CHAOTIC DEMODULATION UNDER INTERFERENCE

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

BY

ÖZDEN ERDEM

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

SEPTEMBER 2006

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) _____

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

İrfan OKŞAR (M.S) (ASELSAN, 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: Özden ERDEM

Signature :

ABSTRACT

CHAOTIC DEMODULATION UNDER INTERFERENCE

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

September 2006, 187 pages

Chaotically modulated signals are used in various engineering areas such as communication systems, signal processing applications, automatic control systems. Because chaotically modulated signal sequences are broadband and noise-like signals, they are used to carry binary signals especially in secure communication systems.

In this thesis, a target tracking problem under interference at chaotic communication systems is investigated. Simulating the chaotic communication system, noise-like signal sequences are generated to carry binary signals. These signal sequences are affected by Gaussian channel noise and interference while passing through the communication channel. At the receiver side, target tracking is performed using Optimum Decoding Based Smoothing Algorithm. The estimation performances of optimum decoding based smoothing algorithm at one dimensional chaotic systems and nonlinear chaotic algorithm map are presented and compared with the performance of the Extended Kalman Filter application.

Keywords: Chaotic Communication Systems, Optimum Decoding Based Smoothing Algorithm, Trellis Diagram, Estimation.

ÖZ

BOZUCU ETKİSİNDE DÜZENSİZ DEMODÜLASYON

ERDEM, Özden
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Kerim DEMİRBAŞ

Eylül 2006, 187 sayfa

Düzensiz (karmaşık) kodlanmış işaretler; haberleşme sistemleri, işaret işleme uygulamaları ve otomatik kontrol sistemleri gibi çeşitli mühendislik alanlarında kullanılmaktadır. Özellikle güvenli haberleşme sistemlerinde genişbantlı ve gürültü benzeri olmaları nedeniyle düzensiz (karmaşık) kodlanmış işaret dizileri, ikili işaretleri taşımak için kullanılırlar.

Bu tezde, bozucu etkisinde kalan düzensiz (karmaşık) haberleşme sistemlerinde hedef izleme incelendi. Düzensiz (karmaşık) haberleşme sistemleri benzetimlerinde ikili işaretleri taşımak için gürültü benzeri işaret dizileri oluşturuldu. Haberleşme kanalından geçen bu işaret dizilerine, Gauss kanal gürültüsü ve bozucu etkisi eklendi. Alıcı tarafında hedef belirleme işlemi, Optimum Kod Çözümüne Dayalı Düzeltme Algoritması kullanılarak gerçekleştirildi. Bir boyutlu düzensiz (karmaşık) sistemlerde ve doğrusal olmayan düzensiz (karmaşık) algoritma haritası üzerinde Optimum Kod Çözümüne Dayalı Düzeltme Algoritması'nın kestirim performansı incelendi ve Genişletilmiş Kalman Süzgeci performansı ile karşılaştırıldı.

Anahtar Kelimeler: Düzensiz (Karmaşık) Haberleşme Sistemleri, Optimum Kod Çözümüne Dayalı Düzeltme Algoritması, Kafes Diyagramı, Kestirim

To My Dear Family & Kayhan

ACKNOWLEDGMENT

I would like to express my sincere gratitude and appreciation to Prof. Dr. Kerim DEMİRBAŞ for his keen interest, guidance, endless patience, and support during this work. Also I would like to thank Prof. Dr. Kemal LEBLEBİCİOĞLU, Assoc. Prof. Dr. Fahrettin ARSLAN, Asist. Prof. Dr. Çağatay CANDAN and M. Sc. İrfan OKŞAR for serving on my committee.

I would also like to extend my gratitude 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.

Additionally, I thank Mr. Semih CAN and Mr. Cengiz EKEN for their guidance, valuable supports, endless patience and tolerance throughout this thesis work.

I would like to extend my special appreciation to my parents 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 Kayhan Çağlar KILIÇ for just being with me…

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The noise-like signals generated by deterministic chaotic systems have been successfully used in various engineering areas. These signals are typically broadband and similar to a stochastic process and therefore can be possibly used in secure communication applications, especially spread spectrum communication systems. [2]

In a chaotic communication, binary digital signals are modulated using chaotic waveforms. These modulated signals are transmitted through the Gaussian noise channel. At the receiver side, there are two estimators having the knowledge of the structure and possible parameters of chaotic systems. Estimators are employed to estimate states of chaotic systems. According to estimation error of two estimators, the decision regarding the transmitted signal can be made. [10]

There are three types of one-dimensional chaotic maps commonly used. Those are the skew tent map, the tent map and the symmetric tent map. These tent maps have a parameter *a*. For different values of the parameter *a*, binary signals can be presented over communication channels. Another example of the tent maps is the nonlinear chaotic algorithm (NCA) map. Unlike one-dimensional tent maps, NCA has two parameters to present binary signals. Moreover, the complexity and the sensitivity to initial states of NCA are much more complicated compared to one dimensional tent maps.

There are various estimation methods proposed for target tracking problem. One of these estimation methods is *optimum decoding based smoothing algorithm* [1] (ODSA) that obtains a trellis diagram for the target motion and estimates the

target track both in clear environment and in presence of interference. Another advantage of ODSA is that it can be used for linear and nonlinear models.

In this thesis, performance of ODSA on the chaotic systems are investigated. Algorithms are implemented in Matlab environment and some simulations are performed in order to evaluate the state estimation performance or bit error rate performance of this method on chaotic communication systems.

In Chapter 2, ODSA using Viterbi decoding algorithm for estimation problems is explained. The meanings of the parameters used in ODSA are given. Moreover, the complexity of the algorithm is analyzed.

In Chapter 3, simulation results for optimum decoding based smoothing algorithm are presented. The effects of the quantization levels and variances of ODSA parameters are discussed.

In Chapter 4, brief information is given about chaos and chaotic systems. Three commonly used one-dimensional chaotic maps and nonlinear chaotic algorithm map are explained. Effects of initial conditions and chaotic parameters on chaotic systems are shown by figures.

In Chapter 5, an application of ODSA on the symmetric tent map, which is one of the one-dimensional chaotic systems, is given. Estimation performance and complexity analysis of ODSA on the symmetric tent map are given in clear environment and under interference. Moreover, applications of ODSA on nonlinear chaotic algorithm (NCA) map are discussed considering BER performance.

In Chapter 6, an application of the EKF on the symmetric tent map is given in order to have a sense about performance of ODSA at chaotic systems. Estimation performance and complexity analysis of EKF on the symmetric tent map are given and compared with ODSA's.

In Chapter 7, the conclusions are given considering simulation results.

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 ODSA while obtaining the trellis diagram for the target motion model.

CHAPTER 2

A BRIEF INTRODUCTION TO OPTIMUM DECODING BASED SMOOTHING ALGORITHM

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

2.1. Models and Assumptions [1]

In Optimum Decoding Based Smoothing Algorithm, motion and observation models can be defined as below:

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

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

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

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,
- $u(k)$ is a $qx1$ input at time k with statistics,
- $w(k)$ is a $px1$ Gaussian distributed disturbance noise vector at time k with zero mean and known variance,
- $v(k)$ is an $lx1$ Gaussian distributed observation noise vector at time k with zero mean and known variance,
- $I(k)$ is an $mx1$ interference vector with mean and known variance,
- $z(k)$ is an $rx1$ observation vector at time k,
- Time k is time $t_0 + kT_0$ where t_0 and T_0 are the initial time and the observation 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 $\{x(0), x(1),\ldots, x(L)\}$ by using the observation sequence $\{z(1), z(2),\ldots, z(L)\}$ where l is a chosen integer.

2.2. Quantization of States and Transition Probabilities [1]

Since the main idea for ODSA is to quantize the states of the models to get a finite set, in this section a type of quantization for states is described.

Also another important factor, transition probabilities, that guides the target motion between these quantization states is defined and difficulties in calculating transition probabilities between quantization levels are mentioned.

Let the state $x(k)$ be a random vector whose range is in the space R^n (n dimensional Euclidian space). In the quantization for states, R^n is divided into nonoverlapping subspaces R_i^n and for each subspace R_i^n , there is a unique value x_{qi} assigned where the subscript q refer the 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.

In order to simplify the explanation of quantization, it can be thought that whenever a random state vector x(.) falls within a given subspace, say R_i^n, the state x(.) is quantized to the unique value x_{qi} .

The target motion between these gates, R_i^n, is based on the transition probabilities.

***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.,

$$\Pi_{jm}(k) \triangleq \text{Prob}\ \{\ x(k+1) \in R_m^n \mid x(k) \in R_j^n\ \} \qquad (2.3)$$

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

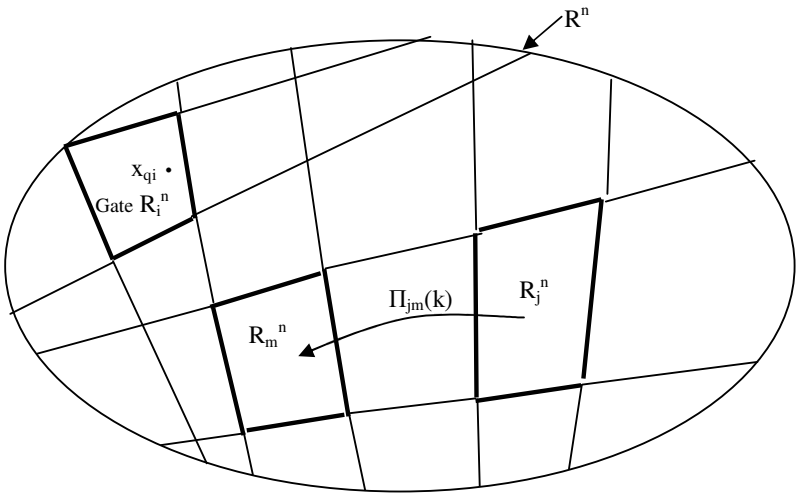

Figure 1 Quantization of states and transition probabilities

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

$$\Pi_{jm}(k) = \frac{Prob\left\{\chi(k+1) \in R_m^n, \chi(k) \in R_j^n\right\}}{Prob\left\{\chi(k) \in R_j^n\right\}} \tag{2.4}$$

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

$$= \left[\int_{R_j^n} p(\chi(k))d\chi(k)\right]^{-1} \times \left\{\int_{R_j^n}\left[\int_{R_m^n} p(\chi(k+1)\,|\,\chi(k))d\chi(k+1)\right]p(\chi(k))d\chi(k)\right\}$$

(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 $x(k)$.

As mentioned 2.1, the initial state vector $x(0)$ and $w(k)$ are Gaussian distributed and the random vectors $x(0)$, $w(k)$, and $w(l)$ are assumed to be statistically independent for all k, l. It is seen that $x(k+1)$ 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. Despite the fact that the motion model is linear, to evaluate the expression, $p\{\chi(k+1)\in R_m^n \mid \chi(k)\in R_j^n\}$, on (2.5) analytically is not easy because of the difficulties which arise from the shapes of the gates R_j^n and R_m^n and the statistics of the disturbance noise vectors $w(.)$ and the initial state vector $x(0)$. When the motion model is nonlinear, evaluation is more difficult.

To overcome these difficulties, an approximate state transition model is given in the next section. This model, called as the finite-state model, 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, … .

Using finite-state model, the transition probabilities can be easily calculated.

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

In the finite state model, there are generalized rectangles, which are called gates. In the center of these generalized rectangles, R_0^n, zero vector 0 (origin) is located.

Let the lengths of the sides of a generalized rectangle, say R_i^n, be g_{i1}, g_{i2} …, g_{in}. These lengths are said to be the sizes of gate R_i^n. Moreover, the quantization levels for gates are assumed the center points of the gates, namely,

$$\chi_q(.) \triangleq Q\{\chi(.)\} = \chi_{qi} \qquad \text{if } \chi(.)\in R_i^n, \qquad (2.6)$$

where x_{qi} is the center of the generalized rectangle (gate) R_i^n.

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 $x_{d1}(0)$, $x_{d2}(0)$, …, $\chi_{dn0}(0)$. The quantization numbers, m_k and n_0, are so important to approximate w(k) and x(k) satisfactorily to the discrete random vectors $w_d(k)$ and $x_d(0)$. 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 (2.6), 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 $x_{q1}(k)$, $x_{q2}(k)$,…, $\chi_{qnk}(k)$ where n_k is the number of possible quantization levels of the state vector $x(k)$. The quantization levels of $x(0)$ are assumed to 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 below,

$$\Pi_{jl}(k) \triangleq Prob\ \{\ x_q(k+1)=x_{ql} \mid x_q(k)=x_{qj}\ \}\ . \tag{2.8}$$

Assume that the quantized state vector $x_q(k)$ is equal to quantization level x_{qj} for gate R_j^n , (i.e., the target is in the R_j^n at time k.) The transitions from this quantization level to others are determined by the discrete random vector $w_d(k)$ and the function $Q(f(k, x_q(k)= x_{qj}, u(k), w_d(k))$. As mentioned before, $w_d(k)$ can take any value in the set $\{w_{d1}(k), w_{d2}(k), \ldots, w_{dm_k}(k)\}$ with the corresponding probabilities $p_{d1}(k)$, $p_{d2}(k)$, …, $p_{dm_k}(k)$. Therefore the quantized state vector $x_q(k+1)$ can take at most m_k 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., $\Pi_{jl}(k) = \sum_n p_{dn}(k) = p_{d1}(k) + p_{d2}(k)$.

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

2.4. Representing Target State Model By A Trellis Diagram [1]

Assuming the quantized state vector $x_q(k)$ has n_k possible values which are $x_{q1}(k), x_{q2}(k),\ldots, \chi_{qnk}(k)$, 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, … .
2. The transition from one quantization level to another is represented by a line having a direction indicating the direction of the state transition.

Hence, the state transition 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 state transition from time zero to time L.

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

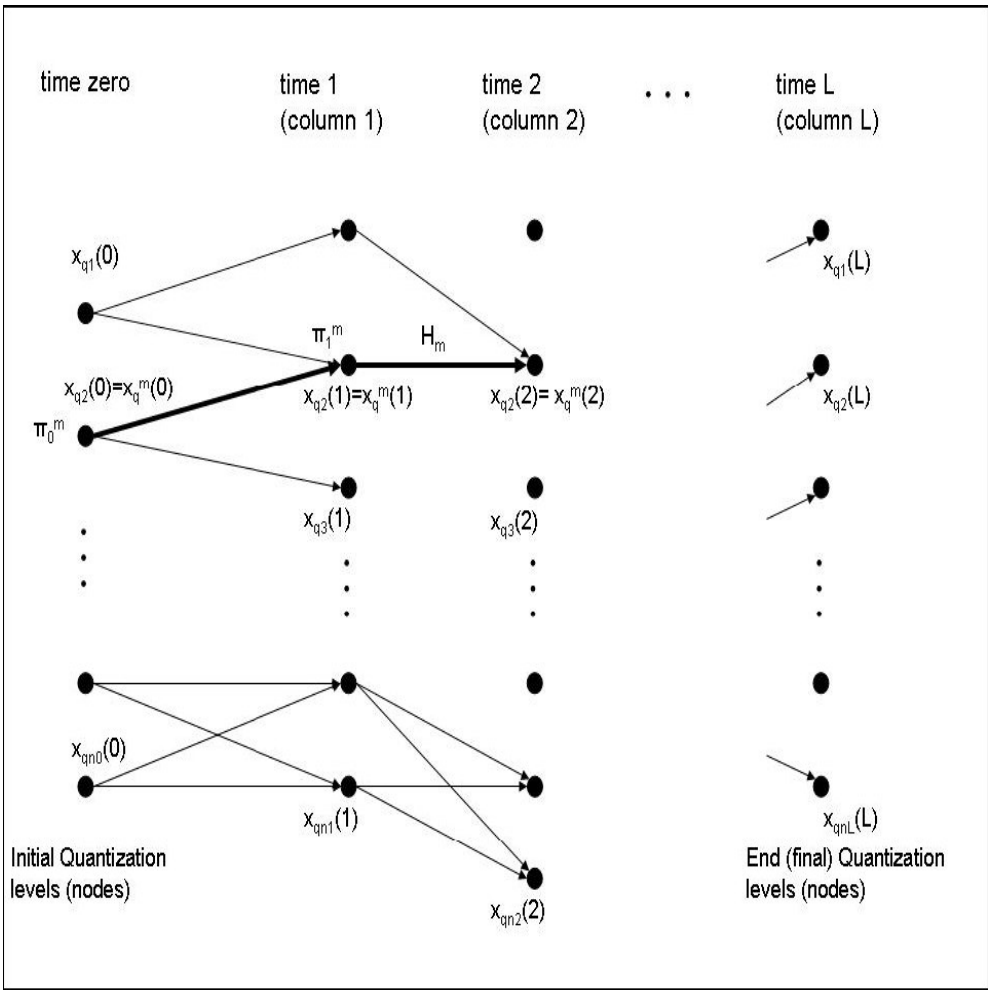

Figure 2 The trellis diagram for the state transition

2.5. Approximate Observation Models [1]

The target motion model has been reduced to a finite-state model that uses the quantized state vector $x_q(.)$ as described in the previous sections. However, the observation model in equation (2.1) and (2.2) uses the state vector $x(.)$. Thus, in the observation model in equation (2.1) and (2.2), by replacing the state vector x(k) with

the quantized state vector $x_q(.)$, the following approximate observation models are obtained:

$$z(k) = \begin{Bmatrix} g(k, x_q(k), v(k)) \\ g(k, x_q(k), I(k), v(k)) \end{Bmatrix} \begin{matrix} in\ clear\ environments, \\ in\ the\ presence\ of\ interference. \end{matrix} \qquad (2.9)$$

From now on, the equation in (2.9) is used to refer the observation 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 that are used for further analyses can be 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.

$\underset{\sim}{\chi}(i)$: Set of all the quantization levels for the gates in which the target may at time i, namely, $\underset{\sim}{\chi}(i) \triangleq \{x_{q1}(i), x_{q2}(i), \ldots, \chi_{qni}(i)$ where i = 0, 1, 2, …, L .

M : Number of possible paths through the trellis diagram. This number is equal to or less than

$$\prod_{j=0} n_j$$

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

$\chi_q^m(i)$: Quantization level for the gate in which the state vector lies at time k 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,

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

$\prod_0^m$: Probability that the possible value of the initial state vector $x_q(0)$ from which the m^{th} path starts occurs, namely,

$\prod_0^m$ =Prob{ $\chi_d(0) = \chi_q^m(0)$ }. For example, in trellis diagram of Figure 2, $\prod_0^m$ =Prob{ $\chi_q(0) = \chi_{q2}(0)$ }.

$\prod_i^m$: Transition probability from the (i-1)th gate for the mth path. In other words, it is the transition probability that the state vector will be at the ith 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 $\prod_i^m \triangleq$Prob{ $\chi_q(i) = \chi_q^m(i) \mid \chi_q(i-1) = \chi_q^m(i-1)$ }.

$\prod_0^{max}$: Maximum of the probabilities that the quantization levels at time zero occur.

$\prod_i^{max}$: 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).

$\prod_0^{min}$: Minimum of the probabilities that the quantization levels at time zero occur.

$\prod_i^{min}$: 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).

$\underset{\sim}{\chi}_L^m \triangleq \{ \chi_q^m(0), \chi_q^m(1), \ldots, \chi_q^m(L) \}$, Sequence of the quantization levels (nodes) which the mth path passes through; obviously, $\chi_q^m(i) \in \underset{\sim}{\chi}(i)$, where i =0, 1, 2, …, L

z^L = {z(1), z(2), …, z(L)} Observation 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 real state transition 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 real state variable by using the observation sequence z^L. Because of randomness in the models, our approach

must be statistical, i.e., a statistical optimization problem. Based on the observations, the path,which was (most likely) followed by the real state variable, will be guessed. Therefore, a criterion is needed. A suitable criterion may be the minimum error probability criterion. By using this criterion, the problem is reduced to find the path that is most likely followed by the real state variable to a multiple- (composite) 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 observation sequence, which hypothesis is true will be decided. A decision rule is developed which assigns each point in the observation space D to one of the hypotheses. The decision rule divides the whole observation space D into M subspaces D_1, D_2, …, D_m. If the observations 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 . The decision regions, therefore, 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

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

where

$$p'(z^L \mid H_j) = \left\{ \begin{array}{c} p(z^L \mid H_j) \\ \int_{I^L} p(z^L \mid H_j, I^L)\, p(I^L)\, dI^L \end{array} \right\} \begin{array}{l} \textit{in clear environment} \\ \textit{in the presence of interference} \end{array}, \tag{2.11}$$

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

$p(z^L|H_j)$: Conditional probability of the observation sequence z^L ($z(1), z(2),\ldots,z(L)$) given that hypothesis H_j is true ,

$p(z^L \mid H_j, I^L)$: Conditional probability of the observation 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;

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

2.7. Optimum Decision Rule for the Target Paths [1]

Consider the motion model in equation (2.7) and the observation 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 $j \neq k$, a priori probability of hypothesis H_i can be rewritten as:

$$p(H_i) = \prod_{k=0}^{L} \Pi_k^{\,i} \tag{2.13}$$

where $\Pi_k^i = prob(x_q(k)=x_q^i(k) | x_q(k-1)=x_q^i(k-1))$, and $x_q^i(k-1)$ *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

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

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

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

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

where

$$p'(z(k)|\chi_q^j(k)) = \begin{cases} p(z(k)|\chi_q^j(k)) \\ \int_{I(k)} p(z(k)|\chi_q^j(k), I(k)) \times p(I(k))dI(k) \end{cases} \begin{matrix} \text{in clear environment} \\ \text{in the presence of interference} \end{matrix} \quad (2.16)$$

and $p(z(k)|x_q{}^i(k))$ is the conditional probability of the observation z(k) in clear environments in Eq.(2.9) given that $x_q(k)=x_q^i(k)$, and $p(z(k)|\chi_q^i(k), I(k))$ is the conditional probability of the observation 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 $I_{d1}(k)$, $I_{d2}(k)$, …, $I_{drk}(k)$, with corresponding probabilities $p(I_{d1}(k))$, $p(I_{d2}(k))$, …,$p(I_{drk}(k))$, then the integral in Eq. (2.17) is reduced to a summation

$$\int_{I(k)} p(z(k)|\chi_q^i(k), I(k)) \times p(I(k))dI(k) \quad (2.17)$$

$$\approx \sum_{l=1}^{rk} p(z(k)|\chi_q^i(k), I_{dl}(k)) \times p(I_{dl}(k)), \quad (2.18)$$

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

$$z(k) = g(k, x_q(k), I(k)=I_d(k), v(k))$$
$$\triangleq g(k, x_q(k), I_d(k), v(k)). \quad (2.19)$$

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

Choose H_i if

$$\Pi_0{}^i \prod_{k=1}^{L} \Pi_k{}^i p'(z(k)|x_q{}^i(k))$$
$$> \quad \Pi_0{}^j \prod_{k=1}^{L} \Pi_k{}^j p'(z(k)|x_q{}^j(k)) \quad (2.20)$$

for all $j \neq i$.

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), we get the following:

Choose H_i if

$$\ln(\Pi_0^{\,i})+\sum_{k=1}^{L}\left\{\ln(\Pi_k^{\,i})+\ln\left(p'(z(k)\left|x_q^{\,i}(k)\right.)\right)\right\}>\ln(\Pi_0^{\,j})+\sum_{k=1}^{L}\left\{\ln(\Pi_k^{\,j})+\ln\left(p'(z(k)\left|x_q^{\,j}(k)\right.)\right)\right\} \quad (2.21)$$

for all $j\neq i$.

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

$$MN(x_{qi}(0)) = \ln [prob(x_q(0) = x_{qi}(0)] \quad (2.22)$$

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

$$M(x_{qj}(k-1) \rightarrow x_{qi}(k)) \triangleq \ln [prob(x_q(k)=x_{qi}(k)|x_q(k-1)=x_{qj}(k-1))] + \ln p'(z(k)|x_{qi}(k)) \quad (2.23)$$

Definition 2.9: The metric of a path from time 0 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 of.

Definition 2.10: The density function of the observation 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 method, 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. This method can be summarized by following steps:

Step 1. 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 2. For each node at time 1, using the observation *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 3. For each node at time k, using the observation $z(k)$, 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.

Step 4. 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.

2.9. An Example of ODSA [1]

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

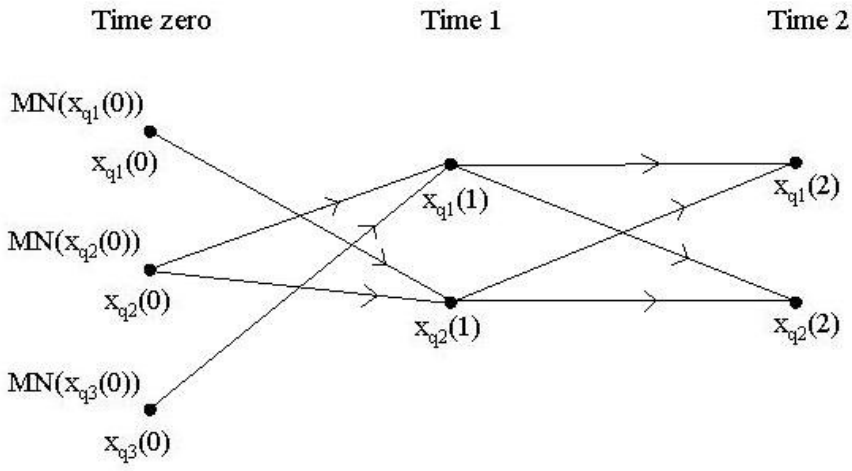

Figure 3 Trellis diagram for the target motion from time zero to time 2

Step 1. 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 2. 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:

$$A_{11} \triangleq M(x_{q2}(0) \rightarrow x_{q1}(1)) + MN(x_{q2}(0)), \tag{2.24}$$

$$A_{12} \triangleq M(x_{q3}(0) \rightarrow x_{q1}(1)) + MN(x_{q3}(0)). \tag{2.25}$$

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 Step 2, Figure 4 is obtained.

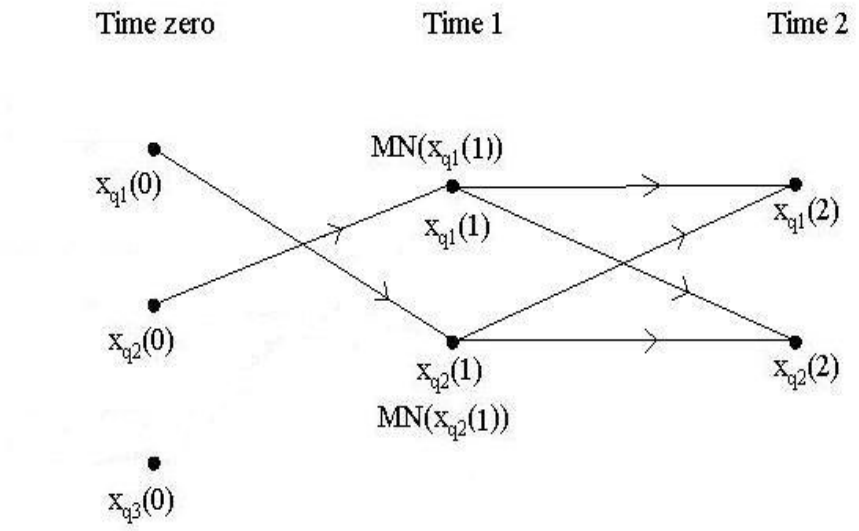

Figure 4 Trellis diagram for the target motion from time zero to time 2 at the end of first step

Step 3. 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:

$$A_{21} \triangleq M(x_{q1}(1) \rightarrow x_{q1}(2)) + MN(x_{q1}(1)), \tag{2.26}$$

$$A_{22} \triangleq M(x_{q2}(1) \rightarrow x_{q1}(2)) + MN(x_{q2}(1)). \tag{2.27}$$

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 Step 3. 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.

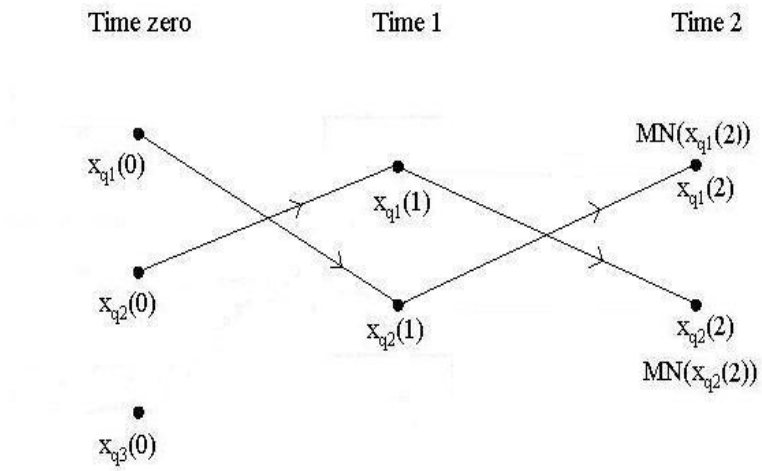

Figure 5 Trellis diagram for the target motion from time zero to time 2 at the end of second step

2.10. The Metric of A Branch [1]

The calculation of a metric of a branch depends on applications effected by the arbitrary random interference or not.

Considering a target model with only Gaussian disturbance and observation noises as described in section 2.1, the motion and observation models can be rewritten as

$$\begin{aligned} &\textit{Motion model}, && x(k+1) = f(k, x(k), u(k), w(k)) \\ &\textit{Observation model}, && z(k) = g(k, x(k)) + v(k) \end{aligned} \tag{2.28}$$

where x(0) is an n x 1 initial state Gaussian random vector with mean m_0 and covariance R_0; w(k) is a p x 1 Gaussian disturbance noise vector with zero mean and covariance $R_w(k)$; x(k) is an n x 1 state vector at time k; z(k) is an r x 1 observation vector at time k; f(k, x(k), w(k)) and g(k, x(k)) are linear or nonlinear vectors with appropriate dimensions; and v(k) is an r x 1 Gaussian observation noise vector with zero mean and covariance $R_v(k)$. Moreover, the random vectors x(0), w(j), w(k), v(l) and v(m) are assumed to be independent for all j, k, l, m.

The z(k) in the observation model given in (2.28) is a linear function of the Gaussian observation noise v(k). Hence given that $x(k)=x_q^i(k)$, the conditional probability density function of z(k) is a multivariate Gaussian density function. Thus, we have

$$p(z(k) \mid \chi_q^i(\mathrm{k})) \triangleq p(z(k) \mid \chi(\mathrm{k}) = \chi_q^i(k)) \tag{2.29}$$

$$= \frac{\exp -\left\{ \left[z(k) - g(k, \chi_q^i(k))\right]^T R_v^{-1}(k) \left[z(k) - g(k, \chi_q^i(k))\right] / 2 \right\}}{(2\pi)^{r/2} \left[\det R_v(k)\right]^{1/2}}$$

Substituting Eq. (2.29) into Eq. (2.23) yields the metric of the branch between the nodes $\chi_q^i(k-1)$ and $\chi_q^i(k)$; that is

$$\mathrm{M}(\chi_q^i(k-1) \rightarrow \chi_q^i(k))$$

$$\ln(\Pi_k^i) - \ln\left|(2\pi)^{r/2}\left[\det R_v(k)\right]^{1/2}\right| = -\left\{\left[z(k) - g(k, \chi_q^i(k))\right]^T R_v^{-1}(k)\left[z(k) - g(k, \chi_q^i(k))\right]/2\right\} \quad (2.30)$$

Considering a target model with the interference, Gaussian disturbance and observation noises as described in section 2.1, the motion and observation models can be rewritten as

$$\textit{Motion model}, \quad x(k+1) = f(k, x(k), u(k), w(k)) \quad (2.31)$$

$$\textit{Observation model}, \quad z(k) = g(k, x(k), I(k)) + h(k, x(k), I(k)) * v(k))$$

where x(0), x(k), u(k), w(k), z(k) and f(k, x(k), u(k), w(k)) are described in section 2.1. g(k, x(k), I(k)) and h(k, x(k), I(k)) are r x 1 and r x *l*-dimensional (linear or nonlinear) matrices, respectively; v(k) is an *l* x 1 Gaussian observation noise vector with zero mean and covariance $R_v(k)$; and I(k) is an m x 1 interference vector with known statistics. Furthermore, the following assumptions are applied:

1. The random vectors x(0), w(j), w(k), v(*l*), v(m), I(n), and I(p) are independent for all j, k, l, m, n, p.
2. $[h(k, x(k), I(k)) \times R_v(k) \times h^T(k, x(k), I(k))]^{-1}$ exists for all k.

The observation z(k) is a linear function of the normal observation vector v(k). Therefore, the conditional probability density function of z(k), given that $x(k)=x_q^i(k)$ and I(k), is multivariate normal density function, namely,

$$p(z(k) \mid \chi_q^i(k),\ I(k)) \triangleq p(z(k) \mid \chi(k) = \chi_q^i(k), I(k)) = A \exp\text{-}(B/2) \quad (2.32)$$

where

$$A \triangleq (2\pi)^{-r/2}\left\{\det\left[h(k, \chi_q^i(k), I(k)) R_v(k) h^T(k, \chi_q^i(k), I(k))\right]\right\}^{-1/2}$$

$$B \triangleq x\left[h(k, \chi_q^i(k), I(k)) R_v(k) h^T(k, \chi_q^i(k), I(k))\right]^{-1} \quad (2.33)$$
$$x\left[z(k) - g(k, \chi_q^i(k), I(k))\right]^T$$

From Eq. (2.17) and Eq. (2.18) , $p'(z(k)\left|x_q^{\,i}(k)\right.)$

$$= \begin{cases} \int_{I(k)} p(z(k) \mid \chi_q^i(k), I(k)) \times p(I(k)) dI(k) \\ \sum_{l=1}^{r_k} p(z(k) \mid \chi_q^i(k), I_{dl}(k)) \times p(I_{dl}(k)) \end{cases} \begin{matrix} using\ Eq.\ (2.17) \\ using\ Eq.\ (2.18) \end{matrix} \qquad (2.34)$$

where $p(z(k) \mid \chi_{\mathrm{q}}^{\mathrm{i}}(k), I_{dl}(k)) \triangleq p(z(k) \mid \chi_{\mathrm{q}}^{\mathrm{i}}(k), I(k) = I_{dl}(k))$ which is given in Eq. (2.32). Substituting Eq.(2.34) into Eq. (2.23) yields the metric of the branch between the nodes $\chi_{\mathrm{q}}^{\mathrm{i}}(k-1)$ and $\chi_{\mathrm{q}}^{\mathrm{i}}(k)$; that is

$$\mathrm{M}(\chi_{\mathrm{q}}^{\mathrm{i}}(k-1) \rightarrow \chi_{\mathrm{q}}^{\mathrm{i}}(k)) = ln\,\Pi_k^i + \ln p'(z(k) \mid \chi_q^i(k)) \qquad (2.35)$$

2.11. Complexity Analysis of ODSA

A program that uses ODSA to estimate the best path with the largest metric is written in MATLAB. In this program, for simplicity the vectors *x(k), w(k), z(k), u(k), v(k)* and *I(k)* are chosen to be one-dimensional vectors.

Runtime of the program written for ODSA is determined by the number of states and the number of quantization levels of *I(k)*. The number of states is determined by time *L*, the gate size and the number of quantization levels of *x(0), u(k)* and *w(k)*. If the number of states at time *k=0* to *k=L* is not limited, then the possible maximum state number for each time step, which is the worst case, will be $n_x(n_w n_u)^k$ at time *k*. Including time consumption on *I(k),* let the maximum time consumption at each state be t_s. Then, since the time consumption at each state will be approximately same, the time consumption at each time *k* will be maximum $t_s n_x(n_w n_u)^k$, which is the multiplication of the state number at time *k* and t_s. The total program runtime will be equal to:

$$Runtime= \left[t_s n_x + t_s n_x (n_w n_u)^1 + t_s n_x (n_w n_u)^2 + \ldots + t_s n_x (n_w n_u)^L\right] = t_s n_x \frac{(n_w n_u)^{L+1} - 1}{(n_w n_u) - 1} \quad (2.36)$$

where

t_s : maximum time consumption at each state,

n_x : the number of quantization levels of *x(0)*,

n_w : the number of quantization levels of *w(k).*

n_u : the number of quantization levels of *u(k).*

Since $(n_w n_u)^{L+1} >> 1$, the value given at equation (2.36) can be approximated as $t_s n_x \frac{(n_w n_u)^{L+1}}{n_w n_u - 1}$. Rewriting this equation, we obtain $\frac{t_s n_x n_w n_u}{n_w n_u - 1}(n_w n_u)^L$.

This means, the program runtime increases exponentially as time *L* increases. Therefore, if the maximum state number is not limited, the program complexity will increase too much which causes difficulties on the application of this algorithm. However, due to the gate size, some of the states will fall into the same gate and some of the states will be discarded as described at section 2.10. So, most of the time, the state number at time *k* will be smaller than $n_x(n_w n_u)^k$, and the total state number actually will be smaller than the value given at equation (2.36). In fact, this will be the upper bound for computing the program runtime.

If the number of states is limited by some value at each time *k*, the program runtime for the worst case will be:

$$Runtime= \quad t_s n_s L \quad (2.37)$$

where

t_s : maximum time consumption at each state,

n_s : the maximum number of states allowed at time from *k=0 to* time *k=L*,

From (2.37) we can say that program runtime increases linearly as time *L* increases. Therefore, if the maximum state number is limited, the program complexity will not increase as k increases.

Some simulations are performed to measure the program runtime for both limited and not limited state number cases.The simulations are performed for different values of gate size and quantization numbers of *x(0) u(k)* and *w(k)* to obtain various state numbers.The results are analyzed at subsections 2.11.1 and 2.11.2.

In these subsections, Figure 6 , Figure 7, Figure 8 and Figure 9 are obtained. In these figures,

- "nx" refers quantization level of the initial state x(0),
- "nu" refers quantization level of the input vector u(k),
- "nw" refers quantization level of the disturbance noise vector w(k),
- "L" refers the number of samples.

2.11.1 Program Runtime When the State Number is not Limited

In this section, the program runtime is analyzed for the case that the state number is not limited. Motion and observation models are

Motion model : x(k+1) = x(k) + w(k) + u(k),

Observation model : z(k) = x(k) + v(k) + I(k).

Parameters used in these simulations are as follows;

- variance of x(0)=1, expected value of x(0)=0,
- variance of u(k)=1, expected value of u(k)=0,
- variance of w(k)=1, expected value of w(k)=0,
- variance of I(k)=0.01, expected value of I(k)=0, quantization level of I(k)=3.

The gate size and the quantization levels of parameters are presented in Figure 6 and Figure 7.

