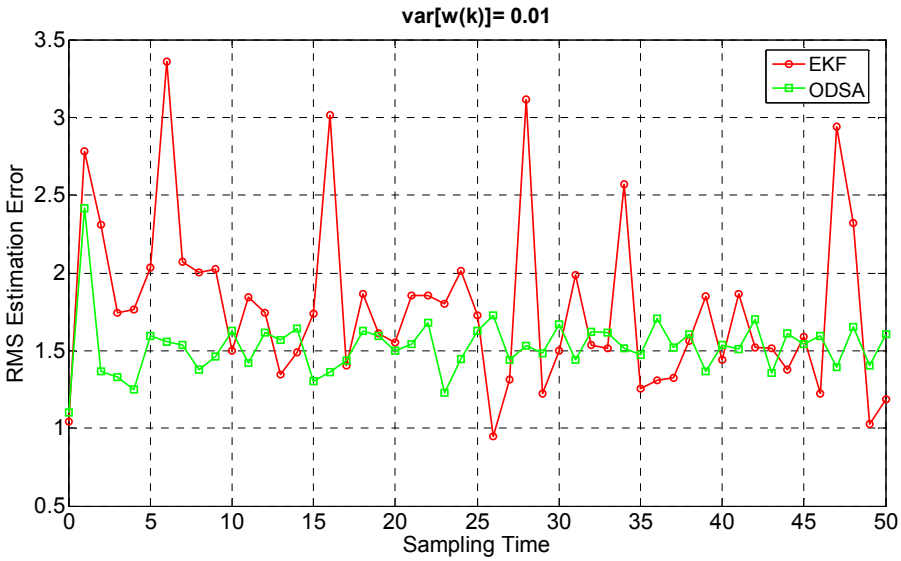

Figure-130: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 0.01

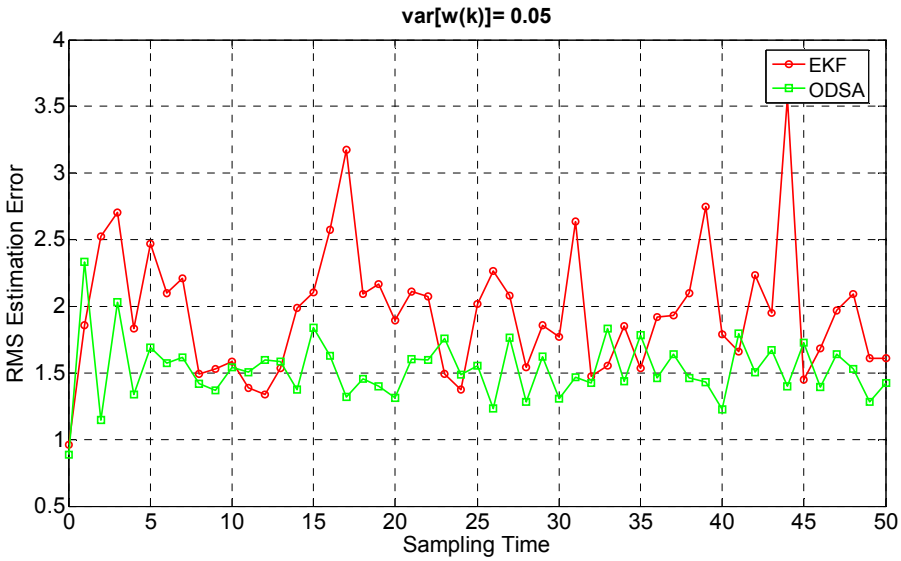

Figure-131: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 0.05

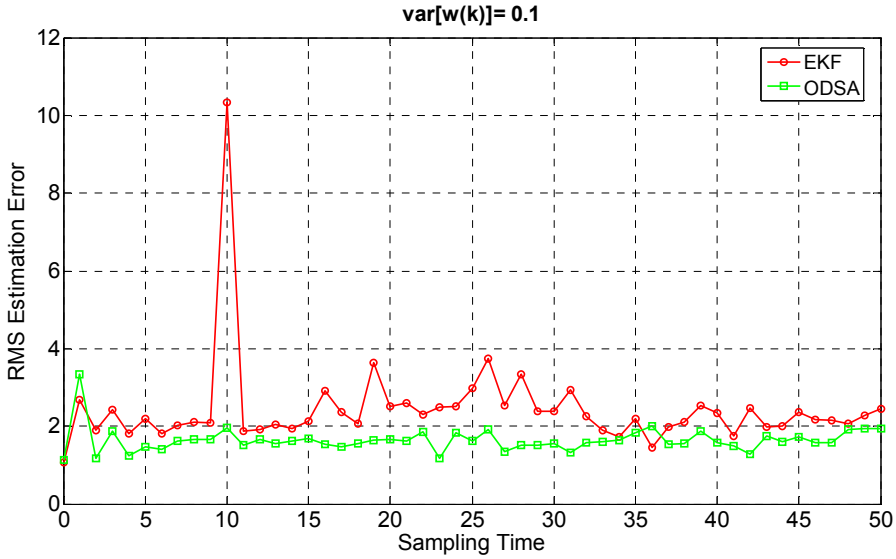

Figure-132: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 0.1

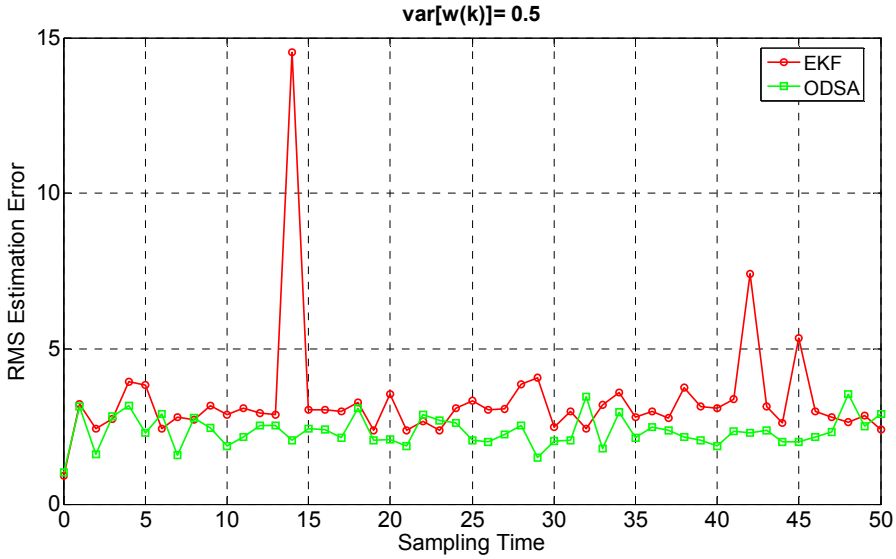

Figure-133: RMS estimation error versus sampling time for both EKF and ODSA when the variance of w(k) equals 0.5

