Summer 2014

Optical Position Sensing In Free Space Optical Communication

Muhammad Salman Bashir
Purdue University

Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Muhammad Bashir

Entitled
Optical Position Sensing in Free Space Optical Communication

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

MARK R. BELL
Chair

JAMES V. KROGMEIER

STANISLAW H. ZAK

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): MARK R. BELL

Approved by: M. R. Melloch 06-06-2014
Head of the Graduate Program Date

OPTICAL POSITION SENSING IN FREE SPACE OPTICAL COMMUNICATION

A Thesis

Submitted to the Faculty

of

Purdue University

by

Muhammad Bashir

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

August 2014

Purdue University

West Lafayette, Indiana

Dedicated to Aimen, Mahnoor and Tyas.

ACKNOWLEDGMENTS

I would like to acknowledge my adviser Prof. Mark R. Bell for his unstinting support, advice and encouragement throughout my graduate studies. His continuous guidance has been a source of motivation for me during the preparation of this thesis.

I would also like to acknowledge my family for their immense cooperation and prayers.

TABLE OF CONTENTS

Page

Page

LIST OF FIGURES

ABSTRACT

Bashir, Muhammad M.S.E.C.E, Purdue University, August 2014. Optical Position Sensing in Free Space Optical Communication. Major Professor: Mark R. Bell.

In this thesis the performance of three estimators (center of gravity, template matching and maximum likelihood (MLE)) to estimate the center of the beam on a photoemissive receiver array in a Free Space Optics (FSO) system is compared in terms of mean square error. Simulations have been conducted in Matlab by generating a two dimensional nonhomogeneous Poisson process and mean square error is computed for three estimators. The cases of continuous and discrete arrays are also considered for various levels of signal-to-noise ratio. Simulations have shown that the MLE gives the least mean squared error and especially performs significantly better than other estimators under poor SNR conditions.

1. INTRODUCTION

1.1 Free Space Optics

Free Space Optics (FSO) is a communication technology that involves transmission of light (both infrared and visible) through free space or terrestrial atmosphere in order to transmit data in the form of point-to-point or point-to-multi-point links. Thus call it the "wireless" or "fiberless" analog of a typical Fiber Optics system with all its advantages of higher bandwidth and none of the drawbacks of difficult installation/maintenance. The chief advantage of FSO system lies in its ability to transfer high volumes of data on an increasingly large bandwidth which is almost impossible to achieve on conventional radio frequency networks.

The primary application of FSO, in the framework of this thesis, is in the realm of Deep Space Communications. This involves interacting with a satellite from earth once its launched in a space mission. Then the only means of malfunction diagnosis and its solution is possible through a communication network since a large number of such missions never return to earth. Without a reliable communication network it would be hard to launch a successful space mission.

Another equally important advantage is the ease of installation/deployment of an FSO system in very short periods of time e.g. in case of extension or continuation of a fiber optic link across a stretch of river or harsh terrain where it might not be feasible to lay cables etc. Similarly this advantage is of primary importance in disaster recovery situations where time delays are not acceptable to set up communication links for relief administering infrastructure.

Further the deployment of FSO system is cheaper as opposed to fiber driven technologies as no additional expenses are incurred to acquire fiber optic cables, their laborious laying and installation. Other advantages of FSO communication

are security from intruders, immunity from electromagnetic interference by nearby networks, virtual non-existence of frequency allocation and planning issues, smaller power consumption by FSO based terminals and relatively smaller dimensions of the same.

Last but not the least it is the solution to last mile problem (insufficient bandwidth for end user arising from lack of higher bandwidth access networks) which has plagued many consumers with low data speeds. With the deployment of FSO the concept of fiber-to-home has been realized. In a typical scenario scientists are promising FSO deployments up to two to three miles in length in near future with data rates of the order of hundreds of gigabits per second or higher using Wavelength Division Multiplexing (WDM).

1.1.1 Applications

The applications of FSO systems are manifold. They have been chiefly deployed by military and are finding niche applications in services and commercial sectors as well. Depending on the needs of the users and dedicated application scenarios they can be deployed on fixed or mobile platforms, can be used for interstellar communications, ship to ship or ship to ground communications, ground to spacecraft, spacecraft to spacecraft and high altitude platforms (HAP), and between elements of satellite constellations. In addition to stand-alone usage of FSO for communication for communication between airborne terminals, satellites and terrestrial links, it is used as a complementary communication technology to aid and reinforce existing RF or microwave based networks. The typical scenario could be back-haul traffic or conventional networks where FSO could find diverse applications for high speed data transfer. Thus it is very likely that these hybrid FSO-RF and FSO-microwave networks would gain popularity in an age where users demand more and more bandwidth for advanced multimedia applications. Last but not the least FSO systems are a very

important factor in NASA's deep space communication. A good description of FSO applications is given in [1].

1.2 Thesis Motivation

FSO systems require some kind of optical beam tracking/sensing mechanism whereby the position of the beam center can be tracked in real time. These systems transmit highly directional and narrow beams of light for data communication. A typical transmitter transmits one ore more light beams of diameter 5-8 cm which flares out to 1-5 m within a distance of 1 kilometer. Moreover the problem to receive this narrow beam becomes more challenging due to the receiver's limited Field of View (FOV). It is not uncommon for an optical transmitter-receiver system to lose alignment for the following reasons:

1. Relative motion between the transmitter and receiver as in satellite communications.

2. Atmospheric effects e.g. turbulence might cause the beam to wander away temporarily.

3. Mechanical vibrations.

4. Thermal expansion and wind sways (building sways where the transmitter or receivers are mounted also referred to "base motion").

In addition the angular motion (azimuth and elevation) causes more degradation in optical communication than linear motion between transmitter and receiver. Base motion can be further subdivided into the following three categories:

1. **Low Frequency**: This type of relative motion between transmitter and receiver has periods from minutes to months and is caused by diurnal or seasonal temperature variations. The greater the height of the building or rooftop the more pronounced would be the effect.

2. **Moderate Frequency**: This type is caused by wind and again is more pronounced in tall buildings. Its periods lasts within seconds.

3. **High Frequency**: This type of motion has period less than a second and is caused by the mechanical vibrations from large machines installed in the building.

In this thesis the performance of three estimators in accurately tracking the beam position is formulated. Incident beam of light is assumed to have a Gaussian shape and the photo emissions on detector array are assumed to follow a non-homogeneous Poisson process with a Gaussian intensity function. These estimators are:

1. **Center of Gravity or Centroid**: Arithmetic average of the position of photons.

2. **Template Matching**: Convolution of Intensity function with Dirac Delta functions present at position of the photons and finding the maximum of the resulting function gives the template matching estimate.

3. **Maximum Likelihood**: Computing the maximum of log likelihood function given the position of photons gives maximum likelihood estimate.

The criterion to judge the performance of these estimators is the mean square error. The above mentioned list is formed in order of complexity of these estimators: centroid has the least computational complexity followed by template matching and the maximum likelihood is the most computationally intensive. As shown in later chapters the results of the simulation indicate that the maximum likelihood estimate has the smallest mean square error of all the three estimators.

In the next chapter the simulation of non-homogeneous Poisson process modeling the intensity function of the beam and photo emissions on detector array is discussed.

2. GENERATION OF TWO DIMENSIONAL NONHOMOGENEOUS POISSON PROCESS

2.1 Introduction

The following two factors are noted about a two dimensional non-homogeneous Poisson Process with rate function $\lambda(x, y) > 0$ on $C \subset \mathbb{R}^2$

(a) If $\{(X_i, Y_i)\}$ are events corresponding to the 2-D process, then the abscissa X_i correspond to a 1-D non-homogeneous Poisson Process with rate function $\lambda_x(x) = \int_{C(x)} \lambda(x, y)\, dy$, where $C(x) = \{y : (x, y) \in C\}$

(b) If (x, y) denotes the location of an event from the 2-D process, the conditional random variable $Y | X = x$ has the probability density function $\lambda(x, y)/\lambda_x(x)$.

Thus the algorithm for generating points for a 2-D non-homogeneous Poisson Process is as follows:

(a) Initialize $i = 0$.

(b) Generate x_i according to the 1-D non-homogeneous Poisson Process with rate function $\lambda_x(x)$.

(c) Generate y_i according to probability density function $\lambda(x_i, y_i)/\lambda_x(x_i)$.

(d) Deliver (x_i, y_i).

(e) Set $i = i + 1$ and go to step 2.

To carry out step (b), we generate a 1-D non-homogeneous Poisson Process with rate $\lambda_x(x)$ as follows in the following section.

2.2 Generation of Non-homogeneous Poisson Process with rate $\lambda_x(x)$

(a) Initialize $x = a, n = 0, \lambda = \max_{x\in[a,b]} \lambda_x(x)$;

(b) Set $x = x - \ln(\text{Uni}(0,1))/\lambda$, if $x > b$ stop;

(c) If $\text{Uni}(0,1) \leq \lambda_x(x)/\lambda$, set $n = n+1, S_n = x$;

(d) Go to step (b)

Output n is the number of events in (a,b) and event times are $S_1, \ldots, S_n$.

