# CHAOTIC DIGITAL MODULATION AND DEMODULATION

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

BY

UYGAR ÖZTÜRK

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

DECEMBER 2005

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 Leblebicioğlu | (METU,EE) | |
| Prof. Dr. Kerim Demirbaş | (METU,EE) | |
| Prof. Dr. Aydan Erkmen | (METU,EE) | |
| Prof. Dr. Ömer Morgül | (BİLKENT,EE) | |
| Assoc. Prof. Dr. Bilgehan Güven | (METU,STAT) | |

**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: Uygar Öztürk

Signature :

# ABSTRACT

CHAOTIC DIGITAL MODULATION AND DEMODULATION

Öztürk, Uygar

M.S., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Kerim Demirbaş

December 2005, 53 pages

This thesis considers a communication system with chaotic modulation. Noise-like signals are generated by chaotic systems with different parameters to modulate binary digital signals. Demodulation is performed by both the Extended Kalman Filter (EKF) and Optimum Decoding Based Smoothing Algorithm (ODSA). Simulations are performed using both of these algorithms for different parameters affecting the performance of the communication system. Simulation results of these algorithms are compared.

Keywords: Chaotic modulation, secure communication, Optimum Decoding Based Smoothing Algorithm

# ÖZ

DÜZENSİZ SİSTEMLİ SAYISAL MODULASYON VE DEMODULASYON

Öztürk, Uygar

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Kerim Demirbaş

Aralık 2005, 53 sayfa

Bu tezde, kaotik kodlamanın kullanıldığı bir haberleşme sistemi ele alınmıştır. Sayısal sinyalleri kodlamak için farklı parametrelere sahip kaotik sistemler kullanılarak gürültü benzeri sinyaller üretilmiştir. Kod çözümünde, Optimum kod çözümüne dayalı düzeltme algoritması ve Genişletilmiş Kalman filtresi kullanılmıştır. Simülasyonlar, her iki algoritmayı da kullanarak, haberleşme sisteminin performansını etkileyen parametreler için gerçekleştirilmiş ve simülasyon sonuçları karşılaştırılmıştır.

Anahtar Kelimeler: Kaotik modülasyon, güvenli iletişim, Optimum kod çözümüne dayalı düzeltme algoritması

***To My Family***

# ACKNOWLEDGMENTS

I would like to thank to my advisor Prof. Dr. Kerim Demirbaş for his advice and criticism throughout the research.

The author would also like to thank Mr. Hasan Çitçi for his support for a fast computer which was used during simulations.

# TABLE OF CONTENTS

# LIST OF FIGURES

FIGURE

# CHAPTER 1

# INTRODUCTION

Noise-like signals generated by deterministic chaotic systems are similar to stochastic processes [5]. Therefore, they can be used for a secure communication. In communication systems, which are conventionally in use, binary digital signals are modulated by using sinusoidal waveforms before they are transmitted. In a chaotic communication, digital signals are modulated using chaotic waveforms.

In this thesis a simple chaotic map, which is the skew tent map [9, 10], is used to modulate digital signals. Skew tent map is used with different parameters to modulate binary digital signals. Modulated signals are transmitted through an Additive White Gaussian Noise (AWGN) channel. At the receiver, Extended Kalman Filters (EKF's) [7] and Optimum Decoding Based Smoothing Algorithms (ODSA's) [1] with different parameters are employed to estimate chaotic parameters of the received signal. Decision making is performed by comparing estimation errors of two (for binary data) EKF's or ODSA's.

Chapter 2 reviews Optimum Decoding Based Smoothing Algorithm (ODSA) which uses the Viterbi decoding algorithm for estimation.

Chapter 3 gives a short review of chaos and chaotic systems. Two chaotic models, skew tent map and symmetric tent map [9, 10], are defined. Some chaotic properties of these maps (sensitivity to initial conditions and chaotic parameters) are shown by figures.

Chapter 4 presents some simulation results of state estimations of dynamic models with skew and symmetric tent maps by using Optimum Decoding Based Smoothing

Algorithm (ODSA).

In Chapter 5, chaotic modulation and demodulation are considered. Both the EKF and ODSA are used for demodulation and some simulation results are presented.

# CHAPTER 2

# STATE ESTIMATION AND SMOOTHING ALGORITHM

This chapter reviews the Optimum Decoding Based Smoothing Algorithm (ODSA) [1], which is used for state estimation of nonlinear dynamic systems. This algorithm quantizes states of models and then uses the Viterbi algorithm for state estimation.

## 2.1. Models and Assumptions

In this chapter we deal with the discrete model in (2.1) and (2.2)

$$x(k+1) = f(k, x(k), w(k)) \qquad \text{State Model} \qquad (2.1)$$

$$y(k) = g(k, x(k), v(k)) \qquad \text{Observation Model} \qquad (2.2)$$

where,

- x(0) is an nx1 initial state vector (at time 0)
- x(k) is an nx1 state vector at time k
- w(k) is a px1 disturbance noise vector at time k with zero mean and known statistics
- v(k) is an lx1 observation noise vector at time k with zero mean and known statistics
- y(k) is an rx1 observation vector at time k

If we call $t_0$ as time 0 or initial time, then ‘time k’ can be defined as $t_0 + kT_0$ where $T_0$ is observation interval.

$f(k, x(k), w(k))$ and $g(k, x(k), v(k))$ are linear or nonlinear vectors of appropriate dimensions. 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 goal is to estimate the state sequence {x(0), x(1), …, x(L)} using observation sequence {y(1), y(2), …, y(L)}, where L is a chosen integer.

The estimation algorithm presented in this chapter is applicable to any type of estimation problem.

## 2.2. Quantization of States and Transition Probabilities

In this section, a quantization type for target states and some difficulties in calculating transition probabilities between quantization levels is described. If we consider state x(k) it is a random vector whose range is in space $R^n$ (n-dimensional Euclidean space). Let us divide $R^n$ into non-overlapping subspaces $R_i^n$ and assign a unique value $x_{qi}$ to each subspace $R_i^n$ , where the subscript q is quantization.

Definition 1: A function $x_q(\cdot) \overset{\Delta}{=} Q\{x(\cdot)\}$ is a quantizer for the state $x(\cdot)$ if following 2 holds:

1. $x_q(\cdot) \overset{\Delta}{=} Q\{x(\cdot)\} = x_{qi}$ whenever $x(\cdot) \in R_i^n$
2. $x_{qi}$ is unique for each $R_i^n$

Definition 2: The function $x_q(\cdot)$ is the quantized state vector at time $(\cdot)$, and its possible values are called the quantization levels of the state $x(\cdot)$.

Definition 3: The value $x_{qi}$ is called the quantization level for the gate $R_i^n$ .

Definition 4: The transition probability $\pi_{jm}(k)$ is the probability that the space 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.,

$$\pi_{jm}(k) \overset{\Delta}{=} \Pr ob\left\{x(k+1) \in R_m{}^n \middle| x(k) \in R_j{}^n\right\} \tag{2.3}$$

Since $\pi_{jm}(k)$ is a conditional probability (2.3) can be rewritten as

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

It is not easy to evaluate the transition probability $\pi_{jm}(k)$ analytically. The difficulties are due to the shapes of the gates $R_j{}^n$ and $R_m{}^n$ and the statistics of the disturbance-noise vectors $w(\cdot)$ and the initial state vector $x(0)$. The problem is more difficult if the model is nonlinear.

Therefore in the next section an approximate target motion model called the finite state model that is obtained by approximating the disturbance noise vector $w(k)$ and the initial state vector $x(0)$ by discrete random vectors.

## 2.3. Finite State Model

In this section gates are assumed to be generalized rectangles such that the zero vector is located in the center of a generalized rectangle, $R_0{}^n$. Let the lengths of the sides of a generalized rectangle be $g_{i1}$, $g_{i2}$, … , $g_{in}$. These lengths are called sizes of gate $R_i{}^n$. Moreover, the quantization levels for gates are assumed to be the center points of the gates, namely,

$$x_q(\cdot) \overset{\Delta}{=} Q\{x(\cdot)\} = x_{qi} \text{ if } x_q(\cdot) \in R_i{}^n \tag{2.5}$$

where $x_{qi}$ is the center of the generalized rectangle $R_i{}^n$. Number of quantization points used defines complexity of the algorithm.

Now we can define finite-state model that approximates the target-motion model. For each k, the disturbance noise vector w(k) is approximated by a discrete random vector $w_d(k)$. $w_d(k)$ can take $m_k$ possible values $w_{d1}(k)$, $w_{d2}(k)$, …, $w_{dm_k}(k)$. Corresponding probabilities are: $p_{d1}(k)$, $p_{d2}(k)$, …, $p_{dm_k}(k)$, where, $m_k$ is a positive integer. Also, the initial state vector x(0) is approximated in the same way: It is approximated by discrete random vector $x_d(0)$ which can take $n_0$ possible values $x_{d1}(0)$, $x_{d2}(0)$, …, $x_{dn_0}(0)$. Corresponding probabilities are: $p_{d1}(0)$, $p_{d2}(0)$, …, $p_{dn0}(0)$, where, $n_0$ is a positive integer. $m_k$, $n_0$ must be big enough so w(k), x(0) are satisfactorily approximated by the discrete vectors $w_d(k)$ and $x_d(0)$.

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 $R_j^{\ n}$, namely,

$$\pi_{jl}(k) = \mathrm{Pr}\,ob\{x_q(k+1) = x_{ql} \,|\, x_q(k) = x_{qj}\} \tag{2.6}$$

is determined as follows.

Let us assume that the quantized state vector $x_q(k)$ is equal to the quantization level $x_{qj}$ for gate $R_j^{\ n}$. The transitions from this quantization level to the others are determined by the discrete random vector $w_d(k)$ and the function $Q\{f(k, x_q(k) = x_{qj}, w_d(k))\}$. The discrete random vector $w_d(k)$ can take any value in the set { $w_{d1}(k)$, $w_{d2}(k)$, …, $w_{dm_k}(k)$ } with corresponding probabilities $p_{d1}(k)$, $p_{d2}(k)$, …, $p_{dm_k}(k)$. Thus, the quantized state vector $x_q(k+1)$ can be equal to at most $m_k$ various quantization levels. If the function $f(k, x_q(k) = x_{qj}, w_d(k))$ 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, i.e., $\pi_{ji}(k) = p_{di}(k)$. However, if the function $f(k, x_q(k) = x_{qj}, w_d(k))$ maps $x_{qj}$ into another gate, say $R_l^n$, for more than one possible value, say $w_{d1}(k)$, $w_{d2}(k)$ of $w_d(k)$, the transition probability $\pi_{jl}(k)$ from gate $R_j^n$ to gate $R_l^n$ is the probability that the discrete random vector $w_d(k)$ is equal to either of the possible values $w_{d1}(k)$ or $w_{d2}(k)$, i.e., $\pi_{jl}(k) = \sum_n p_{dn}(k) = p_{d1}(k) + p_{d2}(k)$, where the summation is over all n such that $Q\{f(k, x_q(k) = x_{qj}, w_d(k))\} = x_{ql}$. Since we determined finite state model, we can represent the target motion by a trellis diagram.

## 2.4. Trellis Diagram for the Target Motion

Let us assume that the quantized state vector $x_q(k)$ has $n_k$ possible values, say $x_{q1}(k)$, $x_{q2}(k)$, …, $x_{qn_k}(k)$ where $n_k$ is a positive integer. To represent target motion by a graph following conventions can be used:

1 Each possible value of $x_q(k)$ is represented on kth column by a point with corresponding quantization level so kth column contains possible quantization levels of x(k).

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

The target motion from time zero to time L can be represented as in Figure2.1

which is called trellis diagram for the target motion from time zero to L.

Finally, definition of a path in trellis diagram is any sequence of directed lines where the final vertex of one is the initial vertex of the next.

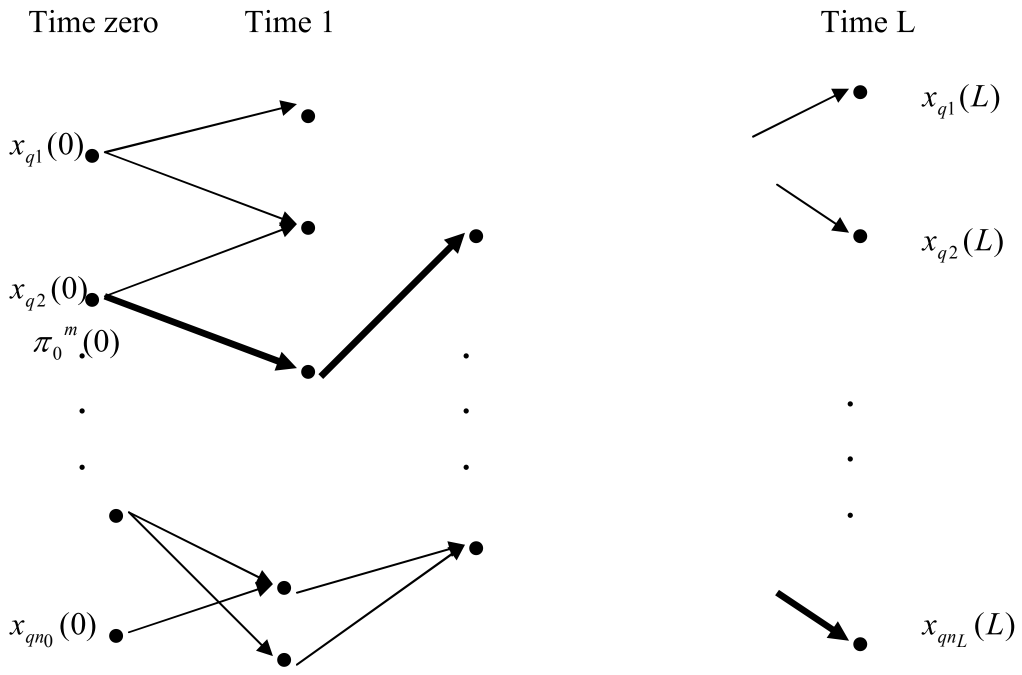

Figure 2.1 The trellis diagram for the target motion.

