SIMULTANEOUS ROBOT LOCALIZATION AND MAPPING OF PARAMETERIZED SPATIO-TEMPORAL FIELDS USING MULTI-SCALE ADAPTIVE SAMPLING

by

MUHAMMAD FAIZAN MYSOREWALA

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2008

In the name of God, the Benevolent, the Merciful.
(Quran, 1:1)

This is the God, Other than which there is no deity: Knower of the invisible
and the evident, the Benevolent, the Merciful.
This is the God, other which there is no deity: The Sovereign, the Holy, Peace,
the Giver of Safety, the Protector, the Almighty, the Omnipotent, the
Overwhelming; glory to God, beyond any association they attribute.
This is God, the Originator, the Creator, the Shaper, to whom refer the most
Beautiful names, celebrated by everything in the heavens and on earth,
being the Almighty, the Perfectly Wise.
(Quran, 59:22-24)

There is no ability or power except through God.
(Sahih al-Bukhari, 3883)

ACKNOWLEDGEMENTS

"Whoever has not thanked people, has not thanked God", said the Prophet Muhammad (Peace be upon him). So I would like to express my sincere appreciation and gratitude to my supervising professor Prof. Dan Popa, and the Director of U.T. Arlington's Automation & Robotics Research Institute (ARRI), Prof. Harry Stephanou, for their guidance, patience, encouragement and financial support which made the completion of this dissertation possible. I am also grateful to my other committee members Prof. Frank Lewis, Prof. Kamesh Subbarao, and Prof. Kai-Shing Yeung for their advice, time and effort expended to improve this work. In addition I wish to thank my colleagues at ARRI and specially the ones in the DIAL Lab, Koushil Sreenath, Jaymala Ghadigaonkar and Rohit Talati, for their contribution and companionship. I am also grateful to the helpful staff of ARRI, especially Norman Spayd, Kathleen Elfrink and Sara Bodine.

I am eternally indebted to my parents Siddiq and Maimoona for their prayers, worries, and support during my long years abroad. Finally my immense gratitude goes to my wife Javeria for her patience, support and prayers during the course of this work.

During this time, I enjoyed the good company of friends at UTA such as: Shahid, Zakir, Zeeshan, Amin, Faraz, Najeeb, Rizwan, Adnan, Zeerak, Bilal, Imran,

Muneeb, Faisal, Aamir, Sami, Naveed, Basu, Fahad, Rana, Tanveer, Hasan, Mohsin Kareem, Aqeel, Raji, Abdul-Wahab, Mohsin Rizwan, Farhan, Abid, and Kashif.

April 3, 2008

ABSTRACT

SIMULTANEOUS ROBOT LOCALIZATION AND MAPPING OF PARAMETERIZED SPATIO-TEMPORAL FIELDS USING MULTI-SCALE ADAPTIVE SAMPLING

Muhammad Faizan Mysorewala, PhD.

The University of Texas at Arlington, 2008

Supervising Professor: Dr. Dan O. Popa

This dissertation presents a Multi-scale Adaptive Sampling (AS) framework for combining measurements arriving from mobile robotic sensors of different scales, rates and accuracies, in order to reconstruct a parametric spatio-temporal field. The proposed sampling algorithm, "EKF-NN-GAS", is based on the Extended Kalman Filter (EKF), Radial Basis Function (RBF) Neural Networks and Greedy Search Heuristics. This novel AS algorithm responds to real-time measurements by continuously directing robots to locations most likely to yield maximum information about the sensed field. EKF is used to derive quantitative information measures for sampling locations. In addition, the localization uncertainty of the robots is minimized by combining the

location states and field parameters in a Joint-EKF formulation. This feature is critical in GPS-denied environment such as inside buildings or underwater.

Secondary objectives such as sampling duration, computational cost and energy are minimized by adding several extensions called "Greedy Adaptive Sampling" (GAS) heuristics. The issue of thorough sampling in dense regions is addressed using *Clustered Adaptive Sampling*. Drawbacks of local searching approach used in GAS are overcome with *Non-uniform Grid Size AS* and *Multi-step AS*. The proposed sampling algorithms are compared with traditional raster-scanning through many examples. Results indicate that that the proposed parametric algorithm provides faster convergence with less number of samples. This dissertation also addresses issues of efficient partitioning of the sampling area, distribution of computations and communication for adaptive sampling with multiple robots. The performance of the algorithm was experimentally validated using indoor multi-robot testbed at ARRI's DIAL lab (Distributed Intelligence and Autonomy Lab).

A real world scenario of mapping of forest fires is addressed in this thesis in conjunction with the proposed sampling algorithm. Our strategy combines measurements arriving at different times from sensors with different field of view (FOV) and resolution, such as ground, air-borne and space-borne observation platforms. In practice, such robots could be equipped with thermal imaging, topographic mapping and other sensors for measuring environmental conditions.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Mobile robots are being increasingly used on land, underwater, and in air, as sensor carrying agents to perform sampling missions, such as searching for harmful biological and chemical agents, search and rescue in disaster areas, and environmental mapping and monitoring. Sampling problems are central aspects of deployment because complete coverage is not possible if the environment is large, has only a few “hotspots”, or if the sampling costs are high. Therefore, it is desirable to build robot teams that can coordinate to maximize the amount of sensor data taken at these hotspots while minimizing resource costs.

In general for environmental sampling, different types of sensors provide different sampling ranges and resolutions. For example, a thermal image taken from an aircraft or a satellite provides a high-spatial field of view (FOV) and low-resolution coverage of the area of interest, whereas a measurement taken from a point measurement thermal sensor on a robot provides a more accurate local reading. This involves estimating a space-time field spread over wide area using multiple scale, rates and accuracies of robotic sensing. An example of a relevant application is the deployment of multiple underwater vehicles for the environmental monitoring of large

bodies of water, such as oceans, harbors, lakes, rivers and estuaries. Environmental models and maps can be created by repeated measurements of physical characteristics, such as water temperature, dissolved oxygen, current strength and direction, or bathymetry.

Another motivating application is wildfire monitoring, which is a major natural disturbance with a tremendous impact on environment, human and wildlife, ecosystem, weather and climate. There appears to be an increasing trend of natural fire activity. In the recent October 2007 California fires several people died, many were injured, and around one million were evacuated from their homes. The fire destroyed approximately 1,500 homes, burned approximately 800 square miles of land and resulted in overall estimated loss of more than 1 billion dollars just in the San Diego area alone [54]. During this time, fire suppression resources were in high demand and there was critical need for air tankers, crews and personnel, but the speed and uncertainty of fire spread surprised forecasters in several instances.

Fire management strategies have changed today with the introduction of new fire suppression tools. The use of advanced technology helps in fire mapping through satellite imagery, accurate weather forecasts, and fire behavior modeling. Air-borne and space-borne observation platforms, air-tankers, bulldozers and tractor plows are commonly used for fire mapping and control, and they improve the safety of firefighters and the general public. Tractors are used for clearing vegetation and building fireline faster more efficiently than human firefighters. Rescue operations consist of searching

for, and removing trapped occupants in hazardous conditions. Although the use of advanced technology has reduced human involvement, self-preservation of the personnel manning the observation platforms remains a critical issue. One of the best ways to reduce the risk for firefighting personnel is to gather information about the spread of the fire in real time using autonomous robots, however, such technology has not yet been adopted.

However, it seems entirely plausible that in the near future, tracking and prediction of the spread of forest fires will be achieved by mobile robots equipped with sensors for measuring environmental conditions such as temperature, fire intensity, humidity, slope, wind strength and direction, etc. Fire spread and intensity information "measures" should then be used to reposition the robots in order to achieve optimal sampling of this spatio-temporal field, e.g. to describe its spread as accurately, and in the shortest time possible.

In the context of optimizing the sampling of spatio-temporal fields with mobile robots, several Adaptive Sampling (AS) algorithms have recently been proposed [4, 15, 31, 34, 41]. The term "adaptive" refers to choosing the sampling points based on the amount of information they provide about the spatio-temporal distribution being mapped. Examples of spatially distributed fields that can be monitored using adaptive sampling are salinity in lakes [34], humidity in forests [41], and chemical leaks in buildings, as it is done in odor sampling [4, 17, 38].

Sampling using mobile robots is a typical example of a real-time density

estimation problem, and has been covered in detail in past work [39-40]. The sensor fusion approaches for density estimation can be classified into 3 different categories. The first is building physical parametric models, the second is parametric and non-parametric feature-based inference, both of which involve clustering of observations, and the third is cognitive-based modeling.

In general, a spatio-temporal distribution can be modeled in a parametric or non-parametric manner. Strictly non-parametric field descriptions are accurate, as they don't assume any functional form of distribution in advance. However, they require a lot of samples to reconstruct the distribution, which seems difficult to apply in practice to situations involving time and energy constraints, and for time-varying fields. An example of solutions with non-parametric distribution assumptions is discrete landmark mapping [7] that involves exploration, instead of planned sampling missions. Other examples are chemical source localization [38] using *Hill (*or *Gradient) Climbing Approach,* and *Adaptive Cluster Sampling* [26] for wide-area sampling.

In contrast, parametric algorithms are computationally efficient and fairly easy to run onboard mobile robots in a distributed fashion. Such models can sometimes be constructed from physical or numerical models such as in the case of weather prediction. With no apriori information, the sampling problem becomes an exploration problem in the beginning, until enough samples are taken and distribution assumptions can be made. The advantage of the parametric model is that, in fact, it can also include a parametric model of sensor observable data and corresponding measurement

uncertainties. Also, the uncertainty in the field estimate can be measured indirectly through uncertainty in the distribution parameters. Furthermore, parametric modeling works better for time-varying fields, in which the field variation can be represented through parameters instead of the entire density function. The main limitation of parametric representations is modeling uncertainty which adds another source of estimation error.

A problem which naturally arises in the process of sampling is making sure that the measurements of sensors are correlated with their position, and that the data from multiple sensors is fused efficiently. Multiple vehicle localization and sensor fusion are by now classic problems in robotics, and there has been considerable progress in the two decades in these areas [10, 37]. Furthermore, distributed field variable estimation is relevant to charting and prediction in oceanography and meteorology, and has also received considerable attention [6]. In both contexts, measurement uncertainty can be addressed using Kalman-Filter Estimation [23]. Recently, Sanderson and Popa described a combined multi-agent AS problem coupling wireless sensor nodes with mobile robots, and using information measures to reposition the robots in order to achieve near-optimal sampling of a distributed field [34]. Unlike other non-parametric sampling methods [15, 41], this approach requires a parametric field description of the sampled field, and a dynamic model for robots. The advantage of this approach was the inclusion of vehicle localization uncertainty, but the disadvantage was the fact that it could only handle very simple field models.

Localization uncertainties are especially relevant for robotic vehicles where dead-reckoning errors are high such as in the case of GPS-denied environments. The field model parameters are combined with uncertainty in the estimation of the robot localization using the overall error covariance. Therefore, localization uncertainty is reduced by building accurate models of distributed fields and vice versa. For instance, if a robot is sampling an unknown field, but its location is accurately known, a distributed parameter field model can be constructed by taking repeated field samples. Later on, this field model can be used to reduce the robot localization error.

In this thesis we follow a similar parametric approach, and use sensor measurements to improve both the field estimate and the localization of the robot. Since this kind of formulation requires a parametric model for a potentially unknown and complex field, we considerably extend and expand on results from prior work. In particular, we address challenges related to sampling of complex, non-linear, and time-varying space-time fields, we investigate considerations of computational complexity, communication overhead, time and energy constraints, and sampling using multiple robots.

Two popular non-adaptive sampling methods are *Systematic Raster-Scan Sampling* and *Random Sampling*. In random sampling, all the points have equal probability of being selected irrespective of the field variation. Raster scanning acquires measurements at uniform intervals and ignores the field variability. Another sampling scheme, called *Stratified or Clustered Sampling,* involves classifying the sampling area

into non-overlapping stratas based on apriori information about the field. This could be accomplished either by parametric or non-parametric classification, and used to sample only in hotspot areas. In practice, classification is also needed because many field distributions are multimodal and require some parameters before sampling starts. Examples of multimodal distributions are complex forest fires and chemical gas leaks dominated by turbulence.

Environmental distributions such as forest fires, chemical source leaks and temperature variations in oceans are examples of phenomena for which exact nonlinear model descriptions are unattainable due to the high-level of complexity. Due to time and energy-critical nature of some of these sampling scenarios, simply requiring that the robots perform a raster scan or randomly sample the field is clearly sub-optimal and highly inefficient. Moreover, many of these time-varying distributions encompass a wide area, and must be observed from sensors of multiple size scales, rates and accuracies.

In order to address these challenges we propose a *Multi-scale Adaptive Sampling* approach with a parametric description of the field. In this approach, sampling strategies continuously adapt in response to real-time measurements from sensors of different scales. This scheme relies on building parametric models from high-spatial sensor measurements that are less accurate, and then improving the models by more accurate spot measurements. The Extended Kalman Filter (EKF) is used to derive quantitative information measure for sampling locations mostly likely to yield optimal

information.

Five approaches are proposed in this thesis for sampling of parameterized fields:

- The first one is *Global Search Adaptive Sampling,* in which sampling points are selected in real time to gain maximum information about the estimates of unknown parameters and entire search space is used to look for optimal sampling location. This approach ignores the time (and distance) cost of sampling.
- The second one is a local search heuristic, *Greedy Adaptive Sampling* (GAS), which is a trade-off between gaining maximum information and minimizing the traveling time and energy associated with sampling. Robot navigation vectors are selected by searching in the neighborhood of the currently sampled location.
- The third approach is *Clustered Adaptive Sampling,* in which low-resolution apriori information is used to parameterize the field using Radial Basis Functions Neural Networks. The centers of the RBF Gaussians are selected using self-organized placement scheme to ensure that more samples are taken in high-variance regions.
- The fourth approach proposes a *Non-uniform Grid Size Greedy Adaptive Sampling* (also called *Non-systematic Sampling*) in which the grid size varies with local variance of the distribution. Grids are smaller in high-variance areas and larger in low-variance areas.
- The fifth approach considers a *Multi-step Greedy Adaptive Sampling Scheme* (as opposed to a single-step GAS), in which an exhaustive search is performed with a k-step horizon.

In this thesis, a real world scenario of mapping of forest fires is considered in conjunction with our sampling algorithms. The sampling strategy combines measurements arriving at different times from sensors of different field of view (FOV) and resolution, such as ground, air-borne and space-borne observation platforms. In practice, such robots could be equipped with thermal imaging, topographic mapping and other sensors for measuring environmental conditions, as illustrated in Figure 1.1.

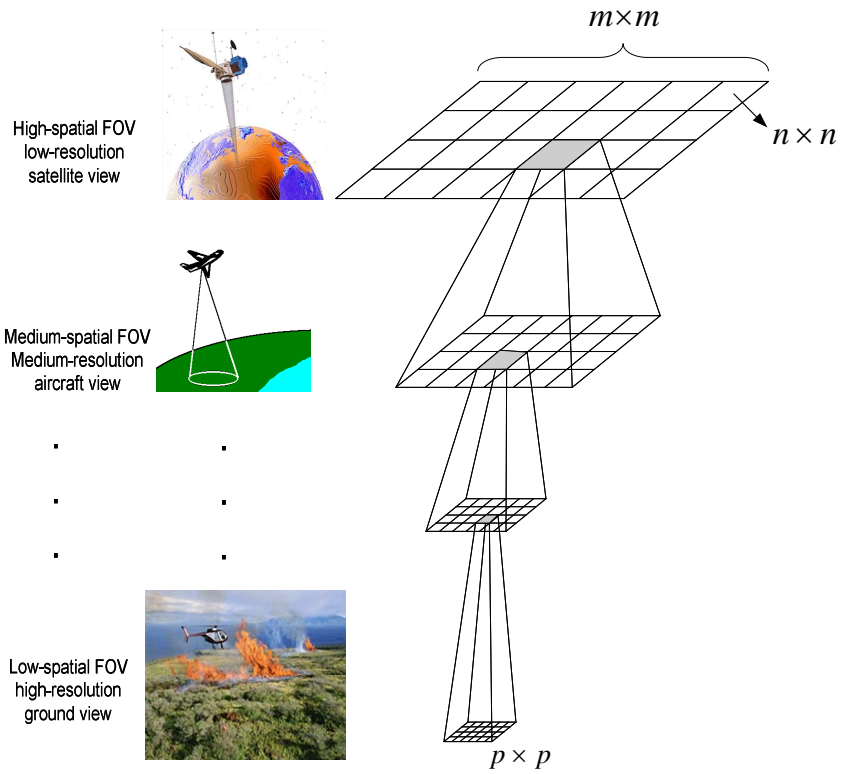


Figure 1.1 Illustration of the multi-scale sampling approach

1.1 Problem statement and approach

This thesis addresses the following problem:

Given *N* robotic vehicles sampling a spatio-temporal field whose distribution is not known, where should the vehicles be directed to sample such that:

i. The uncertainty in the knowledge of the field is minimized.

ii. Localization uncertainty of the robotic vehicles is minimized.

iii. Additional secondary objectives are minimized, such as energy, sampling duration, etc.

iv. Simultaneous sampling is done using robots covering multiple scales and rate of sensing, in a scalable fashion, by addressing issues related to distributed computation and communication environments.

The outline of the proposed approach for solving this problem is described below:

- First, a field distribution model, e.g. the initial estimate, is acquired by clustering and unsupervised RBF parametric classification of the sensory data obtained by robots with higher instantaneous field of view (IFOV). This ensures that more RBFs are placed in "hot spot" regions.
- Next, uncertainty in the knowledge of the field is minimized by adaptive sampling, using the Kalman Filter error covariance of the estimates as an information measure. Localization uncertainty is also minimized by fusing the location states and field parameters in a Joint Extended Kalman Filter.
- Next, secondary objectives such as sampling duration, computational cost and energy are incorporated into the algorithm through the use of heuristics such as *Greedy Adaptive Sampling* (GAS).
- Finally, sensing and computations are distributed among multiple robotic

nodes that wirelessly share field estimates and localization information.

This thesis contains both algorithmic and experimental developments that demonstrate the effectiveness of the proposed approach including:

- It is demonstrated through extensive simulations that our approach along with self-organized clustering is efficient way to accomplish wide-area sampling using measurements from robots of different scales. The approach is scalable, and computations can be distributed among the sampling agents.
- It is demonstrated through experiments and simulations that the proposed EKF based Adaptive and Greedy Adaptive Sampling algorithms provide field reconstruction in significantly less time and number of samples, compared to just random sampling or raster-scanning. This is due to sampling at locations most likely to yield maximum information about the sensed field of interest.
- It is demonstrated experimentally that the localization uncertainty of the robots can be reduced by fusing the robot location states and field parameters in a Joint-EKF and using the field sensor measurement only for simultaneous localization and field parameter estimation. The convergence of this algorithm depends on the initial conditions, because of the inherent linearization of the Extended Kalman Filter.
- It is demonstrated through simulation and experiments results that our proposed multi-scale sampling algorithm is an effective approach for sampling of wide-area time-varying fields by combining sensor measurements of multiple

scales, rates and accuracies. The convergence of this algorithm requires that the rate of spatial field evolution be much slower than sampling (robot) speed.

- We provided a comprehensive simulated verification of our algorithm for mapping spatio-temporal forest fire scenarios, with the help of realistic Cellular Automata (CA) fire spread distributions, used as "truth models". The sampling performance was also experimentally validated with small in-door mobile robots that navigate over a "virtual fire" projected on the floor from a ceiling-mounted projector to emulate the sampling mission performed by aerial robots.

1.2 Contribution of the dissertation

The primary contribution of this thesis is the formulation of a novel parametric multi-scale, multi-rate and multi-robot sampling algorithm called "EKF-NN-GAS" for complex field estimation using RBF Neural Networks based parametric classification, EKF based parameter estimation and a sampling strategy based on Greedy Search Heuristics.

The first component of the algorithm is a multi-scale algorithm for wide-area sampling, which involves unsupervised parametric classification of complex high-spatial complex fields using RBF-NN.

The second component results in further parameter estimate improvements by spot measurements in an EKF-based adaptive sampling scheme. A Joint-EKF structure uses field measurement information to improve robot localization, even in the absence of other location measurements.

The third component is a greedy heuristic strategy with fast convergence leading to considerably less field reconstruction time and number of samples.

Finally, we formulate a multi-robot parallel adaptive sampling algorithm for distribution of computations and communication, and efficient partitioning of sampling area.

Some of the results presented in this Thesis have also been published in several other publications [31-32, 82, 153-155].

1.3 Organization of the dissertation

Chapter 2 summarizes prior research in robotic sensor deployment for sampling missions. Two classes of sampling processes are described: the first one is based on estimating parameters of an assumed processes model; and the second one is non-parametric, in which the field model is either not known or not used. Several steps in parametric density estimation are described, including clustering, parametric approximation and parameter estimation. Several examples from literature of parametric and non-parametric fields and their sampling using robots are discussed. A survey of robot localization schemes is also presented, including: a survey of absolute and relative localization schemes, sensor fusion for mapping and navigation, simultaneous localization and mapping (SLAM), and multi-robot localization.

Chapter 3 presents the mathematical formulation of the proposed Adaptive Sampling algorithm for parameterized fields. It starts with the qualitative and quantitative definition of an Adaptive Sampling problem for a single robotic vehicle

considering the models and uncertainties involved. A Joint-EKF formulation is presented for combining robot location states and field parameters in a single estimator. Several field cases are considered, starting from simple field and vehicle assumptions and working our way to more complex examples. At first, we assumed that a parametric field measurement depends linearly on unknown field parameters, but non-linearly on robot position. Linear least squares is used to find the estimated field parameters in a closed-form. In the second case, a linear Kalman Filter is used to solve the same problem and the results are compared to the first case. In the third case, a robot model was introduced and uncertainties in both the vehicle and location measurements were considered. A Joint Extended Kalman Filter (EKF) was used to estimate both the field parameters and the robot location states. In the fourth case, we assumed that absolute location measurement is not available and we estimated the field parameters and location states by field measurements only. And in the last case, the field was parameterized by Radial Bases Function (RBF) Neural Networks with Gaussian activation. The error covariance of the unknown field parameters was used as an information measure to direct the robots to sample locations minimizing the uncertainty in the knowledge of the field. The performance of different sampling strategies such as Raster scanning, Random Sampling, Adaptive Sampling and Greedy Adaptive Sampling are compared.

Chapter 4 presents a multi-scale, multi-rate adaptive sampling algorithm and its application to mapping of forest fires. Two existing semi-empirical parametric models

were used to describe the spread of forest fires, and the parameterization of the forest fire field was done by interpreting remote-sensing images. We formulate an EKF-based adaptive sampling algorithm for spatio-temporal distributions, and applied it to mapping of forest fires. Finally, we discuss potential field path planning for robots navigating through the estimated fire field as a natural extension of sampling.

Chapter 5 presents detailed simulation results to support our conclusions, using parametric fields of increasing complexity. We present simulation results for linear-in-parameter fields without and with considering uncertainty in localization of the robots. We then present simulations results for single Gaussian fields with and without considering uncertainty in localization of the robots. The final section in Chapter 4 discusses the simulation of complex spatio-temporal forest fire fields, and their parameterization using RBF Neural Networks. Simulation results are included for stationary and time-varying forest fire fields, and potential field based approach for path planning.

Chapter 6 presents experimental results of sampling with robotic hardware. We describe the multi-robot testbed at the ARRI's DIAL (Distributed Intelligence and Autonomy Lab) used to validate the AS algorithms. For linear field distributions, printed color on the lab floor was measured using color sensors mounted on small in-door robots. For time-varying non-linear distributions, a camera-projector system was employed. The ARRI-Bot robots are inexpensive Wireless Sensor Network units built at DIAL, equipped with Javelin-stamp microcontroller, color sensor, cricket transceiver

and dead-reckoning location estimation capabilities. In addition to relative localization using optical encoders, absolute robot localization using the over-head camera and triangulation was implemented. Several simulation cases from Chapter 3 were experimentally validated. Results indicate that the proposed Greedy Adaptive Sampling (GAS) works much better than other alternatives in terms of number of samples and the time it takes to reconstruct the field.

Chapter 7 introduces multi-robot sampling based on existing approaches of centralized, decentralized [57-59] and partially centralized federated filter [56]. We then present the formulation of a distributed federated Kalman filter for multi-robot adaptive sampling. This includes explanation of the scheme for efficient partitioning of the sampling area using fuzzy c-means clustering and Centroidal Voronoi Tesselation (CVT) diagrams. We then discuss the advantages of our approach in terms of reduction in computations and communication overhead, and the proposed decomposition of the filter via three categories of parameters which are unique, common or out-of-the-range of a particular robot's sampling area. Finally, simulation and experimental results of sampling with multiple robots using a centralized filter are presented.

Chapter 8 concludes the dissertation and discusses future work.

CHAPTER 2

BACKGROUND AND RELATED WORK

This chapter discusses background work related to the deployment of mobile robots for sampling. Section 2.1 presents different sampling strategies. Section 2.2 describes the density estimation by sampling and sensor fusion. Sections 2.3 and 2.4 explain existing approaches for sampling using static and mobile sensor nodes, respectively. Parametric and non-parametric solutions to the sampling problem are also discussed. Section 2.5 presents the existing approaches for reducing the localization error in mobile robots while sampling.

2.1 Sampling

The idea of sampling comes from Statistics where only a small set of measurements are available and they are used to estimate the characteristics of a population. Thompson [138] discusses several ways to select few samples from a large dataset. When considering sequential estimation of a field distribution, the simplest method is to randomly select samples for every measurement with each item in the population having equal probability for being chosen irrespective of the field. In other words, each sampling point is selected independently from all other sampling points as shown in figure 2.1 (a). Another approach is *Cluster* or *Stratified Sampling,* in

which the sampling area is classified into homogenous clusters. Such classified stratas can be overlapping or non-overlapping. The formation of strata requires prior information about the field of interest. In practice a combination of these approaches is used, for example a simple *Random Sampling* (figure 2.1 (a)) or *Stratified Random Sampling* (figure 2.1 (b)). Another sampling approach is *Systematic Sampling* (*Uniform Grid-size Sampling*), which involves dividing the sampling area into equal sized grids. Examples of these approaches are *Systematic Raster-scan Sampling* (figure (2.1 (c)) and *Systematic Random Sampling* (figure 2.1 (d)). In *Adaptive Sampling,* sampling locations are selected in real-time, based on the information that is gained by past sampling. In this thesis, an Adaptive Sampling algorithm for complex parametric fields is proposed and its advantages over random or raster non-adaptive sampling are highlighted.

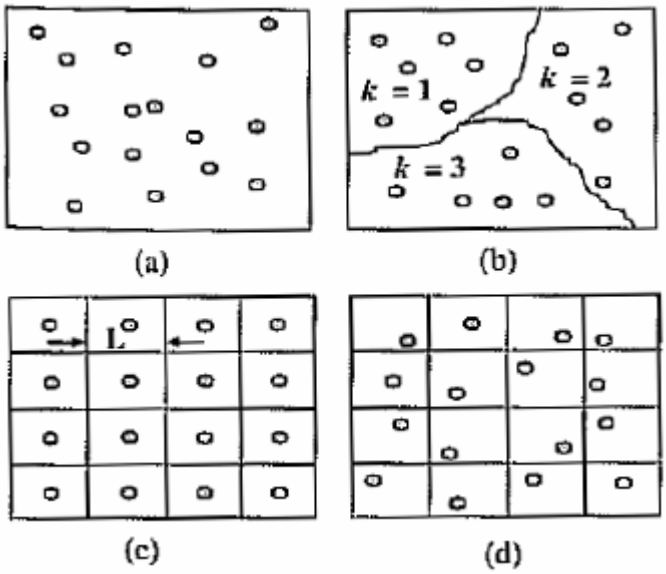


Figure 2.1 Sampling Strategies: (a) simple random sampling, (b) stratified random sampling, (c) systematic raster-scan sampling, (d) systematic random sampling [138]

2.2 Sampling for density estimation

One purpose for sampling a space-time field is density estimation, e.g. the construction of a field estimate of how a certain parameter varies in space and time based on observed data. Data analysis can be performed by assuming either a parametric or a non-parametric distribution. Parametric formulations assume a functional form of the probability density function (PDF), while non-parametric solutions don't assume any particular distribution in advance. In the parametric case, given a density $f(./\theta)$, such as a two-parameter Normal family $N(\mu,\sigma^2)$, where $\theta=(\mu,\sigma^2)$, the emphasis is on obtaining the better estimator $\hat{\theta}$ of θ, and the error criteria can be the $MSE\{\hat{\theta}\}=E\left[\theta-\hat{\theta}\right]^2$. In the non-parametric case, the emphasis is directly on obtaining a good estimate $\hat{f}(.)$ of the entire density function $f(.)$, and the error criteria can be $MSE\{\hat{f}(\theta)\}=E\left[f(\theta)-\hat{f}(\theta)\right]^2$.

There are two types of information for density estimation: *apriori information* and *empirical information*. Apriori information is the knowledge that was already there before a given observation became available. The combination of prior knowledge and empirical knowledge leads to a*posteriori information*. Parametric description of distributions can be either a physical model of the system, or a black or grey box model acquired by clustering the apriori data, and then fitting a linear or non-linear model to it. Non-parametric formulations can also involve clustering. The concepts of clustering, parametric approximation and parameter estimation are introduced in this section in the context of using apriori data for clustering and system identification to acquire a

parametric description and, therefore improve the parameter estimates when empirical knowledge is available.

2.2.1 Clustering

Clustering is the process of organizing a set of data into groups in such a way that observations within a group are more similar to each other than they are to observations belonging to a different cluster. Clustering can be done either by *supervised* or by *unsupervised learning* of the data. Examples of clustering are *hierarchical clustering*, *k-means clustering*, *and mixture of Gaussians clustering*, etc. the details of which can be found in the literature [84]. Two examples of *k-means clustering* and *fuzzy c-means clustering* are relevant to this thesis and are further covered in this section.

2.2.1.1 *K*-means clustering

K-means clustering is one of the simplest unsupervised learning algorithms for clustering discussed in detail in the literature [45]. The objective is to classify a given set of data into *k* clusters where each cluster has a centroid *c*. There are three ways to select centroid locations: randomly, at fixed distances, or at locations where significant data is present. Once the centroid locations are selected, the next step is to assign each of the available data to the nearest centroid. When all the data is assigned, the locations of centroids are recalculated. The procedure is repeated until the centroids no longer move. The objective function which is minimized for recalculating the centroid is:

$$J = \sum_{i=1}^{N} \sum_{j=1}^{K} \left\| x_i - c_j \right\|^2 , \tag{2.1}$$

where N is the number of Gaussian centers, K is the number of clusters and c_j is the centroid of the cluster j.

2.2.1.2 Fuzzy *c*-means clustering

The fuzzy clustering approach was introduced by Bezdek [51]. When compared to *k*-means clustering, fuzzy clustering allows one piece of data to belong to 2 or more clusters. Each Gaussian center belongs to a cluster to some degree which is defined by membership grade by a value u between 0 and 1. The algorithm involves the minimization of the cost function:

$$J_m = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2, \ 1 \le m < \infty . \tag{2.2}$$

where N is the number of Gaussian centers, C is the number of clusters, u_{ij} is the degree of membership of center x_i in cluster j, c_j is the centroid of the cluster j and m is a real number greater than 1. The member function and cluster centroids are iteratively optimized by the following equations:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C}\left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|}\right)^{\frac{2}{m-1}}}, \quad c_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} x_i}{\sum_{i=1}^{N} u_{ij}^{m}}. \tag{2.3}$$

This iteration stops when $\max_{ij}\left\{\left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right|\right\} < \varepsilon$, where ε is the termination criteria, and k is the number of iterations.

2.2.2 Parametric approximation

Parametric approximation is a curve-fitting problem the objective of which is to use the apriori information about a non-parametric field to approximate it with a parametric field. Methods for parametric approximation are well described by Nelles [45] and are some of the ones used in this work are covered below.

2.2.2.1 Linear and polynomial approximations

The simplest of parametric approximation is linear for the fields whose non-linear characteristic is weak. For example:

$$\hat{F} = a_0 + a_1 u_1 + a_2 u_2 + \cdots a_p u_p = \sum_{i=0}^{p} a_i u_i \text{ with } u_0 = 1 \ , \tag{2.4}$$

where a_i are the slope values in the direction of u_i, and a_0 is the ordinate value at $u = 0$. Equation (2.4) can be consolidated in vector form through a parameter vector:

$$\hat{A} = \begin{bmatrix} a_0 & a_1 & \cdots & a_p \end{bmatrix}^T . \tag{2.5}$$

For non-linear fields a polynomial approximation can be considered, which is a straightforward extension of a linear model. The higher the degree of the polynomial, the more flexible the model is. For instance, a 3-dimensional polynomial of degree 2 is given by:

$$\hat{F} = a_0 + a_1 u_1 + a_2 u_2 + a_3 u_3 + a_4 u_1^2 + a_5 u_1 u_2 + a_6 u_1 u_3 + a_7 u_2^2 + a_8 u_2 u_3 + a_9 u_3^2 . \tag{2.6}$$

Equations (2.5) and (2.4) represents static fields for which the values of these parameters stay the same. But time-varying fields can also be modeled through time-dependent parameters.

2.2.2.2 Neural Network approximation

A general formulation of parametric approximation is an artificial neural network. Instead of assuming a linear or polynomial basis, a neural network assumes a general non-linear basis function $\theta_i^{(nl)}$ organized in multiple of layers. The most common neural network architectures are MLP (Multi-Layer Perceptrons), and RBF (Radial Basis Functions) networks. Both MLPs and RBFs are universal approximators, e.g. they can approximate any smooth function [45]. The universal approximation capability is an important property which will be used in this thesis to approximate complex fields. Although MLP has its own advantages in terms of accuracy, its hidden layers are either *log* or *tanh,* which are global activation functions. It may be good to approximate a complex field with parameters of *log* function layer, but it is difficult to distribute computations with global activation functions because the information is not locally contained. In contrast, RBFs activation is local, which is advantageous in sequential robotic sampling, because samples have an effect on neighboring Gaussians only. This is further elaborated in Chapter 7. In short, in RBF networks, the parameters of the hidden layers have a more appropriate sampling interpretation than MLP.

An RBF network consists of three types of parameters:

- Output layer weights, a_i, are linear parameters. They determine the height of the basis functions and the offset value.
- Centers, c_{ij}, are nonlinear parameters of the hidden layer neurons. They determine the positions of the basis functions.

- Standard deviations, σ_i, are also nonlinear parameters of the hidden layer neurons. They determine the width of the basis functions.

Radial basis functions exhibit good approximation properties of their own. Any continuous function can be approximated with RBF neural network according to Universal Approximation Theorem:

Let $G:\Re^{m_0} \to \Re$ be an integrable bounded function such that G is continuous and $\int_{\Re^{m_0}} G(x)dx \neq 0$. Let $\Im_G$ denote the family of RBF networks consisting of functions: $F:\Re^{m_0} \to \Re$ represented by:

$$F(x) = \sum_{i=1}^{N} a_i G\left(\frac{x - c_i}{\sigma_i}\right), \tag{2.7}$$

where $\sigma_i > 0$, $a_i \in \Re$ and $c_i \in \Re^{m_0}$ for $i = 1, 2, \cdots, N$. The universal approximation theorem for RBF networks can then be stated as:

For any continuous input-output mapping function $f(x)$ there is an RBF network with a set of centers c_i and width σ_i where $i = 1, 2, \cdots, N$ such that the input-output mapping function $F(x)$ realized by the RBF network is close to $f(x)$ in the L_p-norm, $p \in [1, \infty]$.[52]

For training a RBF neural network, different strategies exist. In the simplest case, the hidden layer of the network is not optimized, which means that there is no optimal learning strategy for c_i, σ_i. But there are several strategies for placing the center locations before learning of other parameters begins. One such strategy is the

random center placement, which is considered a reasonable initialization method for some of the advanced learning schemes [45]. A straightforward improvement of random selection of centers is the application of clustering techniques discussed in previous section. The greatest advantage of using this approach is that the centers are selected according to the training data distribution in input space. Therefore, more RBFs are placed in regions where data is dense, and few RBFs are placed in regions where data is sparse.

The universal approximation theorem is valid even for a constant σ for all neurons, but there are heuristic methods to calculate standard deviations for training the RBF network. The most common is the *k-nearest neighbor* rule, which assigns each RBF a standard deviation based on average distance between its center and the center of the k nearest neighbor RBFs. Finally, the weights of the output layer can be computed using least squares. The clustering and LS algorithms are both very fast, even for relatively complex fields.

2.2.3 Parameter estimation

Parameter estimation problems involve finding the best value for parameters of a process based on noisy measurements. Least squares estimation is the most common criteria used for batch as well as sequential estimation problems. For recursive estimation problems, an initial estimate and uncertainty (error covariance) is assumed. As noisy measurements become available sequentially, the parameter estimates, as well as the estimation error covariance are updated. The Kalman filter is a fundamental estimation tool, which uses the dynamical description of the system along with

recursive least squares estimation for processing measurements. Robot location states and field parameters estimation using Kalman filters is covered in great detail in the next chapter.

2.3 Sampling for static Wireless Sensor Network (WSN)

Wireless networks of spatially distributed sensors provide an attractive approach to spatially monitor the environment. The obvious advantage of using static nodes compared to mobile robots is that there are no localization errors due to navigation. However, using higher densities of sensors provides more measurements, higher resolution and better accuracy, at the expense of more communication and processing. Several deployment schemes for static nodes have been recently proposed to reduce communication and energy consumption. For the simplest scenario of field estimation, sensors are deployed at the centers of uniformly-space square grids [77]. Based on the initial information about the field to be mapped, the sensors could be deployed more densely in the high-variance areas and far apart in the areas with small variation [15]. Usually the objective is to minimize the energy consumption and communications by activating only a fraction of sensors for measurements and communication while maintaining high-accuracy. However, this kind of approach cannot be used to monitor wide areas. Also, a sparse density of placement of static nodes reduces the accuracy of the estimate.

Other related problems for static sensor network are *coverage* and *connectivity* [16]. The coverage problem is defined as the maximization of the total area covered by the sensors, while connectivity is the requirement that every sensor should be able to

communicate with every other sensor. There are many applications of coverage and connectivity for applications such as target tracking, search and rescue, etc.

Static nodes are also widely used in applications involving navigation and controlling of mobile robots, environmental monitoring, disaster and rescue work and healthcare, etc. When used for tracking mobile targets, static sensors act as anchor nodes with known position and help the mobile robots localize.

2.4 Robotic sensor deployment for sampling

Mobile robots are being increasingly used on land, underwater, and in air, as sensor carrying agents to perform sampling missions, such as searching for harmful biological and chemical agents, search and rescue in disaster areas, and environmental mapping and monitoring. A multitude of research groups have published results on sampling using mobile robots for chemical plume source localization, mine-detection, ocean sampling, forest mapping, etc [4, 8, 15, 17, 38, 41, 73, 74, 76, 97-99].

One of the problem most commonly discussed in mobile WSNs are team objectives for exploration and mapping in an unknown environment. Sensor deployment for sampling can be considered a variant of the exploration and mapping problem. The aim in mapping and exploration is to build a global map of the environment by sequentially visiting each location with one or more robots, whereas in information-aware sampling the aim is choosing the best locations that minimize the uncertainty of the field estimate. In exploration for building a global map it is impractical to have the robots take measurements at all locations.

