8-2009

Plume Source Localization and Boundary Prediction

Samir Sahyoun
University of Tennessee - Knoxville

To the Graduate Council:

I am submitting herewith a thesis written by Samir Sahyoun entitled "Plume Source Localization and Boundary Prediction." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Seddik Djouadi, Major Professor

We have read this thesis and recommend its acceptance:

Aly Fathy, Hairong Qi, Mongi Abidi

Accepted for the Council:
Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

To the Graduate Council:

I am submitting herewith a thesis written by Samir Sahyoun entitled "Plume Source Localization and Boundary Tracking." I have examined the final electronic copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Electrical Engineering.

Seddik Djouadi, Major Professor

We have read this thesis
and recommend its acceptance:

Aly Fathy

Hairong Qi

Mongi Abidi

Accepted for the Council:

Carolyn R. Hodges
Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Plume Source Localization and Boundary Prediction

A Thesis
Presented for the
Master of Science
Degree
The University of Tennessee, Knoxville

Samir Sahyoun
August 2009

Abstract

Plume location and prediction using mobile sensors is the main contribution of this thesis. Plume concentration values measured by chemical sensors at different locations are used to estimate the source of the plume. This is achieved by employing a stochastic approximation technique to localize the source and compare its performance to the nonlinear least squares method. The source location is then used as the initial estimate for the boundary tracking problem. Sensor measurements are used to estimate the parameters and the states of the state space model of the dynamics of the plume boundary. The predicted locations are the reference inputs for the LQR controller. Measurements at the new locations (after the correction of the prediction error) are added to the set of data to refine the next prediction process. Simulations are performed to demonstrate the viability of the methods developed. Finally, interpolation using the sensors locations is used to approximate the boundary shape.

Table of Contents

List of Figures

Chapter 1

Introduction

Accidental gas releases from industrial sites that results in dangerous chemical plumes makes the problem of tracking such plumes extremely important. The fear of biological terrorist attacks is another motive that made this subject a hot research topic to answer the question of what is the fastest and most accurate approach to locate and track a possible chemical plume.

Plume spreading is affected by different factors [10]. Apart from the nature of the gas and the temperature at the release point; weather conditions is the most important factor. It is impossible to track plumes without full knowledge of the weather conditions, especially the wind direction and velocity. The wind factor is included in most of the mathematical plume models in literature [10]. Figure 1.1 shows a real world plume that spreads in the downwind direction.

The objective of this work is to implement an application scenario to examine the performance of a new developed data sharing middleware that is able to handle multiple distributed data sources and dynamically changing items, and to assist in real-time INFOrmation Dissemination (INFOD) across multiple agencies for homeland security purposes. The ultimate goal of the INFOD model is to support the timely delivery of

Figure 1.1 A Real World Plume

valuable information [31]. Figure 1.2 shows an information dissemination plume scenario.

Researchers use the concentration of the gas at any location to study plumes [9]. Special sensors can measure the values of concentrations at the desired locations. The surrounding area around factories that use a certain poisoning gas is covered by a grid of sensors that are able to measure the concentration of this specific gas.

In literature, many source localization techniques have been developed. In [11], the Maximum Likelihood algorithm (MLE) is compared with the Direct Triangulation algorithm. Based on the contaminant attenuation model, they proposed a wireless sensor network (WSN) to estimate the plume source location in a sensor field using the MLE algorithm and the Direct Triangulation algorithm respectively. They showed that better accuracy using the two algorithms is achieved if the sensor nodes reach to appropriate numbers in the field. The MLE algorithm was shown to be robust to the much noise compared with the Direct Triangulation algorithm. In [12], the problem of plume source localization was formulated using multiple intensity sensors as the most likely sequence decoding over a fuzzy hidden Markov model. Under the assumption that each sensor has high detection and low false alarm probability, they proposed a greedy heuristic decoding algorithm with much less computational cost than Viterbi algorithm. The plume localization accuracy of the algorithm was shown to be close to the best decoder using Viterbi algorithm when tracing a single plume using randomly deployed sensors.

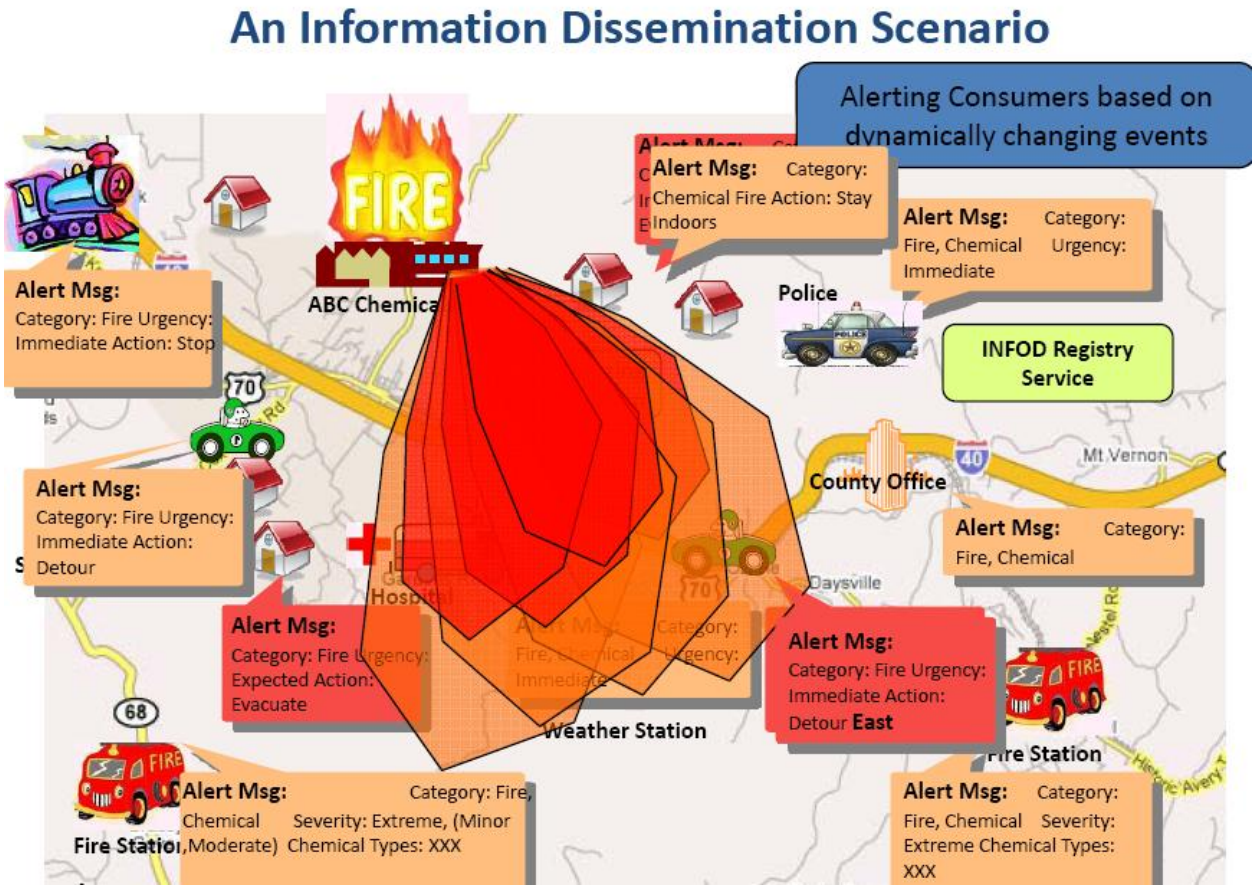

Figure 1.2 An Information Dissemination Scenario. Plume tracking information should be distributed to nearest police offices, fire stations, etc. Figure from an InfoD quarterly review meeting presentation by Raghul Gunasekaran

The problem of predicting the spread of an airborne plume has also been examined by many researchers. The prominent approach among these is to compute the parameters of the advection-diffusion equation which governs the spread of the agent [23]. A non-linear least-squares method for estimating these parameters offline is presented in [24], and an exploration of agent spread under continuous release assumption is given in [25], [26]. In [12], the problem of plume localization was formulated using multiple intensity sensors as the most likely sequence decoding over a fuzzy hidden Markov model. Under the assumption that each sensor has high detection and low false alarm probability, they proposed a greedy heuristic decoding algorithm with much less computational cost than Viterbi algorithm. The plume localization accuracy of the algorithm was shown to be close to the best decoder using Viterbi algorithm when tracing a single plume using randomly deployed sensors. Several other methods have been proposed, and an overview of these is available in [27].

The most common sensors are the semiconductors gas sensors [21]. Chemical sensors based upon semiconductors react to various reducing gases such as carbon monoxide, hydrogen or ethanol [22]. When exposed to air, a layer of oxygen is absorbed onto the bed of semiconductor granules that forms the sensing element. When a reducing gas is present, oxidation occurs and the layer of oxygen on the sensor surface is diminished, increasing the conductivity (and therefore reducing the resistance) of the sensor [22]. Other sensors use p-type semiconductor base materials (instead of the n-type

semiconductors mentioned above) and react to oxidizable gases (such as O_2, NO_2 and Cl_2) [22]. All of these sensors are tuned to target specific gases by changing the operating temperature of the sensor or through the addition of impurities and catalysts. However most still have limited selectivity, reacting strongly to the target gas, but also reacting to a number of other reducing or oxidizing gases [22]. These sensors have fast response and high sensitivity, but have the disadvantage of a return time of approximately 30 seconds [21]. Other types of sensors are discussed in details in [22].

In this thesis, the source of the plume is estimated for a fixed grid of sensors using two methods, Non Linear Least Squares [2] and Kiefer-Wolfowitz stochastic approximation algorithm [4]. Performance comparison between both methods is shown. Moreover, source localization will be performed using a number of mobile sensors that move and converge to the plume source. Plume source is then used as an initial location to start the plume location prediction process.

Thesis is organized as follows; Chapter 1 contains an introduction about the problem and what motives the work. Also it contains a summary of what people have done in this area.

Chapter 2 discusses mathematical models that describe plumes. Lagrangian and Eulerian dispersion models are discussed briefly while Gaussian dispersion model is discussed in details. The other part of the chapter discusses the main three types of

chemical plumes that are classified by the nature of the gas and the temperature at the release source.

Localizing the source of the plume is discussed in chapter 3. The approximate estimate of the source point will be used as the initial point in the tracking process later on. A uniform propagation of the plume is assumed and a time averaging of the measurements at the sensor nodes is performed before solving the nonlinear least square problem. Choosing the initial points to solve the problem is an important factor to reach the best solution faster. The nonlinear least squares approach is compared with the stochastic approximation approach. Stochastic approximation techniques are iterative methods that attempt to find zeros of functions which cannot be computed directly, but only estimated via noisy observations. The basic stochastic approximation algorithms were introduced by Robbins and Monro [3] and by Kiefer and Wolfowitz [4] in the early 1950s. The original work was motivated by the problem of finding a root of a continuous function $g(\theta)$, where the function is not known but the experimenter is able to obtain noisy measurements at any desired value of θ.

Plume location prediction is discussed in chapter 4. The process depends mainly on the measurements of concentrations of the plume provided by sensors at different times and locations. The value of the concentration at the boundary of the plume is pre-defined as the value that after which, the plume is not dangerous. Initial data for the dynamic boundary locations and times are needed to start the prediction process. Starting

from an initial location (the source of the plume), sensors send their measurements and move in different directions as long as they keep sending values of concentrations above the pre-defined boundary threshold value. In all simulations, it is assumed that sensors know their locations.

Chapter 5 discusses the interpolation process used to form the shape of the predicted boundary. The interpolation method used is Spline method. Finally, Chapter 6 contains thesis conclusion and future work.

Chapter 2

Plume Modeling

2.1 Introduction

For simulation purposes, plume mathematical models are based on a mathematical description of physical and chemical processes taking place in the atmosphere. In this chapter, mathematical models of plumes are described. They can be grouped into classes based on different criteria; the spatial scale, temporal scale, pollutant type, and emission source type. The main three models encountered in the literature are Gaussian, Lagrangian and Eulerian models [10]. These models are discussed next.

2.1.1 Gaussian model

Gaussian model is the oldest and the most commonly used model type. It assumes that the plume dispersion has a Gaussian distribution, meaning that the pollutant distribution has a normal probability distribution. Gaussian models are most often used for predicting the dispersion of continuous, buoyant air pollution plumes originating from ground-level or elevated sources. Gaussian models may also be used for predicting the dispersion of non-continuous air pollution plumes (called puff models) [7].

2.1.2 Lagrangian model

Lagrangian dispersion model mathematically follows pollution plume particles as they move in the atmosphere and model the motion of the particles as a random walk process. The Lagrangian model then calculates the plume dispersion by computing the statistics of the trajectories of a large number of the pollution plume particles. A Lagrangian model uses a moving frame of reference as the particles move from their initial location [7].

Lagrangian model is used to develop Second-Order Closure Integrated Puff (SCIPUFF) [6], a Lagrangian plume dispersion model developed by Titan's ARAP (The Alliance for Responsible Atmospheric Policy) Group that uses a collection of Gaussian puffs to represent an arbitrary, three-dimensional time-dependent concentration. The turbulent diffusion parameterization is based on turbulence closure theory, providing a direct relationship between the predicted dispersion rate and turbulent velocity statistics of the wind field. In addition to the average concentration value, the closure model also provides a prediction of the statistical variance in the concentration field resulting from the random fluctuations in the wind field. The closure approach also provides a direct representation for the effect of averaging time. SCIPUFF has been incorporated into the Defense Threat Reduction Agency's (DTRA) Hazard Prediction and Assessment Capability (HPAC) software. HPAC is utilized for planning and analysis as well as in the field by military personnel to rapidly determine consequences of dispersing chemical,

nuclear and biological agents. SCIPUFF has been validated against a number of laboratory and field experiments, demonstrating its usefulness for non-military applications. It has been recommended as an alternative model by the EPA which can be used on a case-by-case basis for regulatory applications [6].