In our case

$$\lambda(x,y) = I_0 e^{\frac{-(x-\alpha)^2-(y-\beta)^2}{2\rho^2}}, \tag{2.1}$$

$$\begin{aligned} \lambda_x(x) &= \int_{C(x)} \lambda(x,y)\, dy \\ &= \int_a^b I_0 e^{\frac{-(x-\alpha)^2}{2\rho^2}} e^{\frac{-(y-\beta)^2}{2\rho^2}}\, dy \\ &= I_0 e^{\frac{-(x-\alpha)^2}{2\rho^2}} \int_a^b I_0 e^{\frac{-(y-\beta)^2}{2\rho^2}}\, dy, \end{aligned} \tag{2.2}$$

and

$$\begin{aligned} f_y(y) &= \frac{\lambda(x_i,y_i)}{\lambda_x(x_i)} \\ &= \frac{I_0 e^{\frac{-(x-\alpha)^2}{2\rho^2}} e^{\frac{-(y-\beta)^2}{2\rho^2}}}{I_0 e^{\frac{-(x-\alpha)^2}{2\rho^2}} \int_a^b I_0 e^{\frac{-(y-\beta)^2}{2\rho^2}}\, dy} \\ &= \frac{e^{-\frac{(y-\beta)^2}{2\rho^2}}}{\int_a^b e^{-\frac{(y-\beta)^2}{2\rho^2}}\, dy} \\ &= M e^{-\frac{(y-\beta)^2}{2\rho^2}}, \end{aligned} \tag{2.3}$$

where $M = \left(\int_a^b e^{-\frac{(y-\beta)^2}{2\rho^2}}\, dy \right)^{-1}$.

So we first generate Poisson points with intensity function $\lambda_x(x)$ in x-direction and then sample the same number of points from $f_y(y)$ independent of $\lambda_x(x)$ along the y-direction.

2.2.1 Generation of sample points from $f_y(x)$

Lets assume that we want to simulate a value W from a given density function $f_y(y)$ which satisfies the following conditions:

1. $f_y(x) > 0$ only for x in the interval $a \leq x \leq b$.
2. $f_y(x) \leq M$ for some M.

We use Rejection Method for simulation as follows:

(a) Generate a Uniform random variable on $[a, b]$; call it X.

(b) Independently generate a Uniform value on $[0, M]$; call it Y.

(c) $\begin{cases} \text{Accept the point } (X, Y) \text{ if } Y \leq f_y(x); \text{ the simulated value then is } W = X, \text{ or,} \\ \text{Reject the point } (X, Y) \text{ if } Y > f_y(X) \end{cases}$

(d) Repeat the first three steps again until the required number of points are obtained.

The above procedure to generate points from an arbitrary distribution will work correctly as explained below:

Probabilistic background

1. The density of X is $1/(b-a)$ for $a \leq x \leq b$.
2. The density of Y is $1/M$ for $0 \leq y \leq M$.
3. The joint density of X and Y is $f(x, y) = 1/M(b-a) \cdot 1_{[a,b]}(x) \cdot 1_{[0,M]}(y)$.

4. Note that

$$\begin{aligned} P(c \le W \le d) &= P(c \le X \le d | Y \le f_y(x)) \\ &= \frac{P(c \le X \le d, Y \le f_y(x))}{P(Y \le f_y(x))} \\ &= \frac{\int_c^d \int_0^{f_y(x)} \frac{1}{M(b-a)} \, dy \, dx}{\int_a^b \int_0^{f_y(x)} \frac{1}{M(b-a)} \, dy \, dx} \\ &= \frac{\int_c^d \int_0^{f_y(x)} \, dy \, dx}{\int_a^b \int_0^{f_y(x)} \, dy \, dx} \\ &= \frac{\int_c^d f_y(x) \, dx}{\int_a^b f_y(x) \, dx} \\ &= \int_c^d f_y(x) \, dx. \end{aligned} \tag{2.4}$$

Hence this shows that sample W is generated from the probability density function $f_y(x)$.

2.3 Alternative way to generate nonhomogeneous Poisson Process

For generating a non-homogeneous Poisson process in a bounded region C, a simple algorithm is based on the intuitive idea that the position (X, Y) of an event given that the event has occurred is distributed according to the following probability density function

$$f(x, y) = \frac{\lambda(x, y)}{\iint\limits_C \lambda(x, y) \, dx \, dy}, (x, y) \in C. \tag{2.5}$$

It is also noted that the number of Poisson events in C are distributed with mean $\iint\limits_C \lambda(x, y) \, dx \, dy$ and thus the following conceptually simpler algorithm results as follows for simulation of the process.

1. Generate $n \sim$ Poisson $(\iint\limits_C \lambda(x, y) \, dx \, dy)$.
2. If $n = 0$, then exit. Otherwise independently generate n events $(x_i, y_i), i = 1, \ldots, n$ in C according to density function $f(x, y)$ given in (2.5).

3. Deliver $(x_i, y_i), i = 1, \ldots, n$.

2.3.1 Derivation of $f(x, y)$

We know that for a Gaussian beam

$$\lambda(x, y) = I_0 \exp\left\{\frac{-(x-\alpha)^2 - (y-\beta)^2}{2\rho^2}\right\}. \tag{2.6}$$

Substituting this value of λ in (2.6), we get

$$f(x, y) = \frac{\lambda(x, y)}{\int_a^b \int_a^b \lambda(x, y)\, dx\, dy} = \frac{\lambda(x, y)}{\lambda_0}, \tag{2.7}$$

where the region of interest C is bounded by values a and b in each direction of 2-D plane and $\lambda_0 = \int_a^b \int_a^b \lambda(x, y)\, dx\, dy$.

Substituting the value of λ_0 in (2.7) gives

$$\begin{aligned} f(x, y) &= \frac{I_0 e^{\frac{-(x-\alpha)^2}{2\rho^2}} e^{\frac{-(y-\beta)^2}{2\rho^2}}}{\lambda_0} \cdot 1_{[a,b]}(x) \cdot 1_{[a,b]}(y) \\ &= \frac{\sqrt{I_0} e^{\frac{-(x-\alpha)^2}{2\rho^2}}}{\sqrt{I_0}\int_a^b e^{\frac{-(x-\alpha)^2}{2\rho^2}}\, dx} \cdot 1_{[a,b]}(x) \frac{\sqrt{I_0} e^{\frac{-(y-\beta)^2}{2\rho^2}}}{\sqrt{I_0}\int_a^b e^{\frac{-(y-\beta)^2}{2\rho^2}}\, dy} \cdot 1_{[a,b]}(y) \\ &= f_x(x) \cdot f_y(y), \end{aligned} \tag{2.8}$$

where

$$f_x(x) = \frac{\sqrt{I_0} e^{\frac{-(x-\alpha)^2}{2\rho^2}}}{\sqrt{I_0}\int_a^b e^{\frac{-(x-\alpha)^2}{2\rho^2}}\, dx} \cdot 1_{[a,b]}(x)$$

and

$$f_y(y) = \frac{\sqrt{I_0} e^{\frac{-(y-\beta)^2}{2\rho^2}}}{\sqrt{I_0}\int_a^b e^{\frac{-(y-\beta)^2}{2\rho^2}}\, dy} \cdot 1_{[a,b]}(y).$$

To simulate this process we simply generate n points, each from the density functions $f_x(x)$ and $f_y(y)$ independently of each other according to the procedure laid out in section 2.2.1.

3. IMPLEMENTATION OF CENTROID, TEMPLATE MATCHING AND MAXIMUM LIKELIHOOD ESTIMATION METHODS

3.1 Introduction

As discussed earlier an optical communication system has a requirement for active tracking of the beam position in order to ensure that the receiver looks as directly as possible at the transmitter. This mechanism requires some kind of a sensor that could detect or estimate the position of the beam arriving at the receiver photocell array. The sensor output then is used as a feedback signal to control the positions of telescopes and mirrors of the receiver so that the receiver-transmitter alignment is maintained even thought the transmitter might be in relative motion to the receiver. A simple device to sense the light beam position could be a photo detector having a photo emissive surface that has a number of built in cells (or cell array).

Photo electron emissions are monitored in different cells of the cell array during a time interval $[0, T)$ and that information is used to sense the position of the incoming beam. It is assumed that the intensity of the light beam has a two dimensional circularly symmetric Gaussian shape, given by the following equation:

$$I(a, b : \alpha, \beta) = I_0 \exp\left(-\frac{1}{2\rho^2}[(a-\alpha)^2 + (b-\beta)^2]\right) \tag{3.1}$$

where α and β are beam position parameters(coordinates of the center of the beam) which are unknown and desired. I_0 is the peak of the intensity function and ρ is beam width parameter, and these values are known. It is thus assumed that photo-emissions on the surface of cell array are modeled by a non-homogeneous Poisson process governed by intensity function given in (3.1).

In this chapter the implementation of Center of Gravity, Template Matching and Maximum Likelihood estimation methods for estimating the optical beam's position would be discussed.

3.2 Center of Gravity

The Center of Gravity estimate of the beam position is given by

$$(\hat{\alpha}, \hat{\beta}) = \left\{ \frac{1}{N(T)} \sum_{i=1}^{N(T)} a_i, \frac{1}{N(T)} \sum_{i=1}^{N(T)} b_i \right\} \tag{3.2}$$

where $N(T)$ is the number of photo-emissions occurring in $[0, T)$ and (a_i, b_i) for $i = 1, \ldots, N(T)$ are their positions. Thus the estimated position is just the arithmetic average of the positions of observed photo-events (or the centroid if each point is assumed to possess unit mass).

3.3 Template Matching

The Template Matching estimate of the beam position is given by the following mathematical expression:

$$\mathcal{T}(\alpha, \beta) = \sum_{i=1}^{N(T)} I(a_i, b_i : \alpha, \beta) \tag{3.3}$$

where $I(a_i, b_i : \alpha, \beta)$ is the intensity function given in equation (3.1), (a_i, bi_i) are the observed data points and $N(T)$ is the data count in time interval $[0, T)$. The Template Matching estimate is given by those values of α and β that maximize equation (3.3). In other words this method comprises finding the convolution of the intensity function with impulse functions present at the location of data points over the region of interest in 2-D plane and computing that value of α and β that maximizes the convolution sum.