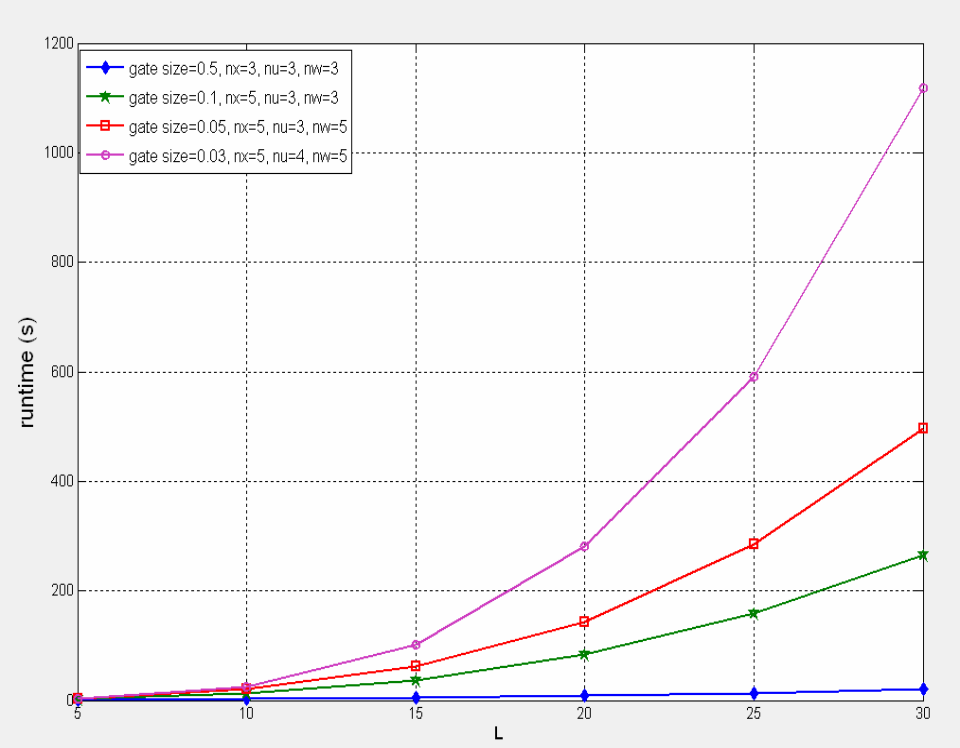

Figure 6 Program runtime when the state number is not limited

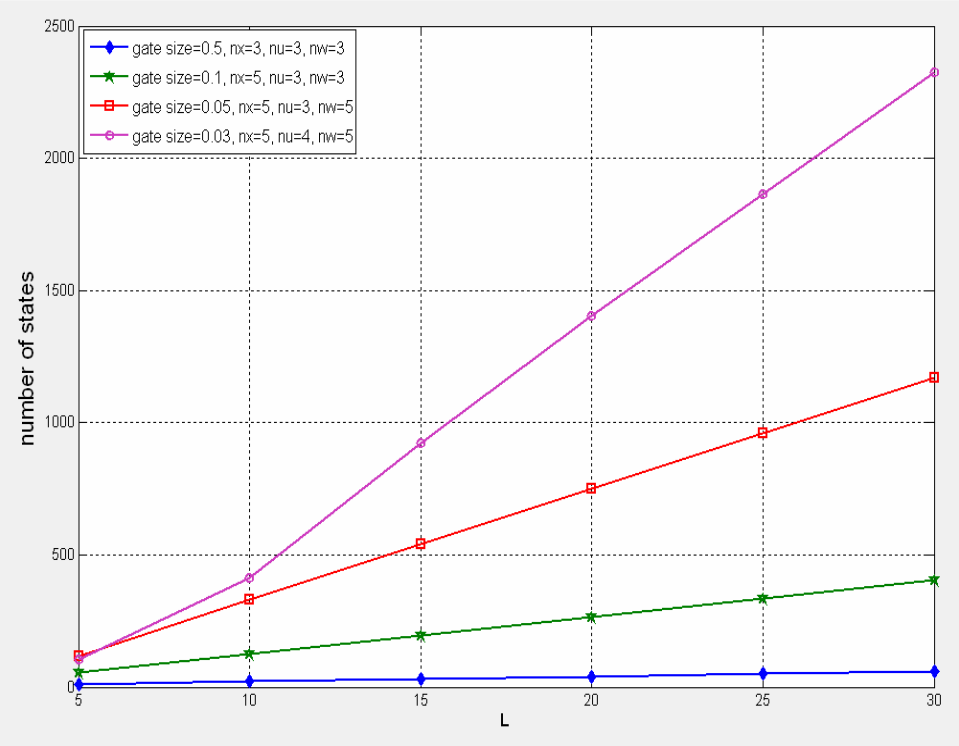

Figure 7. The number of states when the state number is not limited

Analyzing Figure 7, it can be observed that the state number depends on the quantization level of x(0), u(k), w(k) and the gate size. As the quantization levels of these parameters increases and the gate size decreases, the state number increases dramatically. Depending on the state number, the program runtime exponentially increases as L increases as shown Figure 6, whereas the state number increases linearly.

2.11.2 Program Runtime When the State Number is Limited

In this section, the program runtime is analyzed for the case that the state number is limited. Motion and observation models are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters used in these simulations are as follows;

- variance of x(0)=1, expected value of x(0)=0,
- variance of u(k)=1, expected value of u(k)=0,
- variance of w(k)=1, expected value of w(k)=0,
- variance of I(k)=0.01, expected value of I(k)=0, quantization level of I(k)=3,
- maximum number of the states=100.

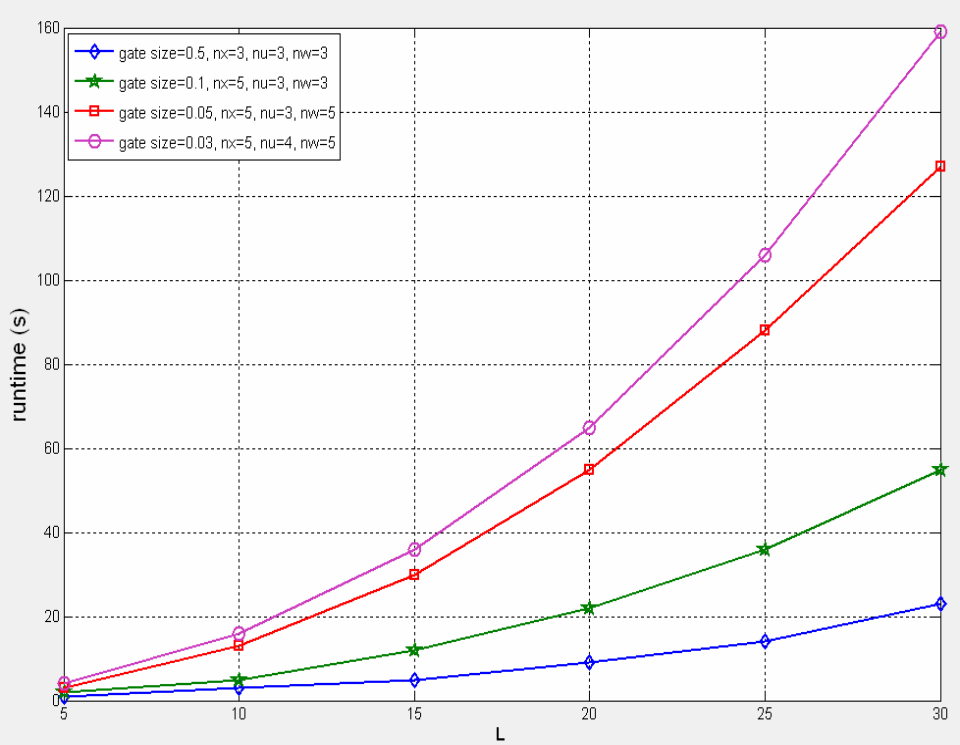

Figure 8 Program runtime when the state number is limited to 100

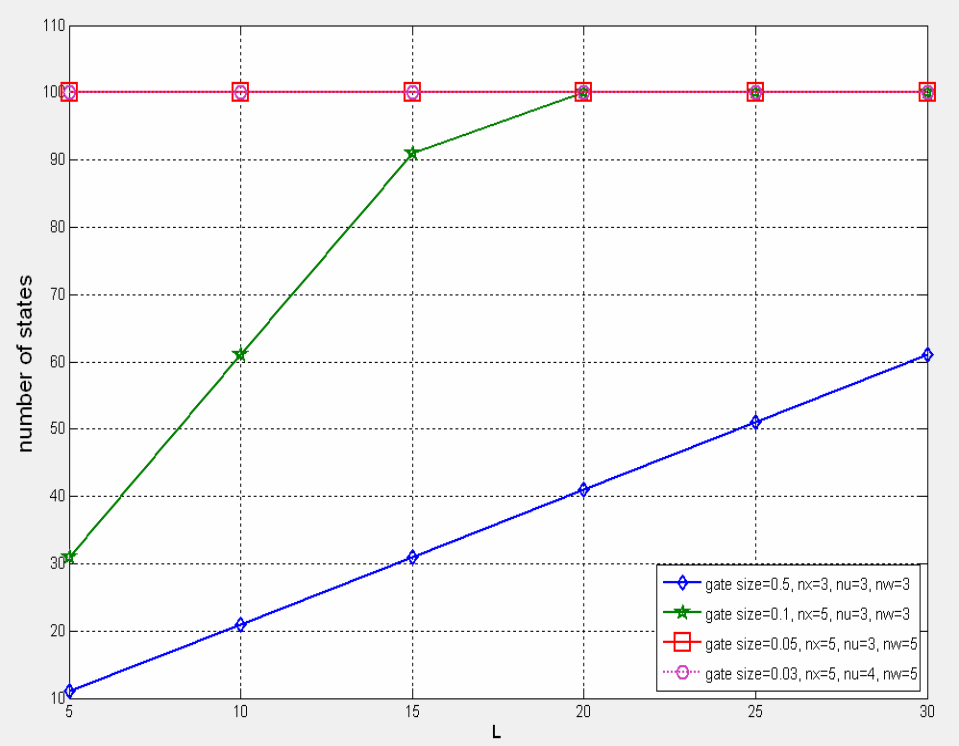

Figure 9 The number of the states when the state number is limited to 100

When Figure 8 is analyzed, it can be observed that the program runtime is increasing with *L*. But, type of increase on runtime can be defined as "Linear" comparing with Figure 6. This is expected, because the state number is limited to 100. When Figure 9 is analyzed, it can be observed that the state number reaches to the limit faster as the quantization level of the parameters increases.

CHAPTER 3

SIMULATION RESULTS OF ODSA

Throughout CHAPTER 3, effects of the parameters, which are used in ODSA, are detaily analyzed. These parameters are;

- The gate size,
- The quantization number of initial state vector, x(0)
- The quantization number of disturbance noise vector, w(k)
- The quantization number of input vector, u(k)
- The initial state variance,
- The disturbance noise variance,
- The input u(k) variance,
- The observation noise variance,
- The quantization number of interference vector, I(k)
- The interference variance,
- The limiting of the maximum state number.

Simulations are performed for one linear and one nonlinear model. First, analyses are studied for the models in clear environments, which do not include the interference vector. Then, the interference vector will be added to the models as simulating the environments in presence of interference. In this way, the variation in the performance will be studied.

To check the performance of the algorithm and the effects of the parameters, actual target path should be simulated. Therefore, observation vector is generated.

Assuming *x(0), w(k), u(k), I(k)* and *v(k)* vectors are the Gaussian distributed random vectors, these vectors are generated using the *"randn(.)"* command of *MATLAB* with appropriate mean and variances. Putting these values to the motion and observation models, the actual values of *x(k)* and *z(k)* values are obtained.

The simulations are obtained after 1000 execution. For each execution, the state vector *x(k)* and the observation vector *z(k)* are regenerated randomly with the same motion and observation equations using independent random vectors *x(0), w(k), u(k), v(k)* and *I(k)*.

From now on in the figures,

- "Q # x(k)" refers the quantization number of x(k),
- "σ^2(x(k))" refers the variance value of x(k).

3.1. Effect of the Gate Size

As explained 2.3, the state number and the value of the states at time k are affected by the gate size indirectly. Due to these effects, the performance of the algorithm varies. There are two models, which are linear and nonlinear.

Figures for each model show RMS error variation depending on the gate size. This error, $z(k)-\hat{x}_q(k)$, is calculated for each time k at 1000 simulations, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k)$.

Parameters are;

number of samples, L = 20,

number of max states = 100,

quantization numbers : x(0) = 5, w(k) = 3, u(k) = 3 ,

variances : x(0) = 1, w(k) = 1, u(k) = 1, v(k) = 1 ,

expected values : x(0) = 0, w(k) = 0, u(k) = 0 ,v(k) = 0 .

For the linear model, Figure 10 is obtained.

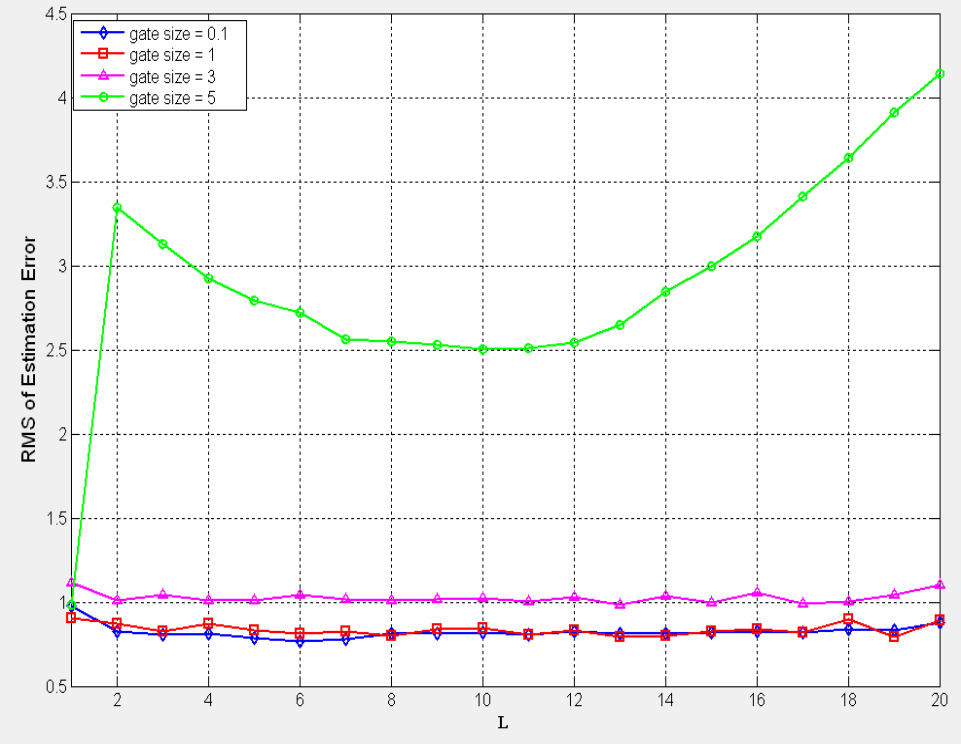

Figure 10 RMS estimation error for the linear model as gate size changing

Table 1 Average values of RMS Estimation error as gate size changing for the linear model

Gate size	Average values of RMS Estimation Error
0.1	0.8236
1	0.8366
2	1.0263
5	2.8931

Nonlinear models used in this simulation are

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

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

Parameters are same used in linear models. For the nonlinear model, Figure 11 is obtained

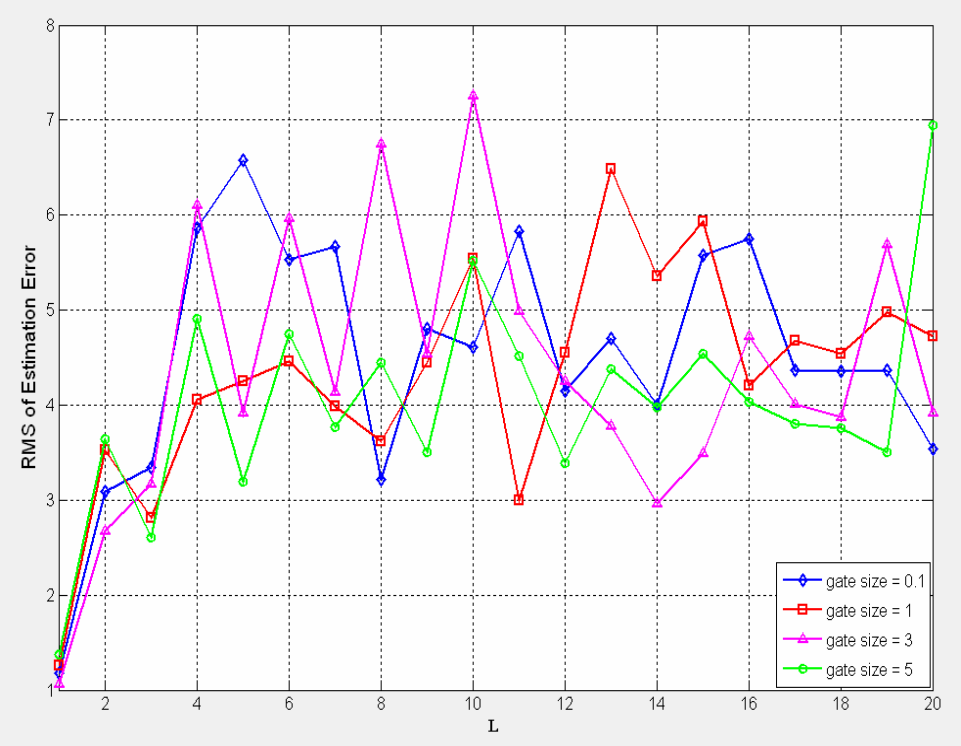

Figure 11 RMS estimation error for the nonlinear model as gate size changing

Table 2 Average values of RMS Estimation error as gate size changing for the nonlinear model

Gate size	Average values of RMS Estimation Error
0.1	4.5266
1	4.3218
2	4.3648
5	4.0279

Comment:

The effect of the gate size is observed more clearly in the linear model. Figure 10 shows that the gate size is directly proportional with the estimation error. Increase on the gate size causes estimation error to become larger. Figure 11 shows that increase on gate size does not affect estimation error too much in nonlinear model.

3.2. Effect of the Quantization Number of the Initial State Vector

In this section, effects of the quantization number of the initial state vector are explained. There are two models, linear and nonlinear models.

Figures show RMS error variation depending on the quantization number of the initial state vector. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 1,

number of max states = 100 ,

quantization numbers : w(k) = 3, u(k) = 3 , I(k) = 3,

variances : x(0) = 1,w(k) =1,u(k) = 1,v(k) = 1, I(k) = 0.01,

expected values : x(0) = 0, w(k) = 0,u(k) = 0, v(k) = 0, I(k) = 0.2

For the linear model, using these parameters Figure 12 is obtained.

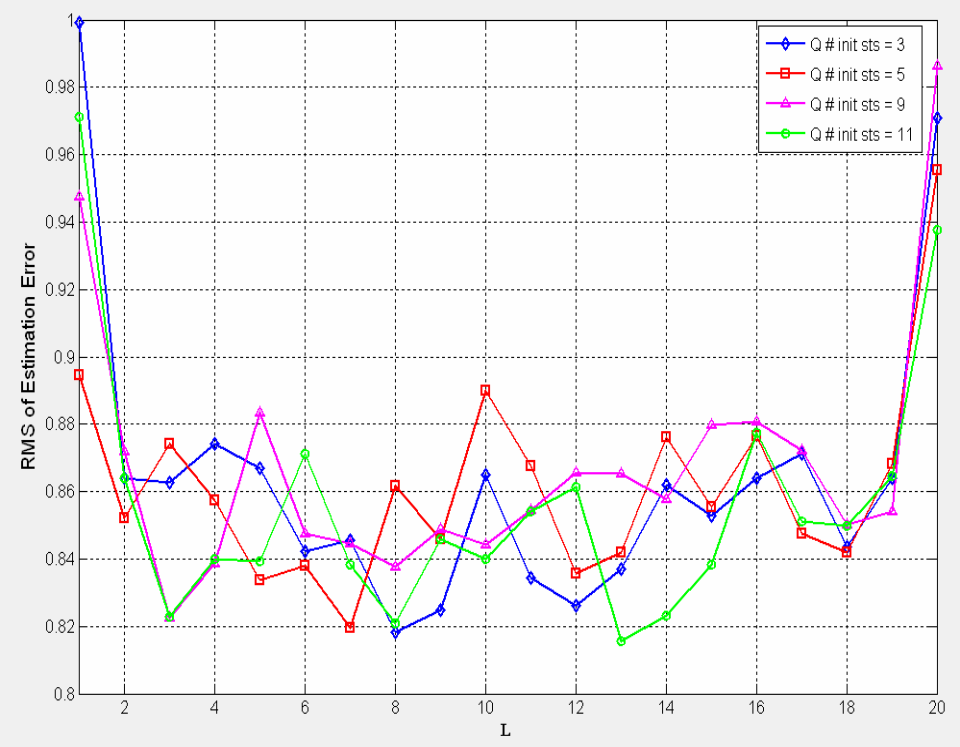

Figure 12 RMS estimation error for the linear model as quantization level of x(0) changing

Table 3 Average values of RMS Estimation error as quantization level of x(0) changing for the linear model

Quantization level of x(0)	Average values of RMS Estimation Error
3	0.86445
5	0.8617
9	0.86767
11	0.85636

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 13 is obtained.

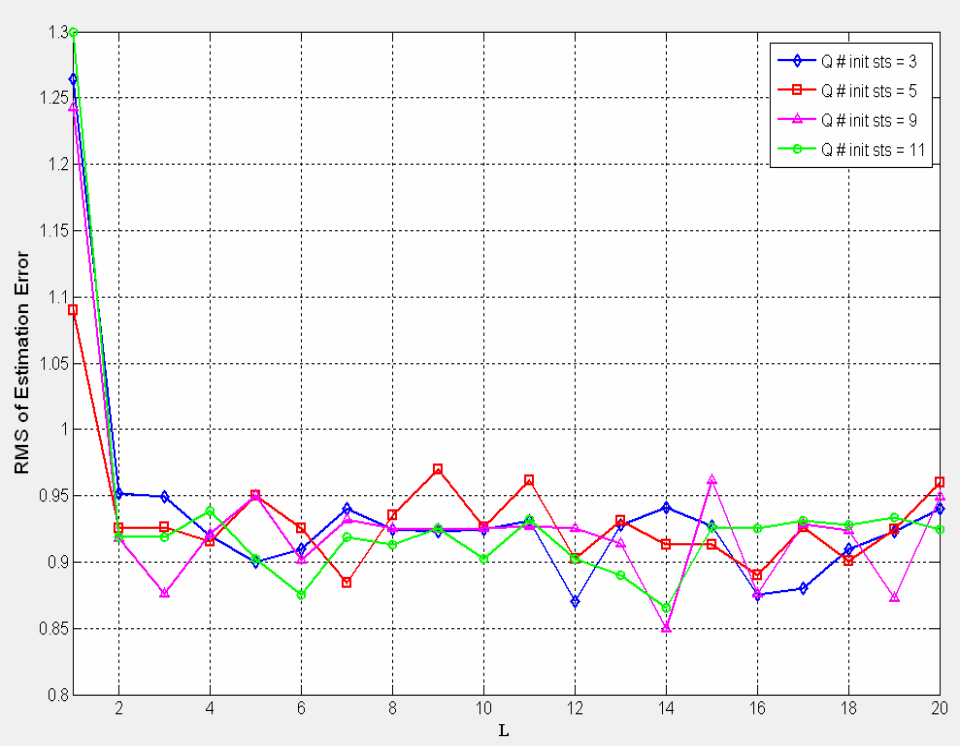

Figure 13 RMS estimation error for the nonlinear model as quantization level of x(0) changing

Table 4 Average values of RMS Estimation error as quantization level of x(0) changing for the nonlinear model

Quantization level of x(0)	Average values of RMS Estimation Error
3	0.9366
5	0.9336
9	0.9323
11	0.9335

Comment:

From Figure 12 and Figure 13, 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.

3.3. Effect of the Quantization Number of the Disturbance Noise

In this section, effects of the quantization number of the disturbance noise vector are explained. There are two models, linear and nonlinear models.

Figures show RMS error variation depending on the quantization number of the disturbance noise vector. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, u(k) = 3 , I(k) = 3,

variances : x(0) = 1, w(k) =1,u(k) =1,v(k) = 1, I(k) = 0.01,

expected values : x(0) = 0, w(k) = 0, u(k)=0, v(k) = 0, I(k) = 0.2.

For the linear model, Figure 14 is obtained.

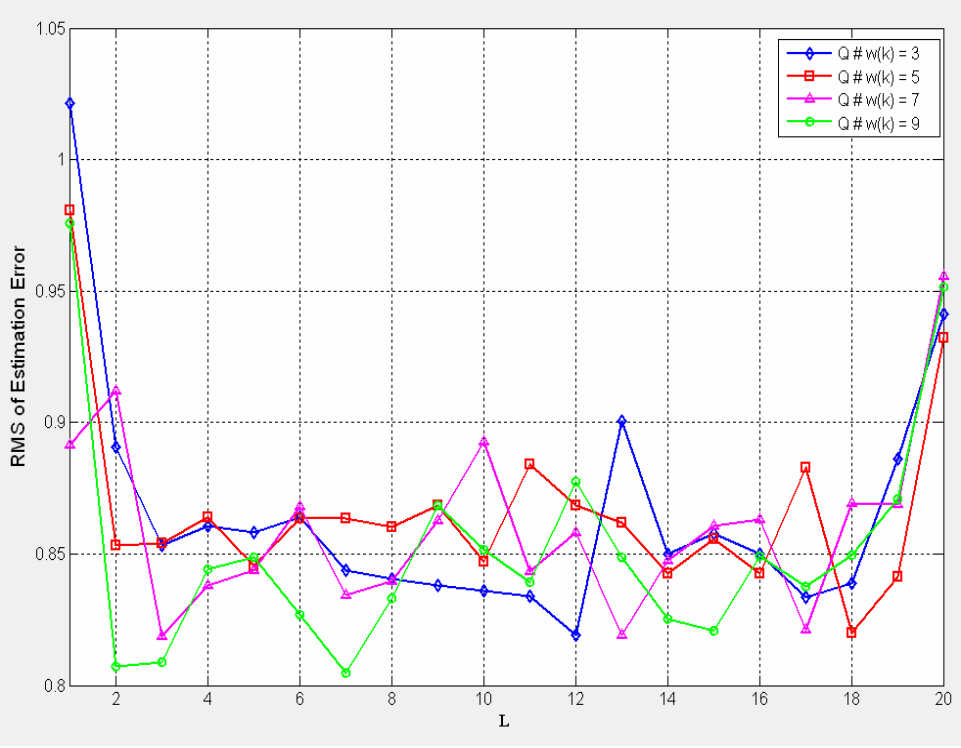

Figure 14 RMS estimation error for the linear model as quantization level of w(k) changing

Table 5 Average values of RMS Estimation error as quantization level of w(k) changing for the linear model

Quantization level of w(k)	Average values of RMS Estimation Error
3	0.8658
5	0.8665
7	0.8604
9	0.8519

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 15 is obtained.

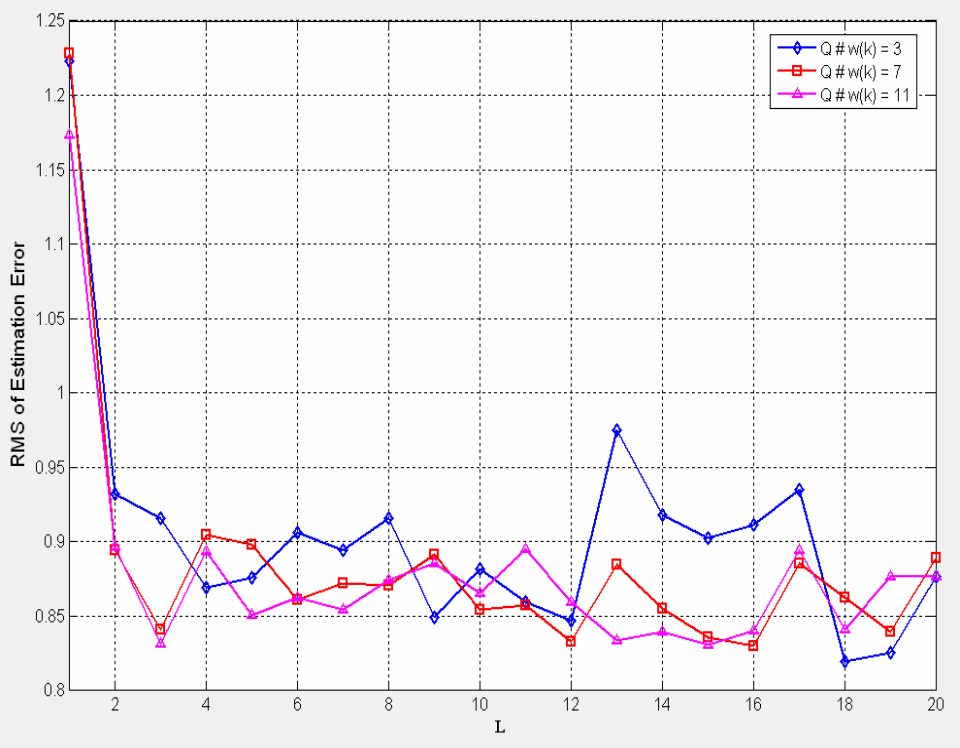

Figure 15 RMS estimation error for the nonlinear model as quantization level of w(k) changing

Table 6 Average values of RMS Estimation error as quantization level of w(k) changing for the nonlinear model

Quantization level of w(k)	Average values of RMS Estimation Error
3	0.90667
7	0.88445
13	0.87881

Comment:

It can be observed from the figures that increasing the number of the quantization levels of the disturbance noise *w(k)* slightly improves the state estimation error (Figure 14 and Figure 15) both linear and nonlinear models.

3.4. Effect of the Quantization Number of the Input Vector

In this section, effects of the quantization number of the input vector are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the quantization number of the input vector. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, w(k) = 3, I(k) = 3,

variances : x(0) = 1,w(k) =1, u(k)=1,v(k) = 1, I(k) = 0.01,

expected values : x(0) = 0, w(k) =0, u(k) =6, v(k) = 0, I(k) = 0.2.

For the linear model, Figure 16 is obtained.

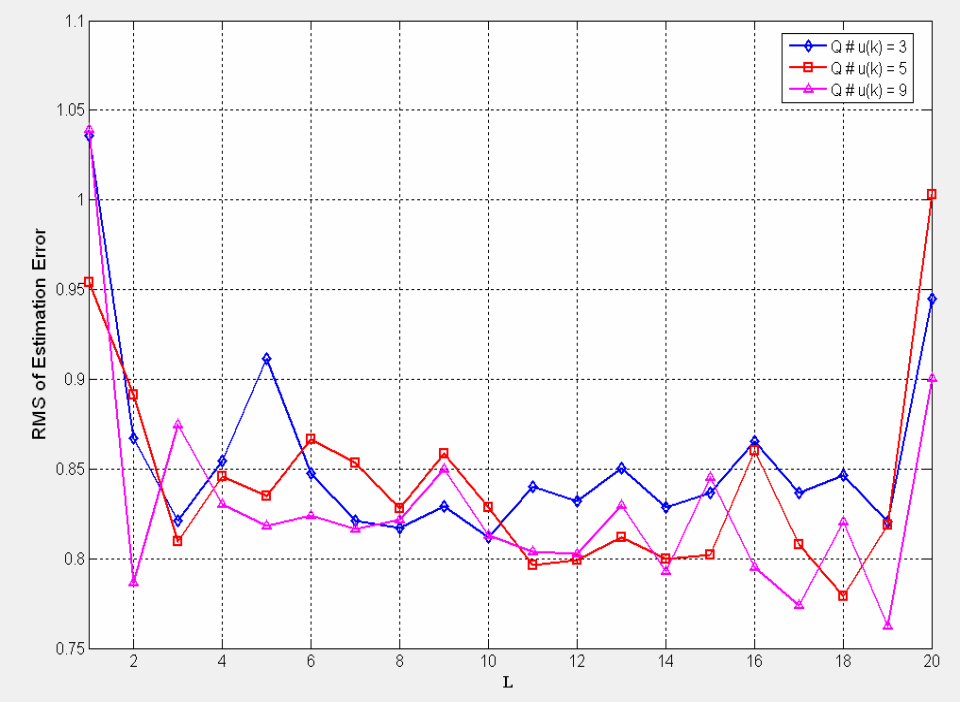

Figure 16 RMS estimation error for the linear model as quantization level of u(k) changing

Table 7 Average values of RMS Estimation error as quantization level of u(k) changing for the linear model

Quantization level of u(k)	Average values of RMS Estimation Error
3	0.8558
5	0.8424
9	0.8299

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1 - |x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 17 is obtained.

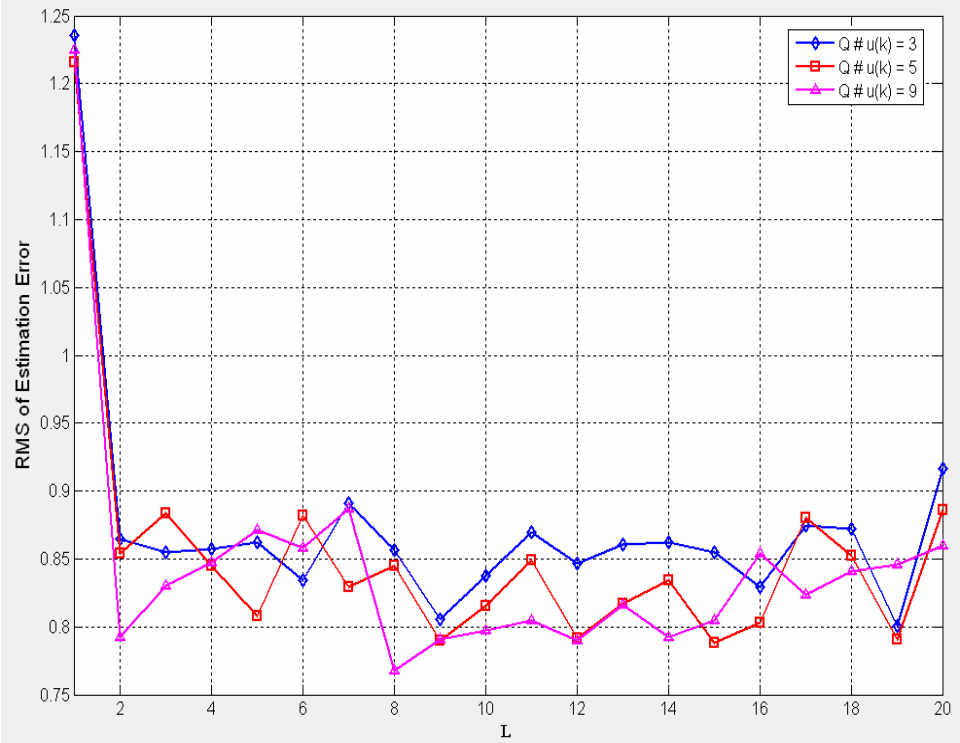

Figure 17 RMS estimation error for the nonlinear model as quantization level of u(k) changing

Table 8 Average values of RMS Estimation error as quantization level of u(k) changing for the nonlinear model

Quantization level of u(k)	Average values of RMS Estimation Error
3	0.8743
5	0.8531
9	0.8449

Comment:

It can be observed from the figures that increasing the number of the quantization levels of the input vector *u(k)* slightly improves the state estimation error (Figure 16 and Figure 17) both linear and nonlinear models. In this simulations, input vector *u(k)* behaves like the disturbance vector *w(k)*. Therefore, the performance of *u(k)*'s state estimation error has similarities with the state estimation error performance of *w(k)*.

3.5. Effect of the Initial State Variance

In this section, effects of the initial state variance are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the initial state variance. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 5, w(k) = 3, u(k) = 3 , I(k) = 3,

variances : w(k) = 1, u(k) = 1, v(k) = 1, I(k) = 0.01,

expected values : x(0) = 0, w(k) = 0, v(k) = 0, I(k) = 0.2 .

For the linear model, Figure 18 is obtained.

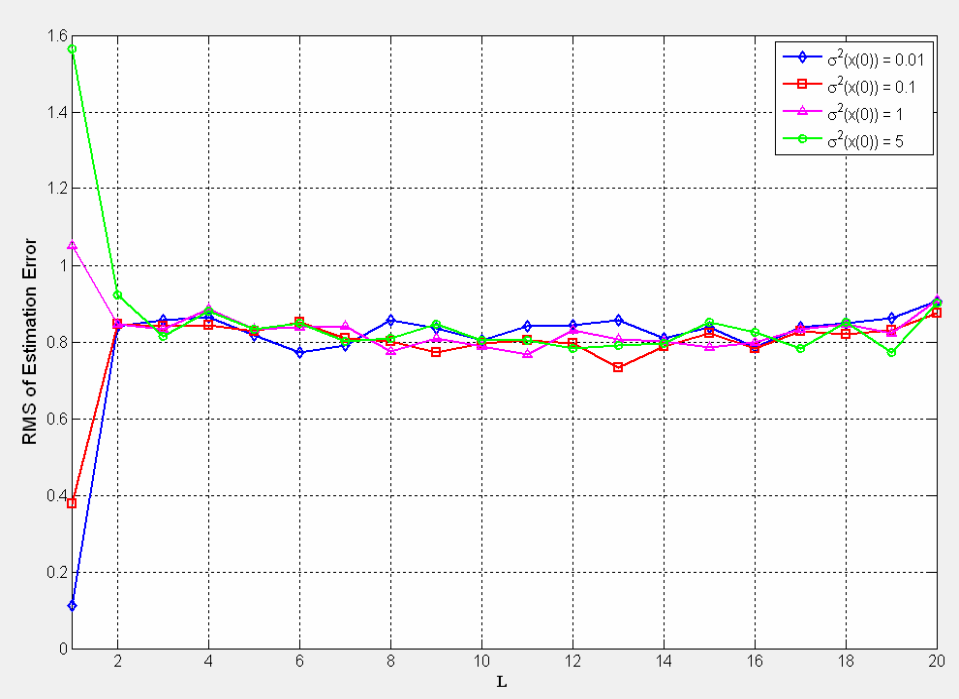

Figure 18 RMS estimation error for the linear model as the initial state variance changing

Table 9 Average values of RMS Estimation error as variance values of x(0) changing for the linear model

Variance values of x(0)	Average values of RMS Estimation Error
0.01	0.79866
0.1	0.79264
1	0.83515
5	0.86382

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k)+ I(k) + v(k)$.

Parameters are same used in linear models. In the simulations for the nonlinear model, Figure 19 is obtained.

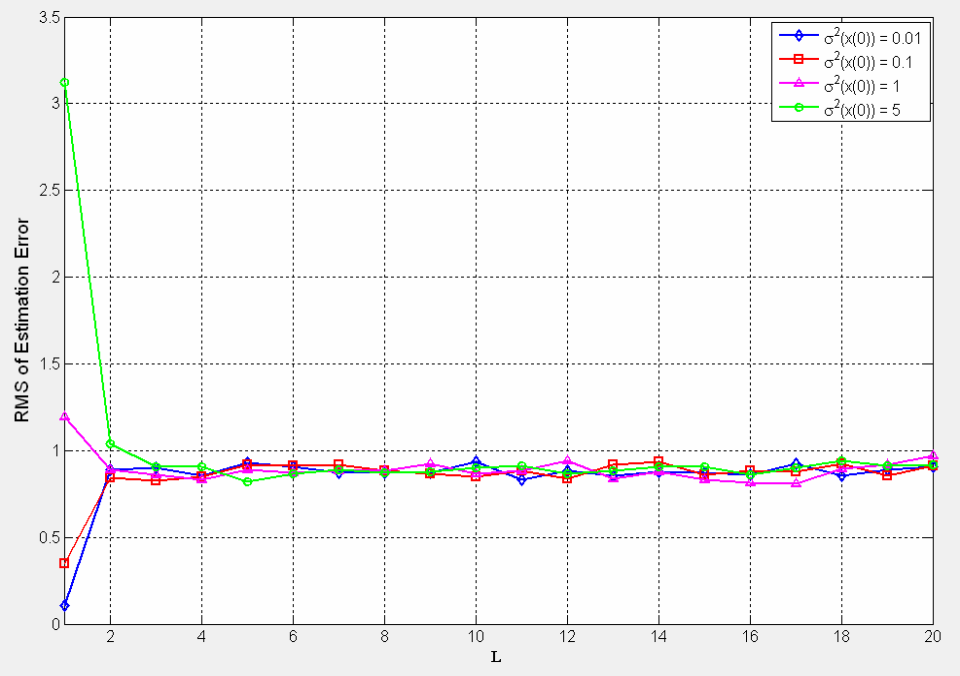

Figure 19 RMS estimation error for the nonlinear model as the initial state variance changing

Table 10 Average values of RMS Estimation error as variance values of x(0) changing for the nonlinear model

Variance values of x(0)	**Average values of RMS Estimation Error**
0.01	0.8444
0.1	0.8549
1	0.8939
5	1.0103

Comment:

When Figure 18 and Figure 19 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 value due to the large initial state variance values, and then decreases significantly to around a value as time k.

3.6. Effect of the Disturbance Noise Variance

In this section, effects of the disturbance noise variance are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the disturbance noise variance. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, w(k) = 3, u(k) = 3, I(k) = 3 ,

variances : x(0) = 1, u(k) = 1, v(k) = 1, I(k) = 0.01 ,

expected values : x(0) = 0, w(k) = 0, u(k) =0, v(k) = 0, I(k) = 0.2

For the linear model, Figure 20 is obtained.