Table-49 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the variance of w(k) changes

Variance of w(k)	0.01	1	3	5
EKF Average of all RMS Errors	1.7745	1.9501	2.4374	3.3209
ODSA Average of All RMS Errors	1.5260	1.5238	1.6433	2.3380

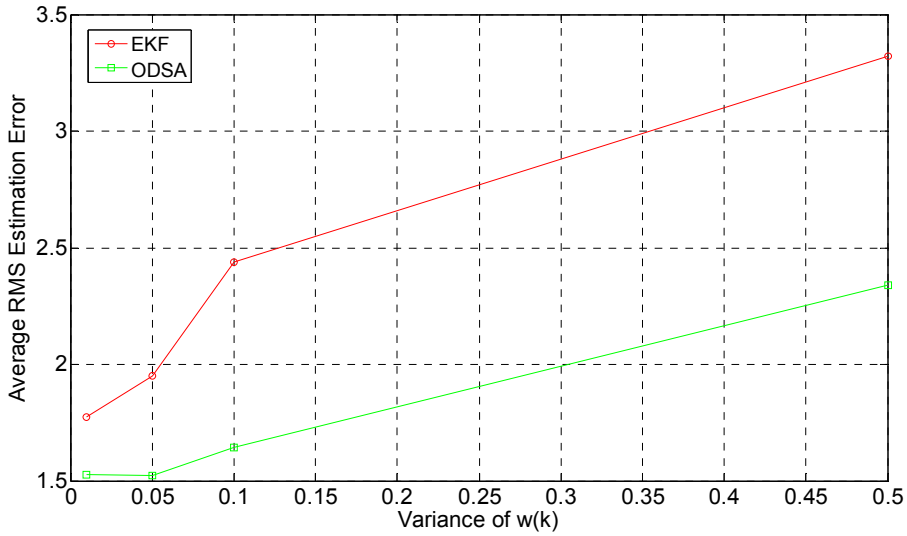

Figure-134: Estimation performance comparison of EKF and ODSA as the correlation coefficient changes

Comment:

Figure 134 and Table 49 show that ODSA shows a better estimation performance than EKF.

9.2.3 Comparision as the Measurement Noise Variance Changes

In this section, effects of the measurement noise variance on both EKF and ODSA are investigated.

Parameters used in ODSA are:

total number of samples, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0 ,E[v(0)] = 0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[x(0)] = 1, var[w(k)]= 0.1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for four different values of correlated measurement noise variance, which are [0.01 0.1 1 3] after 500 executions for each algorithm.