3.4 Maximum Likelihood Estimation

The log likelihood function for beam position estimate is derived in [2]. The final expression is as follows:

$$\mathcal{L}(\alpha, \beta) = \int_{-1}^{1} \int_{-1}^{1} I(a, b : \alpha, \beta)\, da\, db + \sum_{i=1}^{N(T)} \ln\, I(a_i, b_i : \alpha, \beta). \tag{3.4}$$

Here $N(T), I(a, b : \alpha, \beta)$, and (a_i, b_i) are defined as before for Template Matching estimate. The Maximum Likelihood estimate of the beam position is given by α and β that maximize equation (3.4).

3.5 Computing the maximum of template matching and maximum likelihood methods

One can use various methods to find the maximum of a function over a given bounded region. One can resort to iterative maximization procedures, for example, Newton-Raphson method, the scoring approach or expectation maximization algorithm. The advantage these methods offer are less computation complexity, however, it may be that these methods converge to a local maximum instead of a global maximum. There is a good chance that these methods would reach the global maximum if the initial guess is close to the true maximum.

If computational complexity is not an issue then the 'safest' method is to use grid search algorithms. One class of such algorithms is the *Genetic Algorithms* which converge to the global maximum within a finite number of iterations, especially for two dimensional grid searches. These algorithms are discussed in the next chapter.

4. GENETIC ALGORITHMS

4.1 Introduction

A Genetic Algorithm is a randomized, derivative free, population-based search technique that has its roots in principles of genetics used to find the global maximum of a function over the region of interest. One important distinction between a genetic algorithm and other search techniques is that the genetic algorithm searches for the maximum from a set of points rather than a single point at each iteration. Only a brief explanation of genetic algorithms is discussed in this chapter. For details, interested readers should refer to [3].

4.2 Basic Description

Suppose we wish to solve the following optimization problem

$$\text{maximize } f(\boldsymbol{x})$$

$$\text{subject to } \boldsymbol{x} \in \Omega$$

The underlying idea of genetic algorithms is as follows: We begin with an initial set of points denoted by $P(0)$ which we call *initial population*. Objective function is then evaluated on this initial population, and, based on this evaluation, a new generation of points $P(1)$ is created. The creation of $P(1)$ involves certain operations which we call *crossover* and *mutation*. The new populations (or generations) are generated until a stopping criterion is achieved.

4.3 Real-Number Genetic Algorithms

We have used Real-number genetic algorithm to find the global maximum of the objective function which operates directly on points (also called chromosomes) in feasible set Ω (2-D array in our case). Each point has a value associated it with called *fitness*, which is the value of the objective function at that point.

The next stage entails initializing the first population $P(0)$. This is normally done by random selection of points from Ω. After the initial population is formed the operations of crossover and mutation are applied. During each iteration k of the process the fitness $f(\boldsymbol{x}^{(k)})$ of each point $\boldsymbol{x}^{(k)}$ is evaluated. After the fitness of entire population $P(k)$ is evaluated we form new population $P(k+1)$.

4.4 Selection and Evolution

In the first stage of forming $P(k+1)$ we apply *selection*, where we form a set $M(k)$ with same number of elements (population size N) as $P(k)$. Set $M(k)$ is called the *mating pool*, and is formed from $P(k)$ using a random procedure as follows: Each point $\boldsymbol{x}^{(k)}$ in $M(k)$ is equal to $\boldsymbol{x}^{(k)}$ in $P(k)$ with probability

$$\frac{f(\boldsymbol{x}^{(k)})}{F(k)},$$

where

$$F(k) = \sum f(\boldsymbol{x}_i^{(k)})$$

and the sum is taken over whole of $P(k)$. In other words, the greater the fitness of a chromosome, the larger the probability of it making it to the mating pool.

The second stage is called *evolution* (it involves both crossover and mutation). In evolution, crossover is operated upon points in mating pool according to a crossover probability to create offspring chromosomes from parents. For crossover we have used random convex combination of the parents to produce offspring because Ω is a convex set and the resulting offspring is also within Ω. That offspring then can replace

one of the parents. For mutation we replace, with mutation probability, the chosen chromosome with a random convex combination of the chromosome with a random point in feasible set. For further details about crossover and mutation, the reader is again referred to chapter 8 of [3].

After applying crossover and mutation operations we obtain the new population $P(k+1)$.

4.5 Summary of Genetic Algorithm, Elitism and stopping criterion

1. Set $k := 0$. Generate an initial population $P(0)$ at random.
2. Evaluate $P(k)$.
3. If the stopping criterion is satisfied, then stop.
4. Choose $M(k)$ from $P(k)$.
5. Apply Crossover and Mutation on $M(k)$ to obtain $P(k+1)$.
6. Set $k := k+1$ and go to step 2.

4.5.1 Elitism

Also in genetic algorithm we keep track of the 'best' two chromosomes: the chromosomes having the highest fitness and second highest fitness. These two best chromosomes are then passed onto the next population of chromosomes, a practice known as *elitism*. Intuitively also it makes sense to never lose the chromosomes with best fitness at any stage of the algorithm if a global maximum is desired.

4.5.2 Stopping criterion

A reasonable stopping criterion could be to stop the algorithm after a preset number of iterations is achieved. Another criterion could be to stop after the difference in

the fitness of the ‘elitist’ chromosomes stays within some specified small limit from iteration to iteration.

5. SIMULATION

5.1 Introduction

In this chapter the details of simulation for computing mean square error (MSE) for each of the three estimators, Center of Gravity, Template Matching and Maximum Likelihood are discussed.

5.2 Details of Simulation

Matlab has been used to carry out the simulation of two dimensional Optical Position Sensing process. Non-homogeneous Poisson is generated on Matlab as discussed in detail in chapter 2. The values of α and β are generated according to the following two rules:

1. **Non random parameters**: An arbitrary value of the center of the beam is chosen (α, β) to generate a non-homogeneous Poisson process, and that value is kept constant for the given number of iterations.

2. **Random parameters**: A random value for center of beam is chosen from a two dimensional Gaussian distribution with mean zero and given variance for every iteration of the simulation.

The number of iterations is selected to be 200. For each iteration a non-homogeneous Poisson process is generated with given values of I_0, width parameter ρ and center of beam (α, β) on an array of size two-by-two, and the squared error is calculated for each of the three estimators. At the end of 200 iterations the mean is computed for all three squared errors.

5.3 Factors affecting Position Sensing

As previously stated the optical beam (signal) from the transmitter generates photons on the photoemissive array (the receiver) to communicate data in an FSO system. The beam centers position sensing problem involves using the observations (points or photon positions generated using non-homogeneous Poisson process) to estimate the center in order to track the position of the beam. This position sensing problem gets more complicated due to the following two additional factors.

(a) Noise process: Background radiation and dark current

(b) Quantization of photoemissive array

We discuss these two factors briefly as follows:

5.3.1 Noise event process

Noise causes significant degradation in the overall performance for each of the three estimators we discussed. Sources giving rise to noise photons in the array are *background radiation* and *dark current*. Background radiation comprises light from sky, sun, stars and other light sources that pass through the optical system; as a result of which randomly dispersed photo-emissions occur within the array. Field stops can be used in the receiver to minimize the effect of background radiation but the diffuse light still causes photo-conversions with a Poisson intensity function which is modeled as uniform or constant throughout the dimensions of detector array.

Dark current is the *zero input* response of receiver. Even in the absence of signal there would still be photo-emissions in the detector array due to thermal effect. These photons are modeled as being generated by an independent Poisson intensity function which is again assumed to be uniform like the background radiation. Therefore the resultant noise process is modeled as one Poisson process with uniform intensity function λ_n in which the two noise factors are lumped together.

5.3.2 Quantization of detector array

We know that the square detector consists of cells of array which convert light energy into electrical energy. Normally the arrays used are either 4-by-4 (16 cells) or 8-by-8 (64 cells), and the position of a given photon is given by the position of the photocell in which it occurs. Thus the positions of photons are not exact but are rounded to center of the given cell. This quantization effect adds degradation to the position sensing performance as the original positions of the photons are not used as observations in the parameter estimation. If we let the limit of cells or partitions go to infinity the resulting array would become continuous which would give the exact value of photon positions. The performance of a continuous versus the quantized array would be analyzed. Intuitively the higher the number of partitions, the better would be the performance of estimators.

5.4 Log-likelihood function

5.4.1 Unknown or non random parameter

The log likelihood equation for the unknown parameter is given as

$$\mathcal{L}(\alpha,\beta) = \int_{-1}^{1}\int_{-1}^{1} I(a,b:\alpha,\beta)\,da\,db + \sum_{i=1}^{N(T)} \ln\left(I(a_i,b_i:\alpha,\beta)\right). \tag{5.1}$$

The value that maximizes (5.1) is the maximum likelihood estimate of the beam center.

It can be easily verified that when the width parameter ρ and range of possible position becomes smaller relative to the dimensions of the array, the maximum likelihood estimate converges to the center of gravity estimate as given by [2] since the first term in (5.1) would become constant relative to α and β and hence can be ignored. Taking the derivative of second term with respect to α and β and setting that equal to zero yields

$$(\hat{\alpha}_{ML}, \hat{\beta}_{ML}) = \left(\frac{1}{N(T)} \sum_{i=1}^{N(T)} a_i,\ \frac{1}{N(T)} \sum_{i=1}^{N(T)} b_i \right), \tag{5.2}$$

which is the center of gravity estimate.