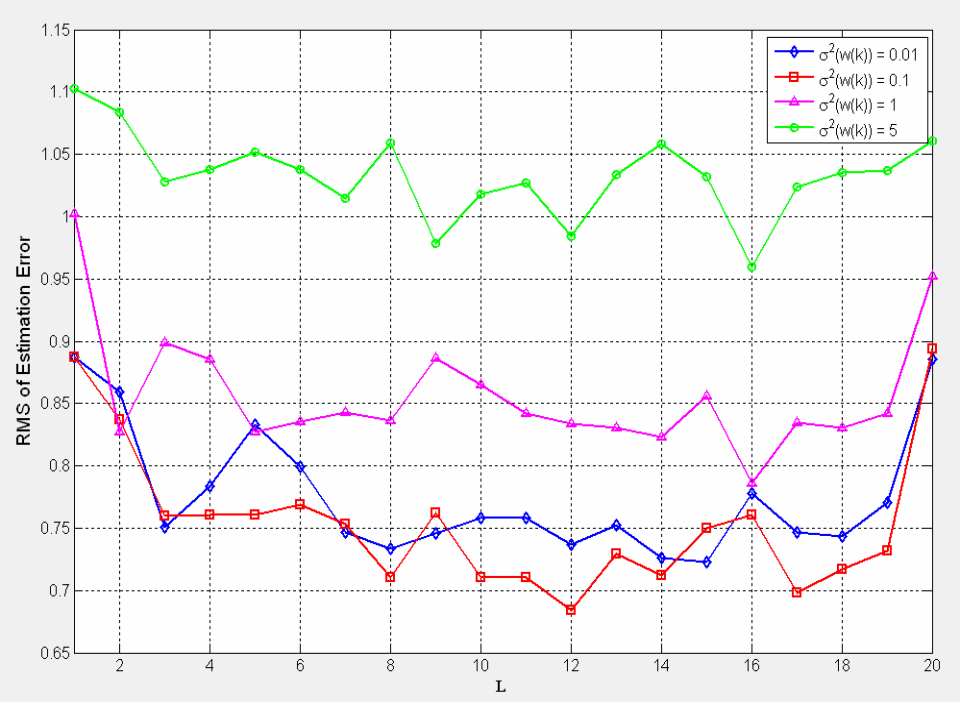

Figure 20 RMS estimation error for the linear model as disturbance noise variance changing

Table 11 Average values of RMS Estimation error as variance values of w(k) changing for the linear model

Variance values of w(k)	Average values of RMS Estimation Error
0.01	0.7758
0.1	0.7548
1	0.8569
5	1.0331

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are same used in linear models. In the simulations for the nonlinear model, Figure 21 is obtained

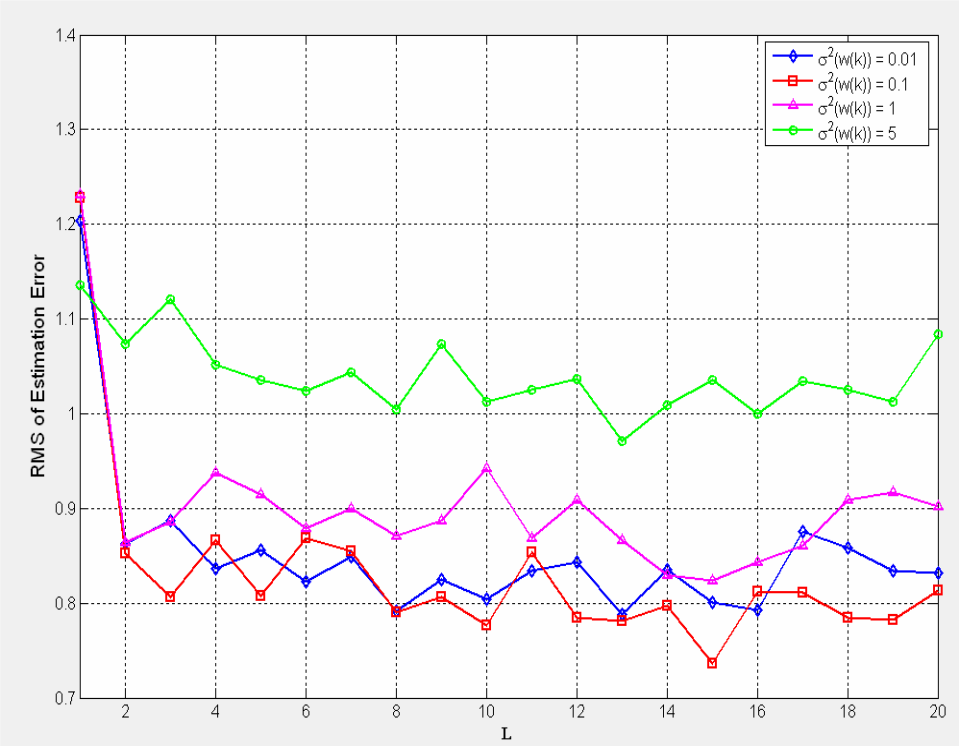

Figure 21 RMS estimation error for the nonlinear model as disturbance noise variance changing

Table 12 Average values of RMS Estimation error as variance values of w(k) changing for the nonlinear model

Variance values of w(k)	Average values of RMS Estimation Error
0.01	0.8517
0.1	0.8306
1	0.9021
5	1.0404

Comment:

Unlike effects of the initial state variances, little change on the disturbance noise variance directly affects performances of the state estimation error both linear and nonlinear models. This effect can be realized during the number of samples, L. As shown by Figure 20 and Figure 21, increase on the disturbance noise variance causes the performance of the estimation to worsen.

3.7. Effect of the Input u(k) Variance

In this section, effects of the input vector variance are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the input vector variance. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, w(k) = 3, u(k) = 3, I(k) = 3 ,

variances : x(0) = 1, w(k) = 1 , v(k) = 1, I(k) = 0.01,

expected values : x(0) = 0, w(k) = 0, v(k) = 0, I(k) = 0.2 .

For the linear model, Figure 22 is obtained.

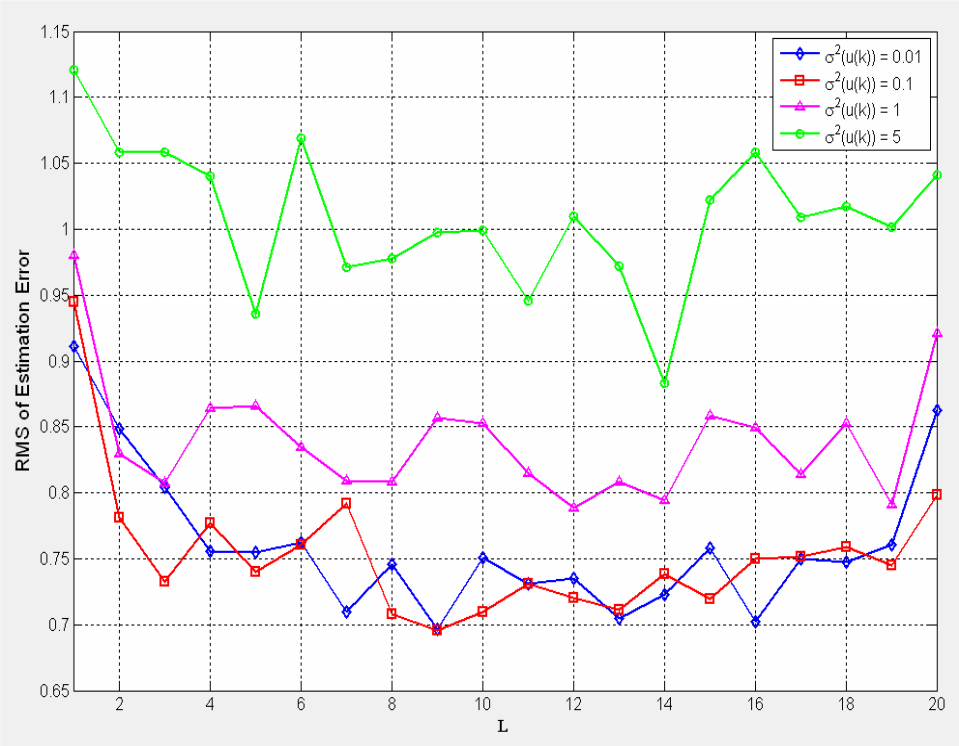

Figure 22 RMS estimation error for the linear model as variance of u(k) changing

Table 13 Average values of RMS Estimation error as variance values of u(k) changing for the linear model

Variance values of u(k)	Average values of RMS Estimation Error
0.01	0.7607
0.1	0.7533
1	0.8400
5	1.0094

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 23 is obtained.

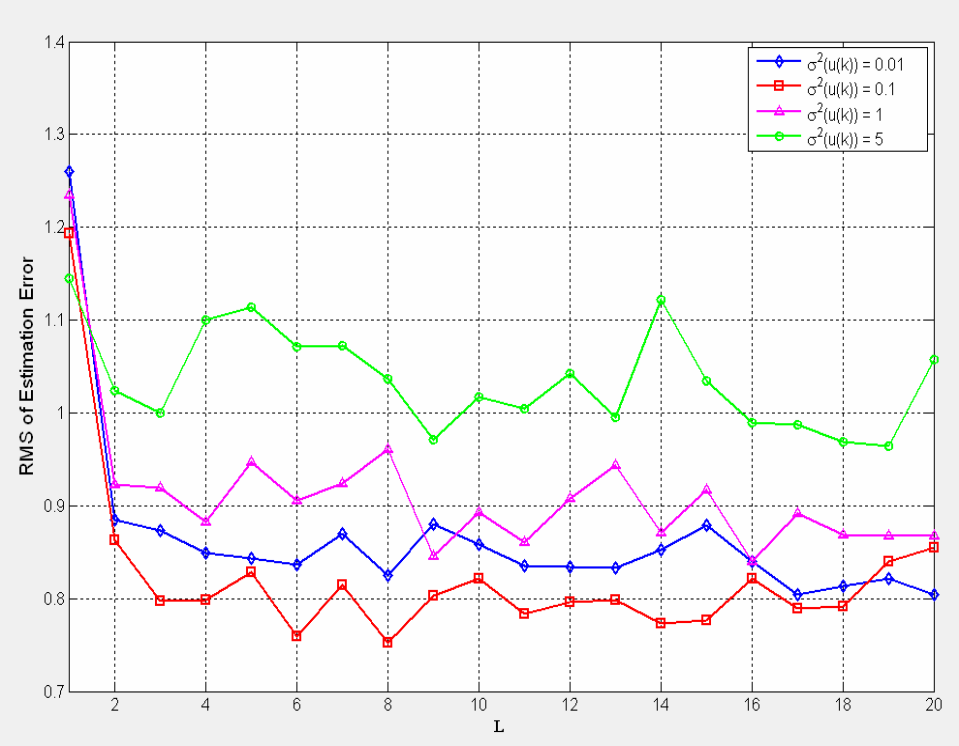

Figure 23 RMS estimation error for the nonlinear model as variance of u(k) changing

Table 14 Average values of RMS Estimation error as variance values of u(k) changing for the nonlinear model

Variance values of u(k)	Average values of RMS Estimation Error
0.01	0.8650
0.1	0.8227
1	0.9135
5	1.0358

Comment:

Decreasing the input *u(k)* variance reduces the state estimation error as expected for both the linear and the nonlinear case. Because the usage of the parameter input *u(k)* in the simulations is similar to the disturbance vector *w(k)*, a little increase on the input u(k) variance causes the state estimation error to get a large value during the sampling time. Figure 22 and Figure 23 show these effects.

3.8. Effect of the Observation Noise Variance

In this section, effects of the observation noise variance are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the observation noise variance. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, w(k) = 3, u(k) = 3, I(k) = 3 ,

variances : x(0) = 1, w(k) = 1, u(k) = 1, I(k) = 0.01,

expected values : x(0) = 0, w(k) = 0,u(k) = 0,v(k) = 0,I(k) = 0.2

.

For the linear model, Figure 24 is obtained.

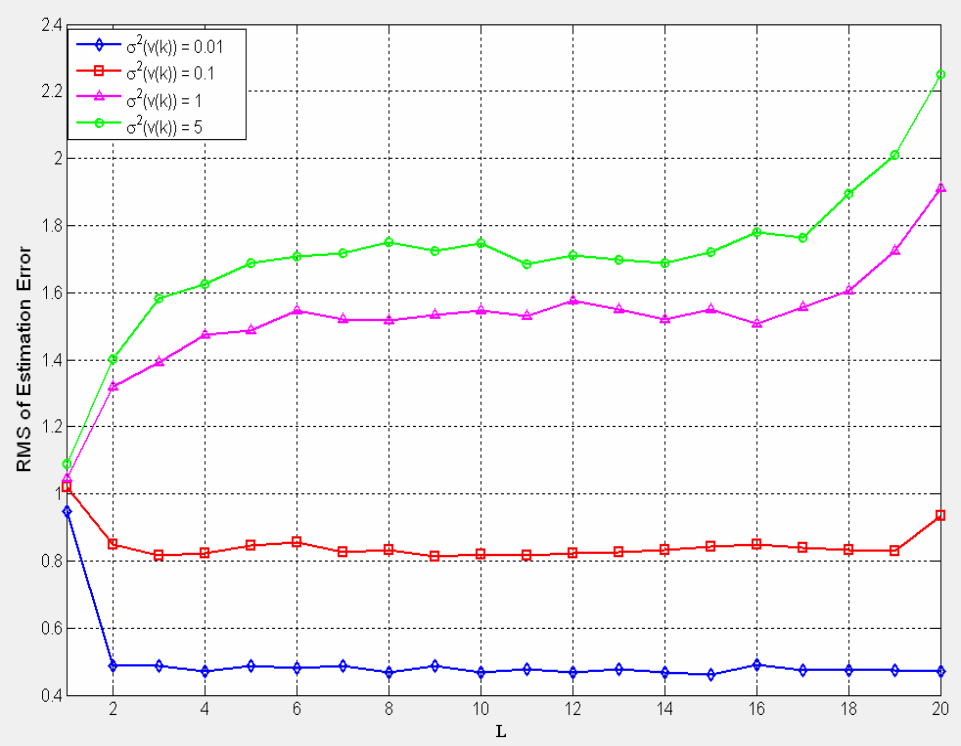

Figure 24 RMS estimation error for the linear model as variance of v(k) changing

Table 15 Average values of RMS Estimation error as variance values of v(k) changing for the linear model

Variance values of v(k)	Average values of RMS Estimation Error
0.1	0.5005
1	0.8461
10	1.5209
15	1.7118

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 25 is obtained.

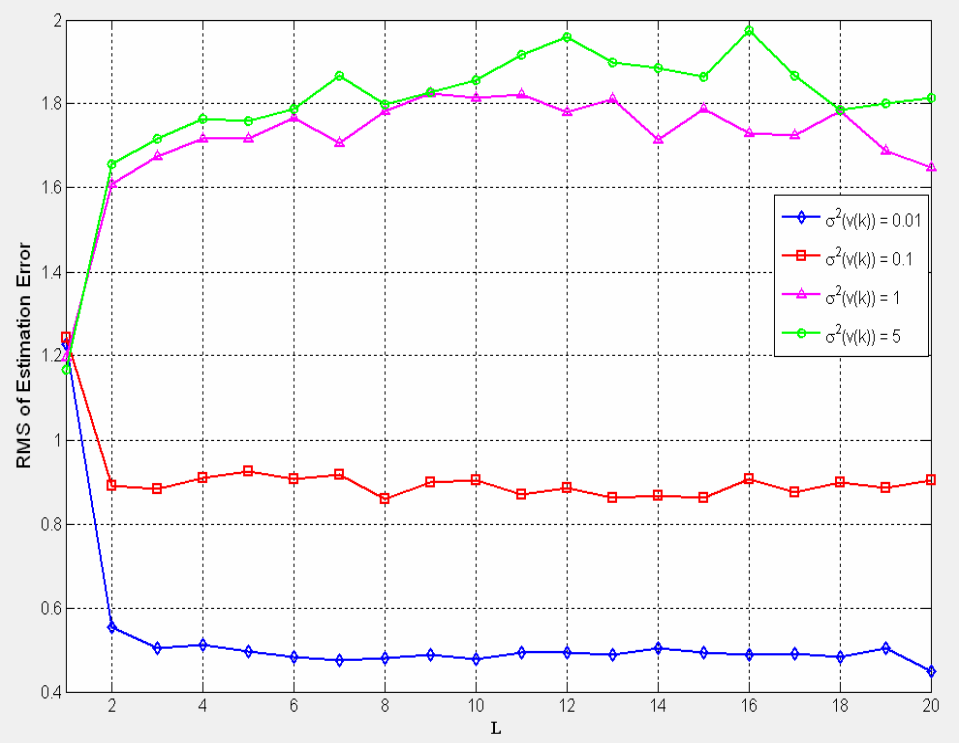

Figure 25 RMS estimation error for the nonlinear model as variance of v(k) changing

Table 16 Average values of RMS Estimation error as variance values of v(k) changing for the nonlinear model

Variance values of v(k)	Average values of RMS Estimation Error
0.1	0.5291
1	0.9073
10	1.7152
15	1.7983

Comment:

Decreasing the observation noise reduces the state estimation error as expected for both the linear and the nonlinear case as observed at Figure 24 and Figure 25.

3.9. Effect of the Quantization Number of the Interference Noise

In this section, effects of the quantization number of the interference vector are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the gate size. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, $L = 20$,

gate size = 0.1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, w(k) = 3, u(k) = 3 ,

variances : x(0) = 1,w(k) = 1,u(k) = 0,v(k) = 1,I(k) =0.01,

expected values : x(0) = 0, w(k) = 0,u(k) = 1,v(k) = 0, I(k) = 0.2

For the linear model, Figure 26 is obtained.

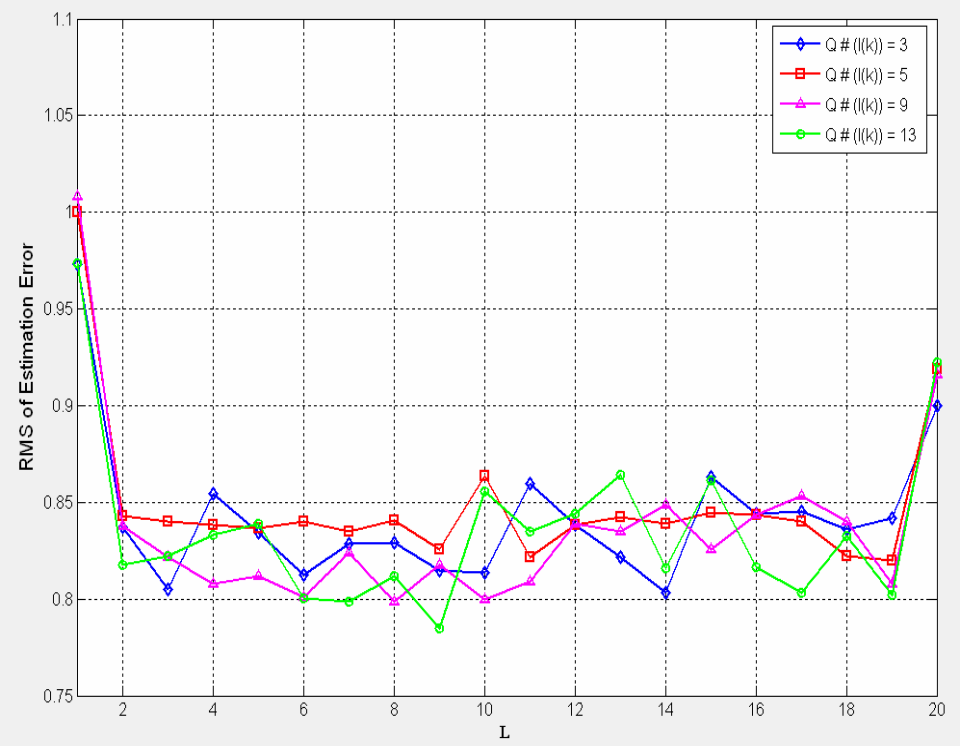

Figure 26 RMS estimation error for the linear model as quantization level of interference changing

Table 17 Average values of RMS Estimation error as quantization level of I(k) changing for the linear model

Quantization Level of I(k)	Average values of RMS Estimation Error
3	0.8428
5	0.8496
9	0.8372
13	0.8366

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 27 is obtained.

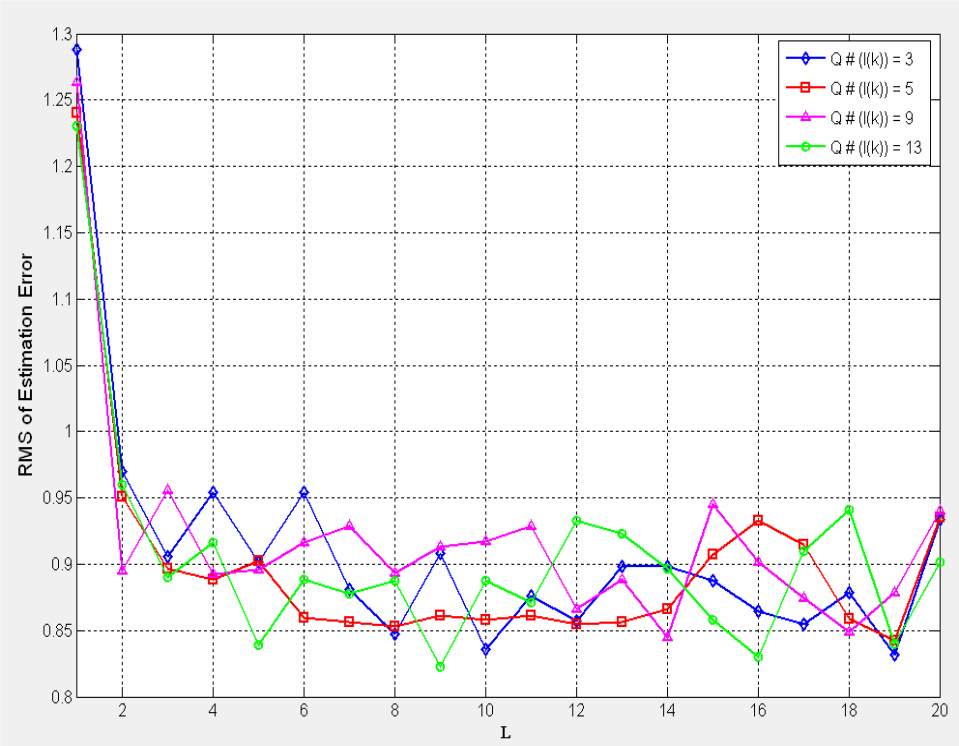

Figure 27 RMS estimation error for the nonlinear model as quantization level of interference changing

Table 18 Average values of RMS Estimation error as quantization level of I(k) changing for the nonlinear model

Quantization Level of I(k)	Average values of RMS Estimation Error
3	0.9113
5	0.8998
9	0.9194
13	0.9050

Comment:

It can be observed from Figure 26 and Figure 27 that the number of the quantization levels of the interference vector *I(k)* slightly affects the performance of the algorithm.

3.10. Effect of the Interference Noise Variance

In this section, effects of the interference noise variance are explained. There are two models, linear and nonlinear models. Figures show RMS error variation depending on the interference noise variance. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, $L = 20$,

gate size = 1 ,

number of max states = 100 ,

quantization numbers : x(0) = 3, w(k) = 3, u(k) = 3 , I(k) = 3 ,

variances : x(0) = 1, w(k) = 1, u(k) = 1, v(k) = 1,

expected values : x(0) = 0, w(k) = 0, u(k) =1, v(k) = 0, I(k) = 0.2

For the linear model, Figure 28 is obtained.

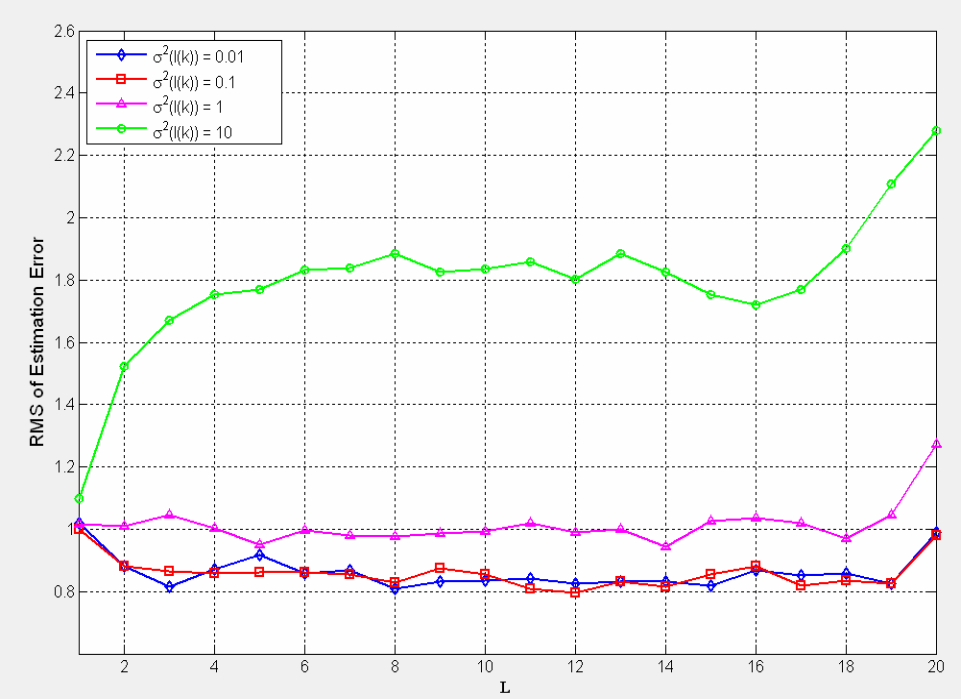

Figure 28 RMS estimation error for the linear model as variance of interference changing

Table 19 Average values of RMS Estimation error as variance values of I(k) changing for the linear model

Variance values of I(k)	Average values of RMS Estimation Error
0.01	0.8625
0.1	0.8591
1	1.0142
10	0.7965

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k) + I(k) + v(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 29 is obtained.

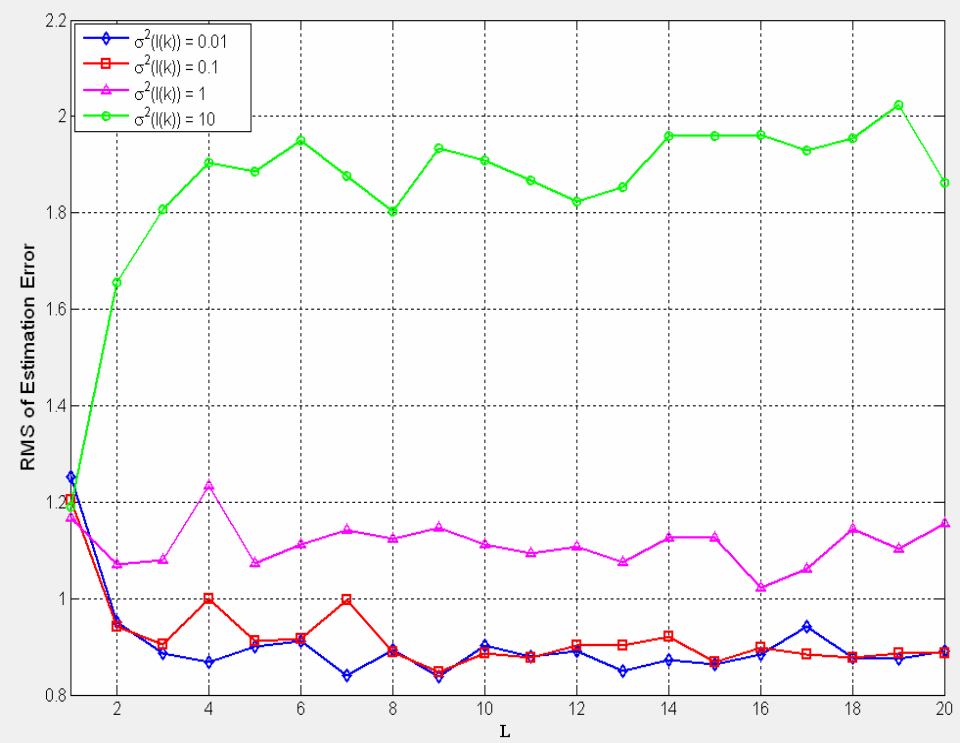

Figure 29 RMS estimation error for the nonlinear model as variance of interference changing

Table 20 Average values of RMS Estimation error as variance values of I(k) changing for the nonlinear model

Variance values of I(k)	Average values of RMS Estimation Error
0.01	0.9034
0.1	0.9198
1	1.1138
10	1.8557

Comment:

Analyzing Figure 28 and Figure 29, it can be seen that the interference noise variance directly affects the performance of the state estimation error both in linear and nonlinear models.

3.11. Effect of the Limiting the Maximum State Number

In this section, effects of the limiting the maximum state number are explained. There are two models, linear and nonlinear models.

Figures show RMS error variation depending on the limitation of the state number. This error, $z(k)- \hat{x}_q(k)$, is calculated for each k time at 1000 simulation, then RMS of errors are obtained.

Linear models used in this simulation are

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

Observation model : $z(k) = x(k) + v(k) + I(k)$.

Parameters are;

number of samples, L = 20 ,

gate size = 0.1 ,

quantization numbers : x(0) = 3, w(k) = 3, u(k) = 3 , I(k) = 3 ,

variances : x(0) = 1, w(k) = 1, v(k) = 1, I(k) = 0.01 ,

expected values : x(0) = 0, w(k) = 0, v(k) = 0, I(k) = 0.2 .

For the linear model, Figure 30 is obtained.

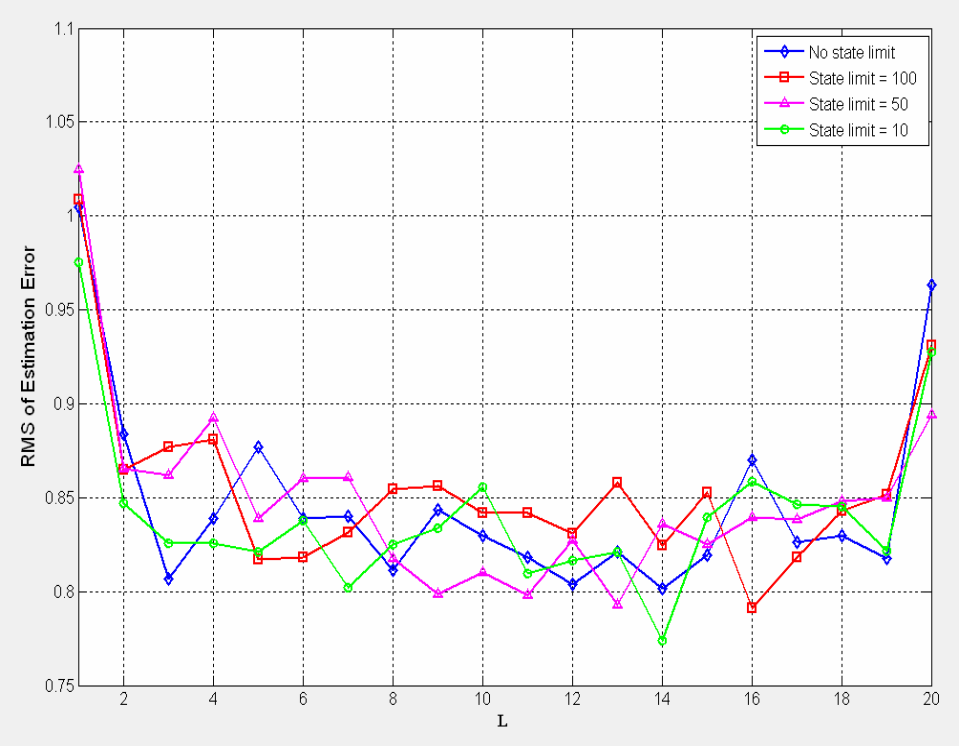

Figure 30 RMS estimation error for the linear model when maximum state number is not limited

Table 21 Average values of RMS Estimation error as the number of state limit values changing for the linear model

# of the state limit	Average values of RMS Estimation Error
No limit	0.8473
100	0.8547
50	0.8490
10	0.8404

Nonlinear models used in this simulation are

Motion model : $x(k+1) = 1-|x(k)| + w(k) + u(k)$,

Observation model : $z(k) = x(k)+ I(k) + v(k)$.

Parameters are same used in linear models. For the nonlinear model, Figure 31 is obtained.

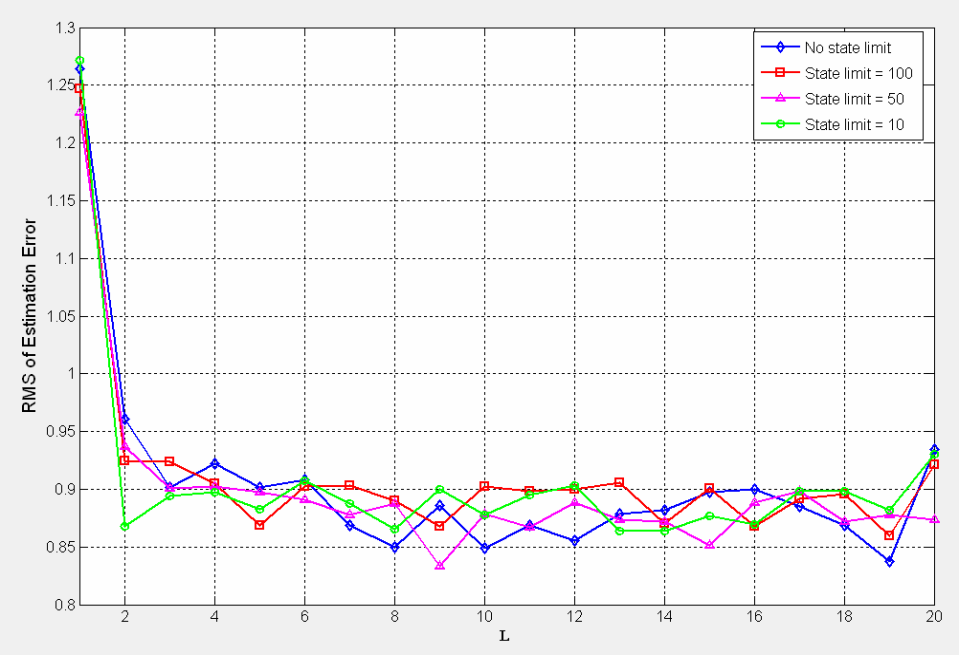

Figure 31 RMS estimation error for the nonlinear model when maximum state number is limited

Table 22 Average values of RMS Estimation error as the number of state limit values changing for the nonlinear model

# of the state limit	**Average values of RMS Estimation Error**
No limit	0.9059
100	0.91233
50	0.89977
10	0.90649

Comment:

It can be observed from the figures that limiting the number of states does not affect the algorithm performance in linear (Figure 30) and nonlinear (Figure 31) 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.

CHAPTER 4

CHAOTIC SYSTEMS

The possibility of using chaotic signals to carry information was first proposed in 1993 and, since then, chaotic communication have been a very important topic in both nonlinear science and engineering. [3]

The noise-like signals generated by deterministic chaotic systems have been successfully used in various engineering areas. These signals are typically broadband and similar to a stochastic process and can therefore be possibly used in secure communication applications, especially spread spectrum communication systems. [3]

In chaotic spread spectrum communication and chaotic modulation applications that have been developed, binary digital symbols are carried by chaotic sequences at the transmitter. A natural way to do is to modulate each digital bit by a chaotic sequence with randomly picked initial condition and different parameters of the system, which greatly spreads the spectrum of the original binary digital signals, and make them quite noise like. At the receiver, a parameter estimator is employed to determine which symbol is represented by the received noise-like and broadband chaotic sequence. Although this scheme allows the communication system to have high security and broadband nature, it has low information carrying efficiency because each sequence can carry only one bit a time. [4]

Throughout this chapter, definitions of chaos and nonlinear dynamics are given. Then common types of chaotic mapping are given detailly.

4.1. Chaos and Nonlinearity

The word "chaos" implies some observation of a system which varies unpredictably. If a measurement does not have regularity or order, it is called "chaotic". However, it is not a set of random events. For example, flipping a coin 100 times is not a chaotic event because chaotic dynamics are deterministic developments with chaotic outcome, i.e., current state of a system depends on the previous state in a rigidly determined way. [5]

Mathematical requirements of a system to be chaotic are given below:

Let V be an interval. f : V $\rightarrow$V is chaotic on V if the following conditions hold [6, 7]

1. f is sensitive to initial conditions

2. f is topologically transitive

3. periodic points are dense in V

Sensitivity to initial conditions means that two points in such a system move in vastly different paths even if the difference between their initial conditions is small [6] . Sensitivity to initial conditions is related to the Lyapunov exponents [6, 7]. The Lyapunov exponent of a map is used to obtain sensitive dependence to initial conditions which is the first necessity for the above characterization of chaotic maps.

If a system is allowed to start from two slightly different initial states, say x and x + a, after n iterations, their divergence may be characterized as [7]

$$\varepsilon(n) \approx \varepsilon e^{\lambda n} \tag{4.1}$$

where, λ is Lyapunov exponent and it gives average rate of divergence. If λ is negative, slightly apart trajectories converge and the evolution is not chaotic. If λ is positive two trajectories diverge and evolution is sensitive to initial conditions. One dimensional map is given by

$$x_{n+1} = f(x_n) \tag{4.2}$$

where, n is iteration number. The difference between two trajectories after n steps, whose initial conditions are close to each other, is given as

$$f^n(x+\varepsilon)-f^n(x)\approx \varepsilon e^{\lambda n} \tag{4.3}$$

or

$$ln\left(\frac{f^n(x+\varepsilon)-f^n(x)}{\varepsilon}\right)\approx \lambda n \tag{4.4}$$

where $f^n(x)$ is iterated value of initial value x after n steps. For small ε, this can be rewritten as

$$\lambda = \lim_{n\to\infty}\frac{1}{n} ln\left|\frac{df^n}{dx}\right| \tag{4.5}$$

Finally, if we use chain rule [7] for the derivative of the nth iteration and take the limit as n tends to infinity we obtain the Lyapunov exponent as

$$\lambda = \lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n-1} ln\, f'(\chi_i) \tag{4.6}$$

Definition 4.1: A mapping f : [0,1]→[0,1] is transitive if for every pair of subintervals I and J of [0,1] there is an n such that $f^n(I)\cap J\neq\varnothing$ [6] where $f^n(I)$ is iteration result of subinterval I after n steps and Ø is empty set.

The last characteristic of chaotic systems, denseness of periodic points on V means that: there are infinitely many points with infinitely different periods. A periodic point is a point which cycles after a number of iterations. However, although theoretically there are infinitely many periodic points, practically it is not possible to find these numbers because of rounding-off errors. Since rounding-off errors will be amplified because of initial sensitivity property, we will never achieve an exact periodicity but quasy-periodicity, which means we can save appearance of periodicity for only limited number of iterations but after that, periodicity becomes unstable because of sensitivity. [8]

4.2. One Dimensional Chaotic Systems

In this section, three popular one-dimensional chaotic systems are studied. These systems are skew tent map, tent map and symmetric tent map.

4.2.1 Skew Tent Map

The skew tent map is defined [9] by

$$f(\chi)=\begin{cases}\dfrac{\chi}{a} & \text{if } 0 \le \chi \le a \\ \dfrac{1-\chi}{1-a} & \text{if } a < \chi \le 1\end{cases} \tag{4.7}$$

and represented in Figure 32.

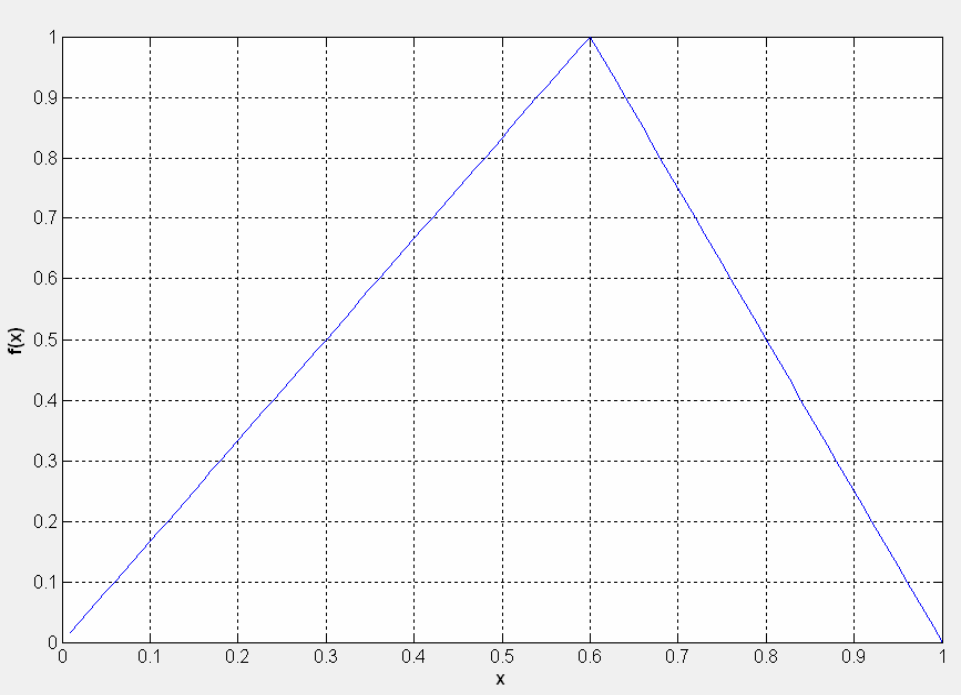

Figure 32 Skew tent map when a=0.6.

It is noninvertible transformation of the unit interval onto itself. It depends on the parameter *a* which satisfies

$$0.5 \le a \le 1 . \qquad (4.8)$$

The result for $0 < a \le 0.5$ are completely analogous. For a = 0.5, f becomes the *tent map*. Typical trajectory of the skew tent map system is given in Figure 33. In the simulation, the parameter a is equal to 0.6 and 1000 samples are given.