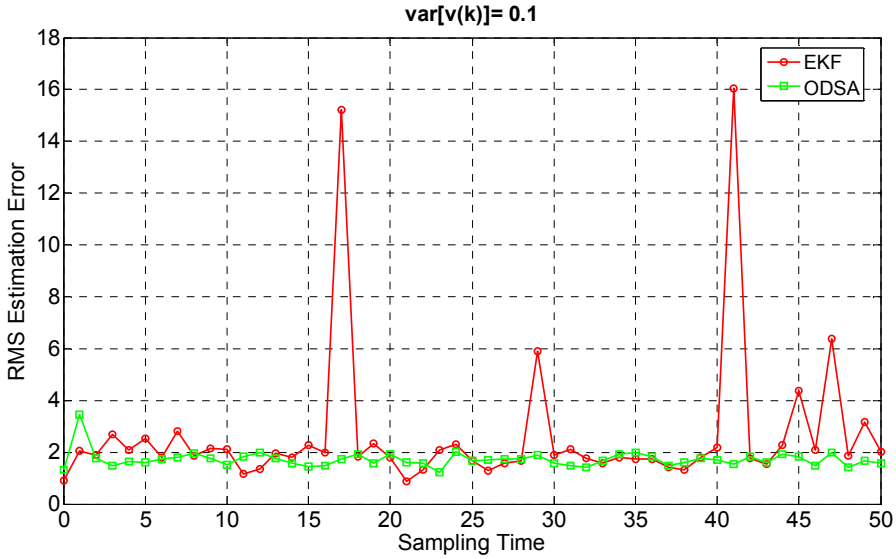

Figure-135: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 0.1

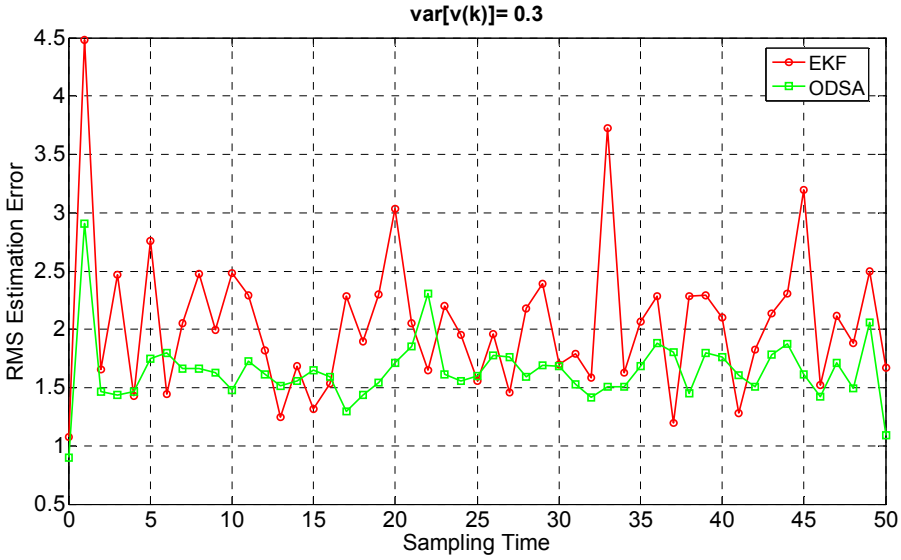

Figure-136: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 0.3

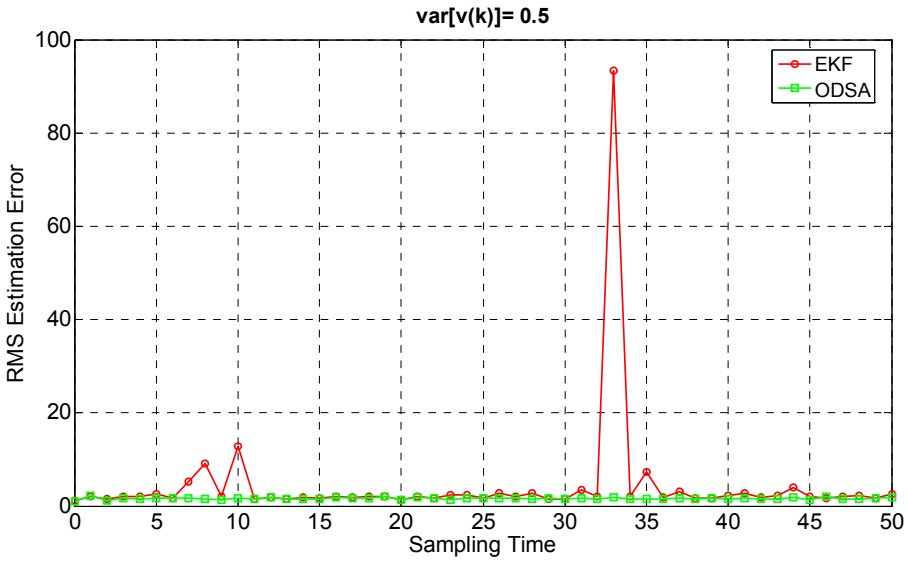

Figure-137: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 0.5

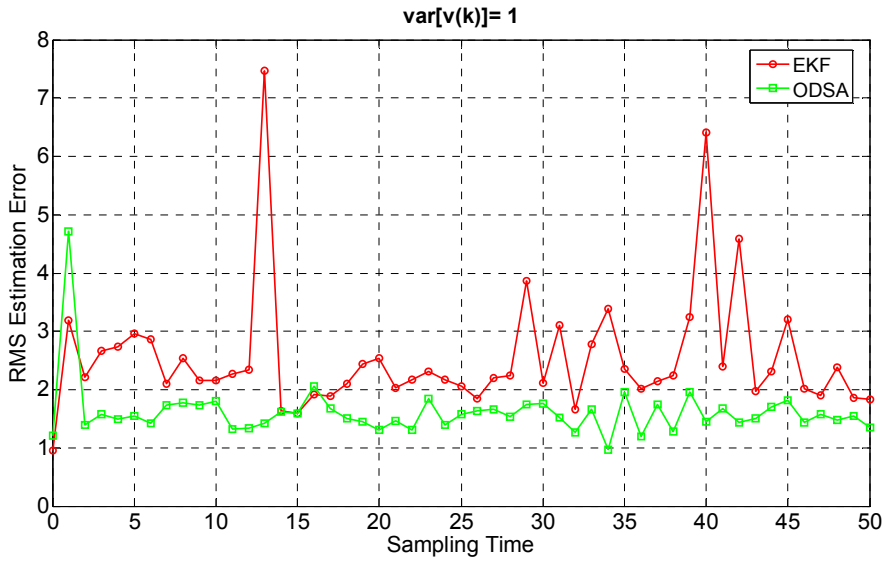

Figure-138: RMS estimation error versus sampling time for both EKF and ODSA when the variance of v(k) equals 1

Table-50 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as the variance of v(k) changes

Variance of v(k)	0.1	0.3	0.5	1
EKF Average of all RMS Errors	2.6233	2.0426	4.3570	2.5384
ODSA Average of All RMS Errors	1.7195	1.6411	1.6151	1.6094

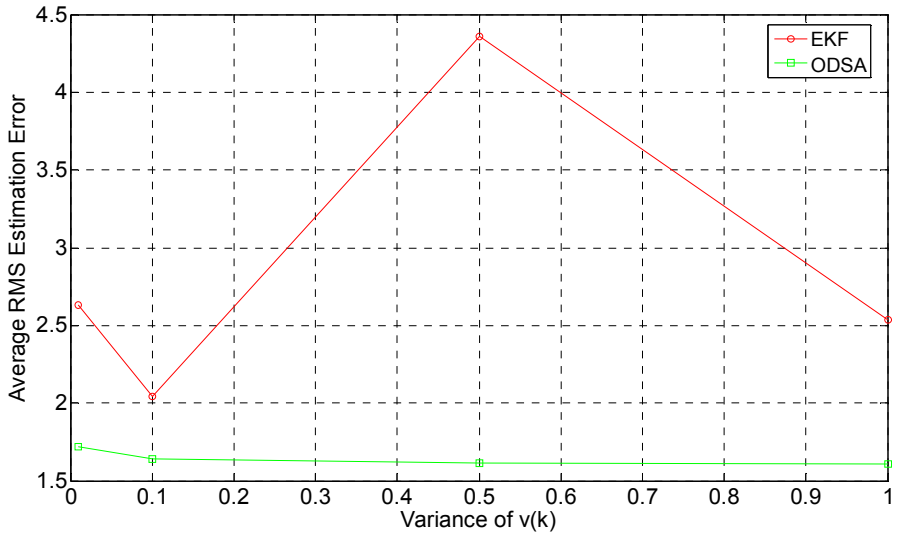

Figure-139: Estimation performance comparison of EKF and ODSA as the variance of v(k) changes

Comment:

Figure 139 and Table 50 show that ODSA shows a better estimation performance than EKF.

9.2.4 Comparision as the Initial State Variance Changes

In this section, effects of the initial state variance on both EKF and ODSA are investigated.

Parameters used in ODSA are:

total number of samples, L = 50

correlation coefficient, a = 0.1

gate size = 0.1

number of max states = 100

number of max v(0) states=50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)]= 0, E[w(k)]= 0, ,E[v(0)] = 0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[w(k)] = 0.1, var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for four different values of initial state variance, which are [0.01 0.1 0.3 0.5], after 500 executions for each algorithm.