5.4.2 Random parameter

The *a priori* distribution of the parameter $\boldsymbol{x} = \begin{bmatrix} \alpha & \beta \end{bmatrix}^T$ is assumed to be circularly symmetric two dimensional Gaussian distribution given as follows:

$$p(\boldsymbol{x}) = \frac{1}{2\pi\sigma^2} \exp\left(\boldsymbol{x}^T C^{-1} \boldsymbol{x}\right), \tag{5.3}$$

where

$$C = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix},$$

and σ^2 is the variance of distribution.

The log-likelihood equation for random parameter case is

$$l(\alpha, \beta) = \int_{-1}^{1} \int_{-1}^{1} I(a, b : \alpha, \beta)\, da\, db + \sum_{i=1}^{N(T)} \ln\left(I(a_i, b_i : \alpha, \beta)\right) + \ln\, p(\alpha, \beta). \tag{5.4}$$

The values of (α, β) that maximize (5.4) is the *maximum a posteriori (MAP)* estimate of the beam's center. According to [4] (5.4) can be reduced to the following estimate for the case of zero noise in the system $\lambda_n = 0$

$$\hat{x} = \frac{\sum_{i=1}^{N(T)} r_i}{N(T) + (\rho/\sigma)^2}, \tag{5.5}$$

where $r_i = (a_i, b_i)$. (5.5) reduces to center of gravity estimate when $(\rho/\sigma) \to 0$.

5.5 Simulation details

In this section general details regarding simulation would be laid out. The whole simulation can be broken down into two major steps:

1. **Simulation of two dimensional non-homogeneous Poisson process**: Photo-emissions for both the signal and noise are modeled as being generated by independent non-homogeneous Poisson processes. For signal photons equation (3.1) is used as the intensity function and for noise emissions a constant value of λ_n is used. The values of α, β, peak intensity I_0 and width parameter ρ are given.

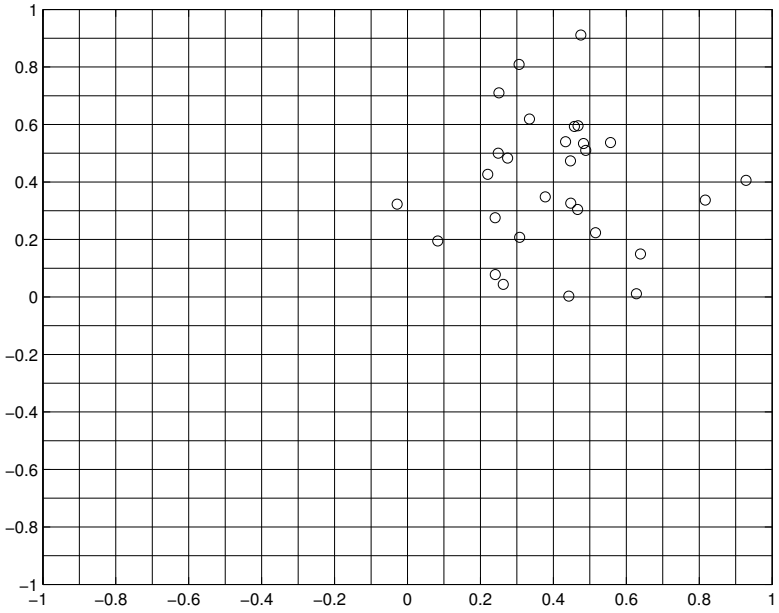

Fig. 5.1. A realization of 2-D non-homogeneous Poisson process with parameters $I_0 = 130$, $\rho = 0.2$, $(\alpha, \beta) = (0.4, 0.4)$.

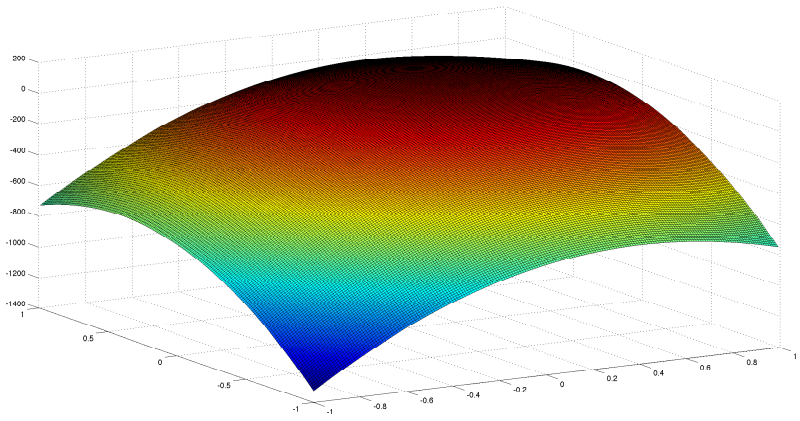

Fig. 5.2. Log likelihood function for the realization of Poisson process showing in figure 5.1.

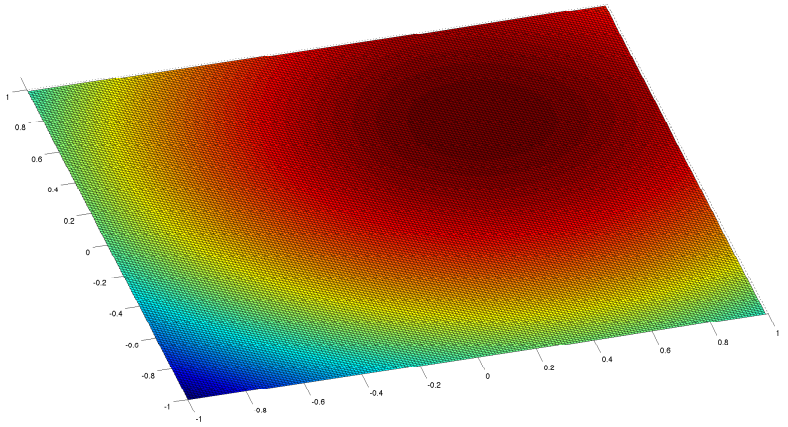

Fig. 5.3. Contour plot for the Log likelihood of Poisson process shown in figure 5.2.

Figures 5.1, 5.2, and 5.3 shows a typical realization of the non-homogeneous Poisson process for a given set of parameter values and its corresponding log likelihood function.

2. **Estimators**: Equations (3.2), (3.3) and (3.4) are implemented for center of gravity, template matching and maximum likelihood estimates by using the photon positions on 2-D surface generated in step 1. Then the square error for each of the three estimators is calculated.

3. **Iteration**: The number of iterations are 200. After the end of two hundredth iteration mean of square error is computed.

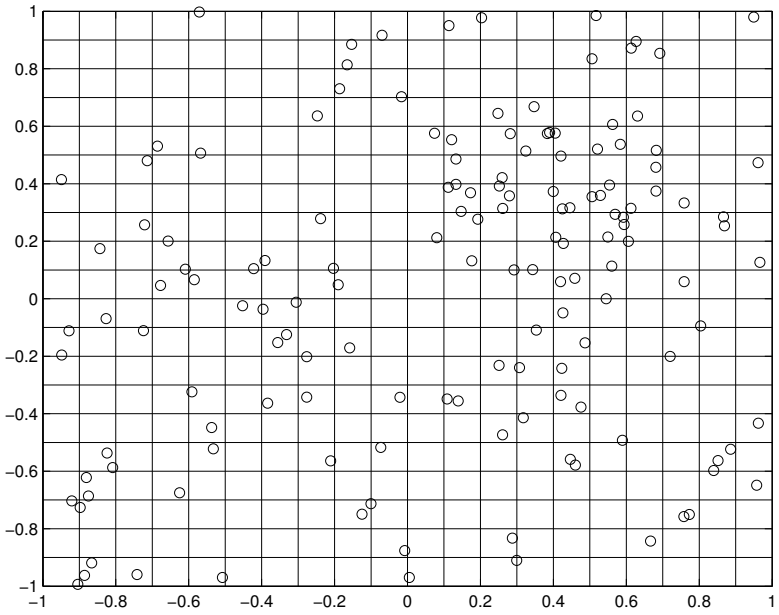

Fig. 5.4. Realization of Poisson process with dark current photoemissions. $I_0 = 130, \rho = 0.2, (\alpha, \beta) = (0.4, 0.4), \lambda_n = 30$.

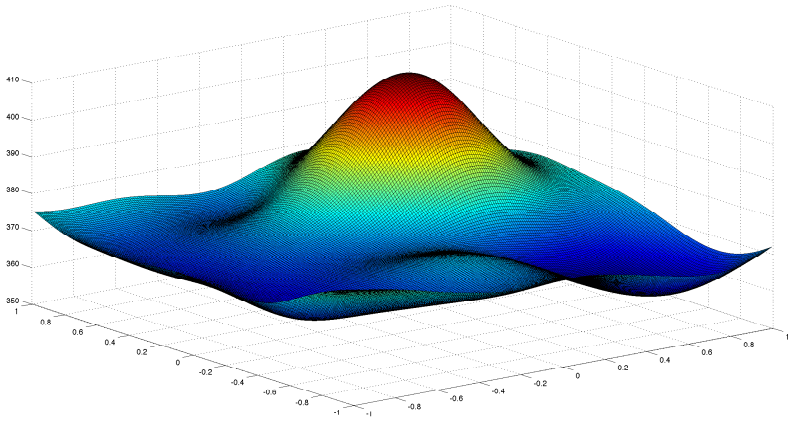

Fig. 5.5. Log likelihood of Poisson process showing in figure 5.4.