## 2.5. Approximate Observation Model

Until that point target-motion model has been reduced to a finite state model which uses the quantized state vector $x_q(0)$. In the observation model in (2.2) by replacing state vector $x\ (k)$ with quantized state vector $x_q(k)$ the following approximate observation model is obtained:

$$y(k) = g(k, x_q(k), v(k)) \qquad \text{Observation Model} \qquad (2.7)$$

From now on, when we discuss about observation model, it refers to (2.7)

When we consider trellis diagram in figure2.1, it is assumed that target will be tracked from time zero to time L. Let us now define the following symbols which will be used throughout the analyses:

$n_i$ Number of quantization levels for the gates in which the target may lie at time i.

$\underset{\sim}{x}(i)$ Set of all the quantization levels for the gates in which the target may lie at time i.

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

$H_m$ The mth path through the trellis diagram (indicated as a bold line in Figure2.1)

$x_q^{\ m}(i)$ Quantization level for the gate in which the target lays at time i when it follows path $H_m$

$\pi_0^{\ m}$ Probability that the possible value of the initial state vector $x_d(0)$ from which the mth path starts.

$\pi_i^{\ m}$ Transition probability from the (i-1)th gate for the mth path to the ith gate for the mth path.

$\pi_0^{\ \max}$ Maximum of the probabilities that the quantization levels at time zero occur.

$\pi_i^{\ \max}$ Maximum of the transition probabilities from the quantization levels at time i-1 to the quantization levels at time i.

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

$\pi_i^{\min}$ Minimum of the transition probabilities from the quantization levels at time i-1 to the quantization levels at time i.

$\underset{\sim L}{x}^{m} \overset{\Delta}{=} \{x_q^m(0), x_q^m(1), ..., x_q^m(L)\}$ Sequence of the quantization levels which the mth path passes through

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

Our aim is to define a path, using observation sequence $y^L$, through the trellis diagram which is most likely followed by the target. Since the model includes randomness our approach must be statistical. The suitable criterion for the problem is the minimum error probability criterion, which is a special case of Bayes' criterion. Using this criterion reduces the problem of finding the most likely path followed by the target to a multiple hypothesis-testing problem.

## 2.6. Minimum Error Probability Criterion

Using the minimum error probability criterion and the observation sequence, we would like to decide which hypothesis is true (we would like to find most likely path followed). To accomplish this we develop a decision rule and this rule divides the whole observation space D into subspaces $D_1$, $D_2$, …, $D_m$. If the observations fall into subspace $D_i$, we decide that target followed path $H_i$. $D_i$ is called the decision region for hypothesis $H_i$. So we must choose the decision regions $D_1$, $D_2$, …, $D_m$ in such a way that error probability is minimized.

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

$$R \overset{\Delta}{=} \sum_{j=1}^{M} \sum_{\substack{i=1 \\ i \neq j}}^{M} \left\{ \int_{Y^L \in D_i} p(H_j) p(y^L | H_j) dy^L \right\} \qquad (2.8)$$

where,

$p(H_j)$ is probability that the hypothesis $H_j$ is true. This is called the a priori probability of hypothesis $H_j$.

$p(y^L | H_j)$ is conditional probability of the observation sequence $y^L$ given that hypothesis $H_j$ is true.

In order to find the optimal decision rule, we vary the decision regions $D_1$, $D_2$, …, $D_m$ so that the risk R is minimized. The optimum decision rule [1] is

$$\text{Choose } H_i \text{ if } p(H_i) p(y^L | H_i) > p(H_j) p(y^L | H_j) \text{ for all } j \neq i \qquad (2.9)$$

For a given observation sequence $y^L$, if the inequality in (2.9) becomes an inequality for one or more hypotheses $H_j$, any one of these and $H_i$ can be chosen as the decision.

Note that the decision regions are non-overlapping and the union of all the decision regions covers the whole observation space D.
The optimum decision rule may be interpreted as follows: If the observation sequence $y^L$ falls within the optimum decision region $D_i$, then choose hypothesis $H_i$.

## 2.7. Optimum Decision Rule for the Target Paths

From previous section, the a priori probability of hypothesis $H_i$ can be rewritten as

$$p(H_i) = \prod_{k=0}^{L} \pi_k^i \qquad (2.10)$$

since the disturbance noise vector w(k) is assumed to be independent of w(j) and x(0) for all $j \neq k$ .

Moreover, since sequence $\underset{\sim L}{x}^i$ describes hypothesis $H_i$ completely and the assumption that the observation noise is independent from sample to sample, the function $p(y^L|H_i)$ can be rewritten as

$$p(y^L|H_i) = \prod_{k=1}^{L} p(y(k)|x_q^i(k)) \qquad (2.11)$$

where, $p(y(k)|x_q^i(k))$ is the conditional probability of the observation y(k) given that $x_q(k) = x_q^i(k)$ , i.e., $p(y(k)|x_q^i(k)) = p(y(k)|x_q(k) \underset{=}{\Delta} x_q^i(k))$

Substituting (2.10) and (2.11) into the optimum decision rule of (2.9) we obtain the following [1]

$$\text{Choose } H_i \text{ if } \quad \pi_0^i \prod_{k=1}^{L} \pi_k^i p(y(k)|x_q^i(k)) > \pi_0^j \prod_{k=1}^{L} \pi_k^j p(y(k)|x_q^j(k))$$

for all j≠i (2.12)

Since it is frequently more convenient to use summations than multiplications taking the natural logarithm of both sides of the inequality in (2.12) we get the following [1]:

Choose $H_i$ if

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

for all j≠i (2.13)

Either of the expression in (2.12) and (2.13) is the optimum decision rule for deciding the path most probably followed by the target.

Let us now present some definitions to be used later in the chapter:

- An initial node is a quantization level at time zero. The metric denoted by

$$MN(x_q^m(0)) = \ln\left[ \Pr ob\left\{ x_q(0) = x_{qi}(0) \right\} \right] \tag{2.14}$$

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

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

$$M(x_{qj}(k-1) \to x_{qi}(k)) \underset{=}{\Delta} \ln\left[ \Pr ob\left\{ x_q(k) = x_{qi}(k) | x_q(k-1) = x_{qj}(k-1) \right\} \right] + \ln p(y(k) | x_{qi}(k)) \tag{2.15}$$

- The metric of a path from time zero to time i is the summation of the metric of the initial node from which the path starts and the metrics of the branches of which the path consists. For example, the metric, denoted by $M(x_q^m(i))$, of the portion between the nodes $x_q^m(0)$ and $x_q^m(i)$ of the path $H_m$ as defined in [1] is:

$$M(x_q^m(i)) = \ln \pi_0^m + \sum_{k=1}^{i} \left[ \ln \pi_k^m + \ln p(y(k) | x_q^m(k)) \right] \tag{2.16}$$

- The error probability of a path, say $H_m^{'}$ through a trellis diagram T with M possible paths $H_1$, $H_2$, ..., $H_M$ is the probability of deciding that a path which is different from $H_m$ is the one most probably followed by the target when the target actually followed the path $H_m$. This error probability is denoted by either $P_{E_m}(H_1,...,H_M)$ or $P_{E_m}(T)$ where subscripts E and m are the error and the mth path respectively. The overall error probability for the detection of the path most likely followed by the target can be expressed in terms of the path error probabilities as follows [1]:

$$P_E = \sum_{m=1}^{M} p(H_m) p_{E_M}(H_1, H_2, ..., H_M) \tag{2.17}$$

- The density function of the observation sequence $y^L$ when the target actually followed the path $H_m$ is referred to as the likelihood function for the path $H_m$.

Optimum decision rule is to choose the path with the largest metric through the trellis diagram as the decision. This can be handled using Viterbi Decoding Algorithm, which is optimum decoding algorithm.

## 2.8. Optimum Decoding Based Smoothing Algorithm

*Preliminary step:* Reduce the target motion model to a finite state model and obtain a trellis diagram for the target motion from time zero to time L. Assign each initial node its metric.

*Step 1:* For each node at time 1 use observations y(1) and evaluate the metrics of the branches connecting the initial nodes to the node at time 1. Add these metrics to the metrics of the initial nodes from which the branches start, find the metrics of the paths merging at the node at time 1. Label the path with the largest metric and discard other paths. Finally assign the largest metric to the node at time 1 which is called the metric of the node at time 1.

*Step k:* For each node at time k; use the observation at this time and calculate the metrics of the branches connecting the nodes at time k-1 to the node at time k. Add these metrics to the metrics of the nodes at time k-1 from which the branches start. Find the metrics of the paths merging at the node at time k, label the path with the largest metric, and then discard the other paths. Finally, assign the largest metric to the node at time k.

At the end of step L, stop and choose from among the nodes at time L that with the largest metric. Then decide that the best path for this node is the path fallowed by the target.

# CHAPTER 3

# NONLINEAR DYNAMICS AND CHAOS

In this chapter, definition of chaos and its relation with nonlinear dynamics are given. The Tent Map as a common type of chaotic mapping and its two versions are given. These are symmetric tent map and skew tent map [9, 10]. Finally, in the last part of the chapter, sensitivities to initial conditions and chaotic parameters of the symmetric tent map are shown by figures.

## 3.1. Chaos and Nonlinearity

The word “chaos” implies some observation of a system which varies unpredictably. For a measurement we say “chaotic” if it does not have regularity or order. 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. [2]

Whenever dynamical chaos is found, it is accompanied by nonlinearity. Nonlinearity in a system means that current state of a system depends on early state(s) of the system in a complicated way. Here complicated means not just proportional (a constant).

In this work only discrete chaotic structure will be used. Mathematical requirements of a system to be chaotic are given below:

Let V be an interval. We say that $f : V \to V$ is chaotic on V if the following conditions hold [9,10]

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 [9]. Sensitivity to initial conditions is related to the Lyapunov exponents [9,10]. 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+ε, after n iterations, their divergence may be characterized as [10]

$$\varepsilon(n) \approx \varepsilon e^{\lambda n} \tag{3.1}$$

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

$$x_{n+1} = f(x_n) \tag{3.2}$$

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

$$f^n(x+\varepsilon) - f^n(x) \approx \varepsilon e^{\lambda n} \tag{3.3}$$

or

$$\ln\left(\frac{f^n(x+\varepsilon)-f^n(x)}{\varepsilon}\right)\approx n\lambda \tag{3.4}$$

where $f^n(x)$ is iterated value of initial value x after n steps.

For small ε, this can be rewritten as

$$\lambda=\lim_{n\to\infty}\frac{1}{n}\ln\left|\frac{df^n}{dx}\right| \tag{3.5}$$

Finally, if we use chain rule [10] for the derivative of the nth iterate and take the limit as n tends to infinity we obtain the Lyapunov exponent as

$$\lambda=\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{n-1}\ln\left|f'(x_i)\right| \tag{3.6}$$

Definition : A mapping $f:[0,1]\to[0,1]$ is transitive if for every pair of subintervals I and J of [0,1] there is an n such that $f^n(I)\cap J\neq$ Ø. [9]

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]

## 3.2. The Tent Map

One dimensional maps have a general form of (3.2) in which f maps variable $x_n$ to $x_{n+1}$. There are two types of tent maps which are symmetric tent map and skew tent map [9, 10].

The symmetric tent map is given by [3]

$$x_{n+1} = \begin{cases} a(1-|2x_n - 1|) & 0 \le x_n \le 1 \\ 0 & elsewhere \end{cases} \tag{3.7}$$

Where $a \in [0.5\ 1]$ for a chaotic behavior.
The skew tent map is given by [4]

$$x_{n+1} = \begin{cases} x_n / a & 0 \le x_n \le a \\ (1-x_n)/(1-a) & elsewhere \end{cases} \tag{3.8}$$

where, the chaotic parameter $a$ determines the behavior of system. Figure 3.1 demonstrates the iteration procedure of the symmetric tent map. It is as follows: From $x_0$ value draw a vertical line, which will give y=$x_1$. From that point follow a horizontal line until it intersects y=x line. From that point follow a vertical line again until it intersects the tent map again and so on.