When using mobile robots for sampling missions, accurate localization is needed, because the uncertainty in localization affects the estimate of the spatio-temporal field. Even though mobility introduces additional degree of complexity for navigation and localization, it allows the repositioning of on-board sensors, thus expanding the coverage and operational lifetime of the sensor network. For instance, Howard *et .al.* [16] discusses the mobile robotic sensor deployment using potential fields for maximization of area covered by the network. In addition to coverage, the problem of distributed field variable estimation by inverse modeling of the measurements has been addressed for oceanography and meteorology [149]. Popa *et. al.* [33] proposes using a potential field framework to control the behavior of the mobile sensor nodes by combining classical robotic team concepts (obstacle avoidance, goal attainment flight formation, environment mapping and coverage) with traditional sensor network concepts (node energy minimization, optimal data rate and congestion control, routing in ad-hoc networks).

In the end, the performance of any deployment scheme for sampling will also depend on the type of field distribution, and its representation.

2.4.1 Parametric field representation

Parametric representation is a model-based method in which parameters of a process model is estimated using techniques such as the Kalman Filter, Least Squares Estimation or other optimization schemes.

Christopoulos and Roumeliotis [4] present an approach for estimating the parameters of diffusion equation that describes the propagation of an instantaneously

released gas. This is accomplished by first building a model considering factors such as the diffusion phenomena of the chemical, wind direction and speed, turbulence, etc. The selection of the set of locations where chemical concentration measurements should be recorded is performed in real-time with the objective of maximizing the accuracy of the parameter estimates and reducing the time to convergence of this estimation problem. The selection of sequence of locations where the measurements are recorded is based on the minimization of the uncertainty of the estimated parameter which is represented by the trace of the error covariance of parameters estimates.

Cannell and Stilwell [3] present two approaches for adaptive sampling of underwater processes using AUVs. The first one assumes a parametric model, while the second one uses an information theoretic approach. For the parametric approach, two methods for estimating the unknown parameters of the process model are considered. The first method is related to mapping algorithms where the unknown parameters in an assumed measurement model are states of a Kalman Filter. In the second method, non-linear least squares are used to estimate the parameters.

2.4.2 Non-parametric field representation

A number of strategies for non-parametric adaptive sampling can also be found in the literature. A solution for non-parametric ocean sampling is proposed in [15] based on classification of the sampling area. The area is divided into non-uniform size grids and the size of the grid is selected based on the local variance of the distribution. Hence the areas with high variance have small size grids and the areas with small variance

have large size grids, which is important for thoroughly sampling the areas with more variation and just taking a few samples in more uniform areas.

In some cases, information theoretic approaches are utilized, which process data via sequential classification and tracking phases. During the classification phase, measurements are collected in the assumed ambient flow, and an empirical distribution is formed. This distribution represents the measurements taken outside the process. As new measurements are taken, updated empirical distributions are computed and compared to the assumed ambient distribution using the *Minimum Description Length (MDL) test*. This test utilizes the *Kullback-Leibler Divergence,* which does not require any explicit defined process model. It characterizes the boundary of a closed *Divergence*, also known as *Relative Entropy*, to determine if two distributions are different in a statistically significant manner. This result is then used as a test for identifying data that is statistically different from the ambient distribution. This information theoretic approach has been applied to ocean sampling and prediction [34].

Another non-parametric approach is *cognitive* or *rule-based* in order to adaptively select sample locations in real-time based on current and past samples to ensure that a field source is localized. This scheme does not involve formation of any distribution, but it is used to find distribution maxima. This non-parametric approach is used very widely in chemical plume tracing on land and in water [8], odor sensing, mine detection [76], etc.

Chemical source localization is usually a gradient climbing problem with disturbances caused by wind, gas diffusion and turbulence from the obstacles and the

environment. A lot of research has been done in the area of odor source localization, land-mine detection, etc. using both parametric and non-parametric approaches. Non-parametric approaches involve taking appropriate samples in a recursive fashion in order to find the source. In practice, such approaches generally involve three phases: detecting the plume, tracking the chemical towards its source and finally locating the source. Some source localization algorithms are biologically inspired because insects and most other animals are very apt at sensing odors in order to find food, locating mates or communicate. The applications for source localization using robots are in locating harmful biological and chemical agents, avalanche and earthquake victims, landmine detection, etc. It would be difficult for humans to find chemical sources because of presence of poisonous gases, nuclear radiation or any other hazardous conditions.

Chemotaxis and *Anemotaxis* are the most common techniques used for estimating chemical distributions. Chemotaxis relies on the local gradient of the chemical agent concentration while Anemotaxis based approaches require that the agent move in the upwind direction. The robots used for source localization are equipped with anemometric sensors and gas sensors, and in some cases with vision. Several algorithms such as step-by-step, zigzag, active sampling can be found in the literature [17, 38, 74, 97-99].

Another similar application of non-parametric sampling is mine detection using demining robots. The sensors used are metal detectors, IR detectors and chemical sensors. Literature on mine detection using robots can be found in [76].

2.5 Robot localization

Knowing the location of a robotic vehicle is a fundamental problem in order to navigate autonomously and perform useful tasks. Irrespective of whether the vehicle is aerial, under-water or a ground vehicle, it is crucial for the robots to get a good estimate of their location. Measurements of robot localization can be categorized into relative and absolute location measurements.

2.5.1 Relative measurement

Relative localization relies on the on-board inertial sensors of the robot, such as encoders, gyroscopes or accelerometers. This technique is also called "dead-reckoning", and uses the kinematic model and odometric data to compute position and orientation for the robot. The advantage of odometry is that it is self-contained, and hence always available. The disadvantage is that there is unbounded accumulation of errors [80, 109-114].

There are two types of errors in relative location measurement: (i) systematic errors, and (ii) non-systematic errors. Systematic errors are due to errors in the robot model parameters. For wheeled robots, the sources of systematic errors could be unequal wheel diameters, uncertainty about effective wheel-base, limited encoder resolution, misalignment of wheels, limited encoder sampling rate, etc. Non-systematic errors are due to the surroundings of the robots such as uneven floors, unexpected object, fast-turning, over-acceleration, etc. As this problem is fundamental in robotics, many researchers have tried to improve odometric errors. Borenstein and Feng [80, 112] present a calibration technique called UMB-Mark test to calibrate the systematic errors

of a mobile robot with differential drive. Martinelli [79, 80] proposes a method to estimate both systematic and non-systematic errors for synchronous drives.

2.5.2 Absolute measurement

Relative localization is prone to errors in the calculation of robot's location. To overcome this problem, absolute localization can be performed using GPS, landmarks or maps. The landmark could be an active beacon or passive landmarks (natural or artificial). The advantage of absolute measurement is that the location is independent of previous location estimates and hence there is no accumulation of errors with time or distance traveled. One of the drawbacks of these measurements is that they are not always available. For example, for AUVs performing ocean sampling, GPS information is only available when the vehicles rise to the surface.

2.5.2.1 Artificial Landmarks (Beacon based localization)

Landmarks are features in the environment that a robot can detect. These landmarks could be active or passive. Active landmarks are the ones that actively transmit signals (radio, sonar, etc) that can be received by the receiver on the robot. Robots then can triangulate or trilaterate to estimate their location, if distance is available from several active beacons. Localization using GPS uses the same idea by using satellite landmarks. Active landmarks are usually artificially placed at known location. These kinds of localization algorithms where some of the nodes knows their location is called 'anchor-based localization' and the nodes with known location are called 'anchor nodes' [143]. The 'incremental algorithms' usually start with a set of 3 or 4 nodes with assigned coordinates. Then they repeatedly add appropriate nodes to this

set by calculating the node's coordinates using the measured distances to previous nodes with already computed coordinates. In 'concurrent algorithms' all the nodes calculate and refine their coordinates in parallel.

The main type of signals used by beacon based localization systems are infrared laser, ultrasound and millimeter wave radar.

2.5.2.2 Natural Landmarks (apriori map based localization)

Landmarks could also be naturally existing objects in a structured environment or it could be artificially placed objects with particular shapes. Many artificial landmark positioning systems are based on computer vision. In addition bar-coded reflectors are also used laser scanners or sonar sensor. The important issues in passive landmark based localization are:

- Landmark based localization requires good initial estimates of robot location. In the absence of such estimate, a time-consuming search process to form an association between different objects [68-69].
- Localization accuracy depends on the distance and angle between the robot and the landmark. Hence line-of-sight is an important issue.
- A database of the landmarks their location needs to be maintained.
- Ambient conditions also affect the accuracy.

Tanaka *et. al.* [66] discusses the robot's localization with incomplete maps in non-stationary environments which is done by detecting changes in the environment.

2.5.2.3 Map building

In map building, a robot that is completely localized uses its sensors to build a map of the environment. This could be map matching problem, involving establishing a correspondence between a current local map and a stored global map. Mapping is achieved by capturing images and then performing “feature extraction”.

2.5.3 Sensor fusion

There is no single sensor measurement that can adequately capture all relevant features of a real environment. To overcome this problem most localization algorithms combine data from different sensors. This multi-sensor fusion is based on probabilistic approaches such as Bayesian estimation, or the use of the Kalman Filter (KF). KF provides an optimal unbiased estimate, and its predictor-corrector structure works very well for combining reading from different sensors at different rates. Fusion algorithms combine relative measurement from encoders and absolute measurement from beacons which are usually available at different rates [102, 106].

This kind of location estimation approaches relies on a model of the vehicle and the sensor. The estimate is updated each time a new measurement is taken, and the vehicle location is corrected using a prediction of its position based on control inputs.

Roumeliotis and Bekey [63] and Roumeliotis and Sukhatme [67] present a general EKF formulation for the fusion of relative and absolute position measurement for improving the localization. In [62, 64], the approach is validated for the fusion of dead-reckoning and ultrasonic measurement. Mourikis and Roumeliotis [66] propose a scheme for using the relative distance measurement and correlating the estimates to

improve the overall state estimate. They fuse dead-reckoning location estimates with relative distances between the two robots when the location of both needs to be estimated. Multi-robot localization using relative observations is discussed in [37, 66].

2.5.4 Simultaneous Localization and Mapping (SLAM)

Simultaneous localization and mapping is currently one of the most important goals of autonomous robots navigation. Solving this problem will allow robotic sensor nodes to be deployed easily with very little initial preparation. The problem of localizing robotic vehicles in a previously unexplored environment is addressed by using the vehicle location to build the map of the environment and simultaneously using the map to localize the vehicle.

A comprehensive survey on the solution of SLAM problems is given in [68-69]. Early work in SLAM assumed that the world could be modeled as a set of simple discrete landmarks. However, in more complex and unstructured environments this assumption does not hold. More realistic solutions to SLAM problems include:

- *Delayed mapping*- involving accumulating information and permitting delayed decision making. Deferred data facilitate reliable data association.
- *Partial observability*- including *bearing-only SLAM* e.g. with a camera, or *range-only SLAM* e.g. with sonar sensors.
- *Unstructured or dynamic environment*- involves detecting and removing the landmarks that are moving.
- *3D SLAM*- Increased complexity compared to 2D case.

- *Multi-Robot SLAM or Cooperative SLAM*- Different robots act as landmarks for each other. One scheme to avoid complexity is that only one robot moves at a time and other remains stationary and act as landmarks for other robots. After some time they can swap their roles.

2.5.5 Multi-robot localization

Collaborative multi-robot localization schemes rely on using robots as landmarks for one another. The key idea is to combine localization through internal sensors (such as encoders) with measurements from the neighboring robots, in order to improve localization. The earliest approach presented for multi-robot localization assumed that only one robot move at one time while the rest of the team forms an equilateral triangle of localized beacons [81]. After some time the robots swap roles. For this kind of approach, only one robot can move at a time, and robots must maintain contact either through vision or sonar sensing. Other approaches are presented in [120-128], in which sensor data from multiple robots is combined in a single Kalman Filter to estimate the position of each robot in the team. Several research groups have also formulated how a centralized KF can be divided into N separate KFs, one for each robot, in order to to allow distributed processing:

- *None of the Robot has absolute positioning information.* By measuring the relative locations, multiple robots can improve their position tracking accuracy, but they are not able to bound the overall uncertainty.
    - Dead-Reckoning: No exchange of information causes the error to grow drastically.

- Continuous relative position measurement: the rate at which positioning uncertainty grows is considerably lower.
- Intermittent relative position measurement: positioning errors are reduced even if the robots exchange position information intermittently.

- *One of the robots has absolute positioning information*. The system is observable if communication takes place.
  - Continuous communication between the robots: positioning error is bounded for all robots.
  - No communication between the robots: rhe robot which does not have absolute position information will just rely on DR and the location estimates will eventually diverge.
- *One of the robots remains stationary.* One of the robots remains stationary, while the others move in the same area and measures their relative locations with respect to the standing on. Even though no absolute positioning information is available to any of the robots in the group the system is observable.

In this Chapter, we presented prior work in sensor fusion and localization for density estimation of space-time fields using mobile robots. The approaches are based on physical models, parametric and non-parametric feature-based inference, and cognitive-based models. Different methods for localization of mobile nodes and related errors were also summarized, because of their relevance to robotic sampling problems.

CHAPTER 3

ADAPTIVE SAMPLING OF PARAMETRIC FIELDS

This chapter focuses on the mathematical formulation of AS problem for parameterized fields, including models, uncertainties, and sampling criteria. Section 3.1 gives both a qualitative and quantitative definition for the Adaptive Sampling problem. Section 3.2 presents the EKF formulation of the AS problem with a single mobile sensor node. Section 3.3 discusses the parameterization of non- parametric fields using RBF Neural Networks. Finally section 3.4 discusses sampling strategies such as Raster Scanning, Random Sampling, Adaptive Sampling and Greedy AS.

3.1 Adaptive Sampling problem

Mobile robots are being used on land, underwater, and in air, as sensor carrying agents to perform sampling missions, such as searching for harmful biological and chemical agents, search and rescue in disaster areas, and environmental mapping and monitoring. Sampling can be simply defined as *a set of elements drawn from and analyzed to estimate the characteristics of a population.* In adaptive sampling*, strategies continuously adapt in response to real-time environmental measurements without human intervention.* Once such scenario involves the

deployment of multiple underwater vehicles for the environmental monitoring of large bodies of water, such as oceans, harbors, lakes, rivers and estuaries [3, 8, 34]. Predictive models and maps can be created by repeated measurements of physical characteristics, such as water temperature, dissolved oxygen, current strength and direction, or bathymetry. However, because the sampling volume could be large, only limited measurements are available. Intuitively, a deliberate sampling strategy based on models will be more efficient than just a random sampling strategy.

Popa and Sanderson [34] described a combined multi-agent AS problem coupling uncertainty in localization as well as in the sensor measurements to achieve effective adaptive sampling using a solar AUV (Autonomous Underwater Vehicle), considering very simple, linear field models. Localization uncertainties are relevant to the sampling problem since position estimates for underwater robots are often inaccurate due to navigational errors from dead-reckoning. In this thesis we expand and extend this parametric approach to include much more complex fields and heuristics with much better sampling performance. Unlike non-parametric sampling methods, such as [15], our approach requires a parametric field description of the sampled field.

Qualitatively, a multi-agent AS problem for spatio-temporal fields can be posed as follows:

We wish to describe an unknown nonlinear spatio-temporal field variable via a parametric approximation $Z = Z(A, X, t)$ *depending on an unknown parameter vector A, position vector X, and time t. The field is recovered by using N robotic vehicles, sampling the field with localization and sensing uncertainty. We wish to:*

- *Decide what sampling locations* $X_k^i, 1 \le i \le N, 1 \le k \le n$ *should be chosen such that the uncertainty in estimating the unknown parameter vector is minimized, and,*
- *Additional secondary objectives such as the number of samples, sampling duration, energy, etc., are minimized.*
- *Distribute the computational and communication load, and the sampling space among N robots in an efficient manner.*

3.2 Basic EKF formulation

To pose the AS problem in a quantitative way, an information measure and uncertainty model needs to be selected. We use the Extended Kalman Filter where the state transition and measurement equations during sampling can be described through non-linear mappings (functions *h*, *m*, *f* & *g*) as follows:

Robotic sensor nodes state dynamics:

$$X_{k+1}^i = X_{k+1}^i + h(X_k^i, U_{1k}^i) + w_k^i, \tag{3.1}$$

where X_k^i, U_k^i are the i^{th} vehicle state and control input, where the state has noise covariance matrix $E[w_k^i (w_k^i)^T] = Q_{1k}^i$.

Field parameter dynamics:

$$A_{k+1} = A_k + m(A_k, U_{2k}) + \alpha_k, \tag{3.2}$$

where A_k is a vector of unknown coefficients describing the field with noise covariance matrix $E[\alpha_k \alpha_k^T] = Q_{2k}$, and U_{2k} is the uncontrollable (but measurable) "field evolution

vector". This parameter is a slow-varying correction factor in the field parameters, assuming that infrequent, low resolution measurements of the entire field are available.

Robot position output measurement:

$$Y_k^i = f(X_k^i) + \xi_k^i, \tag{3.3}$$

where the output noise covariance is $E[\xi_k^i(\xi_k^i)^T] = R_{1k}^i$.

Distributed field variable measurement model:

$$Z_k^i = g(X_k^i, A_k) + \nu_k^i, \tag{3.4}$$

where Z_k^i is the field variable with measurement noise covariance $E[\nu_k^i(\nu_k^i)^T] = R_{2k}^i$.

Robot input measurement:

$$\tilde{U}_{1k}^i = U_{1k}^i + \beta_k^i, \tag{3.5}$$

where $\tilde{U}_{1k}^i$ is the measured control input with measurement noise covariance $E[\beta_k^i(\beta_k^i)^T] = R_{3k}^i$. U_{1k}^i is the output caused by the control input $\tilde{U}_{1k}^i$.

Field evolution factor measurement:

$$\tilde{U}_{2k} = U_{2k} + \eta_k, \tag{3.6}$$

where $\tilde{U}_{2k}$ is the measured field parameter evolution using high-spatial and low-rate measurement with measurement noise covariance $E[\eta_k \eta_k^T] = R_{4k}$. U_{2k} is the actual change in the parameter value since the last estimate.

For the single mobile agent *i* sampling the field, the EKF overall state vector is (X_k^i, A_k), and the system and measurement models are thus summarized as:

$$
\begin{aligned}
&\begin{pmatrix} X^i_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X^i_k \\ A_k \end{pmatrix} + \begin{pmatrix} h(X^i_k, U^i_{1k}) \\ m(A_k, U_{2k}) \end{pmatrix} + \begin{pmatrix} w^i_k \\ \alpha_k \end{pmatrix} \\
&Y^i_k = f(X^i_k) + \xi^i_k \\
&Z^i_k = g(X^i_k, A_k) + \nu^i_k \\
&\tilde{U}^i_{1k} = U^i_{1k} + \beta^i_k \\
&\tilde{U}_{2k} = U_{2k} + \eta_k
\end{aligned}
\tag{3.7}
$$

Using the standard nonlinear, discrete EKF formulation [23], the time and measurement update equations are given by:

$$
\begin{aligned}
&P^-_{k+1} = P_k + Q_k, \begin{pmatrix} \hat{X}^{i-}_{k+1} \\ \hat{A}^-_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}^i_k \\ \hat{A}_k \end{pmatrix} + \begin{pmatrix} h(X^i_k, \tilde{U}^i_{1k}) \\ m(A_k, \tilde{U}_{2k}) \end{pmatrix} \\
&G^i_k = \frac{\partial g_k}{\partial(\hat{X}^{i-}_k, \hat{A}^-_k)},\ F^i_k = \frac{\partial f_k}{\partial \hat{X}^{i-}_k},\ H^i_k = \begin{bmatrix} F^i_k \\ G^i_k \end{bmatrix} \\
&P^i_{k+1} = \left(\left(P^{i-}_{k+1} \right)^{-1} + H^{iT}_{k+1} (R^i_k)^{-1} H^i_{k+1} \right)^{-1} \\
&\begin{pmatrix} \hat{X}^i_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}^{i-}_{k+1} \\ \hat{A}^-_{k+1} \end{pmatrix} + P^i_{k+1} H^{iT}_{k+1} (R^i_{k+1})^{-1} \left[\begin{pmatrix} Y^i_{k+1} \\ Z^i_{k+1} \end{pmatrix} - \begin{pmatrix} f_{k+1}(\hat{X}^{i-}_{k+1}) \\ g_{k+1}(\hat{X}^{i-}_{k+1}, \hat{A}^-_{k+1}) \end{pmatrix} \right]
\end{aligned}
\tag{3.8}
$$

where the state is the combined position of all the robots and the field parameter vector $\begin{bmatrix} X & A \end{bmatrix}^T$, P_k is the state covariance, and the process and measurement error covariances are given by:

$$
Q^i_k = \begin{bmatrix} Q^i_{1k} & 0 \\ 0 & Q_{2k} \end{bmatrix},\ R^i_k = \begin{bmatrix} R^i_{1k} & 0 & 0 & 0 \\ 0 & R^i_{2k} & 0 & 0 \\ 0 & 0 & R^i_{3k} & 0 \\ 0 & 0 & 0 & R_{4k} \end{bmatrix}
\tag{3.9}
$$

By using a norm of the state covariance as information measure, the adaptive sampling algorithm moves the vehicle i from X^i_k to X^i_{k+1} such that the following p-norm of the covariance matrix is minimized over the search space Θ:

$$m(X_k) = \left\| P_k^i \right\|_p, m(X_{k+1}^i) \le m(X_k^i),\ \forall X^i \in \Theta\,. \tag{3.10}$$

The above formulation applies to both single as well as multiple robotic sensor agents on a spatio-temporal field with location and field parameter measurements available at the same rate. In past work, several cases were considered by Sreenath *et. al.* [140-141] to estimate the parameters of stationary field using least squares and the Kalman Filter with single robot, as detailed below:

3.2.1 Least-squares estimation for linear-in-parameters field

If the dynamics of the robot is ignored, and if we assume that the parametric field measurement Z depends linearly on the unknown parameter vector A, but non-linearly on position vector X, then a closed-form solution on the unknown parameters can be obtained, subject to the condition that there is no process and position measurement noise. After n scalar measurements $Z_1,...,Z_n$ are taken at location vectors $X_1,...,X_n$, the field equations can be used to find the least squares solution:

$$\begin{aligned} Z_1 &= a_0 + a_1 g_1(X_1) + ... + a_m g_m(X_1) \\ Z_2 &= a_0 + a_1 g_1(X_2) + ... + a_m g_m(X_2) \\ &\vdots \\ Z_n &= a_0 + a_1 g_1(X_n) + ... + a_m g_m(X_n) \end{aligned}, \tag{3.11}$$

where $A = \begin{bmatrix} a_0 & a_1 & \cdots & a_m \end{bmatrix}^T$ is the parameter vector, and $g_1, g_2, \cdots, g_m$ are nonlinear basis functions of the field. Equation (3.11) can be written in matrix format as:

$$\begin{bmatrix} Z_1 \\ \vdots \\ Z_n \end{bmatrix} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ g_1(X_1) & g_2(X_1) & \cdots & g_m(X_1) \\ \vdots & \vdots & \ddots & \vdots \\ g_1(X_n) & g_1(X_n) & \cdots & g_m(X_n) \end{bmatrix} \begin{bmatrix} a_1 \\ \vdots \\ a_m \end{bmatrix} = M_n A_n\,, \tag{3.12}$$

where M_n is a $n \times m$ matrix. The least-square solution of equation (3.12) is simply:

$$\hat{A}_n = \begin{pmatrix} 1 & g_i(X_j) \end{pmatrix}^{+}_{i \le m, j \le n} \begin{pmatrix} Z_1 \\ \vdots \\ Z_n \end{pmatrix} = M_n^{+} \begin{pmatrix} Z_1 \\ \vdots \\ Z_n \end{pmatrix}, \tag{3.13}$$

where M_n^{+} is the Moore-Penrose inverse of M_n. Moreover, the pseudo-inverse has a closed-form solution given by:

$$M_n^{+} = \left(M_n^T M_n\right)^{-1} M_n^T, \tag{3.14}$$

leading to a parameter estimate closed-form:

$$\hat{A}_n = \left(\sum_{j=1}^{n} \begin{pmatrix} 1 & \cdots & g_i(X_j) & \cdots & g_m(X_j) \end{pmatrix} \begin{pmatrix} 1 \\ \vdots \\ g_i(X_j) \\ \vdots \\ g_m(X_j) \end{pmatrix} \right)^{-1} \sum_{j=1}^{n} Z_j \begin{pmatrix} 1 \\ \vdots \\ g_i(X_j) \\ \vdots \\ g_m(X_j) \end{pmatrix}. \tag{3.15}$$

Assuming that the robot locations are known without uncertainty, the covariance matrix of $\hat{A}_n$ can be related directly to the constant measurement uncertainty as:

$$\mathrm{var}\left(\hat{A}_n\right) = \mathrm{var}(Z_i)\left(M_n^T M_n\right)^{-1} \quad . \tag{3.16}$$

An Adaptive Sampling algorithm should then move the robot from X_n to X_{n+1} such that the *p*-norm of the covariance matrix is minimized over the search space Θ:

$$m(X) = \left\| \begin{pmatrix} & 1 & \\ & \vdots & \\ M_n^T & g_i(X) & \\ & \vdots & \\ & g_m(X) & \end{pmatrix} \begin{pmatrix} & & \\ & M_n & \\ & & \end{pmatrix} \begin{pmatrix} 1 & \cdots & g_i(X) & \cdots & g_m(X) \end{pmatrix} \right\|_p \quad \text{, or} \tag{3.17}$$

$$m(X_n) = \left\| P_n \right\|_p, m(X_{n+1}) \le m(X_n),\ \forall X \in \Theta. \tag{3.18}$$

3.2.2 Kalman filter estimation for linear-in-parameters field with no uncertainty in localization

Like in the previous case, if the parametric form of a measurement field is known, as might be the case, for example, with a bottom profile or systematic variations in temperature or salinity, the field estimation can be integrated with localization in order to improve the estimation process. The assumption that the field distribution is linear in its parameters allows computing a closed-form solution for the information measure used by the sampling algorithm, as we saw in the previous section. The same result can be recovered using the Kalman Filter. After *n* measurements taken at location X_i, the field model depends linearly on *m*-dimensional parameter coefficients $A = (a_j)_{1 \le j \le m}$ via position-dependent functions as in equations (3.11) and (3.12).

If the robot dynamics and localization uncertainty are ignored, the state is simply the parameter vector, and KF equations can be written simply as:

$$\begin{aligned} A_{k+1} &= A_k \\ Z_k &= G_k A_k + \nu_k \end{aligned} \tag{3.19}$$

$$G_k = \begin{pmatrix} 1 & \dots & g_i(X_k) & \dots & g_m(X_k) \end{pmatrix}$$
$$E[\nu_k \nu_k^T] = R$$

The error covariance matrix from the KF is used to decide where to sample next. Because of the simple state update equation, the KF equation can be reduced to:

$$
\begin{aligned}
&A_o \sim (\bar{A}_o, P_{A_o}),\ P_o = P_{A_0} \\
&P_{k+1}^{-1} = P_k^{-1} + G_{k+1}^T R^{-1} G_{k+1} \\
&\hat{A}_{k+1} = \hat{A}_k + P_{k+1} G_{k+1}^T R^{-1} (Z_{k+1} - G_{k+1} \hat{A}_{k+1})
\end{aligned}
\quad . \tag{3.20}
$$

The error covariance is similar to the least squares solution in (3.14) and (3.15), and can be directly calculated by the following formula:

$$
P_k = (P_0^{-1} + \sum_{j=1}^{k} G_j^T R^{-1} G_j)^{-1} \tag{3.21}
$$

3.2.3 Simple Kalman Filter estimation for linear field with uncertainty in localization

In this case, in addition to the parameter state equation update $A_{k+1} = A_k$, the vehicle dynamics must also taken into account. In the simple case when the vehicle motion can be represented by a single particle equation with uncertainty: $X_{k+1} = X_k + U_k + w_k$, and assuming that we use dead-reckoning for localization, and that the function $g(X_k)$ is linear, the KF equations become:

$$
\begin{aligned}
&\begin{pmatrix} X_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \begin{pmatrix} I_3 \\ 0 \end{pmatrix} U_k + \begin{pmatrix} w_k \\ 0 \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + BU_k + \vartheta_k \\
&\begin{pmatrix} Y_k \\ Z_k \end{pmatrix} = \begin{pmatrix} X_k \\ \begin{pmatrix} 1 & g(X_k)^T \end{pmatrix} A_k \end{pmatrix} + \begin{pmatrix} \xi_k \\ \nu_k \end{pmatrix} = \begin{pmatrix} I_3 & 0 \\ 0 & \begin{pmatrix} 1 & g(X_k)^T \end{pmatrix} \end{pmatrix} \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \lambda_k \quad , \\
&X_{k+1}^{DR} = X_k^{DR} + h(X_k^{DR}, u_k^-)
\end{aligned}
\tag{3.22}
$$

where X_k^{DR} is the dead-reckoning estimate for the robot position computed after a command vector u_k^- is applied to the robot to cause a predicted position change equivalent to u_k, and

$$E[\vartheta_k \vartheta_k^T] = Q = \begin{pmatrix} Q_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad E[\lambda_k \lambda_k^T] = R = \begin{pmatrix} R_1 & 0 \\ 0 & R_2 \end{pmatrix} \tag{3.23}$$

are the white noise covariances of the state and output.

The linear Kalman Filter equations become:

$$\begin{aligned}
&P_{k+1}^- = P_k + Q, \begin{pmatrix} X_{k+1}^- \\ A_{k+1}^- \end{pmatrix} = \begin{pmatrix} \hat{X}_k \\ \hat{A}_k \end{pmatrix} + BU_k \\
&P_{k+1} = ((P_{k+1}^-)^{-1} + G_{k+1}^T R^{-1} G_{k+1})^{-1} \\
&G_k = \begin{pmatrix} I_3 & 0 \\ 0 & \begin{pmatrix} 1 & g(X_k^T) \end{pmatrix} \end{pmatrix}, \begin{pmatrix} X_{k+1}^- \\ A_{k+1}^- \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + BU_k \\
&\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} X_{k+1}^- \\ A_{k+1}^- \end{pmatrix} + P_{k+1} G_{k+1}^T R^{-1} \left[\begin{pmatrix} Y_{k+1} \\ Z_{k+1}^- \end{pmatrix} - G_{k+1} \begin{pmatrix} X_{k+1}^- \\ A_{k+1}^- \end{pmatrix} \right]
\end{aligned} \tag{3.24}$$

If on the other hand the robot dynamics or the field is nonlinear in parameters or in the robot state, equation (3.24) will become a Non-Linear Kalman Filter (EKF).

3.2.4 Kalman filter estimation for linear-in-parameters field with location measurement unavailable

In previous cases it is assumed that both the sensor node position measurement (Y_k) and field variable measurement (Z_k) to estimate parameter states (A_k) and position states (X_k). In this case we assume that the node position output is not present and we rely on field variable measurements alone, but we allow for the field to be linear in parameters but nonlinear in the location state. Localization uncertainties are especially relevant in GPS-denied environments, but they can also be present in situations where local visual information is unavailable (for instance due to thick smoke), and GPS data rates are slow.

Here model parameter estimation for the field variable is integrated with estimation of the uncertainty in the mobile robot localization and the overall covariance of the estimate is used for sampling. This way localization uncertainty can be reduced by building accurate models of distributed fields and vice versa. For instance, if a robot is sampling an unknown field, but its location is accurately known, a distributed parameter field model can be constructed by taking repeated field samples. Later on, this field model can be used to reduce the localization error.

In the case of an m-dimensional linear-in-parameters field distribution,

$$g(X_k, A_k) = a_{ko} + \sum_{i=1}^{m} g_i(X_k) a_{ki} = \begin{pmatrix} 1 & g_1(X_k)^T \end{pmatrix} A_k, \tag{3.25}$$

where A_k is the parameter vector and is a nonlinear basis g_1 for the field, the non-linear EKF equations can be written as:

$$\begin{aligned} &\begin{pmatrix} X_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \begin{pmatrix} I_3 \\ 0 \end{pmatrix} U_k + \begin{pmatrix} \omega_k \\ 0 \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + B_k U_k + \vartheta_k \\ &Z_k = \begin{pmatrix} 1 & g_1(X_k)^T \end{pmatrix} A_k + \nu_k \\ &X_{k+1}^{DR} = X_k^{DR} + h(X_k^{DR}, u_k^-) \end{aligned} \tag{3.26}$$

where,

$$E[\vartheta_k \vartheta_k^T] = Q = \begin{pmatrix} Q_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad E[\lambda_k \lambda_k^T] = R. \tag{3.27}$$

The nonlinear Kalman Filter update equations become:

$$
\begin{aligned}
&P_{k+1}^{-} = P_k + Q, \begin{pmatrix} \hat{X}_{k+1}^{-} \\ \hat{A}_{k+1}^{-} \end{pmatrix} = \begin{pmatrix} \hat{X}_k \\ \hat{A}_k \end{pmatrix} + B_k (X_{k+1}^{DR} - X_k^{DR}) \\
&H_k = \frac{\partial g_k}{\partial (\hat{X}_k^{-}, \hat{A}_k^{-})} = \begin{pmatrix} \hat{A}_k^{-T} \begin{pmatrix} 0 \\ \dfrac{\partial g_1(\hat{X}_k^{-})}{\partial \hat{X}_k^{-}} \end{pmatrix} & \begin{pmatrix} 1 & g_1(\hat{X}_k^{-})^T \end{pmatrix} \end{pmatrix}. \\
&P_{k+1} = \left(\left(P_{k+1}^{-} \right)^{-1} + H_{k+1}^T R^{-1} H_{k+1} \right)^{-1} \\
&\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}_{k+1}^{-} \\ \hat{A}_{k+1}^{-} \end{pmatrix} + P_{k+1} H_{k+1}^T R^{-1} \left[Z_{k+1} - g_{k+1}(\hat{X}_k^{-}, \hat{A}_k^{-}) \right]
\end{aligned}
\tag{3.28}
$$

3.2.5 Extended Kalman Filter estimation for a field represented by Radial Basis Function (RBF) neural network

We now consider sampling problems for distributed fields with increased complexity, including nonlinearities in both the parameters and locations. In this case, we can use Gaussian (or radial) basis functions. Hence, the field is represented by a neural network having radial bases (i.e. RBF), with a Gaussian activation function. The use of such model is motivated by ability to approximate complex non-linear parametric fields by the Universal Approximation Theorem. Modeling of complex forest fires by RBF-NN is considered in great detail in Chapter 4, but here a general framework is shown. The measurement model is given by:

$$
Z_k = h(X_k, A_k) + \nu_k = a_0 + \sum_{i=1}^{n} a_i \exp\left[-\sigma_i \left\{ (x_k - x_{0i})^2 + (y_k - y_{0i})^2 \right\} \right] + \nu_k \,. \tag{3.29}
$$

The Neural Network, as well as the corresponding EKF equations can now be summarized as shown in Figure 3.1.

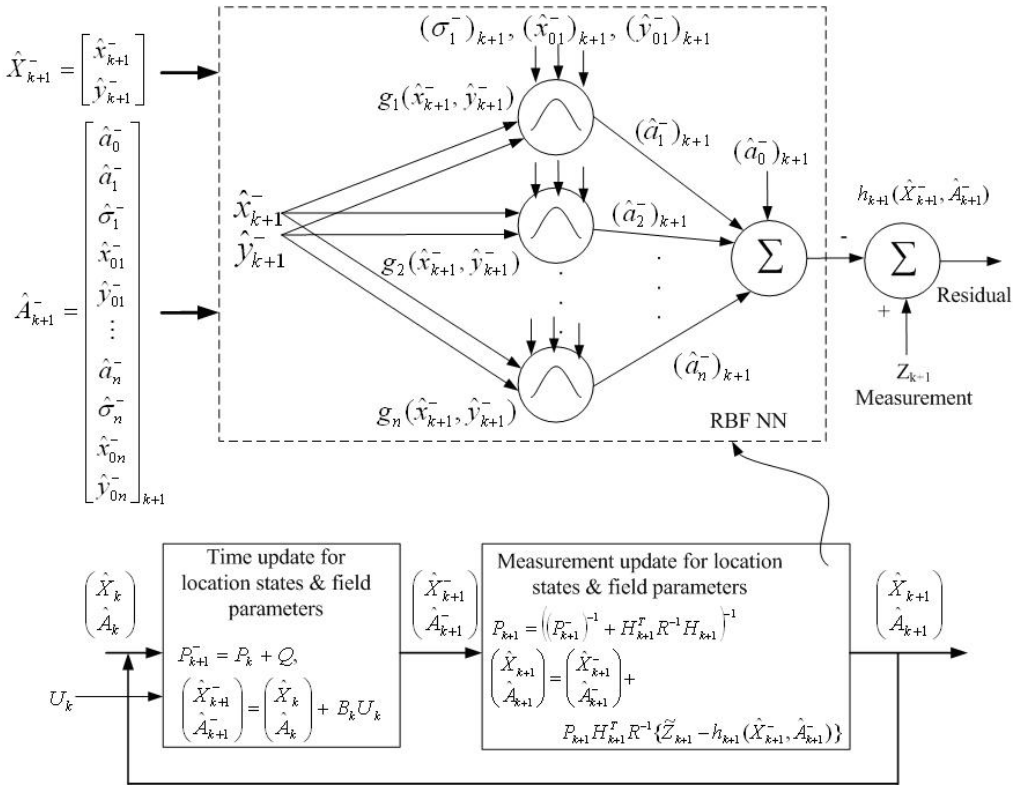


Figure 3.1 A radial basis neuro-EKF to represent a non-linear field and estimate the field parameters and robot location states.

3.3 RBF neural network for parameterization

As explained in Chapter 2, parametric approximations of complex fields can be achieved using artificial neural networks. Apriori information of the field is used to approximate the continuous field with an RBF neural network according to the universal approximation theorem. The inputs (states) and outputs (field measurements) are used to train the neural network. Training is done at a lower rate in a multi-rate EKF estimation process (e.g. of sampling with robots). Using the training algorithm, the network adjusts its weights so that the error between the actual and desired response is minimized.

RBF is a two-layer network. The hidden layer has radial (Gaussian) bases while the output layer is linear. Both layers have biases, and the output of the network is simply:

$$g = \sum_{i=0}^{N} w_i \Phi_i \left(\left\| (X_i - c_i) s_i \right\| \right)$$
$$\Phi_i = \exp\left(- \left\| (X_i - c_i) s_i \right\| \right) = \exp\left\{ - s_i \sum_{j=1}^{P} \left(X_j - c_{ij} \right)^2 \right\}, \tag{3.30}$$

where X is the field location, $X \in \Re^P$, where P is the dimension of the field e.g. two or three dimensional distribution. It is desired to learn parameters $w_0, w_1, ..., w_N$, centers $c_1, c_2, ..., c_N$ and the spread parameter $s_1, s_2, ..., s_N$ of the network. The spread parameters are the measure of the variance of the Gaussians. The hidden and output layer of the RBF network can be optimized separately by a hybrid algorithm, the "self-organized selection of centers" [14, 45, 83]. This algorithm uses *k*-means clustering for placing the Gaussian centers, heuristic *r*-nearest neighbors rule for width optimization, and LMS for estimating the peak value.

A supervised or random approach for the selection of centers will be less useful in this case when compared to the self-organized approach because the former schemes ignore the non-uniformities in the distribution which will give less accurate results. On the other hand, *k*-means clustering adjust the centers using "minimum distance Euclidean distance" criteria, ensuring that there are more Gaussians overlapping in high-variance areas. This results in thorough sampling in high-variance areas due to the

presence of more parameters dominating in those areas when “minimum error covariance” criterion is used to seek optimal sampling locations.