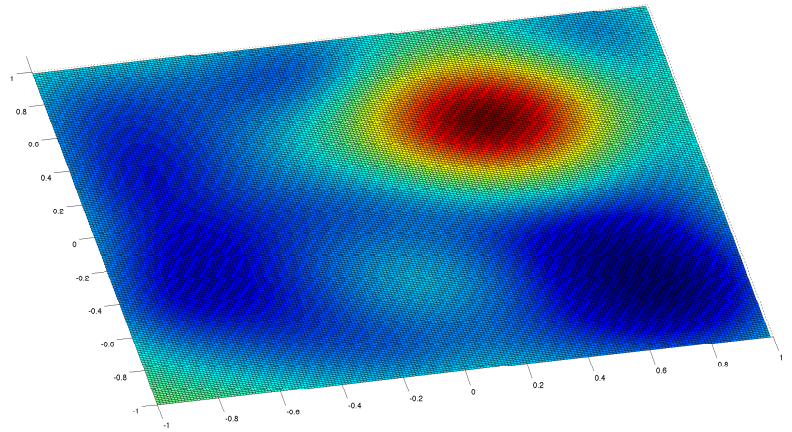

Fig. 5.6. Contour plot for the Log likelihood of Poisson process showing in figure 5.5.

4. **Noise**: We increase the noise parameter λ_n by a fixed value and repeat steps 1, 2 and 3 again until we have data for the whole range of λ_n. Figures 5.4, 5.5 and 5.6 depict a typical realization of signal and noise photons generated by the Poisson process along with the log likelihood function.

5. **Plots**: Plot the mean squared error versus λ_n.

5.5.1 Parameter values for simulation

Throughout the simulation the values of ρ and σ are chosen small enough relative to array dimensions to avoid any "edge effects". Edge effects occur when the beam of light impinging on the array is too wide and a portion of the beam "falls off" the edge of the array. In that case it is not possible to evaluate the performance of the estimators as edge effects result in adding bias to the estimator values. Throughout the simulation the value of ρ is kept within the range $[0.05, 0.2]$ and σ within $[0.1, 0.3]$.

5.5.1.1 Values of $\boldsymbol{\alpha, \beta}$

To generate step 1 of simulation we consider both cases of parameter estimation: random as well as nonrandom parameters. To consider the random parameter first we choose the values of (α, β) from a two dimensional Gaussian distribution for each iteration. For nonrandom or unknown parameter case we assign arbitrary values to both α and β and keep those values same for every run of 200 iterations.

5.5.1.2 Values of $\boldsymbol{I_0}$, $\boldsymbol{\rho}$ and $\boldsymbol{\lambda_n}$

Values of peak intensity I_0 is chosen within the range of 100 and 2000 and beam width parameter ρ between 0.05 to 0.2 depending upon what value of signal power is desired. Simulations are done for different values of I_0 and ρ to assess the performance of estimators for those parameter values. In this simulation λ_n is varied from 0.5 to 30.5 in steps of 5. For typical FSO systems its not uncommon to see on average about four to six signal photons and a similar number of noise photons (distributed uniformly over the array). That scenario is also simulated by using I_0 equal to 100 and ρ to 0.1 which gives about six signal photons on average for every iteration. Since

the size of array is 2-by-2, using noise parameter λ_n in the range $[0.5, 3]$ gives the number of average noise photons in the range $[2, 12]$.

5.5.1.3 Quantization

The simulation would be done for 4-by-4, 8-by-8 and 16-by-16 quantized array and the performance of the estimators would then be compared to that obtained for continuous array.

5.5.1.4 Genetic Algorithm

The genetic algorithm is used to search the peak of functions in the range $[-1, 1]$ for both x and y axes (dimensions of detector array) for template matching and maximum likelihood estimators. The number of chromosomes is set to 30 and iterations at 100. Mutation rate is set equal to 0.01 and Crossover rate at 0.75. It has been proved in [5] that Genetic Algorithm will converge if Elitism (passing on chromosomes with the best fitness value to next generation) is employed.

6. RESULTS OF SIMULATION

In this chapter the performance comparison in terms of mean squared error for each of the three estimators (Center of Gravity, Template Matching and Maximum Likelihood) is carried out based on the results of the simulation, the details of which are laid out in the last chapter.

6.1 Continuous array

6.1.1 Unknown deterministic parameter

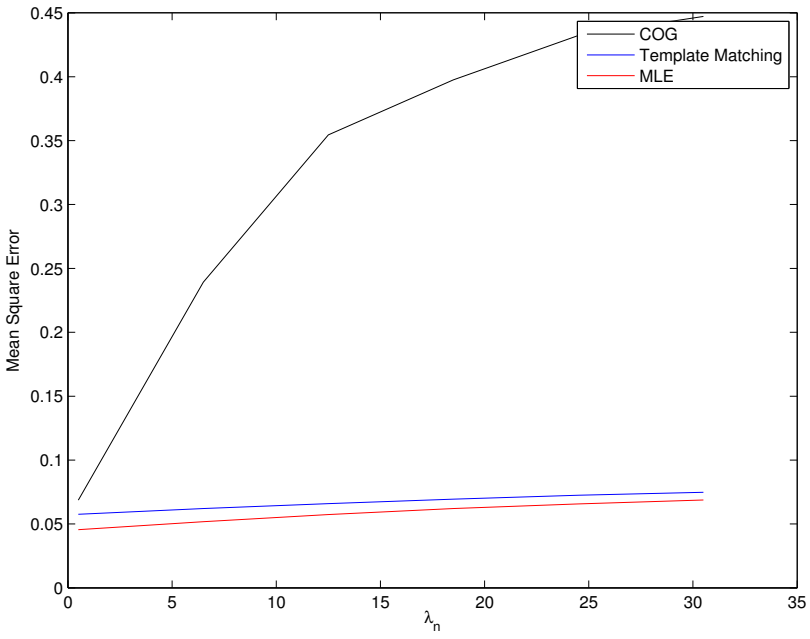

Fig. 6.1. Mean Square Error for $I_0 = 130$, $\rho = 0.2$, $(\alpha, \beta) = (0.4, 0.4)$, continuous array.

Figure 6.1 shows the results of simulation for random parameter case. The value of I_0 and beam spread ρ are set equal to 130 and 0.2 respectively. α and β are set equal to an arbitrary $(0.4, 0.4)$ and the noise parameter λ_n is varied from 0.5 to 30.5 in steps of 5. In the rest of the simulations the signal power is kept constant, noise is gradually added and the behavior of the estimators observed during a varying signal-to-noise scenario.

It can be seen that under high signal-to-noise ratio the performance of the three estimators are identical but as noise power increases the error performance of center of gravity estimate goes down considerably. This is because as the noise power is increased the center of gravity of all the photons would begin to shift from (0.4, 0.4) to the center of array (0, 0) since the noise parameter distributes noise photons uniformly on the array. The other two estimators perform reasonably well even under high noise conditions. However the maximum likelihood estimate gives the best performance.

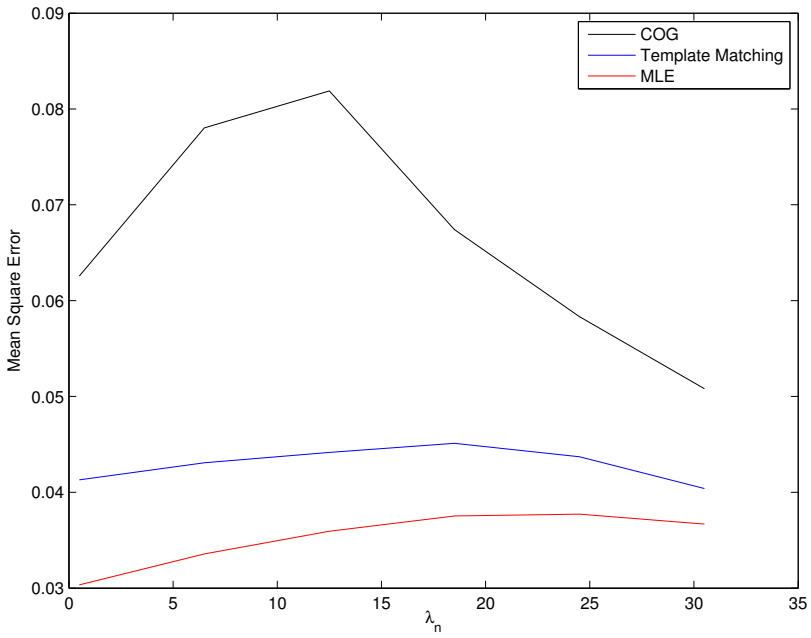

Fig. 6.2. Mean Square Error for $I_0 = 130$, $\rho = 0.2$, $(\alpha, \beta) = (0, 0)$, continuous array.

Figure 6.2 shows the simulation results for the case ($\alpha = 0, \beta = 0$). It must be noted that since we use a continuous array and ρ is sufficiently small as compared to array dimensions the edge effects are minimal and thus all estimators are assumed to be unbiased. Therefore we expect all the estimators to behave the same no matter what the position of beam center is. However under high noise conditions the center of gravity would always shift to the center of array and there would be bias in its performance. That phenomenon can actually be seen in this figure as we see the error performance for center of gravity estimate improves as noise power is increased since the center of the beam is (0, 0) to begin with.