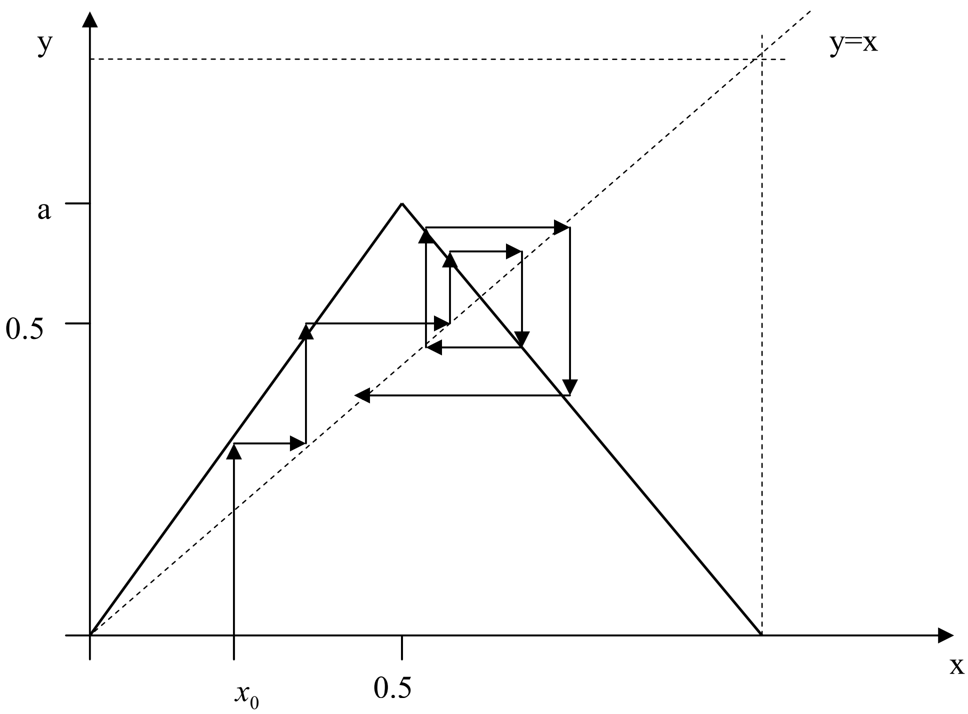

Figure3.1 Sketch of Symmetric Tent Mapping

To see chaotic behavior of tent map and dramatically dependence to chaotic parameter *a* and initial value $x_0$, following simulation results are presented.

Figure 3.2 presents the output of the skew tent map with two different parameters a1=0.8 for continuous line and a2=0.81 for dotted line. The initial value for both cases is $x_0$=0.3.

Figure 3.3 presents the output of the skew tent map with two different initial value $x_0$=0.4 for continuous line and $x_0$=0.41 for dotted line. System parameter a is taken as 0.8 for both cases.

As seen from Figure 3.2 and Figure 3.3, after a few iterations, two maps follow irrelevant paths although system parameters *a* or initial values $x_0$ are very close to each other.

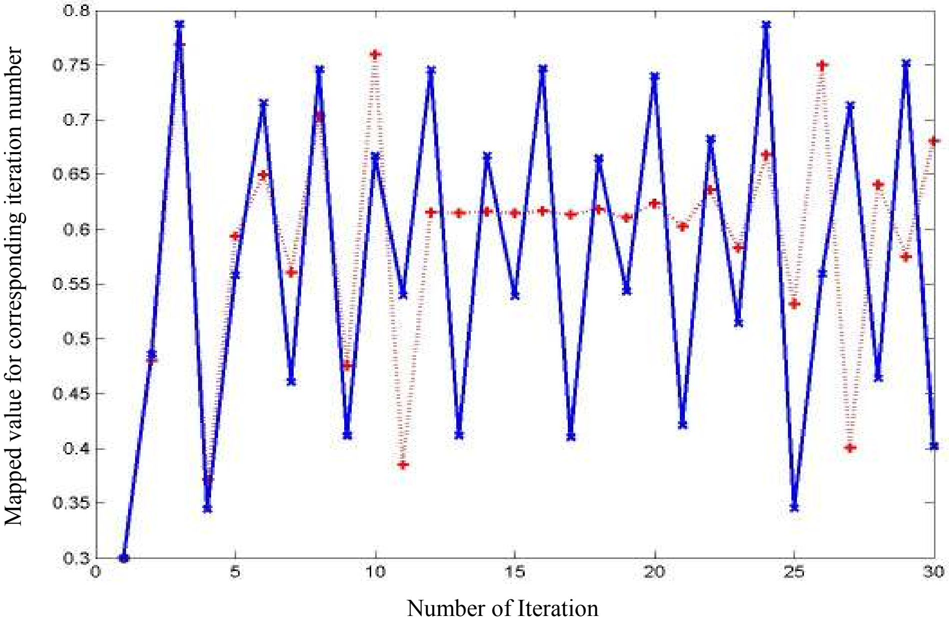

Figure 3.2 Two Tent Maps with different chaotic parameters

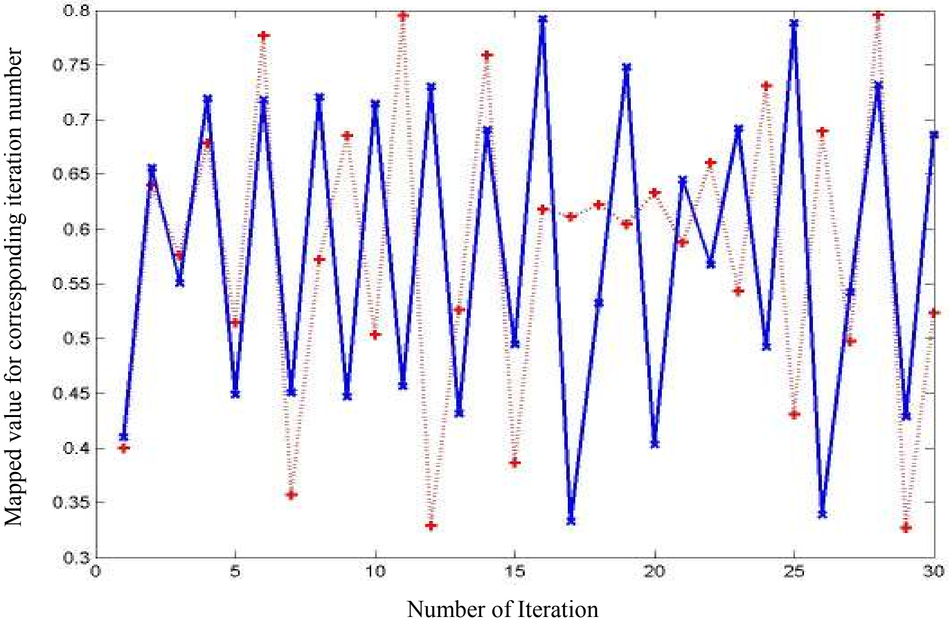

Figure 3.3 Two Tent Maps with different initial values

# CHAPTER 4

# SIMULATION RESULTS FOR STATE ESTIMATION WITH ODSA

In this chapter, results of state estimation for 2 nonlinear systems are presented with simulations. These 2 system models are the skew tent map and the symmetric tent map. State estimation is performed by Optimum Decoding Based Smoothing Algorithm (ODSA) [1], which is reviewed in Chapter 2.

General forms of 2 nonlinear systems used in the simulations are:

$$x_{n+1} = f(x_n) \tag{4.1}$$

$$y_n = x_n + v_n \tag{4.2}$$

where, $x_n$ is the state, $y_n$ is the observation and $v_n$ is the zero mean Gaussian noise with variance $\sigma_v$.

Although all information and formulations about Optimum decoding based smoothing algorithm are reviewed in Chapter 2, it is given below some information that is needed for an practical application of ODSA.

First of all, discrete random variables approximating the Gaussian random variable with zero mean and unit variance is given in Table 4.1 from [1]

Table 4.1 Discrete random variables approximating the Gaussian random variable with zero mean and unit variance

| Number of possible values | y(i) is the possible value and p(i) is the corresponding probability | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| n=1 | y(1)=0<br>p(1)=1 | | | | | | | |
| n=2 | y(1)= -0.675<br>p(1)=0.5 | y(2)= 0.675<br>p(2)=0.5 | | | | | | |
| n=3 | y(1)= -1.005<br>p(1)=0.315 | y(2)= 0<br>p(2)=0.370 | y(3)= 1.005<br>p(3)=0.315 | | | | | |
| n=4 | y(1)= -1.219<br>p(1)=0.223 | y(2)= -0.355<br>p(2)=0.277 | y(3)= 0.355<br>p(3)=0.277 | y(4)= 1.219<br>p(4)=0.223 | | | | |
| n=5 | y(1)= -1.376<br>p(1)=0.169 | y(2)= -0.592<br>p(2)=0.216 | y(3)= 0<br>p(3)=0.230 | y(4)= 0.592<br>p(4)=0.216 | y(5)= 1.376<br>p(5)=0.169 | | | |
| n=6 | y(1)= -1.499<br>p(1)=0.134 | y(2)= -0.767<br>p(2)=0.175 | y(3)= -0.242<br>p(3)=0.191 | y(4)= 0.242<br>p(4)=0.191 | y(5)= 0.767<br>p(5)=0.175 | y(6)= 1.499<br>p(6)=0.134 | | |
| n=7 | y(1)= -1.599<br>p(1)=0.110 | y(2)= -0.905<br>p(2)=0.145 | y(3)= -0.423<br>p(3)=0.162 | y(4)= 0<br>p(4)=0.166 | y(5)= 0.423<br>p(5)=0.162 | y(6)= 0.905<br>p(6)=0.145 | y(7)= 1.599<br>p(7)=0.110 | |
| n=8 | y(1)= -1.683<br>p(1)=0.093 | y(2)= -1.018<br>p(2)=0.123 | y(3)= -0.567<br>p(3)=0.139 | y(4)= -0.183<br>p(4)=0.145 | y(5)= 0.183<br>p(5)=0.145 | y(6)= 0.567<br>p(6)=0.139 | y(7)= 1.018<br>p(7)=0.123 | y(8)= 1.683<br>p(8)=0.093 |

In Table 4.1 possible initial values for a defined “possible number of initial states” are given as y(n) and corresponding probabilities are given as p(n). Note that lnp(n) in table 4.1 gives $\ln \pi_0^i$ which is defined as initial metric in (2.13)

Also $\ln p(y(k) | x_q^i(k))$ from (2.13) is used as below (4.3) in following simulations.

$$\ln p(y(k)|x(k)) = \ln\left\{\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{-\left(y(k)-f(x(k))\right)^2}{2\sigma^2}\right]\right\} \tag{4.3}$$

where, $\sigma^2$ is the observation noise variance.

## 4.1. Type I System

This subsection considers the skew tent map with a observation model, defined by

$$x_{n+1} = \begin{cases} x_n / a, & 0 \le x_n \le a \\ (1-x_n)/(1-a), & a < x_n \le 1 \end{cases} \qquad \text{State model} \qquad (4.4)$$

$$y_n = x_n + v_n \qquad \text{Observation model} \qquad (4.5)$$

where, $v_n$ is the observation noise with zero mean variance $\sigma_v$.

State estimation results of (4.3) with $\sigma^2{}_v$=1, chaotic system parameter a=0.25 and a=0.5 are presented in Figure 4.1 and Figure 4.2 respectively. In simulations 50 iterations were observed. Other parameters for both Figure 4.1 and Figure 4.2 are:

Number of possible values for initial state: 6
Mean of initial value for each sequence: 0.5
Variance of initial value for each sequence: 0.3
Gate size: 0.04

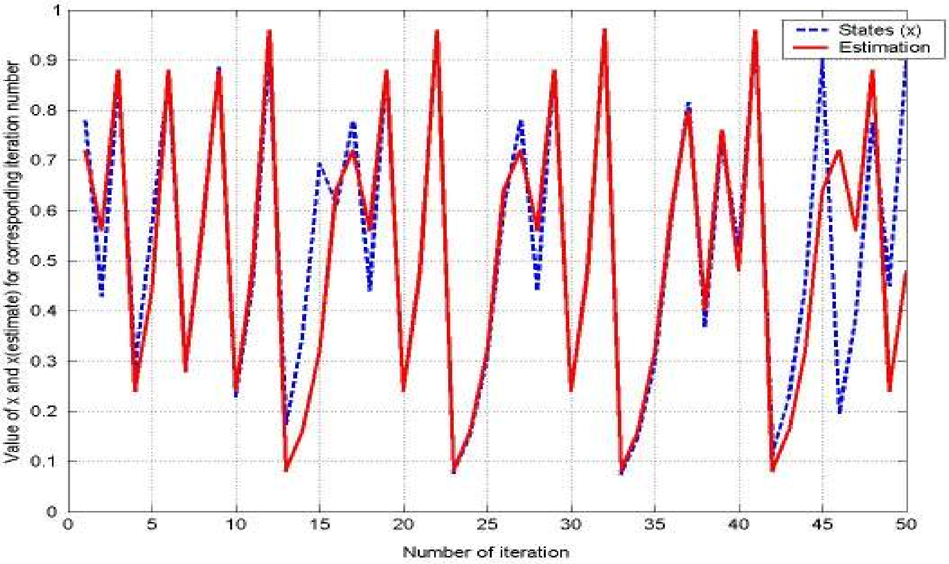

Figure 4.1 Estimated and real values of skew tent map for a=0.25

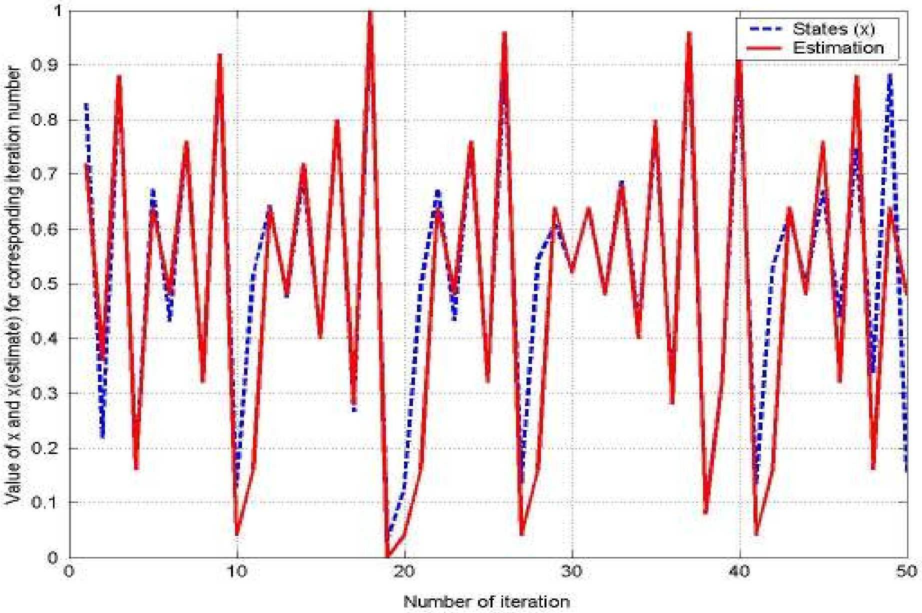

Figure 4.2 Estimated and real values of skew tent map for a=0.5

## 4.2. Type II System

This subsection considers the symmetric tent map with a observation model, defined by

$$x_{n+1} = \begin{cases} a(1-|2x_n - 1|) & 0 \le x_n \le 1 \\ 0 & elsewhere \end{cases} \qquad \text{State model} \qquad (4.6)$$

$$y_n = x_n + v_n \qquad \text{Observation model} \qquad (4.7)$$

where, $v_n$ is the observation noise with zero mean variance $\sigma_v$.