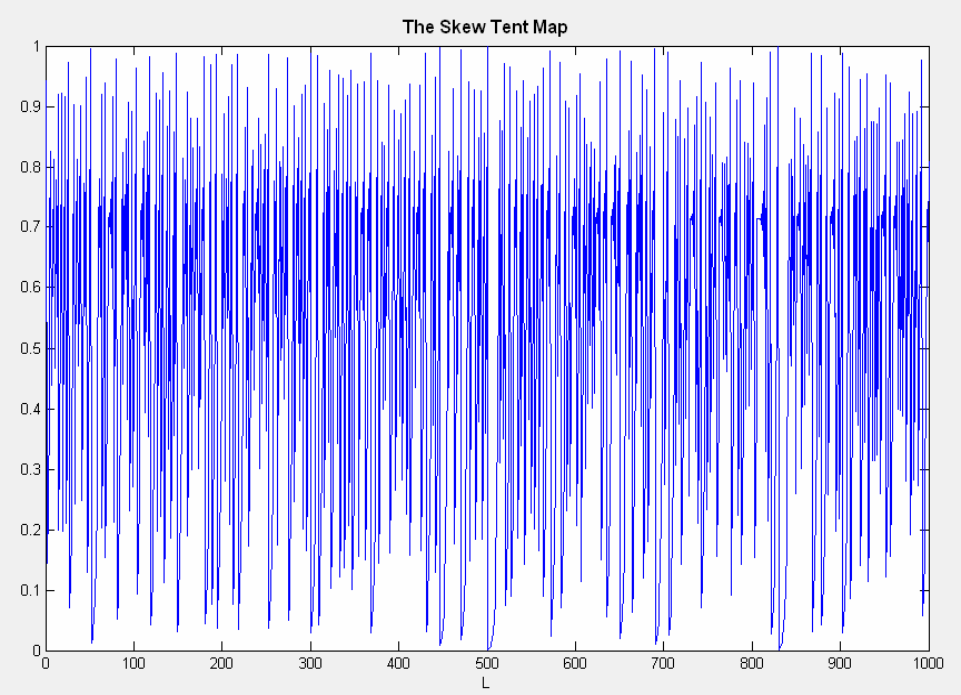

Figure 33 1000 points of typical trajectory of the skew tent map system for a=0.6

4.2.2 Tent Map

The tent map is defined [10] by

$$\chi(k+1) = \begin{cases} a(1-|2\chi(k)-1|) & if\ 0 \le \chi(k) \le 1 \\ 0 & elsewhere \end{cases} \tag{4.9}$$

and represented in Figure 34.

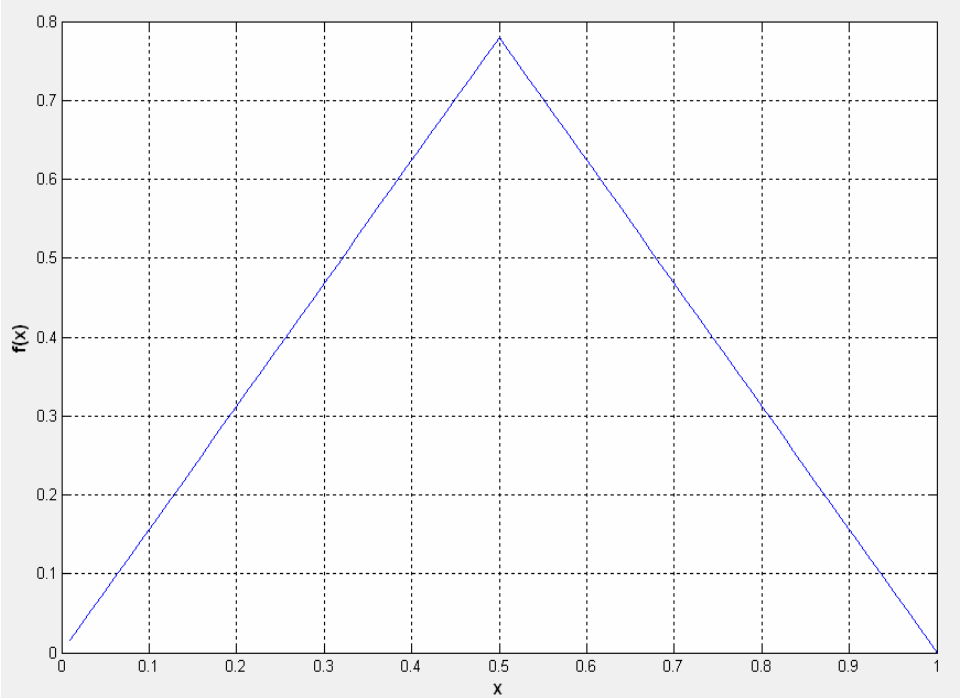

Figure 34 Tent map when a=0.78

When *a* is between 0.5 and 1, trajectory of this map exhibits chaotic behavior. Typical trajectory of the tent map system is given in Figure 35. In the simulation, the parameter a is equal to 0.78 and 1000 samples are given.

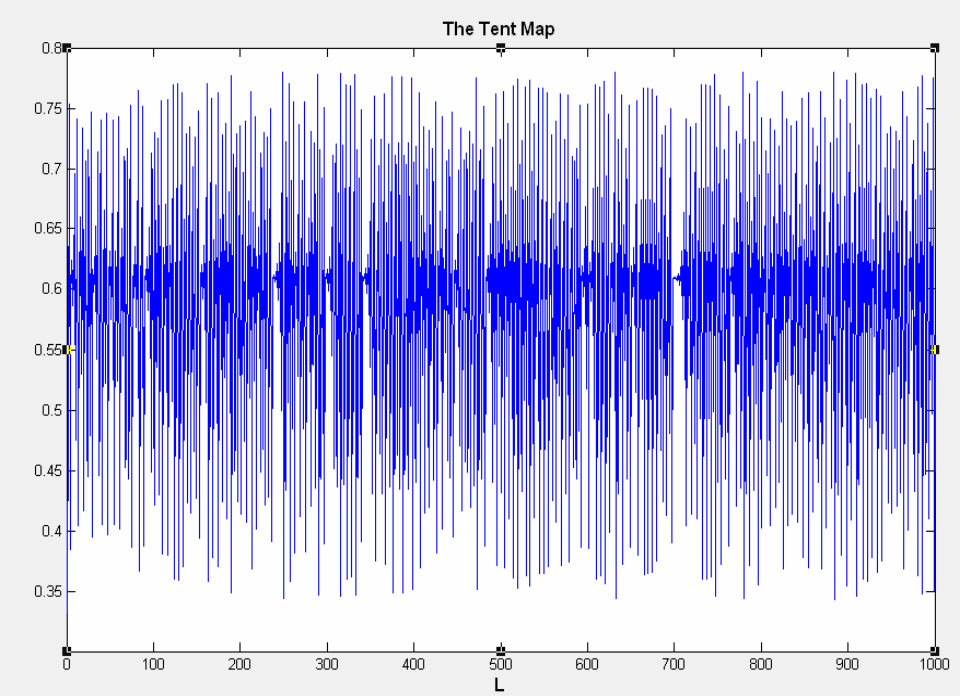

Figure 35 1000 points of typical trajectory of the tent map system for a=0.78

4.2.3 Symmetric Tent Map

The symmetric tent map is defined [10] as

$$\chi(k+1) = a - 1 - a|\chi(k)| \tag{4.10}$$

and represented in Figure 36.

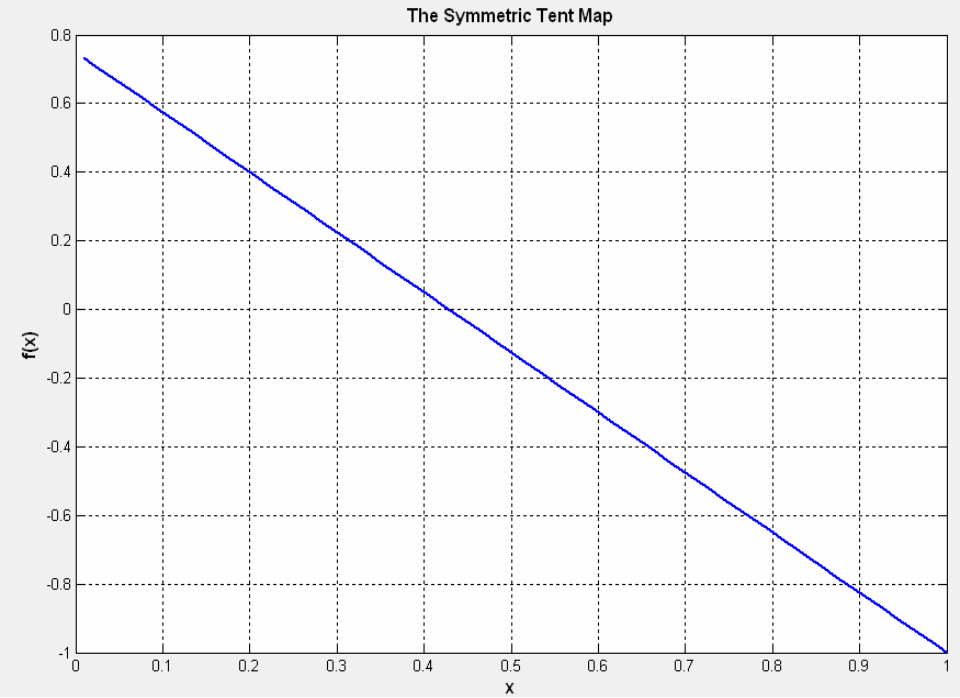

Figure 36 Symmetric tent map when a=1.75.

When a is between 1 and 2, the system has chaotic behavior. Typical trajectory of the symmetric tent map system is given in Figure 37. In the simulation, the parameter a is equal to 1.75 and 1000 samples are given.

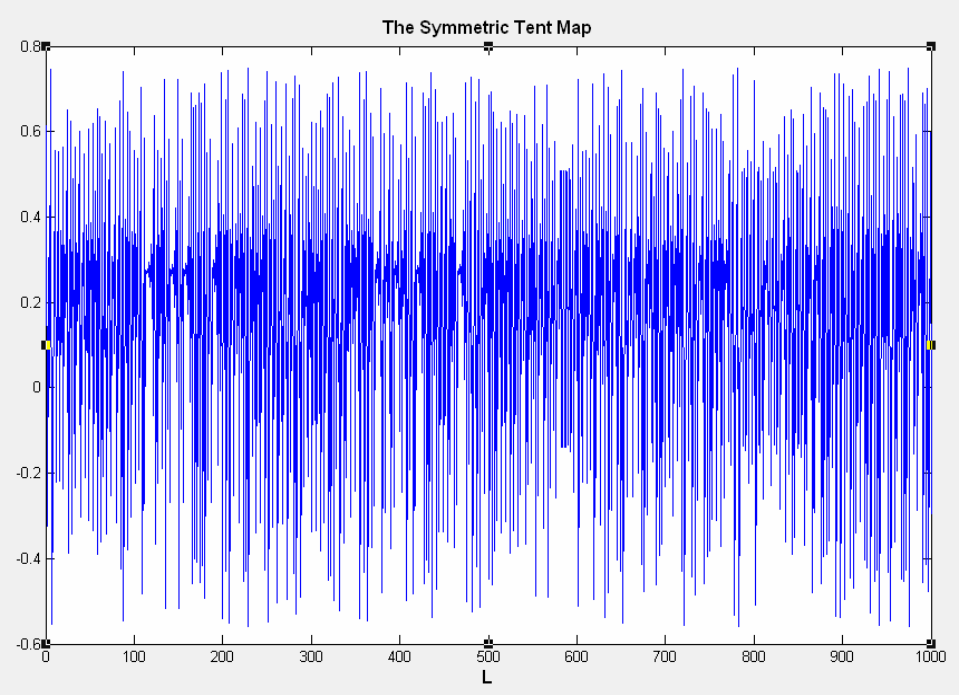

Figure 37 1000 points of typical trajectory of the symmetric tent map system for a=1.75

4.2.4 The Sensitivity to Initial States and Parameter *a*

As explained in section 4.1, chaotic systems are sensitive to initial conditions. Starting from different initial states causes paths to diverge vastly, even though difference between initial states is too small. Therefore, performances of one-dimensional chaotic systems, which are presented in section 4.2, are affected by the difference between initial states.

Effects of the initial state on the skew tent map are analyzed in Figure 38. In the simulation, a reference path is generated, namely x1, taking the initial state 0.87 and the parameter a 0.75. Then second path, x2, is generated with the same parameter a. The initial state of the second path differ by 0.00001 from x1's. Last third path, x3, is generated with the same parameter a and its initial state differs by 0.01 from x1's. In Figure 38, there are two subfigures which shows these paths. In

the first subfigure, x1 and x2 is shown. Despite the small difference between the initial states, two paths diverge with time L. In the second subfigure, x1 and x3 is figured out. Because the difference between initial points becomes larger, i.e., 0.01 , these two paths diverge earlier.

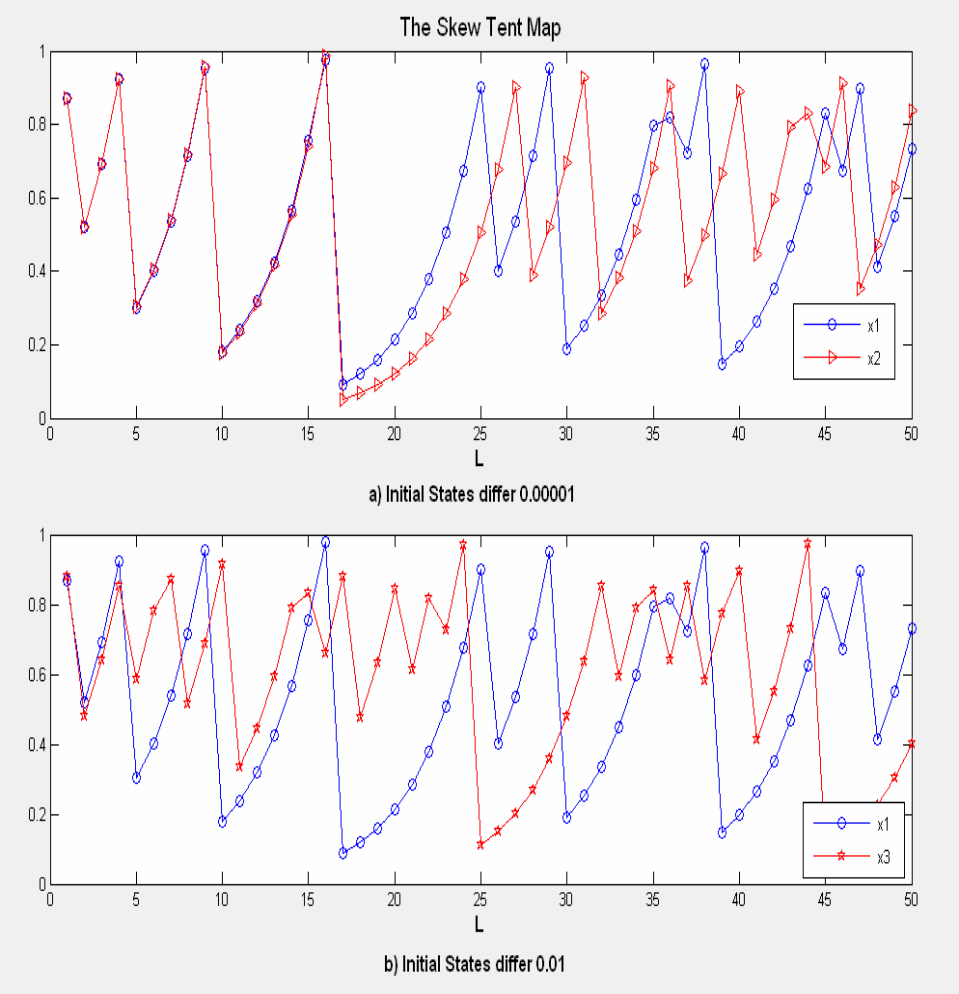

Figure 38 Effect of initial states on skew tent map when a=0.75, a) Initial states differ 0.00001, b) Initial states differ 0.01

The effects of the difference between initial states are similar on the tent map shown by Figure 39. In these simulations, a is taken 0.82 and reference initial state is 0.63.

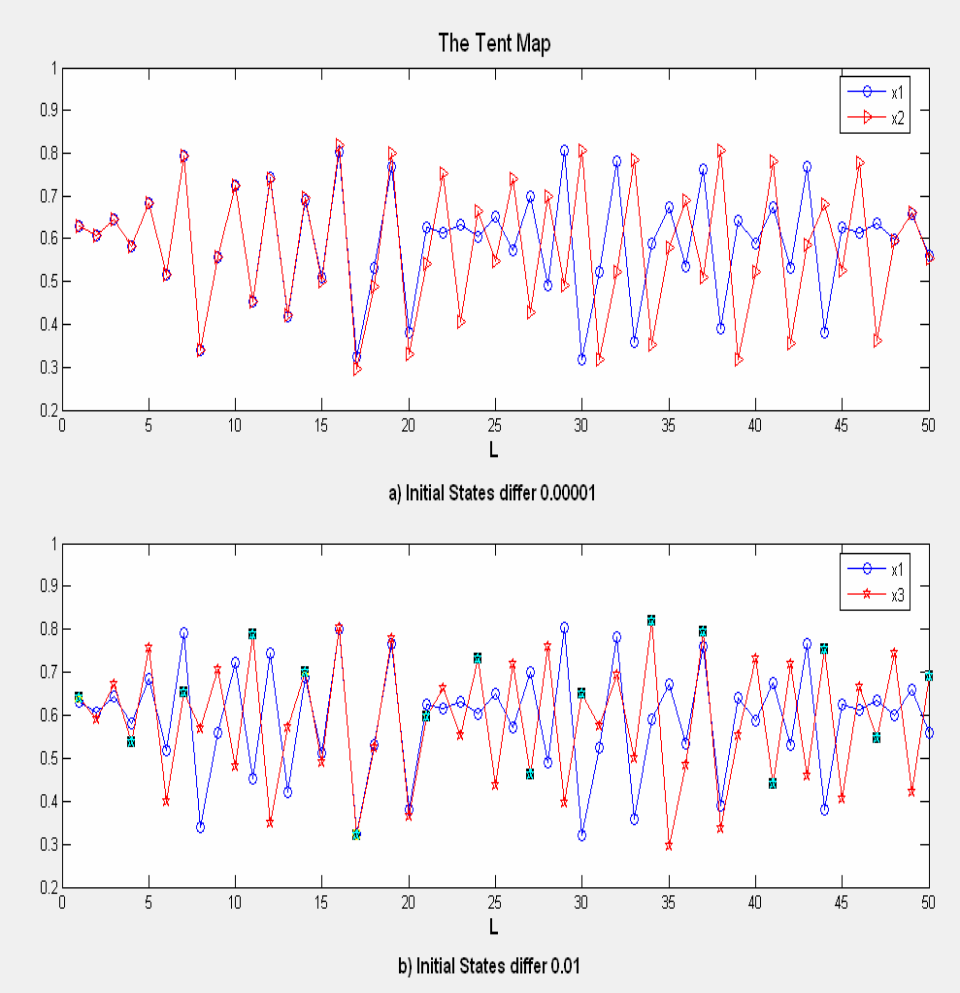

Figure 39 The effects of initial states on the tent map for a=0.82, a) Initial states differ 0.00001, b) Initial states differ 0.01

The effects of the difference between initial states are similar on the symmetric tent map shown by Figure 40. In these simulations, a is taken 1.62 and reference initial state is 0.3.

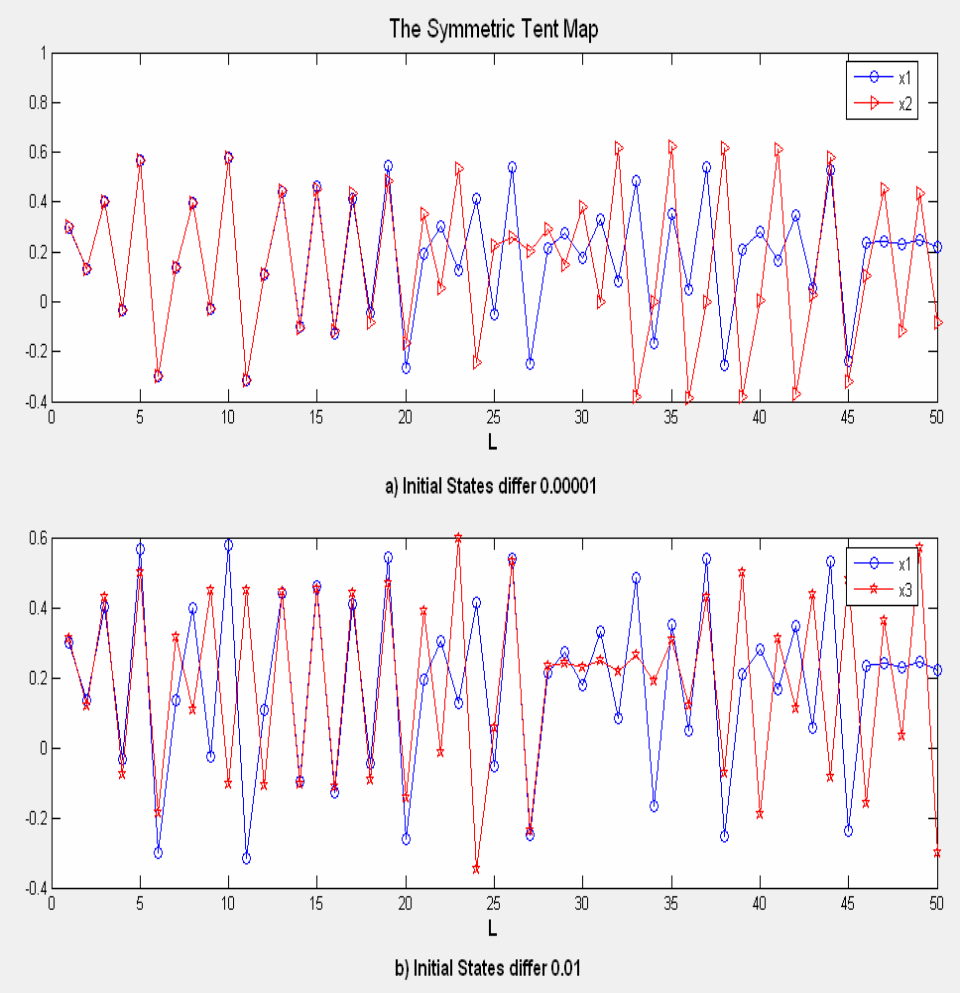

Figure 40 The effects of initial states on the symmetric tent map for a=1. 62 a) Initial states differ 0.00001, b) Initial states differ 0.01

Another important parameter affecting the chaotic systems is the parameter *a*. *a* guides how the system follows the path. A small difference on *a* causes system to follow irrelevant paths. This structure can be analyzed on Figure 41, Figure 42, and Figure 43 for different one-dimensional chaotic systems.

In Figure 41, the initial state of the skew tent map is taken 0.87. The parameter a is taken 0.6 for the reference path. Then for the paths, namely x2 and x3, a is taken 0.61 and 0.8 respectively.

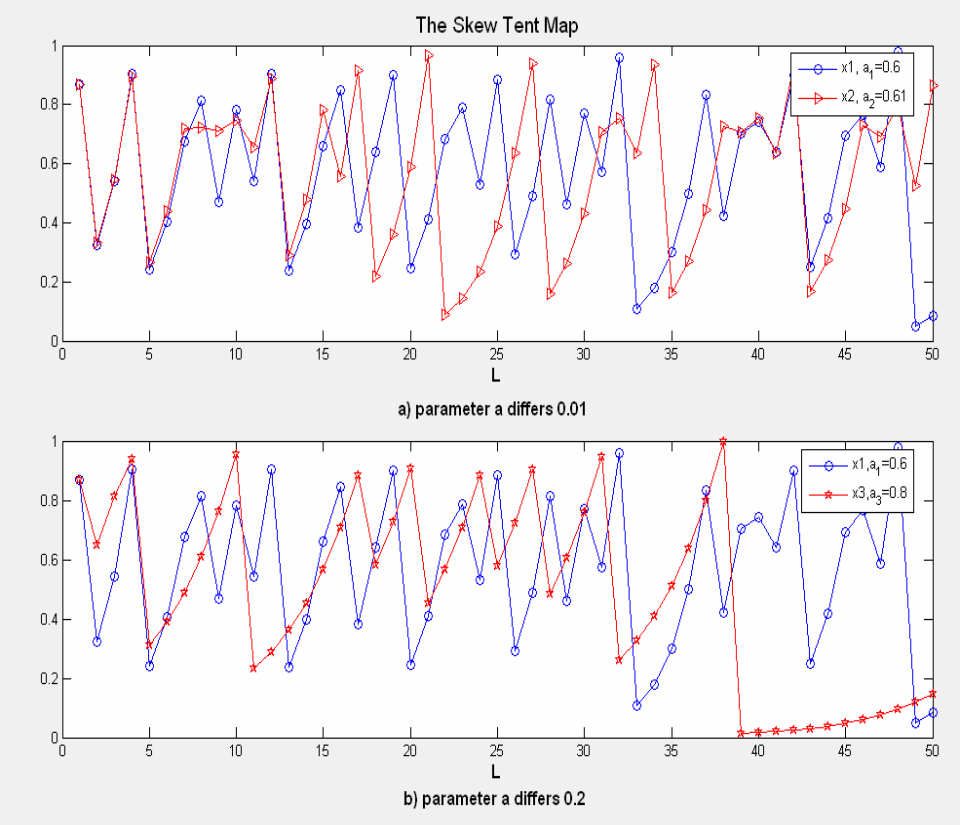

Figure 41 The skew tent map paths with the same initial states and different *a* values, a) a differs 0.01, b) a differs 0.2

In Figure 42, the initial state of the tent map is taken 0.63. The parameter a is taken 0.7 for the reference path. Then for the paths, namely x2 and x3, a is taken 0.71 and 0.8 respectively.

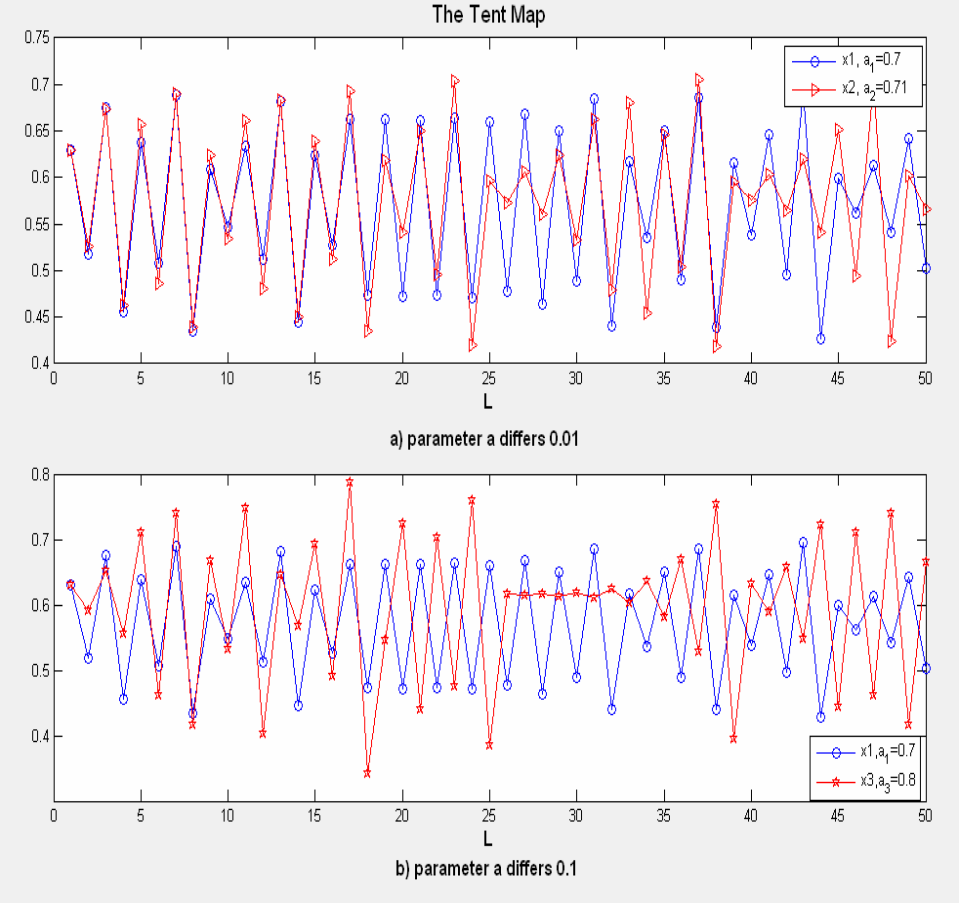

Figure 42 The tent map paths with the same initial states and different *a* values, a) a differs 0.01, b) a differs 0.1

In Figure 43, the initial state of the symmetric tent map is taken 0.3. The parameter a is taken 1.7 for the reference path. Then for the paths, namely x2 and x3, a is taken 1.71 and 1.8 respectively.

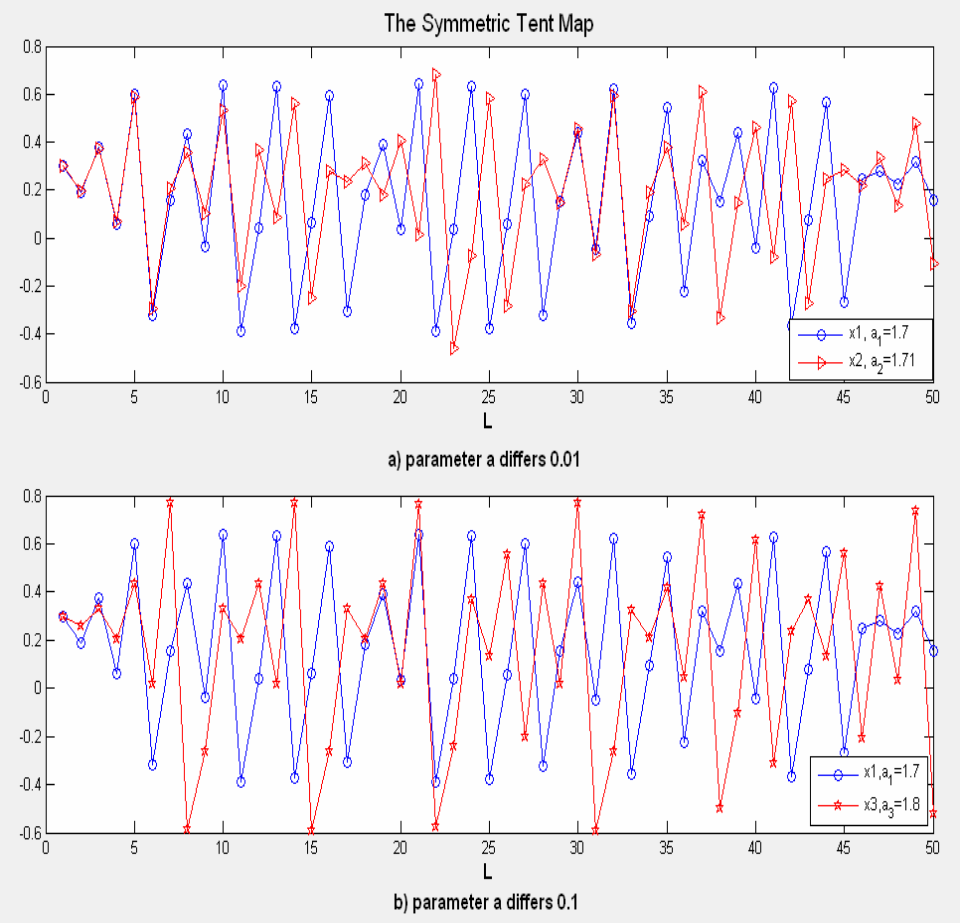

Figure 43 The symmetric tent map paths with the same initial states and different *a* values, a) a differs 0.01, b) a differs 0.1

4.3. The Nonlinear Chaotic Algorithm Map (NCA)

One-dimensional chaotic system with the advantages of high-level efficiency and simplicity has been widely used. However, their weakness, such as small key space and weak security, is disturbing. To overcome these drawbacks, a new chaotic algorithm is designed. The new algorithm has the advantages of high-level security, large key space. [11]

The nonlinear chaotic algorithm map uses power function $(1-x)^{\beta}$ and tangent function instead of linear function. The NCA is defined as

$$\chi_{n+1} = \lambda \cdot tg(\alpha\chi_n) \cdot (1-\chi_n)^{\beta} \tag{4.11}$$

where $x_n \in (0,1)$, n=0,1,2... . [11]

The parameter λ is defined as

$$\lambda = \mu \cdot ctg\left(\frac{\alpha}{1+\beta}\right) \cdot \left(1+\frac{1}{\beta}\right)^{\beta}, \quad \mu > 0. \tag{4.12}$$

Finally, parameter μ is obtained by experimental analysis; as a result $\mu=1-\beta^{-4}$. Therefore, the NCA map is defined as follows:

$$\chi_{n+1} = (1-\beta^{-4}) \cdot ctg\left(\frac{\alpha}{1+\beta}\right) \cdot \left(1+\frac{1}{\beta}\right)^{\beta} \cdot tg(\alpha\chi_n) \cdot (1-\chi_n)^{\beta} \tag{4.13}$$

where $x_n \in (0,1)$, $\alpha \in (0,1.4]$, $\beta \in [5,43]$ or $x_n \in (0,1)$, $\alpha \in (1.4,1.5]$, $\beta \in [9,38]$ or $x_n \in (0,1)$, $\alpha \in (1.5,1.57]$, $\beta \in [3,15]$. The ranges of α and β are obtained by iteration experimental analysis. Figure 44 shows the iteration of the NCA map when x_0=0.3, α=1.57, β=3.5. From these statistical data, it is seen that the new chaotic algorithm spreads the initial region over the entire phase space.

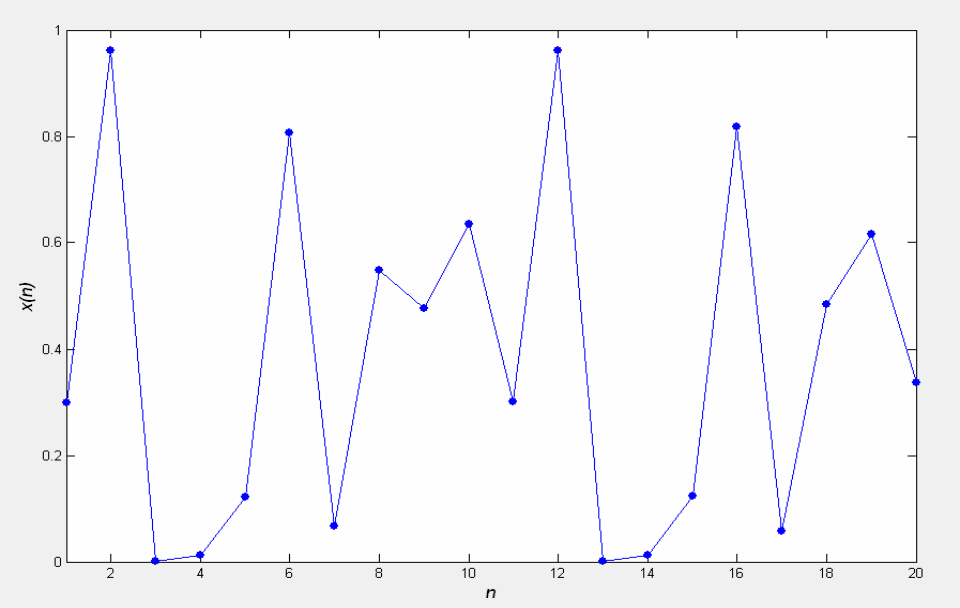

Figure 44 Iteration property of the NCA map

Like other chaotic schemes, one-dimensional chaotic schemes, the NCA is sensitive to initial states and parameters α, β. A small change of these parameters will generate a completely different result. In the simulations, that shows these effects, initial state is 0.3, α is 1.57 and β is 3.5. Figure 45 shows the effect of the difference between initial states, which results different paths. Small changes on parameters α and β cause similar effects as figured out at Figure 46 and Figure 47.

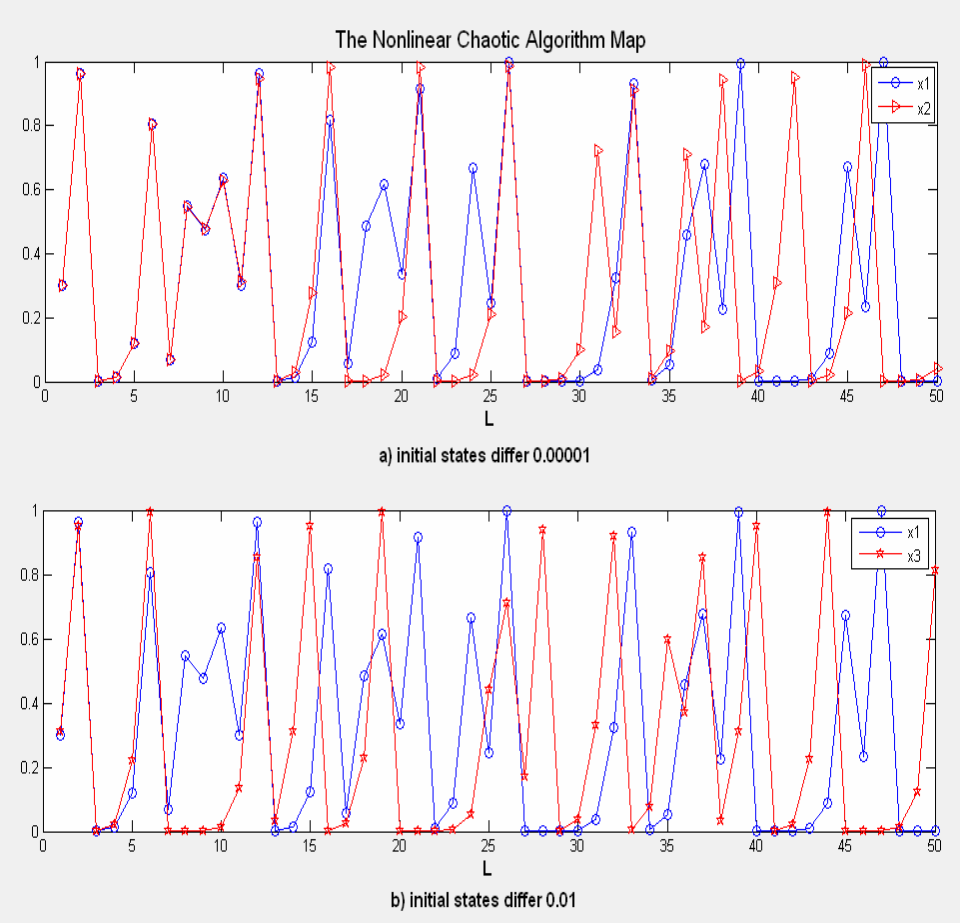

Figure 45 Effect of initial states on the NCA map, a) Initial states differ 0.00001, b) Initial states differ 0.01

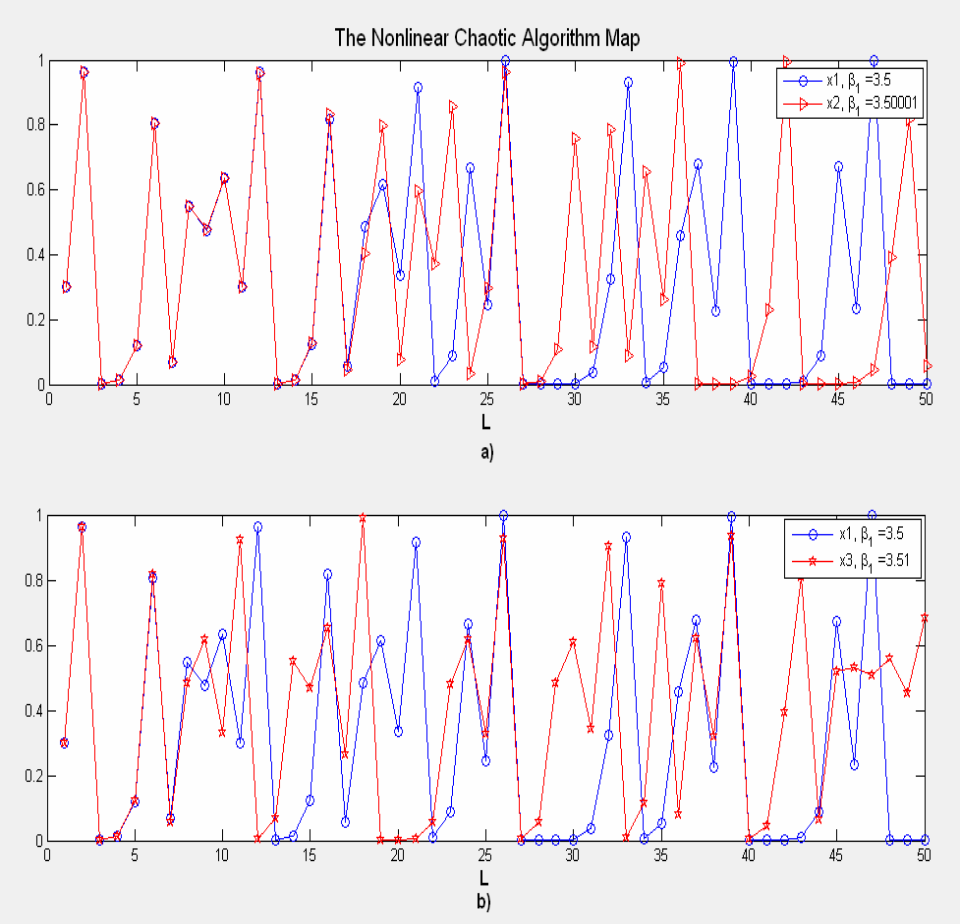

Figure 46 Effect of β on the NCA map, a) β differs 0.00001, b) β differs 0.01

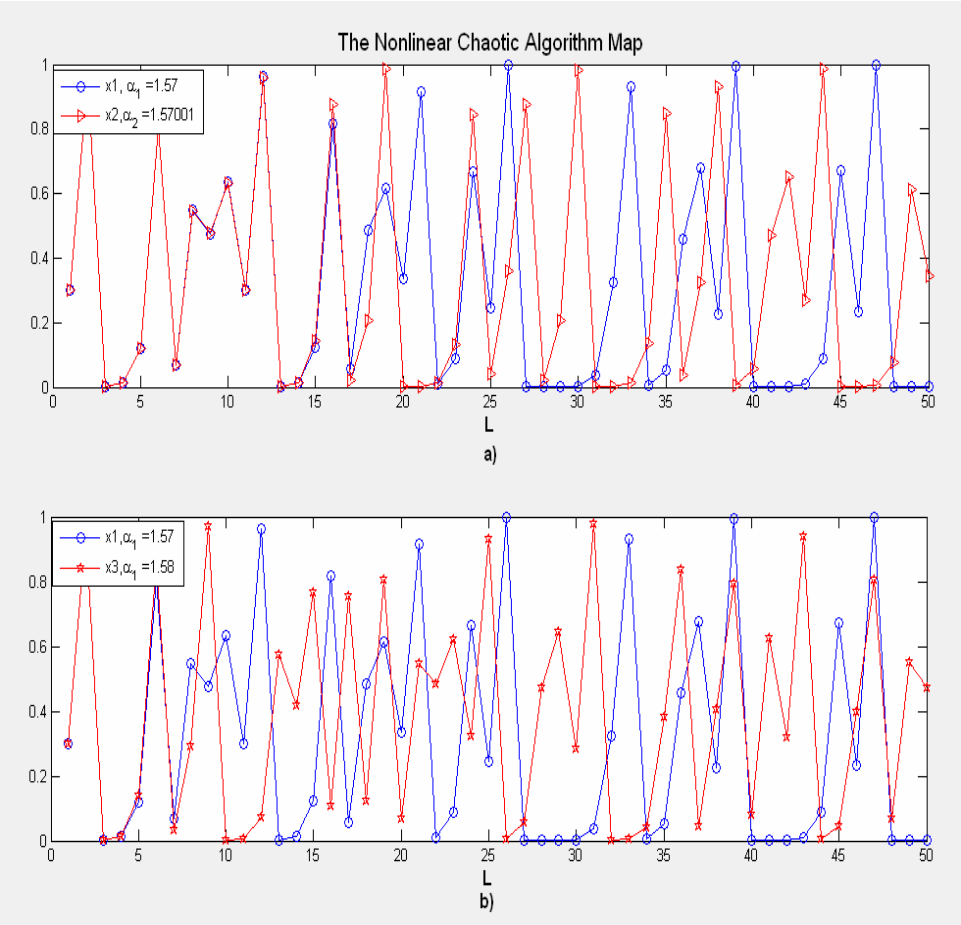

Figure 47 Effect of α on the NCA map, a) α differs 0.00001, b) α differs 0.01

CHAPTER 5

APPLICATION OF ODSA TO CHAOTIC SYSTEMS

As described in Chapter 4, chaotic signals can be used for secure communication applications. Communication systems with chaotic modulation and demodulation can be figured out by Figure 48. Binary digital signals are prepared by chaotic systems with different parameters on the transmitter side. If digital 0 is send to the transmitter, a signal source is switched to SYS1 to generate a chaotic sequence with parameter a_1. However, if digital 1 is send to the transmitter, the signal source is switched to SYS2 which generates a chaotic sequence with parameter a_2 . X1 and X2 are two finite-length chaotic sequences generated by the same type of system with two different parameters, a_1 and a_2. These signals are transmitted through communication channel and affected by Gaussian channel noise.