2.1.3 Eulerian dispersion model

The Eulerian dispersion model is similar to a Lagrangian model in that it also tracks the movement of a large number of pollution plume particles as they move from their initial location. The most important difference between the two models is that the Eulerian model uses a fixed three-dimensional Cartesian grid as a frame of reference rather than a moving frame of reference [7].

In the next section, different types of plumes are discussed.

2.2 Types of plumes

There are three primary types of air pollution emission plumes:

2.2.1 Buoyant plumes

Buoyant plumes are lighter than air because they are at a higher temperature and lower density than the ambient air which surrounds them, or they are at about the same temperature as the ambient air but have a lower molecular weight and hence lower density than the ambient air. For example, the emissions from the flue gas stacks of

industrial furnaces are buoyant because they are considerably warmer and less dense than the ambient air. As another example, an emission plume of methane gas at ambient air temperatures is buoyant because methane has a lower molecular weight than the ambient air [10].

2.2.2 Dense gas plumes:

Dense gas plumes are heavier than air because they have a higher density than the surrounding ambient air. A plume may have a higher density than air because it has a higher molecular weight than air (for example, a plume of carbon dioxide). A plume may also have a higher density than air if the plume is at a much lower temperature than the air. For example, a plume of evaporated gaseous methane from an accidental release of liquefied natural gas (LNG) may be as cold as -161 °C. [10].

2.2.3 Passive or neutral plumes:

Passive or neutral plumes are plumes which are neither lighter nor heavier than air and moves according to the surrounding weather conditions. An example of this type of plumes is mist; a phenomenon of small droplets suspended in air [10].

2.3 One dimension Gaussian plume:

The basic one dimensional (1D) transport equation of a Gaussian plume dispersion is given by the following partial differential equation (PDE) [7]:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} D \frac{\partial c}{\partial x} - v \frac{\partial c}{\partial x} \tag{2.1}$$

which has the solution [7]:

$$c(x,t) = \frac{M}{\sqrt{4\pi t}\sqrt{D}} \exp\left(-\frac{(x - vt)^2}{4tD} \right) \tag{2.2}$$

where the concentration c is a solution of the transport equation (2.1), *M* denotes the total mass per unit area in the fluid system, *D* is the diffusivity, *v* is the wind velocity. A one dimension Gaussian plume for an instantaneous source for 6 time instants is shown in Figure 2.1.

2.4 Two dimension Gaussian Plume:

In analogy to the 1D situation analytical solutions can be derived for the higher dimensional cases. The generalization of the 1D normal distribution for 2D is [7]:

$$c(x,y,t) = \frac{M}{4\pi t\sqrt{D_x D_y}} \exp\left(-\frac{1}{4t}\left(\frac{(x - vt)^2}{D_x} + \frac{y^2}{D_y} \right) - \lambda t \right) \tag{2.3}$$

which is the solution of the 2D PDE [7]:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} D_x \frac{\partial c}{\partial x} + \frac{\partial}{\partial y} D_y \frac{\partial c}{\partial y} - v \frac{\partial c}{\partial x} - \lambda c \tag{2.4}$$

where the concentration c is a solution of the transport equation, M denotes the total mass per unit area in the fluid system, D_x , D_y are the diffusivities, v is the wind velocity and λ

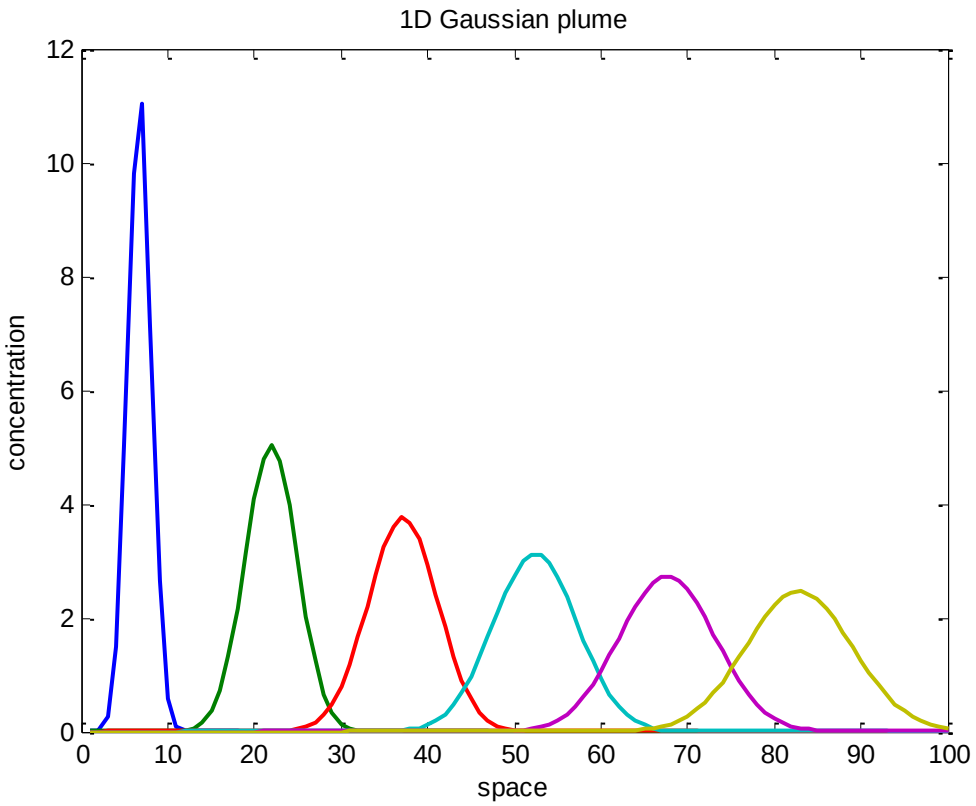

Figure 2.1 1D Gaussian plume for instantaneous source for 6 time instants. Concentration c is a solution of the transport equation 2.1.

is the decay coefficient. Concentration distribution of an instantaneous source Gaussian plume in two dimension is shown in Figure 2.2. Gaussian plume along slices of a constant y axis is shown in Figure 2.3.

2.5 Three dimension Gaussian plume:

The PDE that describes the 3D Gaussian plume is given by [7]:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} D_x \frac{\partial c}{\partial x} + \frac{\partial}{\partial y} D_y \frac{\partial c}{\partial y} + \frac{\partial}{\partial z} D_z \frac{\partial c}{\partial z} - v \frac{\partial c}{\partial x} - \lambda c \qquad (2.5)$$

which has the solution [7]:

$$c(x, y, z, t) = \frac{M}{(4\pi t)^3 \sqrt{D_x D_y D_z}} \exp\left(-\frac{1}{4t} \left(\frac{(x - vt)^2}{D_x} + \frac{y^2}{D_y} + \frac{z^2}{D_z} \right) - \lambda t \right) \qquad (2.6)$$

where the concentration c is a solution of the transport equation, M denotes the total mass per unit area in the fluid system, D_x, D_y and D_z are the diffusivities, v is the wind velocity and λ is the decay coefficient. Concentration distribution from a 3D Gaussian plume is shown in Figures 2.4, 2.5.

2.6 Steady State Continuous source plume:

Solution of the steady state 3D Gaussian plume generated from a continuous emission source is given by [7]:

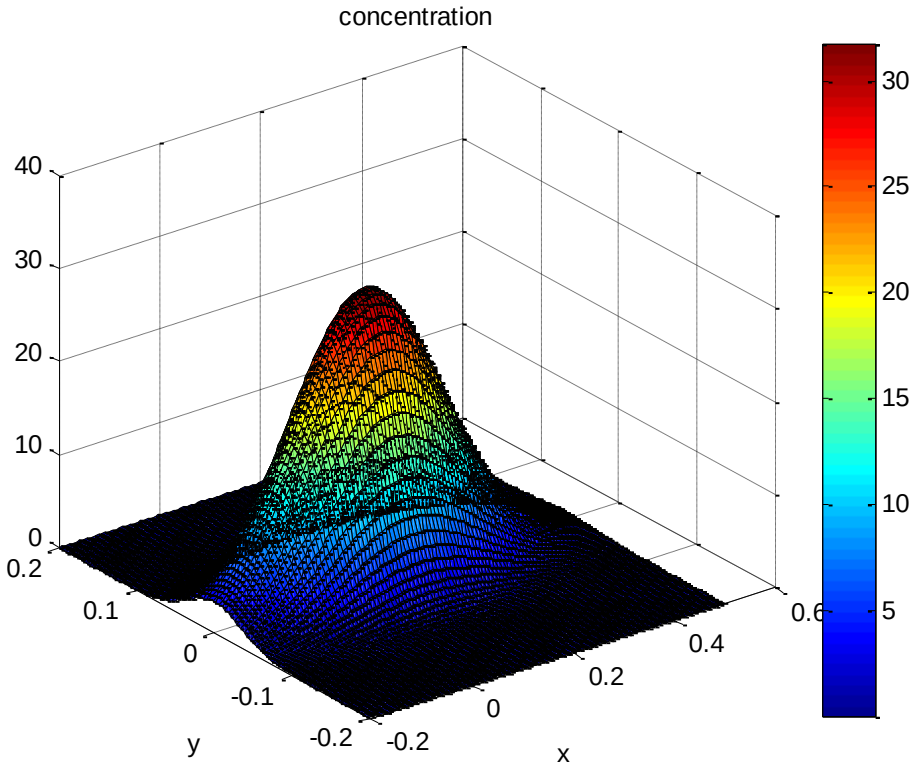

Figure 2.2 Instantaneous source Gaussian plume in 2D

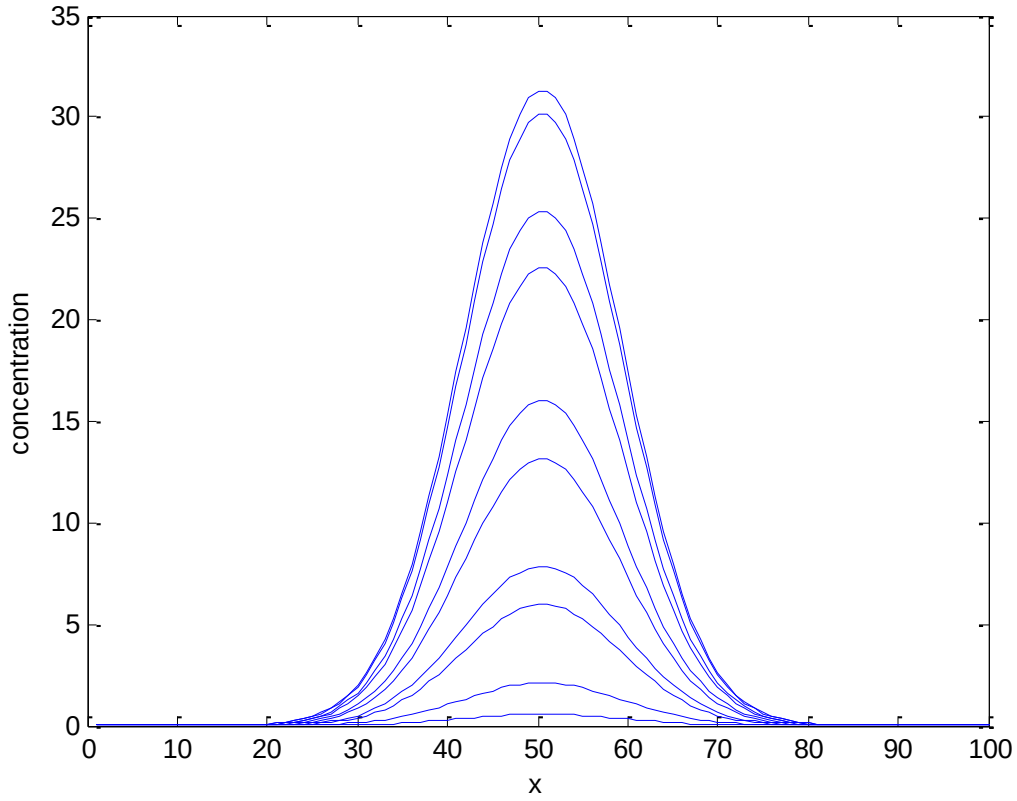

Figure 2.3 2D Gaussian plume along slices y=constant

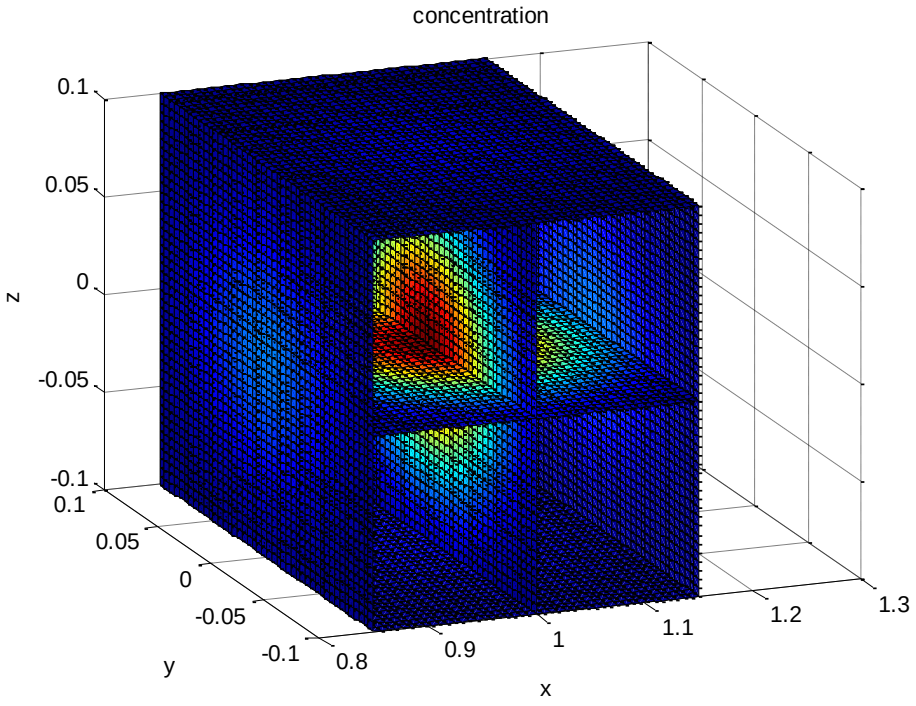

Figure 2.4 Concentration distribution from a 3D Gaussian plume

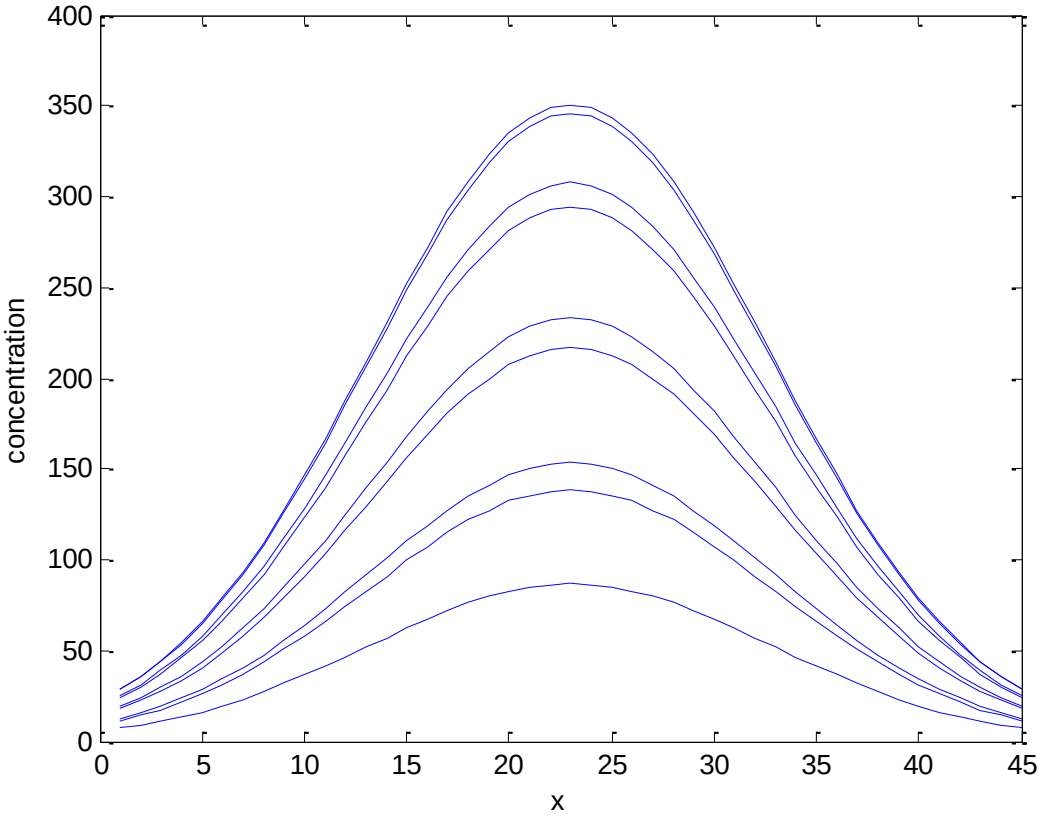

Figure 2.5 3D Gaussian puff along line x=0

$$c(x,y,z)=\frac{M}{4\pi x\sqrt{D_y D_z}}\exp\left(-\frac{vy^2}{4xD_y}\right)\left[\begin{array}{l}\exp\left(-\frac{v(z-H)^2}{4xD_z}\right)\\+\exp\left(-\frac{v(z+H)^2}{4xD_z}\right)\end{array}\right]\exp\left(\frac{\lambda}{v}x\right) \qquad (2.7)$$

where the concentration c is a solution of the transport equation, M denotes the total mass per unit area in the fluid system, D_x, D_y and D_z are the diffusivities, v is the wind velocity and λ is the decay coefficient.

Such models are used extensively for estimations of the local development of a plume in the atmosphere. For the most common application of release from a stack, the parameters are visualized in Figure (2.6). The Gaussian models take into account diffusive processes, advection with a mean air flow direction (wind), and first order decay [7]. The term diffusion here is used as an umbrella term for various processes which have in common the tendency to lower concentration or temperature gradients. Diffusion at the molecular scale can surely be neglected in the atmosphere, while variations and fluctuations at various scales within the velocity field are the cause for the observation of diffusion at a larger scale. Moreover, turbulence adds as another origin of diffusion [17].

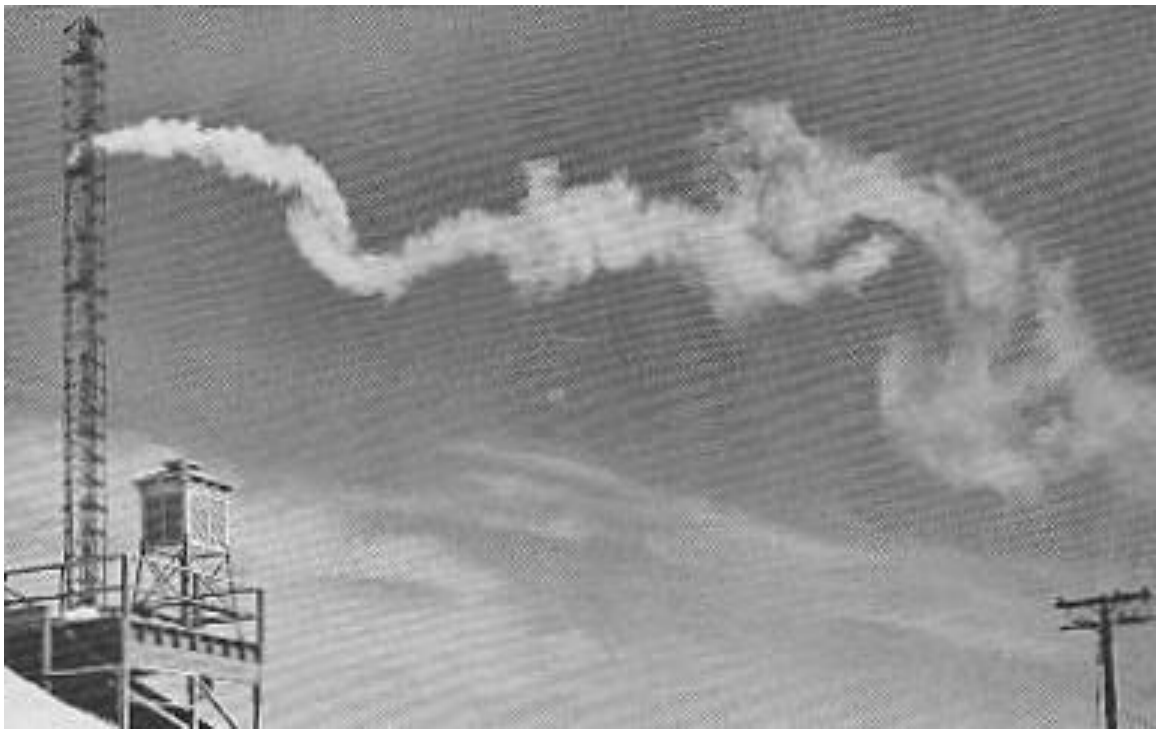

Figure 2.6 Steady state Gaussian plume from a continuous emission source

Chapter 3

Source Localization

3.1 Introduction:

Tracking the source of a chemical plume is a hot research topic [8]. A terrorist attack or an accidental release from a chemical factory might produce a plume of this harmful gas. Accurate estimation of the source of the plume provides an opportunity to stop the release as fast as possible. It is also important to estimate the source of the plume for the purpose of tracking the boundary of the plume, the source is a good initial point for the prediction process.

To estimate the source of a plume we use values of concentrations of the gas at some points within the plume range. These values are measured by sensors that are assumed to be distributed in that area. Various types of sensors can measure releases of potentially harmful chemical, biological and radiological materials. When networked together they can provide real-time detection, identification and assessment of the event.

In the literature, many source localization techniques have been developed. In [11], the Maximum Likelihood algorithm (MLE) is compared with the Direct Triangulation algorithm. Based on the contaminant attenuation model, a wireless sensor network (WSN) to estimate the plume source location in a sensor field using the MLE algorithm and the Direct Triangulation algorithm is proposed. It is showed that better accuracy using the two algorithms is achieved if the sensor nodes reach an appropriate

numbers in the field. The MLE algorithm was shown to be robust to the much noise compared with the Direct Triangulation algorithm [11].

In [12], the problem of plume localization was formulated using multiple intensity sensors as the most likely sequence decoding over a fuzzy hidden Markov model. Under the assumption that each sensor has high detection and low false alarm probability, the authors proposed a greedy heuristic decoding algorithm with much less computational cost than the Viterbi algorithm. The plume localization accuracy of the algorithm was shown to be close to the best decoder using the Viterbi algorithm when tracing a single plume using randomly deployed sensors.

In this chapter, a uniform propagation of the plume is assumed and a time averaging of the measurements at the sensor nodes is performed. To estimate the source location, a nonlinear least square algorithm proposed in [1] is used. Choosing the initial points to solve the problem is an important factor to reach the best solution faster. This nonlinear least squares approach is compared with the stochastic approximation approach.

Stochastic Approximation technique is developed to estimate the source. The Stochastic approximation techniques are iterative methods that attempt to find zeros of functions which cannot be computed directly, but only estimated via noisy observations. The basic stochastic approximation algorithms were introduced by Robbins and Monro [3] and by Kiefer and Wolfowitz [4] in the early 1950s. The original work was motivated

by the problem of finding a root of a continuous function $J(\beta)$, where the function is not known but the experimenter is able to obtain noisy measurements at any desired value of β.

3.2 Non-linear Least squares:

Using the non-linear least squares technique to estimate the source of a plume is one of the standard techniques to solve the problem [1]. We assume a set of *N* stationary sensors that are randomly distributed at positions $(x_i, y_i), i = 1, \dots, N$. The plume source is located at (x_s, y_s). If the concentration at the source is c, the concentration at sensor i is inversely proportional to the distance d between the source and the sensor raised to some power $\alpha \in \mathbb{R}$ which depends on the environment. The measured concentration z at sensor i is given by:

$$z_i = \frac{c}{d_i^{\alpha}} + w_i \tag{3.1}$$

where w_i is additive Gaussian noise with zero mean and variance σ_i while d is the distance from the source, i.e.;

$$d_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{3.2}$$

we assume that sensor i knows its location through GPS or any localization technique. The cost function to be minimized is:

$$J = \sum_{i=1}^{N} \left(\frac{c}{[(\hat{x}_s - x_i)^2 + (\hat{y}_s - y_i)^2]^{\frac{\alpha}{2}}} - \bar{z}_i \right)^2 \tag{3.3}$$

where $\hat{x}_s, \hat{y}_s$ are the source estimated coordinates, x_i, y_i are the location coordinates at sensor i and $\bar{z}_i$ is the measured concentration received from sensor i. The goal is to find the optimum values of $\hat{x}_s, \hat{y}_s$ that minimize the cost function J.

The least squares technique gives better performance at low noise channels than high noise channels. That comes from the fact that this technique does not take noise into account.

The algorithm used in this problem is the Gauss-Newton nonlinear least squares algorithm [2]. The Gauss–Newton algorithm is a method used to solve non-linear least squares problems. It can be seen as a modification of Newton's method for finding a minimum of a function. Unlike Newton's method, the Gauss–Newton algorithm can only be used to minimize a sum of squared function values, but it has the advantage that second derivatives, which can be challenging to compute, are not required [2].

Non-linear least squares problems arise for instance in non-linear regression, where parameters in a model are sought such that the model is in good agreement with available observations.

Given N functions r_i, $(i = 1,...,N)$, where N is the number of sensors and

$$r_i = \frac{c}{\left[(\hat{x}_s - x_i)^2 + (\hat{y}_s - y_i)^2\right]^{\frac{\alpha}{2}}} - \bar{z}_i \tag{3.4}$$

Let $\beta = (\hat{x}_s, \hat{y}_s)$ denotes the minimization factors in (3.3), then the Gauss–Newton algorithm finds the minimum of the sum of squares [2]:

$$J(\beta) = \sum_{i=1}^{m} r_i^2(\beta) \tag{3.5}$$

Starting with an initial guess β^o for the minimum, the method proceeds by the iterations [2]:

$$\beta^{s+1} = \beta^s + \delta\beta, \tag{3.6}$$

with the increment $\delta\beta$ satisfying the normal equations [2]:

$$(\mathbf{J_r^T J_r})\delta\beta = -\mathbf{J_r^T r} \tag{3.7}$$

Here, $\mathbf{r}$ is the vector of functions r_i , and $\mathbf{J_r}$ is the $N \times 2$ Jacobean matrix of $\mathbf{r}$ with respect to β, both evaluated at β^s. The superscript $\mathbf{T}$ denotes the matrix transpose. In data fitting, where the goal is to find the parameters β such that a given model function $y = f(x, \beta)$ fits best some data points (x_i, y_i), the functions r_i are the residuals [2]:

$$r_i(\beta) = y_i - f(x_i, \beta). \tag{3.8}$$

Then, the increment $\delta\beta$ can be expressed in terms of the Jacobean of the function *f*, as [2]:

$$(\mathbf{J_r^T J_r})\delta\beta = \mathbf{J_r^T r} \tag{3.9}$$

In order for $\boldsymbol{J_r^T J_r}$ to be invertible, we always assume that $N \gg 2$, i.e., the number of measurements are always larger than the number of minimization factors, which is 2 in the case of 2D Cartesian coordinates for the plume source. So the normal equation can be solved. For a relatively small m, methods like QR factorization or Choleski factorization can be used to solve the linear equations of the unknown $\delta\beta$. Iterative methods are required when m is large. Conjugate gradient method is one choice [12]. Columns of $\boldsymbol{J_r}$ should be independent, otherwise the iteration will fail because $\boldsymbol{J_r^T J_r}$ becomes singular in this case.