The parameters of the output layer are learned using a LMS algorithm. A hybrid algorithm is also necessary because the hidden layer evolve slowly compared to the linear layer. The selection of spread parameters is crucial. The spread parameters are the measure of the variance of the Gaussians. As there is no optimal scheme to find the spread factor especially in the case of non-uniform distributions, and several fixed and variable width heuristic methods have been proposed in the past [53]. Considering the fact that all the approaches are sub-optimal, a fixed width can be assumed while training as it is corrected by the EKF when high-resolution measurements are taken. In this case the larger spread is, the smoother the function approximation. Too large a spread means a lot of neurons are required to fit a fast-changing function. Too small a spread means many neurons are required to fit a smooth function. Therefore, the accuracy of the initial estimated field depends on resolution of apriori data, number of neurons used and the spread factor. As shown in figure 3.2 & 3.3, input-output data is used for training the RBF neural network which gives an initial estimate of weights in linear layer, non-linear layer and the center of Gaussians. These initial estimates are later used as initial conditions for the parameterized EKF-based sampling algorithm.

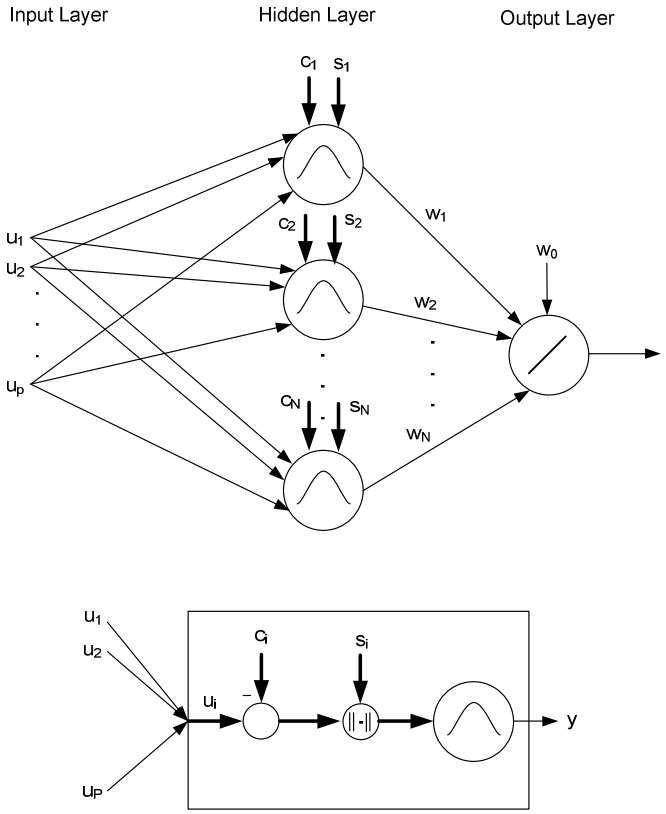


Figure 3.2 RBF NN architecture with p inputs $(u_1, u_2, \cdots, u_p)$ and one output y.

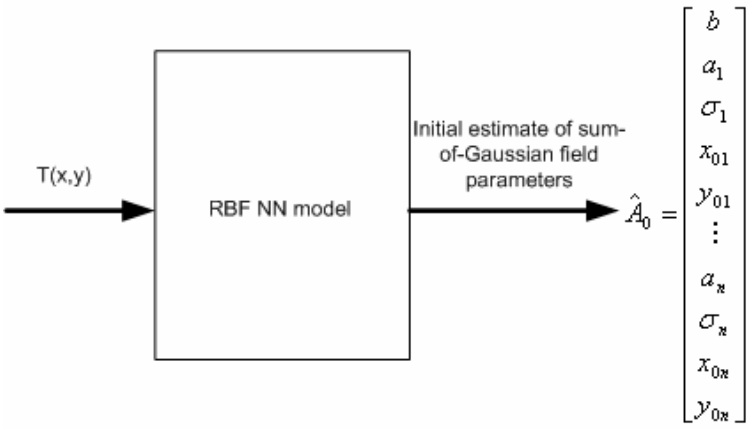


Figure 3.3 Block diagram of inverse modeling for parameter estimation when input to NN are locations (x, y) and field T

3.4 Sampling strategies: where to sample

3.4.1 Global search AS

In the absence of model and measurement uncertainties in localization and field parameters, the sampling task would be a deterministic case and all the sample locations $X_i, (1 \le i \le N)$ will provide same information about the unknown X and A. Hence there would be no difference in random, row-by-row or adaptive sampling. But in the presence of uncertainties discussed, the error covariance (uncertainty) of states can be used to decide the next best sampling location. The error covariance of the parameter states is the measure of uncertainty of the parameters. High value of error covariance indicates higher uncertainty and lower value indicate lower uncertainty.

Adaptive sampling location calculates the anticipated error covariance of all the candidates' location and selects the one which gives minimum value. The minimization in the error covariance of a parameter is achieved by sampling along the extreme (boundary) values of the parameter. A summary of different ways to pick the next sampling location is shown in Figure 3.4.

It will be shown later (Chapters 5, 6) through numerous simulation and experimental results that Adaptive Sampling (e.g. equation (3.18)) requires considerably fewer samples than just Raster Scanning or Random Sampling, because AS minimizes the uncertainty in unknown parameters. However, one drawback of using this standard approach of AS is that it is computationally complex, because it calculates the anticipated error covariance of the locations in the entire search space. To overcome this

drawback, a heuristic greedy AS is used to indirectly incorporate the time-to-move for the robot and reduce complexity.

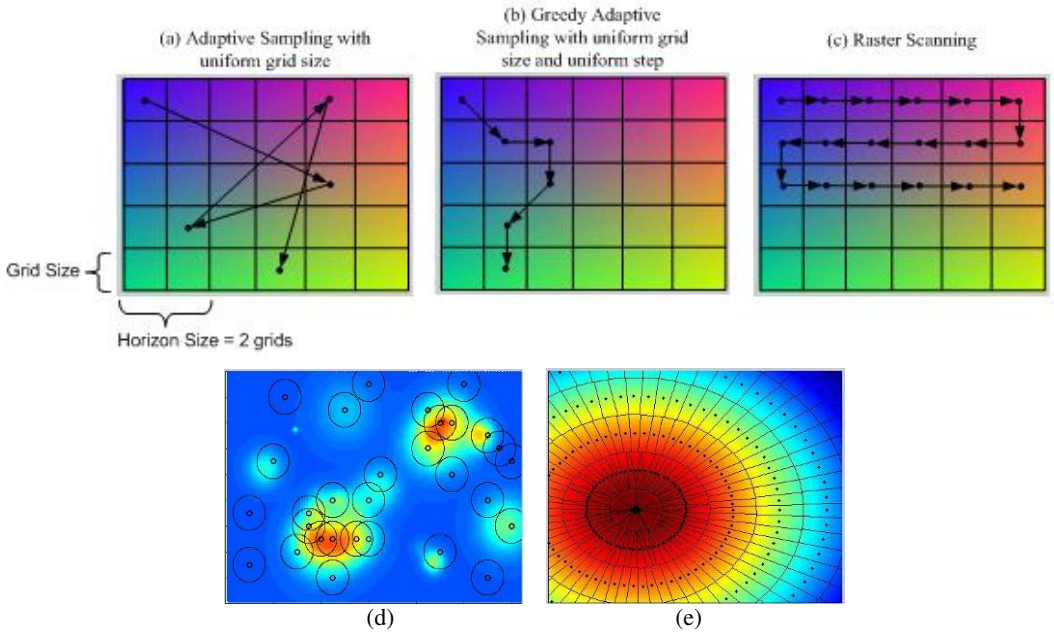


Figure 3.4 (a) Adaptive sampling with uniform grid size, (b) Greedy AS with uniform grid size and uniform step size, (c) Raster scanning, (d) Adaptive Cluster Sampling, (e) Non-uniform grid size sampling

3.4.2 Heuristic Greedy AS

If the sampling area is large, the optimal sampling position could far away from current robot location, and it will not be a efficient in minimize the sampling time. A heuristic greedy approach is adopted instead, where the next sampling location is searched within the vicinity of the currently sampled location. The advantage is that the robot covers less distance which saves time and number of samples. Also it reduces the complexity of the algorithm because only the anticipated error covariance of sampling neighboring locations is calculated. The Heuristic Greedy AS algorithm helps to answer the following question:

Given a parametric field variable $Z = Z(A, X)$ *depending on an unknown parameter vector A and position vector X, and a robotic vehicle/sensor node navigating the field with localization uncertainty, what sampling locations* X_i *(where* $1 \le i \le N$*), in the immediate vicinity of the current sampling location should be chosen such that a trade-off between the uncertainty in estimating the unknown parameter vector and the sampling distance is minimized.*

3.4.3 Multi-step Greedy AS

There are three different cases for choosing the step size, or horizon in greedy AS:

- Look 1-step ahead and move 1-step ahead (single step)
- Look *k*-steps ahead and move *k*-step ahead
- Look k-step ahead and move 1-step ahead

The first case is the simplest one, and involves minimum computations, but since the search space is restricted, it might lead the robot to end up in areas providing limited information about the field. To avoid this, the "look *k*-step ahead and move *k*-step ahead" approach can be adopted with increasing values of k. But the disadvantage of this approach is that it generates a path based on current information about the parameter estimates which are constantly improving. Therefore, the third case ensures that the robot reevaluates sampling location k-steps ahead after each motion command.

3.4.4 Cluster Adaptive Sampling

For a sampling scheme to be efficient, it should consider more samples in regions with large variations compared to regions with small variations. A solution for

non-parametric ocean sampling in [15] requires the adaptive sampling criteria to take more samples in high-variance area and fewer samples in low-variance areas by having a non-uniform size grid formed using Centroidal Voronoi Tesselation (CVT) diagram as shown in figure 3.4 (e). The grid size depends on the local variance of the field.

For parametric fields, more densely sampling non-uniform areas is achieved by placing the centers of the Gaussians while training the RBF such that there are more Gaussians in high-variance areas, as illustrated in figure 3.4 (d). Hence, the adaptive sampling criteria automatically direct the robots to sample thoroughly in dense areas.

Typically, a complex field is represented by numerous of parameters but only a specific set of parameters are dominant in specific regions, which for Gaussian fields is generally within the $1-\sigma$ circle. As it will be shown in the simulations of Chapter 4, the uncertainty in parameters of a Gaussian is most reduced by sampling within this circle. Therefore, taking the 2-*norm* of error covariance of those parameters whose $1-\sigma$ circle includes the next candidate sampling location is a good choice. And it can also be assumed that the candidate sampling location will provide estimates only for few neighboring parameters. This approximation also helps in distributing the computations among multiple robots which will be discussed in Chapter 7.

In summary, this Chapter introduced the use of the EKF for sampling of parametric fields, discussed how to incorporate measurement and robot location uncertainty, discussed the use of RBF Neural Networks to approximate complex fields, and the heuristics that can be used to reduce the computational complexity of the search necessary to select sampling locations for the robots.

CHAPTER 4

ADAPTIVE SAMPLIMG ALGORITHM APPLIED TO FOREST FIRE MAPPING

The chapter is organized as follows: in section 4.1 the two parametric models used to describe the spread of forest fires are presented, section 4.2 discusses the parameterization of the field by interpreting remote-sensing images, section 4.3 presents the formulation for EKF-based adaptive sampling algorithm for spatio-temporal distributions and the multi-scale algorithm for mapping of forest fires using adaptive sampling, and section 4.4 discusses potential fields based path planning for robots navigating through the estimated fire field.

4.1 Parametric description of forest fire spread

The physics of the fire behavior, its spread, and immediate effects have been studied extensively [5, 9, 18, 36]. Fuel, weather and topography are the key considerations in the spreading of fire. Most common are the semi-empirical fire modeling that uses Rothermel's equation for calculation of local rate of fire spread [9, 36], Huygen's principle for modeling the shape of fire front [9, 36], and the use of discrete-event cellular automata models [19, 24, 28, 42]. Furthermore, neural network and other classification schemes have been used for some time for supervised classification of remote sensing data, especially in applications of urban planning and

atmospheric modeling, for instance, in order to classify the area as water, forest, wetlands, lakes, etc [24]. Since the fire spread phenomena is too complex to understand, and the effect of each and every variable is difficult to predict, a neural network can be used as a "black box" for measuring the influence of the independent variables. For instance, representing the fire spread as a parametric time-varying sum of several Gaussians makes for an attractive approximation.

In this thesis, complex forest fires distributions are modeled through the use of RBF (Radial Basis Function) Neural Networks. RBF parametric models resulting from a low-resolution satellite image, for instance, serve as good low-resolution initial approximations for the fire field. When combined with a high-resolution adaptive sampling strategy, the errors introduced by low-resolution sampling and training are reduced. Sampling is done heuristically by mobile robots (agents) that search in the vicinity of the current location for future sampling locations minimizing field parameter uncertainty. This results in a quick, high-resolution reconstruction of the field at the end of the sampling process. Furthermore, the parametric description of the field can be used along with the robot dynamic model to reduce the localization uncertainty of the agents. This may not be needed in scenarios where GPS measurements are available, but it is critical in GPS-denied environments, such as underwater or inside buildings.

The validity and performance of the sampling algorithm is tested here with increasingly complex fire spread scenarios. First, a simple time-varying elliptical forest fire spread model is considered that has been fitted to empirical data. This model considers a decreasing intensity Gaussian distribution from the head of fire to the fire

tail. The model is accurate at the beginning of the fire spread, and less accurate as time goes on. In later sections of the Chapter, we consider more realistic fire spread models simulated by cellular automata (CA), using discrete rules to decide the burning of certain cells based on conditions at neighboring cells. A low-resolution satellite image is used to approximate this field by a RBF neural network training algorithm that also acts as a classifier. The objective of the classification is to introduce more neurons in high-variance areas of the field, and fewer neurons in low-variance areas. Training of the neural net is only needed infrequently, and it need not be very accurate.

In addition to sampling and reconstruction of the fire field, it is sometimes necessary to direct the robots or human personnel through this field and avoid dangerous locations, e.g. “hot zones”. This type of navigation can be accomplished using Potential Fields [11], and the estimated fire field model itself can serve as the navigation potential. Extensive research has been done in the area of path planning using potential fields for robotic vehicles [22]. Obstacle avoidance or goal attainment schemes often use penalty functions to bend feasible paths around obstacles as it was introduced by Krogh [21] and Khatib [20]. As the fire intensity field is generated via sampling, we can use it to generate fire-safe trajectories to and from rescue locations for fire-fighter human crews.

The sampling and navigation algorithms presented in this Chapter will be validated in Chapter 5 using extensive realistic, time-varying forest fire simulation models. In Chapter 6, using a camera-projector system and mobile wireless robot nodes, ARRI-Bots, the algorithms will also be validated experimentally with relative ease.

The fire spreading phenomenon is highly complex, and existing mathematical models of its spread require extensive computing resources. Regenerating time-varying fire spread models might be adequate to run in batch mode, but they are difficult to use in real-time fire-fighting scenarios. However, these models help in understanding the influence of different factors on fire propagation without the need for an analytical solution.

Existing fire spread models are divided into three classes: empirical, semi-empirical and physical models. Using semi-empirical models is particularly interesting for our sampling approach because they use empirical measurements in addition to a physical description of the forest, hence providing a parametric space-time distributed field whose parameters can be estimated by adaptive sampling. By contrast, physical models are too simplistic, while empirical models involve too many parameters for the EKF. The most common fire modeling approaches include semi-empirical models using Rothermel's equation for calculation of local rate of fire spread, Huygen's principle for modeling the shape of fire front and Discrete-Event Cellular Automata (CA) models. Several powerful software applications such as FARSITE, FireGIS, Fire!, SPREAD, etc are available to demonstrate the fire spread .

Many factors influence fire behavior, with fuels, weather and topography being the primary factors. Parametric models have been empirically fitted to observe data using four primary inputs: fuel type, fuel moisture, wind and slope. Second-order variables such as temperature, humidity, shading and shelter operate through one of the four primary inputs. The effect of the primary inputs is described below:

- Slope: fire can spread significantly faster up a slope than on level terrain in the same fuels. In many cases topographic maps are very helpful for understanding the fire spread.
- Wind: wind speed and direction are the most critical factors required for predicting fire behavior.
- Fuel: fuel types are lumped together into different models. Fuel loading, fuel depth, fuel particle density, head content of fuel, moisture of extinction and surface area to volume ratio are the factors used to further describe it.
- Fuel moisture: fuel moisture is an expression of the amount of water in the fuel component. It determines both fire intensity and the heat required to bring the fuel ahead of spreading fire up to the ignition temperature.

4.1.1 Simple elliptical fire-spread model

Given homogenous fuel and weather conditions, and assuming a constant, moderate wind, a fire growing from an ignition point is initially close to an elliptical shape. Strong winds or steep slopes can elongate the shape, but it remains consistent until fuels or wind change. The spread distance is the product of projection time and the rate of spread.

The most common simulation model for forest fires growth from point sources uses the Huygens' principle, which considers the elliptical nature of fire growth with (a,b) as the minor and major axes respectively and (x_s, y_s) as the distance from the center of the fire ellipse to the fire source. Figure 4.1 depicts the elliptical spread of a fire starting at an ignition point. Since this function is a constrained Gaussian, it will

later be used as a Radial Basis Function (RBF) function with constraints. For a typical case the dimensions of the elliptical axis can be calculated by the following equations [9, 36]:

$$a = 0.5\frac{R + R/HB}{LB}, b = \frac{R + R/HB}{2}, c = b - R/HB$$
$$LB = 0.936.e^{0.2566U} + 0.461.e^{-0.1548U} - 0.937 \quad (4.1)$$
$$HB = \frac{LB + (LB^2 - 1)^{0.5}}{LB - (LB^2 - 1)^{0.5}}$$

where R is the fire spread rate in ft/min, LB is the length to breadth ratio, HB is the head to back ratio, and U is the wind speed.

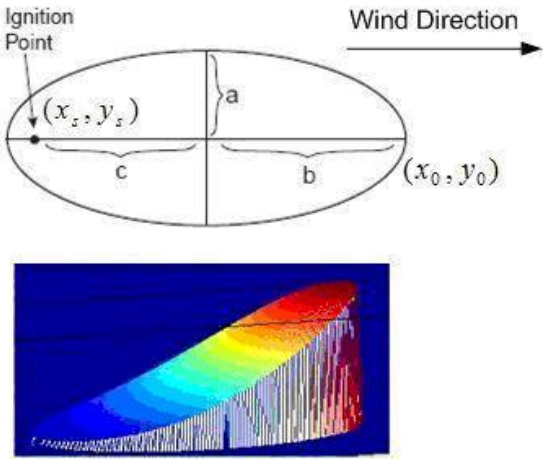


Figure 4.1 Elliptical Fire growth model represented by an elliptically constrained RBF function.

The variable which will later need to be mapped is the temperature of the fire field, a Gaussian distribution only defined in an elliptical region. It is further assumed that the maximum fire temperature I, variance σ and y_0 are time invariant, while the firehead $(x_0(t), y_0(t))$ location is proportional to the fire spread rate which in turn is

proportional to the wind speed. If $y_0(t) = C$ (e.g. fire spreads horizontally), then the fire intensity is given by:

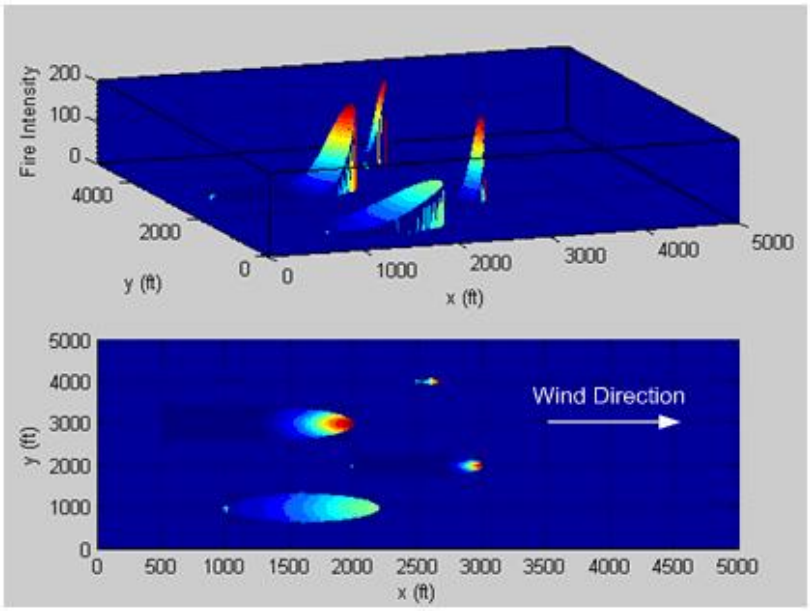


Figure 4.2 Four fire ellipses originated at different points and with different spread rate, peak intensity and variance. The head of fire is the mean of the Gaussian distribution

$$g(X,A,t) = \begin{cases} I\exp\left[-\dfrac{(x-x_0(t))^2+(y-y_0)^2}{2\sigma^2}\right], & \dfrac{(x-x_0(t)+b(t))^2}{b(t)^2}+\dfrac{(y-y_0)^2}{a(t)^2} \le 1 \\ 0 & otherwise \end{cases}, \tag{4.2}$$

where

$$x_0(t) = x_s + b(t) + c(t) \ . \tag{4.3}$$

If wind is blowing at an angle ϕ to the x-axis, the new parametric equation of ellipse can be derived by a rotation around the fire source (x_s, y_s) by angle ϕ. In this case in equation (4.2) becomes:

$$g(X_{new},A,t) = I\exp\left[-\frac{\| X_{new}-X_s \|^2 + \| X_s - X_0 \|^2 + 2(X_{new}-X_s)^T R_\phi (X_s - X_0)}{2\sigma^2}\right], \tag{4.4}$$

where R_{ϕ}^{T} is a 2x2 rotation by angle ϕ, and

$$X_{new} = \begin{bmatrix} x_{new} & y_{new} \end{bmatrix},\ X_o = \begin{bmatrix} x_o & y_o \end{bmatrix},\ X_s = \begin{bmatrix} x_s & y_s \end{bmatrix}. \quad (4.5)$$

4.1.2 Complex Cellular Automata (CA) based discrete event model

The Huygens fire-spread model is quite simple, but it is quite reasonable for short intervals of time after ignition. Because it assumes uniform, continuous fuel, uniform wind velocity throughout the burning area and flat terrain, the model becomes inaccurate if the fire spreads over large distances. Cellular Automata (CA) are very good for modeling and simulating complex dynamical systems whose evolution depends on the state of the current cell, neighboring cells, wind, slope and fuel. This scheme can also recover the Huygens fire spread model discussed in previous section at a cellular level. The fire spread region is divided into cells, as shown in figure 4.3, and the fire propagates from cell to cell governed by discrete-event rules which iterate the temperature at the next sample time as a function of the temperature of its cell neighbors, and of several local fire parameters. The transition rules can be written as:

$$T_{i,j}^{k+1} = f(T_{i,j}^{k}, T_{i,j,1}^{k}, ..., T_{i,j,4}^{k}, wind, slope, fuel), \quad (4.6)$$

where $T_{i,j}^{k}$ are temperatures of the current cell and its neighbors.

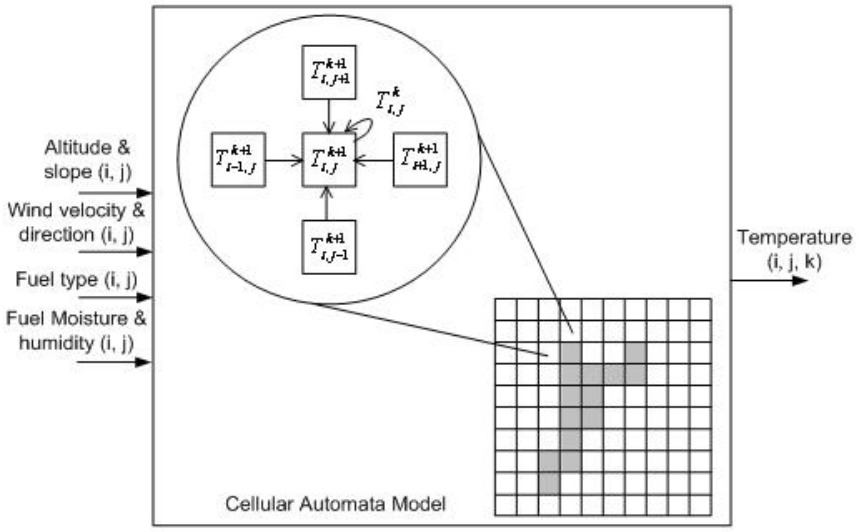


Figure 4.3 CA model inputs, outputs and transition rules

Each cell is considered to be in one of the following four states (*inactive*, *unburned*, *burning* and *burned*) as shown in figure 4.4. The fire starts from a hotspot in an elliptical shape in the sense that each burning cell generates an ellipse with focus on the cell center. Depending on fuel type, moisture, slope, wind and the state of the surrounding cells, each may have different rate of increase of temperature (α), maximum temperature $(T_{\max})$, ignition temperature $(T_{ignition})$, rate of temperature decay (β), time threshold for switching to unburned state (t_{thresh}) and ignition time $(t_{ignition})$.

The following set of equations govern the temperature variation with respect to time, as suggested by [42]:

$$T(t) = \begin{cases} T_a & ; t \le t_{th} \\ T_a + e^{(t-t_{th})\alpha} & ; t_{th} < t \le t_{\max} \\ T_{\max} & ; t_{\max} < t \le t_{\max f} \\ T_a + (T_{\max} - T_a).e^{(t_{\max f} - t)\beta} & ; t > t_{\max f} \end{cases} \quad (4.7)$$

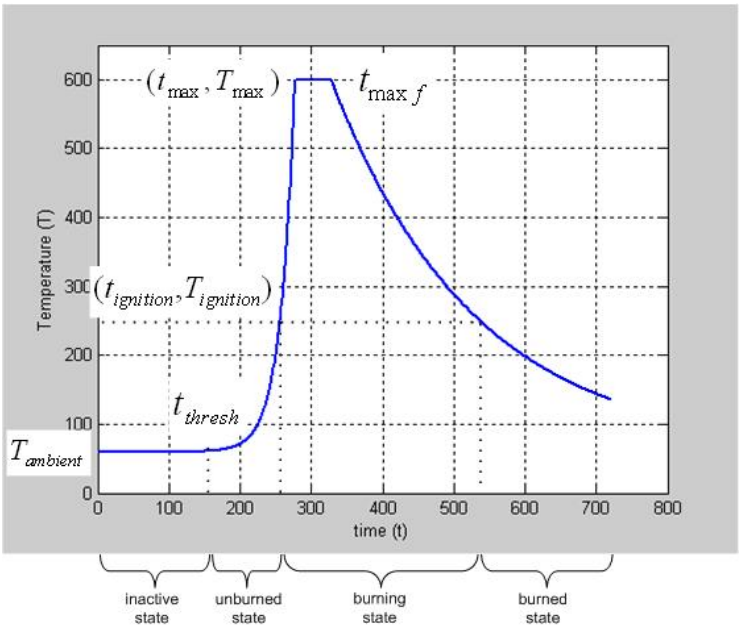


Figure 4.4 Cell temperature variations with respect to time

The rules can be simply defined as:

- A cell remains in *inactive* state until it is outside the *burning* ellipse. The state switches to *unburned* state when one of its neighbors starts *burning*.
- Once the cell switches from *inactive* state to *unburned* state, its temperature starts to rise at a rate of α, which depends on the fuel availability, wind and terrain.
- The cell starts *burning* after it reaches the ignition temperature. The combustion duration and rate of temperature decay β depends on fuel, wind and terrain.

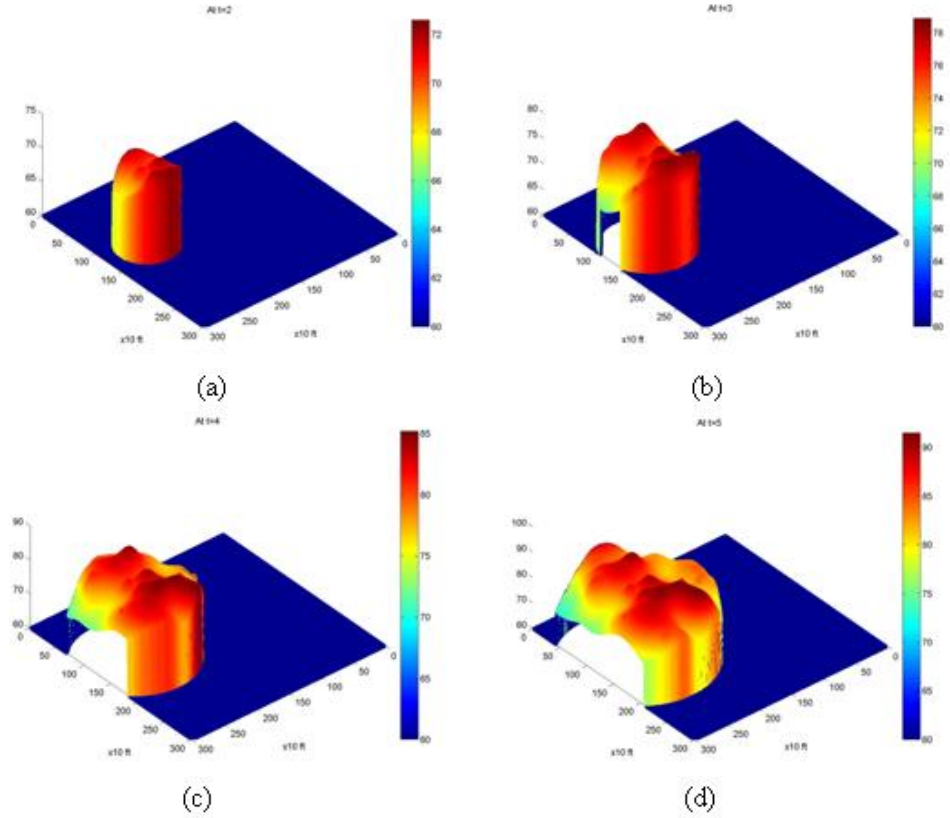


Figure 4.5 Fire at time t = 2, 3, 4 & 5 minutes shown in (a), (b), (c) and (d) respectively

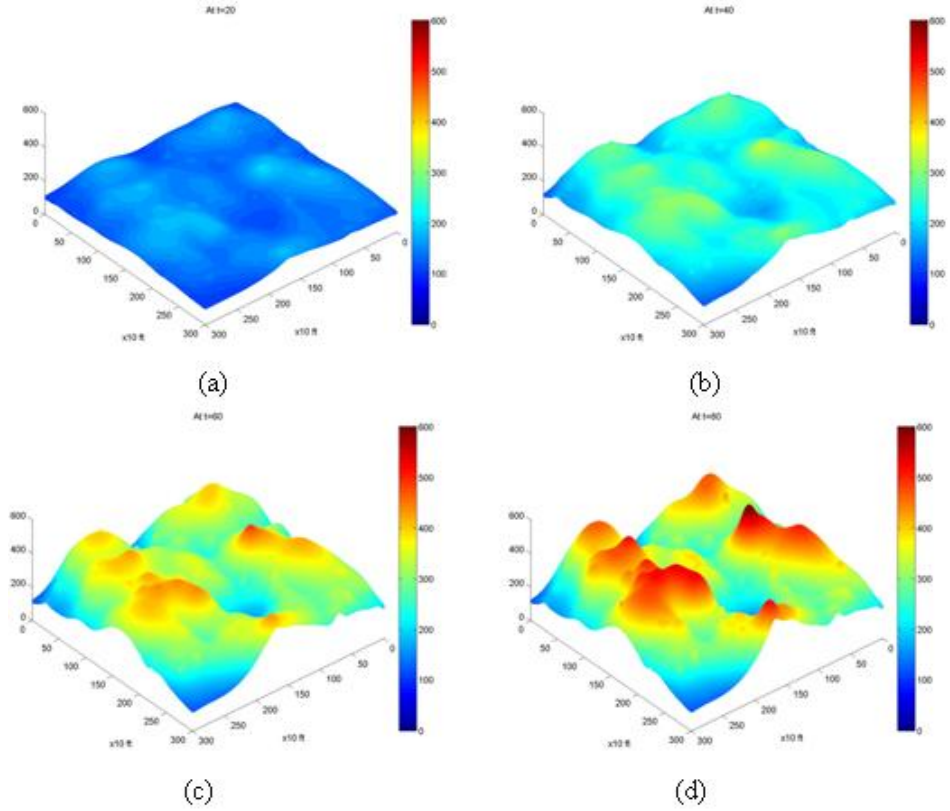


Figure 4.6 Fire at time t = 20, 40, 60 & 80 minutes shown in (a), (b), (c) and (d) respectively

Figure 4.5 shows a MATLAB simulation of a fire generated from the hot spot at (244,127) using CA at time instants t = 2, 3, 4 and 5 minutes. The number of pixels for the fire field was $x \times y = 300 \times 300$. The fire is circular in shape because slope variation and wind direction are not considered. But because of the different fuel present in each cell the temperature is different and keeps increasing. Figure 4.6 shows the fire at time t = 20, 40, 60 & 80 minutes in figures (a), (b), (c) & (d) after some time elapsed and most of the area with significant fuel content is burning. In Chapter 5, these fire field models will be sampled and reconstructed using the algorithms discussed in Chapter 3.

4.2 Neural network for parameterization

Digital remote sensing images of forests can be acquired from field-based, airborne, and satellite platforms. Imagery from each platform can provide a data set with which forest analysis and modeling can be performed. Airborne images typically offer greatly enhanced spatial and spectral resolution over satellite images. In addition, airborne images provide greater control over capturing images only from specific areas of interest and at different altitudes. In section 3.3, we discussed the parameterization of complex time varying fields which is applied here to a fire spread model using RBF neural nets. We assume that low-resolution images are available from overhead imagery at infrequent time intervals.

In the case of fire distribution, an exact nonlinear model description is unattainable due to the high level of complexity. Instead, a parameterized approximation of the field is used, which is acquired by means of a neural network. In order to obtain an initial approximation of the field, a neural network is trained with a low-resolution “fire field image”. Training is done at lower rates in this EKF estimation process (e.g. of sampling with robots). The network is presented with training pairs, which in our case are temperatures at different locations taken from a low-resolution infra-red image. Therefore, we roughly approximate the complex spatio-temporal field with a sum-of-Gaussian parametric field by means of the Universal Approximation Theorem.

4.3 EKF Adaptive Sampling of spatio-temporal distributions using mobile agents

The presented nonlinear EKF approach is an efficient framework for combining the uncertainty in robot localization with errors in field sensor measurements. State-measurement minimization of the EKF covariance matrix norm is used to achieve effective adaptive sampling using a variety of mobile robotic platforms including underwater and in-door vehicles. Localization uncertainties are especially relevant in GPS-denied environments, but they can also be relevant to fire fields in situations where local visual information is unavailable (for instance due to thick smoke), and GPS data rates are slow. Time-varying complex models are considered here stemming from realistic fire-spreading simulations of the previous sections.

Model parameter estimation for the field variable (fire temperature) is integrated with estimation of the uncertainty in the mobile robot localization and the overall estimate covariance is used for sampling. This way, localization uncertainty can be reduced by building accurate models of distributed fields and vice versa. For instance, if a robot is sampling an unknown field, but its location is accurately known, a distributed parameter field model can be constructed by taking repeated field samples. Later on, this field model can be used to reduce the localization error of the robot.

The multi-agent AS problem considered can be described as follows:

Assumptions:

i) A nonlinear spatio-temporal field variable is described via a parametric approximation Z=Z(A, X, t) depending on an unknown parameter vector A, position vector X, and time t.

ii) N robotic vehicles (agents) sample the field with localization and sensing uncertainty in order to obtain higher resolution estimates of the field, and also to improve their own location estimates.

iii) The number of field parameters (M) and their initial guess is based on a hypothesis originating from prior knowledge of the field consistent with a low-resolution image of the entire field.

In this section, the mapping of a slow time-varying complex forest fire field is considered, with measurements performed at different rates. For sampling slow-varying fields, the time update is performed at a slower rate compared to the measurement update. The time update (Equation 3.8) is performed at a sampling rate of T_d by high-spatial infra-red imaging with an uncertainty represented by the process covariance noise Q_{Td}. The field evolution is measured by the difference $(\tilde{U}_{Td+1} - \tilde{U}_{Td})$ between these consecutive measurements. High rate field measurement updates are then done by robotic sampling, with uncertainty represented by covariance noise R_{2k}.

Assuming that the i^{th} robot location Y_k^i is measured using some absolute localization scheme such as GPS, and if, for simplicity, we ignore the robot dynamics

and localization uncertainty, we can write the state and output equations as:

$$\begin{aligned} A_{k+1} &= A_k + (\tilde{U}_{Td+1} - \tilde{U}_{Td}) + \alpha_{Td+1} \\ \tilde{U}_{Td+1} &= U_{Td+1} \\ Y_k^i &= X_k^i \\ Z_k &= g(Y_k^i, A_k) + \nu_k \end{aligned} \quad . \tag{4.8}$$

Therefore, the EKF equations become:

$$\begin{aligned} & P_{k+1}^- = P_k + Q_{Td+1},\ \hat{A}_{k+1}^- = \hat{A}_k + (\tilde{U}_{Td+1} - \tilde{U}_{Td}) \\ & H_k = \frac{\partial g_k}{\partial (Y_k^i, \hat{A}_k^-)} \\ & P_{k+1} = \left(\left(P_{k+1}^- \right)^{-1} + H_{k+1}^T R^{-1} H_{k+1} \right)^{-1} \\ & \hat{A}_{k+1} = \hat{A}_{k+1}^- + P_{k+1} H_{k+1}^T R^{-1} \left[Z_{k+1} - g_{k+1}(Y_{k+1}^i, \hat{A}_{k+1}^-) \right] \end{aligned} \tag{4.9}$$

where, $Q_k = Q_{Td+1},\ R_k = R_{2k}$.

4.3.1 Formulation for elliptically constrained single Gaussian time-varying field

In this section, a time-varying field is assumed where the input U_{2k} depends on the measured velocity of spread (wind velocity). The Kalman Filter equations are used to estimate the peak intensity, variance and mean of the time-varying forest fire field. Using equation, the fire model equation can be written as:

$$\begin{aligned} & A_{k+1} = A_k + B_{2k} U_{2k} + \alpha_k \\ & \begin{bmatrix} I \\ \sigma \\ x_0 \\ y_0 \end{bmatrix}_{k+1} = \begin{bmatrix} I \\ \sigma \\ x_0 \\ y_0 \end{bmatrix}_k + \begin{bmatrix} 0 \\ 0 \\ r_{k+1} \Delta t_{k+1} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \alpha \\ 0 \end{bmatrix} \quad , \\ & \alpha \sim N(0, Q_2) \end{aligned} \tag{4.10}$$

where Δt_{k+1} is the time from sample k to $k+1$, x_0 is a continuous function which is sampled at time $t_0, t_1, ...t_{k+1}$, and r is a time-varying function describing the velocity of spread, and r_{k+1} is the velocity of spread r for $(k+1)$ th sample.