At the receiver side, there are two optimum decoding based smoothing algorithm (ODSA) decoders to extract chaotically transmitted digital signals. These decoders have the knowledge of the parameters a_1 and a_2. A comparator, which is at the end of ODSA decoders, compares the outputs of ODSA parts and decides whether the transmitted signal is 0 or 1.

To decide the transmitted binary digital signal, the comparator compares average estimation errors from the two ODSA decoders. Mean Square Errors (MSE's) for ODSA is defined as

$$MSE1 = \frac{1}{L}\sum_{n=1}^{L}\left(y_n - \hat{x}_{1n}\right)^2 \tag{5.1}$$

$$MSE2 = \frac{1}{L}\sum_{n=1}^{L}\left(y_n - \hat{\chi}_{2n}\right)^2 \qquad (5.2)$$

where, y_n is the observation sequence for both MSE1 and MSE2. $\hat{\chi}_{1n}$ and $\hat{\chi}_{2n}$ are estimated state values with ODSA1 which uses chaotic parameter a_1 and ODSA2 which uses chaotic parameter a_2 respectively. L is the length of chaotic sequence for each bit. For each transmitted bit, MSE1 and MSE2 are calculated. If the mean square error corresponding to parameter a_1 (demodulated by ODSA1) is smaller than the mean square error corresponding to parameter a_2 (demodulated by ODSA2), then comparator decides the transmitted signal as digital 0. Otherwise (MSE2 < MSE1) digital 1 is decided.

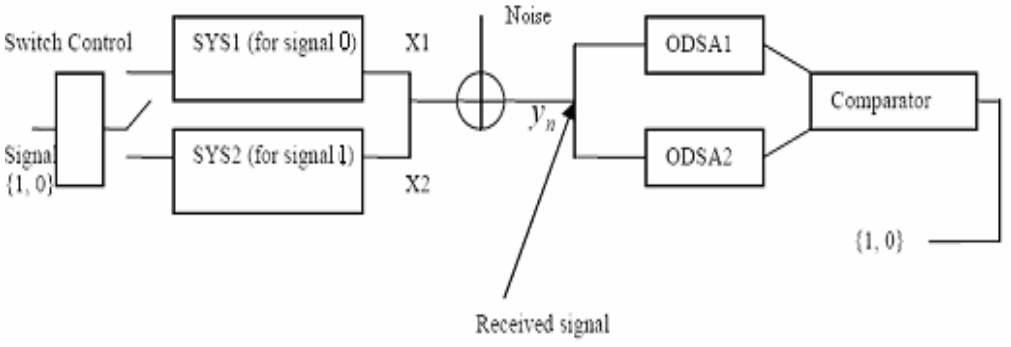

Figure 48 Communication Scheme with ODSA

5.1. Application of ODSA on One Dimensional Chaotic Systems

There are three types of one-dimensional chaotic systems mostly used in secure communication. A brief introduction about these chaotic systems is given in section 4.2. To transmit binary digital signals over the communication channels, one

dimensional chaotic modulation schemes can be used. To demodulate transmitted signal, which is affected by channel noise, ODSA could be a solution.

In this section, the symmetric tent map, which is one of the one-dimensional chaotic systems, is used as the modulator that is figured out as SYS1 or SYS2 in Figure 48. First, models and assumptions to apply ODSA on the symmetric tent map are given. Then the performance of ODSA for different observation noise variances is studied in clear environment. To analyze the performance of ODSA on chaotically modulated signals in the presence of interference, simulations are run for different variance values of interference. After that, BER performance of the symmetric tent map for the application which has the initial states information is analyzed. Last, complexity analysis of ODSA on one dimensional chaotic tent map is given.

5.1.1 Models and Assumptions

To analyze the application of ODSA on one-dimensional chaotic systems, it is assumed that signals are modulated using the symmetric tent map scheme. The symmetric tent map is formulated as

$$\chi(k+1) = a - 1 - a|\chi(k)| \tag{5.3}$$

where a is between 1 and 2.

The motion and observation models in Eq. (2.1) are changed as

$$\begin{aligned} &\text{Motion model} \quad : \quad \chi(k+1) = a - 1 - a|\chi(k)| \ , \\ &\text{Observation model} \quad : \quad z(k) = x(k) + v(k) \end{aligned} \tag{5.4}$$

in clear environment, and

$$\begin{aligned} &\text{Motion model} \quad : \quad \chi(k+1) = a - 1 - a|\chi(k)| \ , \\ &\text{Observation model} \quad : \quad z(k) = g(k, x(k), I(k), v(k)) \end{aligned} \tag{5.5}$$

in presence of interference. g (.) function is described in 5.1.2.2.1 and 5.1.2.2.2.

The constant parameter *a* refers as the input parameter *u(k)* in (2.1) . However, parameter *a* is constant throughout time k unlike *u(k)*.

To analyze the performance of ODSA on chaotic communication systems, the transmitter and the receiver sides are simulated. In the transmitter side, randomly generated binary signals with equal probability are used. These binary signals are modulated using the Eq. (5.5) with the parameter a_1 for binary signal *0* and a_2 for binary signal *1*. Modulated signals are transmitted over the noisy communication channel. This channel adds only Gaussian noise, v*(k)*, to the modulated signals in clear environment whereas in presence of interference I(k) is added to the modulated signals.

In the receiver side, these signals that are modulated at the transmitter side and affected by channel noise are used to estimate transmitted states. As shown Figure 48, there are two ODSA' s which use a_1 and a_2. Using differences between the estimated states and observation sequences, comparator decides whether transmitted signal parameter is a_1 or a_2 that refers to 0 or 1.

As explained in 4.2.4, the receiver side should know the parameter a_1 and a_2. In addition, the initial states of the chaotically modulated sequence are important. In the simulations, it is assumed that the initial states of the transmitted signals are random.

At the last state, the path which variates minimum from the observation values is chosen for each ODSA' s. Then the comparator uses the sum of these variations to decide whether the transmitted signal is a_1 or a_2.

In the presence of interference, the path that has the maximum metric values is chosen for each ODSA's at the last state. Then the comparator uses these metric values to decide whether the transmitted signal is a_1 or a_2.

In some applications, the receiver side can have the knowledge of the initial states. Having the information about the initial states, receiver parts estimates transmitted signals, which are affected by observation noise and interference.

5.1.2 Performance of ODSA on the Symmetric Tent Map Not Having the Knowledge of Initial States

In this section, bit error rate (BER) performance of ODSA on the symmetric tent map is analyzed for the applications which are in clear environment and under interference.

Simulations are run to get BER performance of ODSA. For each simulation, 1000 binary signals with equal probability are generated randomly. For each bit, chaotically modulated signal sequence is generated. Initial state of chaotically modulated signal sequence is assumed uniformly distributed.

In the applications the receiver sides does not have the knowledge of initial states.

The parameters which are used to refer 0 and 1 are taken as $a_1=1.4,\ a_2=1.8$. Also, simulations are run for $a_1=1.4,\ a_2=1.6$.

5.1.2.1 Performance of ODSA on the Symmetric Tent Map in Clear Environment

In order to determine the effects of the observation noise on the performance of the system, simulations are run for different variance values of observation noise which are 0.01, 0.05, 0.1 and 1 . The expected value of the observation noise is taken as 0. Variance values of observation noise are written on the first column of the BER performance tables, Table 23, Table 24, and Table 25. To select the enough the number of samples, simulations are run for 3 different numbers of samples L.

System parameters are;

$[a_1,\ a_2]$ = [1.4, 1.8] and [1.4, 1.6],

expected value of observation noise = 0 ,

quantization # of initial states = 1000 ,

gate size = 1/5000 .

Table 23 BER performance of the symmetric tent map in clear environment, the number of samples L=10

	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.6
	BER	**BER**
σ^2(vk)=0.01	0.017	0.111
σ^2(vk)=0.05	0.188	0.297
σ^2(vk)=0.1	0.327	0.412
σ^2(vk)=1	0.483	0.468

Table 24 BER performance of the symmetric tent map in clear environment, the number of samples L=20

	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.6
	BER	**BER**
σ^2(vk)=0.01	0.027	0.048
σ^2(vk)=0.05	0.051	0.179
σ^2(vk)=0.1	0.150	0.333
σ^2(vk)=1	0.457	0.488

Table 25 BER performance of the symmetric tent map in clear environment, the number of samples L=30

	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.6
	BER	**BER**
σ^2(vk)=0.01	**0.005**	**0.069**
σ^2(vk)=0.05	0.056	0.109
σ^2(vk)=0.1	0.073	0.223
σ^2(vk)=1	0.475	0.474

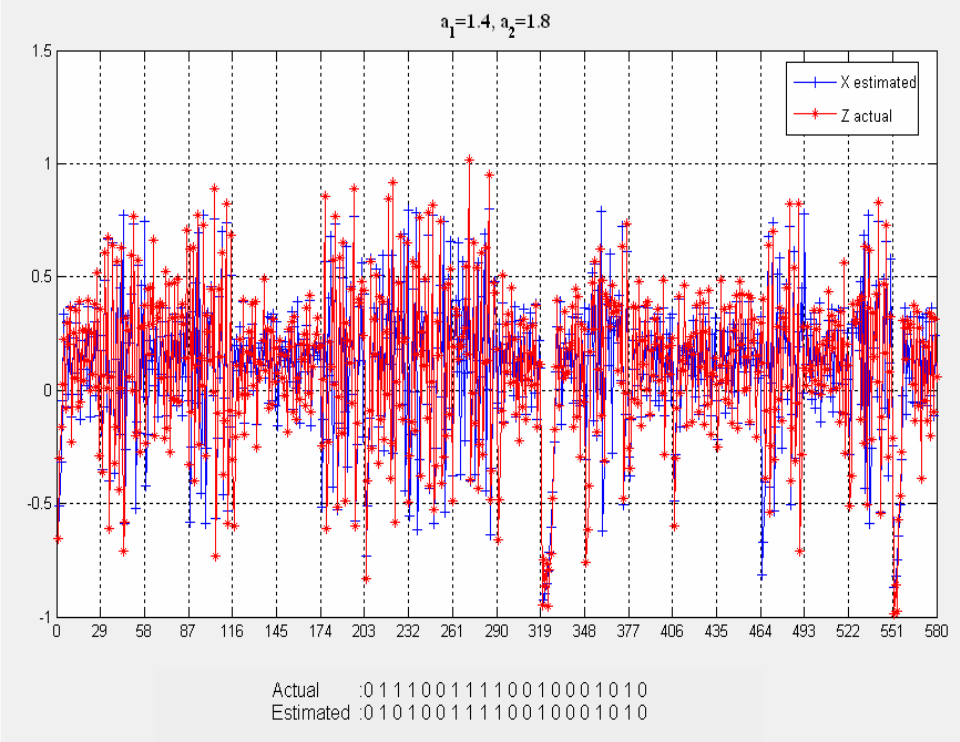

Figure 49 The Symmetric Tent Map with a_1=1.4 , a_2=1.8, σ^2(v(k))=0.01, the number of samples L=30

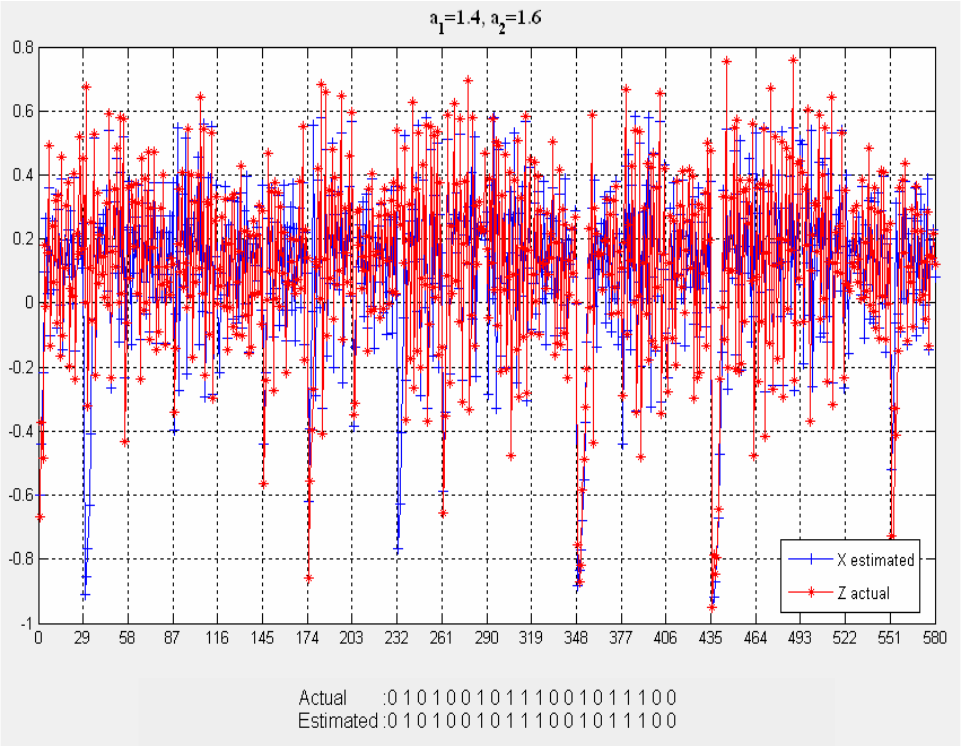

Figure 50 The Symmetric Tent Map with a_1=1.4 , a_2=1.6, σ^2(v(k))=0.01, the number of samples L=30

To see the effect of the number of samples, L, Table 23, Table 24, and Table 25 can be summarized by following figures (Figure 51 and Figure 52). From these figures, it is observed that the number of samples, L can be chosen as 30.

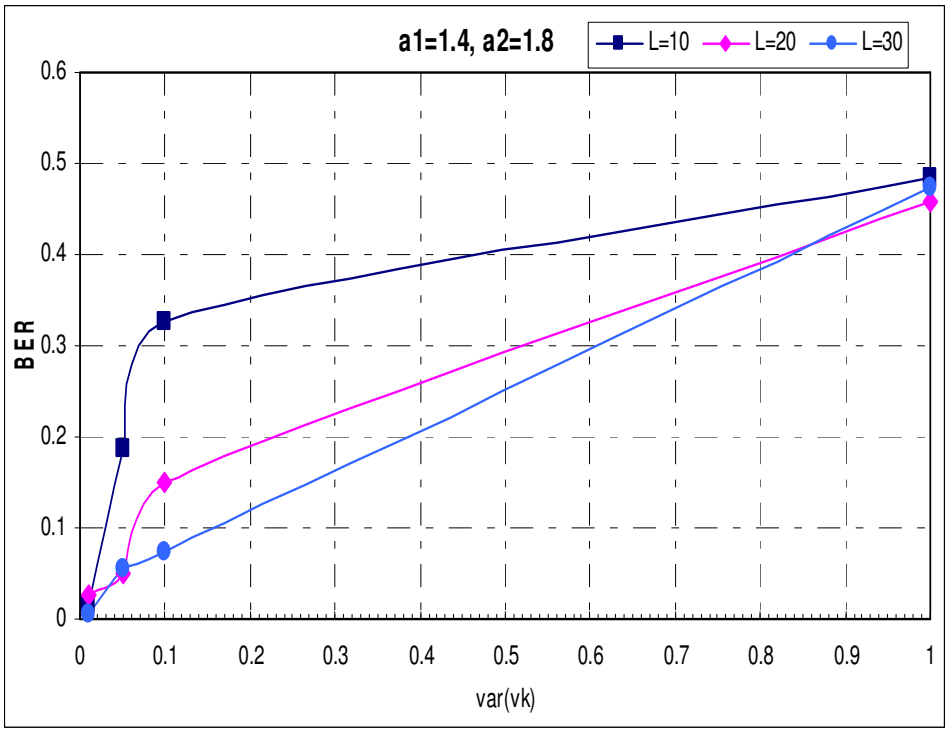

Figure 51 BER performance for different numbers of samples, L

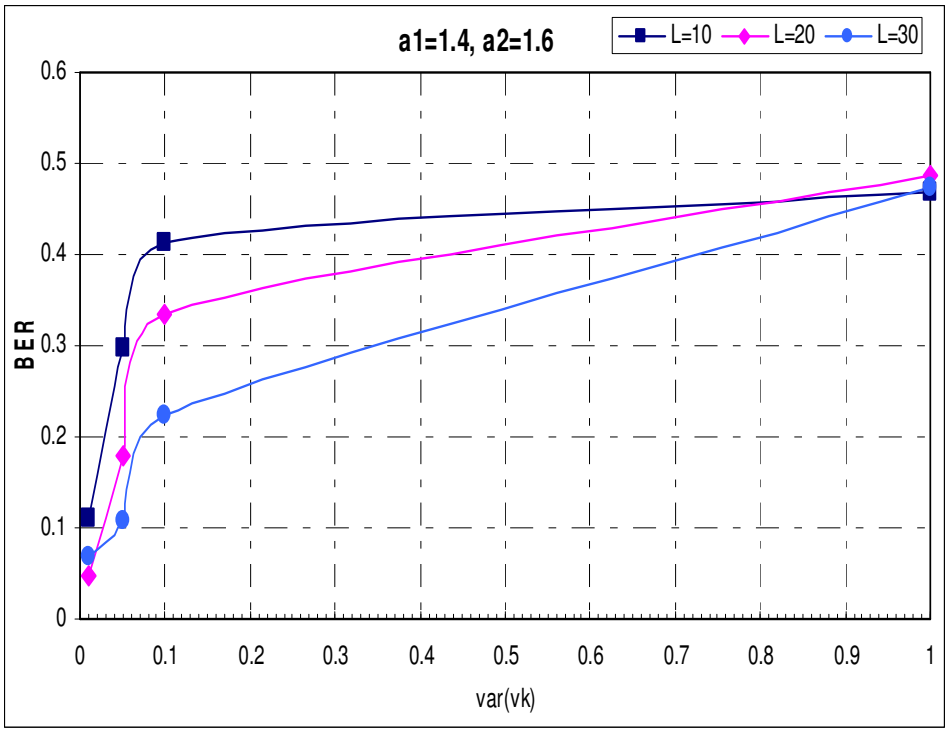

Figure 52 BER performance for different numbers of samples, L

As shown from the above tables, BER performances of ODSA on the symmetric tent map are affected by observation noise variance directly. When the variance of the observation noise increases, BER performance of ODSA decreases.In addition, BER performance depends on parameters a_1 and a_2. While a_1 and a_2 is apart from each other, BER performance increases. However, this situation decreases the security of the chaotic system because of being easily distinguishable. Moreover, Table 23, Table 24 and Table 25 shows that, BER performance of ODSA on the symmetric tent map is better when the number of samples, L is equal to or greater than 20.

5.1.2.2 Performance of ODSA on the Symmetric Tent Map in Presence of Interference

In the previous section, the performance of ODSA was given on the symmetric tent map in clear environments. In this section, the performance of ODSA on one-dimensional chaotic system under interference is analyzed. Simulations are run for three different observation models. In the first model, interference parameter I(k) is directly added to observation noise parameter v(k). Whereas in the second model, observation noise parameter v(k) is multiplied by interference parameter. In the last model, both x(k) and v(k) are multiplied by I(k).

5.1.2.2.1 Observation model I

In these simulations, motion and observation models are described as

Motion model : $\chi(k+1) = a - 1 - a|\chi(k)|$,

Observation model : $z(k) = x(k) + I(k) + v(k)$ (5.6)

where the random vectors x(0), v(*l*), v(m), I(n), and I(p) are independent for all l, m, n, p.

As mentioned in 2.10, the observation z(k) is a linear function of the normal observation vector v(k) and the conditional probability density function of z(k), given that $x(k)=x_q^i(k)$ and I(k), is multivariate normal density function. Then it can be rewritten as

$$p(z(k) \mid x_q^i, I(k)) \approx N(x_q^i + I_{dl}, \sigma_{vk})$$
$$p(z(k) \mid x_q^i) \approx \sum_l p(z(k) \mid x_q^i, I(k) = I_{dl}) * p(I_{dl}) \quad (5.7)$$

Using (5.7) for metric calculation, simulations are run. For each simulation, 1000 binary signals with equal probability are generated randomly. Initial states are generated uniformly in the range of [0 , 1] and maximum number of the states is not limited.

The parameters which are used to refer 0 and 1 are taken as a_1=1.4, a_2=1.8. Also, simulations are run for a_1=1.4, a_2=1.6.

In order to determine effects of the observation noise on the performance of the system, simulations are run different variance values of observation noise which are 0.01, 0.05, and 0.1 . The expected value of the observation noise is taken as 0.

Because BER performances of ODSA on chaotic systems is better when number of samples, L is equal to 20 or 30, simulations are run for these sampling number values. Variance values of observation noise are written in the first row of the BER performance tables, Table 26 and Table 27. Variance values of interferences are given in the first column of the same tables.

System parameters are;

$[\ a_1\ ,\ a_2\ ]$ = [1.4, 1.8] and [1.4, 1.6],

expected value of observation noise = 0 ,

expected value of interference = 0 ,

quantization level of interference = 5,

quantization # of initial states = 1000 ,

gate size = 1/5000

Table 26 BER performance of the symmetric tent map under interference, the number of samples L=20

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0.027	0.048	0.051	0.179	0.150	0.333
σ^2 **(I(k))**=0.001	0.006	0.099	0.101	0.273	0.232	0.389
σ^2 **(I(k))**=0.01	0.043	0.160	0.133	0.274	0.224	0.393
σ^2 **(I(k))**=0.1	0.234	0.387	0.298	0.444	0.369	0.437

Table 26 can be summarized by the following figures (Figure 53 and Figure 54).

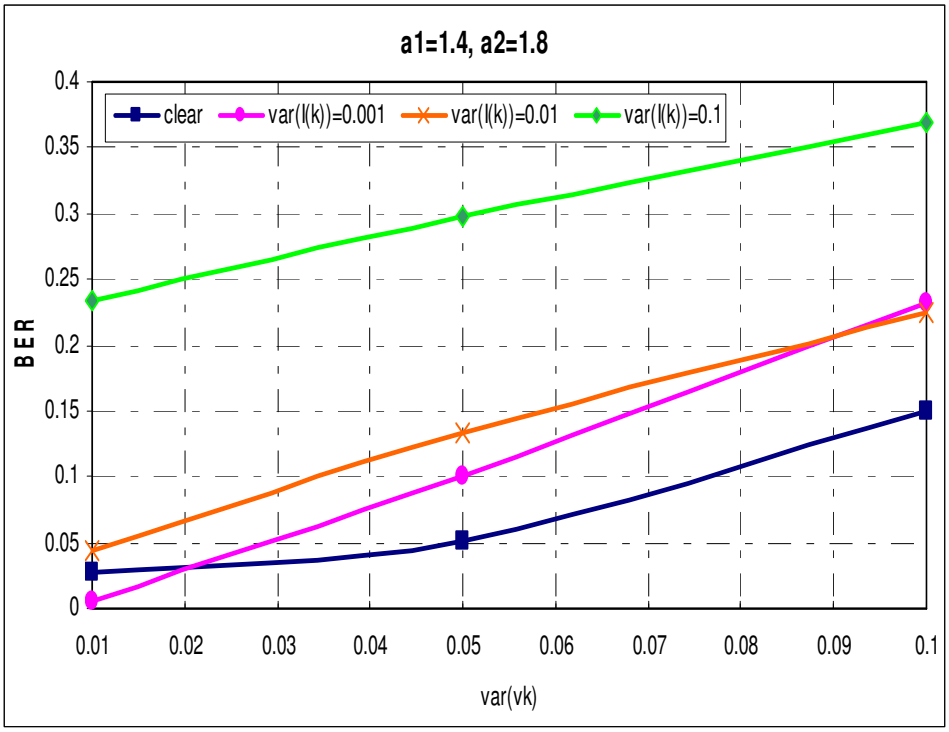

Figure 53 BER performance for different interference variance values at model I, the number of samples L=20

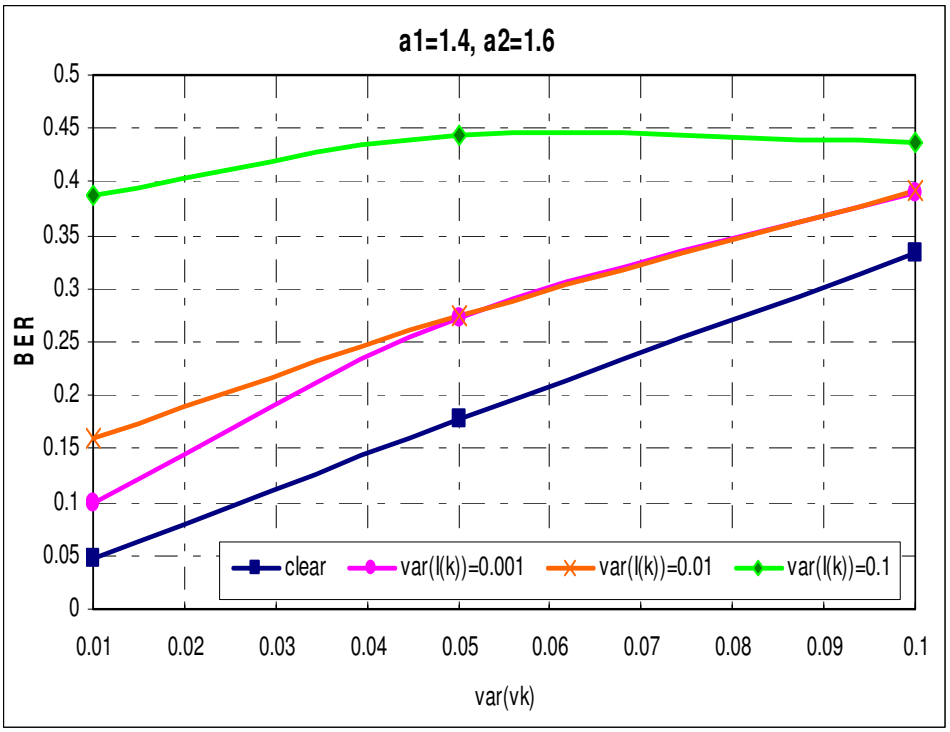

Figure 54 BER performance for different interference variance values at model I, the number of samples L=20

Table 27 BER performance of the symmetric tent map under interference, the number of samples L=30

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0.005	0.069	0.056	0.109	0.073	0.223
σ^2 **(I(k))**=0.001	0.014	0.046	0.053	0.183	0.128	0.336
σ^2 **(I(k))**=0.01	0.019	0.088	**0.063**	**0.215**	0.163	0.328
σ^2 **(I(k))**=0.1	0.150	0.333	0.222	0.383	0.310	0.427

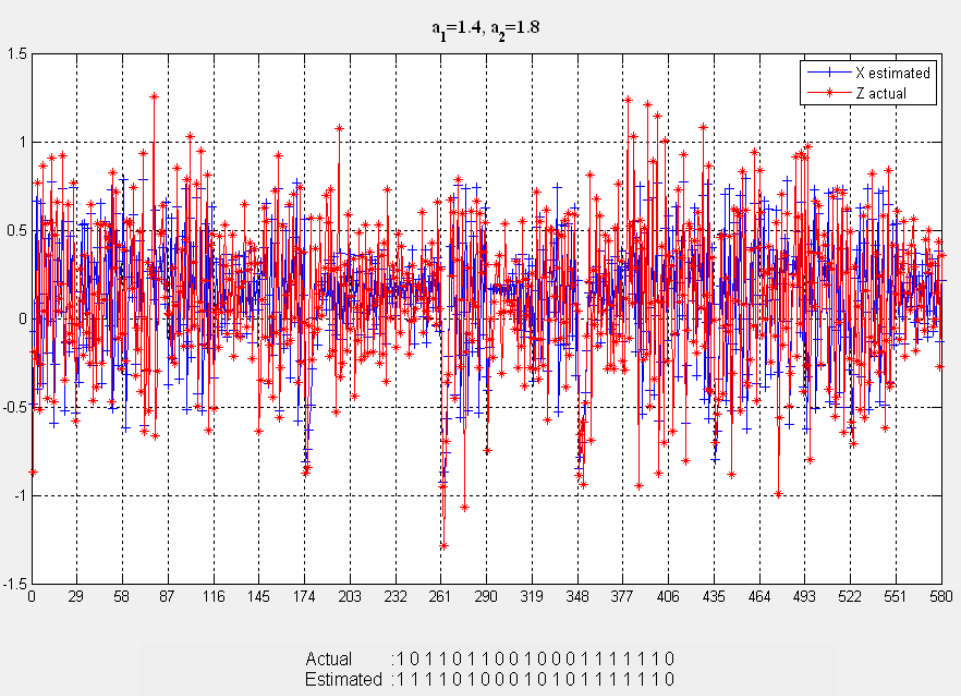

Figure 55 a_1 =1.4, a_2 =1.8, σ^2 (v(k))=0.05, σ^2 (I(k))=0.01, number of samples L=30

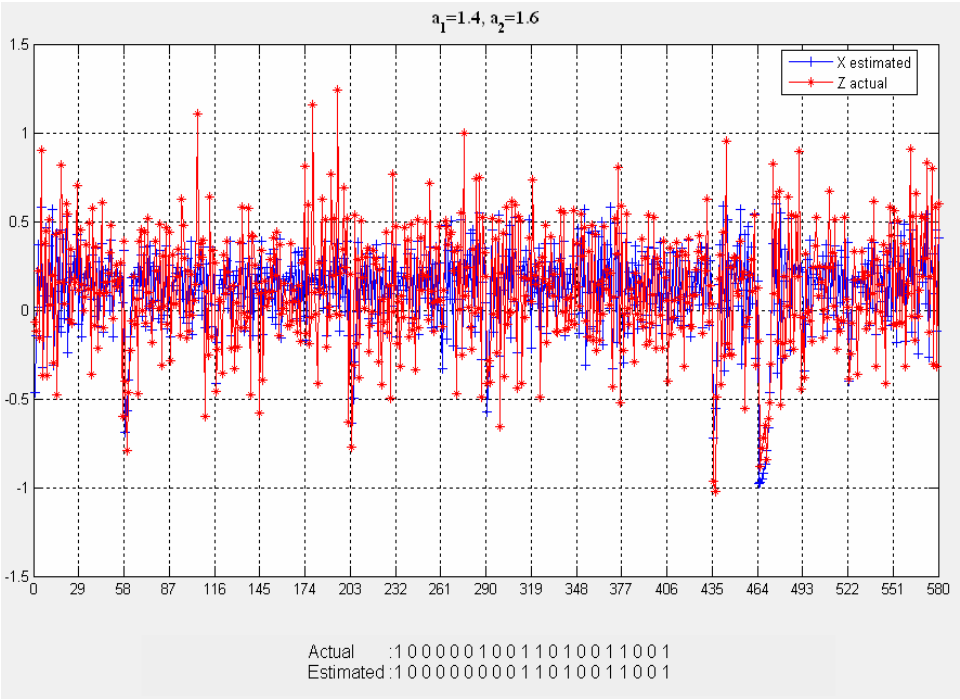

Figure 56 a_1=1.4, a_2=1.6, σ^2(v(k))=0.05, σ^2(I(k))=0.01, number of samples L=30

Table 27 can be summarized by the following figures (Figure 57 and Figure 58).

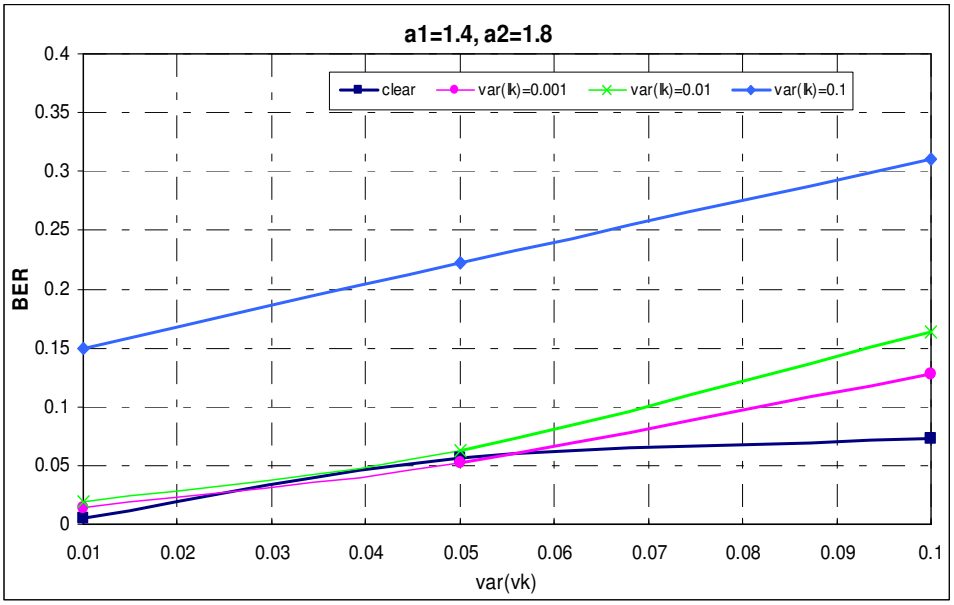

Figure 57 BER performance for different interference variance values at model I, the number of samples L=30

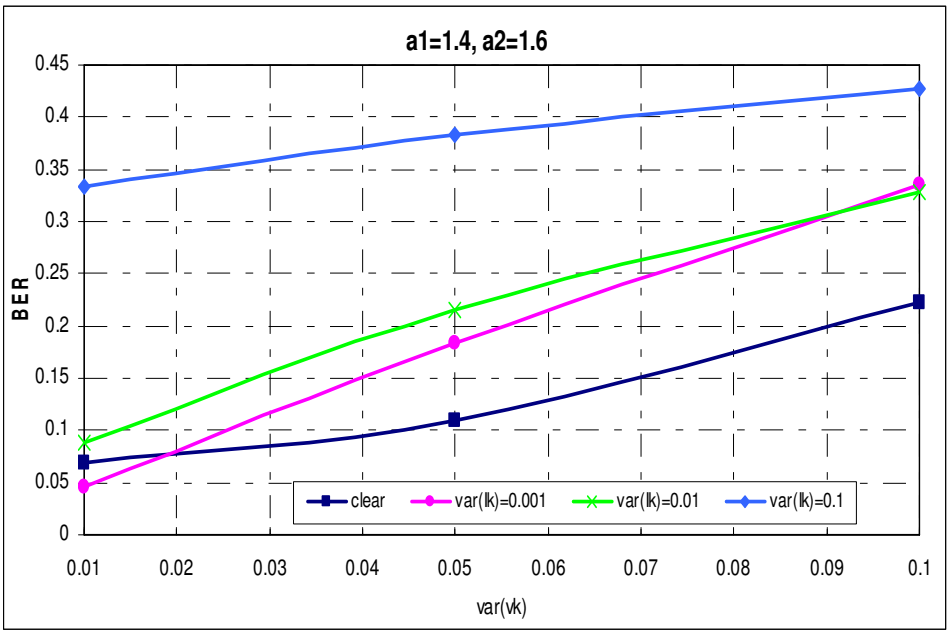

Figure 58 BER performance for different interference variance values at model I, the number of samples L=30

Basically, BER performance of the system is more successful when L is 30. Also it can be said that, the interference is more effective on the symmetric tent map when a's are close to each other.

5.1.2.2.2 Observation model II

In these simulations, motion and observation models are described as

Motion model : $\chi(k+1) = a - 1 - a|\chi(k)|$,

Observation model : $z(k) = x(k) + (1+I^2(k))\ v(k)$ (5.8)

where the random vectors x(0), v(*l*), v(m), I(n), and I(p) are independent for all l, m, n, p.

As mentioned in 2.10, the observation z(k) is a linear function of the normal observation vector v(k) and the conditional probability density function of z(k), given that $x(k)=x_q^i(k)$ and I(k), is multivariate normal density function with mean $x_q^i(k)$ and variance value $\sigma^2_{dl} = (1 + I_{dl}^2)^2 \sigma^2_{vk}$.

Then it can be rewritten as

$$p(z(k) \mid x_q^i, I(k)) \approx N(x_q^i, \sigma^2_{dl})$$

$$p(z(k) \mid x_q^i) \approx \sum_l p(z(k) \mid x_q^i, I(k) = I_{dl}) * p(I_{dl}) \qquad (5.9)$$

Using (5.9) for metric calculation, simulations are run. For each simulation, 1000 binary signals with equal probability are generated randomly. Initial states are generated uniformly in the range of [0 , 1] and maximum number of the states is not limited.

The parameters which are used to refer 0 and 1 are taken as a_1=1.4, a_2=1.8. Also, simulations are run for a_1=1.4, a_2=1.6.

In order to determine effects of the observation noise on the performance of the system, simulations are run different variance values of observation noise which are 0.01, 0.05, and 0.1 . The expected value of the observation noise is taken as 0.

Simulations are run for 30 sample values. Variance values of observation noise is the first row of the BER performance table, Table 28. Variance values of interferences are given in the first column of this table.