As stated earlier, Gauss-Newton algorithm can be derived also from Newton's method. For minimizing a function J of a parameter β , the recurrence relation for Newton's method is [2]:

$$\beta^{s+1} = \beta^s - H^{-1} g \tag{3.10}$$

where g is the gradient vector of J and H is the Hessian matrix of J, and [2]:

$$g_j = 2 \sum_{i=1}^{m} r_i \frac{\partial r_i}{\partial \beta_j} \tag{3.11}$$

Differentiating the gradient elements g_j with respect to β_k produces the elements of the Hessian matrix [2]:

$$H_{jk} = 2 \sum_{i=1}^{m} \left(\frac{\partial r_i}{\partial \beta_j} \frac{\partial r_i}{\partial \beta_k} + r_i \frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k} \right) \tag{3.12}$$

In the Gaussian-Newton method, the Hessian matrix is approximated by ignoring the second derivative terms [2]:

$$H_{jk} \cong 2\sum_{i=1}^{m} J_{ij}J_{ik} \tag{3.13}$$

where $J_{ij} = \frac{\partial r_i}{\partial \beta_j}$ are entries of the Jacobean $\boldsymbol{J}_r$. In matrix notation, the gradient and the Hessian matrix can be written as [2]:

$$\mathbf{H} \cong 2\mathbf{J}_r^T\mathbf{J}_r\,, \qquad \boldsymbol{g} = \mathbf{2}\boldsymbol{J}_r^T\boldsymbol{r} \tag{3.14}$$

Substitution into the recurrence relation above gives the operational equations [2]:

$$\beta^{s+1} = \beta^s + \delta\beta \tag{3.15}$$

where

$$\delta\beta = -(\boldsymbol{J}_r^T\boldsymbol{J}_r)^{-1}\boldsymbol{J}_r^T\boldsymbol{r}$$

Conversion is expected as long as the approximation is relatively accurate; that is [2]:

$$\left|\frac{\partial r_i}{\partial \beta_j}\frac{\partial r_i}{\partial \beta_k}\right| \gg \left|r_i\frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k}\right|$$

Figure 3.1 shows the simulation result of this technique with different noise levels.

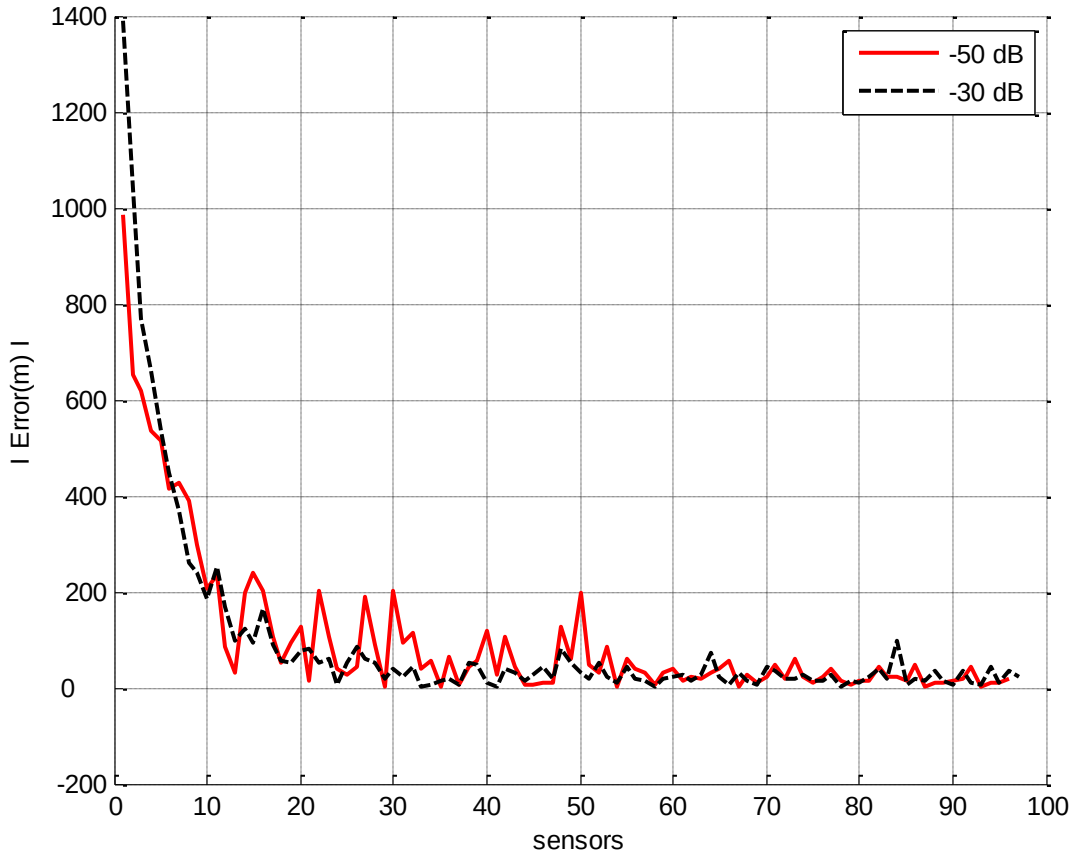

Figure 3.1 Performance of the Non linear least squares source localization technique at different noise levels. Better results at higher signal to noise ratios.

3.3 Stochastic Approximation:

Stochastic approximation (SA) techniques are iterative methods that attempt to find zeros of functions which cannot be computed directly, but only estimated via noisy observations. The basic stochastic approximation algorithms were introduced by Robbins and Monro [3] and by Kiefer and Wolfowitz [4] in the early 1950s.

The original work was motivated by the problem of finding a root of a continuous function $J(\beta)$, where $\beta = (\hat{x}_s, \hat{y}_s)$ denotes the minimization factors $\hat{x}_s, \hat{y}_s$ in (3.3). The function is not known but the experimenter is able to take noisy measurements at any desired value of β [3].

An important feature of SA is the allowance for noisy input information in the algorithm [5]. SA methods are often better at coping with noisy input information than other search methods. Moreover, the theoretical foundation for SA is deeper than the theory for other stochastic search methods with noisy measurements. In the case of root-finding SA, the noise manifests itself in the measurements of $J(\beta)$ used in the search as β varies.

The recursive procedure in the general SA form [3]:

$$\hat{\beta}_{k+1} = \hat{\beta}_k - a_k \hat{J}_k(\hat{\beta}_k), \tag{3.16}$$

where $\hat{J}_k(\hat{\beta}_k)$ is the estimate of J at the iterate $\hat{\beta}_k$based on measurements of the loss function. Under appropriate conditions, the iteration in (3.16) converges to β^* in some stochastic sense.

Using the Kiefer-Wolfowitz algorithm [4], we wish to minimize the function $J(\beta)$ over the $\mathbb{R}^r$ valued parameter β. Let c_n (goes to zero) be a finite difference interval and let e_i be the standard unit vector in the ith coordination direction. Let β_n denotes the nth estimate of the minimum. Suppose that for each i, n, and random vectors $\chi_{n,i}^{+}$, $\chi_{n,i}^{-}$ we can observe the finite difference estimate [5]:

$$Y_{n,i} = -\frac{[J(\beta_n + c_n e_i \chi_{n,i}^{+}) - J(\beta_n - c_n e_i \chi_{n,i}^{-})]}{2c_n} \tag{3.17}$$

Let $Y_n = (Y_{n,1}, \dots, Y_{n,r})$ and update β_n by:

$$\beta_{n+1} = \beta_n + \epsilon_n Y_n \tag{3.18}$$

The importance of dealing with noise in the stochastic approximation approaches is shown in Figure 3.2. Noise in the shown function leads to produce a false minimum. Using Least Squares method discussed in the previous section gives the false minimum as a result but SA gives a correct one as a result.

Figure 3.3 shows that the SA technique gives better results than the Gaussian-Newton non-linear least squares technique at the same noise level. This comes from the fact that the SA methods are often better at coping with noisy input information than other search methods.

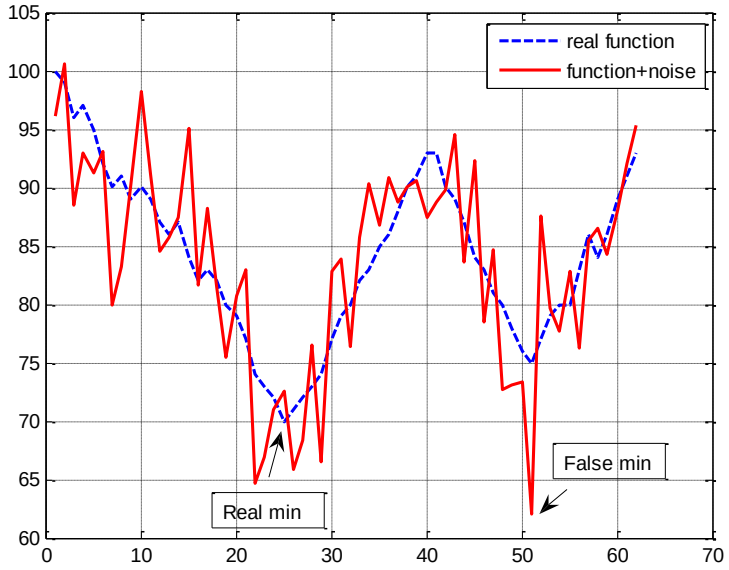

Figure 3.2 Effects of noise in finding the minimum of functions

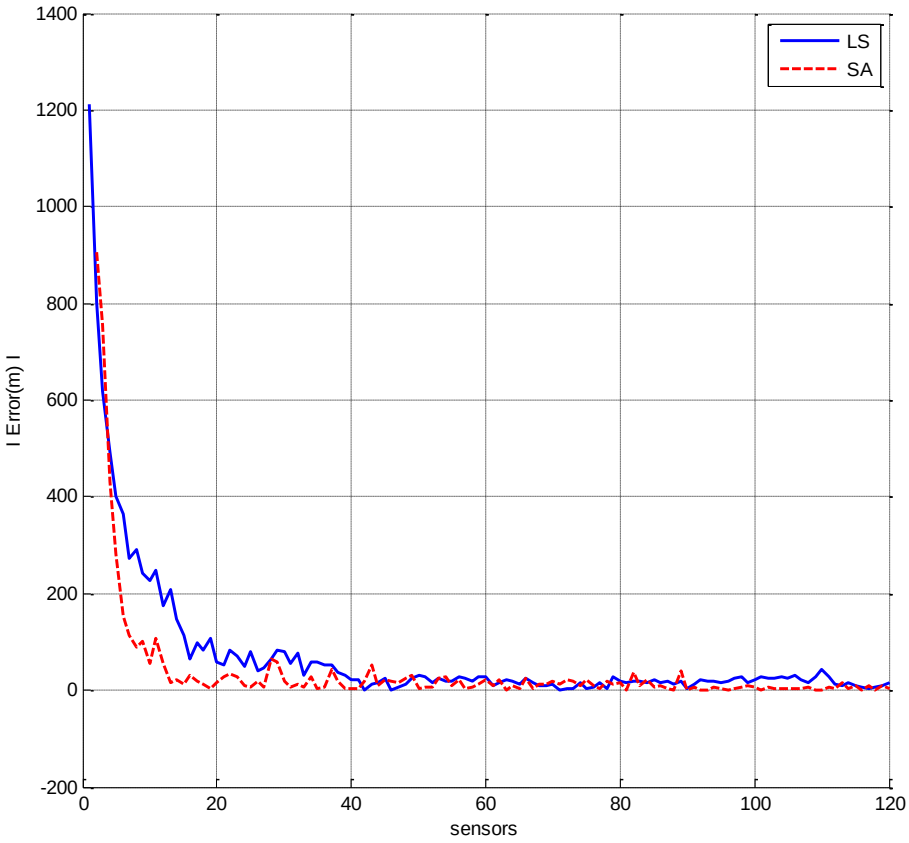

Figure 3.3 Performance of Stochastic Approximation technique compared with the least squares technique

Chapter 4

Plume Location Prediction

4.1 Introduction

Plume concentrations in general are very discontinuous. Gradient-based algorithms are not feasible in environments with high Reynolds numbers [17], [18]. The evolution of the chemical distribution in the flow at high Reynolds numbers, is turbulence dominated [17]. The result of the turbulent diffusion process is a highly discontinuous and intermittent distribution of the chemical [17], [19]. For a dense array of sensors distributed over an area, through which a turbulent flow was advecting a chemical, and the output of each sensor were averaged for a suitably long time (i.e., several minutes), then this average chemical distribution would be Gaussian. At low Reynold numbers, the evolution of the chemical distribution is dominated by molecular diffusion and the concentration field is well defined by a continuous function [19].

The problem of predicting the spread of an airborne plume has also been examined by many researchers. The prominent approach among these is to compute the parameters of the advection-diffusion equation which governs the spread of the agent [23]. A non-linear least-squares method for estimating these parameters offline is presented in [24], and an exploration of agent spread under continuous release assumption is given in [25], [26]. In [12], the problem of plume localization was formulated using multiple intensity

sensors as the most likely sequence decoding over a fuzzy hidden Markov model. Under the assumption that each sensor has high detection and low false alarm probability, they proposed a greedy heuristic decoding algorithm with much less computational cost than Viterbi algorithm. The plume localization accuracy of the algorithm was shown to be close to the best decoder using Viterbi algorithm when tracing a single plume using randomly deployed sensors. Several other methods have been proposed, and an overview of these is available in [27].

Plume location prediction process depends mainly on the measurements of concentrations of the plume sent by sensors at different times and locations. The value of the concentration at the boundary of the plume is pre-defined depending on the nature of the plume, for example it can be defined as the values of concentrations that after which, the plume is not dangerous. A few initial data for the dynamic boundary locations and times are needed to start the prediction process. Starting from an initial location (the source of the plume), sensors sent their measurements and move in different directions as long as they keep sending values of concentrations above the pre-defined boundary threshold value as shown in Figure 4.1.