The state estimates are updated using equation (4.10). The measurement equations are given by equations (4.11) and (4.12) with noise covariance $v_1 \sim N(0, R_1)$. The time and measurement update equations are given by:

$$\begin{aligned}
&\hat{I}_{k+1}^{-} = \hat{I}_k,\ \hat{\sigma}_{k+1}^{-} = \hat{\sigma}_k,\ (\hat{y}_0)_{k+1}^{-} = (\hat{y}_0)_k \\
&(\hat{x}_0)_{k+1}^{-} = (\hat{x}_0)_k + r_{k+1}\Delta t_{k+1} \qquad , \\
&P_{k+1}^{-} = P_k + Q_{k+1}
\end{aligned} \tag{4.11}$$

and

$$\begin{aligned}
&\widehat{H}_{k+1} = \frac{\partial \hat{g}_{k+1}}{\partial(\hat{A}_{k+1}^{-})} = \left[\frac{\partial g}{\partial \hat{I}_0^{-}} \quad \frac{\partial g}{\partial \hat{\sigma}^{-}} \quad \frac{\partial g}{\partial \hat{x}_0^{-}} \quad \frac{\partial g}{\partial \hat{y}_0^{-}}\right]_{k+1} \\
&= \left[C \quad \hat{I}_{k+1}^{-}.(\hat{\sigma}_{k+1}^{-})^3.\left\{(x_{k+1} - \hat{x}_{0k+1}^{-})^2 + (y_{k+1} - \hat{y}_{0k+1}^{-})^2\right\}.C \right. \\
&\left. \frac{\hat{I}_{k+1}^{-}.(x_{k+1} - \hat{x}_{0k+1}^{-})}{(\hat{\sigma}_{k+1}^{-})^2}.C \quad \frac{\hat{I}_{k+1}^{-}.(y_{k+1} - \hat{y}_{0k+1}^{-})}{(\hat{\sigma}_{k+1}^{-})^2}.C\right] \\
&where\ \ C = \exp\left[-\frac{(x_{k+1} - \hat{x}_{0k+1}^{-})^2 + (y_{k+1} - \hat{y}_{0k+1}^{-})^2}{2(\hat{\sigma}_{k+1}^{-})^2}\right] \qquad .
\end{aligned} \tag{4.12}$$

$$\begin{aligned}
&K_{k+1} = P_{k+1}^{-}\widehat{H}_{k+1}^{T}(\widehat{H}_{k+1}P_{k+1}^{-}\widehat{H}_{k+1}^{T} + R_{k+1})^{-1} \\
&P_{k+1} = (I_{12} - K_{k+1}\widehat{H}_{k+1})P_{k+1}^{-} \\
&\hat{A}_{k+1} = \hat{A}_{k+1}^{-} + K_{k+1}\{\tilde{z}_{k+1} - \hat{g}_{k+1}\} \\
&= \hat{A}_{k+1}^{-} + K_{k+1}\left\{\tilde{z}_{k+1} - \hat{I}_{k+1}^{-}.\exp\left[-\frac{(x_{k+1} - \hat{x}_{0k+1}^{-})^2 + (y_{k+1} - \hat{y}_{0k+1}^{-})^2}{2(\hat{\sigma}_{k+1}^{-})^2}\right]\right\}
\end{aligned}$$

4.3.2 Formulation of the general multi-scale algorithm EKF-NN-GAS for fire fields

In previous sections we considered different cases of increasing complexity, in which a robots sample a parameterized field distribution. In a practical scenario, measurements will be arriving from different sensors, at different dimensional and time scales, and must all be fused into the EKF filter. We propose the following sampling algorithm to negotiate the increased complexity of mapping the fire field:

Algorithm EKF-NN-GAS (multi-scale, multi-rate adaptive sampling)

Step 1 (Setup): Give the environmental parameters and rules for fire spread as input to the CA model as shown in figure 4.8. This will generate a 2-D temperature field $T(x, y, t)$ which has a dimension of $m \times m$ at time t. In practice, this step is omitted, and replaced by the actual spread of the fire. Go to step 2.

Step 2 (Initialization): Divide the field into square size grid of $n \times n$, $n < m$, and average values in each grid. This gives a low-resolution version of the actual field of size $m/n \times m/n$, illustrated in figure 4.8. In practice this low-resolution temperature distribution is acquired by an infra-red image taken from an airborne platform. Go to step 3.

Step 3 (Training): Train the RBF neural network using this low-resolution temperature data. In the training algorithm, the number of neurons and smoothness factor are specified. The number of neurons depends on the complexity of the field so that the error is minimized within an acceptable threshold. This gives the parameterized version of the field with N neurons and each neuron has parameters a, σ, x_0 and y_0

representing this RBF field. The error in the actual field and the initial estimate using neural network also gives a guess for initial error covariance P in later EKF steps. Go to step 4.

Step 4 (High rate sampling): Spot measurement robots sample locations in a grid of size $p \times p$, (where $p \leq n < m$) based on a Greedy Adaptive Sampling criterion to minimize the error covariance. The EKF framework shown in figure 4.8 is used to correct the estimates as the subsequent measurements are available one by one. The robot location is calculated by GPS measurement, via dead-reckoning, or relative position measurements. Localization uncertainty is ignored in the simulation results of the next Chapter, but should be considered along with the robot dynamical model. The EKF sampling rate, T should be as fast as sensory measurements from robots are available. Repeat Step 4 until new low resolution updates of the entire field are available, else go to step 5.

Step 5 (Low rate sampling): Because of the time varying nature of the field, a low resolution update of the field is performed at a sampling rate T_d, and the evolution of the parameters is updated by comparing the low resolution field distribution at time T_{d+1} and T_d. This involves repeating step 3 by retraining the neural network at time T_{d+1} and updating the parameters since the last training at time T_d. The low rate parameter update $(\tilde{U}_{Td+1} - \tilde{U}_{Td})$, and its uncertainty Q, are given as inputs to the EKF block for high rate sampling update shown in figure 4.8. Low rate sampling is performed at a very low rate (approximately every 5 minutes for the simulation

examples we considered), compared to the high rate of sampling (which takes several sensor measurements every minute for the simulation examples). This scheme gives a better estimate of the time-varying field than if it was not included in the parameter time update. Go to step 4.

Since the sampling robot dynamics is ignored here for the sake of simplicity, the parameter estimation model is simply:

$$\begin{aligned} &A_{k+1} = A_k + \alpha_k = \begin{bmatrix} b & a_1 & \sigma_1 & x_{01} & y_{01} & \dots & a_N & \sigma_N & x_{0N} & y_{0N} \end{bmatrix}_k^T + \alpha \\ &\alpha \sim N(0, Q) \end{aligned} \quad .(4.13)$$

Moreover, if the field is time-varying the field parameters propagate as:

$$A_{k+1} = A_k + (\tilde{U}_{Td+1} - \tilde{U}_{Td}) + \alpha \ , \quad (4.14)$$

where A_{k+1} is the field parameters update when $(k+1)^{th}$ sample is taken (step 4) and $(\tilde{U}_{Td+1} - \tilde{U}_{Td})$ is the field parameters propagation obtained by performing step 5. The measurement model is:

$$\begin{aligned} &Z_k = h(A_k) + \nu_k = b + \sum_{i=1}^{N} a_i \exp\left[-\sigma_i \left\{ (x_k - x_{0i})^2 + (y_k - y_{0i})^2 \right\} \right] + \nu \\ &\nu \sim N(0, R) \end{aligned} \ , \quad (4.15)$$

while the low rate sampling update equations are given by:

$$\begin{aligned} &\hat{A}_{k+1}^{-} = \hat{A}_k + (\tilde{U}_{Td+1} - \tilde{U}_{Td}) \\ &P_{k+1}^{-} = P_k + Q \end{aligned} \quad . \quad (4.16)$$

Step 5 does not reset the previous state estimates, but calculates the evolution of parameters $(\tilde{U}_{Td+1} - \tilde{U}_{Td})$ based on the difference between current and past field images added to the old parameter estimate $\hat{A}_k$ to predict a new estimate $\hat{A}_{k+1}^{-}$.

Finally, equations similar to (4.12) can be used in the EKF sampling update. Figure 4.8 shows a block diagram for algorithm *EKF-NN-GAS*. Its full update equations are given by the following block equations:

$$\hat{H}_{k+1} = \frac{\partial \hat{h}_{k+1}}{\partial (\hat{A}_{k+1}^{-})} =$$

$$\left[\frac{\partial \hat{h}_{k+1}}{\partial \hat{b}_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{a}_1)_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{\sigma}_1)_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{x}_{01})_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{y}_{01})_{k+1}^{-}} \right.$$

$$\left. \cdots \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{a}_N)_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{\sigma}_N)_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{x}_{0N})_{k+1}^{-}} \quad \frac{\partial \hat{h}_{k+1}}{\partial (\hat{y}_{0N})_{k+1}^{-}} \right]$$

$$\hat{H}_{k+1} = \left[1 \quad (\hat{C}_1)_{k+1} \quad -2.(\hat{a}_1)_{k+1}^{-}.(\hat{\sigma}_1)_{k+1}^{-}.(\hat{C}_1)_{k+1}.(\hat{D}_1)_{k+1} \right.$$
$$2.(\hat{a}_1)_{k+1}^{-}.(\hat{\sigma}_1)_{k+1}^{-}.(\hat{C}_1)_{k+1}.\left\{x_{k+1} - (\hat{x}_{01})_{k+1}^{-}\right\} \quad 2.(\hat{a}_1)_{k+1}^{-}.(\hat{\sigma}_1)_{k+1}^{-}.(\hat{C}_1)_{k+1}.\left\{y_{k+1} - (\hat{y}_{01})_{k+1}^{-}\right\}$$
$$\cdots \quad (\hat{C}_N)_{k+1} \quad -2.(\hat{a}_N)_{k+1}^{-}.(\hat{\sigma}_N)_{k+1}^{-}.(\hat{C}_N)_{k+1}.(\hat{D}_N)_{k+1}$$
$$\left. 2.(\hat{a}_N)_{k+1}^{-}.(\hat{\sigma}_N)_{k+1}^{-}.(\hat{C}_N)_{k+1}.\left\{x_{k+1} - (\hat{x}_{0N})_{k+1}^{-}\right\} \quad 2.(\hat{a}_N)_{k+1}^{-}.(\hat{\sigma}_N)_{k+1}^{-}.(\hat{C}_N)_{k+1}.\left\{y_{k+1} - (\hat{y}_{0N})_{k+1}^{-}\right\} \right]$$

$$\textit{where } (\hat{C}_i)_{k+1} = \exp\left\{-(\hat{\sigma}_i)_{k+1}^{-2}(\hat{D}_i)_{k+1}\right\}, (\hat{D}_i)_{k+1} = \left\{(x_{k+1} - (\hat{x}_{0i})_{k+1}^{-})^2 + (y_{k+1} - (\hat{y}_{0i})_{k+1}^{-})^2\right\}$$

$$K_{k+1} = P_{k+1}^{-}\hat{H}_{k+1}^{T}(\hat{H}_{k+1}P_{k+1}^{-}\hat{H}_{k+1}^{T} + R_{k+1})^{-1}$$
$$P_{k+1} = (I - K_{k+1}\hat{H}_{k+1})P_{k+1}^{-}$$
$$\hat{A}_{k+1} = \hat{A}_{k+1}^{-} + K_{k+1}\left\{\tilde{z}_{k+1} - \hat{h}_{k+1}\right\}$$
$$= \hat{A}_{k+1}^{-} + K_{k+1}\left\{\tilde{z}_{k+1} - \hat{b}_{k+1}^{-} - \sum_{i=1}^{N}(\hat{a}_i)_{k+1}^{-}\exp\left[-\left((\hat{\sigma}_i)_{k+1}^{-}\right)^2\left\{(x_{k+1} - (\hat{x}_{0i})_{k+1}^{-})^2 + (y_{k+1} - (\hat{y}_{0i})_{k+1}^{-})^2\right\}\right]\right\}$$

(4.17)

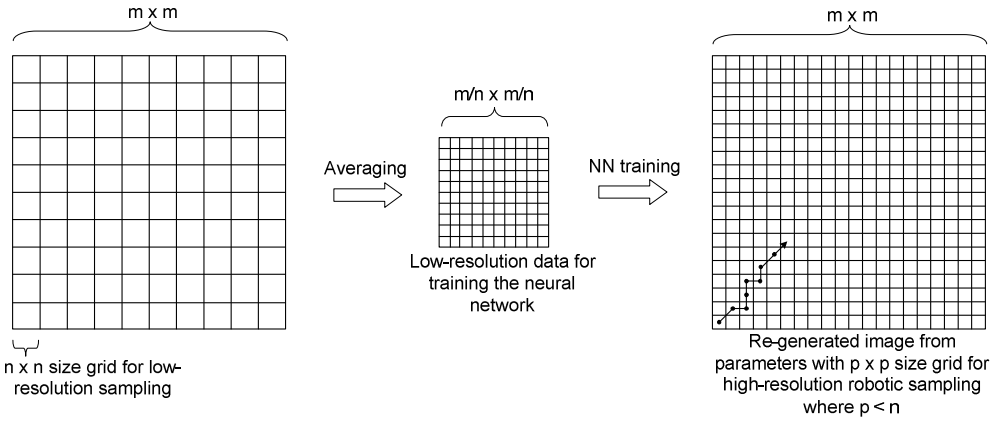


Figure 4.7 Change in spatial resolutions for multi-scale sampling

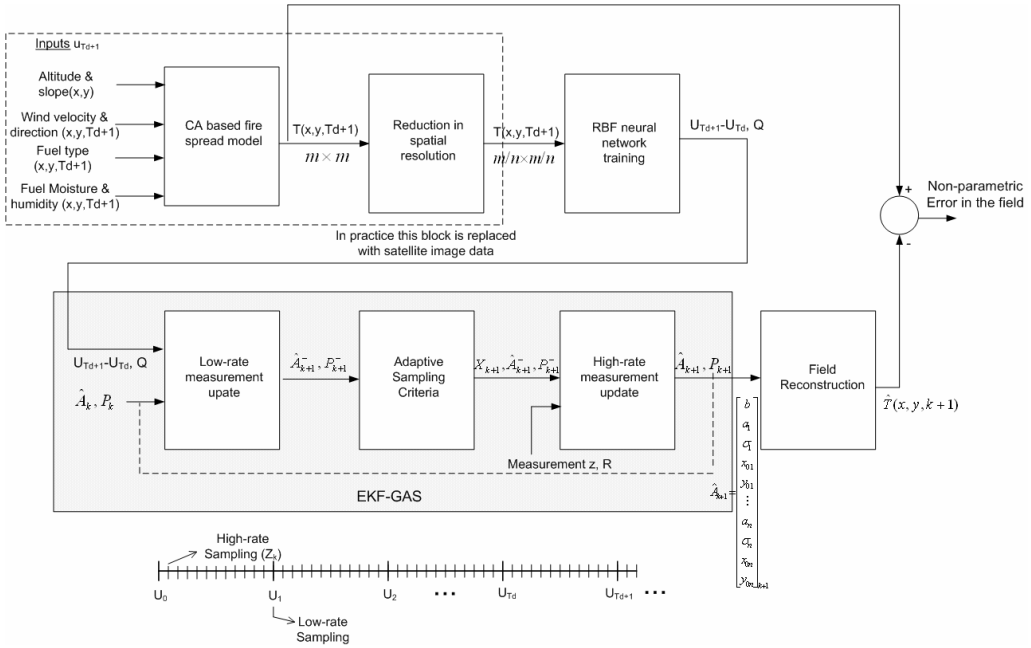


Figure 4.8 Block diagram for temporal field model identification and parameter estimation

4.4 Potential field to aid navigation through fire field using mobile agents

The estimated fire field intensity distribution can be used as a repulsive potential to keep the fire fighters away from dangerous areas in the field and show them safe paths towards important destinations. Potential field methods create a vector field representing a navigational path based on a potential function. The sampled field

variable intensity can be used to plan collision-free paths around the fire "obstacles". Given a scalar potential function $U(X)$ where $X=(x,y)$, that depends on the rescuer position and the field intensity at that point, one can calculate forces governing the rescuer motion based on the gradient of the scalar potential field:

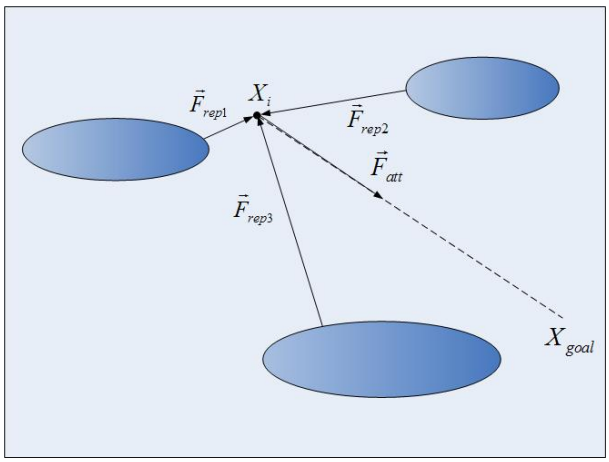


Figure 4.9 The attractive forces on point X_i towards the goal X_{goal} and repulsive forces from the obstacles

$$\vec{F}(X) = -\vec{\nabla}U(X) \ . \tag{4.18}$$

The following forces can be considered in the potential field:

- Attractive forces towards goals:

$$\begin{aligned} &\vec{F}_{att}(X) = -\vec{\nabla}U_{att}(X) = -\xi\rho_{goal}(X).\vec{\nabla}\rho_{goal}(X) \\ &= \xi(X - X_{goal}) \end{aligned} \tag{4.19}$$

- Repulsive forces from obstacles which are fire ellipses

$$U_{rep}(X) = \sum_{k=1}^{n} U_k(X), \quad U_k(X) = \lambda \frac{I_k}{\rho_k(X)^2} \tag{4.20}$$

$$\vec{F}_{rep}(X) = \sum_{k=1}^{n} \vec{F}_k(X), \quad \vec{F}_k(X) = -\vec{\nabla}U_k(X), \tag{4.21}$$

where λ is the positive scaling factor, $\rho(X)$ is the Euclidean distance from X to the center of ellipse, r is the number of elliptical components close to current location and I_k is the fire intensity at the point on ellipse at shortest distance from X. The trajectory can then be updated using a depth-first planning algorithm, which constructs a path as the product of successive segments starting at the initial configuration X_i:

$$\vec{X}_{i+1} = \vec{X}_i + \delta_i \vec{F}_p, \tag{4.22}$$

where X_i and X_{i+1} be the origin and end extremities of the i^{th} segment in the path.

In summary, this Chapter discusses the application of the Adaptive Sampling algorithm to the mapping of forest fires. In addition to the basic approach discussed in Chapter 3, we also offer a complete framework encapsulated in the Algorithm EKF-NN-GAS, which combines sensor measurements from different scales, rates and accuracies to map the time-varying spread of forest fires

Finally, we need to briefly discuss issues related to the convergence of the proposed algorithms. Since we are using the Extended Kalman Filter, any estimation scheme that utilizes it must be initialized sufficiently close to the actual field. Overhead satellite imagery of the field provides a reasonable initial estimate, but the absolute algorithm convergence cannot be guaranteed, as widely discussed in the literature [152]. In addition, the use of Heuristic Search methods in the algorithm can also lead to the presence of local minima . We can avoid such minima by restricting the search space so that we do not re-visit already sampled points, but such heuristics may not always work for time-varying fields. While studying convergence conditions for our algorithms was

beyond the scope of our thesis, simulation results presented in the next sections indicate that the algorithms converge in numerous instances. The algorithms of Chapters 3 and 4 will be simulated and experimentally validated in subsequent Chapters of the thesis.

CHAPTER 5

SIMULATION RESULTS

This Chapter focuses on simulation results with Adaptive Sampling for parametric fields of increasing complexity, in order to validate the formulations presented in previous Chapters. Section 5.1 presents simulation results for linear-in-parameter field first without considering uncertainty in localization and then considering uncertainty in localization of the robots. Section 5.2 discusses simulations results for a single Gaussian field without considering uncertainty in localization and then considering uncertainty in localization of the robots. Section 5.3 presents the simulation results for parameterization of complex space-time fields using the neural network. Finally, section 5.4 discusses simulations of complex, stationary and time-varying spatio-temporal forest fire fields, and potential field based approach for path planning for fire fighters.

5.1 Linear parametric field

5.1.1 Kalman filter estimation for the linear-in-parameters field without considering uncertainty in localization

For the algorithm described in section 3.2.2, consider a stationary linear field, with two spatial variables (x,y), and three unknown parameters to be estimated. The field model is simply defined as:

$$Z = a + bx + cy\,. \tag{5.1}$$

For simplicity, assume that we wish to recover the parameters by sampling with a robot with no localization uncertainty. The process, measurement models, and update equations are given by:

- System Model: $A_{k+1} = A_k = \begin{bmatrix} a & b & c \end{bmatrix}^T$. (5.2)
- Measurement Model: $\tilde{z}_k = H_k A_k + v_k = \begin{bmatrix} 1 & x & y \end{bmatrix} A_k + + v_k$. (5.3)
- Model and measurement uncertainties:

  $Q = 0,\ v_k \sim N(0, R),\ x_0 \sim N(\bar{x}_0, P_0)$.

- Effect of System Dynamics: $\hat{A}_{k+1}^- = \hat{A}_k, P_{k+1}^- = P_k$. (5.4)
- Effect of Measurement:

$$\begin{aligned} K_{k+1} &= P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1} \\ P_{k+1} &= (I_{12} - K_{k+1} H_{k+1}) P_{k+1}^- \\ \hat{A}_{k+1} &= \hat{A}_{k+1}^- + K_{k+1} \left\{ \tilde{z}_{k+1} - H_{k+1} \hat{A}_{k+1}^- \right\} \end{aligned} \quad . \tag{5.5}$$

We simulate the sampling process in MATLAB assuming the following numerical values:

- Actual parameters - $\begin{bmatrix} a & b & c \end{bmatrix} = \begin{bmatrix} 10 & 0.2 & 0.7 \end{bmatrix}$
- Initial parameter estimates - $\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$
- Parameters to estimate - $A = \begin{bmatrix} a & b & c \end{bmatrix}$
- Initial Error covariances - $P = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}$
- Model error covariance - $Q = 0$
- Measurement error covariance - $R = 1$

- Convergence Criteria = $\left\| A - \hat{A}_{k+1} \right\| < 0.7$

The sampling area is divided into grid sizes of 5 and 10 units and we assume that only the center of the grid is sampled. For the greedy adaptive sampling heuristic, we pick a horizon size equal to 2, to limit the search for the next appropriate sampling location, as depicted in Figure 5.1.

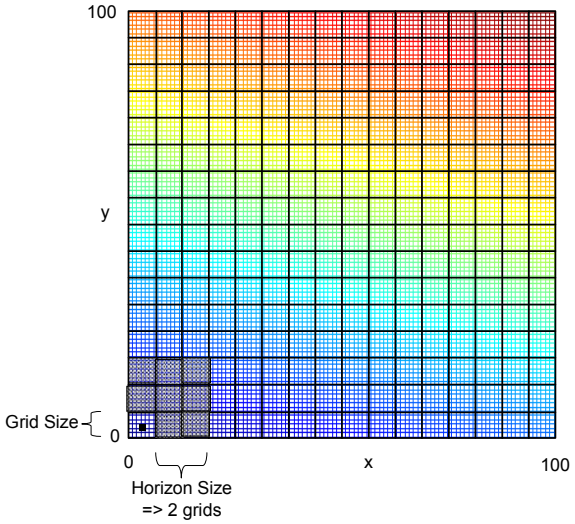


Figure 5.1 Grid size and horizon size are shown for sampling a linear parametric field

In this very simple case, there are three parameters to estimate and the AS algorithm looks for the next sampling location which minimizes the 2-norm of the error covariance matrix of these three parameters. If we wish to focus on minimizing the uncertainty of a specific parameter, instead of the 2-norm aggregate, the next sampling location would be the point which minimizes the error covariance of that particular parameter only. In our simulation, a grid size of 5 is chosen, and therefore a maximum horizon size of 19 will cover the entire field. Simulations results are plotted in figures 5.2-5.6 for a global search AS. These plots are organized to include a snapshot of the

first few samples (top left), many samples (top right), the plots of error covariance, the parameter estimate, and values of error in parameter estimates with respect to number of samples (bottom left), and the plots of error in parameter estimates and error covariances with respect to the distance travelled (bottom right).

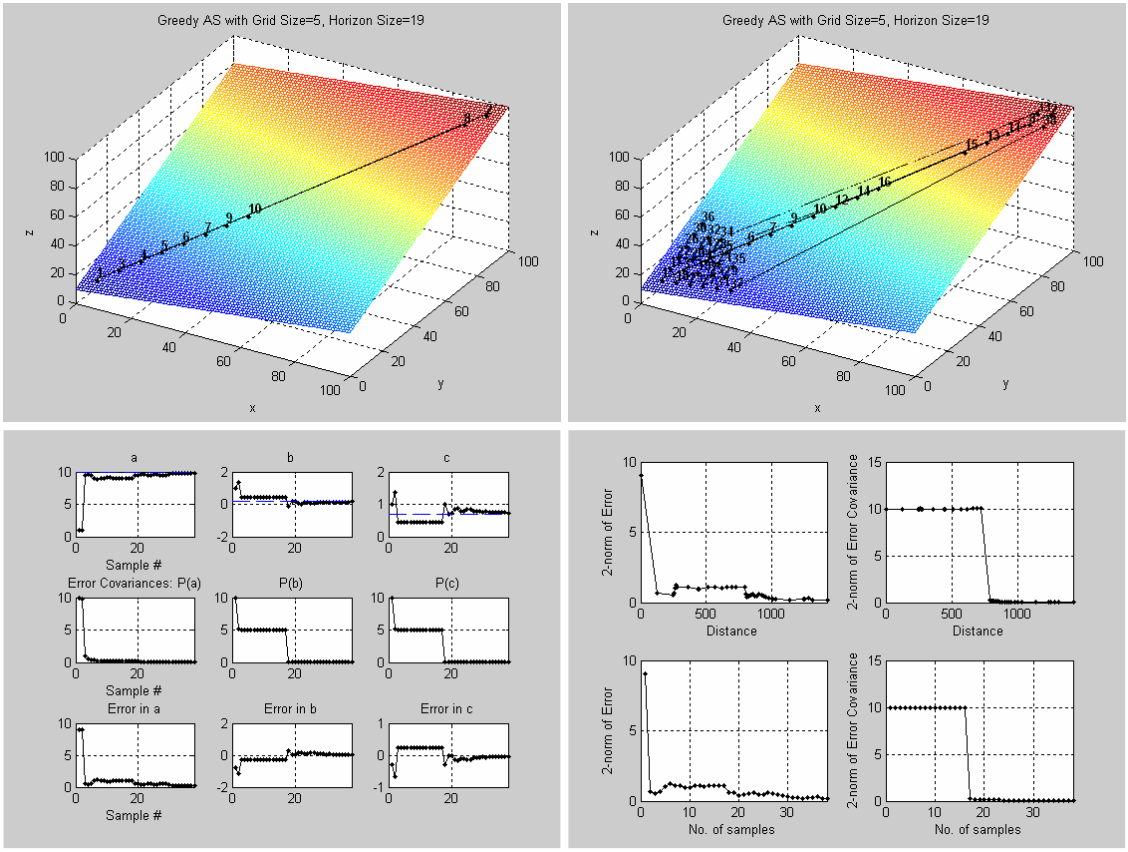


Figure 5.2 Sampling points when error covariance of a is minimized which takes 38 samples and a total distance of 1417 is covered

Figure 5.2 shows the results when the information measure of the algorithm is the error covariance of a estimates only. We observe that the maximum information about parameter a is achieved by first sampling the extreme (x, y) locations. The plots show that parameter a converges faster i.e. within 5 samples is this case. Results in Figure 5.3 indicate that the maximum information about parameter b is achieved by

sampling along x-axis, first starting at the extreme locations. Similarly, Figure 5.4 shows that the maximum information about parameter c is achieved by sampling along y-axis first starting at the extreme locations. And, in the last case, when the 2-norm of error covariance of all parameter estimates is used as information measure, the AS algorithm produces the combined effect to select appropriate locations. In all cases, we also computed the combined distance traveled by the sampling agent, displayed it in the captions, and summarized it in Table 5.1.

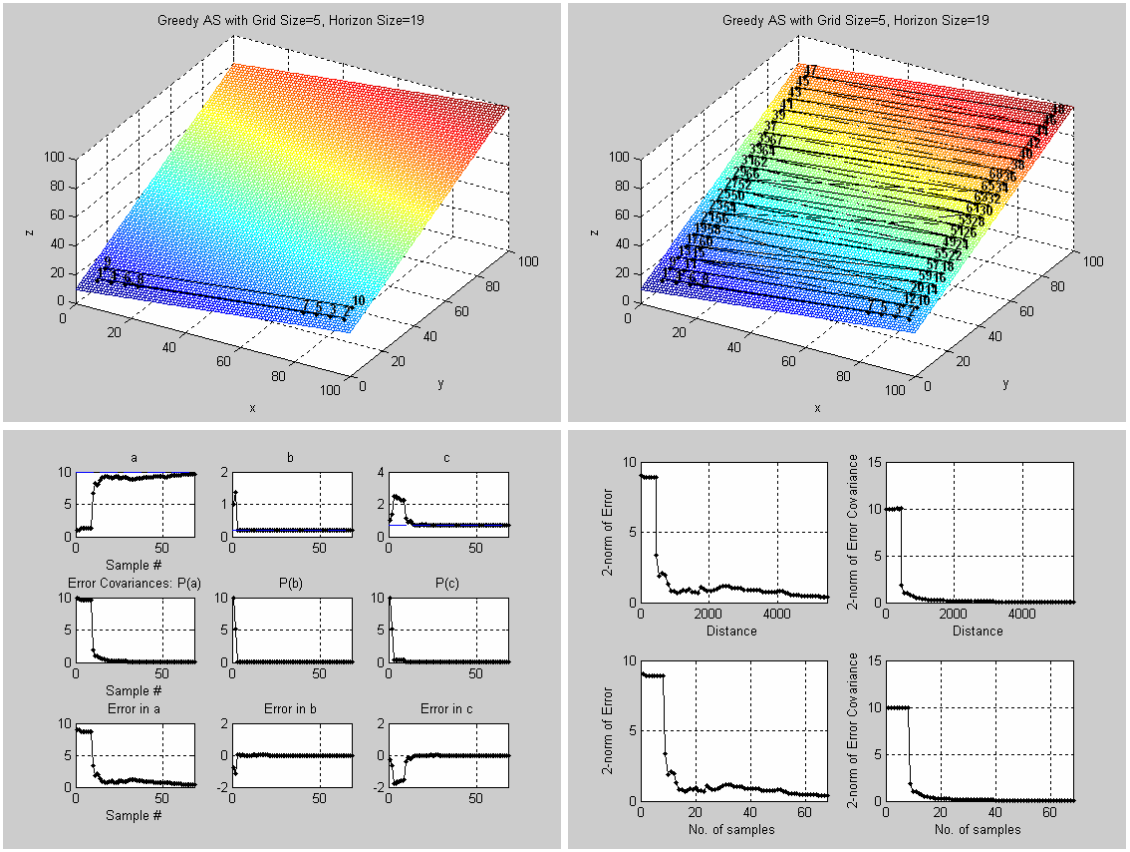


Figure 5.3 Sampling points when error covariance of b is minimized which takes 68 samples and a total distance of 5458 is covered

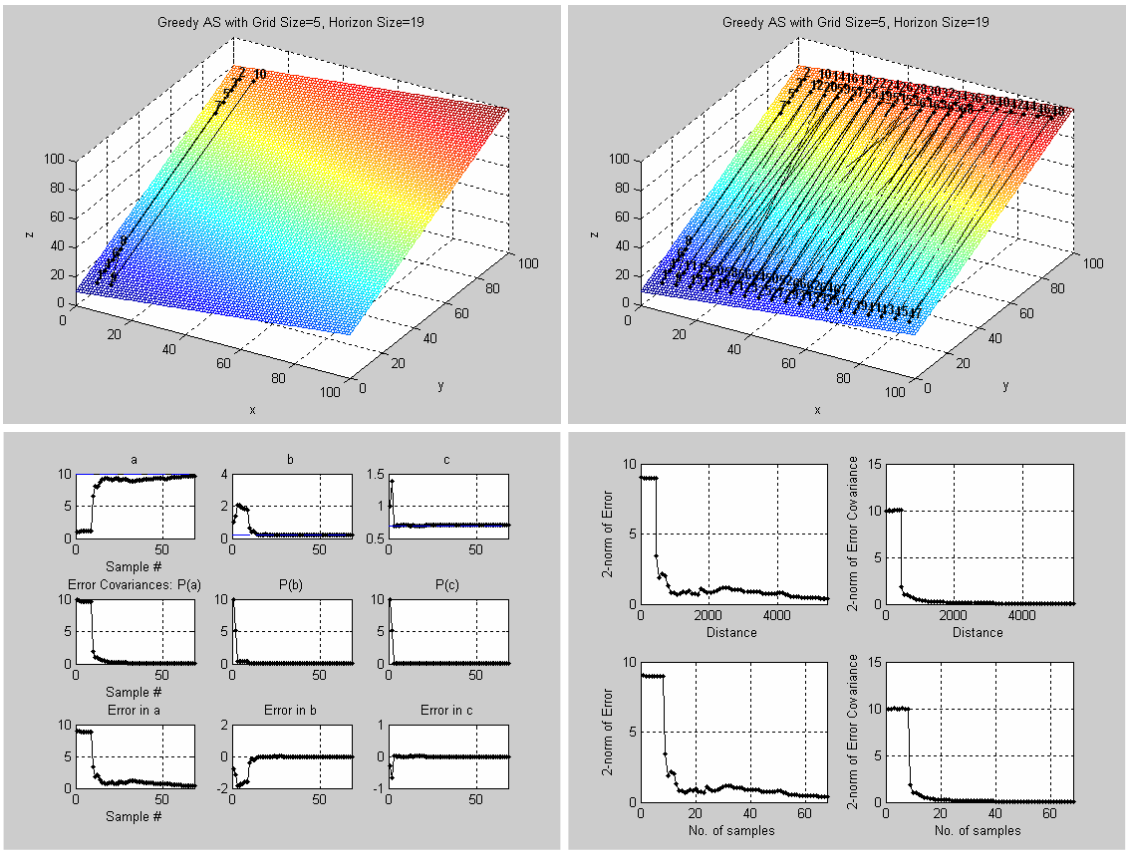


Figure 5.4 Sampling points when error covariance of c is minimized which takes 68 samples and a total distance of 5458 is covered

An interesting observation emerges from examining Table 5.1 – namely that minimizing the error covariance of parameters *b* and *c* requires more distance and number of samples to achieve the same convergence threshold. This is due to the fact that sampling along extreme x and y axes by minimizing uncertainty in *b* and *c* does not provide enough information about rest of the parameters. This can also be observed in Figure 5.3, where minimization of uncertainty in *b* causes *a* and *c* to converge very slowly compared to case in figure 5.2, where minimization of uncertainty in *a* causes *b* and *c* converge at significantly faster rate within 5 samples.

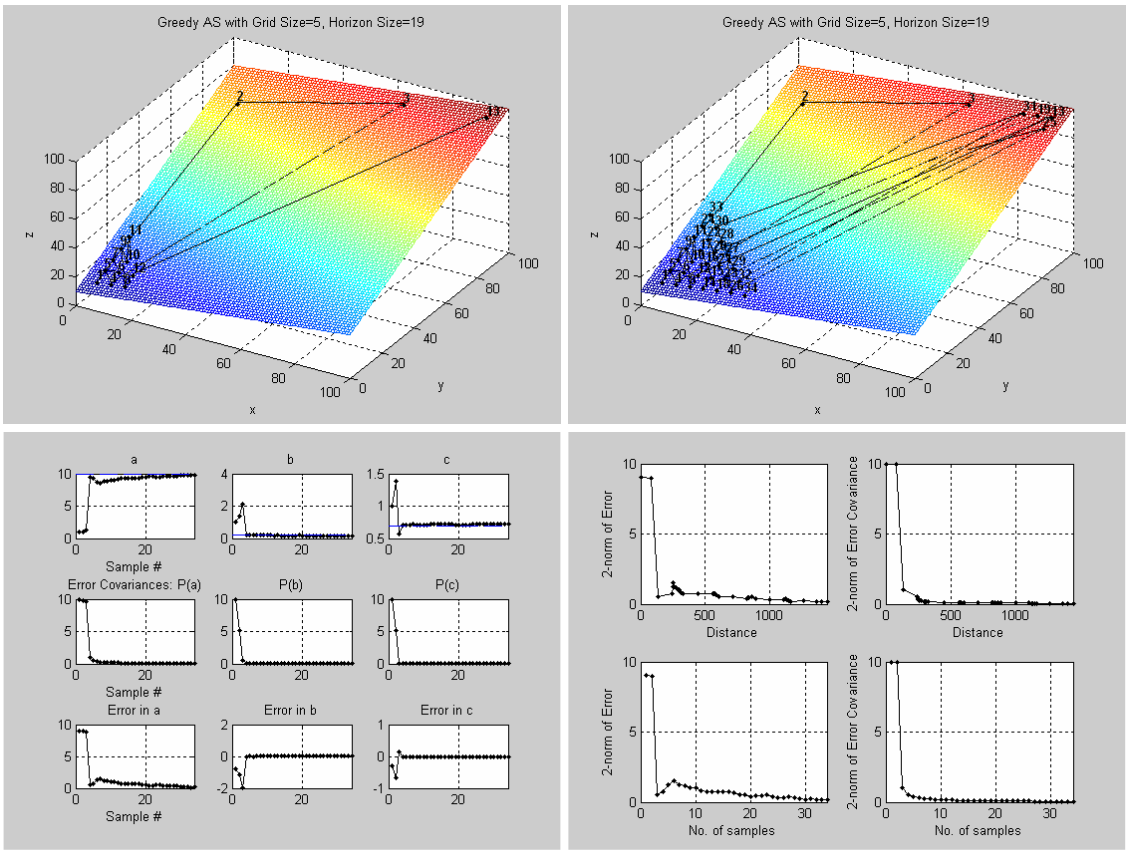
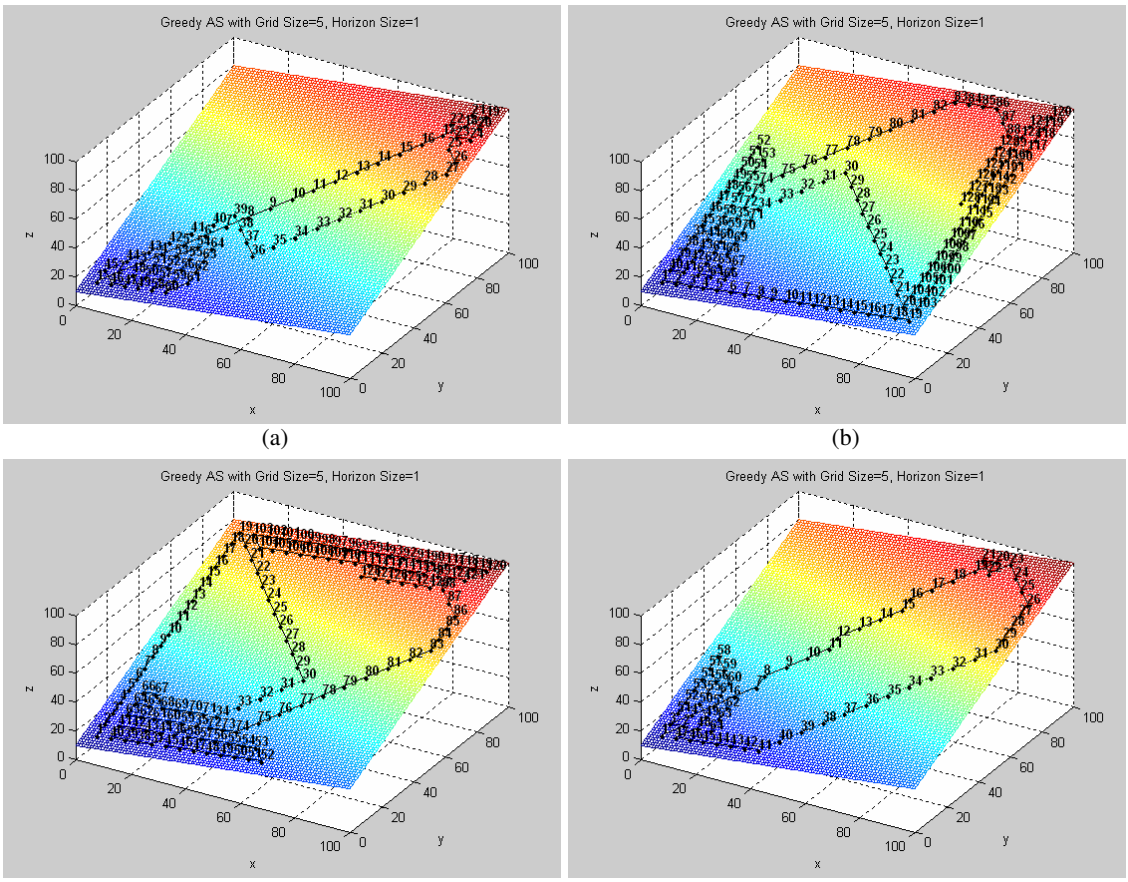


Figure 5.5 Sampling points when 2-norm of error covariance matrix P is minimized which takes 34 samples and a total distance of 1445 is covered

Table 5.1 Comparison of GAS and AS for number of samples and distance when error covariances of different parameters are minimized

	Minimizing Error Covariance	No. of Samples	Distance
Adaptive Sampling (Grid Size=5, Horizon Size=19)	$P(a)$	38	1417
	$P(b)$	68	5458
	$P(c)$	68	5458
	$\|P\|$	34	1445
Greedy Adaptive Sampling (Grid Size=5, Horizon Size=1)	$P(a)$	64	412
	$P(b)$	128	720
	$P(c)$	128	720
	$\|P\|$	64	396



(a) (b)

(c) (d)

Figure 5.6 For GAS (a) the error covariance of a is minimized, (b) the error covariance of b is minimized, (c) the error covariance of c is minimized, (d) the 2-norm error covariance matrix P is minimized

Next, we run the AS algorithm with a smaller horizon size of 2, as shown in the sampling sequences of Figure 5.6. And, in Figures 5.7 to 5.9, a comparison is shown for sampling with a grid size of 5 and 10 for raster scanning (RS), adaptive sampling AS (e.g. a horizon of 19), greedy adaptive sampling (GAS) (e.g. a horizon of 2). In Figure 5.7 it is apparent that for RS increasing the grid size reduces the number of samples required to meet the convergence criterion. The number of samples per row is smaller and hence more rows are scanned by covering more area on x-axis. Figure 5.9 shows a comparison of simulations with grid sizes 5 and 10 for GAS. A larger grid size of 10

gives better results because it covers more area with fewer samples, while the field does not have local variance because it is linear.