6.1.2 Random parameter

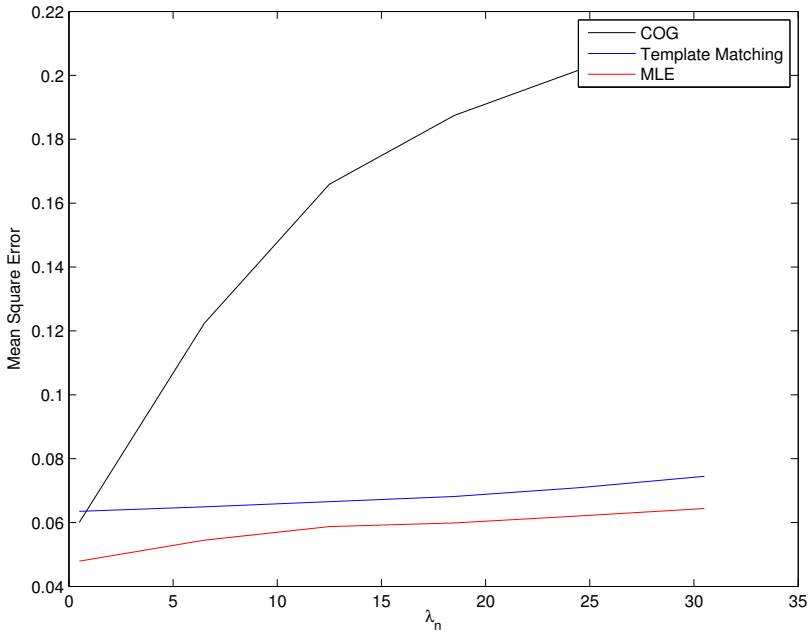

Fig. 6.3. Mean Square Error $I_0 = 130$, $\rho = 0.2$, $\sigma = 0.2$, continuous array.

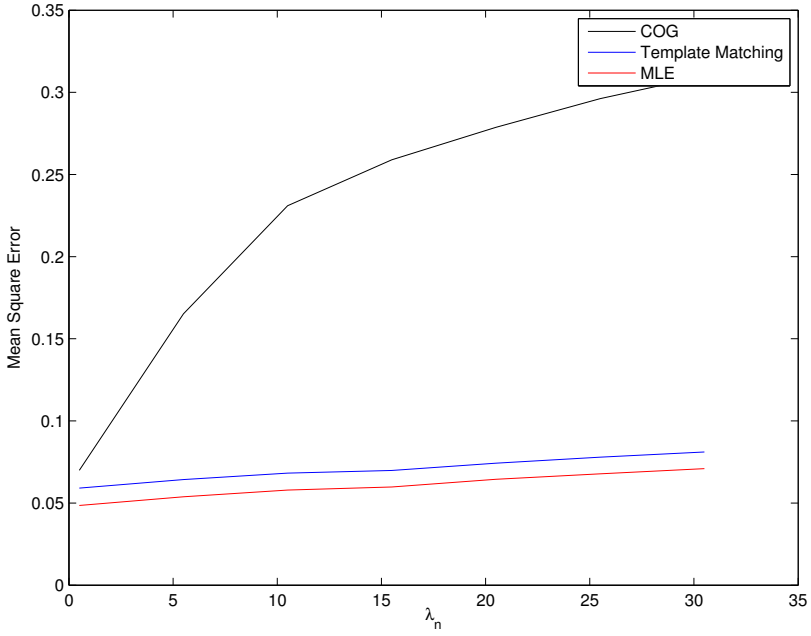

Fig. 6.4. Mean Square Error $I_0 = 130$, $\rho = 0.2$, $\sigma = 0.3$, continuous array.

For random parameter case the values of I_0 and ρ are set the same as before but now the center of beam is chosen at random from a two dimensional Gaussian distribution with mean zero and variance $\sigma^2 = 0.2$ for every iteration. Figures 6.3, 6.4 and shows the performance of the three estimators for random parameter case for σ^2 equal to 0.2 and 0.3 respectively. Maximum likelihood estimate gives best results.

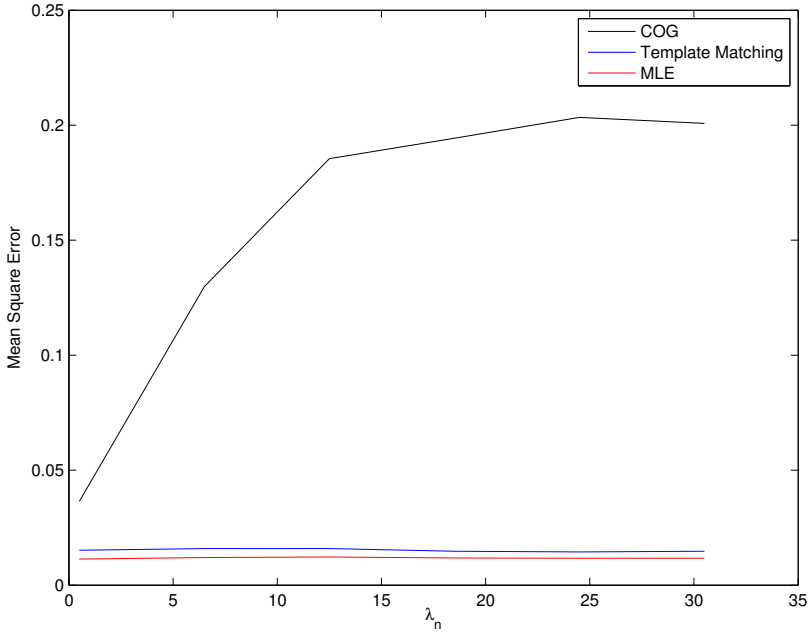

Fig. 6.5. Mean Square Error $I_0 = 2000$, $\rho = 0.05$, $\sigma = 0.2$, continuous array.

In figure 6.5 the simulation is repeated for $I_0 = 2000$ and $\rho = 0.05$ with value of σ set at 0.2.

6.2 Quantized array

6.2.1 Unknown deterministic parameter

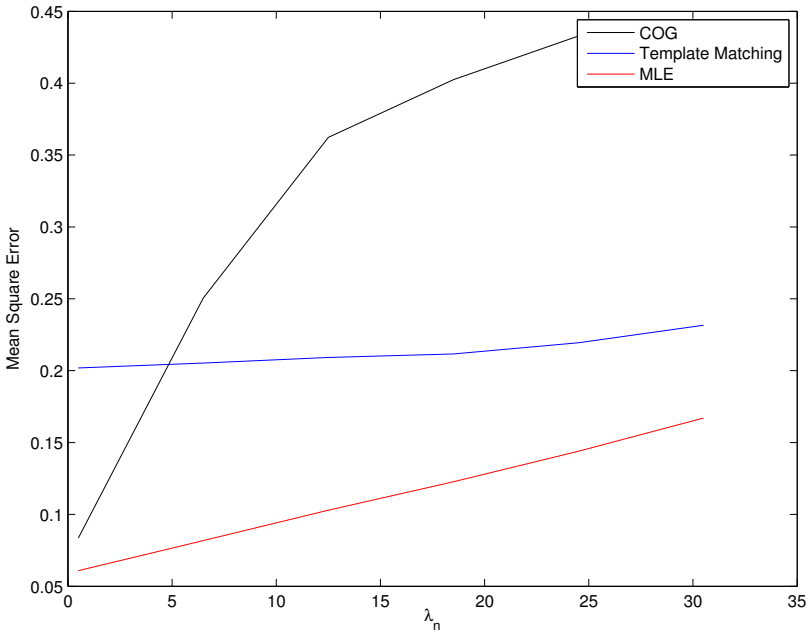

Fig. 6.6. Mean Square Error $I_0 = 300$, $\rho = 0.2$, $(\alpha, \beta) = (0.4, 0.4)$, 4×4 array.

Figure 6.6 shows the results for $I_0 = 300$, $\rho = 0.2$, $(\alpha = 0.4, \beta = 0.4)$ for a 4-by-4 (16 cells) quantized array. Here the difference in the performance of maximum likelihood and template matching estimates is much pronounced, with the maximum likelihood estimate giving a much better performance. Mean square error for maximum likelihood does not degrade as fast as the other two estimators as the noise rate is increased. Error is relatively large for Template Matching estimator even at high SNR.

6.2.2 Random parameter

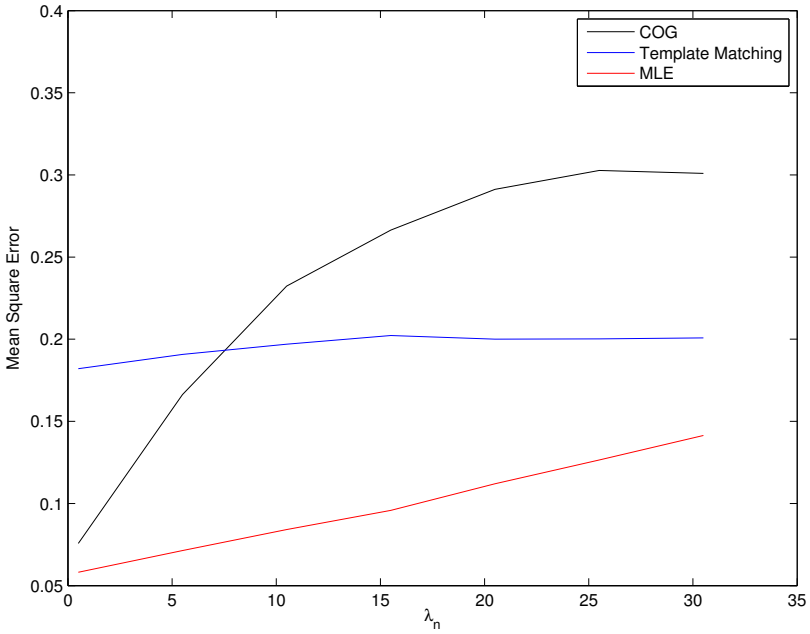

Fig. 6.7. Mean Square Error $I_0 = 130$, $\rho = 0.2$, $\sigma = 0.3$, 4×4 array.