Assuming that the sensors know their locations (using GPS or any other location determination techniques), when they send measurements less than the boundary threshold, they stop moving and their location is considered to be the boundary location S_1 at time t_1. Because of the dynamic behavior of the plume, the boundary location

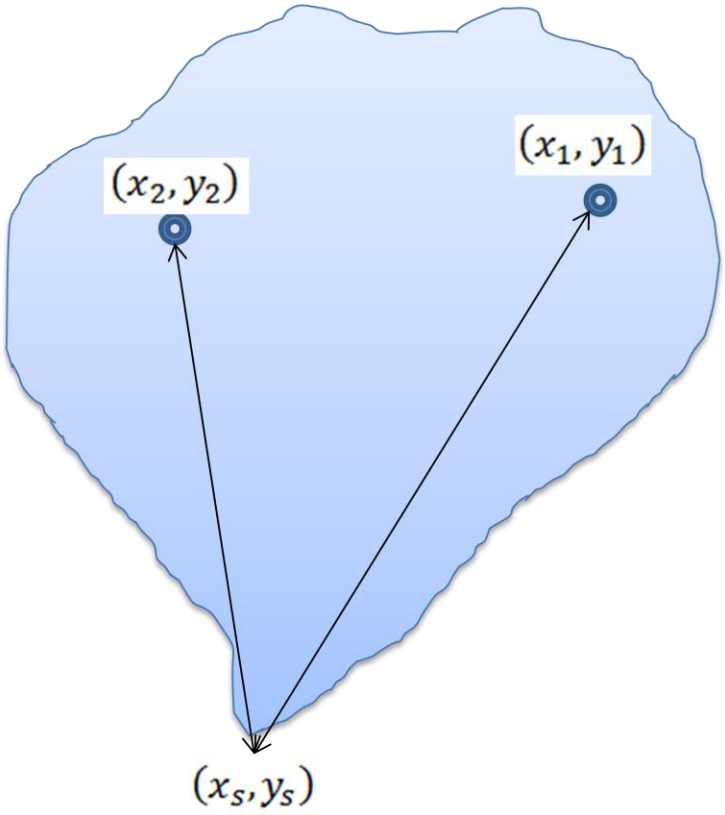

Figure 4.1: Sensors tracking plume boundary

changes with time. This is noticed when the sensor at that location starts sending values that are above the boundary threshold value, so they start moving again until they find the boundary location S_2 at time t_2. This process is repeated until we have the necessary values to begin the prediction process.

A practical plume tracking and prediction system requires the following [15]:

1- A mobile robot with the ability to negotiate the target environment (airborne, submersible, rough terrain, indoors, etc.) with appropriate speed and maneuverability.

2- Environmental sensors and appropriate control algorithms to allow the robot to safely negotiate its environment.

3- Sensors specific to the target chemical that have all of the attributes of sensitivity, selectivity, speed, and so on as outlined in Section 2.1 to the level required by the particular application.

4.2 State space model

The stochastic state space model for the plume boundary location estimation is assumed to take the following form:

$$\mathbf{x}_{k+1,i} = \mathbf{A}_{k,i}\mathbf{x}_{k,i} + \mathbf{w}_{k,i} \tag{4.1}$$

$$\mathbf{y}_{k,i} = \mathbf{C}_{k,i}\mathbf{x}_{k,i} + \mathbf{v}_{k,i} \tag{4.2}$$

where $\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$ is the state vector,

$(x_{k,i}, y_{k,i})$ is the plume location at step k for sensor i,

$(\dot{x}_{k,i}, \dot{y}_{k,i})$ is the plume velocity at step k for sensor i,

$\mathbf{y}_{k,i}$ denotes the concentration measurements step k for sensor i,

$\mathbf{w}_{k,i}, \mathbf{v}_{k,i}$ are assumed to be sequences of independent random variables with zero mean values and covariances:

$$E\mathbf{w}\mathbf{w}^T = R_1, \quad E\mathbf{v}\mathbf{v}^T = R_2, \quad E\mathbf{w}\mathbf{v}^T = R_{12}.$$

$\mathbf{A}_{k,i}, \mathbf{C}_{k,i}$ are the state space parameters to be estimated in the next section.

4.3 Parameter Estimation

Given measurements of concentration, a least squares technique is used to estimate the parameters $\mathbf{A}$ and $\mathbf{C}$ of the stochastic state space model. Equations (4.1) and (4.2) can be represented in the following expression:

$$\begin{bmatrix} \mathbf{x}_{k+1,i} \\ \mathbf{y}_{k,i} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{k,i} \\ \mathbf{C}_{k,i} \end{bmatrix} \mathbf{x}_{k,i} + \begin{bmatrix} \mathbf{w}_{k,i} \\ \mathbf{v}_{k,i} \end{bmatrix} \tag{4.3}$$

$$\mathbf{Y}_{k,i} = \mathbf{\Theta}_{k,i}\,\mathbf{x}_{k,i} + \mathbf{E}_{k,i} \tag{4.4}$$

where,

$$\mathbf{Y}_{k,i} = \begin{bmatrix} \mathbf{x}_{k+1,i} \\ \mathbf{y}_{k,i} \end{bmatrix},\ \mathbf{\Theta}_{k,i} = \begin{bmatrix} \mathbf{A}_{k,i} \\ \mathbf{C}_{k,i} \end{bmatrix},\ \mathbf{E}_{k,i} = \begin{bmatrix} \mathbf{w}_{k,i} \\ \mathbf{v}_{k,i} \end{bmatrix}$$

For the least squares problem in (4.2) we find $\mathbf{\Theta}_{k,i} = \begin{bmatrix} \mathbf{A}_{k,i} \\ \mathbf{C}_{k,i} \end{bmatrix}$ such that:

$$J_i = \frac{1}{N}\sum_{k=1}^{N} \left\| \mathbf{Y}_{k,i} - \mathbf{\Theta}_{k,i}\,\mathbf{x}_{k,i} \right\|^2 \tag{4.5}$$

is minimized. The Gauss Newton algorithm [2] is iteratively used to solve this problem. Gauss Newton algorithm was discussed in section 3.2.

4.4 State estimation using The Kalman filter

The next step is the state estimation. The Kalman filter is used to estimate and predict the states $x, y, \dot{x}, \dot{y}$ (i.e. location and plume velocity). The estimates are used as a reference input to the feedback controller feedback loop in section 4.5.

The Kalman filter is a group of mathematical equations that recursively estimates the states of a process by minimizing the mean of the squared error [28]. Kalman filter estimates past, present and future states of a process. In our state space model we have four states; the position of the boundary in x and y and the velocity at which the

boundary of the plume moves in the two dimension space. The State space equations of the process (4.1) and (4.2):

$$\mathbf{x}_{k+1,i} = \hat{\mathbf{A}}_{k,i}\mathbf{x}_{k,i} + \mathbf{w}_{k,i}$$

$$\mathbf{y}_{k,i} = \hat{\mathbf{C}}_{k,i}\mathbf{x}_{k,i} + \mathbf{v}_{k,i}$$

where $\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$ is the state vector,

$(x_{k,i}, y_{k,i})$ is the plume location at step k for sensor i,

$(\dot{x}_{k,i}, \dot{y}_{k,i})$ is the plume velocity at step k for sensor i,

$\mathbf{y}_{k,i}$ denotes the concentration measurements step k for sensor i,

$\hat{\mathbf{A}}_{k,i}, \hat{\mathbf{C}}_{k,i}$ are are the parameters that have been estimated in the previous part of this section.

$\mathbf{w}_{k,i}, \mathbf{v}_{k,i}$ are assumed to be sequences of independent random variables with zero mean values and covariances:

$$E\mathbf{w}\mathbf{w}^T = R_1, \quad E\mathbf{v}\mathbf{v}^T = R_2, \quad E\mathbf{w}\mathbf{v}^T = R_{12} \tag{4.6}$$

Then the predicted states equation is [28]:

$$\hat{\mathbf{x}}_{k+1,i} = \hat{\mathbf{A}}_{k,i}\mathbf{x}_{k,i} + \mathbf{K}_{k,i}(\mathbf{y}_{k,i} - \mathbf{C}_{k,i}\mathbf{x}_{k,i}) \tag{4.7}$$

and the predicted output equation is:

$$\hat{\mathbf{y}}_{k,i} = \hat{\mathbf{C}}_{k,i}\mathbf{x}_{k,i}\,, \tag{4.8}$$

where **K** is the Kalman filter gain matrix [28]:

$$\mathbf{K}_{k,i} = (\mathbf{A}_{k,i}\mathbf{P}_{k,i}\mathbf{C}_{k,i}^{T} + \mathbf{R}_{12})(\mathbf{C}_{k,i}\mathbf{P}_{k,i}\mathbf{C}_{k,i}^{T} + \mathbf{R}_{2})^{-1}, \tag{4.9}$$

and $\boldsymbol{P}$ is the covariance matrix of the state estimate error [28]:

$$\boldsymbol{P}_{k,i} = \boldsymbol{A}_{k,i}\boldsymbol{P}_{k,i}\boldsymbol{A}_{k,i}^{T} + \boldsymbol{R}_{1} - \mathbf{K}_{k,i}(\mathbf{A}_{k,i}\mathbf{P}_{k,i}\mathbf{C}_{k,i}^{T} + \boldsymbol{R}_{12})^{T} \tag{4.10}$$

The next section discusses the optimum position controller which is the linear quadratic regulator (LQR).

4.5 Design of a linear-quadratic regulator (LQR):

The optimum position controller that is used to control the positions of the moving sensors is an LQR controller which is a feedback controller [32]. For a discrete-time linear system described by:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_{k} + \mathbf{B}\mathbf{u}_{k} \tag{4.11}$$

with a performance index defined as [32]:

$$\mathbf{J} = \sum_{k=0}^{\infty}(\mathbf{x}_{k}^{T}\mathbf{Q}\mathbf{x}_{k} + \boldsymbol{u}_{k}^{T}\boldsymbol{R}\boldsymbol{u}_{k}) \tag{4.12}$$

the optimal control sequence minimizing the performance index is given by [32]:

$$\mathbf{u_k = -Fx_k} \tag{4.13}$$

where

$$\mathbf{F = (R + B^T PB)^{-1} B^T PA} \tag{4.14}$$

and **P** is the solution to the discrete time Riccati equation [32]:

$$\mathbf{P = Q + A^T (P - PB(R + B^T PB)^{-1} B^T P)A} \tag{4.15}$$

The next section is an overview of types of existing chemical sensors.

4.6 Chemical Sensors:

The most common sensors are the semiconductors gas sensors [21]. Chemical sensors based upon semiconductors react to various reducing gases such as carbon monoxide, hydrogen or ethanol. When exposed to air, a layer of oxygen is absorbed onto the bed of semiconductor granules that forms the sensing element [22]. When a reducing gas is present, oxidation occurs and the layer of oxygen on the sensor surface is diminished, increasing the conductivity (and therefore reducing the resistance) of the sensor [21].

Other sensors use p-type semiconductor base materials (instead of the n-type semiconductors) and react to oxidizable gases (such as O_2, NO_2 and Cl_2). All of these sensors are tuned to target specific gases by changing the operating temperature of the sensor or through the addition of impurities and catalysts. However most still have

limited selectivity, reacting strongly to the target gas, but also reacting to a number of other reducing or oxidizing gases [21]. These sensors have fast response and high sensitivity, but have the disadvantage of a return time of approximately 30 seconds [21]. Other types of sensors are discussed in details in [21], [22].

4.7 Sensor Dynamics:

We assume that the dynamics of a moving sensor has the behavior of two coupled second order differential equations, with constants that depends on the physical specifications of the sensor as well as the surrounding environment [30]:

$$M_1\ddot{x}(t)+(B+B_1)\dot{x}(t)+(K+K_1)x(t)-B\dot{y}(t)-Ky(t)=u_1(t) \qquad (4.16)$$

$$M_2\ddot{y}(t)+(B+B_2)\dot{y}(t)+(K+K_2)y(t)-B\dot{x}(t)-Kx(t)=u_2(t) \qquad (4.17)$$

where:

x, y are the 2D Cartesian coordinates,

$\dot{x}$, $\dot{y}$ are the velocities in 2D,

$\ddot{x}, \ddot{y}$ are the accelerations in 2D,

u_1, u_2 are the input signals.

The state space representation in discrete time of (4.16) and (4.17) is:

$$\mathbf{x}_{k+1,i} = \mathbf{A}\mathbf{x}_{k,i} + \mathbf{B}\mathbf{u}_{k,i} \tag{4.18}$$

$$\mathbf{y}_{k,i} = \mathbf{C}\mathbf{x}_{k,i} \tag{4.19}$$

where:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \frac{-(K+K_1)}{M_1} & \frac{-(B+B_1)}{M_1} & \frac{K}{M_1} & \frac{B}{M_1} \\ 0 & 0 & 0 & 1 \\ \frac{K}{M_2} & \frac{B}{M_2} & \frac{-(K+K_1)}{M_2} & \frac{-(B+B_1)}{M_2} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ \frac{1}{M_1} & 0 \\ 0 & 0 \\ 0 & \frac{1}{M_2} \end{bmatrix}, \text{ and } \mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

The states vector $\mathbf{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}$, inputs vector $\mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$ and the outputs vector

$\mathbf{y} = \begin{bmatrix} x \\ y \end{bmatrix}$.

Linear Quadratic Regulator (LQR) which was discussed in section 4.5 is used to calculate the optimum control inputs. The feedback loop is shown in Figure 4.2.

4.8 Boundary Prediction

Plume location prediction process depends mainly on the measurements of concentrations of the plume sent by sensors at different times and locations.