State estimation results of (4.3) with $\sigma^2{}_v$=1, chaotic system parameter a=0.8 and a=0.9 are presented in Figure 4.3 and Figure 4.4 respectively. In simulations 50 iterations were observed. Other parameters for both Figure 4.3 and Figure 4.4 are:

Number of possible values for initial state: 6

Mean of initial value for each sequence: 0.5

Variance of initial value for each sequence: 0.3

Gate size: 0.04

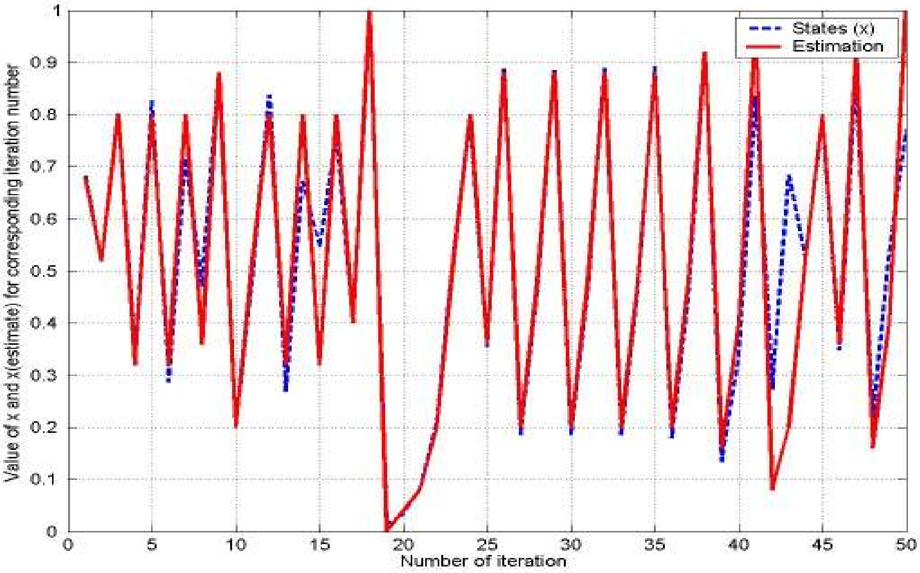

Figure 4.3 Estimated and real values of symmetric tent map for a=0.8

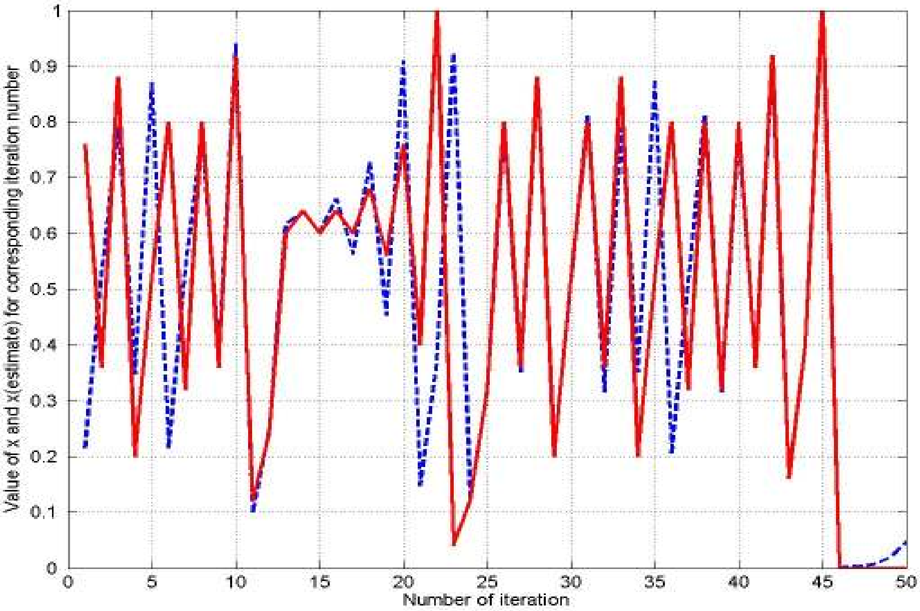

Figure 4.4 Estimated and real values of symmetric tent map for a=0.9

## 4.3. Complexity Analysis

In this part, ODSA run time will be investigated. Effect of gate size and sequence length over program run-time will be analyzed with simulations. After simulations, a formula will be derived for obtained computation run times. For simulations a personal computer with Intel Pentium II 400MHz processor and 256 MB RAM is used.

In Figure 4.5, complexity analysis results are given for gate sizes 0.01, 0.02, 0.04 and finally 0.08. Chaotic system parameter in (4.4) is taken as 0.5. For each plot, sequence length is a changing parameter from 50 to 300 with 50 steps and number of possible values for initial step is taken as 6. Results in Figure 4.5 are mean of 10 repeated runs.

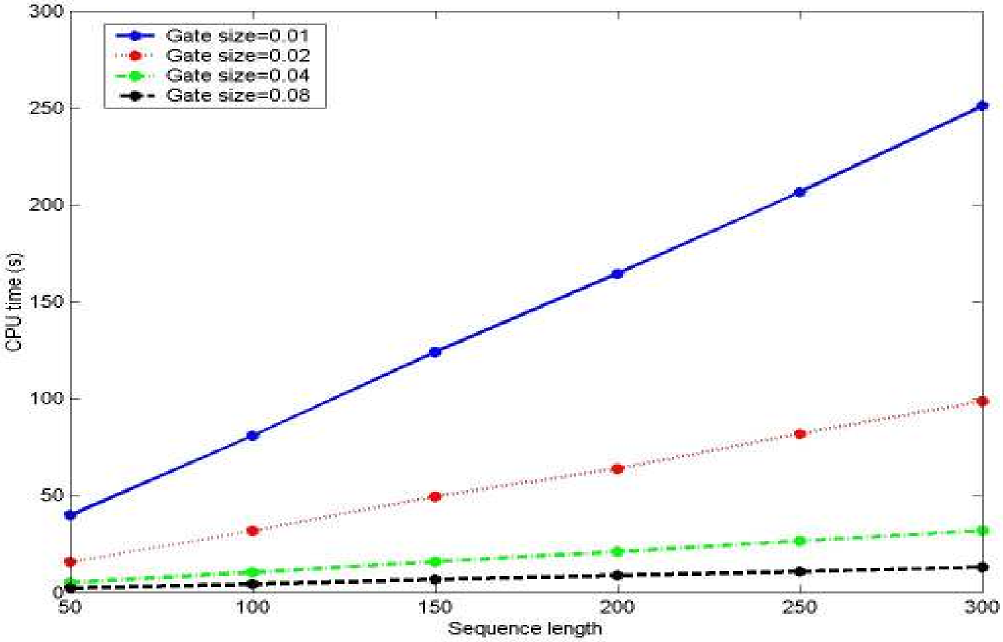

Figure 4.5 Effect of gate size and sequence length over computation (CPU) time

From Figure 4.5, there is a linear relation between sequence length and computation time for all gate sizes. However, effect of gate size is not clear enough from Figure

4.5. So, to better investigate effect of gate size over computation time, next simulation (in Figure 4.6) is performed. In Figure 4.6, gate size is changed from 0.01 to 0.08 with a sequence length of 50 for each gate size. As previous simulation, number of possible values for initial step is taken as 6 and results in Figure 4.6 are mean of 10 repeated runs.

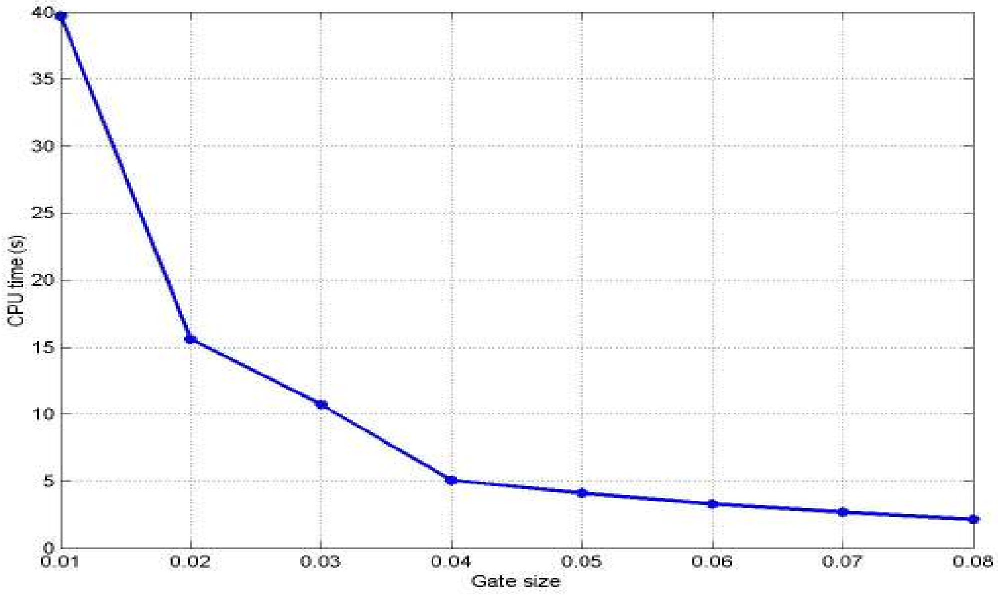

Figure 4.6 Effect of gate size over computation (CPU) time

From Figure 4.6 there is a non-linear relation between gate size and computation time. From these results, an approximation formula for gate sizes smaller than 0.1 can be written as:

$$CPUtime(s) = k * 1.6 * \left(\frac{L}{50}\right)\left(\frac{0.1}{gs}\right)^{1.4} \tag{4.8}$$

where L is the sequence length, gs is the gate size and k is a constant changing for different computers and processor speeds. k is simply 1 for above mentioned computer with which these simulations are performed. Results of approximation formula (4.8) and simulation results are given in Figure 4.7 to better see how closely we approximate computation time with (4.8).

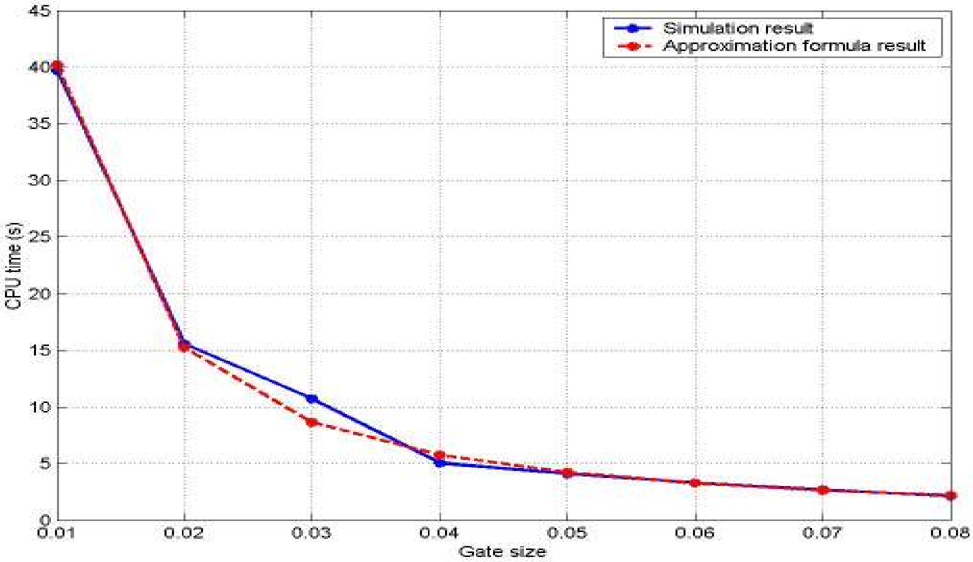

Figure 4.7 Simulation and approximation formula results together for CPU time

At that point another way to see reliability of approximation formula in (4.8) is to try predicting some CPU times for different gate sizes and sequence lengths. In Table 4.2, for 4 different gate size and sequence length pairs; computation time of above mentioned computer and approximation formula results are given together.