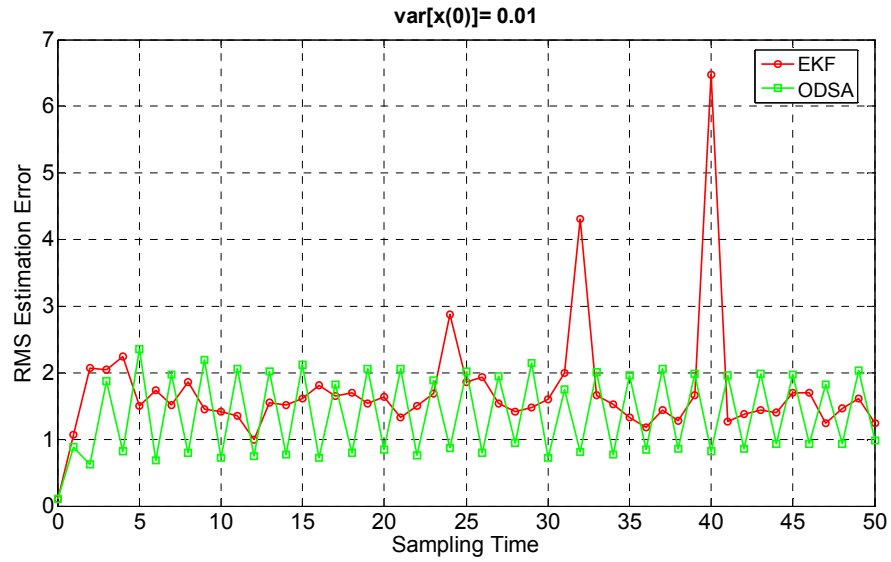

Figure-140: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 0.01

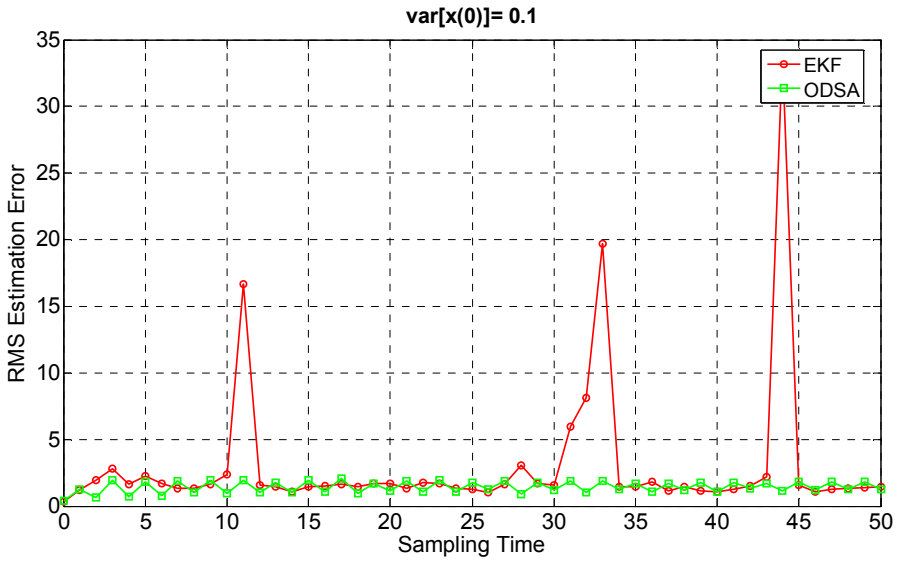

Figure-141: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 0.1

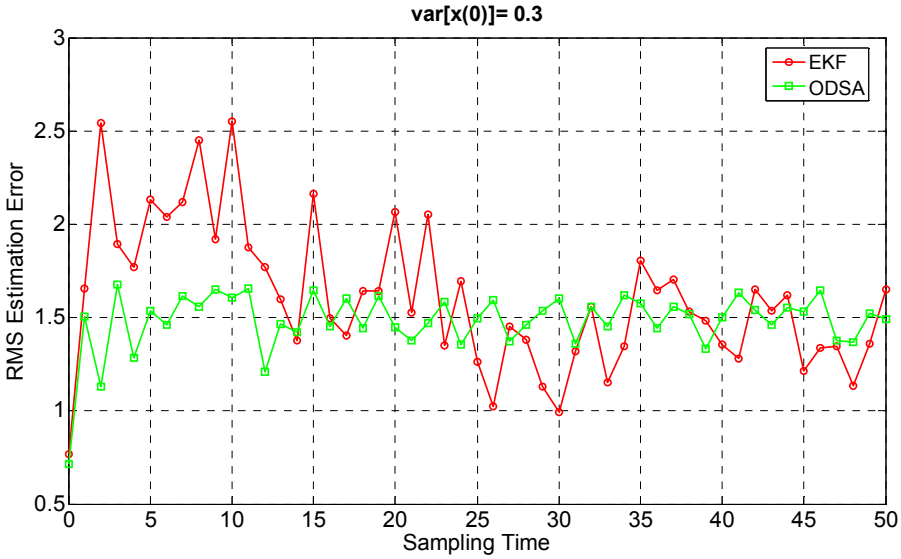

Figure-142: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 0.3

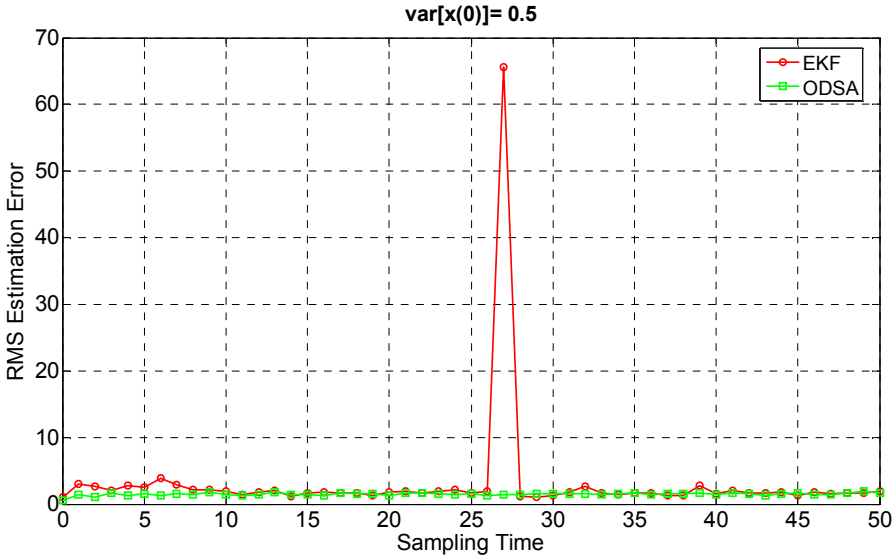

Figure-143: RMS estimation error versus sampling time for both EKF and ODSA when the variance of x(0) equals 0.5

Table-51 : Average values of all RMS estimation errors from k=0 to k=L for both EKF and ODSA as variance of x(0) changes

Variance of x(0)	0.01	0.1	0.3	0.5
EKF Average of all RMS Errors	1.7054	3.0473	1.6029	3.1083
ODSA Average of All RMS Errors	1.3622	1.4272	1.4820	1.4895