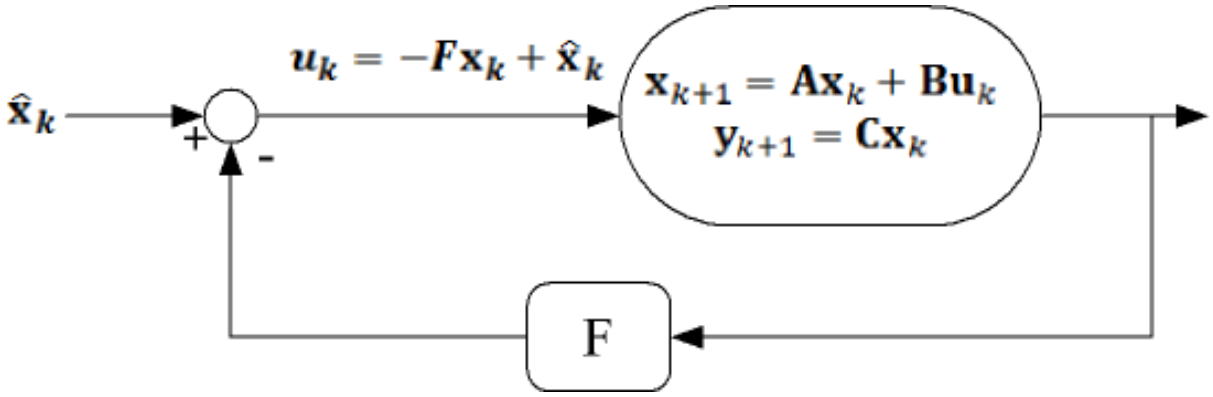

Figure 4.2 LQR controller feedback loop

A practical plume tracking and prediction system requires mobile robots with the ability to negotiate the target environment (airborne, submersible, rough terrain, indoors, etc.) with appropriate speed and maneuverability [27].

It also requires environmental sensors and appropriate control algorithms (we use LQR) to allow the robot to safely negotiate its environment. It is assumed that we have sensors specific to the target chemical that have all of the attributes of sensitivity, selectivity, speed, and so on.

The value of the concentration at the boundary of the plume is pre-defined depending on the nature of the plume, for example it can be defined as the values of concentrations that after which, the plume is not dangerous. A few initial data for the dynamic boundary locations and times are needed to start the prediction process. Starting from an initial location (the source of the plume), sensors send their measurements and move in different directions as long as they keep sending values of concentrations above the pre-defined boundary threshold value as shown in Figure 4.3. They move according to the dynamics described in section 4.7 using the optimal LQR controller described in the same section.

4.9 Full Process demonstration and simulation results:

In this experiment, three sensors move in three different directions. They keep moving until they sense a concentration value below the boundary threshold value as

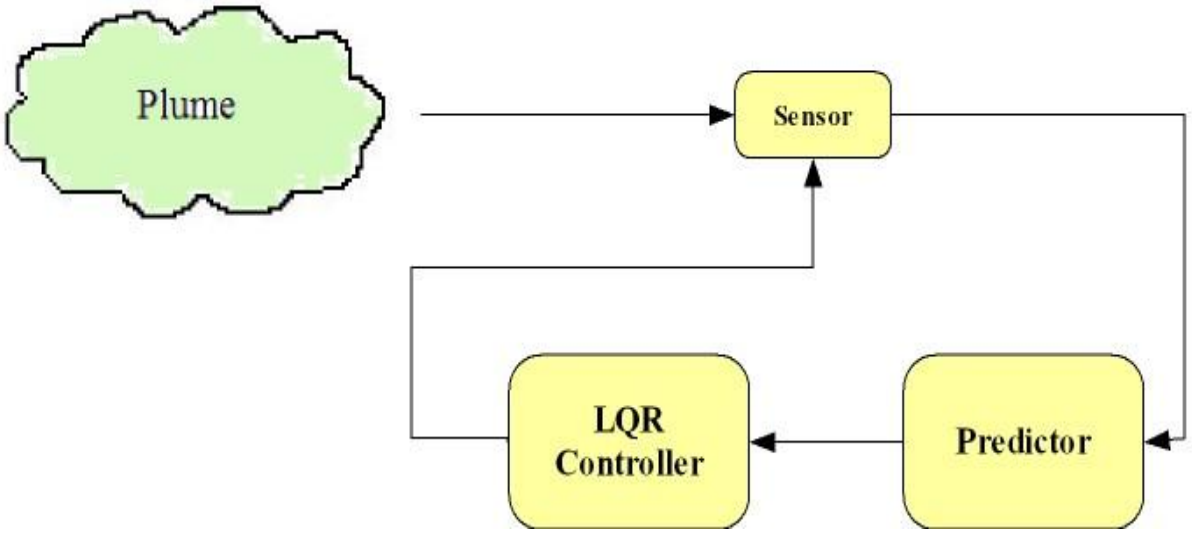

Figure 4.3 Full process block diagram. Sensors send measurements to the predictor. The predictor output is the reference input signal to the LQR controller.

shown in Figure 4.3. At this location they stop and keep sending measurements. Meanwhile, the plume spreads and the value of the measurements exceeds the boundary threshold value, then sensors move again to find the second boundary location. This process is repeated as shown in Figure 4.6 until we have sufficient data to begin the prediction process. Sensors move to the predicted locations and sense the concentration values there.

Due to the prediction error, they will be either in the plume or out of it; this is determined by the concentration measurement there. If the measurement is larger than the boundary level, the sensor is still in the plume and should move until it reaches the boundary. If the measurement is less than the boundary level, the sensor is out of boundary and should reverse its direction back until it reaches the boundary. The new location is added to the previous data measurements in order to make a new prediction. The full process is shown in Figures 4.4 to 4.10. Prediction Error using 30 min. step prediction is shown in Figure 4.11.

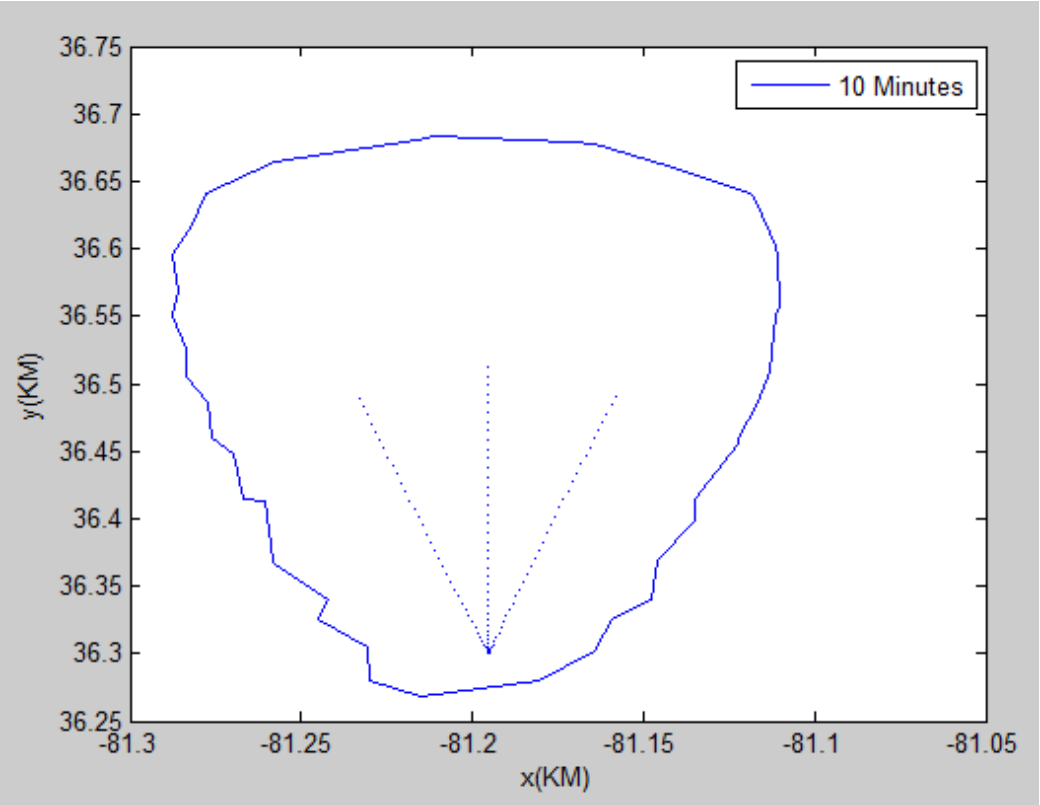

Figure 4.4 Sensors keep moving until they sense a concentration value below the boundary threshold value