Table 4.2 Reliability of approximation formula in (4.8)

| | gs=0.005<br>L=50 | gs=0.025<br>L=150 | gs=0.015<br>L=90 | gs=0.009<br>L=113 |
|---|---|---|---|---|
| Computation time needed (s) | 100.94 | 38.67 | 41.11 | 110.86 |
| Approximation formula (4.8) results (s) | 106.06 | 33.47 | 41.00 | 105.26 |

From Table 4.2, results of approximation formula are not far to real computation times and (4.8) gives an enough idea about computation time for different gate sizes and sequence lengths.

Consequently, Effect of 2 important parameters, sequence length and gate size, over computation time are different. Computation time changes linearly with sequence length. However, effect of decreasing gate size on computation time is faster (see (4.8) ). Between successive gate sizes with small difference on performances, biggest one should be selected to decrease computation time minimum.

# CHAPTER 5

# CHAOTIC DIGITAL MODULATION AND DEMODULATION

Chaotic signals are similar to stochastic processes and can be used for secure communication applications. Figure 5.1 yields the block diagram of a communication system with chaotic modulation and demodulation. On the transmitter side, chaotic systems with different parameters are used to represent binary digital signals. If digital 1 is send to the transmitter, a chaotic sequence generated by SYS1 with a chaotic parameter *a1* is generated by the transmitter. However, if digital 0 is send to the transmitter, a chaotic sequence generated by SYS2 with parameter *a2* is generated by the transmitter. In all simulations, digital 1 and digital 0 are generated randomly and with equal probabilities. Signals are transmitted through an Additive White Gaussian Noise (AWGN) channel. At the receiver, two ODSA's [1] matched to chaotic parameters on the transmitter side are used. Finally outputs of state estimators are used for the decision-making.

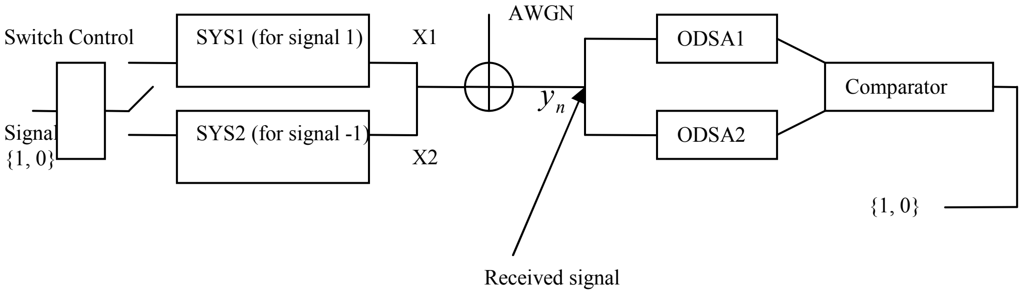

Figure 5.1 Communication Scheme with ODSA

In [5], EKF is used instead of ODSA in Figure 5.1

For decision making, comparator compares average estimation errors from the two ODSA's to decide the transmitted signal. Throughout the thesis, a1 is used for digital 1 and a2 is used for digital 0. Mean Square Errors (MSE's) for ODSA and EKF estimates are defined as

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

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

where, $y_n$ is the observation and it is the same for both MSE1 and MSE2. $\hat{x}_1$ and $\hat{x}_2$ are estimates with ODSA1 (or EKF1) which uses chaotic parameter a1 and ODSA2 (or EKF2) which uses chaotic parameter a2 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 a1 (demodulated by ODSA1) is smaller than the mean square error corresponding to parameter a2 (demodulated by ODSA2), then digital 1 is decided. Otherwise (MSE2<MSE1) digital 0 is decided.

If the length of chaotic sequences is L and if the frequency of data bits is *f,* then the sampling frequency will be L*f.* In other words, for only one bit, the transmitted sequence will be [x1,x2,…,xL]. Another important point is that the beginning of each bit is generated from the end of the previous bit. So, every sample in whole chaotic sequence depends on the previous sample although there are more than 1 bit. In addition, it is investigated if it is important or not for the performance of communication to have a chaotic structure that every sample depends on the previous sample. For this purpose, the beginning of every bit is also chosen in the interval [0 1] randomly. They are compared in Table 5.1. From the performance of the communication system, bit error rate (BER) is intended which is defined as the ratio of erroneous bits over all transmitted bits. In simulations, it is given as the percentage of the transmitted bits.

Of course we can modulate not only binary signals (1 or 0). For example to modulate two bits at once (00 01 10 11) we should use 4 chaotic modulators with 4 different parameters (a1, a2, a3, a4) and at the receiver side we should use 4 ODSA's. Decision-making rule will be the same: Compare estimation errors of all ODSA's and choose the one with minimum Mean Square Error.

A final note is that if we choose L, the length of chaotic sequence, longer; probability of bit error will be better (This is shown in table 5.1 and 5.2 and will be mentioned later).

For comparison purposes, system in figure 5.1 is repeated with all simulations using Extended Kalman Filter (EKF) [7,12] based estimators instead of ODSA based estimators.

## 5.1. ODSA Based Digital Demodulation

Two important parameters directly effecting performance of communication are the parameters a1 and a2 of chaotic systems SYS1 and SYS2. These parameters are user defined.

a1 and a2 parameters of SYS1 and SYS2 can be between 0 and 1. In the following simulation, a1 and a2 parameters are changed from 0.04 to 0.96 with 0.04 steps with all other parameters staying constant. A total of 24x24=576 different a1 - a2 pairs are tried and corresponding bit error rates are shown. An important note on simulation result is that, the simulation is repeated for 50 times and results in figure 5.5 and 5.6 are the average over these 50 simulations. Figure 5.5 presents the bit error rate (BER) versus chaotic parameters a1 and a2. Figure 5.6 (top view) does not have any different information from Figure 5.5. It is given to have another view angle and so making easier comments.

Parameters of each one of 50 simulations are:

Number of bits sent: 100 (Totally, 100x50=5000 bits for each a1-a2 pair and 576x5000=2880000 bits for all surface data)

Length of chaotic sequence for each bit: 50

Gate size: 0.04

Number of possible values for initial state: 6

Mean of initial value for each sequence: 0.5

Variance of initial value for each sequence: 0.3

Mean of observation noise for each sequence: 0

Variance of observation noise for each sequence: 0.2

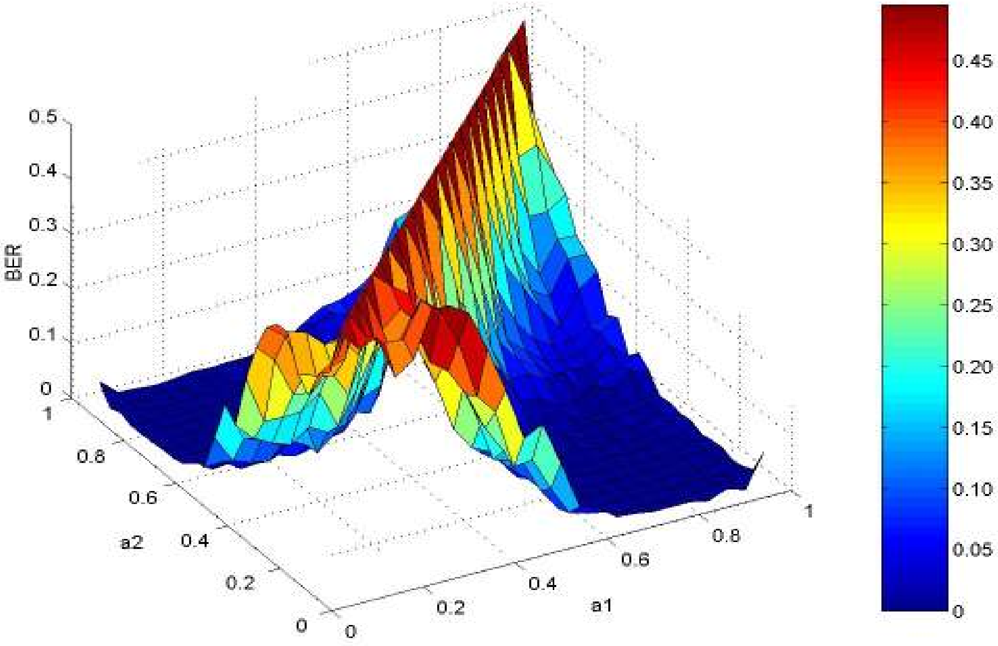

Figure 5.2 BER change with a1 and a2 (side view)

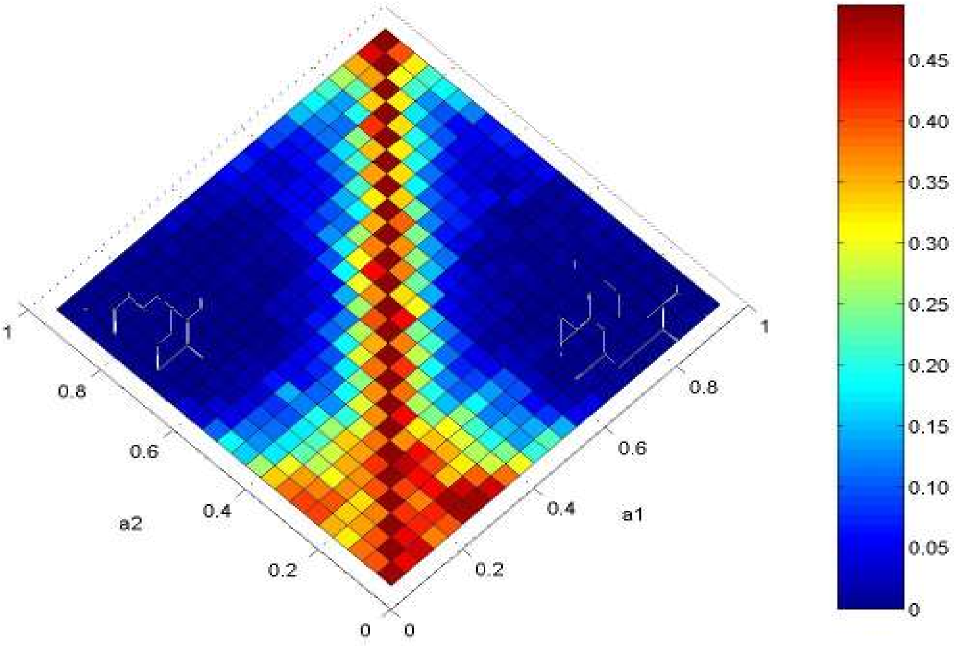

Figure 5.3 BER change with a1 and a2 (top view)

As seen from Figure 5.2 and Figure 5.3, it is apparent that, closer a1 and a2 values results in higher bit error rate (BER). From Figure5.5, error density for low a1-a2 pairs is higher than high a1-a2 pairs. This is a result of signal characteristics of chaotic sequences that with small chaotic parameters (a1 or a2) fluctuations on signal are higher than large chaotic parameters (see Figure5.4). So tracking a high fluctuating signal is more difficult and error rates are higher for low value a1-a2 pairs. One may think that it should be chosen a1 value near 0 and a2 value near 1 (or vice versa) for lowest BER. In fact this is a true observation from figures 5.2 and 5.3. However, the situation is not so simple because noise-like structure of generated chaotic sequence should also be considered since it is a secure communication system. Let us explain it with some graphical simulations:

From Figures 5.4 to 5.8, a 5 bit output sequence is given, having a chaotic sequence length of 50 for each bit (total sequence length is 250 for 5 bits). Different a1 and a2 pairs are tried (0.05-0.95, 0.15-0.85, 0.25-0.75, 0.35-0.65, 0.45-0.55 as a1-a2).

Used bit sequence for each pair was [1 0 1 0 1]. Especially for figures 5.4 and 5.5, one can easily divide graphics into 5 parts with 2 kinds of patterns (one can guess that a sequence [x y x y x] is sent). This is of course not good for secure communication because in this case communication scheme loses its noise-like behavior. Not to lose noise-like property at the output of chaotic modulator, a1 and a2 should not be chosen close to 0 and 1 (or far away to each other).