Figure 6.7 shows results for $I_0 = 130$ and $\rho = 0.2$. α and β are chosen randomly from $\mathcal{N}(\mathbf{0}, \sigma^2 I)$, where $\sigma = 0.3$.

6.3 Low signal power, low noise power regime

For a typical FSO system used in deep space communications, it is not unusual to operate in a regime where the signal photon count is around 4 or 5 and average number of noise photons is 10 for the whole array. To achieve these number we selected $I_0 = 10$, $\rho = 0.1$, and $(\alpha, \beta) = (0.4, 0.4)$. For this typical scenario the mean squared error performance for the three estimators was evaluated for each of the 4×4, 8×8, 16×16 and continuous arrays.

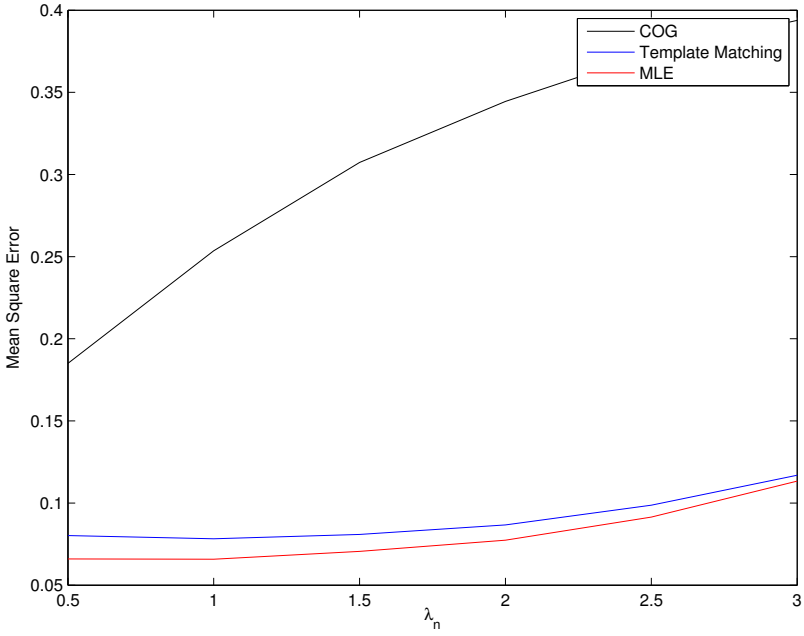

Fig. 6.8. Mean Square Error $I_0 = 100$, $\rho = 0.1$, $(\alpha, \beta) = (0.4, 0.4)$, continuous array.

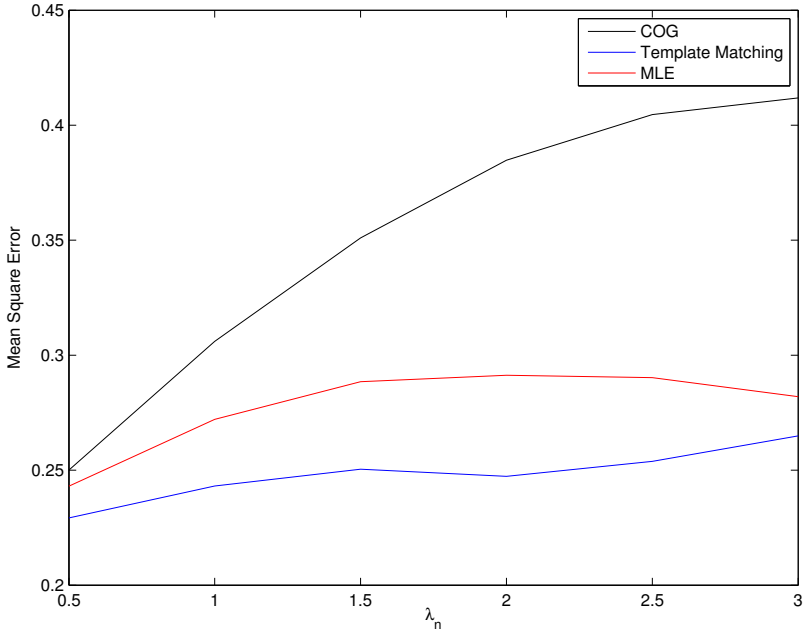

Fig. 6.9. Mean Square Error $I_0 = 100$, $\rho = 0.1$, $(\alpha, \beta) = (0.4, 0.4)$, 4×4 array.

Figure 6.8 shows the mean square error for the three estimators for continuous array. Contrast this with figure 6.9 which shows the error performance curves for 4×4 array. Since now the photons (both data and noise) are now reported to the center of the cell, the true data observations are lost and we now have access only to approximate version of real data. This would probably induce bias in the performance of the estimated values. As a result we can see that the template matching estimate does better than each of the other estimators.

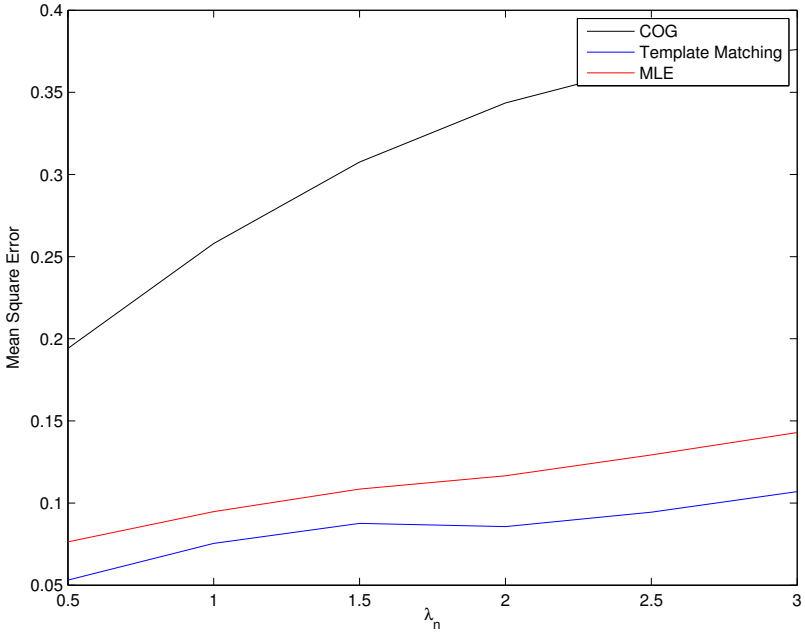

Fig. 6.10. Mean Square Error $I_0 = 100$, $\rho = 0.1$, $(\alpha, \beta) = (0.4, 0.4)$, 8×8 array.

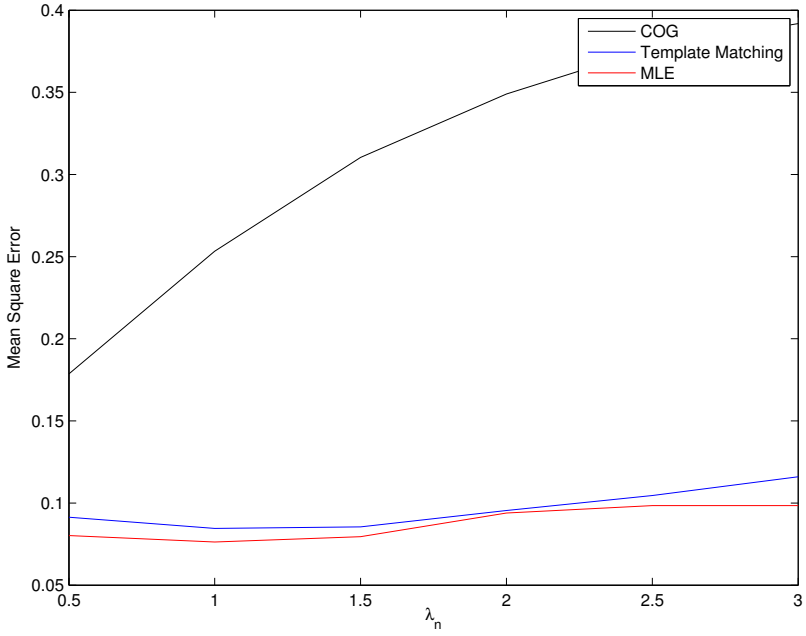

Fig. 6.11. Mean Square Error $I_0 = 100$, $\rho = 0.1$, $(\alpha, \beta) = (0.4, 0.4)$, 16×16 array.

We expect the things to improve as the quantization levels become smaller and smaller and the performance converging to the continuous array case. 8×8, and 16×16 cases are shown in figures above, and we can see the transition happening to the continuous case of figure 6.8.

7. CONCLUSION

In this chapter the final conclusions drawn from the results of the simulation are laid out.

7.1 Continuous array

For the continuous array we can see that the maximum likelihood estimator gives the least means squared error and the best performance especially when SNR was low. The performance of center of gravity estimator degrades noticeably during poor SNR conditions and has the largest mean squared error whereas all the three estimators give relatively comparable performance when SNR is high.

7.1.1 Unknown deterministic parameter

For the unknown parameter case the results show the superiority of maximum likelihood estimator whereas the performance of template matching estimator also gives a relatively reasonable performance. As we can see in figure 6.1 that the mean square error for maximum likelihood case is below 0.05 for 12.13 dB and the degradation is very little: the error is 0.07 for $\lambda_n = 30$ which in terms on SNR is -5.72 dB. However for center of gravity the error starts at around 0.05 and goes up to 0.45 for SNR -5.72 dB.