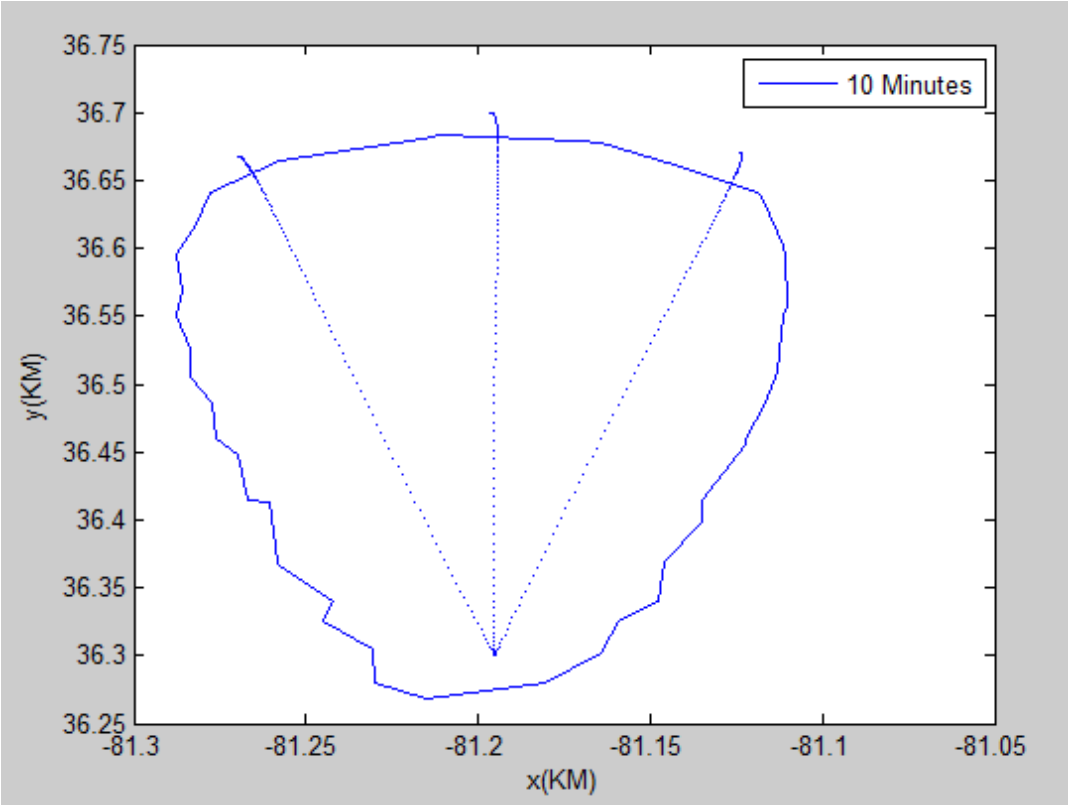

Figure 4.5 Sensors stop when they reach the boundary

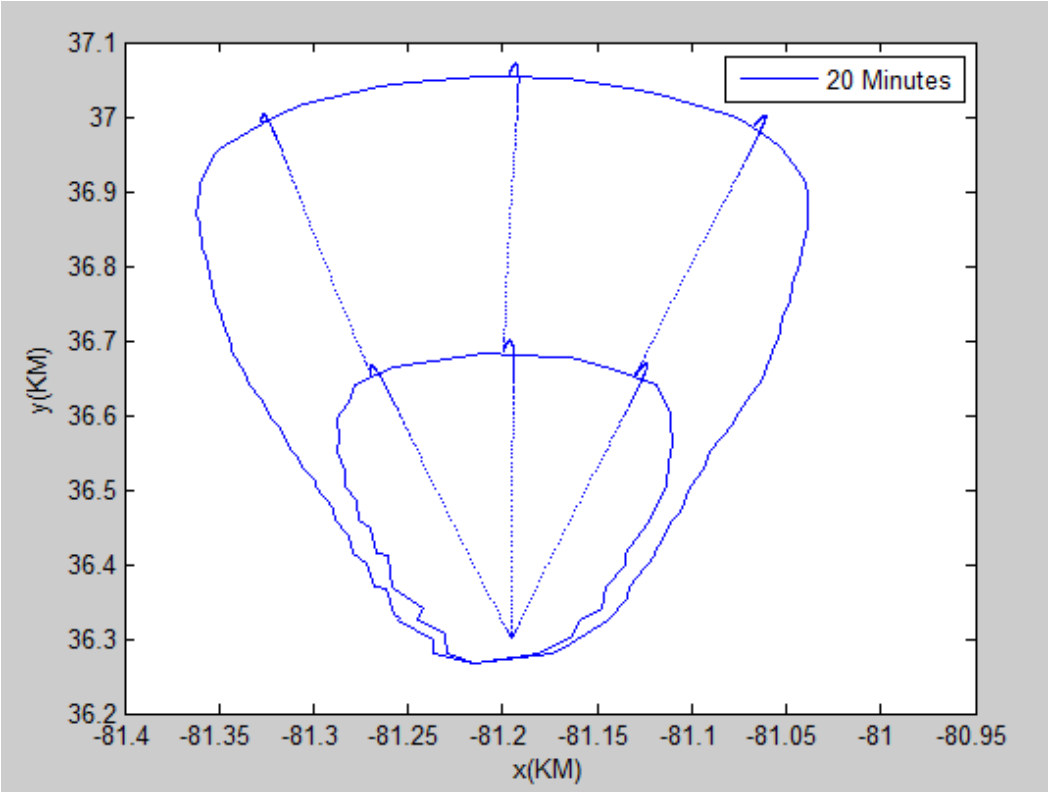

Figure 4.6 As the boundary moves, the sensors continue to track it

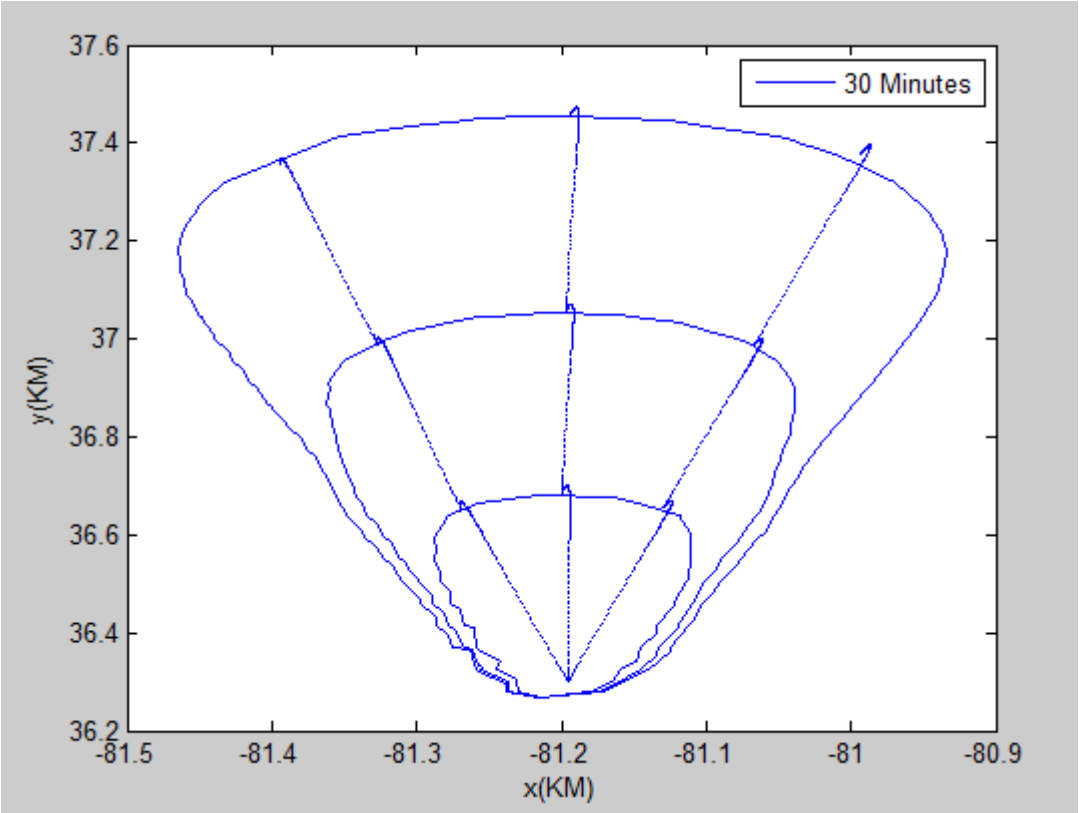

Figure 4.7 A set of initial measurements is ready to start prediction process

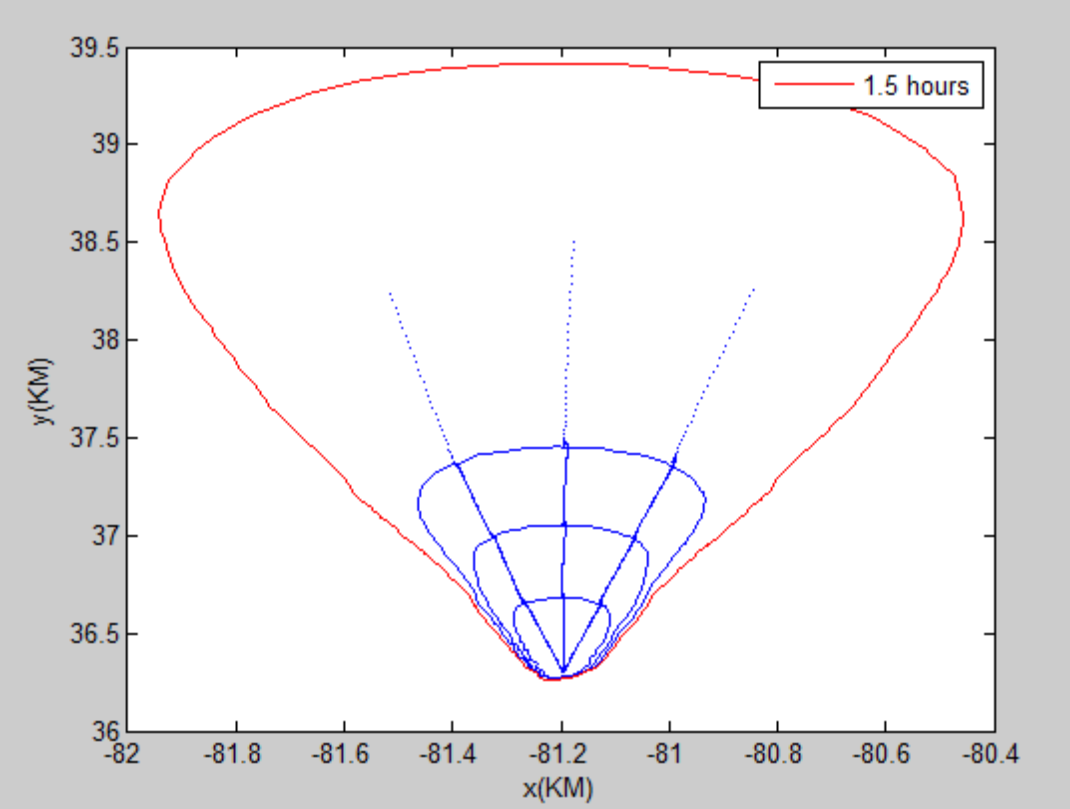

Figure 4.8 Sensors move to the predicted locations of the boundary

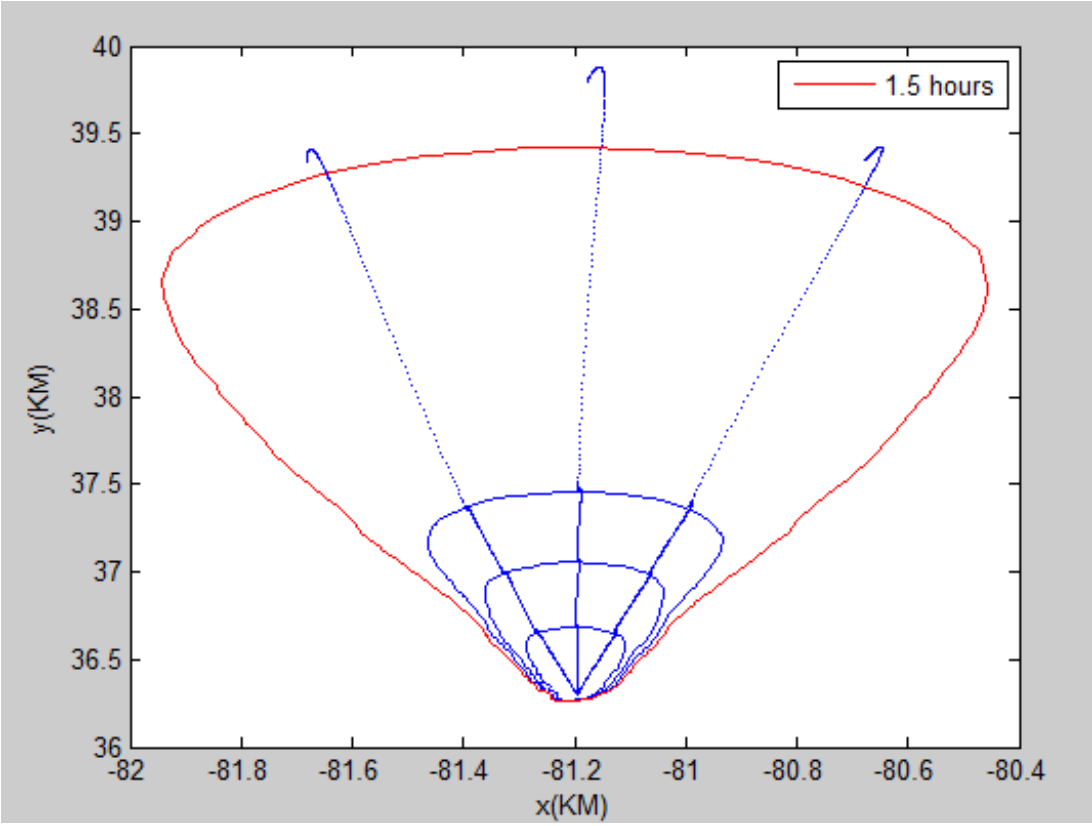

Figure 4.9 Sensors reach the predicted locations of the new boundary, but they send measurements less than the boundary threshold

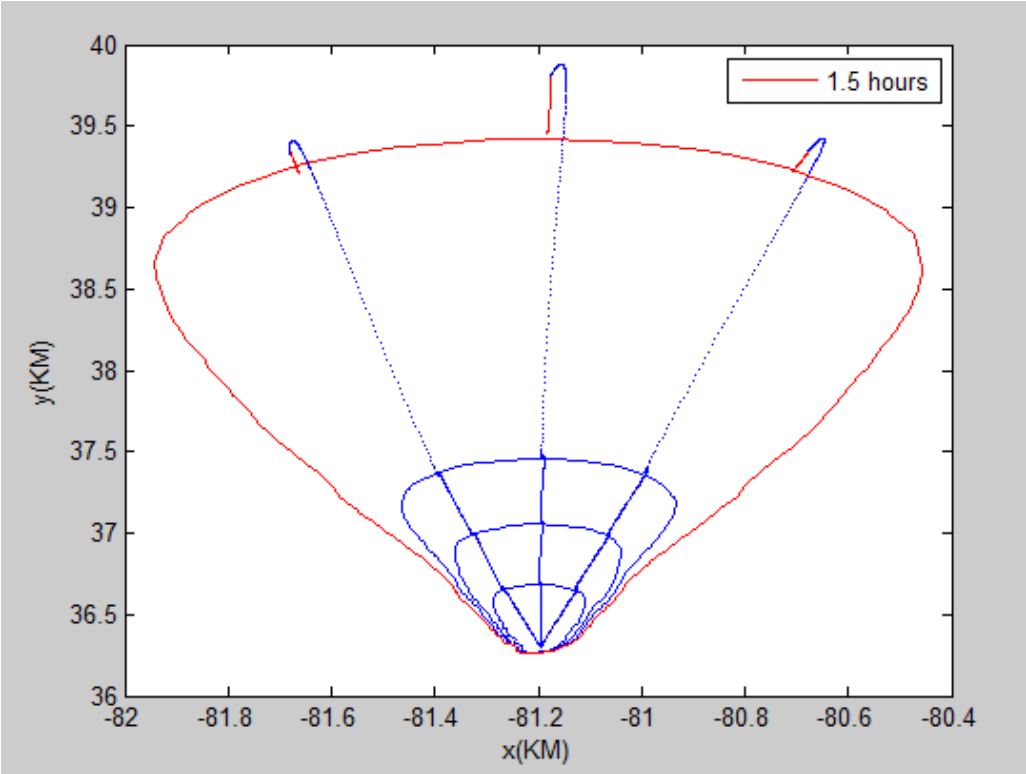

Figure 4.10 Sensors move back until they sense values above the boundary threshold, these locations are added to the previous set of boundary locations to predict the next boundary location at a desired time in the future.

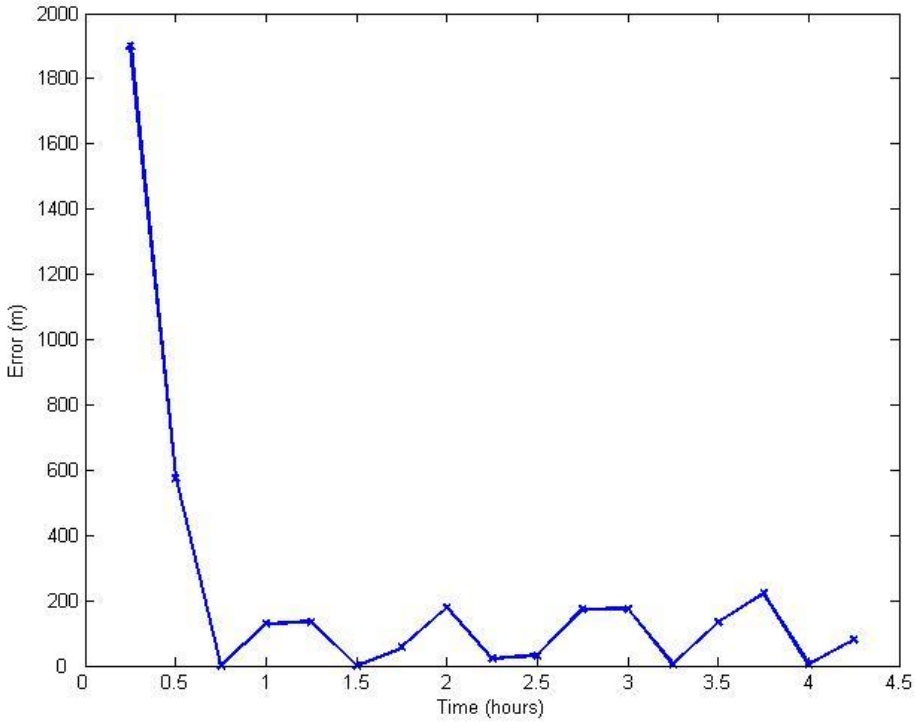

Figure 4.11 Prediction Error (30 min. step prediction)

Chapter 5

Interpolation

Interpolation process is needed to estimate the shape of the plume boundary. Up to this moment some locations of the boundary are known depending on the number of sensors used. Interpolation is a numerical method of constructing new data points within the range of a discrete set of known data points. In other words, we use interpolation to construct a function which closely fits the data points. The more data points we have (i.e. sensors), the closer shape to the real plume boundary we obtain.