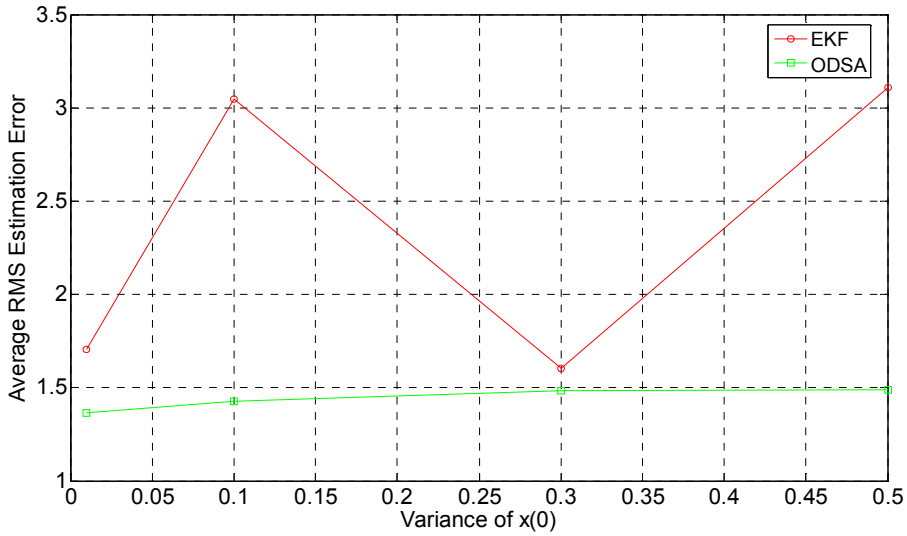

Figure-144: Estimation performance comparison of EKF and ODSA as the variance of x(0) changes

Comment:

Figure 144 and Table 51 show that ODSA shows a better estimation performance than EKF.

9.2.5 Comparision as the Sampling Number Changes

In this section, effects of the sampling number on both EKF and ODSA are investigated.

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a= 0.1

gate size = 0.1

number of maximum states = 100

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 5, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[v(0)]=1, var[w(k)] = 0.1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)]=0

Parameters used in EKF are:

correlation coefficient=0.1

variances : var[x(0)] = 1, var[w(k)] = 0.1 var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for five different values of sampling time, which are [10 20 30 40 50] after 500 executions for each algorithm.

Table-52 : Average values of all RMS estimation errors from k=0 to k=L for the linear model of both EKF and ODSA as the sampling number changes

Sampling Time	10	20	30	40	50
EKF Average of all RMS Errors	11.1363	7.2940	4.3605	2.6872	2.5554
ODSA Average of All RMS Errors	1.7487	1.6685	1.6562	1.7214	1.6570

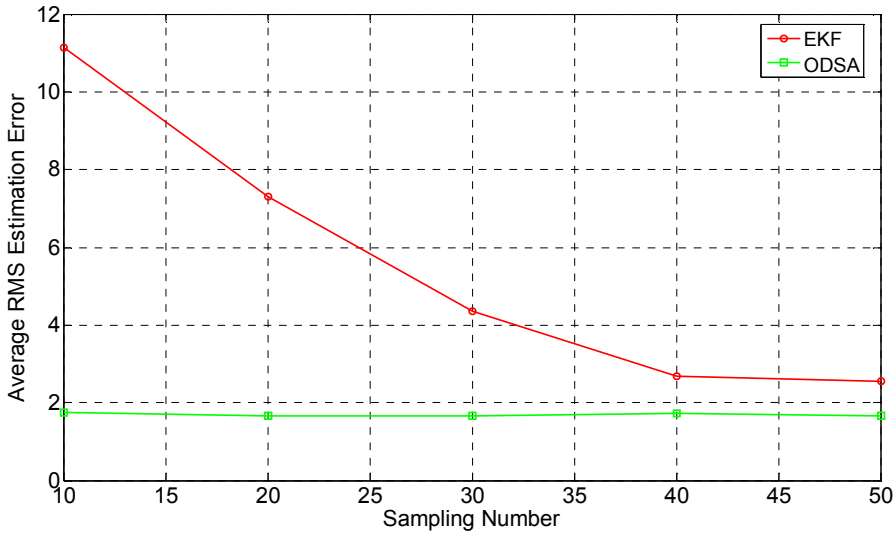

Figure-145: Estimation performance comparison of EKF and ODSA as the sampling number changes

Comment:

Figure 145 and Table 52 show that ODSA shows a better estimation performance than EKF.

9.2.6 Run-Time Comparision of ODSA And EKF

In this section, run time comparision of EKF and ODSA will be done as the sampling number changes. To get approximately the same estimation error performance for both algorithms, the parameters below are used:

Parameters used in ODSA are:

total sampling time, L = 50

correlation coefficient, a= 0.1

gate size = 1

number of maximum states = 50

number of maximum v(k) states =50;

quantization numbers : Q # of x(0) = 3, Q # of w(k)=3,

Q # of v(0) = 3, Q # of r(k)=3

variances : var[x(0)] = 1, var[w(k)] = 0.1, var[v(0)]=1

expected values : E[x(0)] = 0, E[w(k)] = 0, E[v(0)] = 0 , E[r(k)]=0

Parameters used in EKF are:

total sampling time, L = 50

correlation coefficient=0.1

variances : var[x(0)] = 1 var[w(k)] = 0.1, var[v(k)]= 1

expected values: E[x(0)] = 0, E[w(k)] = 0, E[v(k)]= 0 , E[r(k)]=0

In this simulation, results are obtained for five different values of sampling time, which are [10 20 30 40 50].

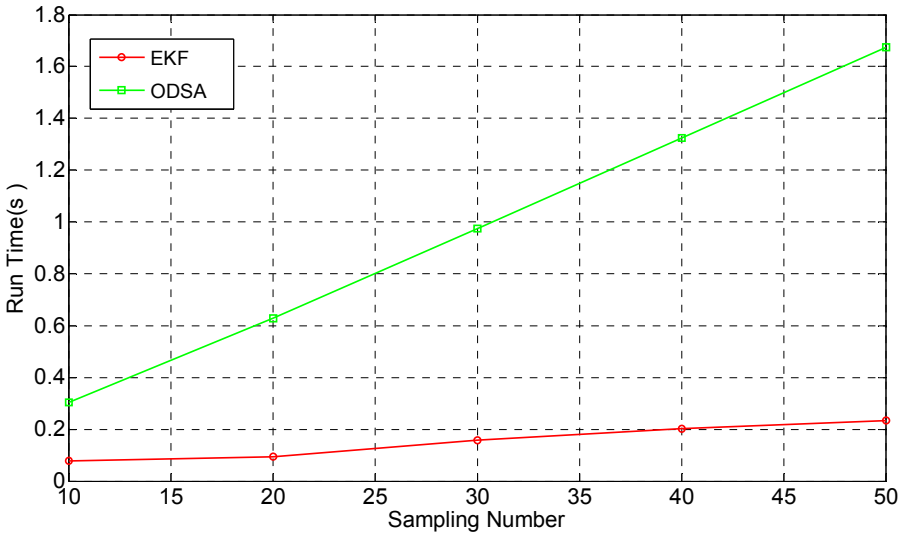

Figure-146: Run time comparison of EKF and ODSA as the sampling number changes

Table-53 : Run Time of both EKF and ODSA as the sampling number changes

Sampling Time	10	20	30	40	50
EKF Run Time(s)	0.0780	0.0930	0.1560	0.2030	0.2340
ODSA Run Time(s)	0.3031	0.6266	0.9735	1.3235	1.6735

Comment:

From Figure 146 and Table 53, it can be observed that EKF is faster than ODSA. For both algorithms, run time consumption increases as the sampling time increases, but EKF shows smaller linear increase, whereas ODSA shows a larger linear increase. This is not surprising, since EKF has less computational work than ODSA.