It is summarized in Table 5.2, and apparent from Figure 5.8, that changing the grid size does not reduce the number of samples and distance for AS because the next best sampling point is searched in the entire area. Moreover, in some cases increasing the grid size increases the number of samples and the distance to convergence because the data set becomes smaller.

In summary, these simple simulation results reveal that the field discretization grid, the information measure, and the horizon size greatly impacts the route taken by the robot to accomplish sampling missions. The sampling behavior can be intuitively understood, and shows that GAS requires a much smaller sampling distance than RS, while AS offers the least number of samples at the cost of the greatest travel distance. This observation is still valid for all simulation and experimental results presented in the Thesis.

Table 5.2 Comparison of RS, GAS and AS for number of samples and distance when different grid size and horizon size are considered for sampling a linear field

	Grid Size	Horizon Size	No of Samples	Distance
Raster Scan	5	-	113	560
	10	-	78	770
Adaptive Sampling	5	19	27	1124
	10	9	21	1205
Greedy Adaptive Sampling	5	1	62	386
	10	1	28	336

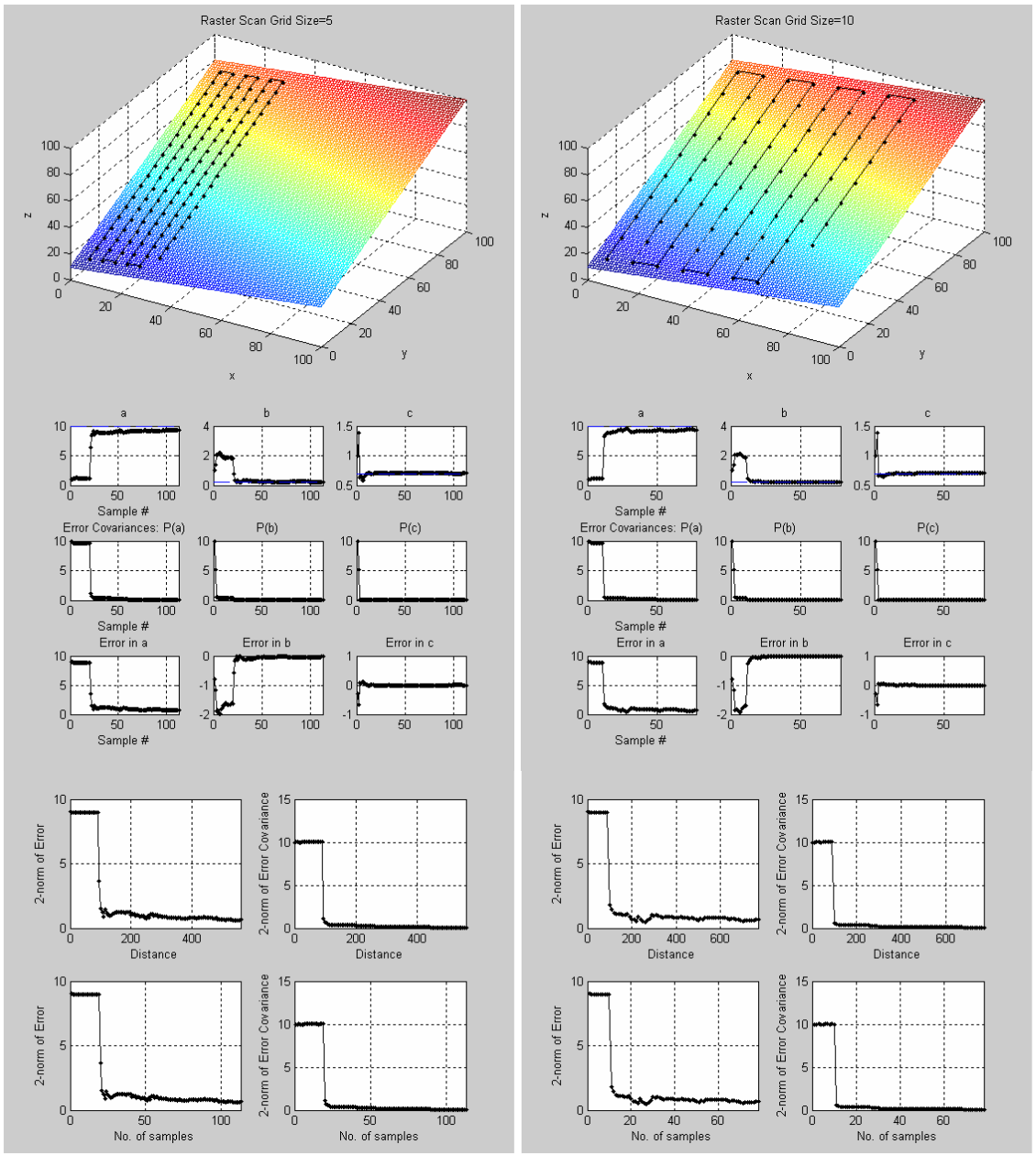
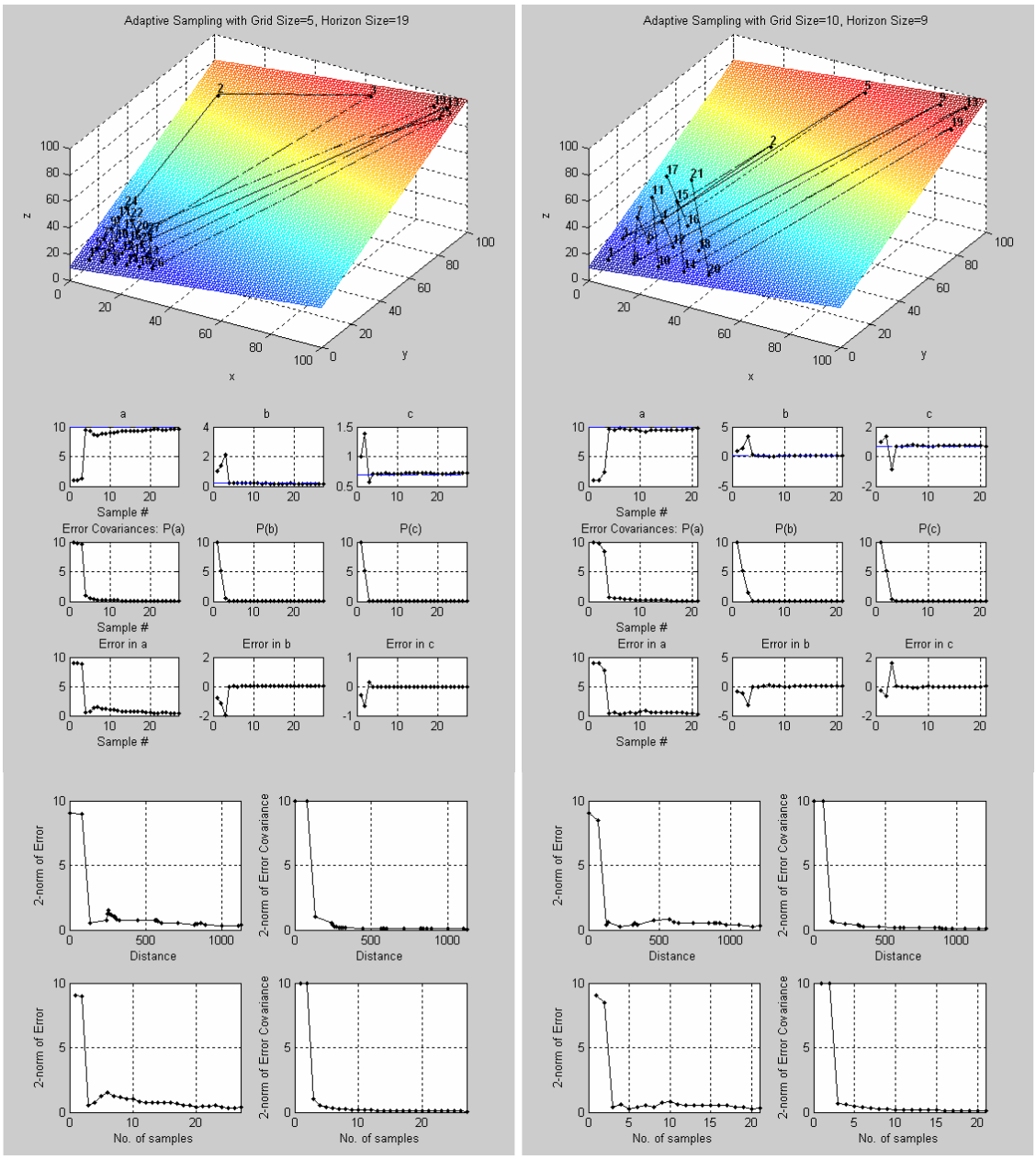
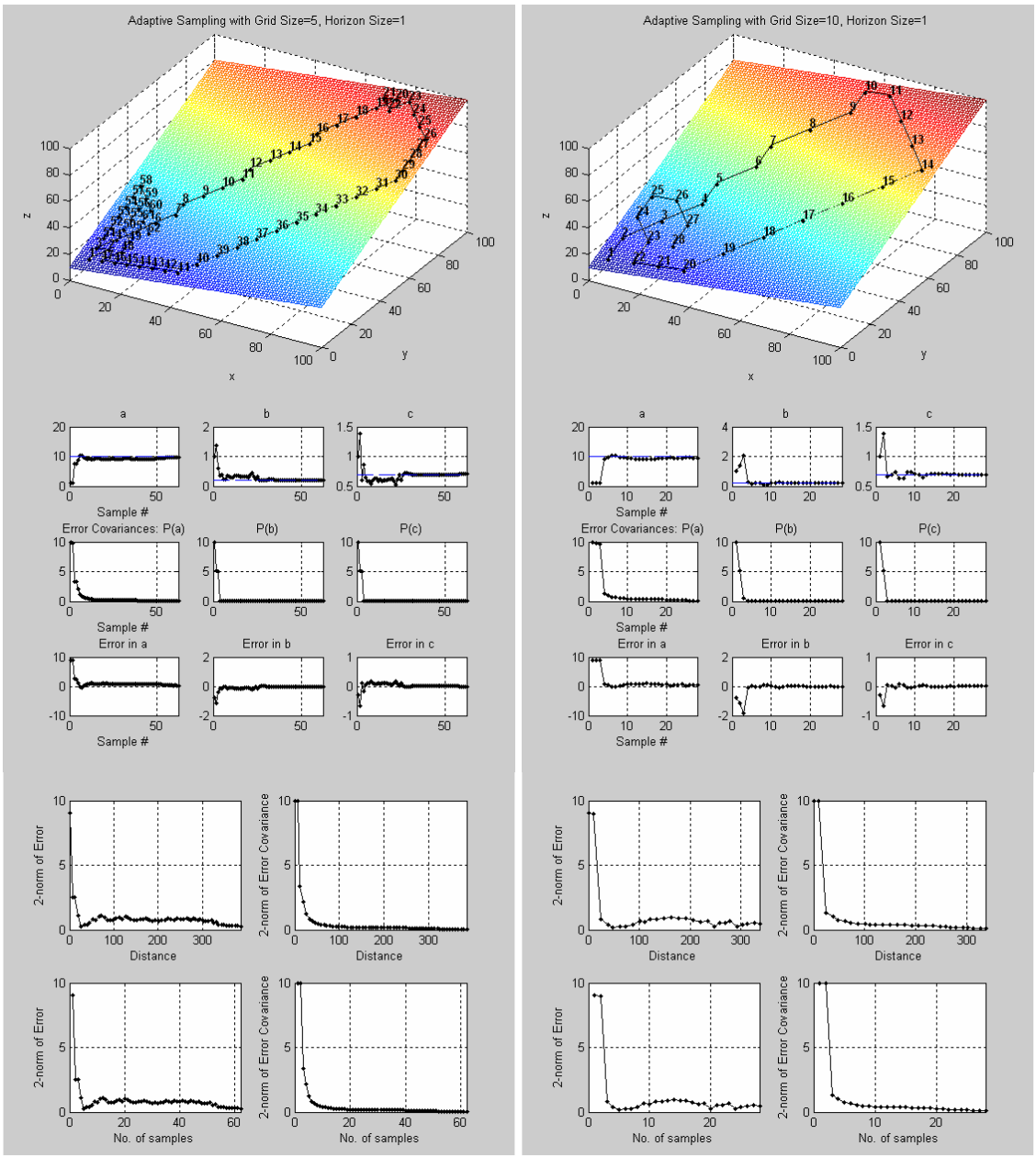


Figure 5.7 Simulation results for raster scan when grid size of 5 and 10 are considered



(a) (b)

Figure 5.8 Simulation results for adaptive sampling with: (a) grid size 5 and horizon size 19, (b) grid size 10 and horizon size 9



(a) (b)

Figure 5.9 Simulation results for greedy adaptive sampling with: (a) grid size 5 and horizon size 1, (b) grid size 10 and horizon size 1

5.2 Gaussian parametric field

5.2.1 Estimating parameters for a single Gaussian field

We now consider sampling of a stationary nonlinear field represented by a Gaussian with four unknown parameters, the center, amplitude, and variance $A = [a \quad \sigma \quad x_0 \quad y_0]$. This distribution is represented by the equation:

$$Z = a.\exp\left\{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}\right\} . \tag{5.6}$$

5.2.1.1 Minimization of uncertainty in single parameter

Four sampling routes are shown in figures 5.10-5.13 in which the sampling criterion seeks to minimize uncertainties in estimates of individual parameters. The numerical assumptions for simulation were:

- Actual parameters - $[a \quad \sigma \quad x_0 \quad y_0] = [100 \quad 20 \quad 40 \quad 60]$
- Initial parameters - $[a \quad \sigma \quad x_0 \quad y_0] = [105 \quad 15 \quad 45 \quad 65]$
- Model error covariance - $Q = 0$
- Measurement error covariance - $R = 1$

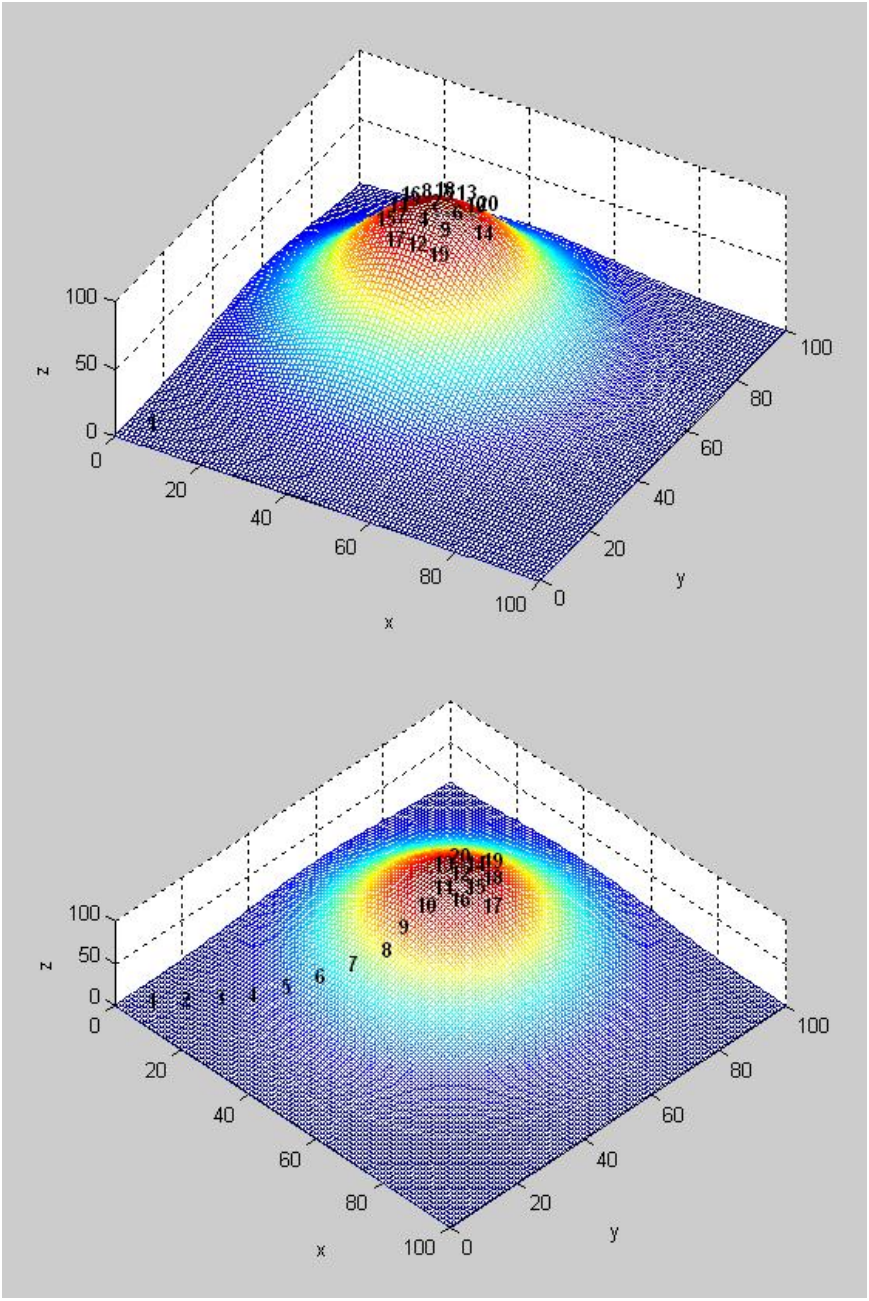


Figure 5.10 Sampling locations for minimizing the uncertainty in a for adaptive sampling when grid size 20 (top) and Greedy adaptive sampling with grid size 1 (bottom). Initial estimate for a is 105 and initial error covariance of 10

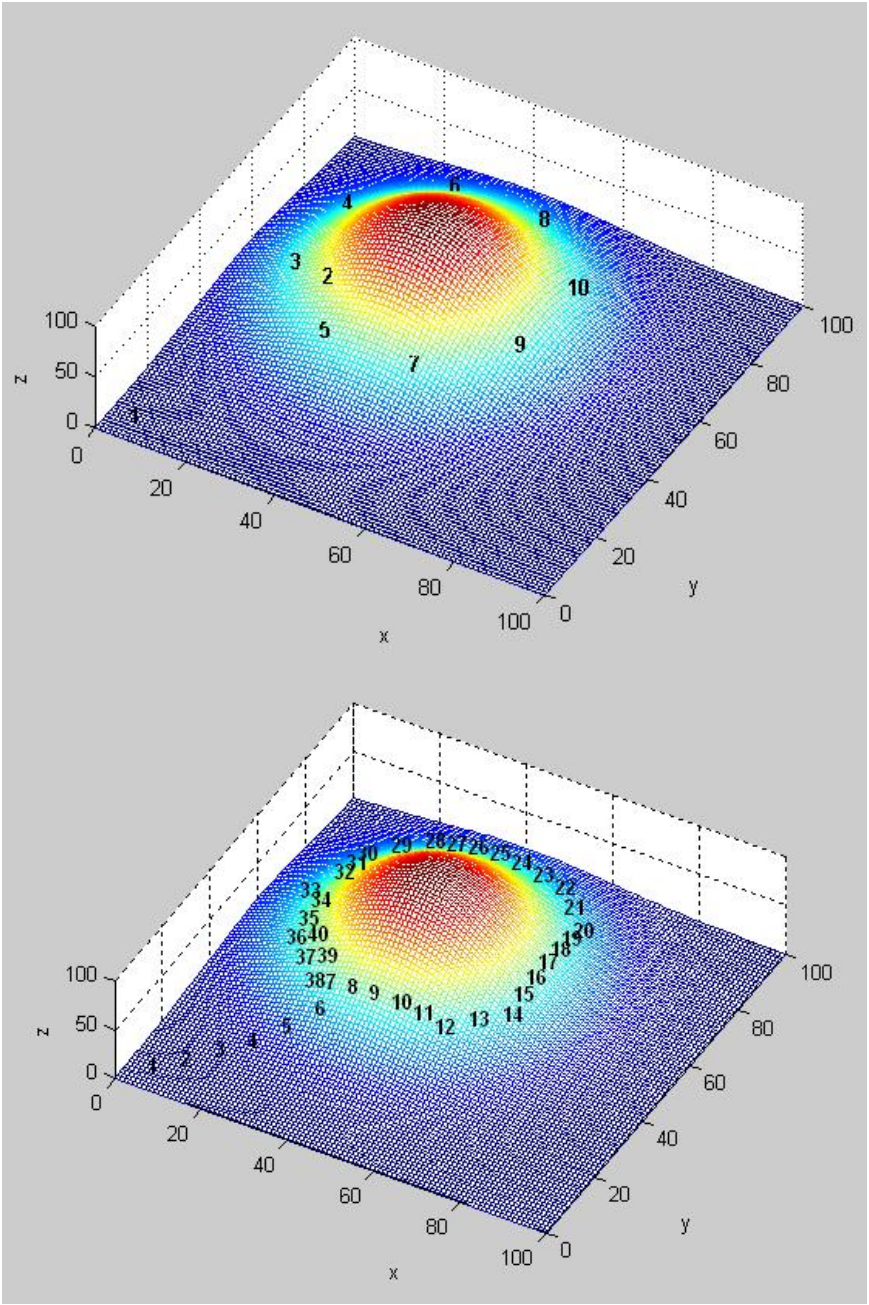


Figure 5.11 Sampling locations for minimizing the uncertainty in σ for adaptive sampling with grid size 20 (top), and Greedy adaptive sampling with grid size 1 (bottom) when initial estimate for σ is 15 and initial error covariance of 10

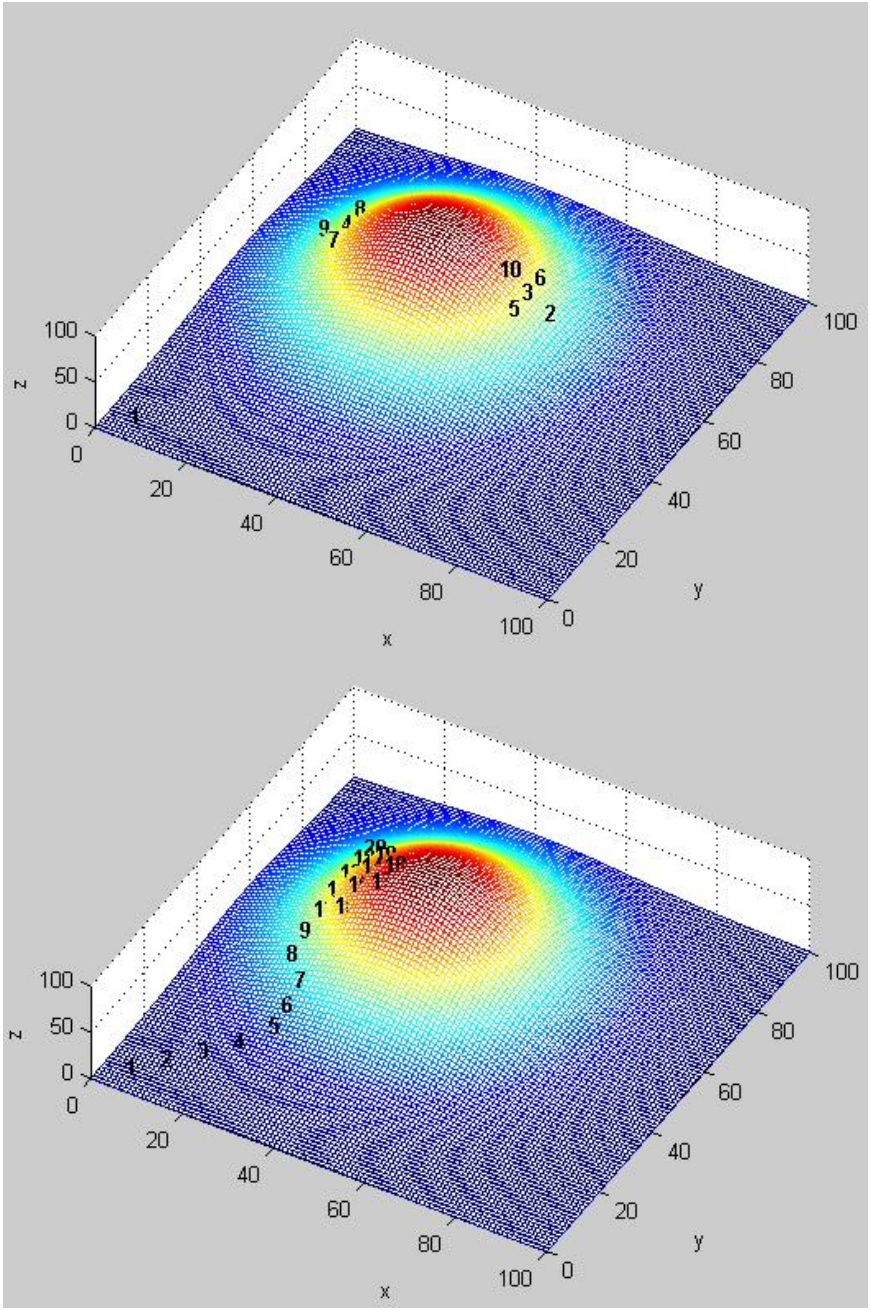


Figure 5.12 Sampling locations for minimizing the uncertainty in x_0 for adaptive sampling with grid size 20 (top) and Greedy adaptive sampling with grid size 1 (bottom) when initial estimate for x_0 is 45 and initial error covariance of 10

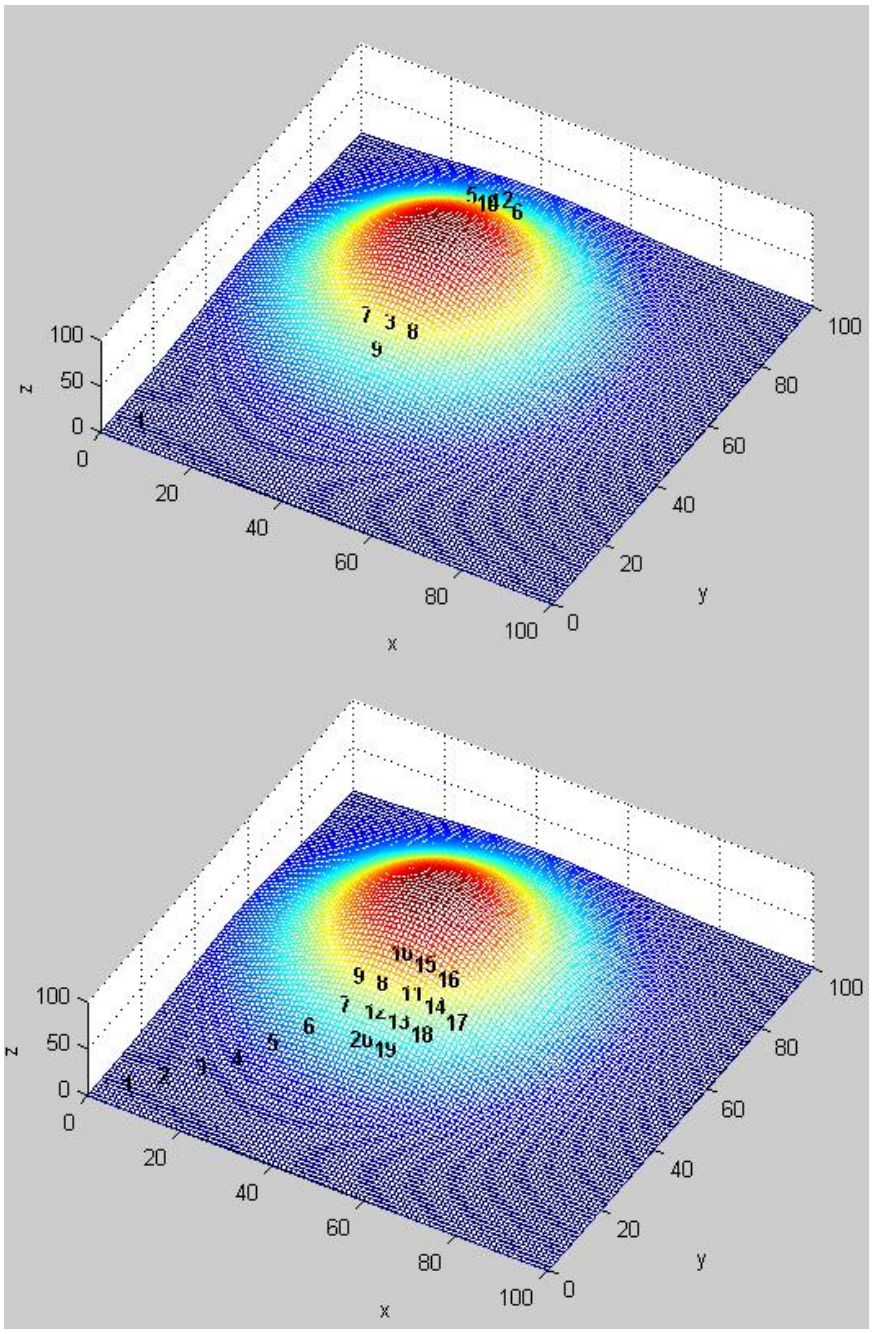


Figure 5.13 Sampling locations for minimizing the uncertainty in y_0 for adaptive sampling with grid size 20 (top) and Greedy adaptive sampling with grid size 1 (bottom) when initial estimate for y_0 is 65 and initial error covariance of 10

We can observe that minimizing the error covariance of a, while disregarding the uncertainty in other parameters, leads to samples being taken at the current estimate of (x_0, y_0) and in its neighborhood. Similarly, minimizing the uncertainty in σ, x_0, y_0 reslts in samples taken around $(x_0 \pm \sigma, y_0 \pm \sigma)$, $(x_0 \pm \sigma, y_0)$ and $(x_0, y_0 \pm \sigma)$ respectively. This idea is illustrated in figure 5.14.

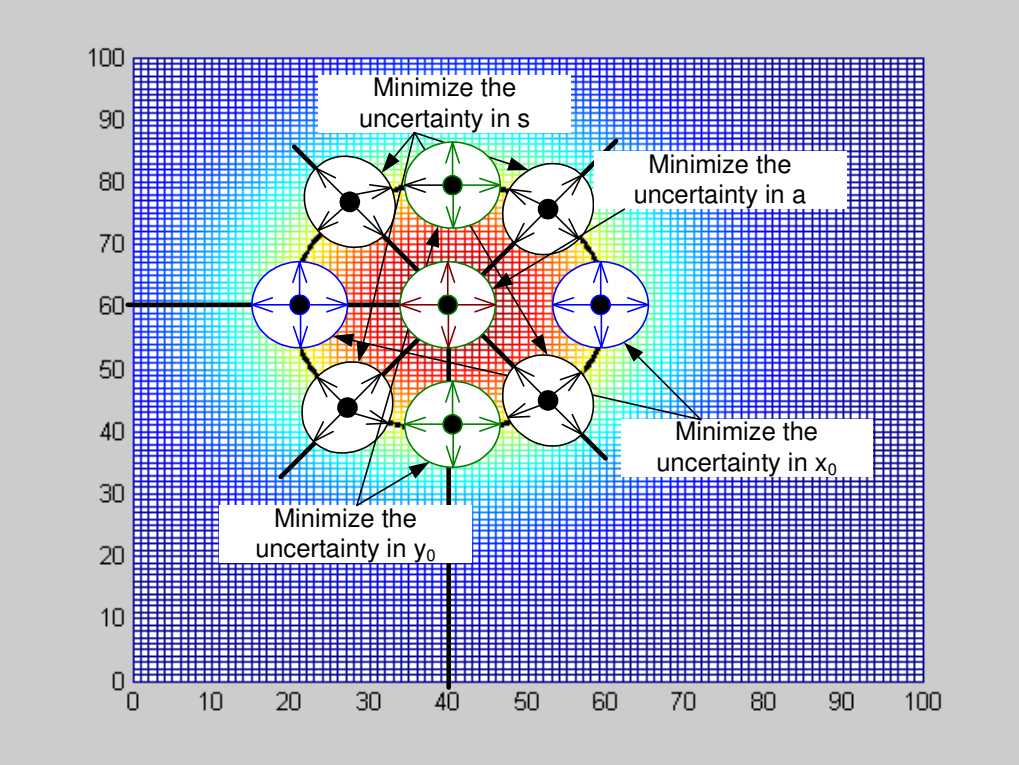


Figure 5.14 Applying minimum error covariance criterion for different parameters results in specific areas of the Gaussian to be sampled

5.2.1.2 Minimization of uncertainty in multiple parameters

It is apparent that a single scalar value cannot capture all the aspects of the error covariance matrix. For cases involving combined uncertainty of several parameters, information measures that can be considered include the 2-norm, inf-norm, trace or log-determinant of the error covariance matrix, as discussed in [49]. The 2-norm of error

covariance is a popular choice, however, most of these measures lead to parameter convergence as illustrated in Figures 5.15-5.19. The *Log-det(P)* measure can take negative values. *Inf-norm(P)* minimizes the uncertainty of the parameter which has the maximum uncertainty, and, as a result oscillations can be seen before the estimates converge to final values. These oscillations can also cause the EKF filter to diverge due to operating in non-linear regions. *Trace(P),* although it shows convergence, might not be appropriate because it doesn't consider the cross correlation terms of *P*. The following numerical assumptions were used for initial uncertainty:

- Initial parameter estimate vector - $\begin{bmatrix} 95 & 25 & 45 & 65 \end{bmatrix}$, and
- Initial error covariance - $P = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$.