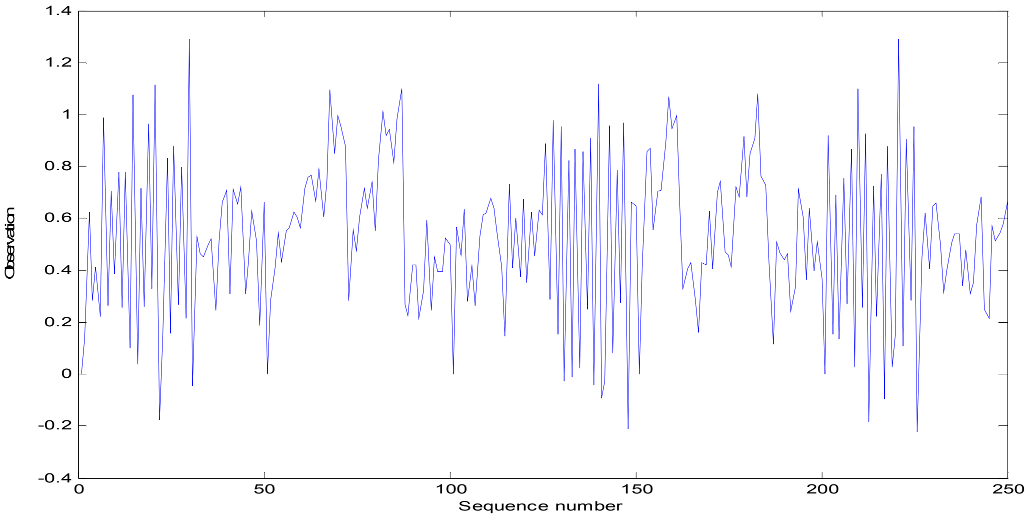

Figure 5.4 Received signal for sequence [1 0 1 0 1] with a1=0.05 a2=0.95

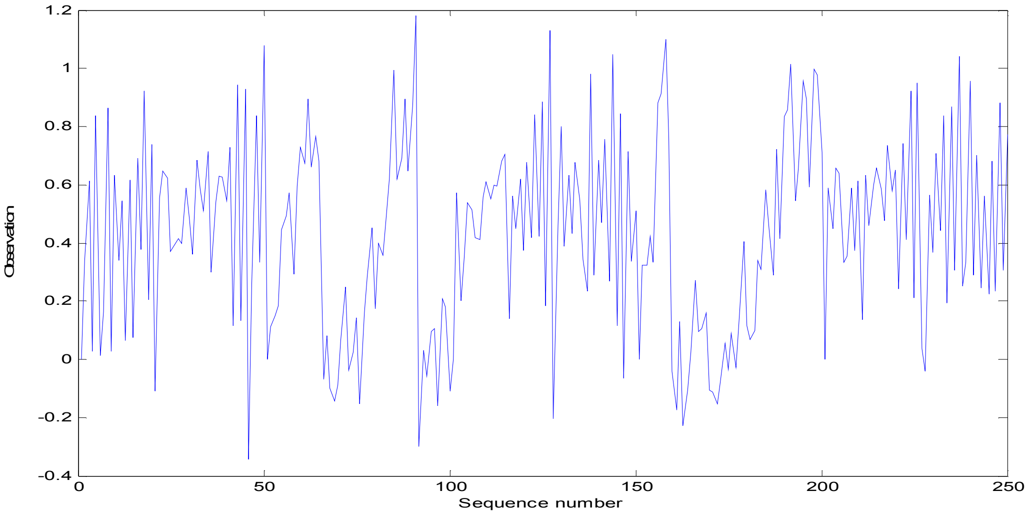

Figure 5.5 Received signal for sequence [1 0 1 0 1] with a1=0.15 a2=0.85

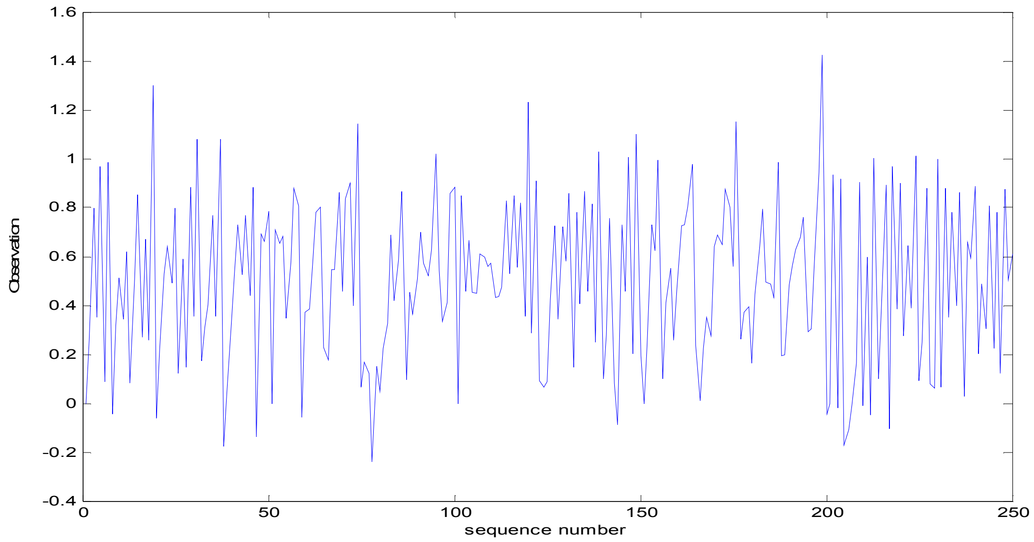

Figure 5.6 Received signal for sequence [1 0 1 0 1] with a1=0.25 a2=0.75

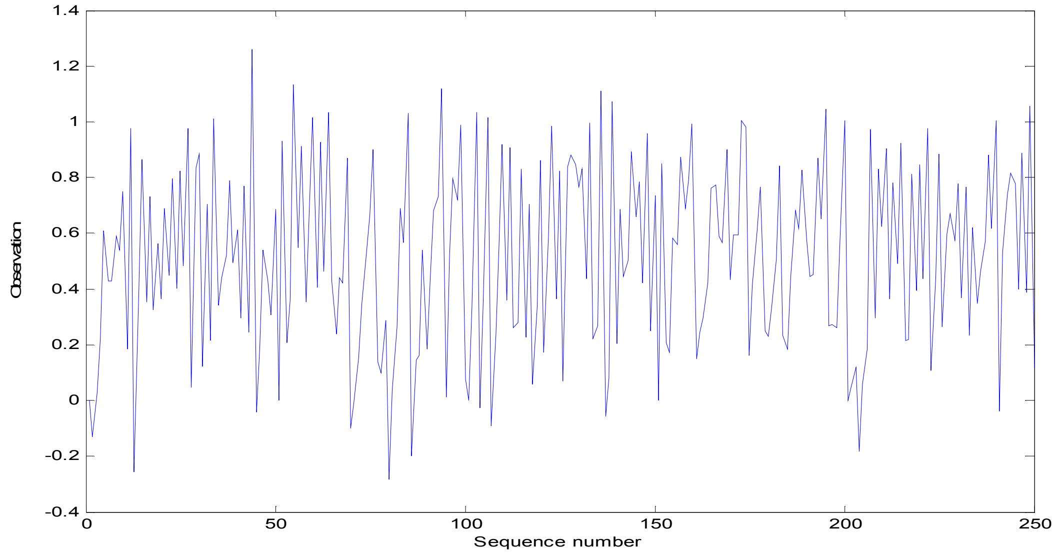

Figure 5.7 Received signal for sequence [1 0 1 0 1] with a1=0.35 a2=0.65

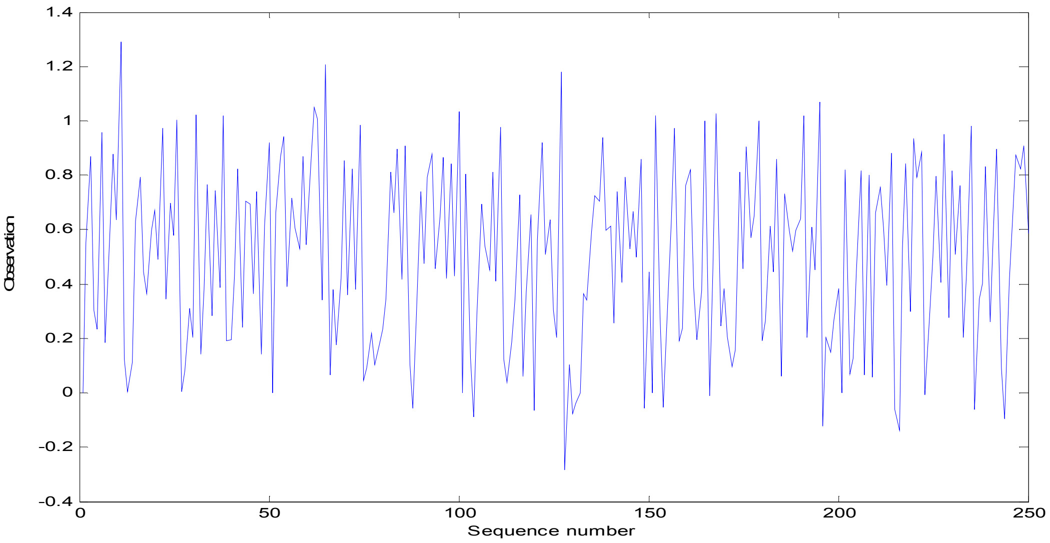

Figure 5.8 Received signal for sequence [1 0 1 0 1] with a1=0.45 a2=0.55

a1 and a2 values must be chosen after a trade of between bit error rate (BER) and noise-like structure considerations. From now on, for all simulations, 0.25 and 0.75 will be accepted for a1 and a2 respectively.

After deciding a1 and a2, we should investigate how BER is affected by observation noise variance and length of chaotic sequence for each bit.

Figure 5.9 shows BER change for observation noise variance from 0.01 to 1 with 0.01 step sizes. 3 different gate sizes are used for comparison purposes and simulation is repeated 50 times. In Figure 5.9, the mean of these 50 runs is given. Other system parameters are as follows:

a1=0.25, a2=0.75

Number of bits sent: 100 (Totally, 50x100=5000 bits for each variance value from 0.01 to 1)

Length of chaotic sequence for each bit: 50

Gate sizes= 0.02, 0.04, 0.08

Number of possible values for initial state: 6

Mean of initial value for each sequence: 0.5

Variance of initial value for each sequence: 0.3

Mean of observation noise for each sequence: 0

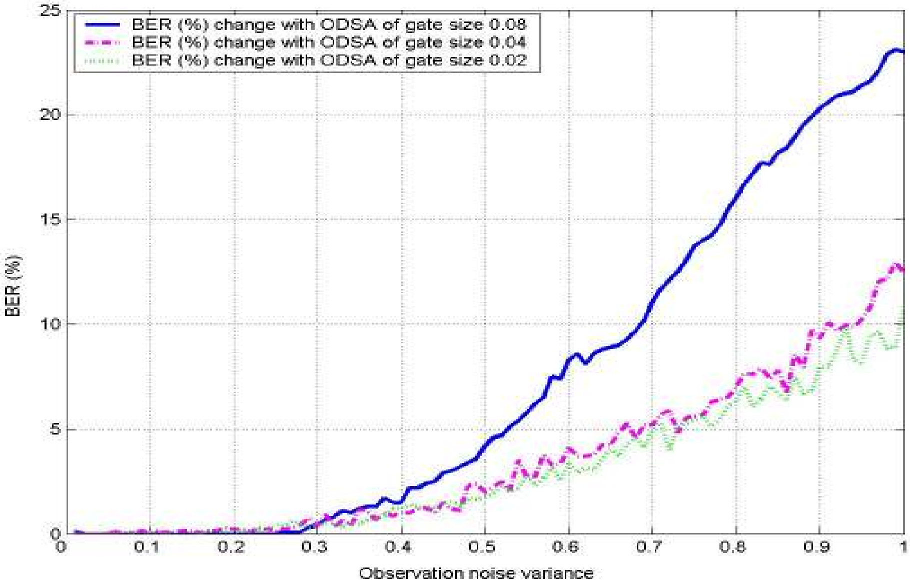

Figure 5.9 BER (%) change with observation noise variance

From Figure 5.9 bit error rate increases rapidly with increasing observation noise variance.

Although performances of ODSA's with gate sizes 0.02 and 0.04 are very close to each other; when we increase gate size to 0.08 there is a fast decrease on performance of communication system in comparison with gate sizes 0.02 and 0.04.

Table 5.1 shows the effect of the length of chaotic sequence over BER. However it should be kept in mind that the length of chaotic sequence for each bit directly effects computation time. In this simulation, length of chaotic sequence is changed from 50 to 600 with 50 step sizes (total of 12 chaotic sequence lengths are tried).

In Table5.1, the second row was obtained when the starting value of one bit (initial value of chaotic structure) follows end value of previous bit (last value of chaotic structure of previous bit). In such a case, all samples in the whole sequence is depending on previous sample. Third row in Table 5.1 is obtained when first value of each bit starts randomly in [0 1] interval. So, whole sequence is a combination of sequences with random chaotic starts for each bit.

Other system parameters are

a1=0.25, a2=0.75
Number of bits sent: 100 (Totally, 50x100=5000bits for each value from 50 to 600)
Gate size: 0.04
Number of possible values for initial state: 6
Mean of initial value for each sequence: 0.5
Variance of initial value for each sequence: 0.3
Mean of observation noise for each sequence: 0
Observation noise variance for each sequence: 1

Table 5.1 Effect of Length of Chaotic sequence on BER

| Length of chaotic sequence for each bit | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 | 550 | 600 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BER (%)-First value of each bit is generated from end of previous bit | 9.6 | 3.26 | 1.58 | 0.8 | 0.34 | 0.18 | 0.14 | 0.06 | 0.02 | 0 | 0 | 0 |
| BER (%)–Each bit starts with a random number in [0 1] interval | 10.82 | 3.6 | 1.48 | 0.9 | 0.48 | 0.15 | 0.12 | 0.1 | 0.02 | 0 | 0 | 0 |

From table 5.1, it is apparent that an increase of the length of chaotic sequence dramatically affects BER. For example, with a length of 500, low BER values were obtained although observation noise variance is high (Noise variance is 1 and signal varies between [0 1] interval because of self nature of tent mapping). There is a slight difference between BER's due to starting type of first value of chaotic sequences for each bit. If the first value of each bit is randomly chosen, BER is higher. But if the starting value of each bit is chosen from the end of previous bit, BER is lower. This difference between two cases is higher for short length chaotic sequences in comparison with high length chaotic sequences.