CHAPTER 10

CONCLUSION

In this study, the Optimum Decoding Based Smoothing Algorithm [1, 2] and the Optimum Decoding Based Smoothing Algorithm with correlated measurement noise are investigated.

The ODSA algorithm is based on Viterbi decoding algorithm. By reducing the target motion to a finite state model which uses the quantized state vector, a trellis diagram is obtained, and then, the state vector is estimated by finding the most probable path in the trellis diagram.

In order to use ODSA in the presence of correlated measurement noise, some modifications are done. To handle the correlation effect, firstly ODSA is implemented by applying decorrelation method [3, 4] on correlated measurements and secondly, ODSA is implemented by a method proposed in this thesis, which treats correlated measurement noise as interference. According to the the proposed method, correlated measurement noise is modeled by a first-order Markov model [3, 4]. The effect of correlation is thought as interference, and ODSA in the presence of interference [1, 2] is applied. The simulation results show that decorrelation method and proposed method show approximately close performances for linear models. Proposed method can be applied for both linear and nonlinear models while decorrelation method can only be applied for linear models.

For the proposed method, simulations are performed and the effects of parameters on the estimation performance of ODSA with correlated measurement noise are observed. The results can be summarized as below:

- Correlation coefficient affects the estimation performance significantly. As the correlation coefficient becomes smaller, the state estimation performance increases,
- Gate size affects the estimation performance significantly. As the gate size becomes smaller, the state estimation performance increases,
- Quantization number of the initial state vector affects the estimation performance slightly,
- Quantization number of the disturbance noise vector affects the estimation performance slightly,
- Quantization number of the initial correlated measurement noise vector affects the estimation performance slightly,
- Quantization number of the white measurement noise vector affects the estimation performance slightly,
- Initial state variance affects only the performance for only initial sampling times,
- Increasing the disturbance noise variance degrades the performance significantly,
- Initial measurement noise variance affects the estimation performance slightly,
- The maximum number of states can be limited without degrading the estimation performance,
- The maximum number of measurement noise states can be limited without degrading the estimation performance.

Gate size, quantization numbers, maximum state number and maximum initial measurement noise states are important factors for determining the computation time of the algorithm. Choosing these values properly, the computation time can be decreased while getting a good estimation performance. There is a trade-off between the precision and the computational time.

Alpha-Beta Filter Algorithm with correlated measurement noise is also implemented which was proposed by Rogers [3, 4, 5]. Some simulations are carried out to demonstrate the effects of the parameters on the performance of the algorithm. The results can be summarized as below:

- Correlation coefficient affects the estimation performance significantly. As the correlation coefficient becomes smaller, the state estimation performance increases,
- Measurement interval affects the estimation performance significantly. As the measurement interval becomes smaller, the state estimation performance increases,
- Increasing the disturbance noise variance degrades the performance significantly,
- Increasing the measurement noise variance degrades the performance significantly,
- Initial state variance does not affect the estimation performance very much.

Estimation performances of ODSA with correlated measurement noise proposed in this thesis are compared with Alpha-Beta Filter Algorithm with correlated measurement noise proposed by Rogers [3, 4, 5] for linear models. The advantages and the disadvantages of these algorithms can be summarized below:

- The most important advantage of ODSA is that it can be applied for both linear and nonlinear target tracking systems while Alpha-Beta Filter Algorithm can only be applied for linear target tracking systems.

- The estimation performance of ODSA is better than the Alpha-Beta Filter Algorithm as the correlation coefficient, initial state variance and sampling number parameters change.

- As the disturbance noise variance changes, for small values of disturbance noise variance: ODSA shows a better estimation performance, whereas for high disturbance noise variance values: Alpha-Beta Filter Algorithm shows a better estimation performance.
- As the measurement noise variance changes, for small values of measurement noise variance: Alpha-Beta Filter Algorithm shows a better estimation performance, whereas for high measurement noise variance values: ODSA shows a better estimation performance.
- Alpha-Beta Filter Algorithm is much faster than the ODSA algorithm, since the computational work of the Alpha-Beta Filter Algorithm algorithm is far less than that of the ODSA algorithm.

Additionaly estimation performances of ODSA with correlated measurement noise proposed in this thesis are compared with EKF with correlated measurement noise for two nonlinear models. The estimation performance of ODSA and EKF Algorithm change depending on the chosen nonlinear models. In this thesis comparision of two algorithms were done on two nonlinear models. The advantages and the disadvantages of these algorithms can be summarized below:

For the first nonlinear system model given in section 9.1:

- For small correlation coefficient, disturbance noise variance and measurement noise variance values, it can be said that generally EKF shows better performance than ODSA,
- For high correlation coefficient, disturbance noise and measurement noise values, it can be said that ODSA shows better performance than EKF,
- EKF is much faster than the ODSA algorithm, since the computational work of EKF is far less than that of the ODSA algorithm.

For the second nonlinear system model given in section 9.2:

- ODSA generally shows better performance than EKF as the correlation coefficient, disturbance noise variance, measurement noise variance, initial state variance and sampling time simulation parameters are changed,
- EKF is much faster than the ODSA algorithm, since the computational work of EKF is far less than that of the ODSA algorithm.

Finally, it can be said that ODSA can be implemented for correlated measurement noise models by applying proper modifications to the algorithm, with the advantage of being applicable to both linear and non-linear systems. Compared with the Alpha-Beta Filter Algorithm for the linear case, the computational work of ODSA is higher, but ODSA is generally more robust when estimation error performances are considered. Compared with the EKF for the nonlinear case, the computational work of ODSA is higher, but ODSA is generally more robust when estimation error performances are considered.

REFERENCES

[1] K. Demirbaş, *"Information Theoretic Smoothing Algorithms for Dynamic Systems with or without Interference: Advances in Control and Dynamic Systems"*, Vol. XXI, Academic Press, pp. 175-295, 1984.

[2] K. Demirbaş, *"Maneuvering Target Tracking with the Viterbi Algorithm in the Presence of Interference"*, Vol.136, No. 6, December 1989.