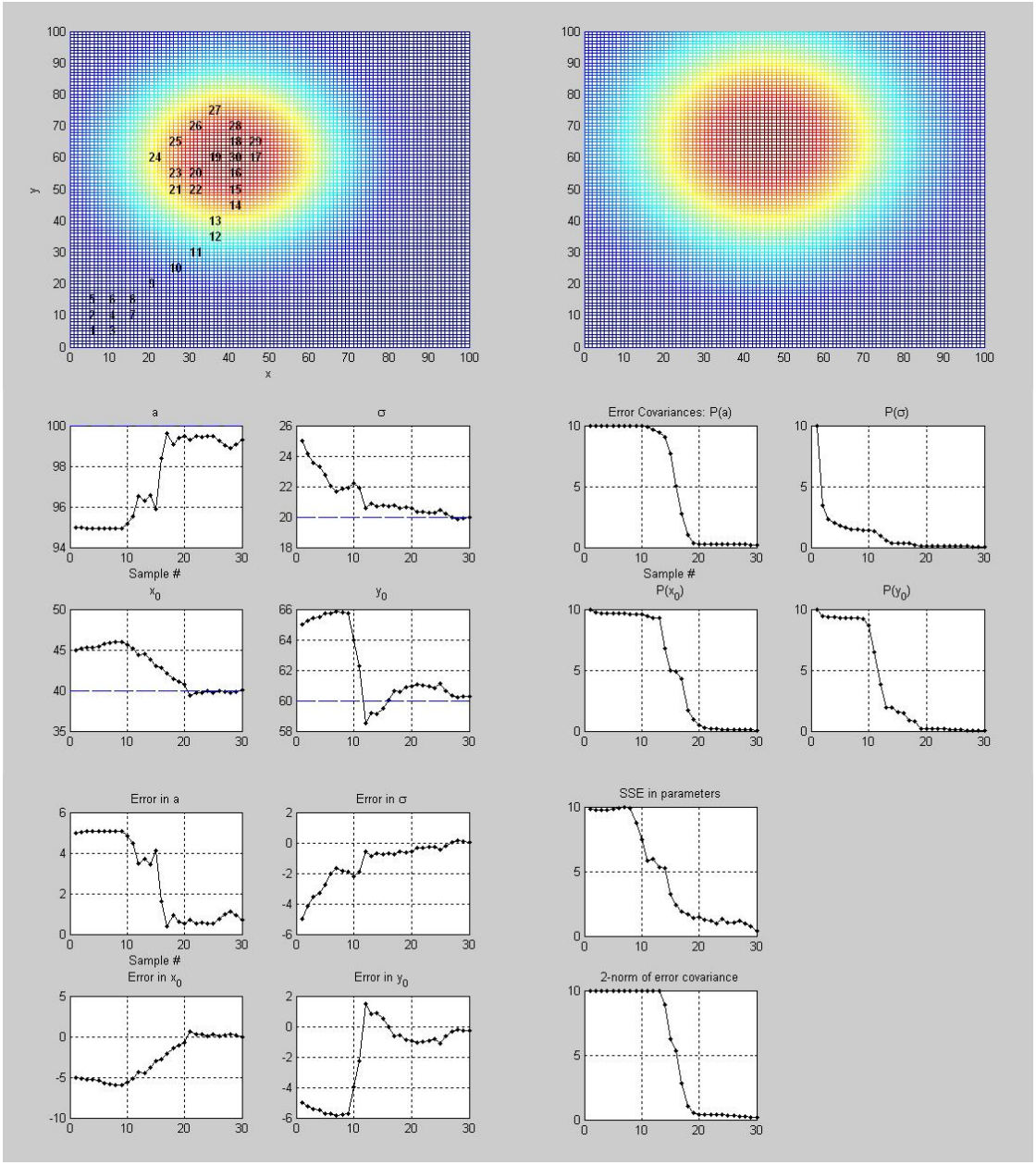


Figure 5.15 GAS (grid size 1) results for 20 samples with *2-norm* of error covariance criterion

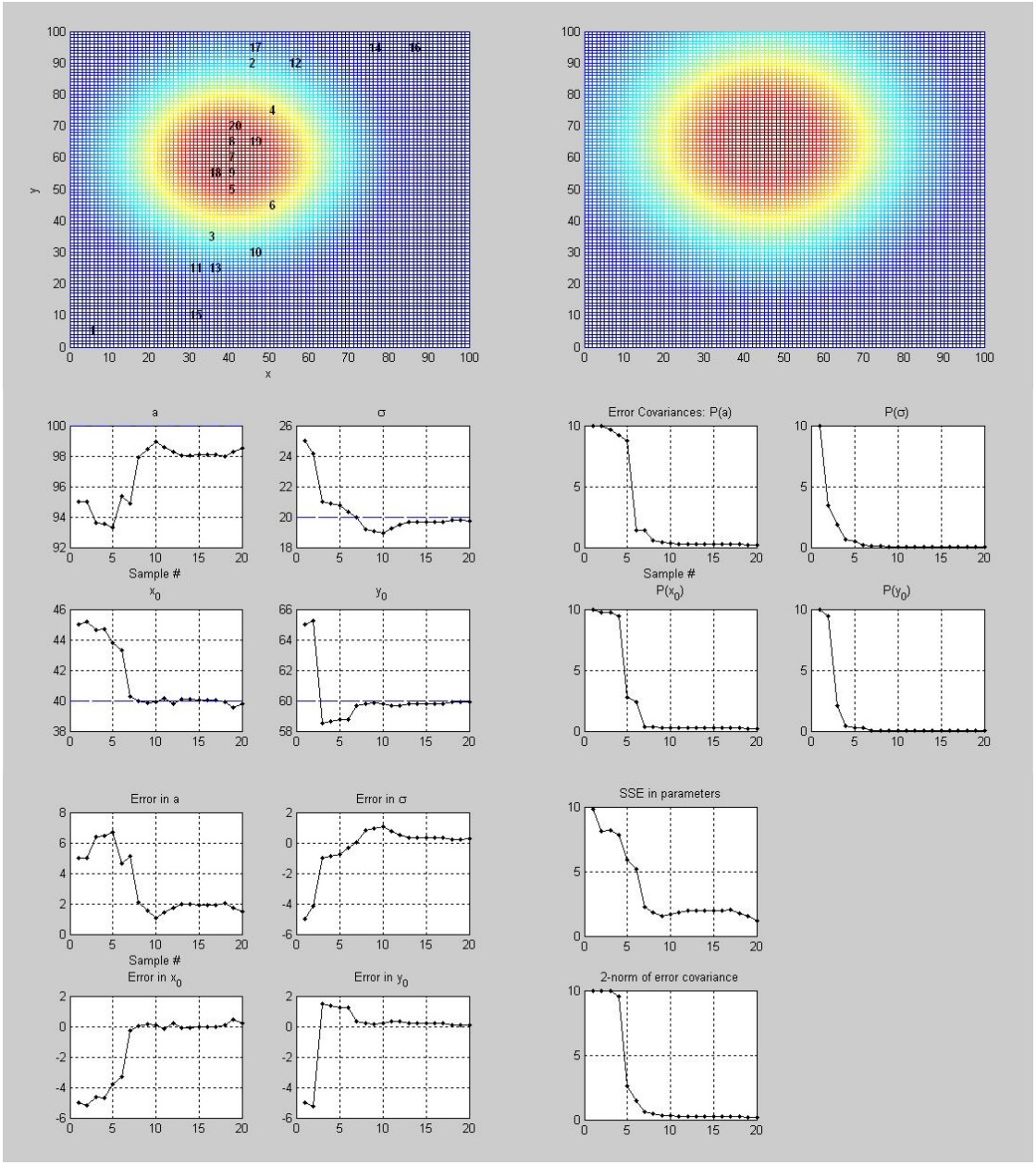


Figure 5.16 Adaptive sampling results for 20 samples with *2-norm* of error covariance criterion

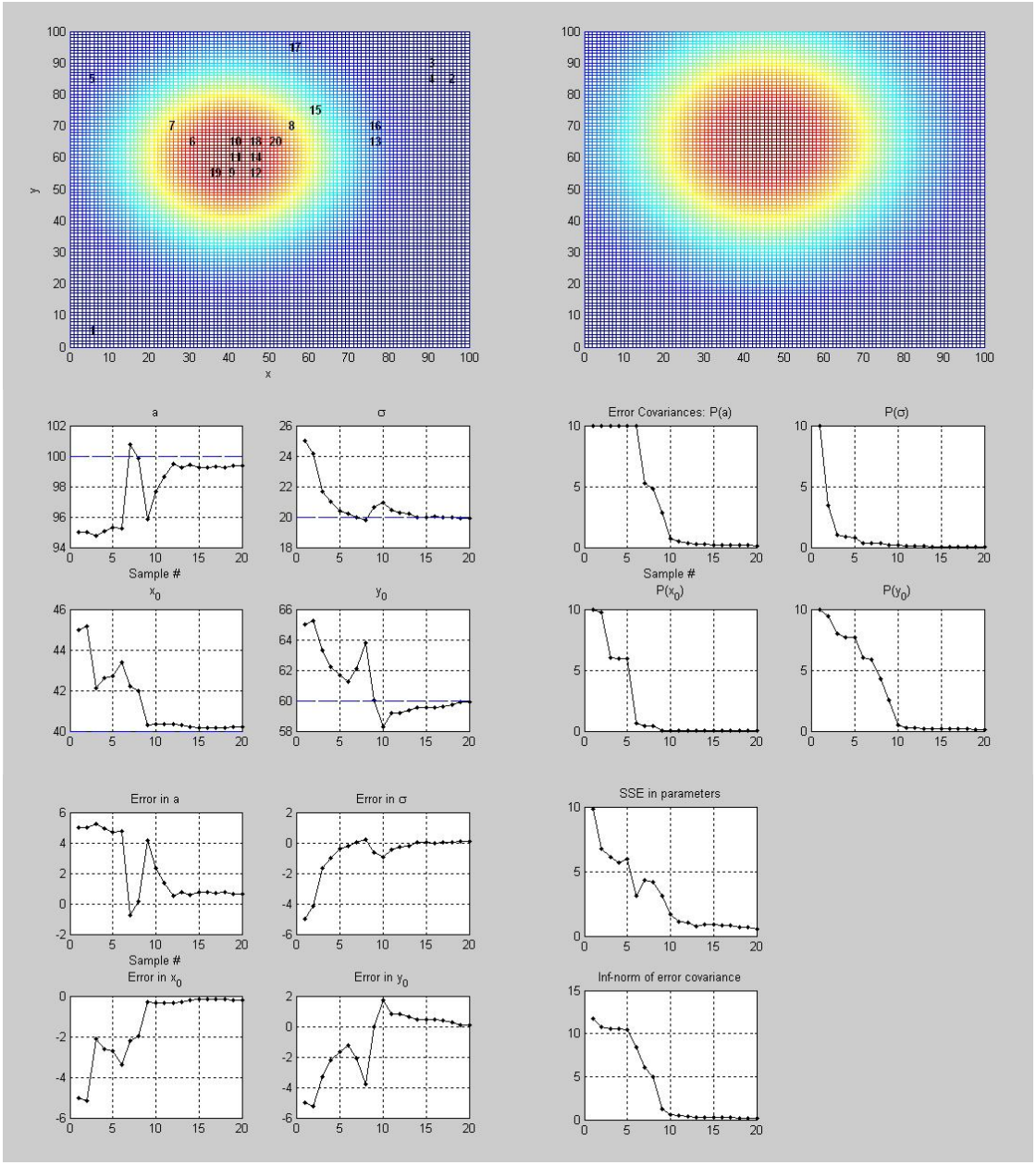


Figure 5.17 Adaptive sampling results for 20 samples with ∞-*norm* error covariance criterion

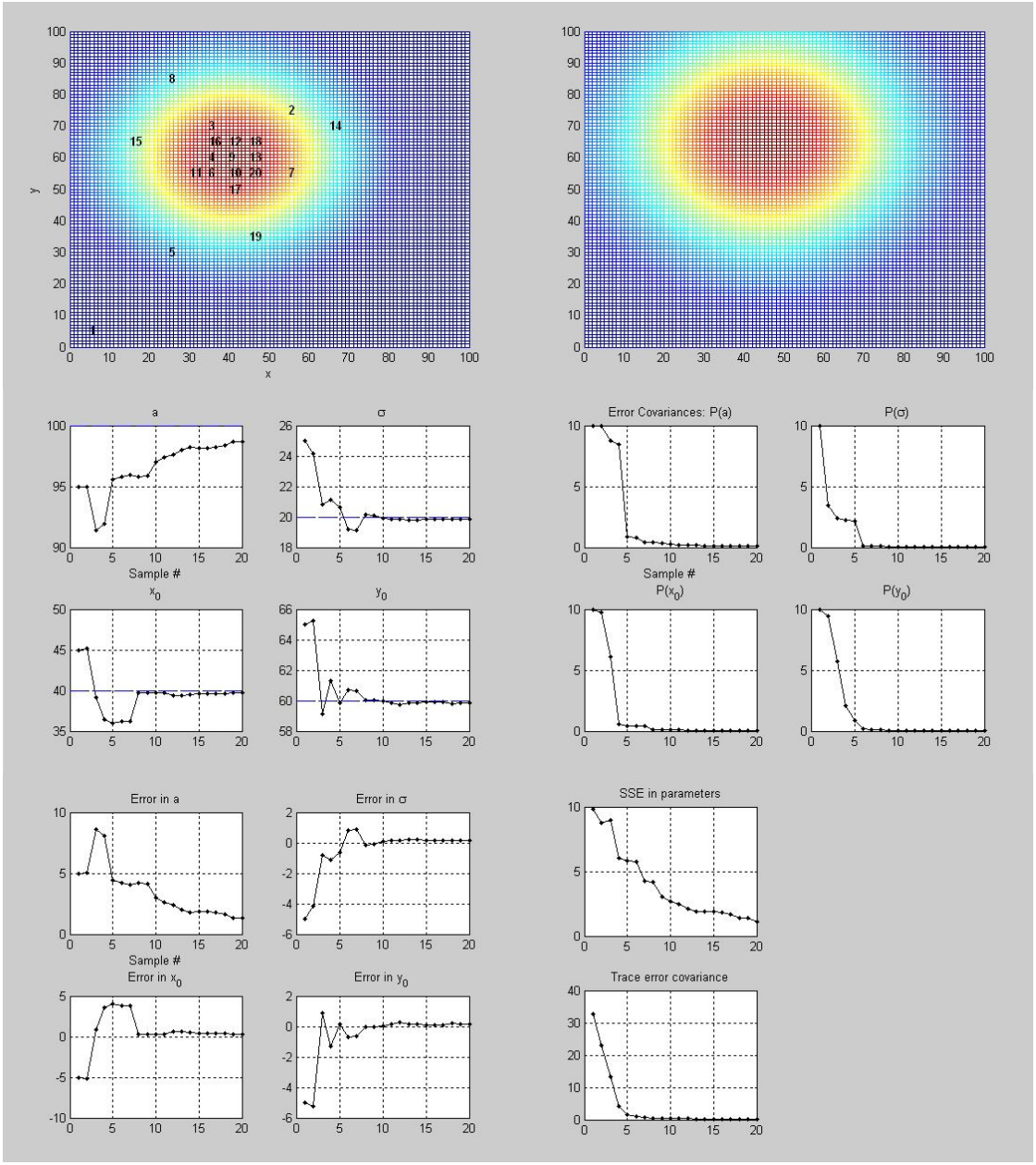


Figure 5.18 Adaptive sampling results for 20 samples with *Trace* of error covariance criterion

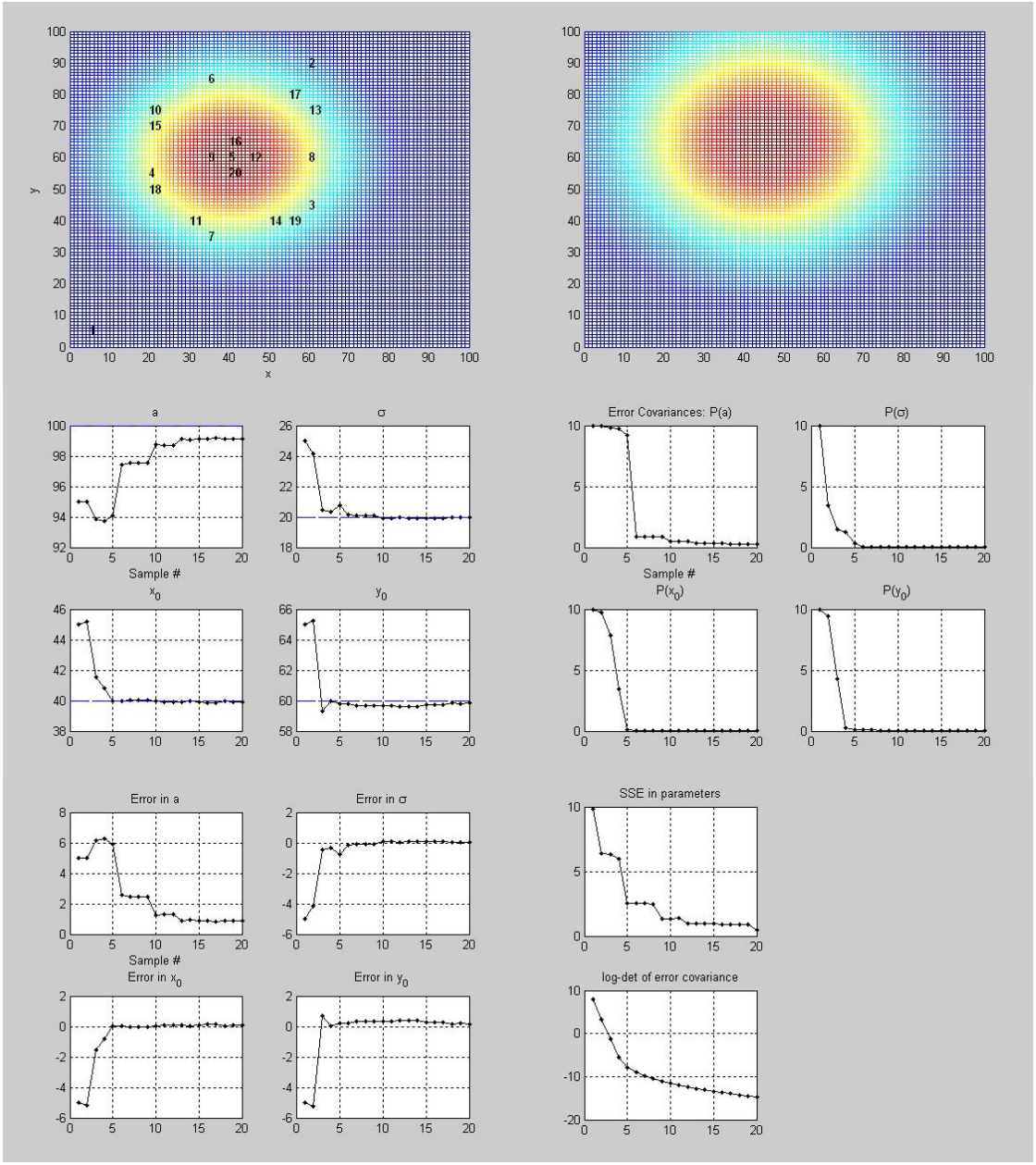


Figure 5.19 Adaptive sampling results for 20 samples with *log-det* of error covariance criterion

We conclude that the choice of covariance matrix norm will affect the performance of the sampling algorithm. In the absence of any prior information about the field, the 2-norm is an appropriate choice.

5.2.1.3 Effect of sampling grid size

We also investigated the effect of changing grid sizes for the sampling simulations. We used the following numerical assumptions:

- Actual parameters - $\begin{bmatrix} a & \sigma & x_0 & y_0 \end{bmatrix} = \begin{bmatrix} 100 & 50 & 40 & 60 \end{bmatrix}$,
- Initial parameter estimates - $\begin{bmatrix} 80 & 40 & 35 & 75 \end{bmatrix}$,
- Initial error covariance - $P = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{bmatrix}$,
- Model error covariance - $Q = 0$,
- Measurement error covariance - $R = 1$
- Convergence Criteria - $\left\| A - \hat{A}_{k+1} \right\| < 2.5$

Table 5.3 Comparison of RS, GAS and AS for number of samples and distance when different grid size and horizon size are considered for sampling a Gaussian field

	Grid Size	Horizon Size	No. of Samples	Distance
Raster Scan	5	-	205	1020
	10	-	96	950
Adaptive Sampling	5	19	24	1022
	10	9	34	1835
Greedy Adaptive Sampling	5	1	68	430
	10	1	44	496

Simulation results are shown in Figures 5.20 and 5.21 for raster scan with different grid sizes the plots for individual parameter estimates, error covariance of individual parameters, error in individual parameter estimates, error in individual error covariance, 2-norm of parameter estimates, and 2-norm of error covariance of

parameters. We notice that in both cases parameters a and x_0 converge slowly because the minimizing the uncertainty of a involve sampling near the Gaussian peak, and of x_0 by sampling along x-axis. However, raster scanning in this case is performed by sampling along y-axis, and therefore convergence will take a while. With a grid size of 10 instead of 5, fewer samples are required because in fewer samples most of the area will be covered. The results are tabulated in Table 5.3. As in the previous sections, GAS leads to a much smaller sampling distance, while AS requires the least number of samples.

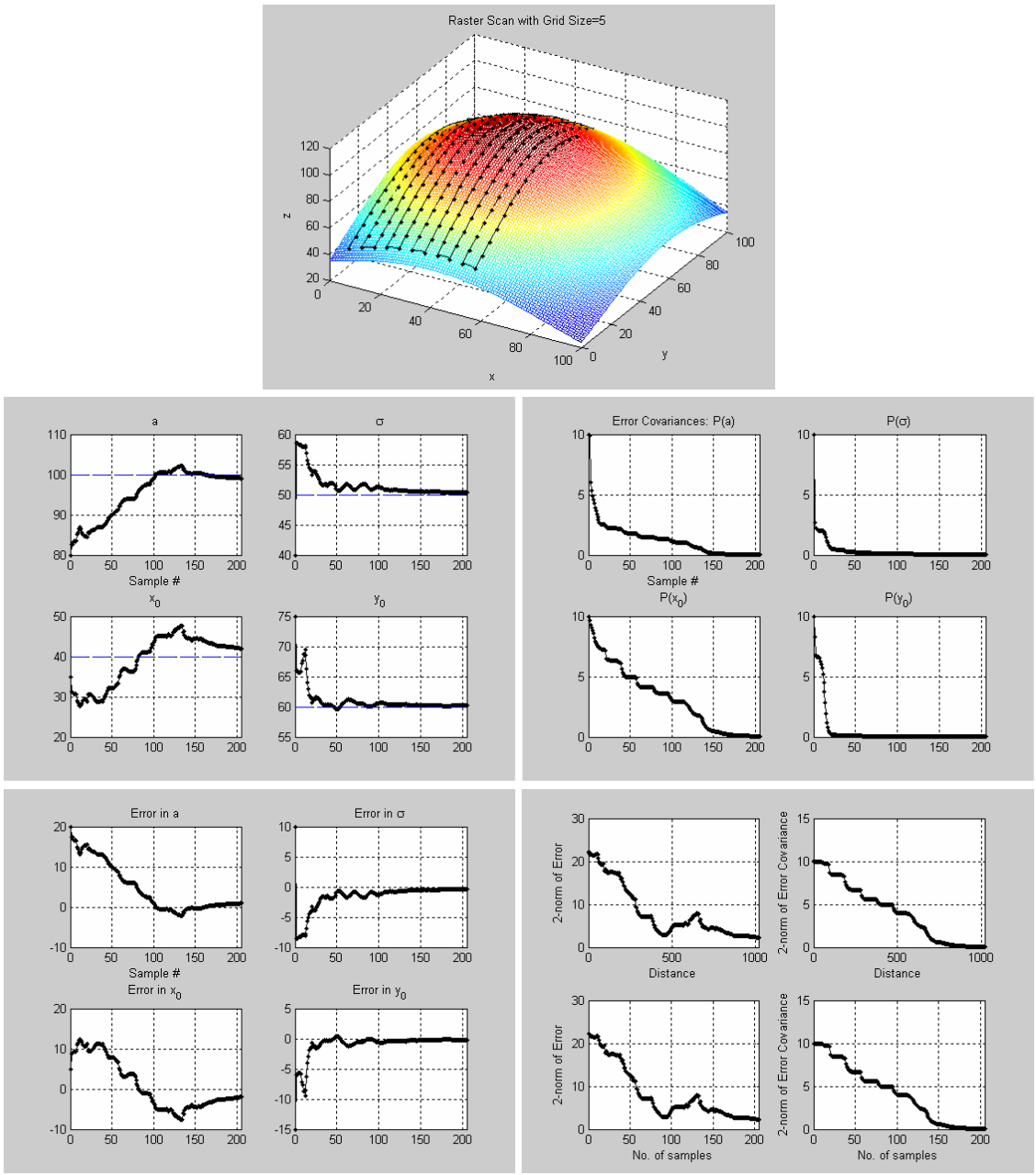


Figure 5.20 Raster scanning of a Gaussian field with grid size 5 which takes 205 samples and a distance of 1020

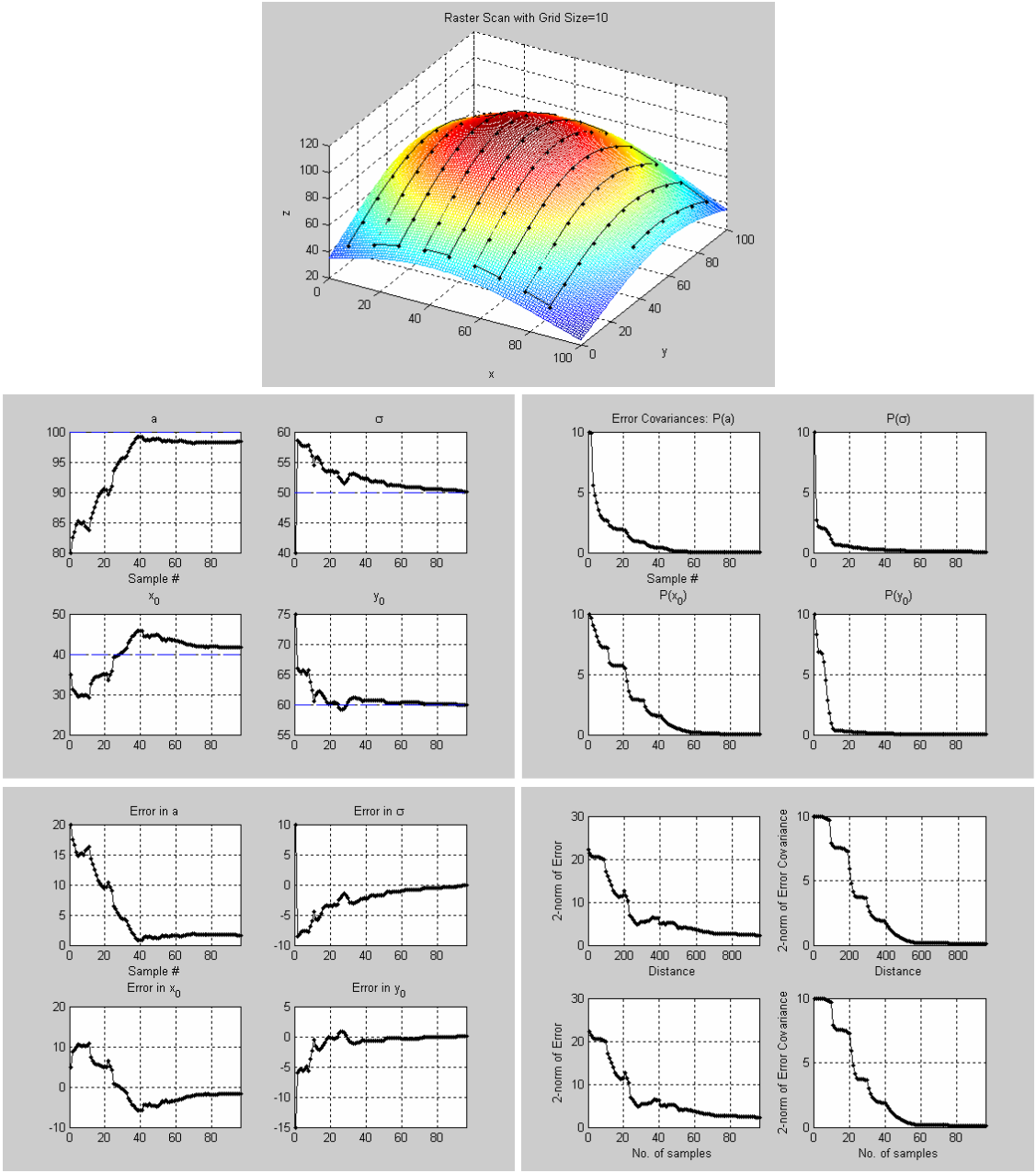


Figure 5.21 Raster scanning of a Gaussian field with grid size 10 which takes 96 samples and a distance of 950

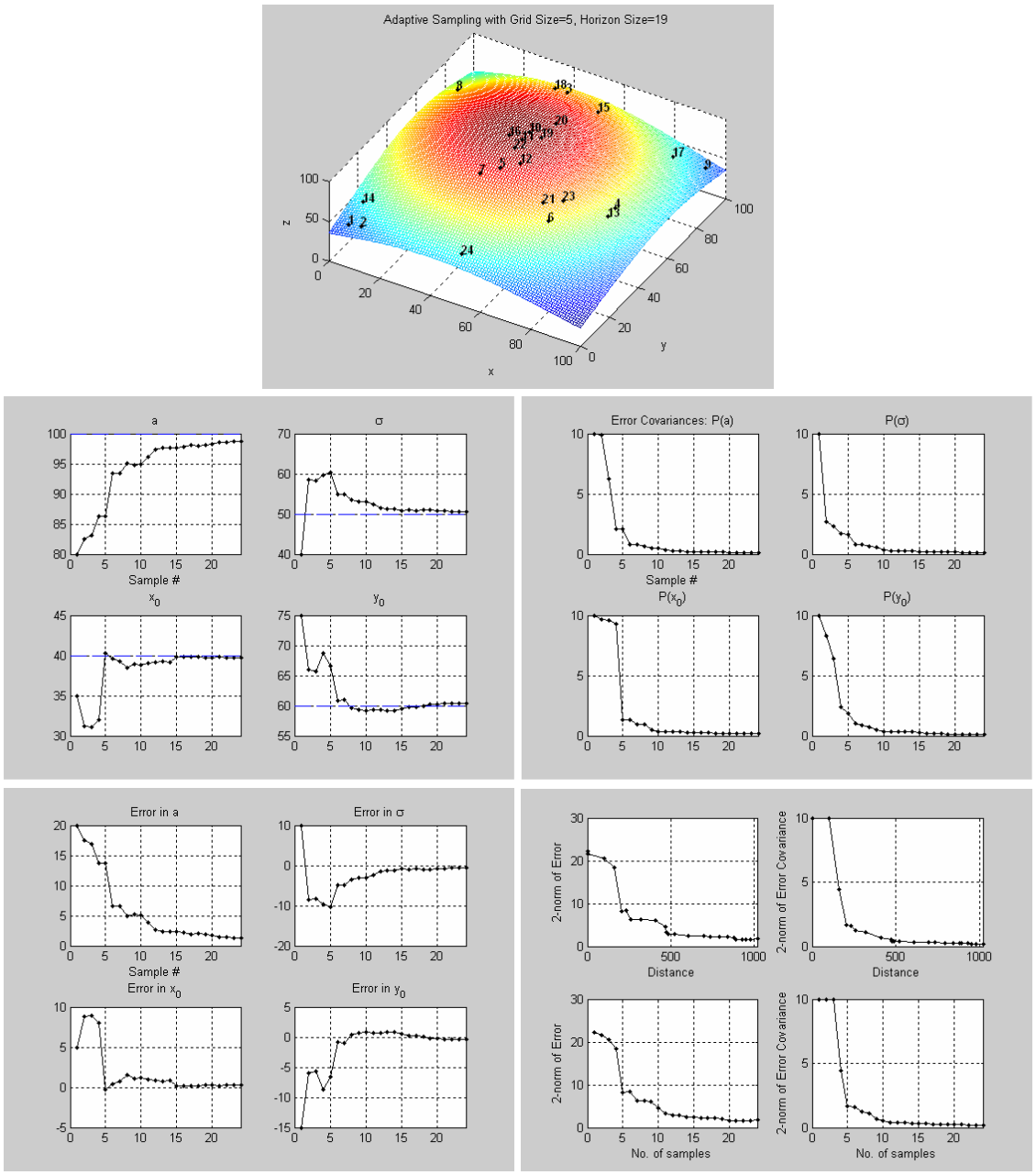


Figure 5.22 Adaptive sampling of a Gaussian field with grid size 5 and horizon size 19

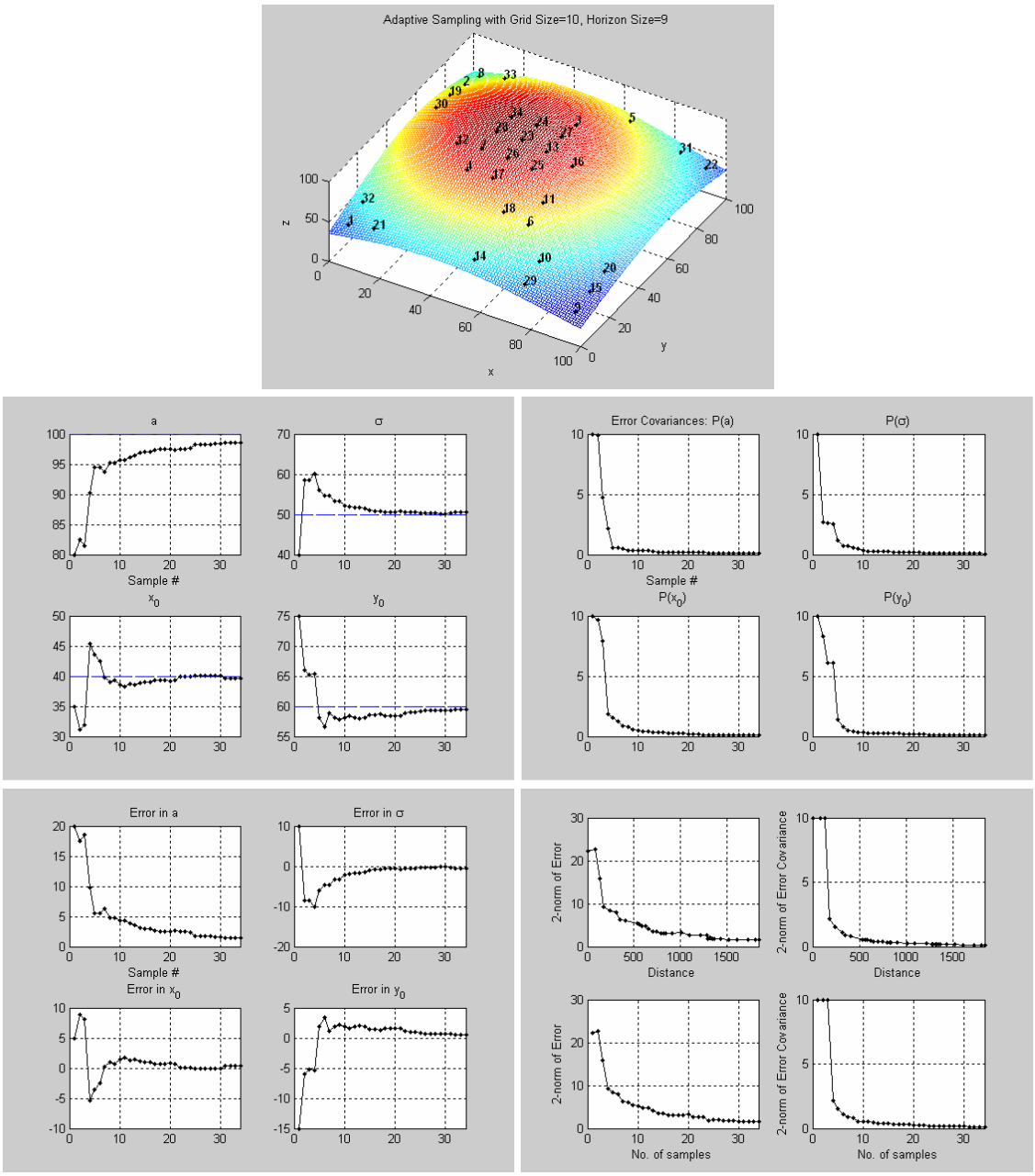


Figure 5.23 Adaptive sampling of a Gaussian field with grid size 10 and horizon size 9

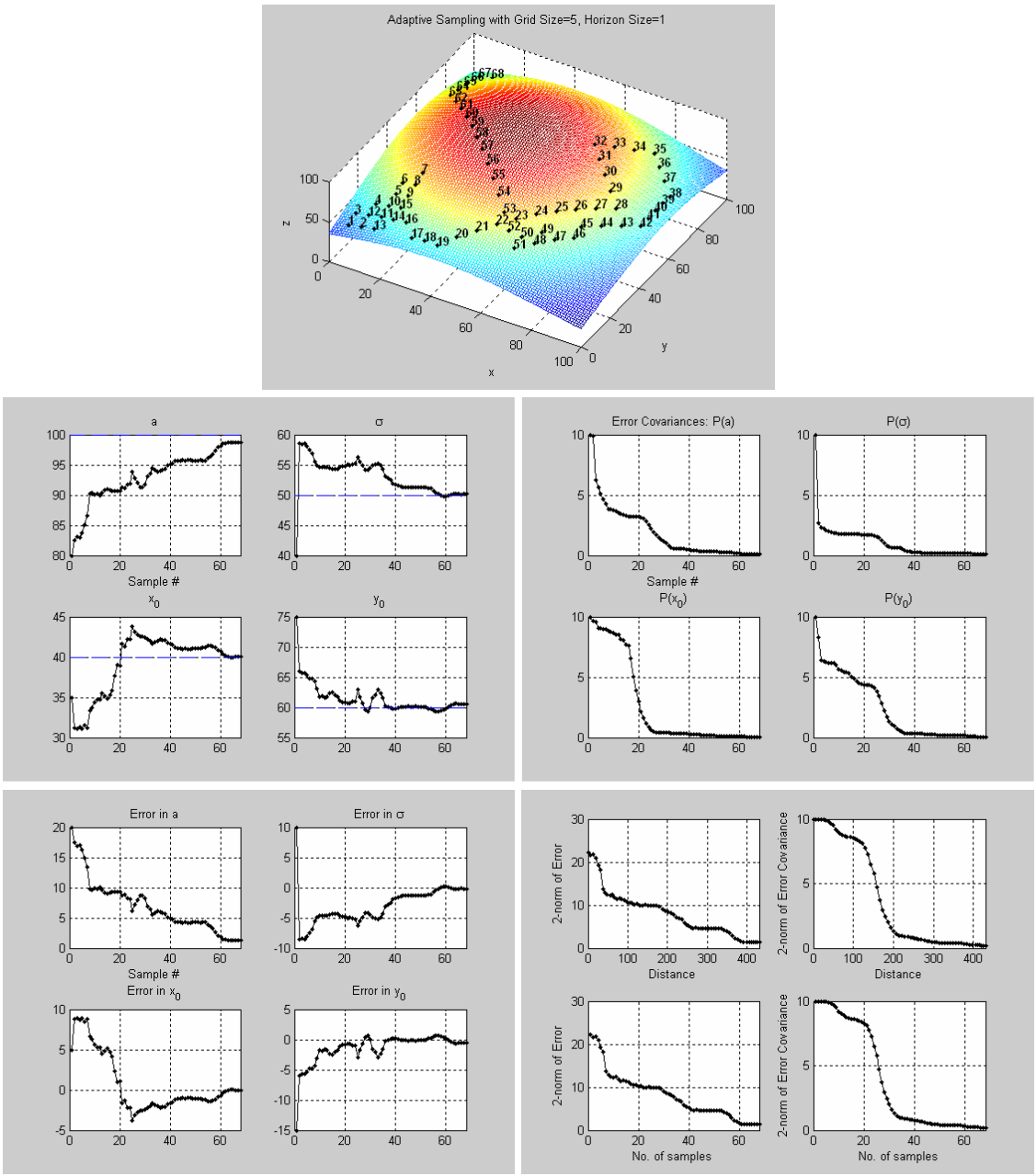


Figure 5.24 Greedy adaptive sampling of a Gaussian field with grid size 5 and horizon size 1

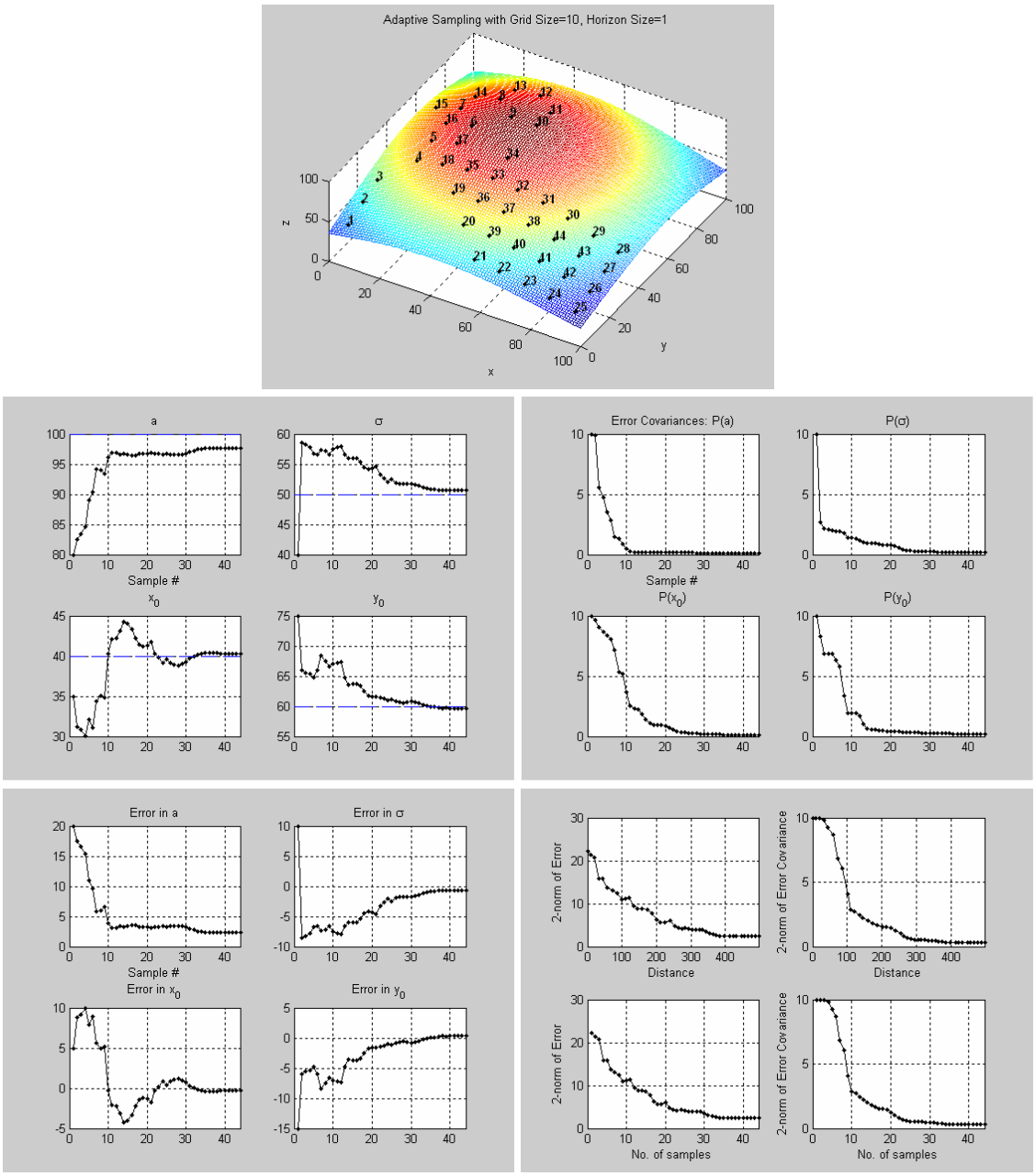


Figure 5.25 Greedy adaptive sampling of a Gaussian field with grid size 10 and horizon size 1

5.2.2 Estimating parameters of single Gaussian field and robot location states by field measurements only

We now consider the case of sampling a stationary Gaussian field with four unknown parameters defined by equation (5.6) with a differential drive robot. Both location states for the robot $X_k = [x_k \quad y_k]$ and field parameters $A = [a \quad \sigma \quad x_0 \quad y_0]$ need to be estimated. The kinematic model for the robot will be discussed later, in Chapter 6, and is shown in Figure 6.6. The robot is equipped with wheel encoders that are used to estimate its position, however, the source of error in robot localization is the difference in wheel radii between the actual and the nominal robot models as discussed in section 6.1.3. Therefore, during simulation the robot ended up at locations different from where it thinks it is based on its dead-reckoning model. The robot kinematic model now included in the Joint EKF as described in equations (6.2), and the following numerical assumptions were used in the simulation:

- Actual field parameters - $[a \quad \sigma \quad x_0 \quad y_0] = [100 \quad 50 \quad 40 \quad 60]$,
- Initial parameter estimates - $[80 \quad 30 \quad 45 \quad 55]$,
- Final parameters estimate - $[99 \quad 50 \quad 42 \quad 57]$,
- States to estimate - $[a \quad \sigma \quad x_0 \quad y_0 \quad x \quad y]$,
- Initial error covariances
- $P_{0a} = 100,\ P_{0\sigma} = 100,\ P_{0x0} = 100,\ P_{0y0} = 100,\ P_{0x} = 100,\ P_{0y} = 100$
- Model error covariances - $Q_a = 0,\ Q_\sigma = 0,\ Q_{x0} = 0,\ Q_{y0} = 0,\ Q_x = 1,\ Q_y = 1$
- Measurement error covariance - $R = 1$

- The robot wheel radii are different: $r_l = 1,\ r_r = 1.05$.