System parameters are;

[a_1 , a_2]	= [1.4, 1.8] and [1.4, 1.6],
expected value of observation noise	= 0 ,
expected value of interference	= 0 ,
quantization level of interference	= 5,
quantization # of initial states	= 1000 ,
gate size	= 1/5000

Table 28 BER performance of the symmetric tent map under interference, the number of samples L=30

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0.005	0.069	0.056	0.109	0.073	0.223
σ^2 **(I(k))**=0.001	0.009	0.047	0.063	0.189	0.141	0.332
σ^2 **(I(k))**=0.01	0.010	0.049	0.065	0.206	0.147	0.353
σ^2 **(I(k))**=0.1	0.011	0.060	0.073	0.215	0.169	0.339

Table 25can be summarized by following figures (Figure 59 and Figure 60).

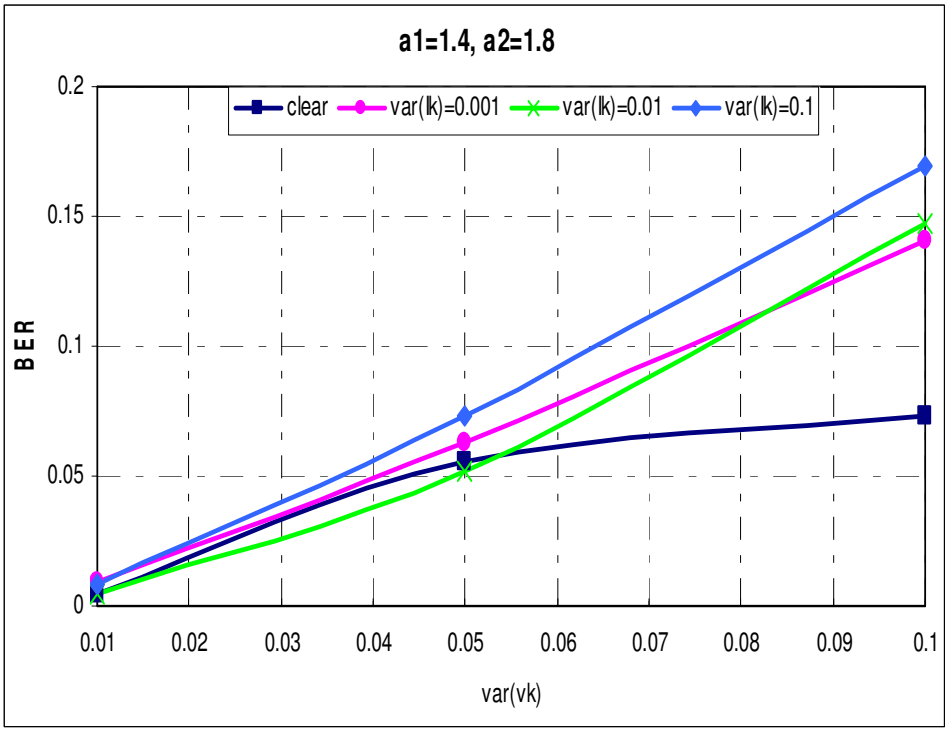

Figure 59 BER performance for different interference variance values at model II

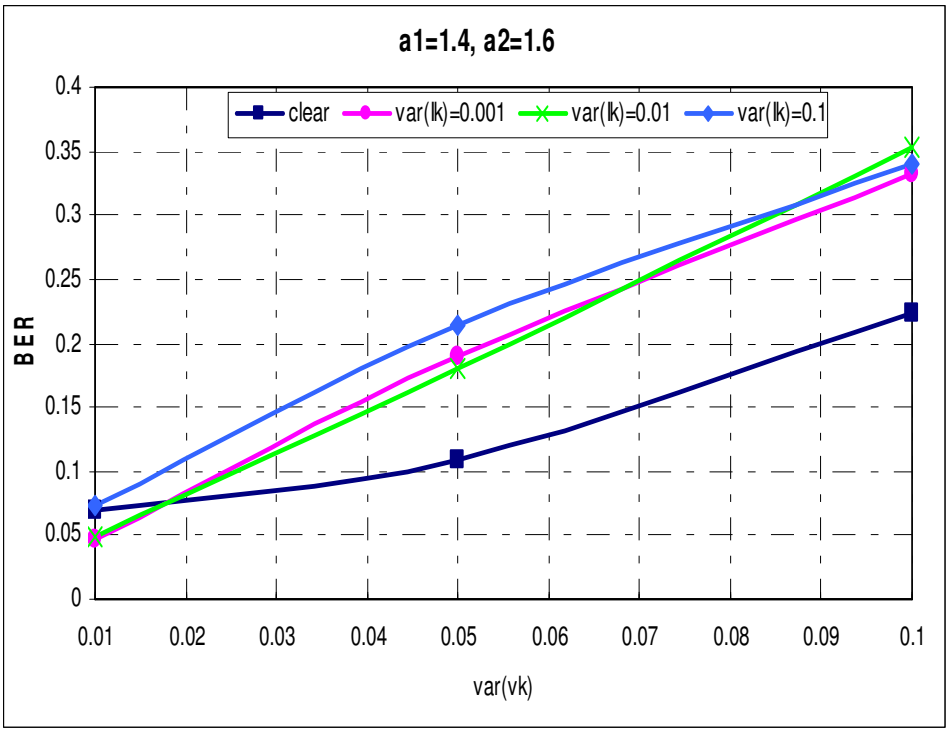

Figure 60 BER performance for different interference variance values at model II

5.1.2.2.3 Observation model III

In these simulations, motion and observation models are described as

Motion model : $\chi(k+1) = a - 1 - a|\chi(k)|$,

Observation model : $z(k) = x(k)\, I(k) + (1+I^2(k))\, v(k)$ (5.10)

where the random vectors x(0), v(*l*), v(m), I(n), and I(p) are independent for all l, m, n, p.

Because the state vector x(k) is multiplied by interference vector I(k) in the observation model, then the state vector fades. Because of this, this channel can be thought as fading channels.

As mentioned in 2.10, the observation z(k) is a linear function of the normal observation vector v(k) and the conditional probability density function of z(k), given that x(k)=x_q^i(k) and I(k), is multivariate normal density function with mean $x_q^i * I_{dl}$ and variance value $\sigma^2{}_{dl} = (1 + I_{dl}^2)^2 \sigma^2{}_{vk}$.

Then it can be rewritten as

$$p(z(k) \mid x_q^i, I(k)) \approx N(x_q^i * I_{dl}, \sigma^2{}_{dl})$$

$$p(z(k) \mid x_q^i) \approx \sum_l p(z(k) \mid x_q^i, I(k) = I_{dl}) * p(I_{dl}) \qquad (5.11)$$

Using (5.11) for metric calculation, simulations are run. For each simulation, 1000 binary signals with equal probability are generated randomly. Initial states are generated uniformly in the range of [0 , 1] and maximum number of the states is not limited.

The parameters which are used to refer 0 and 1 are taken as a_1=1.4, a_2=1.8. Also, simulations are run for a_1=1.4, a_2=1.6.

In order to determine effects of the observation noise on the performance of the system, simulations are run different variance values of observation noise that are 0.01, 0.05, and 0.1. The expected value of the observation noise is taken as 0.

Simulations are run for 30 sample values. Variance values of observation noise is the first row of the BER performance table, Table 29. Variance values of interferences are given in the first column of this table.

System parameters are;

[a_1 , a_2]	= [1.4, 1.8] and [1.4, 1.6],
expected value of observation noise	= 0 ,
expected value of interference	= 1 ,
quantization level of interference	= 5,
quantization # of initial states	= 1000 ,
gate size	= 1/5000

Table 29 BER performance of the symmetric tent map under interference, number of samples L=30

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0.005	0.069	0.056	0.109	0.073	0.223
σ^2 **(I(k))**=0.001	0.031	0.105	0.243	0.390	0.427	0.479
σ^2 **(I(k))**=0.01	0.050	0.096	0.343	0.436	0.448	0.48
σ^2 **(I(k))**=0.1	0.139	0.166	0.330	0.450	0.490	0.498

Table 29 can be summarized by following figures (Figure 61 and Figure 62).

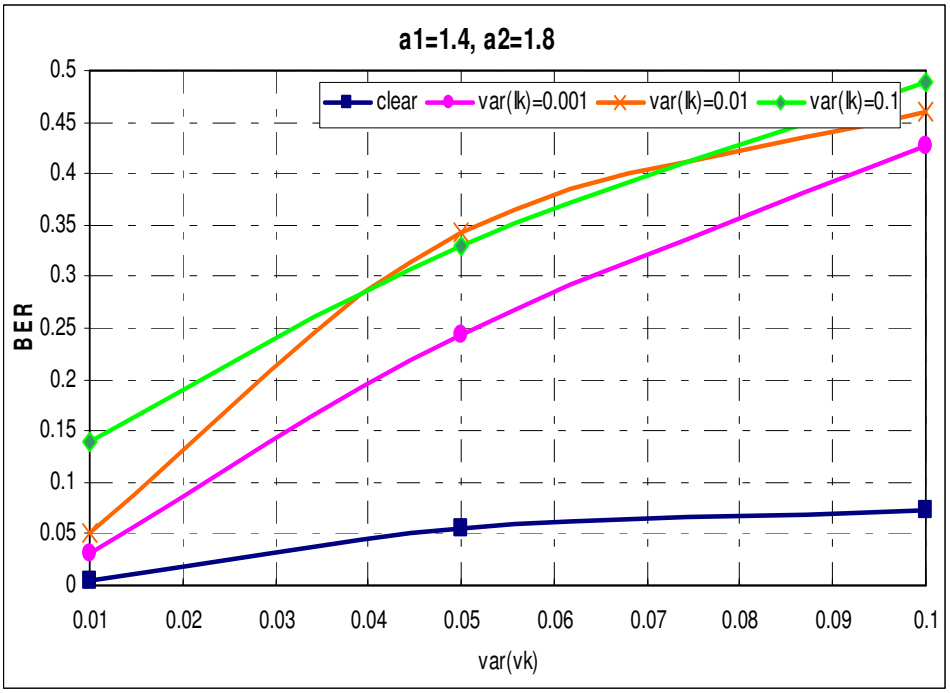

Figure 61 BER performance for different interference variance values at model III

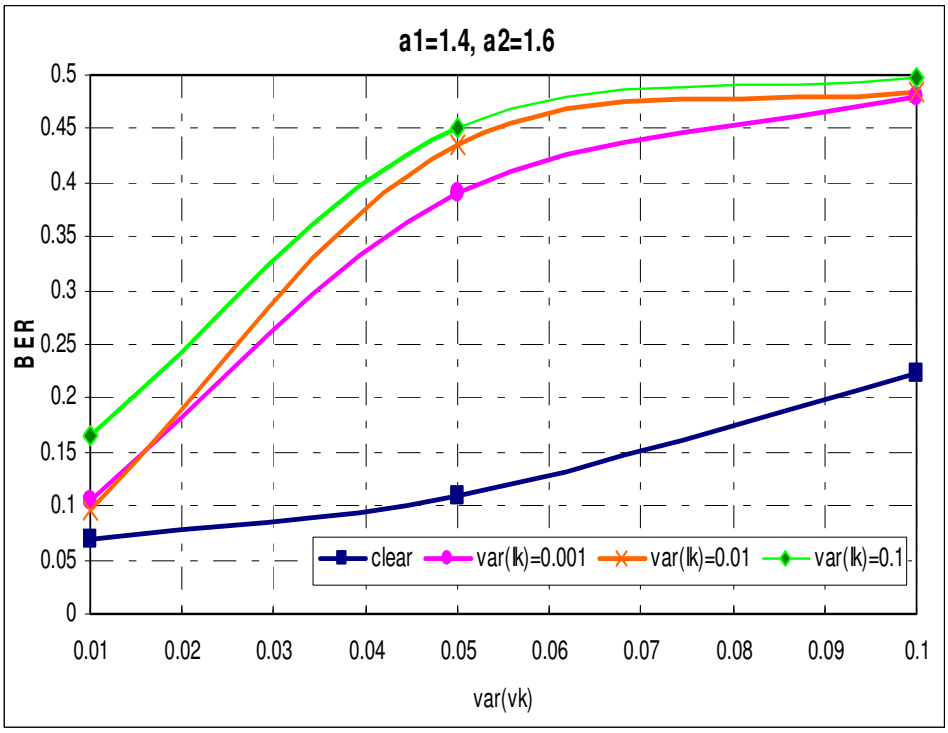

Figure 62 BER performance for different interference variance values at model III

Analyzing Table 27, Table 28, and Table 29, it is observed that, when interference is added to the observation models as a multiplier, BER performance of the system decreases depending on the variance values of interference. Interference is more effective when it is multiplied by x(k) and v(k).

5.1.3 Performance of ODSA on the Symmetric Tent Map Having the Knowledge of Initial States

As described in section 5.1.2, chaotic communication is based on the transmition in which binary signals are transmitted using the chaotic signals. In the transmitter side, to represent 0 or 1 a signal block is generated which is transmitted over the communication channel and reached to the receivers. These receivers do not

have any information about initial states of the transmitted signals. For this case, BER performance of ODSA on the symmetric tent map was given in 5.1.2 and 5.1.2.2.

In this section, it is assumed that receiver sides have information about initial states of the transmitted signal in addition to a_1 and a_2 , and simulation are run to have BER performance of ODSA for this case.

For each simulation, 1000 binary signals with equal probability are generated randomly. Initial states of chaotically modulated signal sequences are taken constant. The parameters which are used to refer 0 and 1 are taken as $a_1=1.4,\ a_2=1.8$. Also, simulations are run for $a_1=1.4,\ a_2=1.6$.

In order to determine the effects of the observation noise on the performance of the system, simulations are run for different variance values of observation noise which are 0.01, 0.05, 0.1 and 1. The expected value of the observation noise is taken as 0. Variance values of observation noise are written on the BER performance tables. The number of samples L is taken 30.

In the simulations, the receiver sides have the information about the initial state value. This information is that the initial state is one of the element of sequences which is formulated as;

$x_k = \frac{2k-1}{2 \cdot n}, \quad k = 1,2,3...,n.$ where n is equal to initial state quantization level, 100. In the simulations, chaotically modulated signal sequences are generated as 0.305 by taking k=31.

System parameters are;

$[a_1,\ a_2]$ = [1.4, 1.8] and [1.4, 1.6],

expected value of observation noise = 0 ,

quantization # of initial states = 100 ,

gate size = 1/5000

initial state = 0.305 .

For the clear environment application, BER performance of ODSA on the symmetric tent map is given by Table 30 Variance values of observation noise are written in the first column of the BER performance table, Table 30. Analyzing both Table 23 and Table 30, there is an improvement on BER performance of ODSA in clear environment applications in which receiver sides know the initial states.

Table 30 BER performance of the symmetric tent map in clear environment

	$a_1=1.4,\ a_2=1.8$	$a_1=1.4,\ a_2=1.6$
	BER	**BER**
σ^2 **(vk)=0.001**	0	0
σ^2 **(vk)=0.01**	0	0.009
σ^2 **(vk)=0.05**	0.032	0.079
σ^2 **(vk)=0.1**	0.070	0.209
σ^2 **(vk)=1**	0.467	0.494

In presence of interference for the model I application as formulated in (5.6), BER performance of ODSA is summarized by Table 31. Variance values of observation noise is the first row of the BER performance table, Table 31. Variance values of interferences are given in the first column of this table. When Table 31 is compared with Table 27, in the applications having the knowledge of initial states there is significant decrease on BER performance of ODSA.

Table 31 BER performance of the symmetric tent map under interference, model I

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0.009	0.032	0.079	0.070	0.209
σ^2 **(I(k))**=0.001	0.001	0.007	0.041	0.092	0.095	0.202
σ^2 **(I(k))**=0.01	0.006	0.028	0.045	0.09	0.104	0.217
σ^2 **(I(k))**=0.1	0.136	0.27	0.151	0.274	0.232	0.349

Table 31 can be summarized by the following figures, Figure 63 and Figure 64.

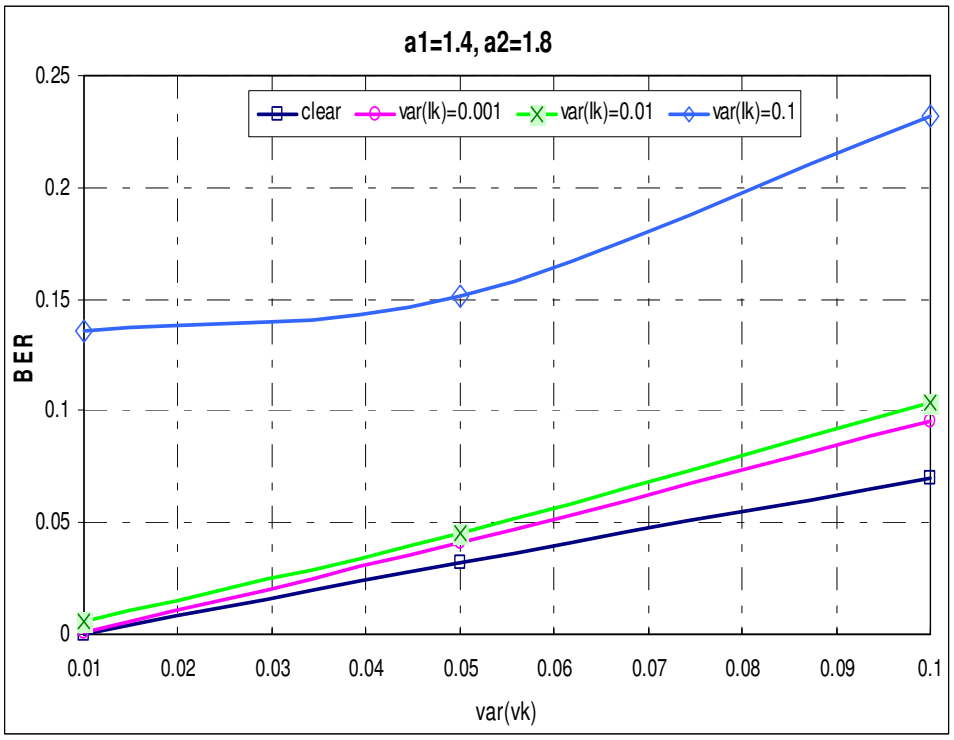

Figure 63 BER performance of ODSA knowing the initial states at model I

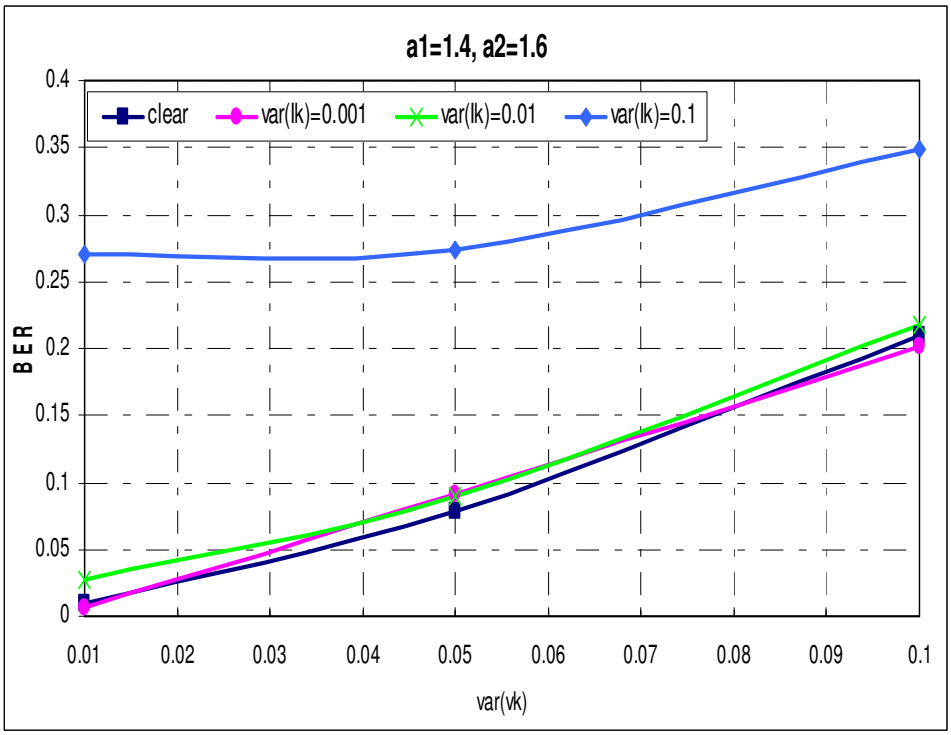

Figure 64 BER performance of ODSA knowing the initial states at model I

In presence of interference for the model II application as formulated in (5.8), BER performance of ODSA is summarized by Table 32. Variance values of observation noise are written in the first column of the BER performance table, Table 32.When Table 32 is compared with Table 28, it can be concluded that having the knowledge of initial states causes a decrease on BER performance of ODSA.

Table 32 BER performance of the symmetric tent map under interference, model II

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0.009	0.032	0.079	0.070	0.209
σ^2 **(I(k))**=0.001	0	0.004	0.031	0.097	0.077	0.195
σ^2 **(I(k))**=0.01	0	0.005	0.036	0.094	0.101	0.194
σ^2 **(I(k))**=0.1	0	0.008	0.034	0.117	0.113	0.226

Table 32 can be summarized by the following figures (Figure 65 and Figure 66).

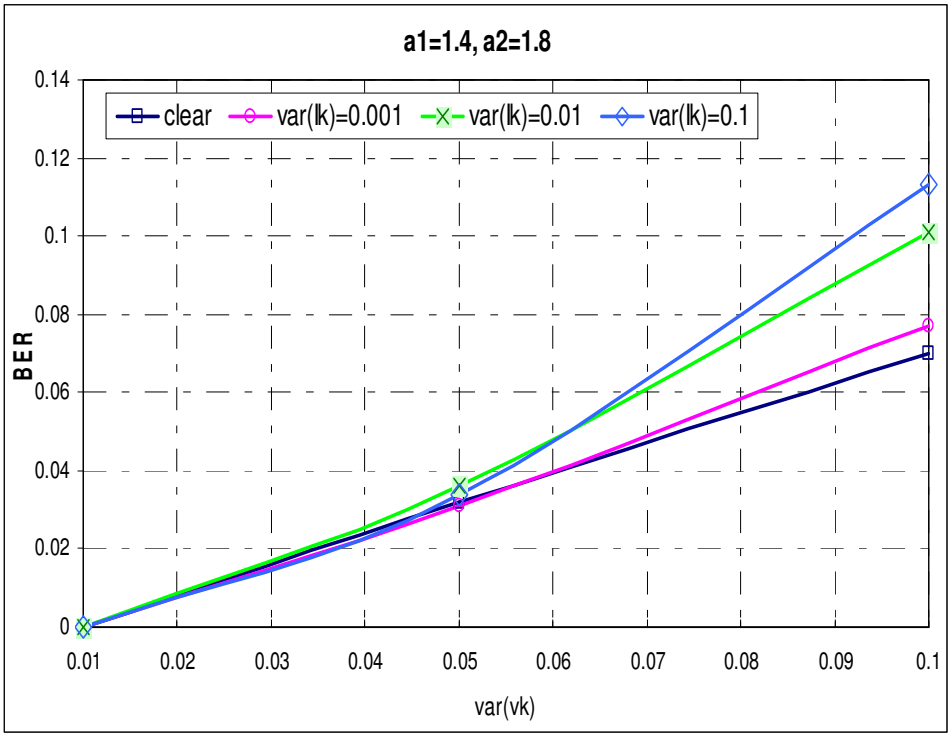

Figure 65 BER performance of ODSA knowing the initial states at model II

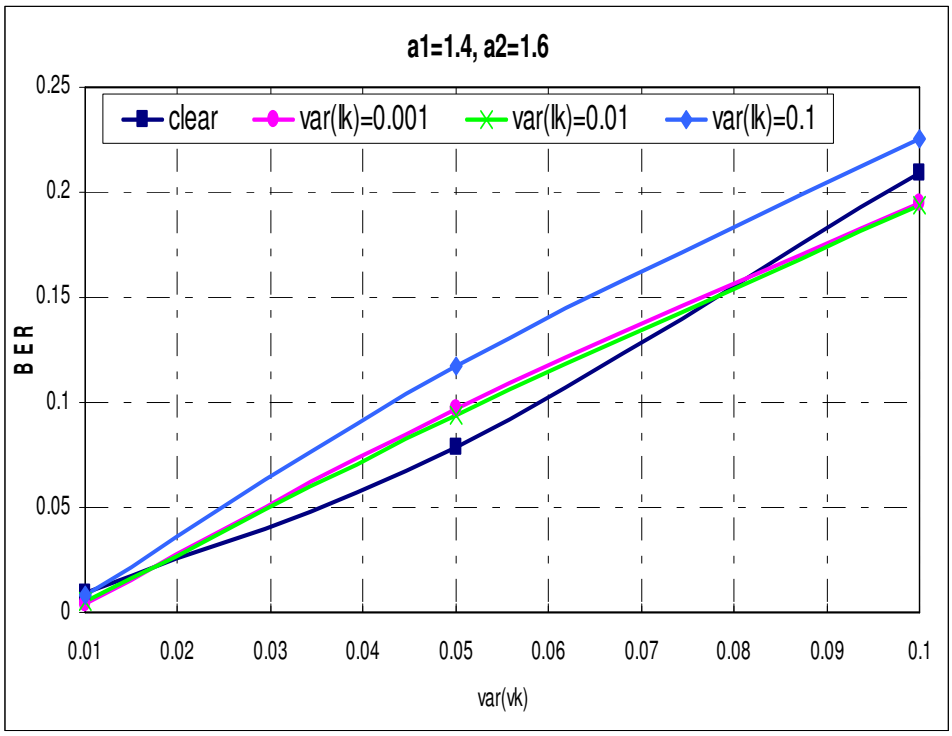

Figure 66 BER performance of ODSA knowing the initial states at model II

In presence of interference for the model III application, BER performance of ODSA is summarized by Table 33. Similar effects of having information about initial states on BER performance of ODSA are realized when comparing Table 33 and Table 29.

Table 33 BER performance of the symmetric tent map under interference, model III

	σ^2(vk)=0.01		**σ^2(vk)=0.05**		**σ^2(vk)=0.1**	
	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6	a_1=1.4 a_2=1.8	a_1=1.4 a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0.009	0.032	0.079	0.070	0.209
σ^2 **(I(k))**=0.001	0.128	0.254	0.468	0.432	0.496	0.479
σ^2 **(I(k))**=0.01	0.144	0.284	0.435	0.465	0.503	0.486
σ^2 **(I(k))**=0.1	0.346	0.373	0.385	0.445	0.468	0.493

Table 26 can be summarized by the following figures (Figure 67 and Figure 68).

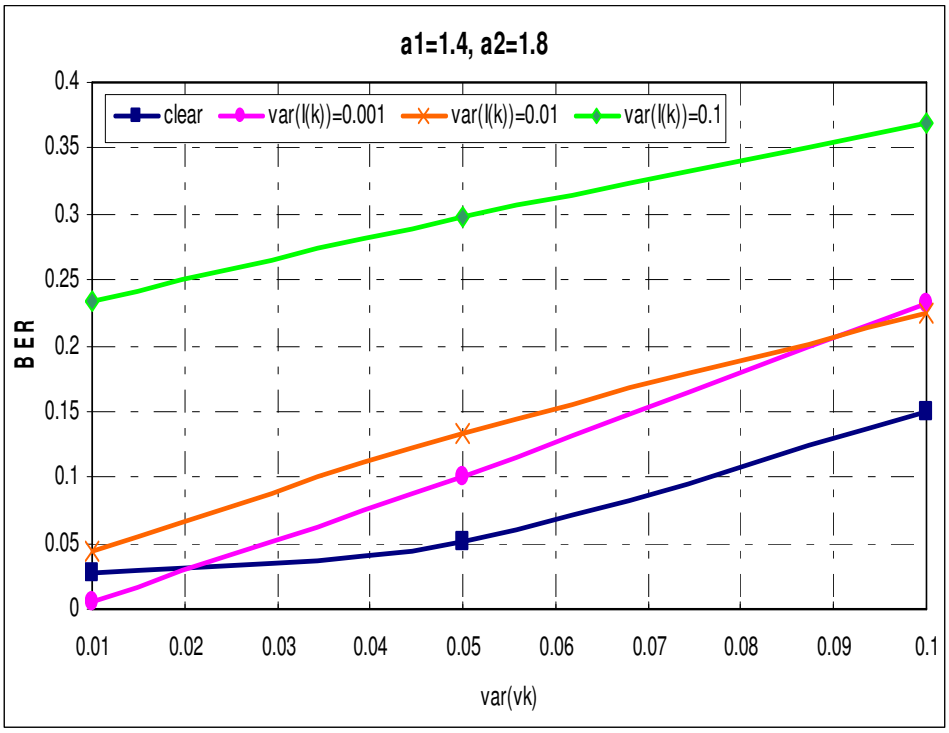

Figure 67 BER performance of ODSA knowing the initial states at model III

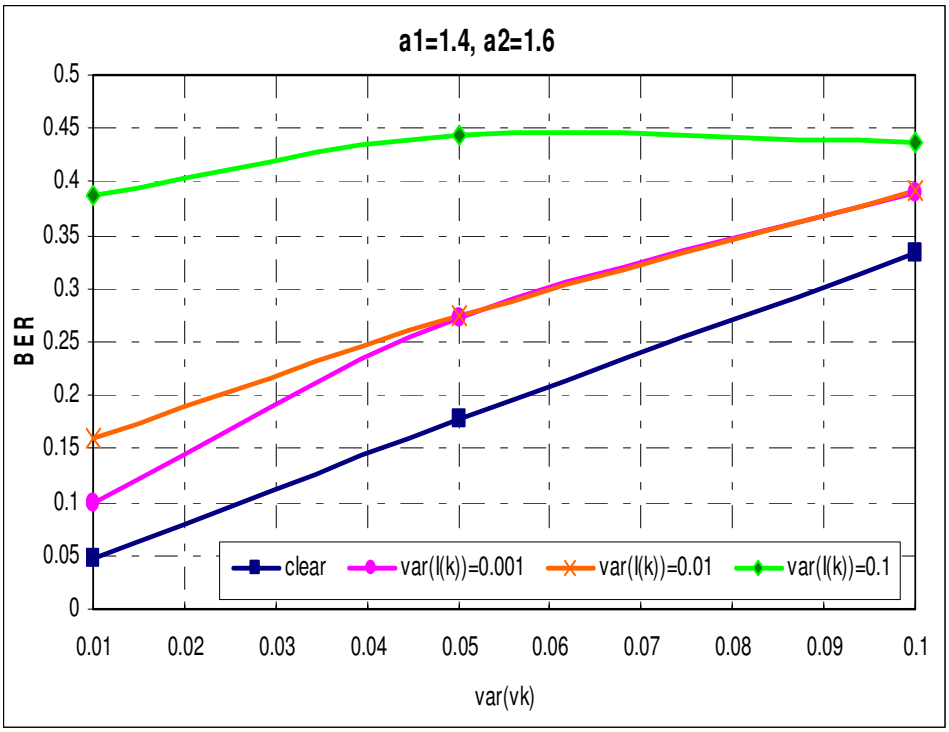

Figure 68 BER performance of ODSA knowing the initial states at model III

5.1.4 Complexity Analysis of ODSA on One Dimensional Chaotic Systems

Brief information about one-dimensional chaotic systems is given in 4.2. In one-dimensional chaotic systems with or without interference, motion models use *x(k)* and *a* as parameters. Only parameter which is quantizated is *x(k)* at time k=0, 1, 2 …, L. Other parameters defined in ODSA such as *w(k)* and *u(k)* are not used in one-dimensional chaotic systems. Therefore, complexity of ODSA is decreased.

In addition, observation models use *I(k)* only in the presence of interference. In clear environment, there is not any quantizated parameter and complexity of ODSA is decreased one more time.

The runtime of the program written for ODSA is determined by the number of initial states. Let the maximum time consumption at each state be t_s at time k in presence of interference. Then, since the time consumption at each state will be approximately same, the time consumption will be $t_s n_x L$. The total program runtime will be equal to 2 $t_s n_x L$ because of two ODSA's.

Moreover, in clear environment, time consumption at each state is smaller than t_s , then the total program runtime is less than 2 $t_s n_x L$.

To have a sense about the program runtime, following simulations are run. The parameters are kept similar with in the case 2.11.2.

Motion and observation models are

Motion model : $\chi(k+1) = a - 1 - a|\chi(k)|$,

Observation model : z(k) = x (k) + I (k) + v (k).

Parameters used in these simulations are as follows;

- Initial states are uniform in the range of 0 and 1,
- variance of I(k)=0.01, expected value of I(k)=0, quantization level of I(k)=3,
- quantization level of initial states, Q(x(0)) =100.

As shown in Figure 8 and Figure 69, runtime of simulations in one dimensional chaotic systems is less than general applications of ODSA which is due to the number of quantized parameters.

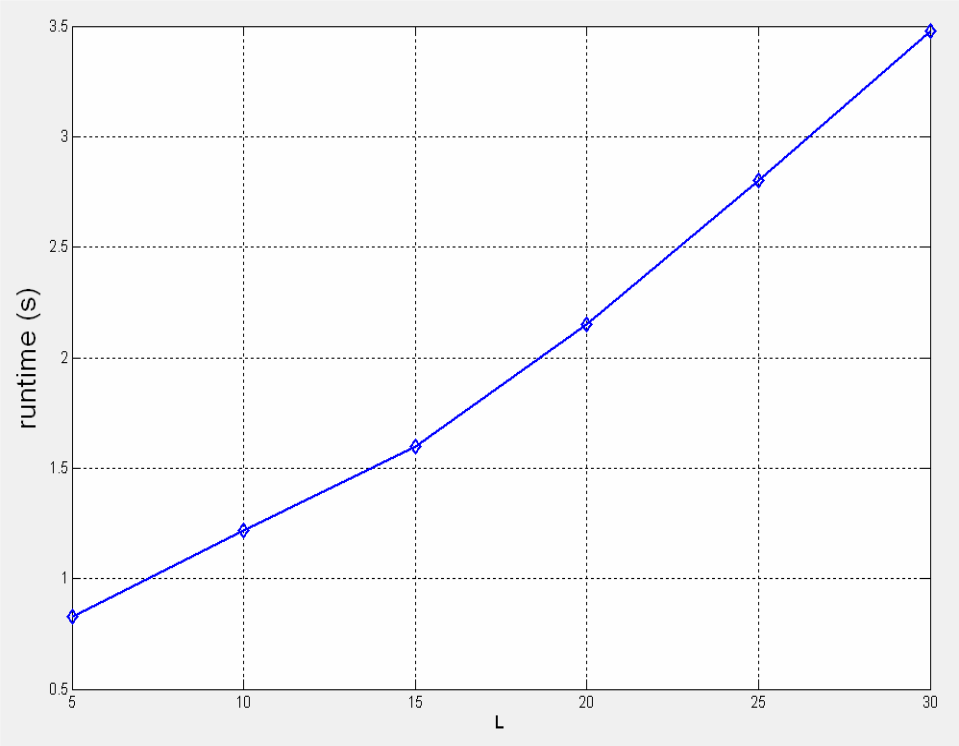

Figure 69 Program runtime for one bit execution on the symmetric tent map

5.2. Application of ODSA on the NCA Map

Another application of ODSA on chaotic system is the NCA map that is described in 4.3.

In this section, the nonlinear chaotic algorithm map is used as the motion model. First, models and assumptions to applicate ODSA on NCA are given. Then the performance of ODSA for different observation noise variances is studied.

5.2.1 Models and Assumptions

To analyze the application of ODSA to the NCA map, it is assumed that signals are modulated using parameters in the range of,

$x_n \in (0, 1)$,

$\alpha \in (0, 1.4]$,

$\beta \in [5, 43]$.

Motion model is

$$\chi_{n+1} = (1-\beta^{-4}) \cdot ctg\left(\frac{\alpha}{1+\beta}\right) \cdot \left(1+\frac{1}{\beta}\right)^{\beta} \cdot tg(\alpha\chi_n) \cdot (1-\chi_n)^{\beta}, \quad (5.12)$$

and observation model is ;

$$y(k) = x(k) + v(k) \quad (5.13)$$

in clear environment. In presence of interference observation model is ;

$$y(k) = g(k, x(k), I(k), v(k)) \quad (5.14)$$

g (.) function is described in 5.1.2.2.1 and 5.1.2.2.2. In the motion model, there are two constant parameter α and β .These parameters refer as the input parameter *u(k)* in Eq. (2.1) .

To analyze the performance of ODSA on the nonlinear chaotic communication system, the transmitter and the receiver sides are simulated. In the transmitter side, randomly generated binary signals with equal probability are used. Whether being 0 or 1, these binary signals are modulated using the Eq. (5.12). In Eq. (5.12) there are two parameters α and β and binary signals can be modulated one of these parameters. For example, assuming β is constant, binary signals are modulated with the parameters α_1 or α_2. On the other hand, binary signals can be modulated with the parameters β_1 or β_2 where α is constant. Moreover, to increase the security of the communication different α and β pairs can be used where 0 is represented by [α_1, β_1] and 1 is represented by [α_2, β_2].

After modulating binary signals, modulated signals are transmitted over the noisy communication channel. This channel adds Gaussian noise, v*(k)*, and interference noise to the modulated signals.

In the receiver side, these noisy signals are used to estimate transmitted states. As explained in 5.1, on the receiver side there are two ODSA's. Each ODSA

uses different parameters, for example, if one of two ODSA's uses α_1, other one uses α_2 where β is constant. Using these parameters, they calculate differences between the estimated states and observation sequence. Each ODSA sends total differences to comparator that decides whether transmitted signal parameter is α_1 or α_2. that refers 0 or 1.

As explained in 4.3, the receiver side should know the parameters α_1, α_2.and β. In addition, initial states of the chaotically modulated sequence are important. In the simulations, the receiver sides have the information about the initial state value. This information is that the initial state is one of the element of sequences which is formulated as;

$x_k = \frac{2k-1}{2 \cdot n}$, $k = 1,2,3...,n.$ where n is equal to initial state quantization level, 10. In the simulations, chaotically modulated signal sequences are generated as 0.25 by taking k=3.

5.2.2 Performance of ODSA on the NCA Map in Clear Environment

In this section, to analyze performance of ODSA on the nonlinear chaotic system, simulation results are given. For each simulation, 1000 binary signals with equal probability, 0 and 1, are generated randomly. For each bit, initial state is taken constant as given 5.2.1 because of the sensitivity to initial states . In addition, maximum number of the states is not limited and in the observation model, interference parameter is not added to simulate clear environment.

In order to determine effects of the parameters on BER performance simulations are run both α and β.

5.2.2.1 Simulations for constant parameter α

These simulations are run for different values of the parameter β as variance values of observation noise changing. The parameter α is taken constant. The parameters values used in the simulations are;

variance of interference noise $= 0$,

expected value of interference noise = 0 ,

expected value of observation noise = 0 ,

quantization # of initial states = 10 ,

number of samples, L = 30 ,

gate size = 1/5000 ,

α = 0.9,

initial state = 0.25 .

BER performance of the system with parameter β is given Table 34. In the first column of Table 34, variance values of observation noise are given. Figure 70 and Figure 71 show chaotically modulated signal sequences for first 10 bits of randomly generated 1000 bits. In these figures, observation signal sequences and estimated state values are figured out on the same time.