Another way to see the effect of chaotic sequence length on the performance is to look at the mean square error (MSE) plot. Figures 5.10,11,12 give such MSE plots for 100 random bits for chaotic sequence lengths of 50, 100, 200 respectively. Solid line is for real (sent) bit sequence and dashed line is for false bit sequence (1 for 0 and 0 for 1). As it is stated, decision making rule is that: compute and compare the mean square errors for a1 and a2 matched filter outputs. Choose smaller one and look for corresponding bit (1 for a1 and 0 for a2). For a bit, if MSE of false bit is smaller than MSE of true bit then false bit will be decided. In Figures 5.10, 11 and 12, the solid and dashed lines which apart from each other are better. Otherwise these lines cross each other which results in bit errors.

Other system parameters for Figures 5.10, 11, 12 are the same as used for Table 5.1.

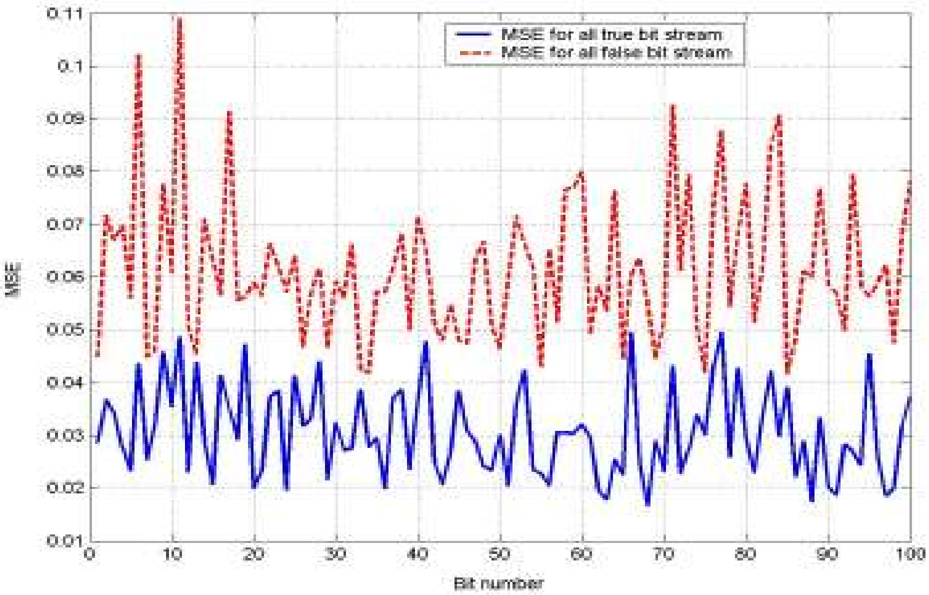

Figure 5.10 MSE plot for 100 bits with chaotic sequence length 50

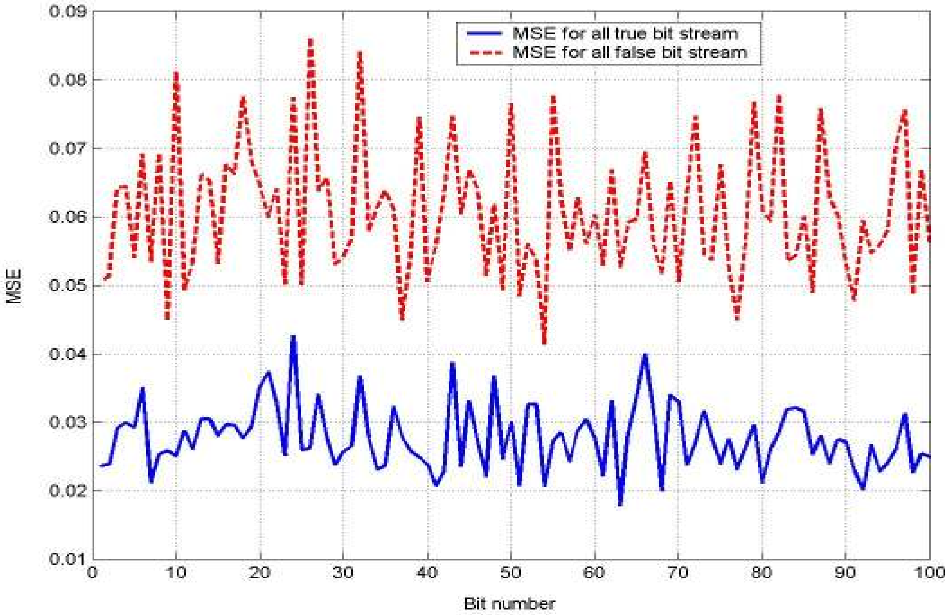

Figure 5.11 MSE plot for 100 bits with chaotic sequence length 100

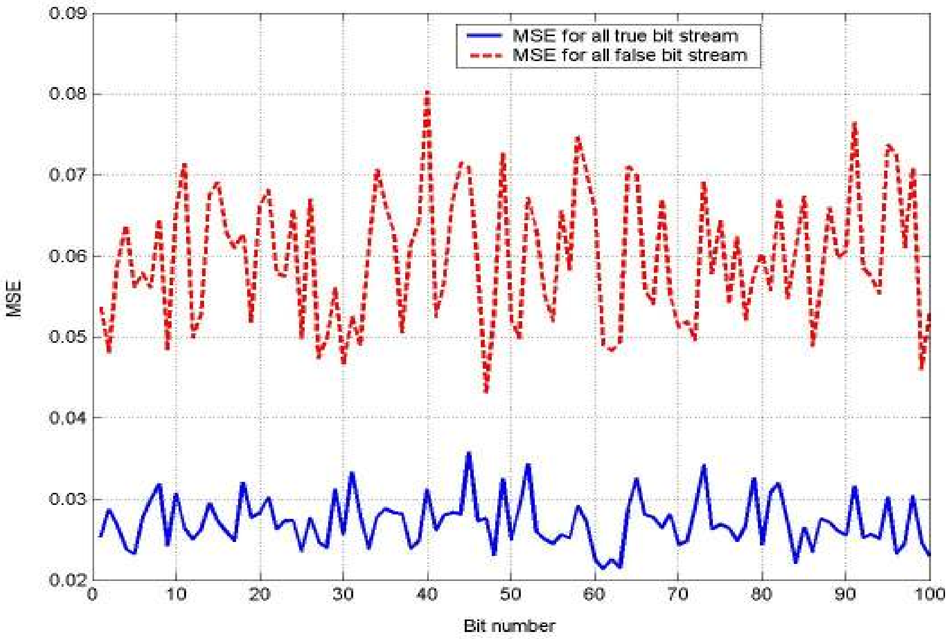

Figure 5.12 MSE plot for 100 bits with chaotic sequence length 200

From figures 5.10-11-12, with increasing chaotic sequence length for each bit, fluctuations on MSE of true bit stream are decreasing.

## 5.2. EKF Based Digital Demodulation

In this part, simulations of "5.1 ODSA Based Digital Demodulation" will be applied to the Extended Kalman Filter (EKF) [7, 11, 12] based demodulation system.

State space model equations are:

$$x_{n+1} = f(x_n) = \begin{cases} a(1-|2x_n - 1|) & 0 \le x_n \le 1 \\ 0 & elsewhere \end{cases} \qquad \text{State model} \qquad (5.3)$$

$$y_n = g(x_n, v_n) = x_n + v_n \qquad \text{Observation model} \qquad (5.4)$$

The EKF produces a state estimate

$$\hat{x}_{n+1} = f(\hat{x}_n) + K_n(y_n - \hat{x}_n) \tag{5.5}$$

$K_n$ is the Kalman gain updated at each step and is given by:

$$K_n = \frac{A_n P_n}{P_n + \sigma^2} \tag{5.6}$$

where, $A_n = \frac{\partial f(\hat{x}_n)}{\partial x}$ and $\sigma^2$ is observation noise variance. Pn is given as:

$$P_{n+1} = A_n P_n A_n^T - \frac{A_n P_n^2 A_n^T}{P_n + \sigma^2} \tag{5.7}$$

At the beginning of simulations and analyses; effects of a1 and a2, chaotic system parameters, on BER's are considered. Similar to the ODSA case, a1 and a2 are changed from 0.04 to 0.96 with 0.04 steps and number of erroneous bits is found over 100 bits for each one of 576 a1-a2 pairs. This simulation is repeated for 50 times and the results in Figure 5.13 and Figure 5.14 are the average of these 50 simulations.

Parameters of each one of 50 simulations are:
Number of bits sent: 100 (Totally, 100x50=5000bits for each a1-a2 pair and 576x5000=2880000 bits for all surface data)
Length of chaotic sequence for each bit: 50
Gate size: 0.04
Number of possible values for initial state: 6
Mean of initial value for each sequence: 0.5
Variance of initial value for each sequence: 0.3
Mean of observation noise for each sequence: 0
Variance of observation noise for each sequence: 0.2

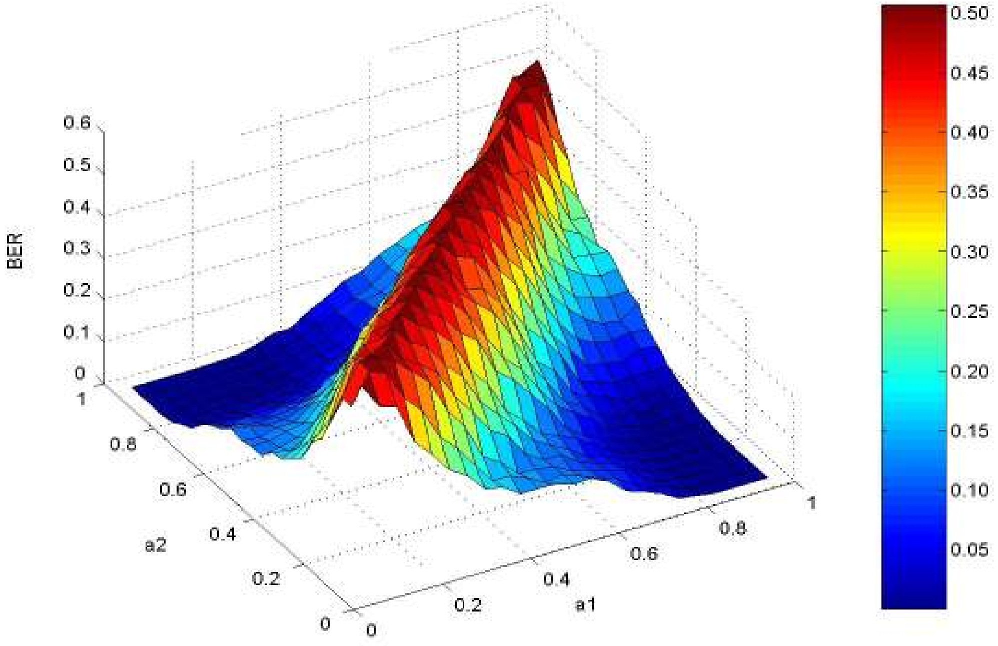

Figure 5.13 BER change with a1 and a2 (side view)

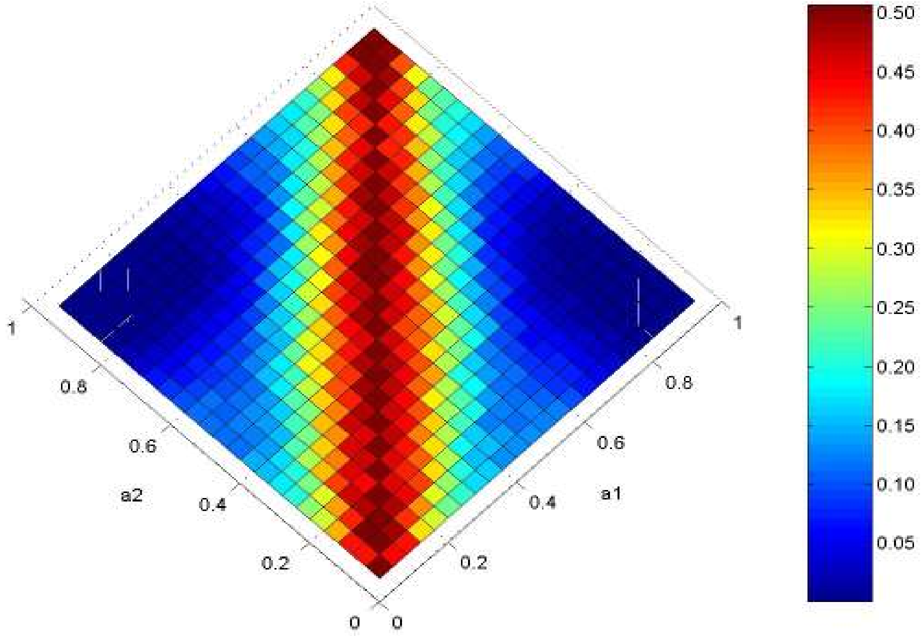

Figure 5.14 BER change with a1 and a2 (top view)

From these figures, situation is similar to one in ODSA based communication scheme. Closer a1 and a2 values results higher BER values. To be able to make a fair comparison between ODSA and EKF based demodulation schemes, a1 and a2 are chosen the same as in previous part, 0.25 and 0.75 respectively.