For the case when α and β are both zero, figure 6.2 indicates that maximum likelihood estimate error is 0.03 for 12.13 dB and goes up to 0.036 as SNR approaches -6 dB. However the center of gravity begins to give less error as it would ultimately

converge to the center of beam as noise power increases due to uniform noise (the center of gravity of uniform noise is the center of array).

7.1.2 Random parameter

For random parameter case the maximum likelihood estimator again gives least mean squared error. There was so significant improvement over the performance for unknown parameter case. Here the center of gravity error is smaller for high noise case (e.g. it is 0.21 for -5.72 dB of SNR). This is due to the fact that the values of α and β are chosen from $\mathcal{N}(\mathbf{0}, 0.2^2 I)$ and therefore most of parameter values would be close to zero and we know the center of gravity estimate gives smaller error when parameter values are closer to the center of array to begin with.

7.2 Quantized Array

7.2.1 Unknown deterministic parameter

It can be readily seen from figures pertaining to quantized arrays that the maximum likelihood estimator has the minimum variance as in the continuous case.For SNR -5 dB the mean square error for MLE is 0.11 as against 0.45 for center of gravity for same value of SNR. At 15 dB the MLE's error is 0.0589 versus center of gravity's 0.0714.

7.2.2 Random parameter

The prior information about the parameter gives us advantage in terms of additional information about the parameter and we exploit that while calculating the error. That shows up as an extra term in equation (5.4) which is the natural logarithm of the parameter's density function and its effect is to improve the performance of maximum likelihood estimate, especially for the low photon rate. For the 4×4 case the maximum likelihood error begins around 0.059 for 12.13 dB and moves up to 0.14 for -6 dB (for center of gravity the error is 0.075 and 0.43 respectively for the same SNR values).

7.3 Performance analysis of center of gravity, template matching and MLE in poor SNR conditions

As discussed earlier, dark current noise along with quantization, are the two major phenomenon affecting the beam center estimation. In this section we take a look at how this uniform noise affects the three estimators.

Since the noise is uniform in nature, we expect the center of gravity to converge to the center of the array under high noise scenario. Therefore the center of gravity estimate would gradually deviate from the true parameter values towards the center of array as the noise power increases.

The degradation in case of template matching and maximum likelihood estimates happen because of the occurrence of several peaks in the maximum likelihood or template matching functions due to noise photons. As described in [6] when noise is low there is only one unique maximum and it is relatively stable from one realization to the next. However in presence of noise photons the peak may shift to a region where there might be a cluster of noise photons or regions of high density of dark current emissions. These 'noise' peaks are larger than the 'true signal' peak value and would cause estimation errors, and ultimately larger variance. These significantly large errors in estimate are known as *outliers*. Even though we know that maximum likelihood estimate achieves the *Cramer Rao Lower Bound* asymptotically, low or poor SNR affects the performance of ML estimate and prevents it from attaining the bound even for a large number of data records.

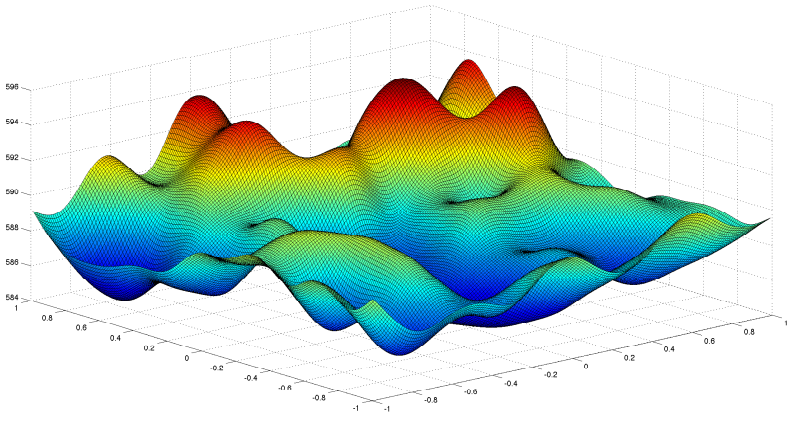

Fig. 7.1. Log-likelihood function for parameter values $I_0 = 100, \rho = 0.1, (\alpha, \beta) = (0.5, 0.5), \lambda_n = 50$.

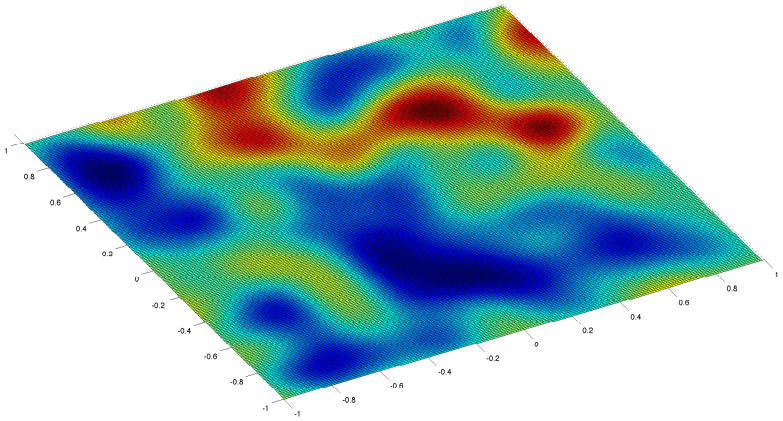

Fig. 7.2. Contour plot of log-likelihood function for parameter values $I_0 = 100, \rho = 0.1, (\alpha, \beta) = (0.5, 0.5), \lambda_n = 50$.

Figures 7.1 and 7.2 show the log likelihood and contour plots for the case of SNR -15dB. It can be clearly seen that noise has caused one extra peak to occur in the vicinity of (0.5, 0.5) and the maximization of the log likelihood might yield a poor

estimate in the presence of large "noise" peaks. To remedy we must use a sufficiently large peak for the beam for the same signal power. For this purpose we set I_0 equal to 2000 and ρ at 0.0225 to keep the signal power the same as in previous case. Below are the results for those beam parameters.

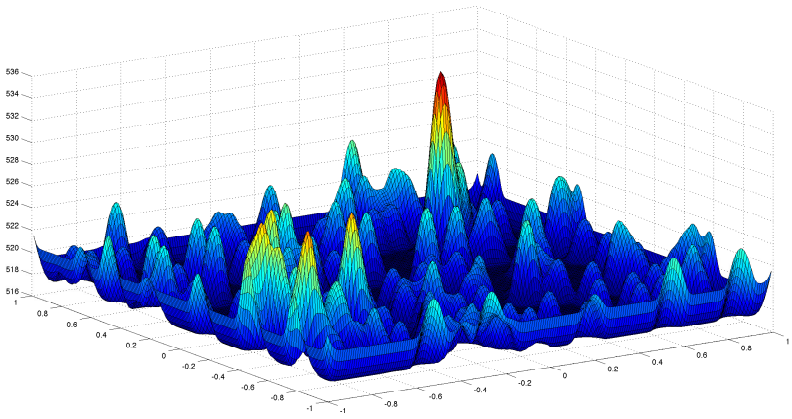

Fig. 7.3. Log-likelihood function for parameter values $I_0 = 2000, \rho = 0.0225, (\alpha, \beta) = (0.5, 0.5), \lambda_n = 50$.

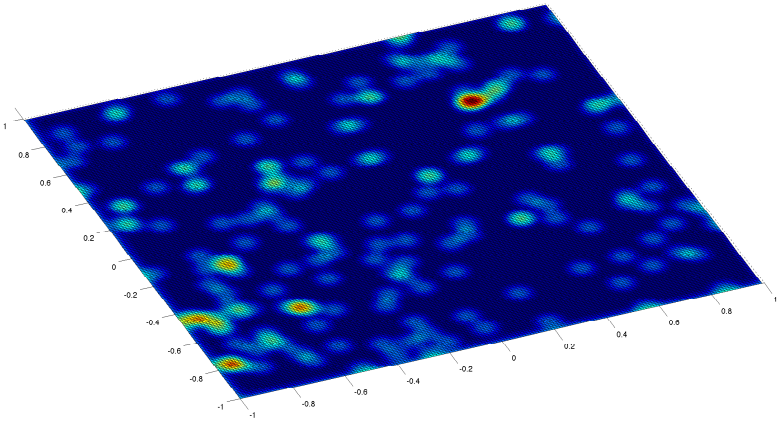

Fig. 7.4. Contour plot of log-likelihood function for parameter values $I_0 = 2000, \rho = 0.0225, (\alpha, \beta) = (0.5, 0.5), \lambda_n = 50$.

Thus as can be seen in 7.3 and 7.4, making the peak of the signal intensity function "larger" for the same SNR it can be seen that there is one distinct maximum at location (0.5, 0.5) and the maximum likelihood function would yield better result in this case.

LIST OF REFERENCES

LIST OF REFERENCES

[1] R. Gagliardi and S. Karp, *Optical Communication*. New York: John Wiley and Sons, 2nd ed., 1995.

[2] D. L. Snyder and M. I. Miller, *Random Point Processes in Time and Space*. New York: Springer-Verlag, 1991.

[3] E. K. Chong and S. H. Żak, *An Introduction to Optimization*. Hoboken New Jersey: John Wiley and Sons, fourth ed., 2013.

[4] B. Slocumb and D. Snyder, "Maximum likelihood estimation applied to quantum-limited optical position sensing," *SPIE*, vol. 1304 Acquisition, Tracking, and Pointing, no. IV, 1990.

[5] G. Rudolph, "Convergence of evolutionary algorithms in general search spaces," (Piscataway, NJ), pp. 50–54, Proceedings of the Third IEEE conference on Evolutionary Computation, 1996.

[6] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River NJ: Prentice Hall, 1996.