There are many different interpolation methods like linear, polynomial and spline interpolation. Cubic spline interpolation will be used as it relatively gives more accurate results [29].

An spline function consists of polynomial pieces on subintervals joined together with certain continuity conditions. Formally, suppose that $n+1$ points $t_0, t_1, \ldots, t_n$ have been specified and satisfy $t_0 < t_1 < \cdots < t_n$. These points are called *knots*. Suppose also that an integer $k \geq 0$ has been prescribed. An spline function of degree *k* having *knots* $t_0, t_1, \ldots, t_n$ is a function *S* such that [29]:

a. On each interval $[t_{i-1}, t_i)$ *S* is a polynomial of degree $\leq k$

b. *S* has a continuous (*k*-1)st derivative on $[t_0, t_n]$.

Cubic splines ($k = 3$) are more often used in practice. Let:

$$(x_i, y_i) = (x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

denotes the locations of the available data points at the plume boundary. A cubic spline *S* is to be constructed to interpolate the set of those available locations. On each interval $[x_0, x_1], [x_1, x_2], \ldots, [x_{n-1}, x_n]$, *S* is given by a different cubic polynomial. Let S_i be the cubic polynomial that represents *S* on $[x_{i-1}, x_i]$. Thus [29]:

$$S(x) = \begin{cases} S_0(x) & x \in [x_0, x_1] \\ S_1(x) & x \in [x_1, x_2] \\ \vdots & \vdots \\ S_{n-1}(x) & x \in [x_{n-1}, x_n] \end{cases} \tag{5.1}$$

The polynomials S_{i-1} and S_i interpolate the same value at the point x_i and therefore

$$S_{i-1}(x_i) = y_i = S_i(x_i) \qquad (1 \le i \le n-1) \tag{5.2}$$

Hence, *S* is automatically continuous. Moreover, the first and second derivatives *S*' and *S*" are assumed to be continuous, and these conditions will be used in the derivation of the cubic spline function. After some derivations, detailed in [29] the final cubic spline interpolation is

$$S_i(x) = y_i + (x - x_i)\left[c_i + (x - x_i)[B_i + (x - x_i)A_i]\right] \tag{5.3}$$

Where

$$A_i = \frac{1}{6h_i}(z_{i-1} - z_i) \tag{5.4}$$

$$B_i = \frac{z_i}{2} \tag{5.5}$$

$$c_i = -\frac{h_i}{6}z_{i+1} - \frac{h_i}{3}z_i + \frac{1}{h_i}(y_{i+1} - y_i) \tag{5.6}$$

$$z_i = S_i''(x_i), \qquad z_{i+1} = S_i''(x_{i+1})$$

Cubic spline interpolation using three data point from sensor measurements is shown in Figures 5.1 and 5.3. The more sensors used, the more accurate boundary shape is obtained.

Interpolation is the last step of the plume tracking process; data is now ready to be sent to the interested destinations like the police department or the nearest fire station.

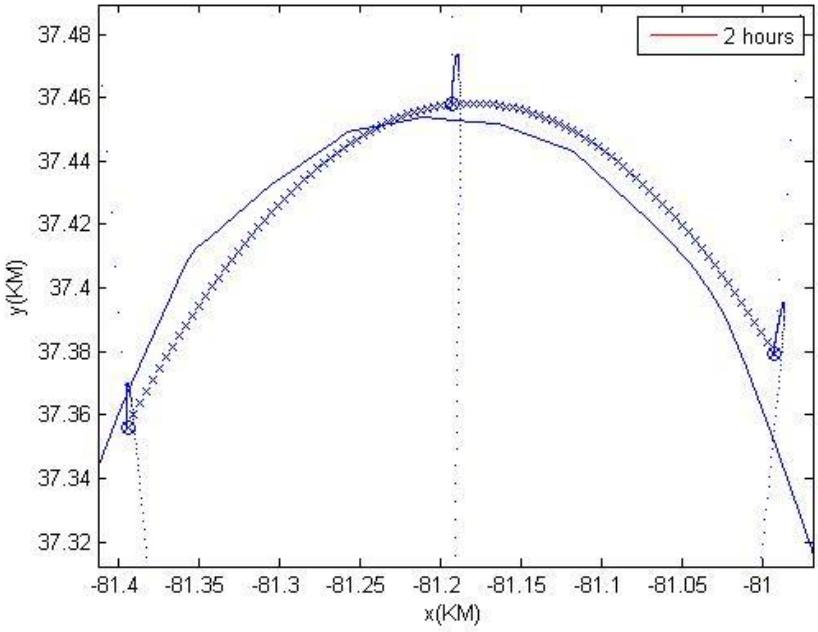

Figure 5.1 Interpolation using 3 sensors

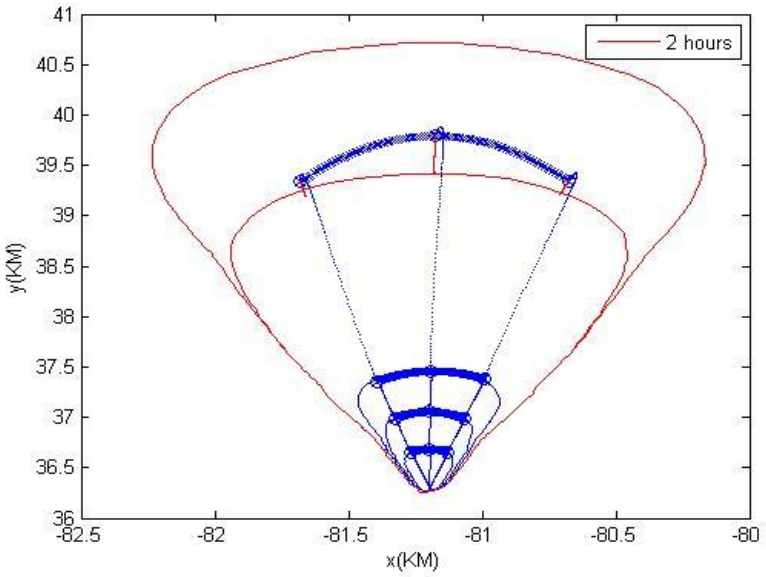

Figure 5.2 Interpolation of predicted boundaries

Chapter 6

Conclusion and Future Work

The source of a chemical plume has been located using two methods; Nonlinear Gauss-Newton least squares method and Kiefer-Wolfowitz stochastic approximation algorithm. Plume data have been generated using SCIPUFF; a well known plume model that uses multiple Gaussian plume models to generate the desired plume. It has been shown that stochastic approximation methods give more accurate results than least square methods when dealing with noise corrupted data. Indeed, Stochastic Approximation algorithms are derived for this purpose. Convergence of Stochastic Approximation is expected and depends on the noise variance of the data.

Plume boundary tracking using mobile sensors has been performed starting from the plume source. A novel state space plume model is used in the estimation and prediction of the plume evolution. The predicted states of the plume progression are used as reference signals to deploy the sensors using optimal controllers. New measurements obtained from the sensors are then used to update the plume state estimates recursively at every time step. The estimates are further used in estimating the plume boundary thanks to spline interpolation. The process is repeated till the sensors converge to the plume boundary.

In our future work, we plan to localize the source of chemical plumes using other Stochastic Approximation algorithms like Stochastic Gradient Descent algorithm [12] and The SPSA algorithm (Simultaneous Perturbation Stochastic Approximation) [13]. These stochastic approximation algorithms converge faster than Kiefer-Wolfowitz algorithm that is used in this thesis [ ]. We will also use the results of source localization as an initial starting point in the boundary tracking process with the knowledge of the plume source location, along with all the needed data like the weather conditions and the nature of the gas.

Currently the sensors are controlled independently. We plan to develop autonomous consensus and rendez-vous distributed coordination motion algorithms for the sensors, to achieve tracking. A test bed in the lab using a non poisoning chemical, such as, water vapor, will be built to implement the developed methods.

References

[1]. M. P. Michaelides, C. G. Panayiotou, "*Plume source position estimation using sensor networks*". Proceedings of the 13th Mediterranean Conference on Control an Automation[C].Limassol Cyprus:IEEE CNF,2005.731-736.

[2]. Bjorck, A., *Numerical methods for least squares problems*. SIAM, Philadelphia, 1993.

[3]. H. Robbins, S. Monro, (1951). "*A Stochastic Approximation Method,"* Annals of Math Stat., 22,400 407.

[4]. J. Kiefer and J. Wolfowitz, "*Stochastic Estimation of the Maximum of a Regression Function*". Annals of Mathematical Statistics 23, #3 (September 1952), pp. 462–466.

[5]. H. J. Kushner, G. G. Yin, "*Stochastic Approximation and Recursive Algorithms and applications*", second edition, Springer, 2003.

[6]. R. I. Sykes and R. S. Gabruk, "*A second order closure model for the effect of averaging time on turbulent plume dispersion".* J. Appl. Met., 36, 165-184.

[7]. E. Holzbecher, "*Environmental modeling using matlab*". Springer 2007.

[8]. R. A. Russell, "Survey of robotic applications for odor-sensing technology," *The International Journal of Robotics Research*, vol. 20, no. 2, pp. 144- 162, February 2001.

[9]. Pasquill, F. (1961). *The estimation of the dispersion of windborne material*, The Meteorological Magazine, vol 90, No. 1063, pp 33-49

[10]. Bosanquet, C.H. and Pearson, J.L. (1936).*The spread of smoke and gases from chimney*, Trans. Faraday Soc., 32:1249.

[11]. X. Kuang and H. Shao Maximum, *"Likelihood Localization Algorithm Using Wireless Sensor Networks"* Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06).

[12]. Huimin Chen, *"Plume Localization Using Fuzzy Hidden Markov Model: An Efficient Decoding Method"* Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International.

[13]. Cauwenberghs, G. A fast stochastic error-descent algorithm for supervised learning and optimization. In *Advances in Neural Information Processing Systems 5*, San Mateo, CA: Morgan Kaufman Publishers: 244–251, 1993.

[14]. *Introduction to Stochastic Search and Optimization* by James C. Spall, ISBN 0-471-33052-3, 2003

[15] C. D. Jones, "On the structure of instantaneous plumes in the atmosphere," Journal of Hazardous Materials 7: 87--112, 1983.

[16] J. Murlis, J. S. Elkinton, R. T. and Cardé, "Odor plumes and how insects use them," Annual Review of Entomology 37: 505--532, 1992.

[17] B. I. Shraiman and E. D. Siggia, "Scalar Turbulence," Nature, 405: 639--646, 2000.

[18] K. R. Mylne, "Concentration fluctuation measurements in a plume dispersing in a stable surface layer," Boundary-Layer Meteorology, 60: 15--48, 1992.

[19] O. G. Sutton, “The problem of diffusion in the lower atmosphere,” *Q. J. R. Meteorol. Soc.*, vol. 73, pp. 257–281, 1947.

[20] Sahyoun, S.M. Djouadi and H. Qi, *Source Localization using Stochastic Approximation and Least Squares Methods*, the 2nd Mediterranean Conference on Intelligent Systems and Automation (CISA'09).

[21] H. Nanto and J. R. Stetter, *Handbook of Machine Olfaction. Electronic Nose Technology*. Wiley-VCH, 2002, ch. Introduction to chemosensors.

[22] H. Ishida, T. Nakamoto, T. Moriizumi, T. Kikas, and J. Janata, "Plumetracking robots: A new application of chemical sensors," *Biological Bulletin*, pp. 222-226, April 2001.

[23] H. Versteeg and W. Malalasekera, *Introduction to computational fluid dynamics. The finite volume method*. Longman Scientific and Technical, 1995.

[24] R. M. P. Kathirgamanathan, R. McKibbin, “Source term estimation of pollution from an instantaneous point source,” *Research Letters in the Information and Mathematical Sciences*, vol. 3, no. 1, April 2002, pp. 59–67.

[25] ——, “Sourse release-rate estimation of atmospheric pollution from non-steady point source - part [1]: Source at a known location,” *Research Letters in the Information and Mathematical Sciences*, vol. 5, June 2003, pp. 71–84.

[26] ——, “Source release-rate estimation of atmospheric pollution from a non-steady point source - part [2]: Source at an unknown location,” *Research Letters in the Information and Mathematical Sciences*, vol. 5, June 2003, pp. 85–118.

[27] V. Christopoulos and S.I. Roumeliotis, “Adaptive Sensing for Instantaneous Gas Release Parameter Estimation”, in *Proc. 2005 International Conference on Robotics and Automation*, Barcelona, Spain, April 18-22, 2005, pp. 4461-4467.

[28] R. Kalman, “A New Approach to Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, March 1960, pp. 35-46.

[29] Ahlberg, Nielson, and Walsh, *The Theory of Splines and Their Applications*, 1967.

[30] P. J. Antsaklis and A. N. Michel. *A Linear Systems Primer*, Birkhäuser, Boston, 2007. ISBN: 0817644601.

[31] Infod

[32] Kwakernaak, Huibert and Sivan, Raphael (1972). *Linear Optimal Control Systems. First Edition*. Wiley-Interscience. ISBN 0-471-511102

Appendix

Publications

Sahyoun, S.M. Djouadi and H. Qi, Source Localization using Stochastic Approximation and Least Squares Methods, accepted for presentation in the 2nd Mediterranean Conference on Intelligent Systems and Automation (CISA'09).

S. Sahyoun, S.M. Djouadi and H. Qi, Plume Estimation and Tracking using Mobile Sensors. Submitted to IEEE Conference on Decision and Control 2009.

Vita

Samir S. Sahyoun was awarded a BSc in Electrical Engineering from Misr University for Science and Technology, Egypt in 2003. After ranking in the top 1% students, he was awarded a Fulbright scholarship in 2005 to study in the US. In 2006 He joined the Department of Electrical Engineering and Computer Science at the University of Tennessee, where he is pursuing a PhD degree. Mr. Sahyoun is a member of the IEEE. His research interests include control and estimation, chemical threats and countermeasures, computational fluid dynamics and model reduction.