Figure 5.15 shows the BER's for observation noise variances from 0.01 to 1 with 0.01 steps. Results are given for EKF based estimator and ODSA [1] based estimators of gate sizes 0.02, 0.04 and 0.08 together. From the figure, BER of ODSA with gate size 0.02 and 0.04 is less than with EKF for all variances from 0 to 1. However, another result from figure is that EKF based simulations tend to give better results for high noise variances whereas ODSA based simulations tend to give better results for comparatively small noise variances.

Simulation is repeated 50 times. In Figure 5.15 the mean of these simulations is given. Other system parameters are as follows:

a1=0.25, a2=0.75
Number of bits sent: 100 (Totally, 50x100=5000 bits for each variance value from 0.01 to 1)
Length of chaotic sequence for each bit: 50
Number of possible values for initial state: 6
Mean of initial value for each sequence: 0.5
Variance of initial value for each sequence: 0.3
Mean of observation noise for each sequence: 0

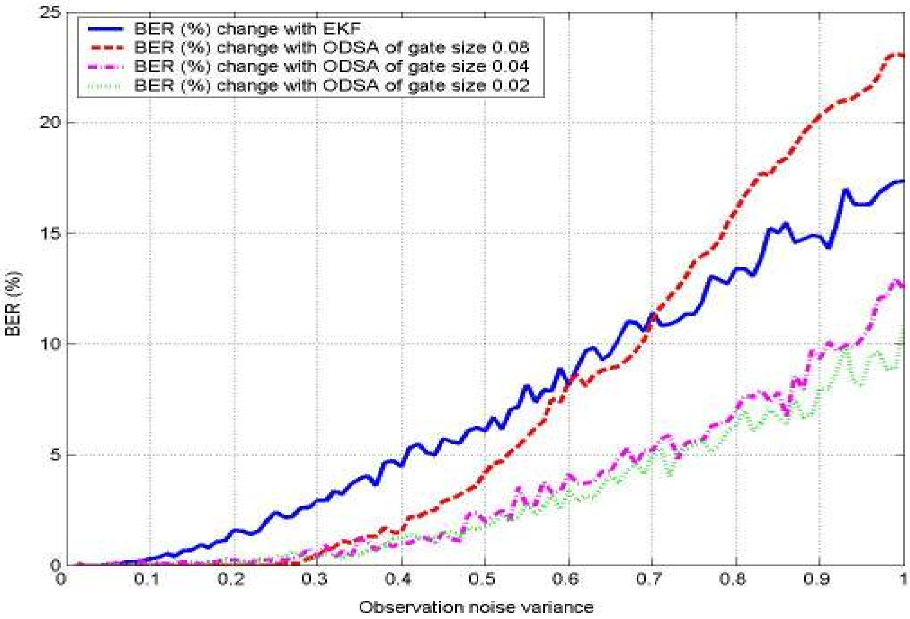

Figure 5.15 BER change with observation noise variance

From Figure 5.15, better results of ODSA over EKF for gate sizes 0.02 and 0.04 can be seen.

Table 5.2 shows the effect of the chaotic sequence length on BER. Simulation parameters are the same as used in ODSA. Length of chaotic sequence for each bit is changed from 50 to 600 with 50 steps and simulations are repeated 50 times. Results in Table 5.2 are the mean of these 50 runs. Other parameters are given below:

a1=0.25, a2=0.75
Number of bits sent: 100 (Totally, 50x100=5000bits for each value from 50 to 600)
Number of possible values for initial state: 6
Mean of initial value for each sequence: 0.5
Variance of initial value for each sequence: 0.3
Mean of observation noise for each sequence: 0
Observation noise variance for each sequence: 1

Table 5.2 Effect of Length of Chaotic sequence on BER

| Length of chaotic sequence for each bit | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500 | 550 | 600 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| BER (%)-First value of each bit is generated from end of previous bit | 21.4 | 12.2 | 7.2 | 3.88 | 2.38 | 1.42 | 0.4 | 0.42 | 0.06 | 0.04 | 0.04 | 0.04 |
| BER (%)– Each bit starts with a random number in [0 1] interval | 21.5 | 13.28 | 7.31 | 3.44 | 2.51 | 1.43 | 0.37 | 0.31 | 0.2 | 0.06 | 0.04 | 0.04 |

Figure 5.15 and Table 5.2 gives similar results. For both of them, BER for ODSA is lower than EKF with gate size 0.04.

Figures 5.16, 17 and 18 gives the mean square error (MSE) plots for chaotic sequence lengths 50, 100, 200 respectively. Figures include MSE's for both true and false bit streams. Other system parameters are the same as used for Table 5.2.

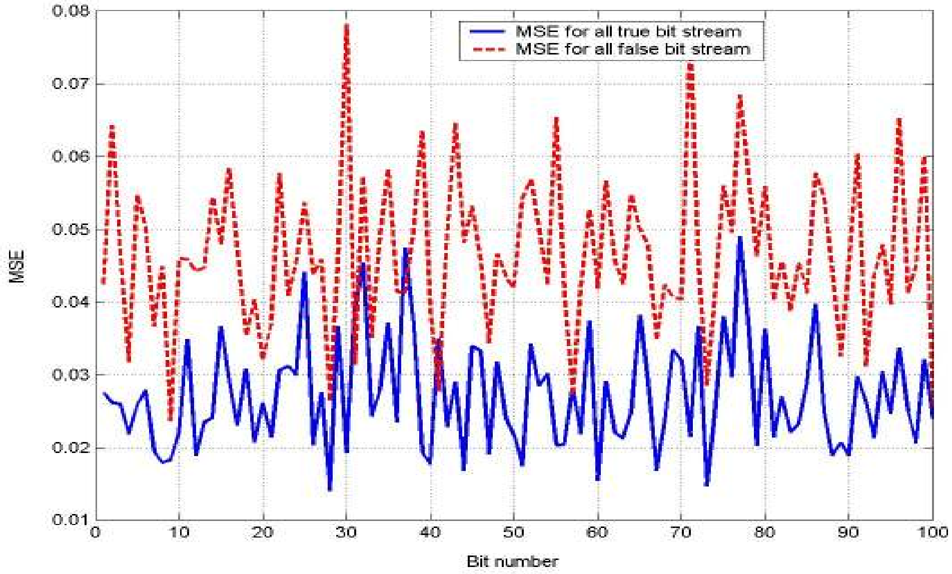

Figure 5.16 MSE plot for 100 bits with chaotic sequence length 50

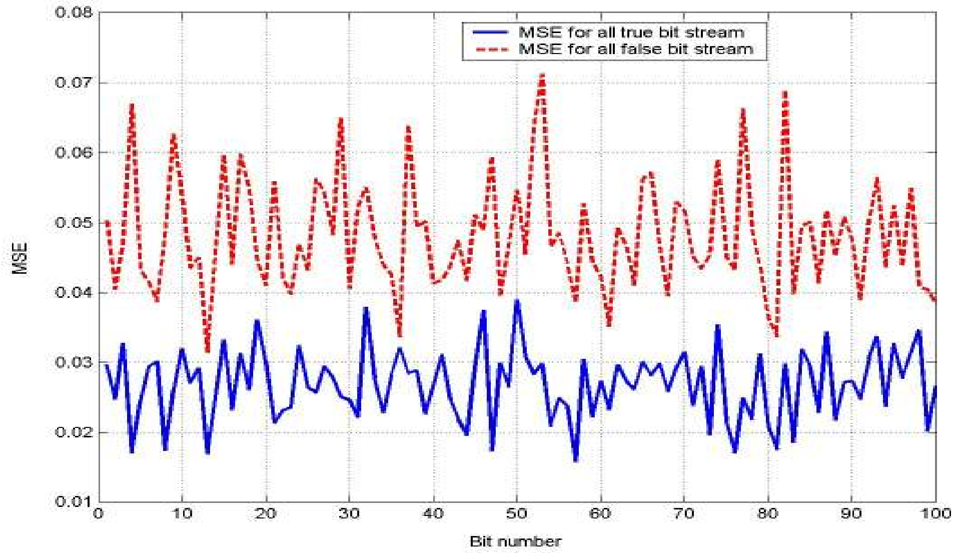

Figure 5.17 MSE plot for 100 bits with chaotic sequence length 100

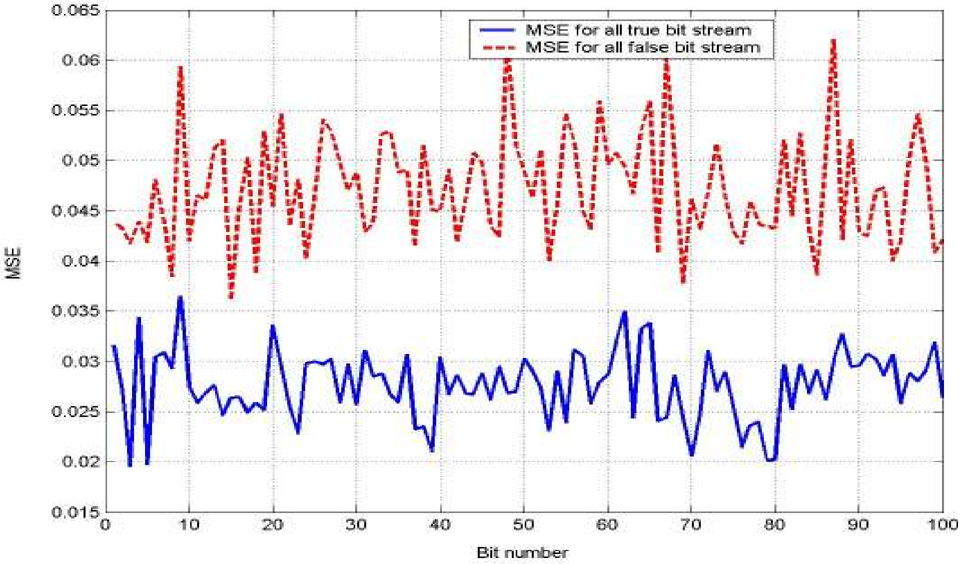

Figure 5.18 MSE plot for 100 bits with chaotic sequence length 200

For Figures 5.16, 17 and 18, the effect of chaotic sequence length on MSE of true bit stream is similar to the case in ODSA. With increasing chaotic sequence length, fluctuations of MSE of true bit stream are decreasing. For Figure 5.16 at some points (bits) MSE of false bit is computed smaller than true bit (solid line over dashed line). That means there is an error (a false decision) at corresponding bit.

# CHAPTER 6

# CONCLUSION

In this thesis, the performances of chaotic communication systems with EKF and ODSA demodulators are compared. Used chaotic model is the skew tent map, which has noise like properties and this property makes it useful for secure communication. Also ODSA is a useful algorithm for use at the receiver of communication scheme for chaotic modulated signals.

When we compare BER's of ODSA and EKF, ODSA gives better results (lower BER) with gate sizes 0.04 and 0.02 than the EKF based demodulation, and performances for gate sizes 0.02 and 0.04 are very close. However it should be noted here that computation time (CPU time) of EKF demodulator based simulations are about 7 times faster than the ODSA demodulator based simulations with gate size 0.04.

With large lengths of chaotic sequences and appropriate chaotic parameters (a1 and a2) it is possible to obtain low BER for both ODSA and EKF based demodulators. As it is seen from Table 5.1 and 5.2 with a chaotic sequence length of 600, BER's are less than 1/5000 for ODSA and around 2/5000 for EKF (with observation noise variance equal to 1 which is a high value for a chaotic generated sequence which is changing between 0-1 interval). The performance with the ODSA is better than the performance with the EKF.

From the results, selection of chaotic parameters, a1 and a2, is important not only for bit error rate but also for noise like property of modulated signal waveform. Choosing closer chaotic parameters results in large bit error rates but apart chaotic parameters will result worse noise-like property. Also for chaotic sequence length of each bit, there is a trade off: Longer sequence lengths result in smaller bit error rates but longer computation times.

The performance of the ODSA based presented communication system seems satisfactory for secure communication applications. One drawback of used demodulation scheme is computation time which uses two estimators (ODSA1, ODSA2).

# REFERENCES

[1] Kerim Demirbaş, “Advances in Control and Dynamic Systems”, vol. 21, Academic Press, pp175-295, 1984

[2] S. Neil Rasband, “Chaotic Dynamics of Nonlinear Systems”, A Wiley-Interscience Publication, 1989

[3] Tongyan Zhai, Huawei Ruan, Edwin E. 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 Maui, Hawaii USA, December 2003

[4] 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

[5] Tongyan Zhai, Huawei Ruan, Edwin E. Yaz, “A Demodulation Scheme Based on State Estimation for Chaotic Digital Communication”, Proceedings of the American Control Conference Denver, Colorado June 4-6, 2003

[6] Tongyan Zhai, Huawei Ruan, Edwin E. Yaz, “A Chaotic Secure Communication Scheme with Extended Kalman Filter Based Parameter Estimation”, IEEE International Conference on Control Applications, Istanbul, Turkey, pp. 404-408, 2003

[7] Bishop Gary, Welch Greg, “An Introduction to the Kalman Filter” UNC-Chapel Hill, TR95-041, April 5, 2001

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

[10] G.L.Baker, J.P.Gollup, “Chaotic Dynamics: an Introduction”, Cambridge University Press, pp 84-86, 1996

[11] M. Norgaard, N.K.Poulsen, O.Ravn, “New Developments in State Estimation for Nonlinear Systems Automatica”, (36:11), pp 1627-1638, November 2000

[12] M. Norgaard, “KALMTOOL versiyon 2 for Use with MATLAB”, Technical Report IMM-REP-2000-6, December 2002