[3] S. Rogers, *"Alpha-Beta Filter with Correlated Measurement Noise"*, IEEE Trans. on Aerospace and Electronic Systems, Vol. AES-23, No. 4, pp. 592-594, July 1987.

[4] S. Rogers, *"Steady-State Kalman Filter with Correlated Measurement Noise-An Analytical Solution"*, IEEE Proc. of National Aerospace and Electronics Conf.(NAECON), May 1989.

[5] S. Rogers, *"Track filter for Correlated Measurements and Target Maneuvers"*, IEEE, 1990.

[6] S. Rogers, *"Accuracy of the Decoupled Kalman Filter"*, IEEE Transactions on Automatic Control, AC-31, pp.274-275, March 1986.

[7] W. Wu, D. Chang, *"Maneuvering Target Tracking with Colored Noise"*, IEEE Transactions on Aerospace and Electronic Systems, Vol.32, No. 4, October 1996.

[8] J. Guu, C. Wei, *"A Maneuvering Target Tracking System With Correlated Measurement Noises"*, IEEE International Radar Conference, 1990.

[9] A.W.Bridgewater, *"Analysis of Second and Third Order Steady-State Tracking Filters"*, AGARD Conference on Strategies for Automatic Track Initiation, Monterey, Calif., October,1978.

[10] W. D. Blair, J. E. Gray, M.D. Boyd, *"Design Analysis for the Two-stage Alpha, Beta, Gamma Estimator"*, IEEE 1991.

[9] Y. Zhu and P. R. Pagilla, *"A Note on the Necessary Conditions for the Algebraic Riccati Equation"*, IMA Journal of Mathematical Control and Information (22), pp.181-186, 2005.

[10] P. R. Kalata, K. R. Murpy, *"α-β Target Tracking with Track Rate Variations"*, IEEE Southeastern Symposium System Theory, pp. 70-74, 1997.

[11] M. I. Skolnik, *"Radar Handbook"*, McGraw Hill , New York, 1990.

[12] A. Papoulis, *"Probability, Random Variables, and Stochastic Process"*, McGraw Hill, New York, 1991.

[13] H. L. Van Trees, *"Detection Estimation and Modulation"*, Part 1, Wiley, New York, 1968.

[14] M. H. Hayes, *"Statistical Dıgital Signal Processing and Modelling"*, John Wiley & Sons,Inc., United States of America, 1996.

[15] Y. B. Shalom, T. E. Fortmann, *"Tracking and Data Association"*, Academic Press, Inc., San Diego, California, 1988.

[16] Y. B. Shalom, *"Multitarget-Multisensor Tracking: Priciples and Techniques"*, Box U-157, Storrs, 1995.

[17] Y. B. Shalom, X. R. Li, Thiagalingam Kirubarajan, "Estimation with Applications to Tracking and Navigation", John Wiley & Sons, Inc., United States of America, 2001.

[18] B. R. Mahafza, A. Z. Elsherbeni *"Matlab Simulations for Radar Systems Design"*, Chapman & Hall/CRC, United States of America, 2004.

[19] M. S. Grewal, A. P. Andrews, *"Kalman Filtering Theory and Practice Using Matlab"*, John Wiley & Sons, Inc., United States of America, 2001.

[20] G.Welch, G.Bishop, *"An Introduction to the Kalman Filter"*, SIGGRAPH 2001 Course 8, ACM Inc., USA, 2001.

[21] E. Gazioğlu, *"Target tracking with Input Estimation"*, A Master Thesis, METU, December 2005.

[22] Ö. Erdem, *"Chaotic Demodulation Under Interference"*, A Master Thesis, METU, September 2006.

[23] S. Ergüven, *"Path Extraction of Low SNR Dim Targets from Gray Scale 2-D Image Sequences"*, A Master Thesis, METU, September 2006.

APPENDIX A

APPROXIMATION OF A CONTINUOUS RANDOM VARIABLE WITH A DISCRETE RANDOM VARIABLE

In order to find the optimum discrete random variable with n possible values that approximates an absolutely continuous random variable x with distribution function $F_x(.)$, we must find a distribution function $F_{y_0}(.)$ which minimizes the objective function $J(.)$:

$$\begin{aligned} J(F_{y_0}(.)) &= \min_{F_y(.)} J(F_y(.)) \\ &= \min_{g(.)} J(g(.)) \end{aligned} \tag{A.1}$$

where

$$J(F_y(.)) = \int_{-\infty}^{\infty} \left[F_x(a) - F_y(a)\right]^2 da \tag{A.2}$$

The aim is to find a step function $g_0(.)$ which minimizes the objective function $J(.)$:

$$\begin{aligned} J(g(.)) = &\int_{-\infty}^{y_1} F_x{}^2(a) da + \int_{y_1}^{y_2} \left[F_x(a) - P_1\right]^2 da + \int_{y_2}^{y_3} \left[F_x(a) - P_2\right]^2 da + \ldots \\ &+ \int_{y_{n-1}}^{y_n} \left[F_x(a) - P_{n-1}\right]^2 da + \int_{y_n}^{\infty} \left[F_x(a) - 1\right]^2 da \end{aligned} \tag{A.3}$$

$$g_0(x)=\begin{cases}0, & x<y_{1,0},\\ P_{i,0}, & y_{i,0}\le x<y_{i+1,0}, \quad i=1,2,...,n-1\\ 1, & x\ge y_{n,0},\end{cases} \qquad \text{(A.4)}$$

If $g_0(x)$ is a step function that minimizes (A.3), it must satisfy the following set of equations:

$$\begin{aligned}
&P_{1,0}=2F_x(y_{1,0});\\
&P_{i,0}+P_{i+1,0}=2F_x(y_{i+1,0}), \qquad i=1,2,...,n-2;\\
&1+P_{n,0}=2F_x(y_n);\\
&P_{i,0}(y_{i+1,0}-y_{i,0})=\int_{y_{i,0}}^{y_{i+1,0}}F_x(a)da \qquad i=1,2,...,n-1
\end{aligned} \qquad \text{(A.5)}$$

The discrete random variables which approximate the normal random variable with zero mean and unity variance (with up to 8 possible values) are given by Demirbaş [1]. In order to increase the possible values of the discrete random variables, a Matlab function called *"discretegaussian"* is written which evaluates the values according to the equations given in (A.4) and (A.5) [13]. The program runs in a recursive manner and finds the discrete values (y values) and the corresponding probabilities (p values) of the continuous Gaussian distributed random variable with zero mean and unity variance. Finally, if the mean (μ) and the variance (σ) of the random variable are different from 0 and 1 respectively, it maps the new discrete values according to the mean and variance of the random variable by using the formula given in (A.6).

$$y'=\sigma y_{i,0}+\mu, \qquad P'_{i,0}=P_{i,0} \qquad i=1,2,...,n \qquad \text{(A.6)}$$

The y and p values of approximated x are given at Table 54.