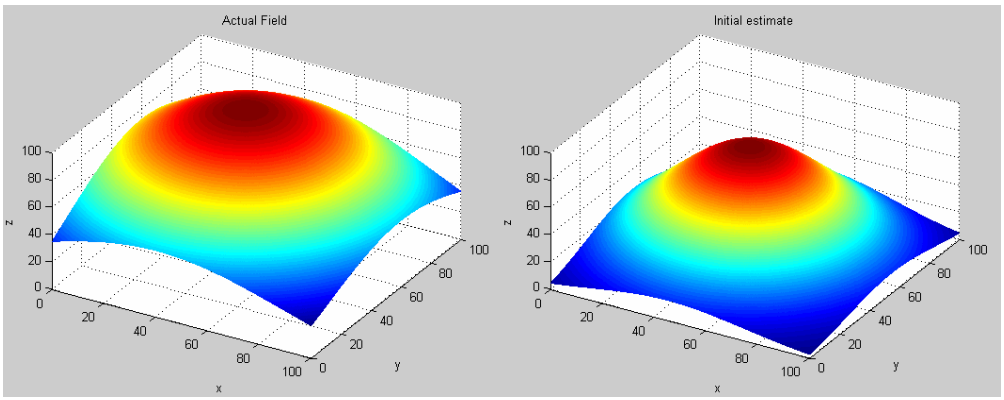


Figure 5.26 The actual field (left), and the initial estimate (right)

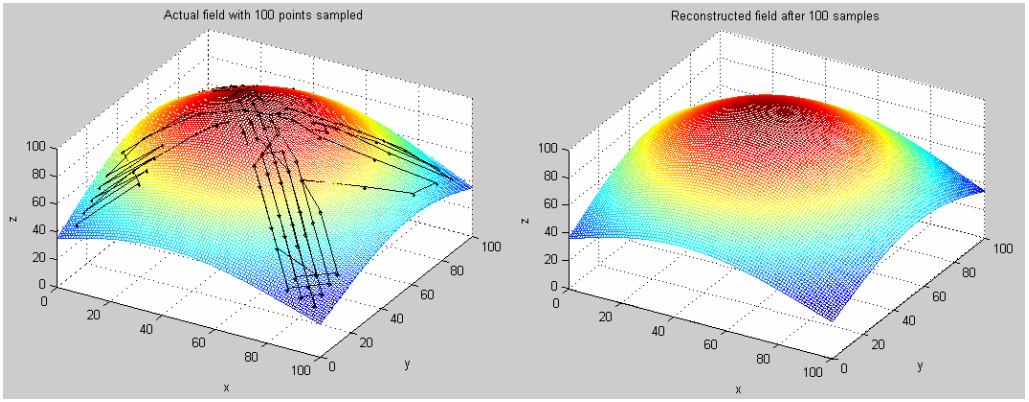


Figure 5.27 The actual field with points sampled using GAS (left), the reconstructed field after 100 samples (right)

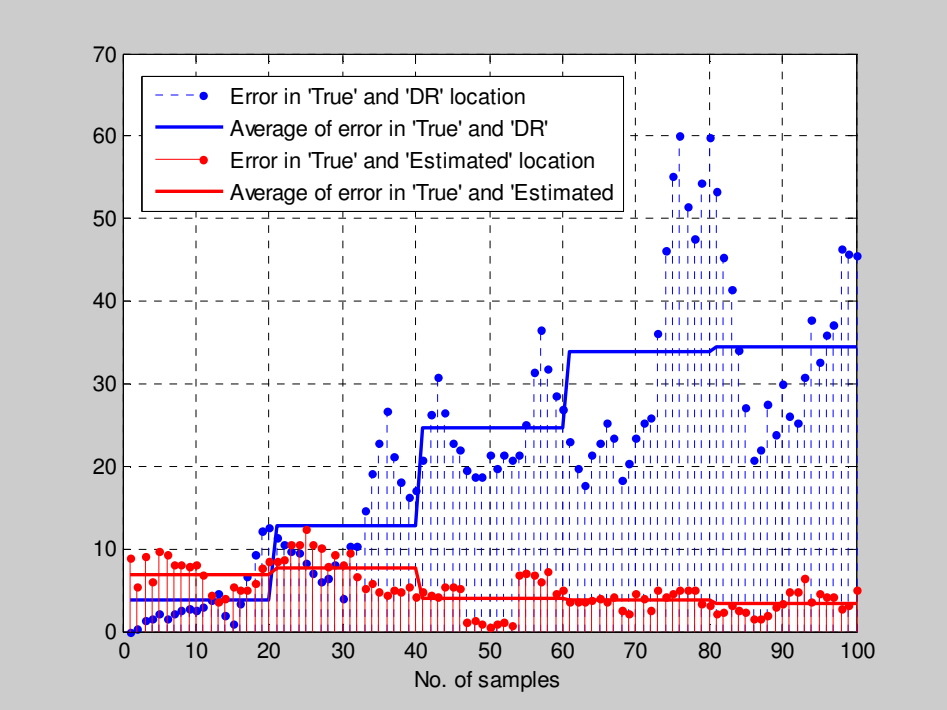


Figure 5.28 Error between true, dead-reckoning and estimated robot locations

Figures 5.26 and 5.27 show the actual Gaussian field, the initial estimate, the path taken by the robot during sampling, and the reconstructed field after 100 samples of GAS. Figure 5.28 plots the 2 norm of the error between the actual robot position (based on the actual wheel radii), the dead-reckoning estimate (based on incorrect values in the model), and the robot location estimate as computed by the joint EKF. It can be observed from the Figure 5.28 that initially the dead-reckoning error (in blue) is small but it grows as time passes. Also, initially the estimated location based on field measurements is not very good (in red). However, while the dead reckoning error keeps accumulating, the robot localization error stays bounded with increasing number of samples because as more samples are taken the algorithm has better knowledge of the field, and this helps localize the robot. A windowed average for the parameter estimates

is taken every 20 samples, and the trends are shown by solid horizontal lines in Figures 5.29-5.31.

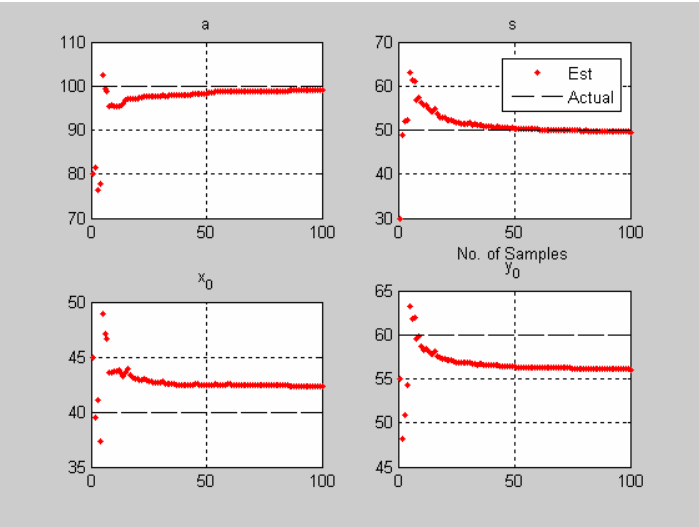


Figure 5.29 Convergence of field parameters to actual values

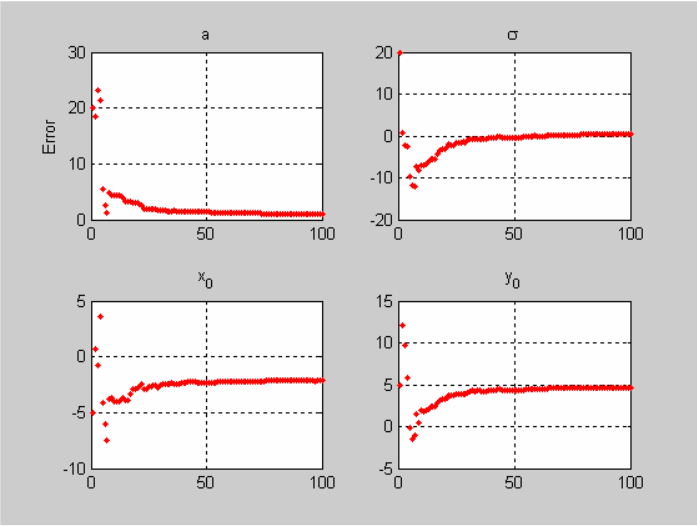


Figure 5.30 Error in actual and estimated parameter values

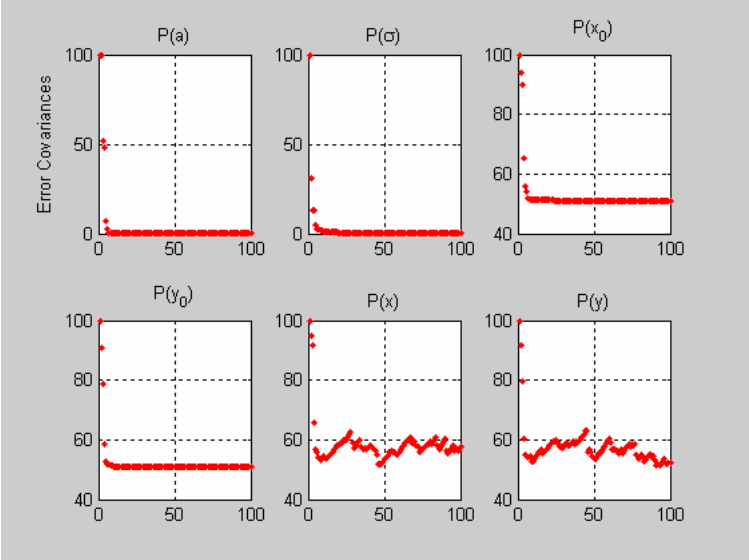


Figure 5.31 Error covariance of location states and field parameters

5.3 Simulations for forest fire mapping

In this section, increasingly realistic fire spread models are used to assess the effectiveness of the sampling algorithm presented in Chapter 4, by comparing it to a basic raster scanning sampling approach. Also the performance of the multi-rate EKF scheme to estimate the time-varying fire field model is also investigated by simulation. First, sampling results on a single Huygen's spread model are presented. We then consider the case of a field represented by a sum of five Gaussians with slow spreading over time, and finally, the case of a time-varying CA fire spread complex model. In all simulations, sampling is performed with a single robotic vehicle. For multiple vehicles, the EKF computations can be distributed among robots as discussed in Chapter 7. The model of the robots is ignored in order to focus on quantifying the accuracy of field estimation.

5.3.1 Elliptically constrained single Gaussian time-varying forest fire field

Simulations are performed to estimate four fire field parameters given in equation (4.10)-(4.12) for a single ellipse. The numerical simulation model assumed a forest area of 1 sq. mile, in which multiple robots take fire intensity measurements (or some measure of temperature) and estimate the desired parameters. The location of firehead x_0 is time-varying. The fire can spread at arbitrary rates, but in this simulation, the case of a slow sinusoidal fire spread rate was considered as an example:

$$\dot{r} = \frac{2\pi}{60}\cos(\frac{2\pi}{60}t) \quad . \tag{5.7}$$

Other choices of rate spread will not affect the rate of convergence as long as the spread model is known, and the rate of spread is much smaller than the speed of sampling. As a result of the fire spread rate in equation (5.7), x_0 will be sinusoidal with respect to time and space, as shown in Figure 5.32, while the other fire field parameters are stationary. A comparison sampling simulation was conducted between sampling using raster scanning and heuristic greedy adaptive sampling. For GAS, the algorithm looks for the next best sampling location in a circle of 50 ft radius around the currently sampled location as shown in figure 5.32. Raster scanning does a row-by-row scanning.

Table 5.4 summarizes the sampling results of Greedy AS and Raster Scanning when sampling performed for 60 minutes. We assumed that the robot navigation speed was 30 ft/min, that the robot sampling and processing time are neglected, while $Q = 0,\ R_1 = 0.1$.

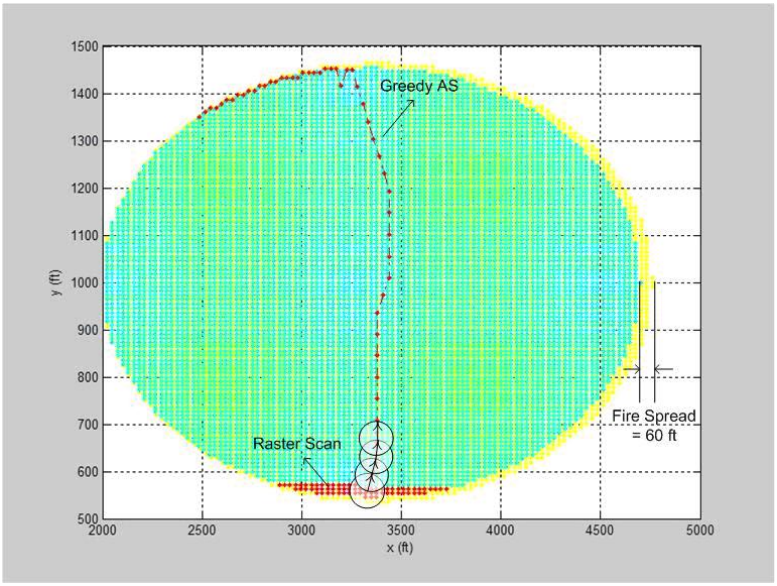
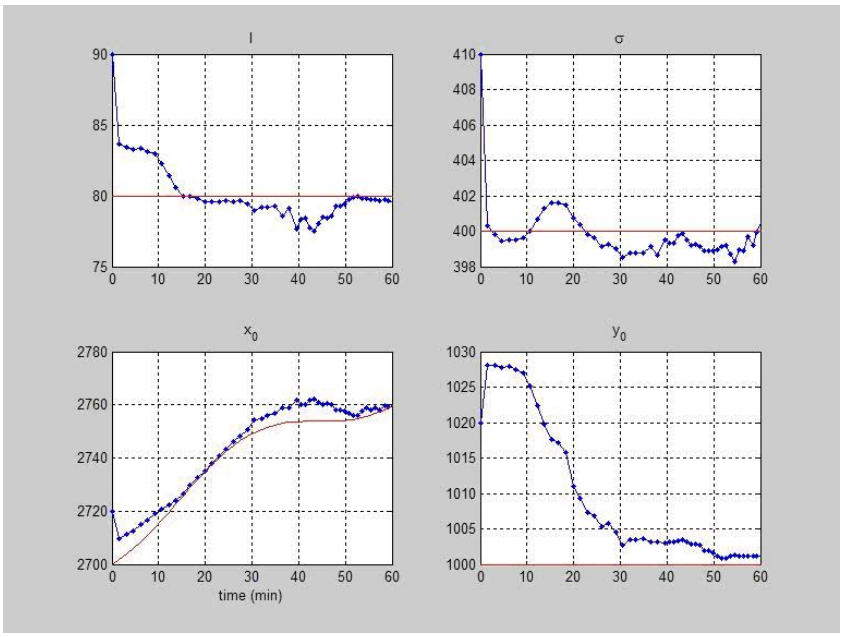
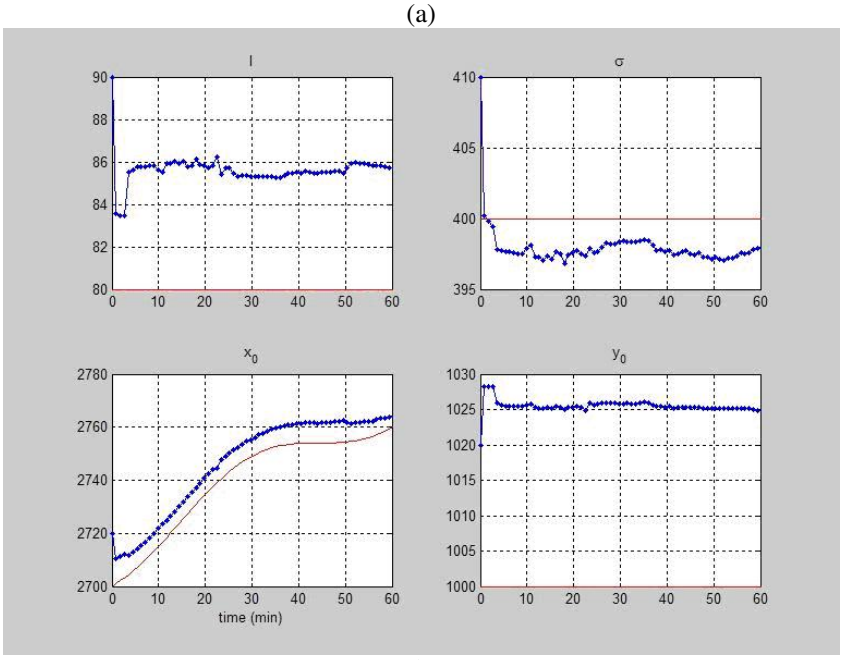


Figure 5.32 Elliptical fire spread with greedy adaptive sampling algorithm which looks for the appropriate location on the elliptical fire

Results indicate that GAS leads to faster convergence, and requires considerably less numbers of sampling points than RS or AS. In Table 5.4, the 2-norm of the error covariance P_k is 7.3 for GAS, and 44.2 for Raster Scanning after 60 minutes of sampling. If sampling continues further, raster scanning takes almost 6 times more time than GAS to converge to same parameter estimate values. We also observed that the error covariance decreases very slowly for raster scanning, and it will require almost the entire sampling area to reduce the parameter estimate uncertainty. Raster scanning performs even worse when sampling is being conducted in an area where the parameter of interest does not vary significantly.



(a)



(b)

Figure 5.33 Actual (red) and estimated (blue) fire field parameters I, σ, x_0 and y_0 and versus time when sampling operation is performed for 60 minutes using greedy approach (a) & raster scanning (b)

Table 5.4 Comparison of Raster Scan and Greedy AS for sampling of elliptically constrained single Gaussian time-varying field

		A_0	A_k	$\hat{A}_0$	$\hat{A}_k$	P_0	P_k
Greedy AS (k=49)	I	80	80	90	79.4	10	0.7
	σ	400	400	410	401.1	20	4.9
	x_0	2700	2760	2720	2759.7	50	3.8
	y_0	1000	1000	1020	1001.2	50	3.8
Raster Scan (k=68)	I	80	80	90	85.7	10	2.4
	σ	400	400	410	397.9	20	5.1
	x_0	2700	2760	2720	2765.5	50	21.8
	y_0	1000	1000	1020	1024.8	50	38.1

5.3.2 RBF-NN parameterization using low-resolution information

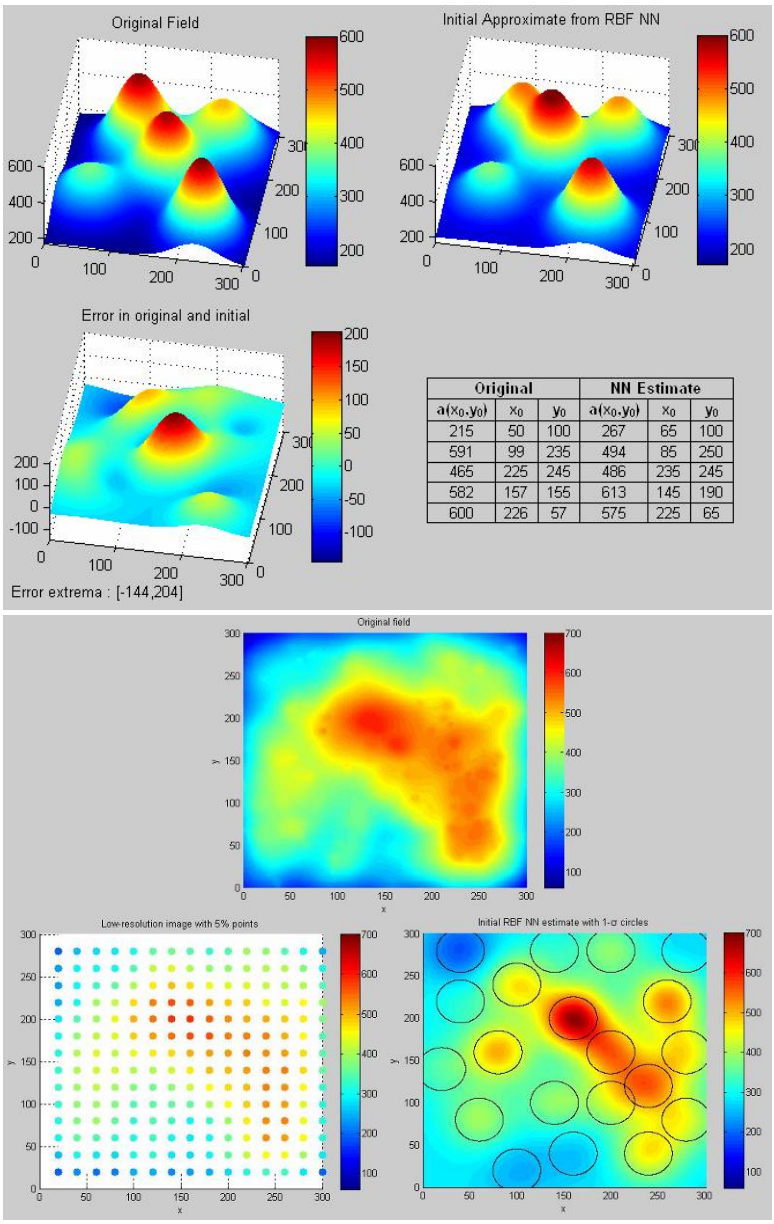
Radial bases neural network using "newrb" is available in Neural Network toolbox of MATLAB. "newrb" creates a two-layer network, with the first layer containing "radbas" neurons and the second, "purelin" neurons. In this section, we make use of this Toolbox to simulate the sampling process of a slightly more complex field than a single Gaussian.

In Figure 5.34 (a), a simple field is presented where a sum of 5 Gaussians field is approximated by a RBF neural network with 5 neurons, while in Figure 5.34 (b) a more complex field is displayed. A low-resolution version of the original fields is acquired by averaging points in a square such that only a small percentage of the total numbers of points are used for training the neural network. The number of neurons and

the spread factor are chosen such that the 'normalized SSE' between low-resolution of the actual field and the estimated field is kept below an acceptable threshold of 1.

As discussed in Chapter 2, the neural network learning algorithm doesn't train for spread parameters. Although some heuristic algorithms exist to select different values for spread factor, we are assuming a constant spread value. For our simulations, p=300, a typical case a grid size is nxn = 20x20, spread factor is 30 with 5 neurons are assumed as shown in figure 5.34 (top). The 2-norm of error between actual and initial estimated field is 38.7 and error extrema are -144 and 204.

A similar scheme is used to approximate the complex field, which is generated using CA discussed in Chapter 4. Since the field is more complex, more neurons are required for a good approximation. At a particular time, a low-resolution version of field is taken by averaging on a grid size of 20x20 and passed through RBF NN training for 20 neurons and spread factor of 40. The training using the hybrid algorithm only takes 2-3 seconds for 60 neurons neural network. Hence, the training time is much smaller than the speed of field evolution.



Original			NN Estimate		
$a(x_0,y_0)$	x_0	y_0	$a(x_0,y_0)$	x_0	y_0
215	50	100	267	65	100
591	99	235	494	85	250
465	225	245	486	235	245
582	157	155	613	145	190
600	226	57	575	225	65

Figure 5.34 Sum-of-five Gaussian fire field approximation using 5 RBF neurons (top), Complex field approximation using 20 neurons and a grid size of 20x20 (bottom)

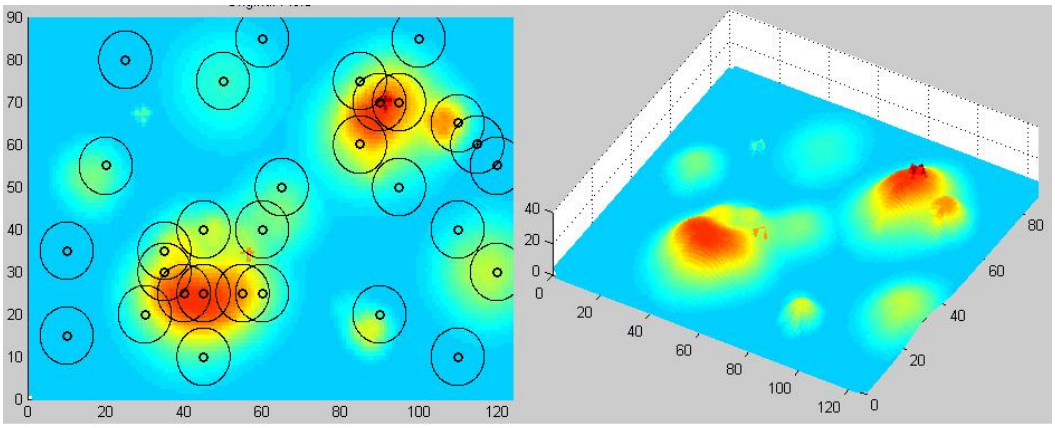


Figure 5.35 Illustration of the presence of more overlapping Gaussians in areas of large variance for thorough sampling

Figure 5.35 also illustrates the use of self-organized selection of centers classification algorithm used for selection of centers while training the neural network. The objective of the classification is to introduce more neurons in high-variance areas of the field, and fewer neurons in low-variance areas. Figure 5.35 shows a sum of 30 Gaussians approximated with a 30 neuron RBF neural network.

The plots shown in Figure 5.36 are the 2-norm of relative error of all the points between the actual fire field and the field estimated by the neural network. The estimated field is achieved by considering lower resolutions of the actual field with grid sizes of *nxn* = 5x5, 10x10 and 20x20. In Figure 5.36 (a) a 5x5 size grid is considered, hence the error in smaller compared to Figure 5.36 (b) where a 10x10 size grid is considered. An increase in the number of neurons decreases the error but after a while, the error doesn't reduce any further. For a spread factor of 40 and 60 the error stays the same even if more than 40 neurons are considered. Figure 5.36 (c) illustrates the obvious fact that taking smaller size grid (indicating a higher resolution) increases the accuracy of initial estimate with same number of neurons. As the number of neurons

increases, the initial estimate gets better. This is valid until the neural network becomes over-trained. Figure 5.36 (d) shows the error in approximating a sum-of-five Gaussians field with RBF NN with different number of neurons when a 20x20 size grid and different spread factors are considered. Increase in the spread factor decreases the error as the number of neurons increase but leads to saturation as shown in Figure 5.36 (d) where the error for spread factor of 80 is higher compared to 60. Simulations are performed to estimate the 21 parameters of the field represented by a sum of five Gaussians. The system is parameterized using NN as explained in Chapter 3 and the sampling results are shown in Figure 5.36.

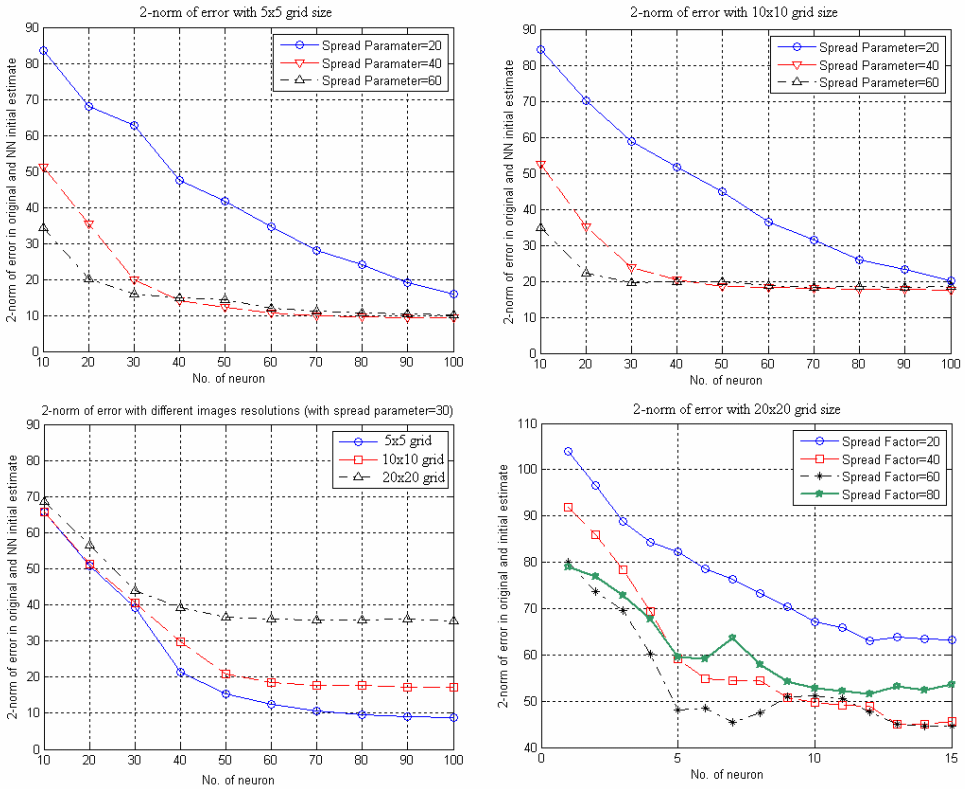


Figure 5.36 Effect of number of neurons, spread factor and number of training points on the error in estimate

In our simulations, additional numerical assumptions for uncertainties were as follows:

$$P_{b0} = 200, P_{a0} = 50, P_{\sigma 0} = 10^{-8},\ P_{x00} = 4,\ P_{y00} = 4$$
$$Q = 0,\ R = 1 \tag{5.8}$$

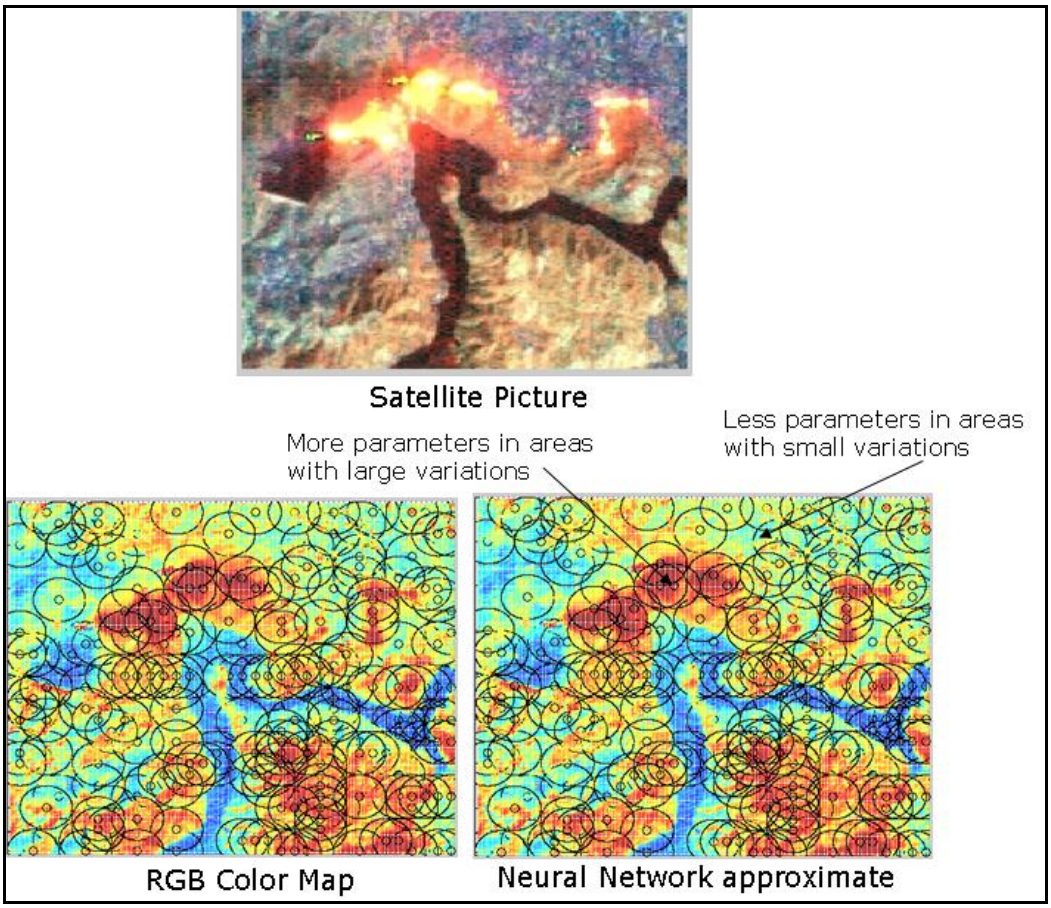


Figure 5.37 Actual image of California 2007 fires taken from a NASA satellite (top), RGB intensity map (bottom left), and RBF-NN approximation with 300 neurons (bottom right)

Figure 5.37 shows the NN approximate when an actual fire field satellite image was used to train the neural network. For a complex continuous field, the following heuristic value for the spread factor was used to improve the NN training error:

$s = \sqrt{\frac{xy}{N}}$, where s is the spread parameter, x and y are maximum values of coordinates, and N is the number of neurons.

5.3.3 Sum-of-Gaussians stationary field

A comparison simulation of GAS and Raster Scan sampling was carried out for the sum of 5 Gaussians nonlinear field. The sampling area was divided into square grids and several search horizons were considered. For Greedy sampling, a grid size of n = 5, p = 5, and horizon size of 5 square grids were assumed. The simulation stopped when the 2-norm error between the actual and estimated fields reduces below 15. It is depicted in Figures 5.38 & 5.39 (f) that at the start of sampling represented by 'black' circles with center and radius $(\hat{x}_{k+1}, \hat{y}_{k+1}, \hat{\sigma}_{k+1})$, coincides with the 'red' circles with center and radius $(\hat{x}_0, \hat{y}_0, \hat{\sigma}_0)$, but start chasing the 'blue' circles which have center and radius (x, y, σ), as the sampling continues. Simulation results for raster scan sampling are shown in Figure 5.38, where it took 170 samples to achieve the norm of error in the original and estimated field less than 15, while the 2-norm of error covariance reduced from 229.5 to 10.72. Simulation results for GAS are shown in Figure 5.39, where sampling required 41 points to achieve the norm of error in the original and estimated field less than 15, and the 2-norm of error covariance to reduce from 229.5 to 2.04. A comparison between initial and final errors is shown in Figures 5.38 & 5.39 (h) and (i).

Table 5.5 clearly indicates the Greedy AS performs much better than Raster Scan in terms of sampling distance (time) as well as number of samples. Also it is apparent in Figures 5.38 & 5.39 that the norm of error between the original and

estimated fields, as well as the norm of state error covariances decrease faster in the case of GAS.