Table 34 BER performance of the system with parameter β

	β_1=5 β_2=5.1	β_1=15 β_2=25	β_1=10 β_2=30	β_1=10 β_2=11.5
	BER	**BER**	**BER**	**BER**
σ^2 **(v(k))**=0.001	0	0	0	0
σ^2 **(v(k))**=0.01	0	0	0	0
σ^2 **(v(k))**=0.05	0	0.008	0	0
σ^2 **(v(k))**=0.1	0	**0.035**	**0.008**	0.007
σ^2 **(v(k))**=0.5	0.141	0.264	0.222	0.268

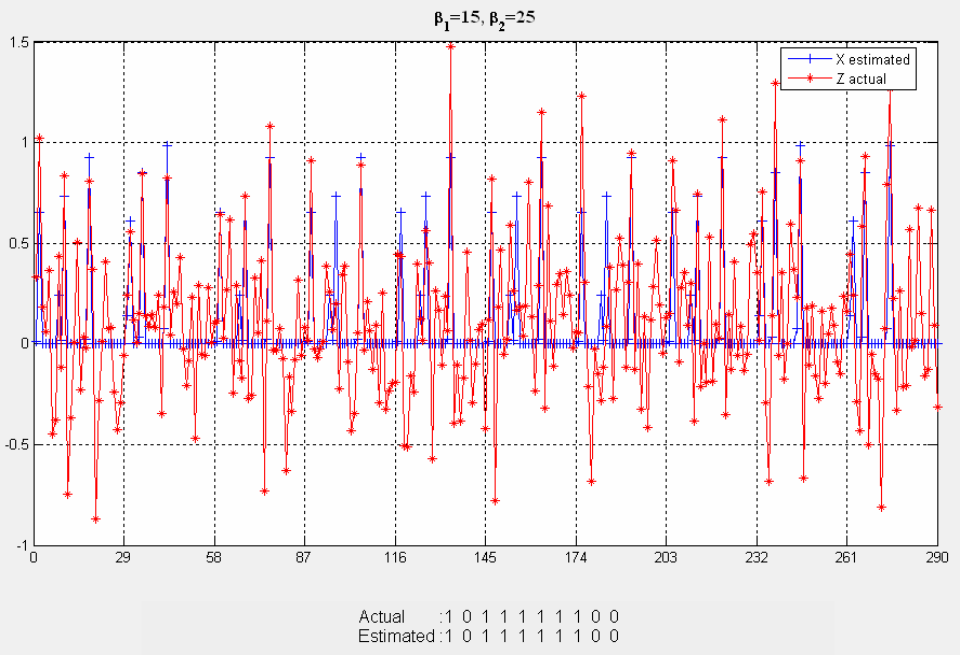

Figure 70 The NCA Map with parameter β_1=15, β_2=25

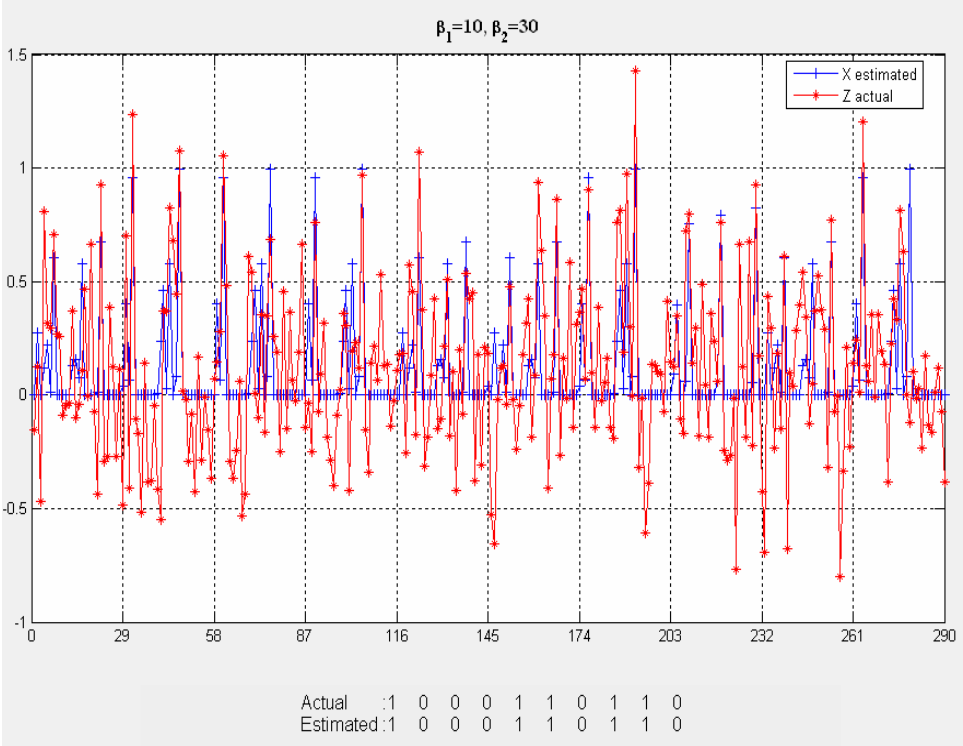

Figure 71 The NCA Map with parameter β_1=10, β_2=30

5.2.2.2 Simulations for constant parameter β

These simulations are run for different values of the parameter α as variance values of observation noise changing. the parameter β is taken constant. The parameters values used in the simulations are;

variance of interference noise = 0 ,

expected value of interference noise = 0 ,

expected value of observation noise = 0 ,

# of initial states = 10,

number of samples, L = 30 ,

gate size = 1/5000 ,

β = 10 ,

initial state = 0.55 .

In the first column of Table 35, variance values of observation noise are given. Figure 72 and Figure 73 show chaotically modulated signal sequences for first 10 bits of randomly generated 1000 bits. In these figures, observation signal sequences and estimated state values are figured out on the same time.

Table 35 BER performance of the system with parameter α

	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2
	BER	**BER**
σ^2 **(v(k))**=0.001	0	0
σ^2 **(v(k))**=0.01	0	0
σ^2 **(v(k))**=0.05	0.001	0.01
σ^2 **(v(k))**=0.1	**0.030**	**0.053**
σ^2 **(v(k))**=0.5	0.245	0.290

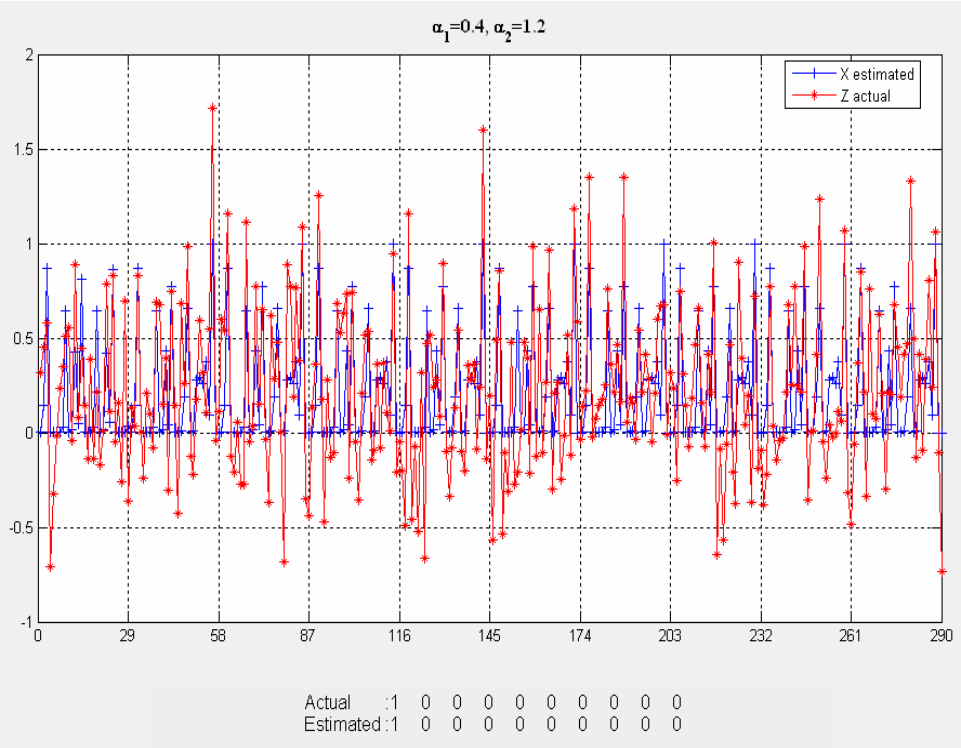

Figure 72 The NCA Map with parameter α_1=0.4, α_2=1.2

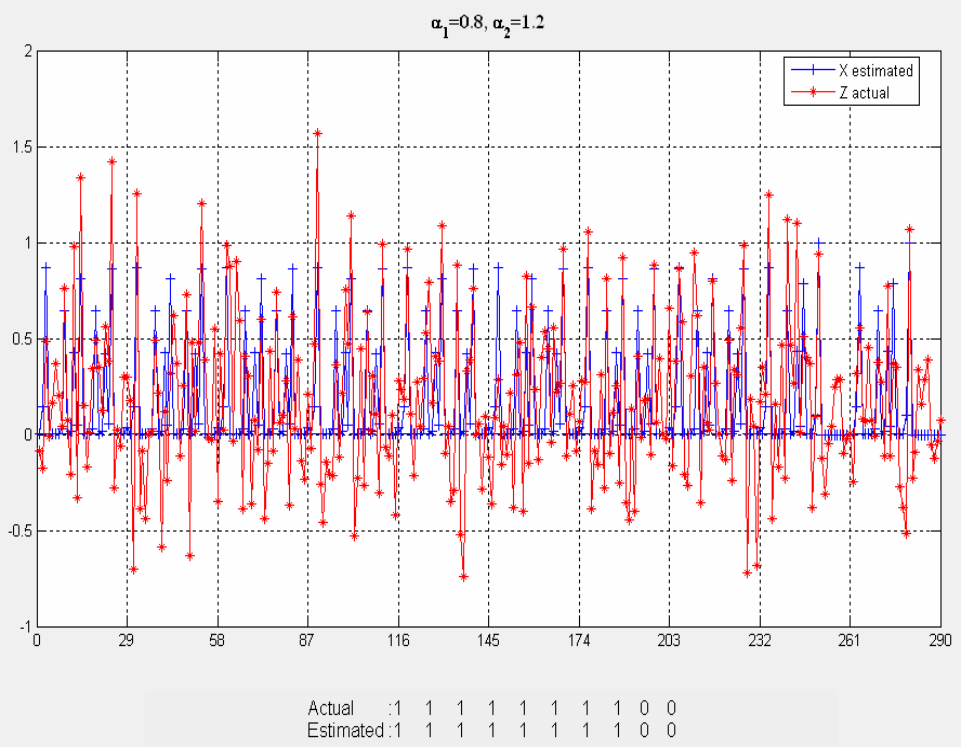

Figure 73 The NCA Map with parameter α_1=0.8, α_2=1.2

5.2.3 Performance of ODSA on the NCA Map in Presence of Interference

In this section, to analyze performance of ODSA on the nonlinear chaotic system, simulation results are given. In the simulations three observation models are used. For each simulation, 1000 binary signals with equal probability are generated randomly. For each bit, initial state is taken constant because of the sensitivity to initial states.

In the first observation model, interference parameter is added to the summation of state vector and observation noise vector.

In the second observation model, observation vector is multiplied by interference parameter whereas the state vector is not affected by interference.

In last observation model, both observation noise vector and state vector are multiplied by interference.

5.2.3.1 Observation model I

In these simulations, motion model is described in (5.12). Observation model is

$$z(k) = x(k) + I(k) + v(k) \qquad (5.15)$$

In order to determine effects of the parameters on BER performance simulations are run both the parameters α and β.

5.2.3.1.1 Simulations for constant parameter α

In this section, effects of the interference noise on the NCA Map are analyzed. On the NCA map the parameter α is taken as constant whereas β is used to refer 0 and 1. Simulations are run for different variances of interference noise. System parameters used in the simulations are;

expected value of observation noise	= 0 ,
quantization # of initial states	= 10,
number of samples, L	= 30 ,
gate size	= 1/5000 ,
α	= 0.9 ,
initial state	= 0.25 .

Following simulations are run for different values of the parameters $[\beta_1, \beta_2]$. Variance values of the interference noise are 0.001, 0.01, 0.1 and 1. Expected value of the interference noise is 0 and its quantization level is 3.

BER performance of the system with parameter β is given Table 36. In the first column of Table 36, variance values of interference are given. In the first row, variance values of observation noise are given.

Table 36 BER performance of the system with different values of σ^2(I(k))

	σ^2 **(vk)=0.001**		σ^2 **(vk)=0.01**		σ^2 **(vk)=0.1**	
	β_1=15 β_2=25	β_1=10 β_2=30	β_1=15 β_2=25	β_1=10 β_2=30	β_1=15 β_2=25	β_1=10 β_2=30
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0	0	0	0.035	0.008
σ^2 **(I(k))=**0.001	0	0	0	0	0.043	0.014
σ^2 **(I(k))=**0.01	0	0	0	0	0.066	0.027
σ^2 **(I(k))=**0.1	0.038	0.019	0.064	0.039	0.153	0.093
σ^2 **(I(k))=**1	0.412	0.396	0.401	0.406	0.412	0.418

Table 36 can be summarized by following figures (Figure 74 and Figure 75).

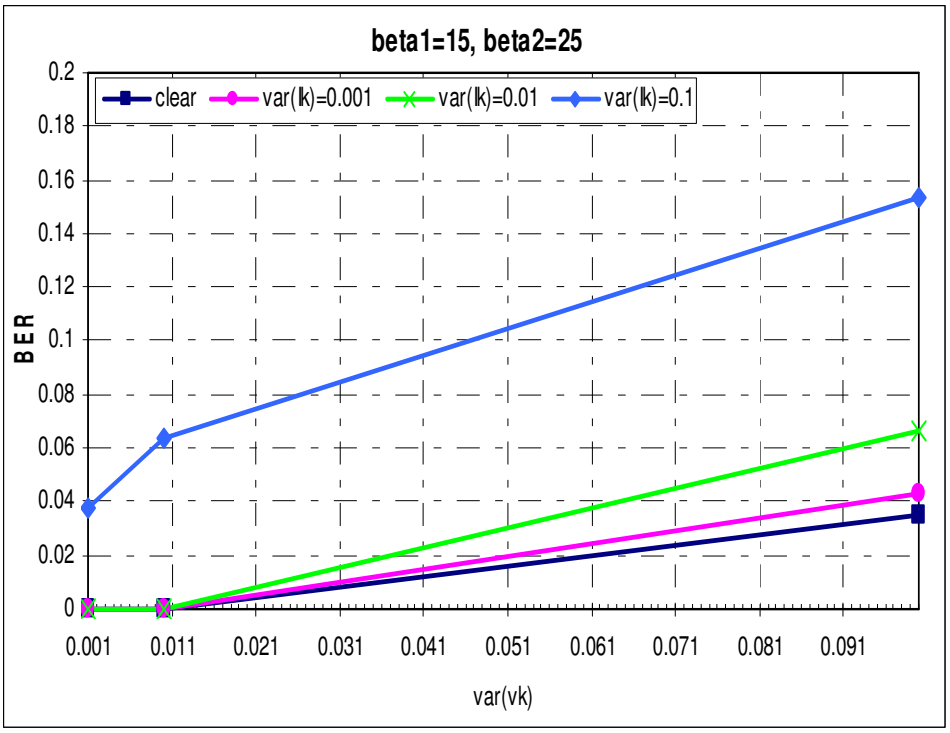

Figure 74 NCA BER performance for interference variance values, β_1 =15, β_2 =25

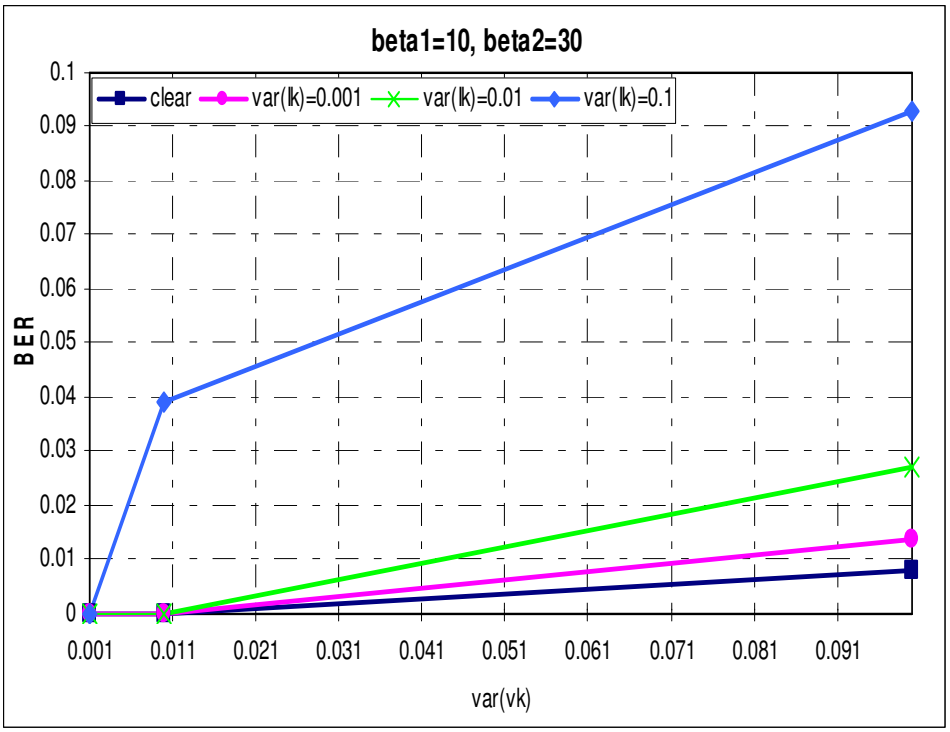

Figure 75 NCA BER performance for interference variance values, $\beta_1=10, \beta_2=30$

5.2.3.1.2 Simulations for constant parameter β

In this section, effects of the interference noise on the NCA Map are analyzed. On the NCA map the parameter β is taken as constant whereas α is used to refer 0 and 1. Simulations are run for different variances of interference noise. System parameters used in the simulations are;

expected value of observation noise = 0 ,

quantization # of initial states = 10,

number of samples, L = 30 ,

gate size = 1/5000 ,

β = 10 ,

initial state = 0.55 .

Following simulations are run for different values of the parameters **[**α_1 α_2**]**. Variance values of the interference noise are 0.001, 0.01, 0.1 and 1. Expected value of the interference noise is 0 and its quantization level is 3.

BER performance of the system with parameter α is given Table 37. In the first column of Table 37, variance values of interference are given. In the first row, variance values of observation noise are given.

Table 37 BER performance of the system with different values of σ^2(I(k))

	σ^2 **(vk)=0.001**		σ^2 **(vk)=0.01**		σ^2 **(vk)=0.1**	
	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0	0	0	0.030	0.053
σ^2 **(I(k))=**0.001	0	0	0	0	0.020	0.068
σ^2 **(I(k))=**0.01	0	0	0	0	0.048	0.067
σ^2 **(I(k))=**0.1	0.038	0.060	0.046	0.100	0.129	0.149
σ^2 **(I(k))=**1	0.405	0.432	0.386	0.427	0.376	0.448

Table 37 can be summarized by following figures (Figure 76 and Figure 77).

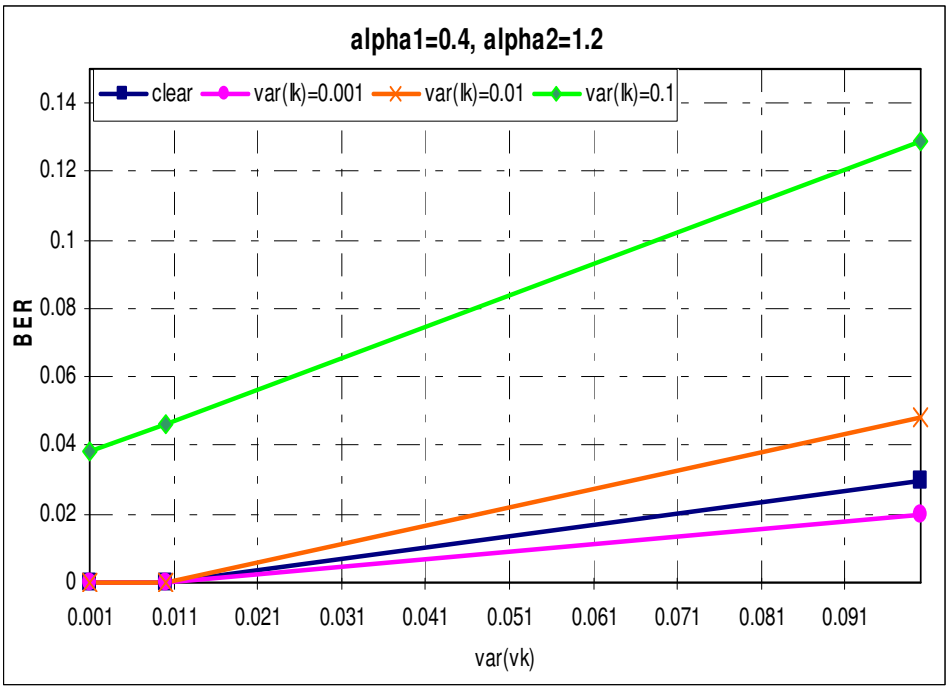

Figure 76 NCA BER performance for interference variance values, α_1=0.4, α_2=1.2

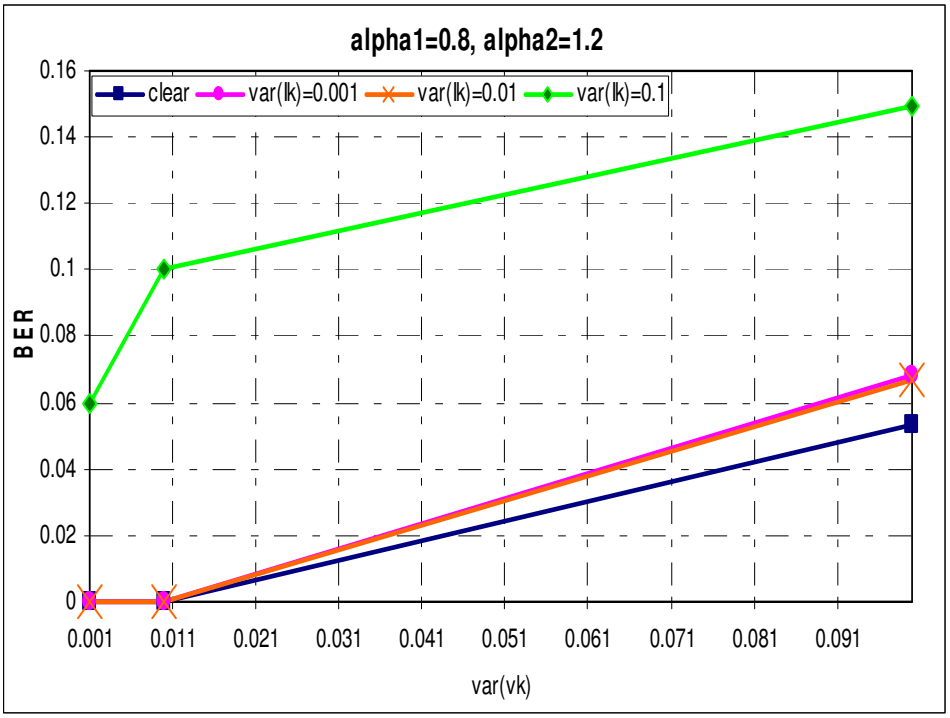

Figure 77 NCA BER performance for interference variance values, α_1=0.8, α_2=1.2

5.2.3.2 Observation model II

In these simulations, motion model is described in (5.12). Observation model is

$$z(k) = x(k) + (1+I^2(k))\ v(k) \qquad (5.16)$$

In order to determine effects of the parameters on BER performance simulations are run both the parameters α and β.

5.2.3.2.1 Simulations for constant parameter α

In this section, effects of the interference noise on the NCA Map are analyzed. On the NCA map the parameter α is taken as constant whereas β is used to refer 0 and 1. Simulations are run for different variances of interference noise. System parameters used in the simulations are;

expected value of observation noise	= 0 ,
quantization # of initial states	= 10,
number of samples, L	= 30 ,
gate size	= 1/5000 ,
α	= 0.9 ,
initial state	= 0.25 .

Following simulations are run for different values of the parameters $[\beta_1, \beta_2]$. Variance values of the interference noise are 0.001, 0.01, 0.1 and 1. Expected value of the interference noise is 0 and its quantization level is 3.

BER performance of the system with parameter β is given Table 38. In the first column of Table 38, variance values of interference are given. In the first row, variance values of observation noise are given.

Table 38 BER performance of the system with different values of σ^2 (I(k))

	σ^2 **(vk)=0.001**		σ^2 **(vk)=0.01**		σ^2 **(vk)=0.1**	
	β_1=15 β_2=25	β_1=10 β_2=30	β_1=15 β_2=25	β_1=10 β_2=30	β_1=15 β_2=25	β_1=10 β_2=30
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0	0	0	0.035	0.008
σ^2 **(I(k))=**0.001	0.001	0	0.002	0	0.07	0.031
σ^2 **(I(k))=**0.01	0	0	0.003	0	0.064	0.041
σ^2 **(I(k))=**0.1	0	0	0.001	0.001	0.097	0.067
σ^2 **(I(k))=**1	0.003	0	0.021	0.004	0.232	0.18

Table 38 can be summarized by following figures (Figure 78 and Figure 79).

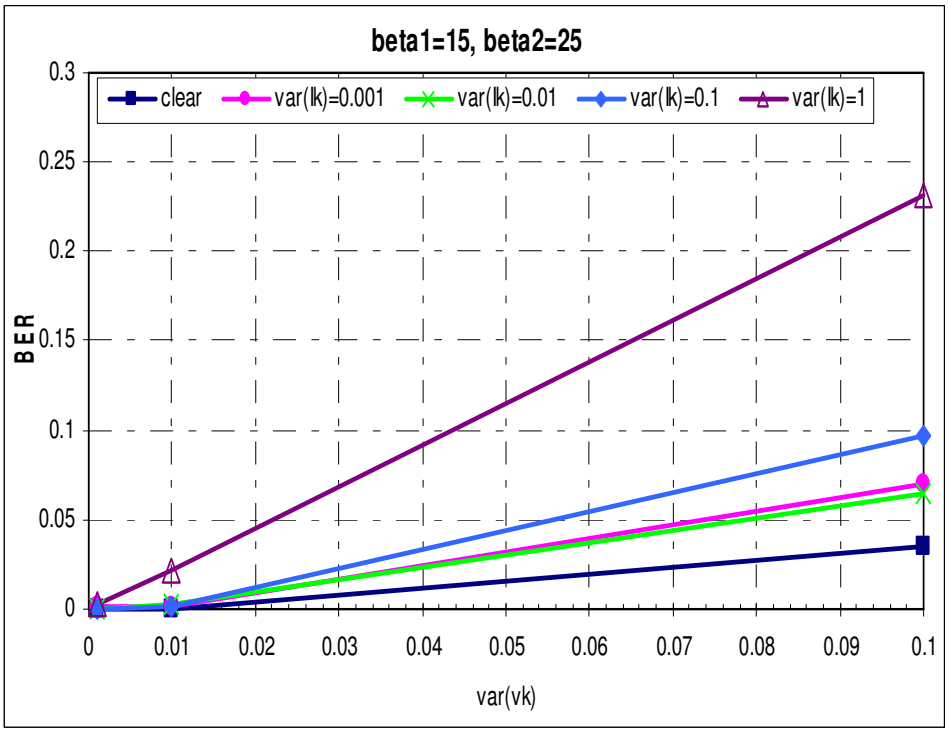

Figure 78 NCA BER performance for interference variance values, β_1 =15, β_2 =25

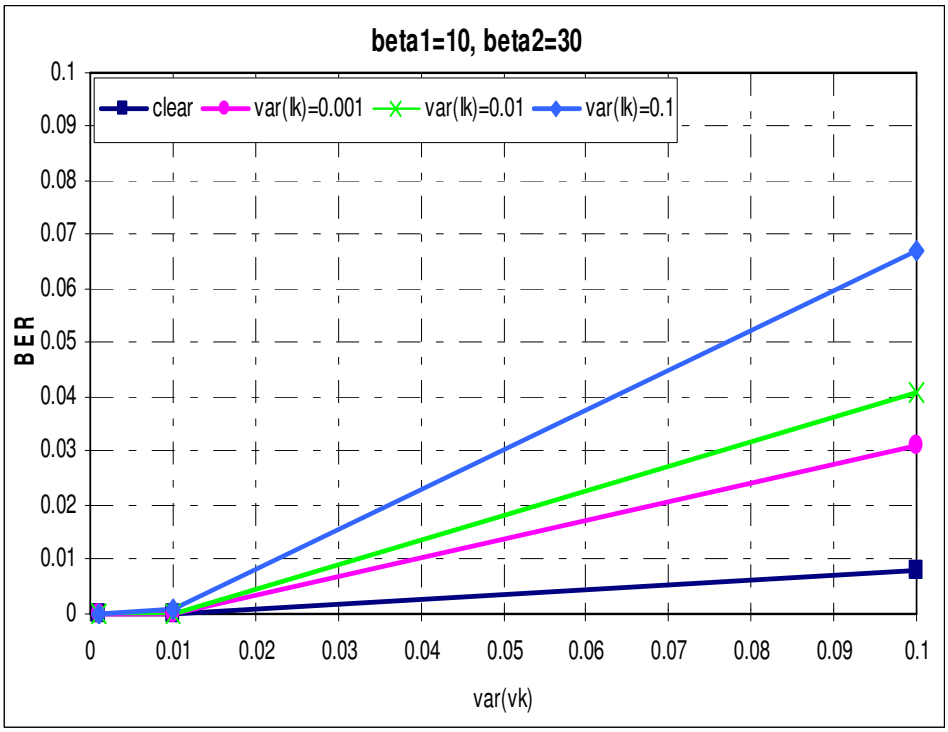

Figure 79 NCA BER performance for interference variance values, $\beta_1=10, \beta_2=30$

5.2.3.2.2 Simulations for constant parameter β

In this section, effects of the interference noise on the NCA Map are analyzed. On the NCA map the parameter β is taken as constant whereas α is used to refer 0 and 1. Simulations are run for different variances of interference noise. System parameters used in the simulations are;

expected value of observation noise = 0 ,

quantization # of initial states = 10,

number of samples, L = 30 ,

gate size = 1/5000 ,

β = 10 ,

initial state = 0.55 .

Following simulations are run for different values of the parameters $[\alpha_1, \alpha_2]$. Variance values of the interference noise are 0.001, 0.01, 0.1 and 1. Expected value of the interference noise is 0 and its quantization level is 3.

BER performance of the system with parameter α is given Table 39. In the first column of Table 39, variance values of interference are given. In the first row, variance values of observation noise are given.

Table 39 BER performance of the system with different values of σ^2(I(k))

	σ^2 **(vk)=0.001**		σ^2 **(vk)=0.01**		σ^2 **(vk)=0.1**	
	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0	0	0	0.030	0.053
σ^2 **(I(k))=**0.001	0	0	0.004	0.006	0.06	0.082
σ^2 **(I(k))=**0.01	0	0	0	0.002	0.066	0.105
σ^2 **(I(k))=**0.1	0	0.001	0.002	0.007	0.075	0.139
σ^2 **(I(k))=**1	0	0.007	0.022	0.031	0.206	0.258

Table 39 can be summarized by following figures (Figure 80 and Figure 81).

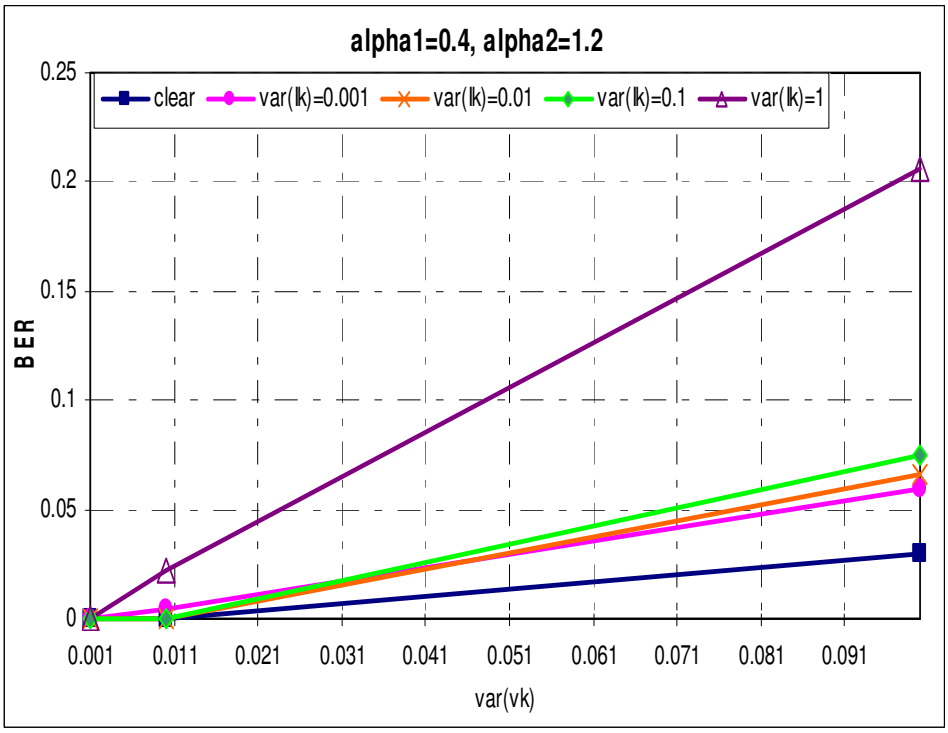

Figure 80 NCA BER performance for interference variance values, α_1=0.4, α_2=1.2

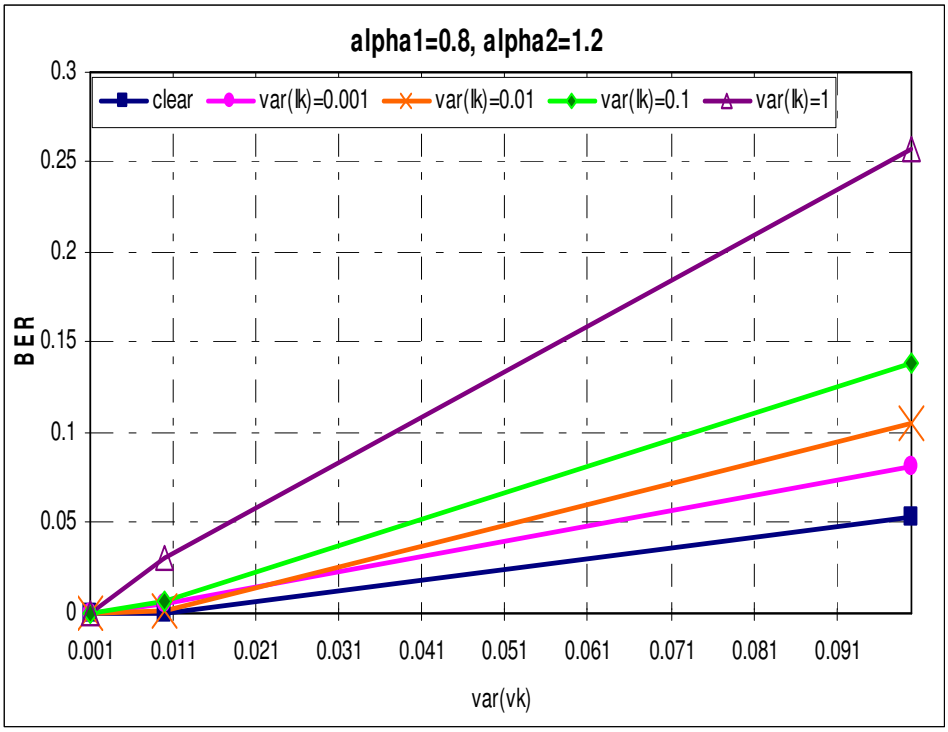

Figure 81 NCA BER performance for interference variance values, α_1=0.8, α_2=1.2

5.2.3.3 Observation model III

In these simulations, motion model is described in (5.12). Observation model is

$$z(k) = x(k)\, I(k) + (1+I^2(k))\, v(k) \qquad (5.17)$$

In order to determine effects of the parameters on BER performance simulations are run both the parameters α and β.

5.2.3.3.1 Simulations for constant parameter α

In this section, effects of the interference noise on the NCA Map are analyzed. On the NCA map the parameter α is taken as constant whereas β is used to refer 0 and 1. Simulations are run for different variances of interference noise. System parameters used in the simulations are;

expected value of observation noise	= 0 ,
quantization # of initial states	= 10,
number of samples, L	= 30 ,
gate size	= 1/5000 ,
α	= 0.9 ,
initial state	= 0.25 .

Following simulations are run for different values of the parameters $[\beta_1, \beta_2]$. Variance values of the interference noise are 0.001, 0.01, 0.1 and 1. Expected value of the interference noise is 0.3 and its quantization level is 3.

BER performance of the system with parameter β is given Table 40. In the first column of Table 40, variance values of interference are given. In the first row, variance values of observation noise are given.

Table 40 BER performance of the system with different values of σ^2(I(k))

	σ^2 **(vk)=0.001**		σ^2 **(vk)=0.01**		σ^2 **(vk)=0.1**	
	β_1=15 β_2=25	β_1=10 β_2=30	β_1=15 β_2=25	β_1=10 β_2=30	β_1=15 β_2=25	β_1=10 β_2=30
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0	0	0	0.035	0.008
σ^2 **(I(k))=**0.001	0.004	0	0.139	0.066	0.416	0.402
σ^2 **(I(k))=**0.01	0.039	0.012	0.145	0.103	0.412	0.407
σ^2 **(I(k))=**0.1	0.246	0.239	0.301	0.297	0.432	0.408
σ^2 **(I(k))=**1	0.313	0.371	0.376	0.371	0.453	0.458

Table 40 can be summarized by following figures (Figure 82 and Figure 83).

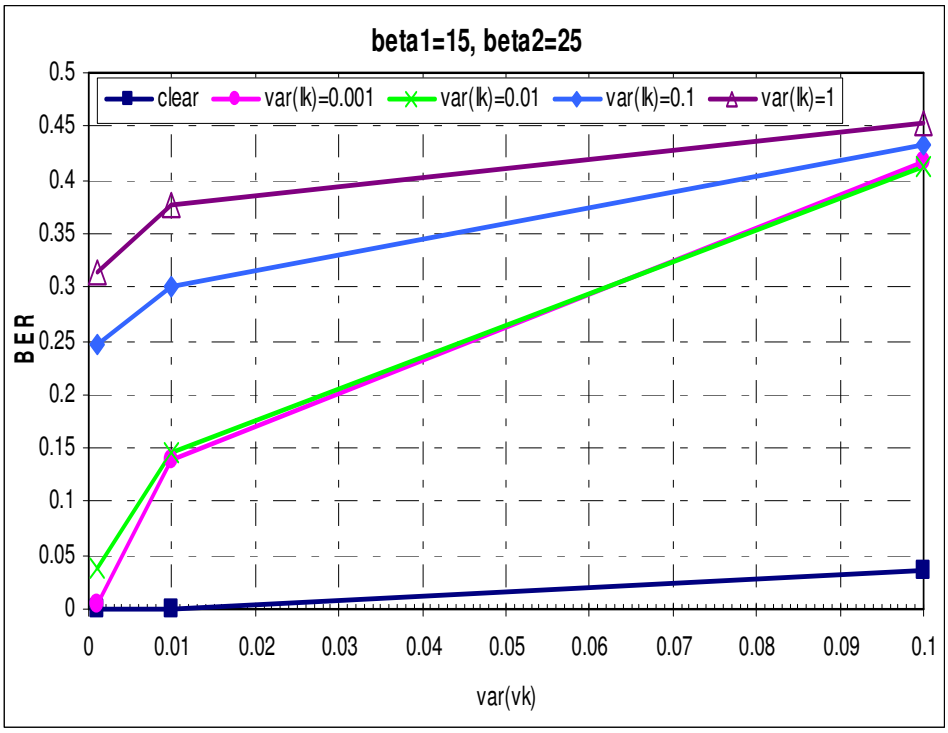

Figure 82 NCA BER performance for interference variance values, β_1 =15, β_2 =25

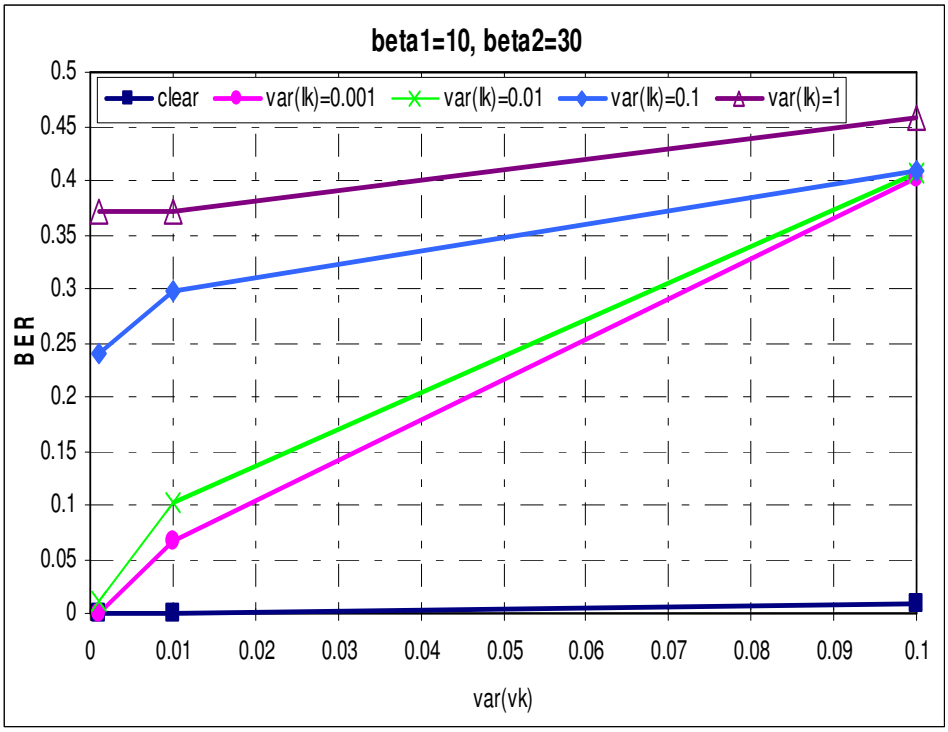

Figure 83 NCA BER performance for interference variance values, β_1 =10, β_2 =30

5.2.3.3.2 Simulations for constant parameter β

In this section, effects of the interference noise on the NCA Map are analyzed. On the NCA map the parameter β is taken as constant whereas α is used to refer 0 and 1. Simulations are run for different variances of interference noise. System parameters used in the simulations are;

expected value of observation noise = 0 ,

quantization # of initial states = 10,

number of samples, L = 30 ,

gate size = 1/5000 ,

β = 10 ,

initial state = 0.55 .