Table-54 : *y* and *p* values of discrete random variables with 8 possible values

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
y	-1.6990	-1.0250	-0.5700	-0.1840	0.1840	0.5700	1.0250	1.6990
p	0.0922	0.1240	0.1394	0.1460	0.1460	0.1394	0.1240	0.0922

Possible values of the discrete random variable approximating the Gaussian random variable with zero mean and unity variance (y values):

N	*y value*							
1	0							
2	-0.675	0.675						
3	-1.0052	0	1.0052					
4	-1.2177	-0.3546	0.3546	1.2177				
5	-1.3767	-0.592	0	0.592	1.3767			
6	-1.4992	-0.7678	-0.2419	0.2419	0.7678	1.4992		
7	-1.6027	-0.9077	-0.4242	0	0.4242	0.9077	1.6027	
8	-1.6897	-1.0226	-0.5694	-0.1839	0.1839	0.5694	1.0226	1.6897

9	-1.7644	-1.1198	-0.6896	-0.3315	0	0.3315	0.6896	1.1198	1.7644
10	-1.8178	-1.1985	-0.7888	-0.4527	-0.1479	0.1479	0.4527	0.7888	1.1985
	1.8178								
11	-1.8799	-1.2737	-0.8779	-0.5575	-0.2716	0	0.2716	0.5575	0.8779
	1.2737	1.8799							
12	-1.9282	-1.3373	-0.9545	-0.6476	-0.377	-0.1239	0.1239	0.377	0.6476
	0.9545	1.3373	1.9282						
13	-1.9714	-1.3942	-1.0226	-0.727	-0.4688	-0.2301	0	0.2301	0.4688
	0.727	1.0226	1.3942	1.9714					
14	-2.0218	-1.4507	-1.0868	-0.7997	-0.5511	-0.3235	-0.1067	0.1067	0.3235
	0.5511	0.7997	1.0868	1.4507	2.0218				
15	-2.0449	-1.4918	-1.1387	-0.8611	-0.622	-0.4047	-0.1996	0	0.1996
	0.4047	0.622	0.8611	1.1387	1.4918	2.0449			
16	-2.0966	-1.5435	-1.1948	-0.9227	-0.6899	-0.4798	-0.2831	-0.0936	0.0936
	0.2831	0.4798	0.6899	0.9227	1.1948	1.5435	2.0966		
17	-2.1372	-1.5879	-1.2443	-0.9777	-0.7509	-0.5474	-0.3581	-0.1771	0
	0.1771	0.3581	0.5474	0.7509	0.9777	1.2443	1.5879	2.1372	

18	-2.1569	-1.6206	-1.2846	-1.0245	-0.804	-0.607	-0.4247	-0.2515	-0.0833
	0.0833	0.2515	0.4247	0.607	0.804	1.0245	1.2846	1.6206	2.1569
19	-2.196	-1.6609	-1.3285	-1.0725	-0.8564	-0.6642	-0.4872	-0.3199	-0.1585
	0	0.1585	0.3199	0.4872	0.6642	0.8564	1.0725	1.3285	1.6609
	2.196								
20	-2.2125	-1.6894	-1.3636	-1.113	-0.902	-0.7149	-0.5432	-0.3816	-0.2265
	-0.0751	0.0751	0.2265	0.3816	0.5432	0.7149	0.902	1.113	1.3636
	1.6894	2.2125							

APPENDIX B

SOLUTION OF THE RICCATI EQUATIONS FOR THE COVARIANCE MATRIX, P, OF ALPHA-BETA FILTER ALGORITHM

The Riccati equation for P can be solved explicitly, as follows:

Define a new state vector, $s'' = L\,s$, where

$$L = \begin{bmatrix} 1 & \gamma^T \\ 0 & 1 \end{bmatrix} \tag{B.1}$$

where $\gamma = \dfrac{a}{1-a}$.

The equation for $P'' = L\,P\,L^{'}$ is identical to Eq.(5.14), with *H, Γ* and *A* replaced by:

$$\begin{aligned} H'' &= HL^{-1} = \begin{bmatrix}(1-a) & 0\end{bmatrix} \\ \Gamma'' &= L\,\Gamma = \begin{bmatrix}(1+\gamma)T & 1\end{bmatrix}^{'} \\ A'' &= L\,A\,L^{-1} = A \end{aligned} \tag{B.2}$$

Writing P'' in component form [8],

$$P'' = r'' \begin{bmatrix} \alpha & \beta/T \\ \beta/T & c/T^2 \end{bmatrix} \tag{B.3}$$

where $r'' = \dfrac{R}{(1-a)^2} = r\dfrac{(1+a)}{(1-a)}$

One obtains from Eq. (5.14) the following algebraic equations for *a*, *b*, and *c*:

$$\frac{\alpha}{1-\alpha} = \alpha + 2\beta + c + \lambda^2 (1+\gamma)$$

$$\frac{\beta}{1-\alpha} = \beta + c + \lambda^2 (1+\gamma)$$

$$\frac{\beta^2}{1-\alpha} = \lambda^2$$

$$\lambda^2 = \frac{qT^3}{r^{"}} \tag{B.4}$$

Solving for (β) in terms of (α) and (*c*) yield the quartic equation:

$$f(x) = x^4 - \lambda x^3 - \left[2 + \lambda^2 \gamma (1+\gamma)\right] x^2 - \lambda x + 1 = 0 \tag{B.5}$$

where

$$\alpha = 1 - x^2$$

$$\beta = \lambda x$$

$$c = \frac{\alpha\beta}{1-\alpha} - \lambda^2 (1+\gamma) \tag{B.6}$$

Observe that *f(x)* is a symmetric polynomial, that is, the coefficients of x^i and x^{4-i} are equal. It follows that:

$$f(x) = (x^2 - d_1 x + 1)(x^2 - d_2 x + 1) = 0 \tag{B.7}$$

where d_1 and d_2 are the roots of

$$d^2 - \lambda d - \left[4 + \lambda^2 \gamma (1+\gamma)\right] = 0 \tag{B.8}$$

Eqs. (B.7) and (B.8) define all four roots of *f(x)*. In order to satisfy the physical constraints $0 \le \alpha \le 1$ and $c \ge 0$, it follows that the correct choice of root is that given in Eq.(5.16). Having found the components of $P^{''}$ by means of the transformation:

$$P^{''} = L^{-1} P^{''} L^{'-1}$$

$$L^{-1} = \begin{bmatrix} 1 & -\gamma^T \\ 0 & 1 \end{bmatrix} \tag{B.9}$$

This completes the solution.