Following simulations are run for different values of the parameters $[\alpha_1, \alpha_2]$. Variance values of the interference noise are 0.001, 0.01, 0.1 and 1. Expected value of the interference noise is 0.3 and its quantization level is 3.

BER performance of the system with parameter α is given Table 41. In the first column of Table 41, variance values of interference are given. In the first row, variance values of observation noise are given.

Table 41 BER performance of the system with different values of σ^2(I(k))

	σ^2 **(vk)=0.001**		σ^2 **(vk)=0.01**		σ^2 **(vk)=0.1**	
	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2	α_1=0.4, α_2=1.2	α_1=0.8, α_2=1.2
	BER	**BER**	**BER**	**BER**	**BER**	**BER**
no interference	0	0	0	0	0.030	0.053
σ^2 **(I(k))=**0.001	0.002	0.011	0.111	0.121	0.434	0.422
σ^2 **(I(k))=**0.01	0.026	0.032	0.135	0.184	0.415	0.428
σ^2 **(I(k))=**0.1	0.203	0.203	0.297	0.287	0.431	0.468
σ^2 **(I(k))=**1	0.314	0.299	0.360	0.359	0.481	0.482

Table 41 can be summarized by following figures.

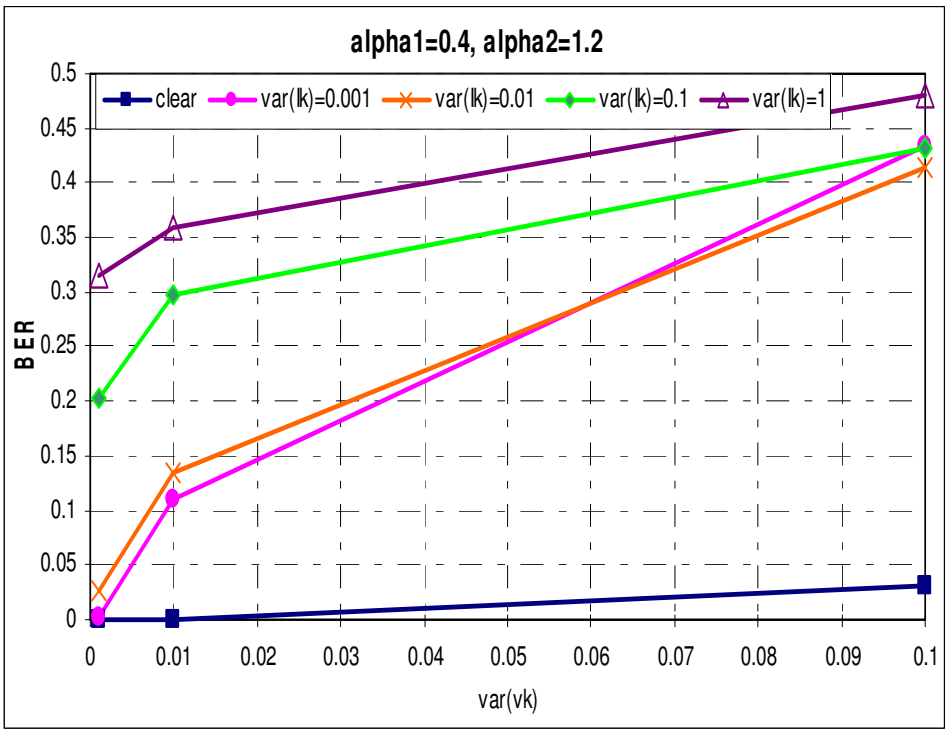

Figure 84 NCA BER performance for interference variance values, α_1=0.4, α_2=1.2

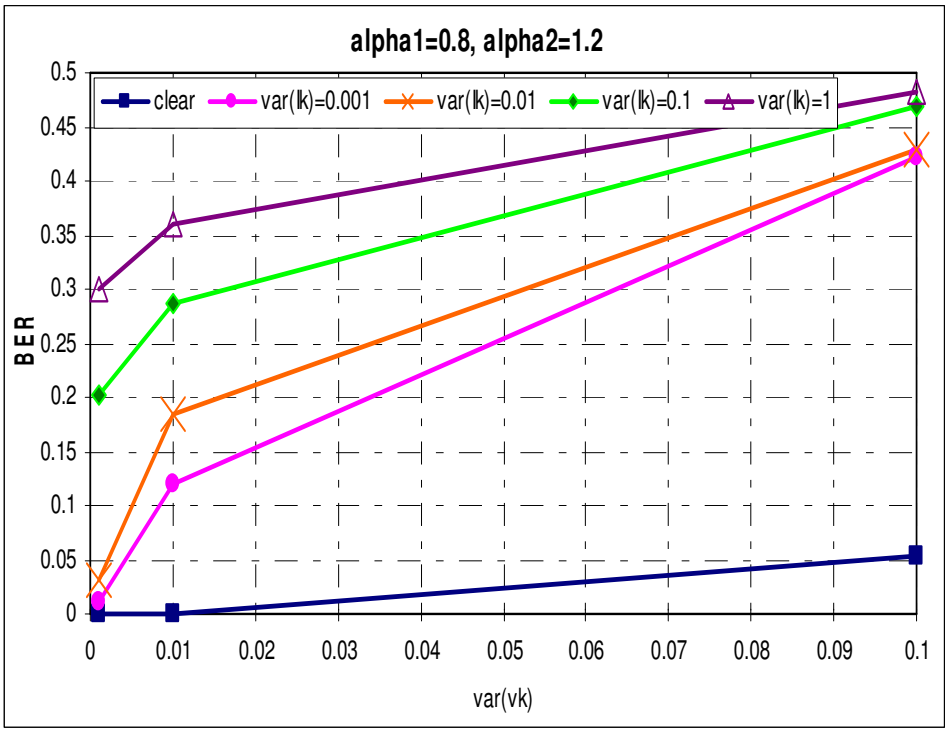

Figure 85 NCA BER performance for interference variance values, α_1=0.8, α_2=1.2

CHAPTER 6

APPLICATION OF THE EXTENDED KALMAN FILTER TO CHAOTIC SYTEMS

The Kalman filter is a mathematical power tool that is playing an increasingly important role in most of engineering areas. The Kalman filter is the best possible (optimal) estimator for a large class of problems and a very effective and useful estimator for an even larger class.

A Kalman filter that linearizes about the current mean and covariance is referred to as an extended Kalman filter or EKF [16].

6.1. The Process to be Estimated [16]

This section describes the filter where the measurements occur and the state is estimated at discrete points in time.

The Extended Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process governed by the non-linear stochastic difference equation.

$$x_{k+1} = f(x_k, u_{k+1}, w_k) \tag{6.1}$$

with a measurement $z(k) \in R^m$ that is

$$z_k = h(x_k, v_k). \tag{6.2}$$

The random variables $w(k)$ and $v(k)$ represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions

$$p(w) \sim N(0, Q) \tag{6.3}$$

$$p(v) \sim N(0, R). \tag{6.4}$$

In practice, *the process noise covariance Q* and *measurement noise covariance R* matrices might change with each time step or measurement, however here we assume they are constant.

The non-linear function *f* in the difference equation (6.1) relates the state at the current time step *k* to the next time step *k+1*. It includes as parameters any driving function u_{k+1} and the zero-mean process noise w_k. The non-linear function *h* in the measurement equation (4.2) relates the state x_k to the measurement z_k .

Approximated the state and measurement vectors are

$$\tilde{x}_{k+1} = f\left(\hat{x}_k, u_{k+1}, 0\right) \tag{6.5}$$

and

$$\tilde{z}_k = h\left(\tilde{x}_k, 0\right). \tag{6.6}$$

where $\hat{x}_k$ is some a posteriori estimate of the state (from a previous time steps *k-1*). It is important to note that a fundamental flaw of the EKF is that the distributions of various random variables are no longer normal after undergoing their respective nonlinear transformations.

To estimate a process with nonlinear difference and measurement relationships, new governing equations that linearize an estimate about Eq.(6.5) and (6.6) are written,

$$x_k \approx \tilde{x}_k + A\left(x_{k-1} - \hat{x}_{k-1}\right) + Ww_{k-1}, \tag{6.7}$$

$$z_k \approx \tilde{z}_k + H(x_k - \tilde{x}_k) + Vv_k. \tag{6.8}$$

where

- x_k and z_k are the actual state and measurement vectors,

- $\tilde{x}_k$ and $\tilde{z}_k$ are the approximate state and measurement vectors form equation (6.5) and (6.6),

- $\hat{x}_k$ is an a posteriori estimate of the state at step k,

- The random variables w_k and v_k represent the process and measurement noise.

- A is the Jacobian matrix of partial derivatives of *f* with respect to x¸ that is

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}\left(\hat{x}_{k-1}, u_k, 0\right) , \tag{6.9}$$

- W is the Jacobian matrix of partial derivatives of *f* with respect to *w,*

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}\left(\hat{x}_{k-1}, u_k, 0\right) , \tag{6.10}$$

- H is the Jacobian matrix of partial derivatives of *h* with respect to *x,*

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}\left(\tilde{x}_k, 0\right) . \tag{6.11}$$

- V is the Jacobian matrix of partial derivatives of *h* with respect to *v,*

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}\left(\tilde{x}_k, 0\right) . \tag{6.12}$$

Note that for simplicity in the notation, the time step subscript *k* is not used with the Jacobians *A, W, H,* and *V*, even though they are in fact different at each time step.

The prediction error is ,

$$\tilde{e}_{x_k} \equiv x_k - \tilde{x}_k , \tag{6.13}$$

and the measurement residual is,

$$\tilde{e}_{z_k} \equiv z_k - \tilde{z}_k . \tag{6.14}$$

Using Eq. (6.13) and (6.14), the governing equations for an error process can be written as

$$\tilde{e}_{xk} \approx A(x_{k-1} - \hat{x}_{k-1}) + \varepsilon_k \ , \tag{6.15}$$

$$\tilde{e}_{zk} \approx H\tilde{e}_{xk} + \eta_k \ , \tag{6.16}$$

where ε_k and η_k represent new independent random variables having zero means and covariance matrices WQW^T and $\mathrm{V}RV^T$, with the process noise covariance matrice Q and measurement noise covariance matrice R .

The random variables of Eq. (6.15) and (6.16) have approximately the following probability distributions

$$p(\tilde{e}_{xk}) \sim N(0, \mathrm{E}[\tilde{e}_{xk}\tilde{e}_{xk}^T]), \tag{6.17}$$

$$p(\varepsilon_k) \sim N(0, WQ_kW^T), \tag{6.18}$$

$$p(\eta_k) \sim N(0, VR_kQ^T). \tag{6.19}$$

The a priori and posteriori estimate error covariances can be defined as

$$P_k^- = \mathrm{E}\{(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T\}$$

$$P_k = \mathrm{E}\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)^T\} \tag{6.20}$$

where $\hat{x}_k^-$ and $\hat{x}_k$ refer for priori and posteriori estimates of x_k.

Given these approximations and letting the predicted value of $\hat{e}_k$ be zero, the Kalman filter equation used to estimate $\hat{e}_k$ is

$$\hat{e}_k = K_k\tilde{e}_{zk} \ . \tag{6.21}$$

By substituting equation (6.21) back into equation (6.19) and making use of equation (6.14) it is seen Kalman filter as:

$$\hat{x}_k = \tilde{x}_k + K_k\tilde{e}_{zk}$$

$$= \tilde{x}_k + K_k\left(z_k - \tilde{z}_k\right) . \tag{6.22}$$

The complete set of EKF equations are follows:

EKF time update equations:

$$\hat{x}_k^- = f\left(\hat{x}_{k-1}, u_k, 0\right), \tag{6.23}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \ . \tag{6.24}$$

EKF measurement update equations:

$$K_k = P_k^- H_k^T \left(H_k P_k^- H_k^T + V_k R_k V_k^T\right)^{-1}, \tag{6.25}$$

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h\left(\hat{x}_k^-, 0\right)\right) , \tag{6.26}$$

$$P_k = \left(I - K_k H_k\right) P_k^- \ . \tag{6.27}$$

Figure 86 shows the complete operation of the EKF.

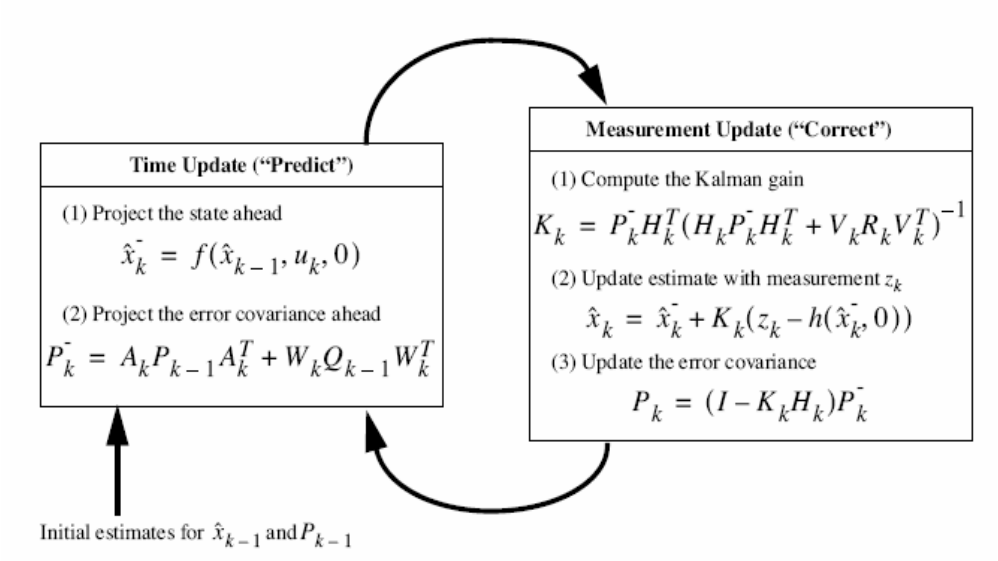

Figure 86 A complete picture of the operation of the EKF

6.2. Application of EKF on One-Dimensional Chaotic Systems

A brief introduction about three types of one-dimensional chaotic systems mostly used in secure communication is given in section 4.2. In this section, the application of the EKF on one-dimensional chaotic tent maps is given. To analyze BER performance of the EKF, the symmetric tent map is chosen as one-dimensional tent map.

Remembering the symmetric tent map, motion and observation models in clear environment can be given as;

Motion model : $\chi(k+1) = a - 1 - a|\chi(k)|$,

Observation model : $z(k) = x(k) + v(k)$ (6.28)

To use the EKF, Jacobians *A, W, H,* and *V* are defined below

- A, the Jacobian matrix of partial derivatives of *f* with respect to *x*

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}\left(\hat{x}_{k-1}, u_k, 0\right) = -a\left|\hat{x}_{k-1}\right|, \tag{6.29}$$

- Because there is not any process noise w_k in the motion model W, the Jacobian matrix of partial derivatives of *f* with respect to *w* is zero,

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}\left(\hat{x}_{k-1}, u_k, 0\right) = 0 \ , \tag{6.30}$$

- H, the Jacobian matrix of partial derivatives of *h* with respect to *v,* is constant for time k=1,2, …, L

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}\left(\tilde{x}_k, 0\right) = 1 \ , \tag{6.31}$$

- V, the Jacobian matrix of partial derivatives of *h* with respect to *v,* is constant for time k=1,2, …, L

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}\left(\tilde{x}_k, 0\right) = 1 \ . \tag{6.32}$$

The EKF time update equations are

$$\hat{x}_k^- = f\left(\hat{x}_{k-1}, u_k, 0\right), \tag{6.33}$$

$$P_k^- = A_k P_{k-1} A_k^T \ . \tag{6.34}$$

The measurement equations are

$$K_k = P_k^-\left(P_k^- + R_k\right)^{-1}, \tag{6.35}$$

$$\hat{x}_k = \hat{x}_k^- + K_k\left(z_k - h\left(\hat{x}_k^-, 0\right)\right) \ , \tag{6.36}$$

$$P_k = \left(I - K_k H_k\right)P_k^- \ . \tag{6.37}$$

The block diagram of the communication system is similar to Figure 48. However in the receiver side there are two EKF parts instead of ODSA's. Like Figure 48, binary messages are modulated by switching the modulator between SYS1 and SY2 when the signal is 0 and 1 respectively. X1 and X2 are two finite length chaotic sequences generated by the same type of system with two different parameters. The comparator is used to compare the estimation errors from the two EKFs to decide if the transmitted signal is 0 or 1 [10].

For each simulation, 1000 binary signals with equal probability, 0 and 1, are generated randomly. For each bit, initial state is generated randomly in the range of [0 , 1]. Since initial estimation error covariance P_0 is normally not known, the filter is initiated with P_0 equal to 10^{10}. This has the effects of treating the initial errors as very large and the filter will ignore the few initial estimates.

In order to determine effects of the observation noise on the performance of the system, simulations are run for different variance values of observation noise. The expected value of the observation noise is taken as 0.

System parameters are;

expected value of observation noise = 0 ,

number of samples, L = 50,

initial estimation error covariance $P_0 = 10^{10}$.

Simulation results using EKF and ODSA are given in Table 42, Table 43, Table 44, and Table 45.

Table 42 BER performances of EKF where σ_{vk}^2 =0.01 in clear environment

	a_1=1.1, a_2=1.8	a_1=1.2, a_2=1.8	a_1=1.3, a_2=1.8	a_1=1.3, a_2=1.7	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.7	a_1=1.4, a_2=1.6	a_1=1.5, a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**
ODSA	0	0.003	0.002	0.002	0.005	0.016	0.069	0.310
EKF	0.003	0	0	0.001	0	0.011	0.070	0.299

Table 43 BER performances of EKF where σ_{vk}^2 =0.05 in clear environment

	a_1=1.1, a_2=1.8	a_1=1.2, a_2=1.8	a_1=1.3, a_2=1.8	a_1=1.3, a_2=1.7	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.7	a_1=1.4, a_2=1.6	a_1=1.5, a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**
ODSA	0.013	0.014	0.032	0.068	0.056	0.082	0.109	0.365
EKF	0.319	0.388	0.427	0.443	0.423	0.467	0.497	0.486

Table 44 BER performances of EKF where σ_{vk}^2 =0.1 in clear environment

	a_1=1.1, a_2=1.8	a_1=1.2, a_2=1.8	a_1=1.3, a_2=1.8	a_1=1.3, a_2=1.7	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.7	a_1=1.4, a_2=1.6	a_1=1.5, a_2=1.6
	BER	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**
ODSA	0.102	0.141	0.120	0.194	0.073	0.207	0.223	0.429
EKF	0.474	0.501	0.483	0.497	0.493	0.491	0.480	0.489

Table 45 BER performances of EKF where $\sigma_{vk}^2 = 1$ in clear environment

	$a_1 = 1.1,$ $a_2 = 1.8$	$a_1 = 1.2,$ $a_2 = 1.8$	$a_1 = 1.3,$ $a_2 = 1.8$	$a_1 = 1.3,$ $a_2 = 1.7$	$a_1 = 1.4,$ $a_2 = 1.8$	$a_1 = 1.4,$ $a_2 = 1.7$	$a_1 = 1.4,$ $a_2 = 1.6$	$a_1 = 1.5,$ $a_2 = 1.6$
	BER	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**	**BER**
ODSA	0.417	0.463	0.476	0.464	0.475	0.506	0.474	0.495
EKF	0.498	0.501	0.500	0.491	0.508	0.498	0.510	0.507

Following figures, which are used to show BER performance of EKF vs ODSA, are given to sum up Table 42, Table 43, Table 44, and Table 45.

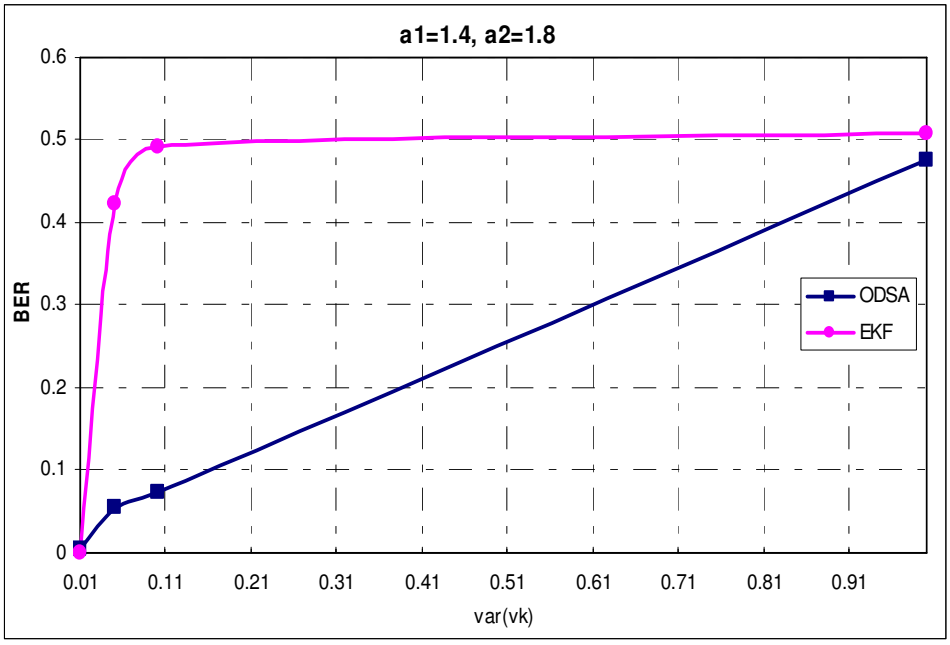

Figure 87 BER performance of EKF and ODSA for a_1=1.4, a_2=1.8

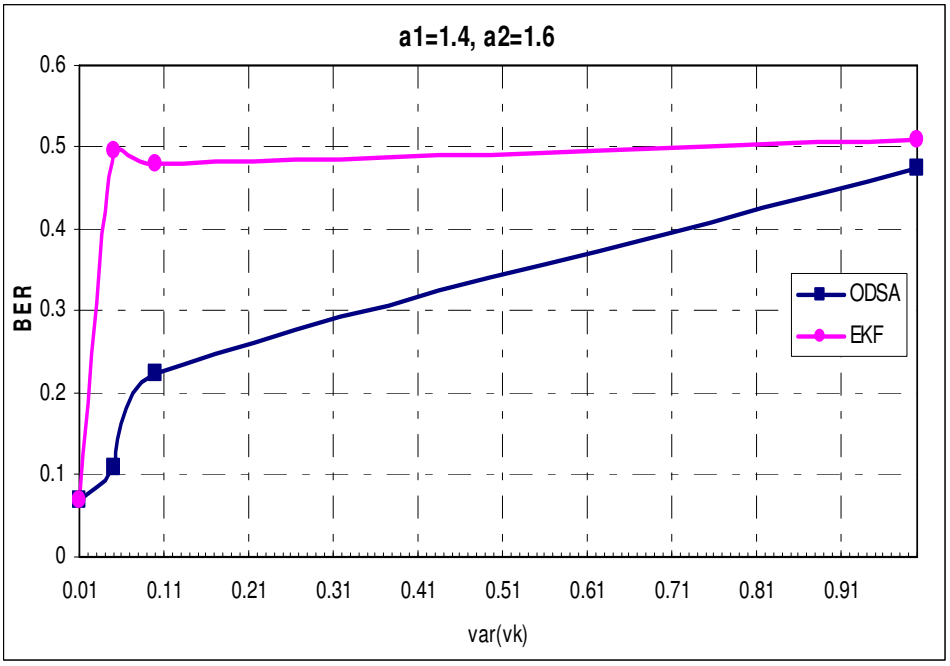

Figure 88 BER performance of EKF and ODSA for a_1=1.4, a_2=1.6

In Table 42, it is seen that, BER performances of the EKF is better than ODSA for case that observation noise variance is equal to 0.01. However, there is a dramatic increase on BER performances of EKF when observation noise variance is equal to or greater than 0.05. To have more information about EKF BER performances depending on observation noise variance, Table 46 is given.

In Table 46, it is observed that a small increase on observation noise variances causes a significant decrease on the performance of the EKF. To improve the performance of the EKF for the observation noise variance is equal to 0.04, number of samples, L is increased as 150. Nevertheless, an improvement on the BER performance could not be obtained.

Table 46 BER performances of EKF for different observation variances in clear environment

	a_1=1.4, a_2=1.8	a_1=1.4, a_2=1.6
	BER	**BER**
σ_{vk}^2=0.01	0	0.070
σ_{vk}^2=0.02	0.093	0.302
σ_{vk}^2=0.03	0.196	0.369
σ_{vk}^2=0.04 (**L**= 50)	0.273	0.393
σ_{vk}^2=0.04 (**L**=150)	0.305	0.491
σ_{vk}^2=0.05	0.423	0.497

To sum up the tables above, BER performances of the EKF is better than ODSA in case small observation noise variances. An increment on observation noise variances influences BER performances of the EKF too much whereas ODSA struggles with observation noise variances.

6.3. Application of EKF on the NCA Map

A brief introduction about the NCA map is described in 4.3. In this section, the application of the EKF on the NCA map is given.

Remembering the NCA map, motion and observation models in clear environment can be given as;

Motion model :

$$\chi_{n+1} = (1-\beta^{-4}) \cdot ctg\left(\frac{\alpha}{1+\beta}\right) \cdot \left(1+\frac{1}{\beta}\right)^{\beta} \cdot tg(\alpha\chi_n) \cdot (1-\chi_n)^{\beta}$$

Observation model : z(k) = x (k) + v (k) (6.38)

where $x_n \in (0,1)$, $\alpha \in (0,1.4]$, $\beta \in [5,43]$ or $x_n \in (0,1)$, $\alpha \in (1.4,1.5]$, $\beta \in [9,38]$ or $x_n \in (0,1)$, $\alpha \in (1.5,1.57]$, $\beta \in [3,15]$.

To use the EKF, Jacobians *A, W, H,* and *V* are defined below

- A, the Jacobian matrix of partial derivatives of *f* with respect to *x*

$$A_{[i,j]} = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_{k-1}, u_k, 0) = c_{EKF} * d_{EKF}, \quad (6.39)$$

where

$$c_{EKF} = (1-\beta^{-4}) \cdot ctg\left(\frac{\alpha}{1+\beta}\right) \cdot \left(1+\frac{1}{\beta}\right)^{\beta} \quad (6.40)$$

and

$$d_{EKF} = \alpha * (1+tg^2(\alpha\hat{\chi}_{k-1})) * (1-\hat{\chi}_{k-1})^{\beta} - tg(\alpha\hat{\chi}_{k-1}) * \beta * (1-\hat{\chi}_{k-1})^{\beta-1}. \quad (6.41)$$

- Because there is not any process noise w_k in the motion model W, the Jacobian matrix of partial derivatives of *f* with respect to *w* is zero,

$$W_{[i,j]} = \frac{\partial f_{[i]}}{\partial w_{[j]}}(\hat{x}_{k-1}, u_k, 0) = 0, \quad (6.42)$$

- H, the Jacobian matrix of partial derivatives of *h* with respect to *v,* is constant for time k=1,2, …, L

$$H_{[i,j]} = \frac{\partial h_{[i]}}{\partial x_{[j]}}(\tilde{x}_k, 0) = 1, \quad (6.43)$$

- V, the Jacobian matrix of partial derivatives of *h* with respect to *v,* is constant for time k=1,2, …, L

$$V_{[i,j]} = \frac{\partial h_{[i]}}{\partial v_{[j]}}(\tilde{x}_k, 0) = 1 \ . \qquad (6.44)$$

The EKF time update and measurement equations are same as given 6.2.

For each simulation, 1000 binary signals with equal probability, 0 and 1, are generated randomly. For each bit, initial state is generated randomly in the range of [0 , 1]. Since initial estimation error covariance P_0 is normally not known, the filter is initiated with P_0 equal to 10^{10}. This has the effects of treating the initial errors as very large and the filter will ignore the few initial estimates.

In order to determine effects of the observation noise on the performance of the system, simulations are run for different variance values of observation noise. The expected value of the observation noise is taken as 0.

System parameters are;

expected value of observation noise = 0 ,

number of samples, L = 50,

initial estimation error covariance P_0 =10^{10}.

Simulation results using EKF and ODSA are given in Table 47 and Table 48

Table 47 BER performance of the system with parameter β

	β_1=15 , β_2=25		β_1=10 , β_2=30	
	BER of ODSA	**BER of EKF**	**BER of ODSA**	**BER of ODSA**
σ^2 **(v(k))**=0.001	0	0.468	0	0.489
σ^2 **(v(k))**=0.01	0	0.504	0	0.511
σ^2 **(v(k))**=0.05	0.008	0.491	0	0.499
σ^2 **(v(k))**=0.1	0.035	0.488	0.008	0.514

Table 48 BER performance of the system with parameter α

	α_1=0.4, α_2=1.2		α_1=0.8, α_2=1.2	
	BER of ODSA	**BER of EKF**	**BER of ODSA**	**BER of ODSA**
σ^2 **(v(k))**=0.001	0	0.490	0	0.485
σ^2 **(v(k))**=0.01	0	0.513	0	0.480
σ^2 **(v(k))**=0.05	0.001	0.476	0.01	0.499
σ^2 **(v(k))**=0.1	0.030	0.502	0.053	0.511

As observed from Table 47 and Table 48, in the NCA map BER performance of EKF increases too much. It can be concluded that the EKF application is not an efficient way on the NCA map.

6.4. Complexity Analysis of the EKF on One-Dimensional Chaotic Systems

The runtime of the program written for the EKF algorithm is determined by the value of number of samples, L. The complexity of the program is linear because at each time step k, the operations, which are performed, does not change. Let the maximum time consumption at each state be t_s at time k. Then, since the time consumption at each state will be approximately same, the time consumption will be maximum t_sL for each bit. Because there is no state quantization on the EKF, the time consumption of EKF algorithm is too smaller than the time consumption of ODSA. Even the time consumptions of EKF on each state is equal to ODSA's, ODSA needs this time consumption for each quantized state.

In the following figure, runtime of ODSA and EKF is compared. In the simulations, the motion and observation models are given as follows;

The motion and observation models are

Motion model : $\chi(k+1) = a - 1 - a|\chi(k)|$,

Observation model : $z(k) = x(k) + v(k)$.

To get approximately same BER performance with ODSA and EKF, the parameters are used;

<u>for ODSA</u>

- Initial states are uniform in the range of 0 and 1,
- quantization level of initial states, Q(x(0)) =1000,
- a_1 =1.4 , a_2 =1.6,
- variance value of observation noise =0.01,
- mean value of observation noise =0,

- number of bits = 1,

<u>for EKF</u>

- initial estimation error covariance P_0 $=10^{10}$
- a_1 $=1.4$, a_2 $=1.6$,
- variance value of observation noise =0.01,
- mean value of observation noise =0,
- number of bits = 1.

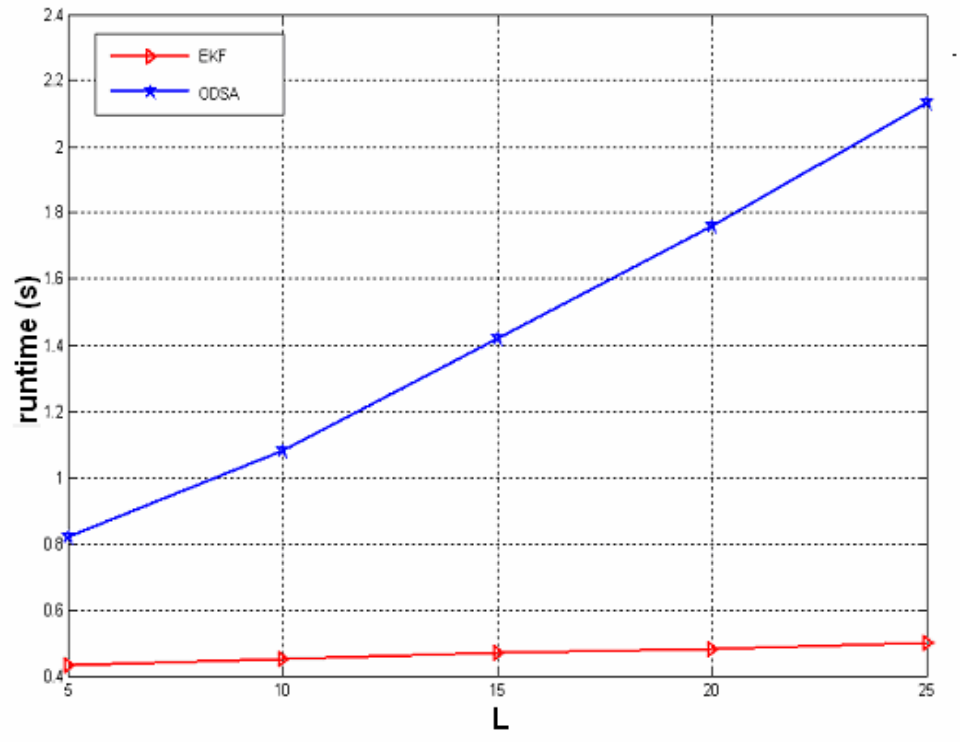

Figure 89 Runtime comparison of ODSA and EKF

CHAPTER 7

CONCLUSION

In this study, optimum decoding based smoothing algorithm [1] and chaotic communication systems under interference [3] are analyzed. Important parameters of this estimation algorithm are explained and some simulations are performed to figure out the performance of ODSA.

ODSA is based on Viterbi decoding algorithm. By reducing the target motion to a finite state model that 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.

To be able to use ODSA on chaotically modulated signals, effect of the parameters of ODSA is analyzed. Therefore, simulations are run and the results can be summarized as below:

- The gate size affects the estimation performance significantly. As the gate size becomes smaller,the state estimation performance increases,
- The quantization number of the initial state vector is effective for only initial times,
- Increasing the disturbance noise variance degrades the performance significantly,
- The quantization number of the disturbance noise vector affects the estimation performance slightly,
- The initial state variance affects only the performance at initial times,

- Increasing the observation noise variance degrades the estimation performance,
- The maximum number of states can be limited without degrading the estimation performance.

The gate size, the quantization numbers and the maximum state number 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 of the simulations and the computational time.

In addition, ODSA can be easily implemented on linear and nonlinear observation model functions that include interference parameters.

When performances of ODSA and EKF algorithms on chaotically modulated systems are compared, EKF algorithm is faster than ODSA. Also, the estimation performance of EKF algorithm is higher than the performance of ODSA when observation noise variance has small values. However, EKF is usable only at clear environment applications and when variance values of observation noise get higher values, EKF algorithm gets useless.

Under interference, ODSA is easily implemented on applications and it gives acceptable BER performance. ODSA is also susceptible to high variance values of the observation noise compared to EKF. Computational complexity is not a problem since today's processors have enough computational power. Therefore, ODSA can be implemented easily when efficiently coded.

As a conclusion, it is seen that ODSA should be the preferred algorithm for applications which are under interference. Also, in clear environments, if there exists comparatively high observation noise, ODSA is again preferable.

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] F. J. Escribano, L. Lopez, M. A. F. Sanjuan, "*Evaluation of channel coding and decoding algorithms using discrete chaotic maps,"* 2006 American Instıtude of Physics, Chaos 16 013103, pp. 1-12, 2006.

[3] H. Ruan, T. Zhai, E. E. Yaz, "*A chaotic Secure communication scheme with extended kalman filter based parameter estimation"*, Proceeding of IEEE International Conference on Control Applications, Istanbul, Turkey pp. 404-408, 2003.

[4] H. Ruan, T. Zhai, E. E. Yaz, Y.I. Yaz, "*A generalization of tent map and its use in EKF based chaotic parameter modulation/demodulation"*, 43rd IEEE Conference on Decision and Control, Atlantis, Bahamas pp. 2071-2075, December 2004.

[5] S. N. Rasband, "*chaotic dynamics of Nonlinear Systems",* a Willey-Interscience Publication, 1989.

[6] J. Banks, V. Dragan, A. Jones, "*Chaos: A Mathematical introduction*", Australian Mathematical Society Lecture Series 18, Cambridge University Press, pp 157-164, 2003

[7] G. L. Baker, J. P. Gollup, "*Chaotic Dynamics: an Introduction*", Cambridge University Press, pp 84-86, 1996

[8] N. H. Gregersen, U. Gorman, H. Meisinger "*A Critical Evaluation of the Use of Chaos in Theology*", Studies in Science & Theology 8. Yearbook of the Europan Society for the Study of Science and Theology 2001-2002. Aarhus, Denmark: University of Aarhus, pp. 277-294, 2002

[9] M. Hasler and Y. Maistrenko, "*An Introduction to the Synchronization of Chaotic Systems :Coupled Skew Tent Maps"*, IEEE Trans. Circuits and Sys., vol44, pp.856-866, Oct.1997

[10] H. Ruan, T. Zhai, E. E. Yaz, " *A Demodulation Scheme Based On State Estimation For Chaotic Digital Communication"*, Proceedings of American Control Conference Denver, Colorado, pp. 1614-1618, June 4-6, 2003.

[11] H. Gao, Y. Zhang, S. Liang, D. Li, " *A New Chaotic Algorithm For Image Encryption"*, Chaos, Solutions and Fractals vol. 29, pp. 393-399,. August 2005

[12] H. L. Van Trees, *"Detection estimation and modulation,"* Part 1, Wiley, New York, 1968.

[13] A. J. Viterbi and J. K. Omura, *"Principles of Digital Communication and Coding,"* McGraw-Hill, New York, 1979

[14] I. B. Rhodes, *"A tutorial introduction to estimation and filtering,"* IEEE Transactions on Automatic Control, AC-16, Dec. 1971.

[15] H. Ruan, T. Zhai, E. E. Yaz, Y. I. Yaz, " *Performance Evaluation of Extended Kalman Filter Based State Estimation for First Order Nonlinear Dynamic Systems"*, Proceedings of the 42nd IEEE Conference on Decision and Control Hawaii, USA pp. 1386-1391, December 2003.

[16] G. Welch, G. Bishop, "An Introduction to the Kalman Filter", SIGGRAPH 2001 Course 8, ACM Inc, USA, 2001.

APPENDIX

APPROXIMATION OF A CONTINUOUS RANDOM VARIABLE WITH A DISCRETE RANDOM VARIABLE [1]

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 $J(.)$:

$$J(F_{y_0}(.)) = \min_{F_y(.)} J(F_y(.)) \qquad (A.1)$$

$$= \min_{g(.)} J(g(.))$$

Where

$$J(F_y(.)) = \int_{-\infty}^{\infty} \left[F_x(a) - F_y(a)\right]^2 da \qquad (A.2)$$

The aim is to find a step function $g_0(.)$ which minimizes the objective function $J(.)$:

$$\begin{aligned} J(g(.)) = & \int_{-\infty}^{y_1} F_x{}^2(a) da + \int_{y_1}^{y_2} \left[F_x(a) - P_1\right]^2 da + \int_{y_2}^{y_3} \left[F_x(a) - P_2\right]^2 da + \ldots \\ & + \int_{y_{n-1}}^{y_n} \left[F_x(a) - P_{n-1}\right]^2 da + \int_{y_n}^{\infty} \left[F_x(a) - 1\right]^2 da \end{aligned} \qquad (A.3)$$

$$g_0(x) = \begin{cases} 0, & x < y_{1,0}, \\ P_{i,0}, & y_{i,0} \le x < y_{i+1,0}, \quad i = 1,2,\ldots,n-1 \\ 1, & x \ge y_{n,0}, \end{cases} \qquad (A.4)$$

If $g_0(x)$ is a step function which minimizes (A.3), it must satisfy the following set of equations:

$$\begin{aligned} &P_{1,0} = 2F_x(y_{1,0}); \\ &P_{i,0} + P_{i+1,0} = 2F_x(y_{i+1,0}), \qquad i = 1,2,...,n-2; \\ &1 + P_{n,0} = 2F_x(y_n); \\ &P_{i,0}(y_{i+1,0} - y_{i,0}) = \int_{y_{i,0}}^{y_{i+1,0}} F_x(a)da \qquad i = 1,2,...,n-1 \end{aligned} \tag{A.5}$$

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 is written which evaluates the values according to the equations given in (A.4) and (A.5). 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 than 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).

$$y' = \sigma y_{i,0} + \mu, \qquad P'_{i,0} = P_{i,0} \qquad i = 1,2,...,n \tag{A.6}$$

The y and p values of approximated x are given at Table 49.

Table 49 y and p values of discrete random variable 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