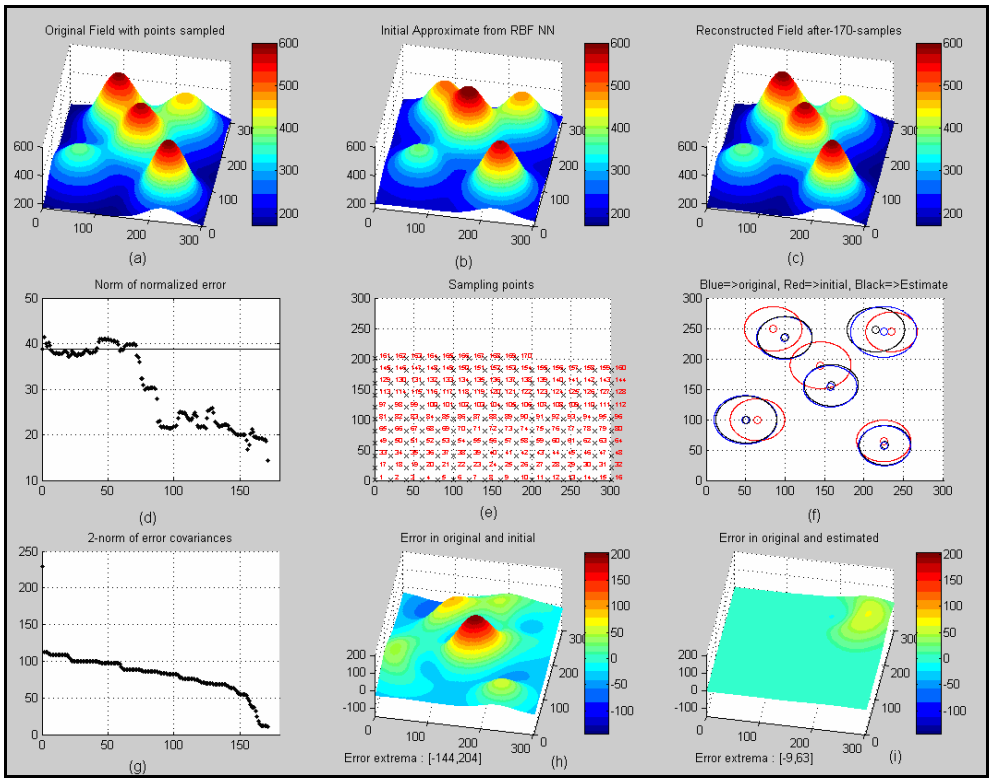


Figure 5.38 Simulation results with raster scanning sampling for sum of Gaussians stationary field

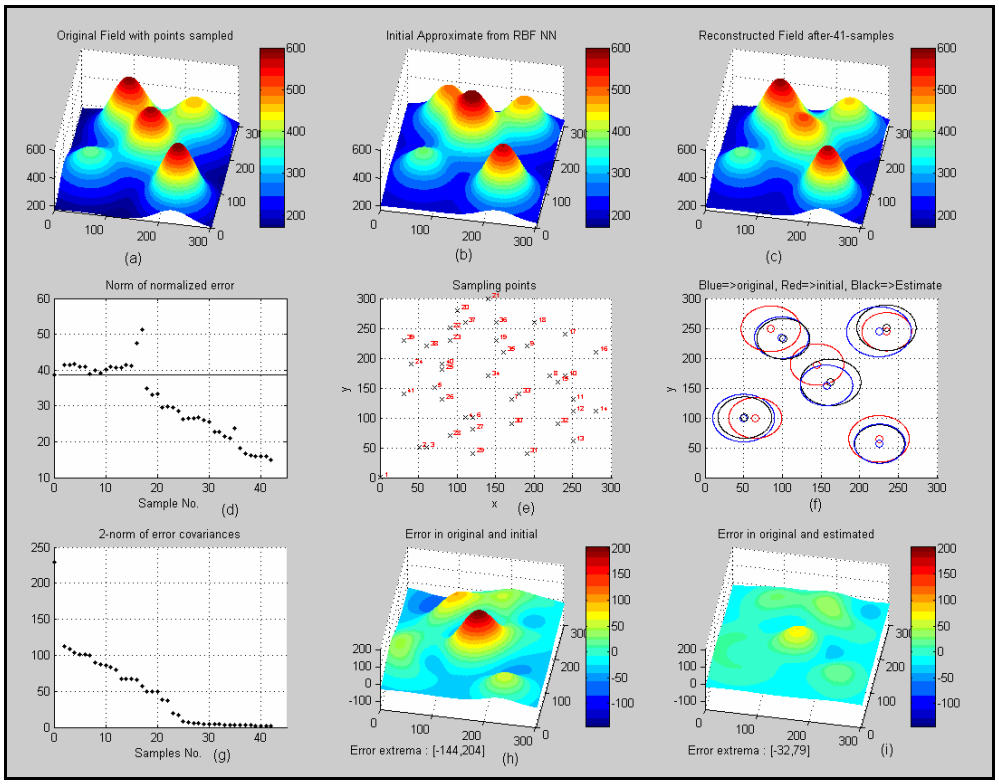


Figure 5.39 Simulation results with GAS for sum of Gaussians stationary field

Table 5.5 Comparison of Raster Scan and Greedy AS for sum of Gaussians field

	Greedy AS	Raster Scan
Distance	2175	3400
No. of samples	41	170
Initial 2-norm of error in actual and estimated field	38.7	38.7
Final 2-norm of error in actual and estimated field	14.97	14.26
Initial 2-norm of error covariance	229.5	229.5
final 2-norm of error covariance	2.04	10.72

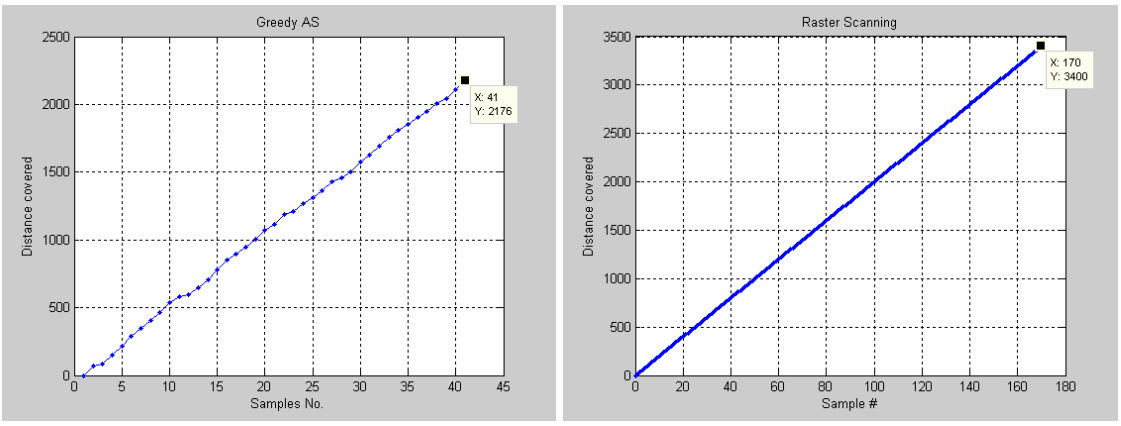


Figure 5.40 Distance covered for Greedy AS (left) and Raster scanning (right)

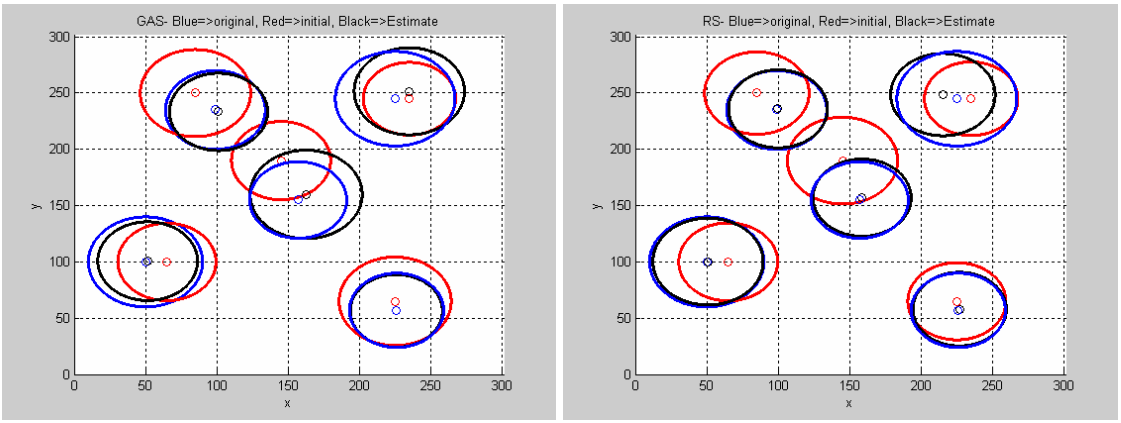


Figure 5.41 Location of Gaussian centers initial and after sampling is done

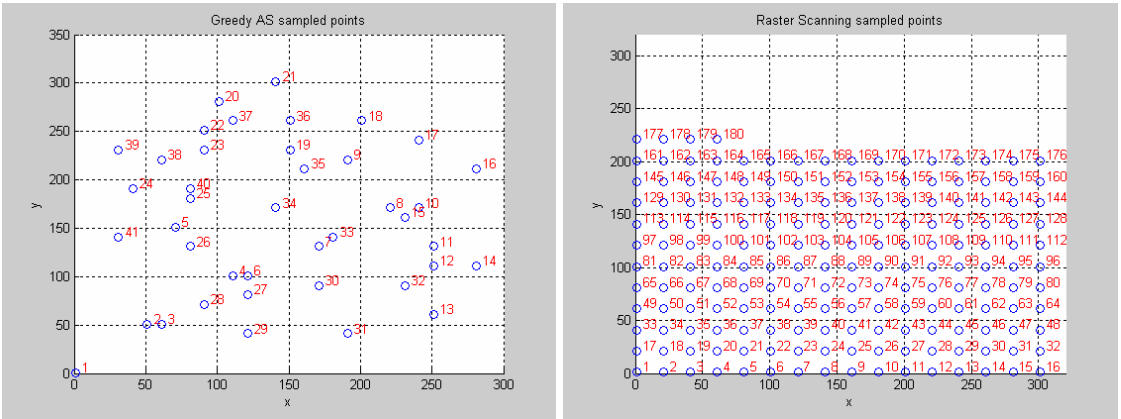


Figure 5.42 Sampling points for GAS (left) and Raster Scanning (right)

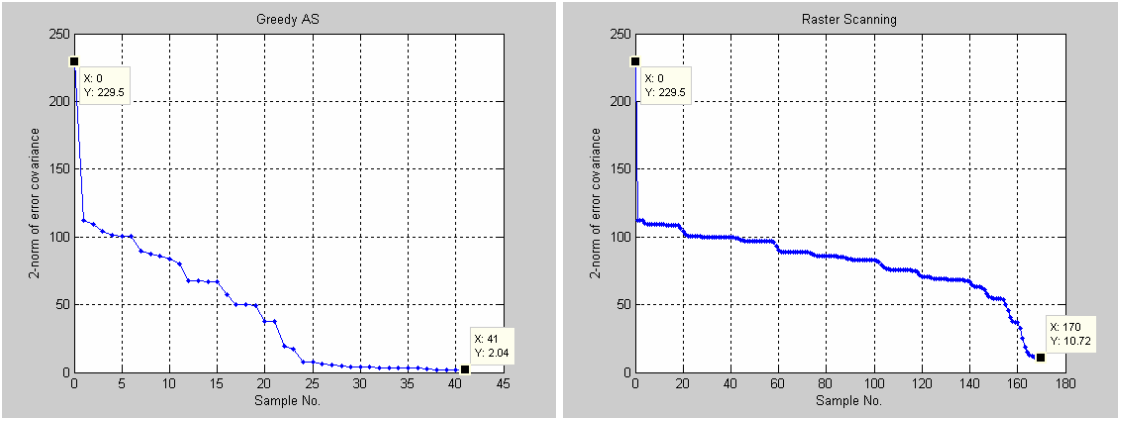


Figure 5.43 *2-norm* of Error Covariance for GAS (left) and Raster scanning (right)

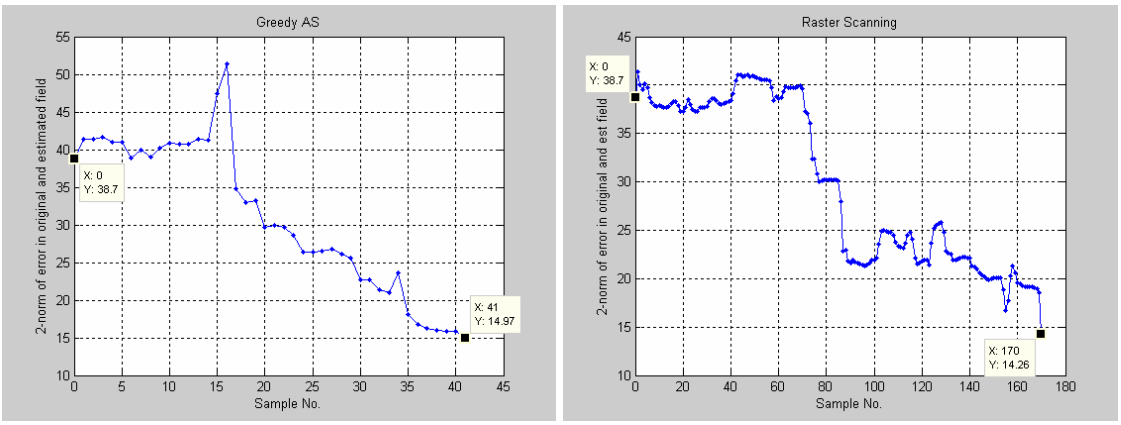


Figure 5.44 *2-norm* of error in original and estimated field

5.3.4 Sum-of-Gaussians time-varying field

Here we make the sum of 5 Gaussians to vary slowly, and we re-run our sampling algorithms. Depending on the anticipated variation of the field parameters based on the CA model and NN inverse model, the inputs $B_k u_k$ will vary, but this variation is available to the EKF estimator according to Equation (4.16).

The numerical values for simulation uncertainties were as follows:

$$P_{b0} = 200, P_{a0} = 50, P_{\sigma 0} = 10^{-8},\ P_{x00} = 4,\ P_{y00} = 4,$$
$$Q_{b0} = 0.5\ Q_{a0} = 3,\ Q_{\sigma 0} = 0,\ Q_{x00} = 0,\ Q_{y00} = 0, \tag{5.9}$$
$$R = 1$$

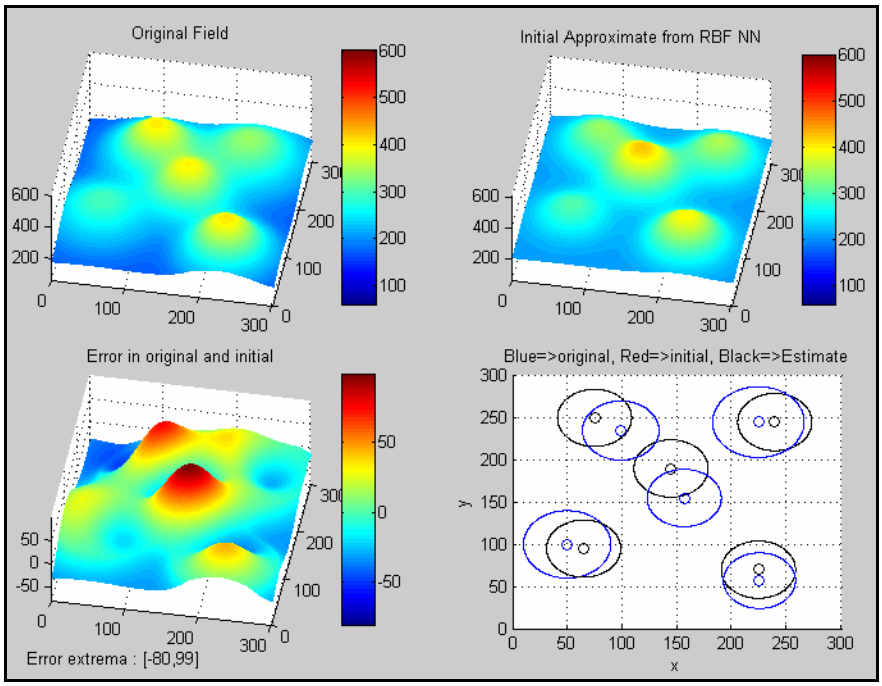


Figure 5.45 Actual and initial estimated field from NN for Sum of Gaussians time-varying field

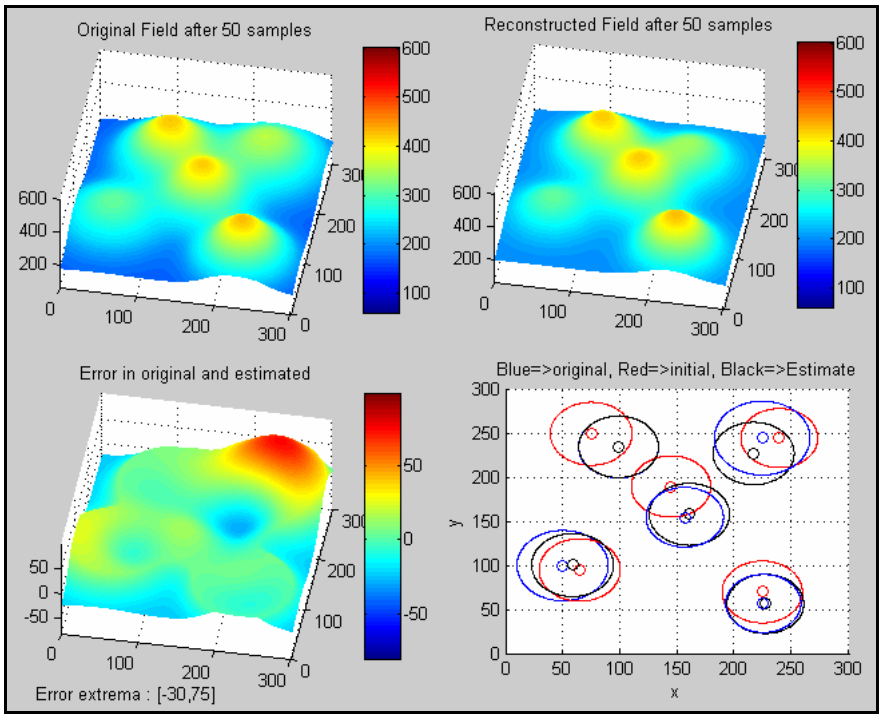


Figure 5.46 Actual and estimated field after 50 samples for Sum of Gaussians time-varying field

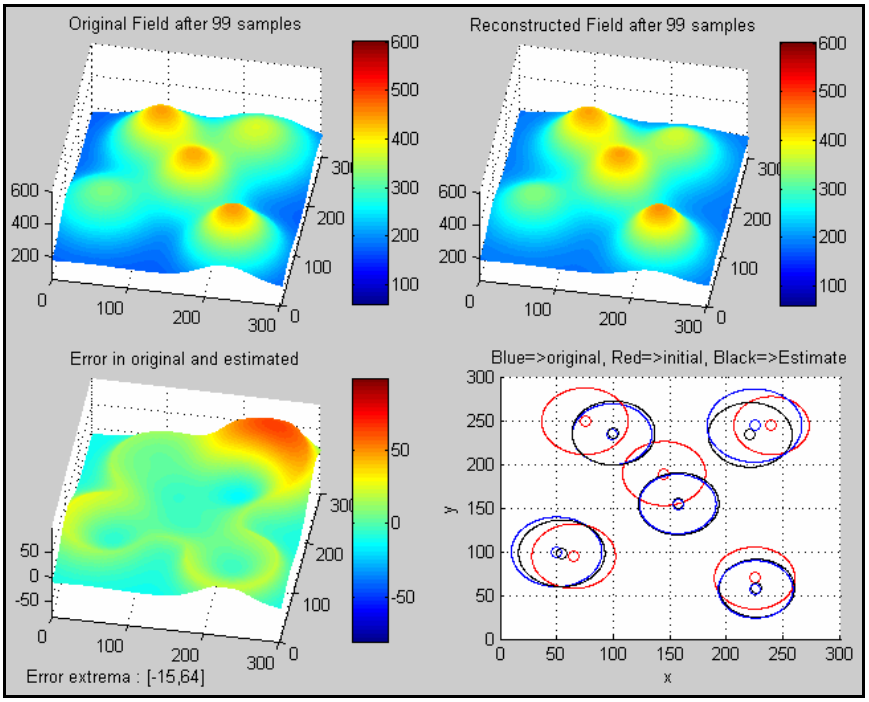


Figure 5.47 Actual and estimated field after 99 samples for Sum of Gaussians time-varying field

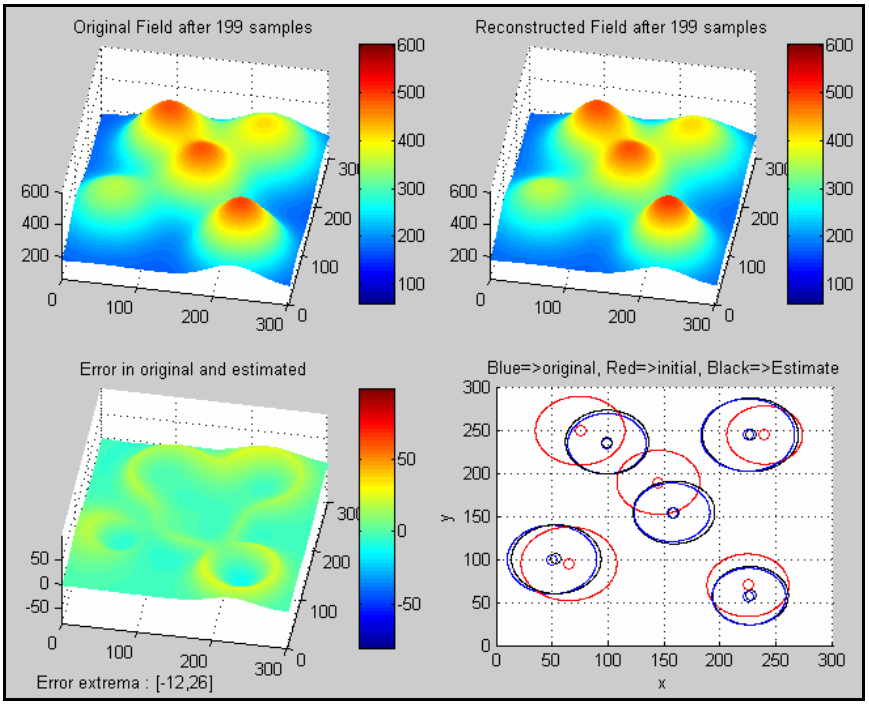


Figure 5.48 Actual and estimated field after 199 samples for Sum of Gaussians time-varying field

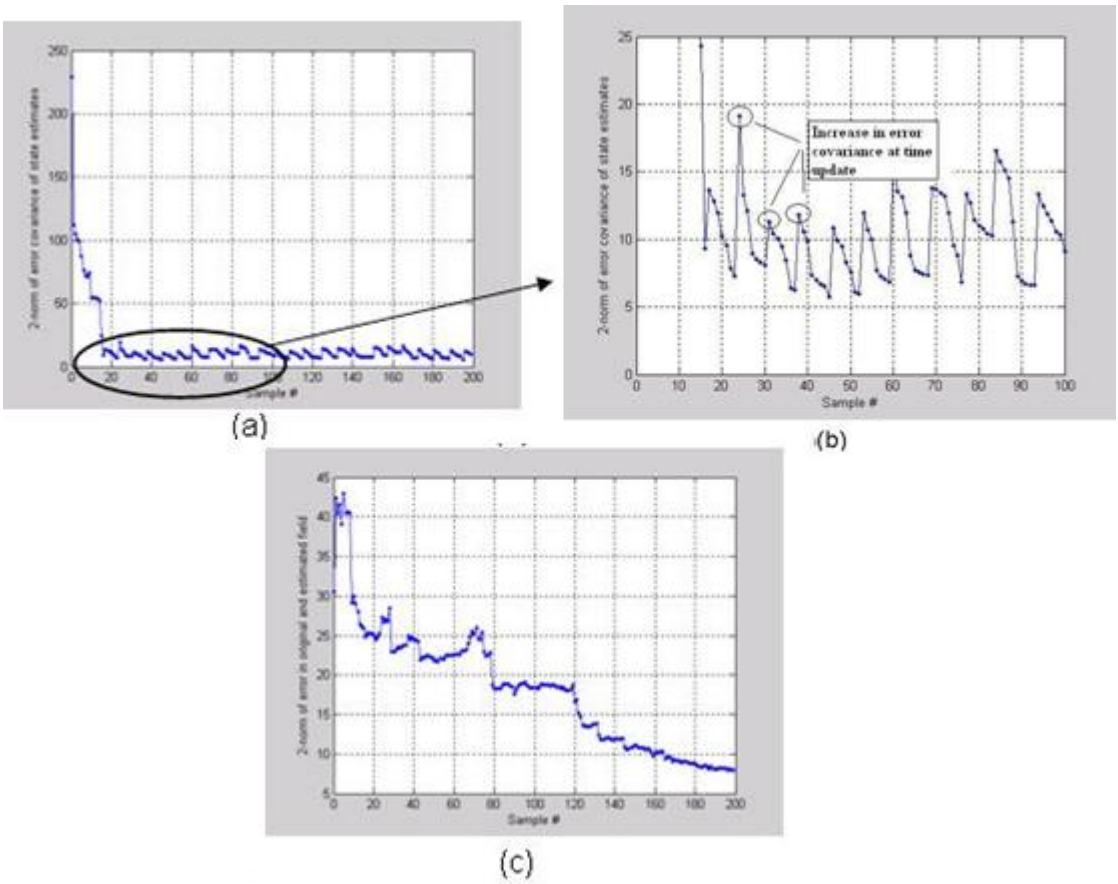


Figure 5.49 (a) 2-norm of error covariance, (b) Increase in error covariance when time update occurs and reduces again when measurements are taken, and (c) 2-norm of error in actual and estimated field

Figure 5.45 shows the original field and the initial NN approximate by taking only 4% samples from the original data. The error between the original and initial estimate is also shown in this figure. 1-σ Gaussian circles of *initial* and *estimate* coincide with one another, and they are different from the *actual* Gaussian centers shown in blue color.

Since the field is slow-varying, we assume that the time update is available slower than the rate at which measurements are taken. In other words, robots take sensor measurements faster than the update in parameters available from remote sensing

measurement. When there is time update, the error covariance of the states goes higher but keeps on reducing again when measurements are taken. This observation is illustrated in Figure 5.49.

5.3.5 Complex RBF time-varying field

In this section, the complexity of the fire field model increases again by considering a slow time varying field generated using the Cellular Automata model presented in Chapter 4. As the sampling algorithm uses an Extended Kalman Filter, the observability of the parameters (algorithm convergence) will depend on the initial conditions. The initial error covariance is selected depending on the error in actual field and the initial estimated field, which in-turn depends on the percentage of data from the actual field which is used for training the neural network, number of neurons and spread parameter. In our simulation models, the following parameters were chosen: the field is defined in a mxm = 300x300 area, and an average of values in a nxn = 30x30 grid is used for training the neural network. 40 neurons are used, with a spread parameter of 30. These parameters can vary as the complexity of the field varies and the goal of sampling is to minimize the sum of squares error. The numerical values for uncertainties in our model were:

$$\begin{aligned} &P_{b0} = 100, P_{a0} = 5, P_{\sigma 0} = 10^{-7},\ P_{x00} = 1,\ P_{y00} = 1, \\ &Q_{b0} = 0.02\ Q_{a0} = 0.02,\ Q_{\sigma 0} = 0,\ Q_{x00} = 0,\ Q_{y00} = 0, \\ &R = 1 \end{aligned} \tag{5.10}$$

Figure 5.50 shows simulation results for GAS illustrating (a) actual field generated using CA, (b) initial approximate with 40 neuron RBF-NN and spread factor

of 30 when grid size of n=30 is used for low-resolution sampling, (c) reconstructed field after 168 samples with GAS heuristic sampling approach when grid size of p=5 is used for high-resolution sampling, (d) sum-of-square Error (SSE) in actual and estimated field which drops faster compared raster scanning, (e) sampled points, (f) red and black circles indicate initial and estimated Gaussian locations respectively, (g) 2-norm of error covariance of parameter estimates, which drops faster compared raster scanning, (h) error in actual and initial estimate, and (i) error in actual and final estimate after 168 samples.

Figure 5.51 shows simulation results for raster scan sampling illustrating (a) actual field generated using CA, (b) initial approximate with 40 neuron RBF-NN and spread factor of 30 when grid size of n=30 is used for low-resolution sampling, (c) reconstructed field after 951 samples with raster scanning when grid size of p=10 is used for high-resolution sampling, (d) sum-of-square Error (SSE) in actual and estimated field, (e) sampled points, (f) red and black circles indicate initial and estimated Gaussian locations respectively, (g) 2-norm of error covariance of parameter estimates, (h) error in actual and initial estimate, and (i) error in actual and final estimate after 951 samples.

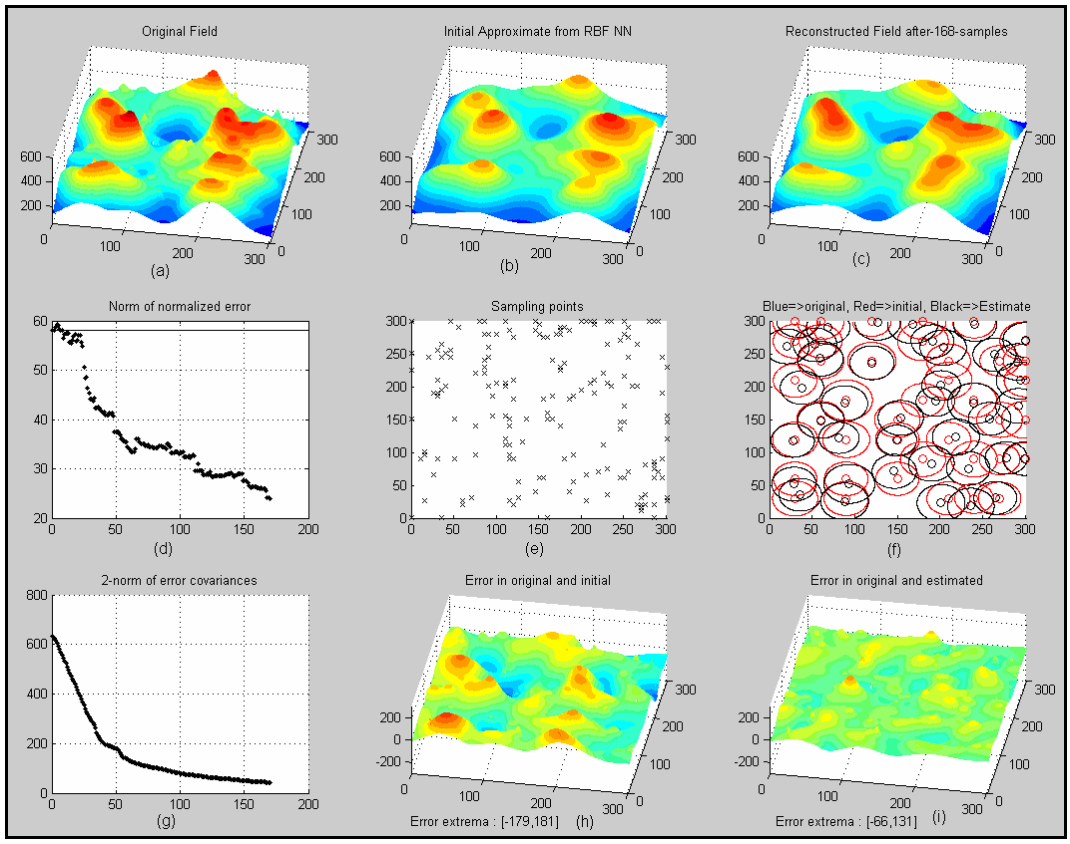


Figure 5.50 Simulation Results for GAS for complex time-varying field

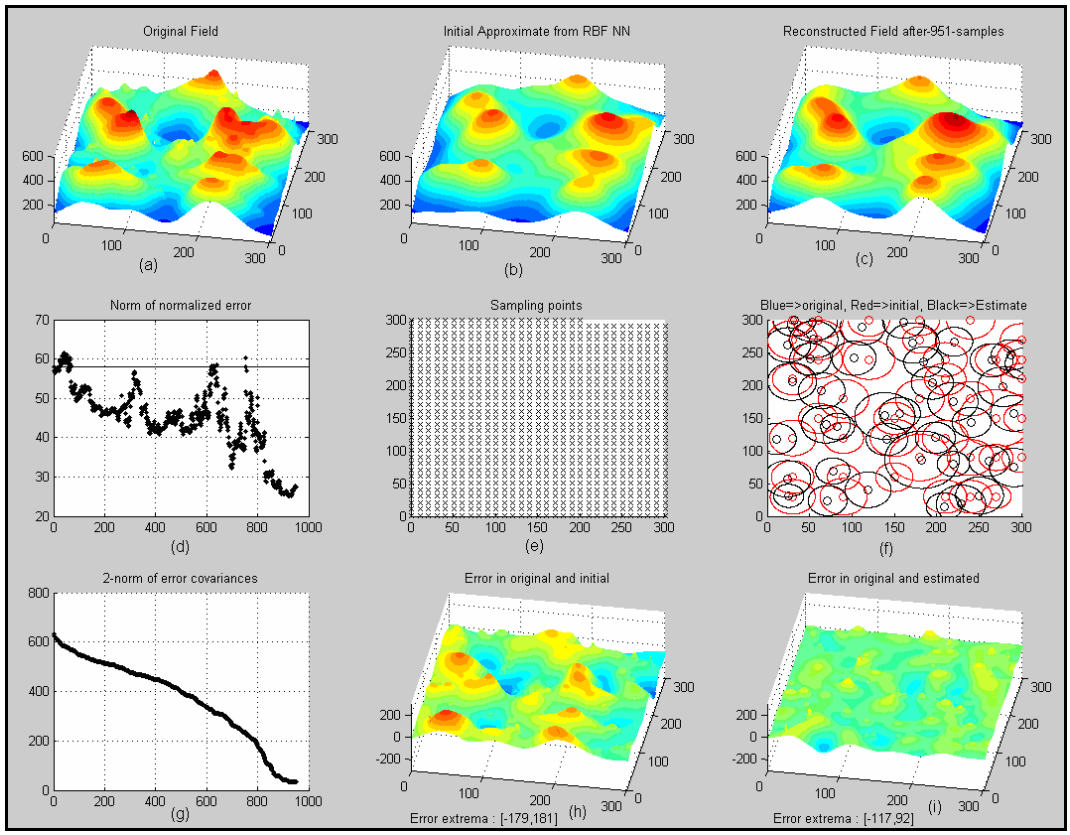


Figure 5.51 Simulation Results for Raster scan sampling for complex time-varying field

Table 5.6 Comparison of Raster Scan and Greedy AS for complex time-varying field

	Greedy AS	Raster Scan
Distance	3380	19020
No. of samples	168	951
Initial 2-norm of error in actual and estimated field	58.07	58.07
Final 2-norm of error in actual and estimated field	23.95	27.49
Initial 2-norm of error covariance	632.5	632.5
final 2-norm of error covariance	43.23	33.62

Figure 5.51 shows the raster scanning simulation results when sampling in performed row-by-row in a grid size of pxp = 20x20. Figure 5.50 shows the GAS sampling simulation results when a grid size of p = 5 and a horizon size of 20 are considered. Grid size is an important parameter. A very large grid size will not reduce the error significantly, even if the whole area is scanned, because thorough sampling is required in high-variance areas. As the robot start sampling with given initial estimates and uncertainties, the uncertainty of the parameters does not decrease until the robot reaches the area where those parameters have a significant influence. In other words, the uncertainty of the Gaussian is most reduced when sampling is performed within a few variance values away from its center.

A comparison of GAS and Raster Scanning is summarized in Table 5.6. We notice that since raster scan performs a row-by-row scanning, it takes a longer time and many more samples than GAS. The simulation stops when 2-norm of error in actual and estimated field reduces below a threshold. It can be seen from Figures 5.50 & 5.51 that SSE (sum of squares error) between the actual and the estimated field, as well as the 2-norm of state error covariances decrease faster in case of GAS.

5.3.6 Potential fields for safe trajectory generation

Simulations were also performed to generate fire-safe paths through the estimated field, as shown in Figure 5.52, as presented in Section 4.4. In our simulations, we assumed that 4 fire ellipses are ignited simultaneously, and that 4 robots are simultaneously sampling in designated areas. Every robot runs a separate EKF based AS algorithm to estimate the parameters of its local fire field. Fire field data is then

aggregated in a central processing location, which is also responsible for on-line fire-safe trajectory generation. The trajectory is dynamically updated from X_i to X_{goal} every time the field parameters estimate updates. In a practical implementation scenario, a human firefighter can carry a wireless device receiving estimates from the robots in order to generate a collision-free path around fire obstacles towards the goals using equations (4.19) & (4.20). Assuming the human crew is at location $X_i = (2000, 2000)$ and needs to go to rescue location $X_{goal} = (2500, 4500)$, the path is divided into 50 segments and repulsive forces from each of these virtual obstacles, and attractive force towards the goal is calculated. The numerical coefficients used in the simulation are: $\xi = 1, \lambda = 10^6$, $\delta = 0.01$. The trajectory is updated using this suitable δ and net force $\vec{F}_p$ value using equation (4.21), and a fire-safe trajectory is generated.

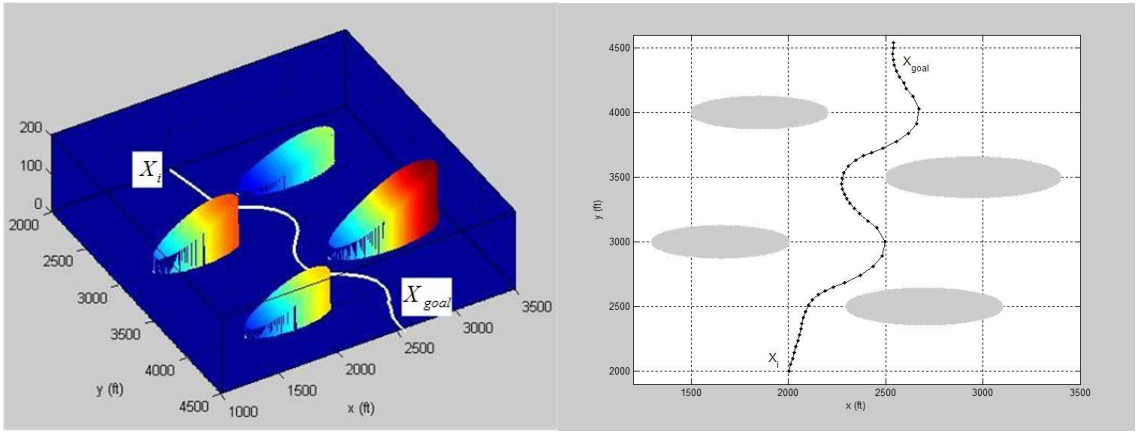


Figure 5.52 Estimated fire ellipses and dynamically generated path using potential fields for the human firefighter to go from X_i to rescue location X_{goal}

In this Chapter were presented extensive simulation results to validate the algorithms proposed in Chapters 3 and 4. We progressively increased the complexity of the field distribution from simple spatial-stationary fields represented by only a few

parameters, to a fairly complex spatio-temporal field represented with a RBF-NN and hundreds of parameters. In each case, convincing evidence was presented to support the conclusion that the EKF-based adaptive sampling algorithms using the GAS heuristic and a RBF-NN approximation perform more efficiently than a simple raster scan, and that the robot uncertainty can be reduced by sampling.

CHAPTER 6

EXPERIMENTAL RESULTS

This Chapter discusses experimental sampling results obtained using a mobile wireless testbed located at ARRI's DIAL (Distributed Intelligence and Autonomy Lab). It includes a brief description of the ARRI-Bot hardware, its localization and navigation, distributed parametric field and its sensing and finally experimental results validating the proposed adaptive sampling algorithms.

6.1 Description of testbed

Team members at the Distributed Intelligence and Autonomy Lab (DIAL) at UTA's Automation & Robotics Research Institute (ARRI) validated the algorithms proposed in Chapter 3 on mobile wireless sensor nodes (MWSN). These nodes can be thought of as robotic platforms delivering a sensor pack to a location of interest in order to take samples, and transmit sensor information via radio signals through a wireless network. A first version of the sampling setup is shown in Figure 6.1. In it, an inexpensive wide-angle overhead camera system is used to estimate the true robot location as an in-door GPS, and is used to compare it with the estimated robot location via the EKF.

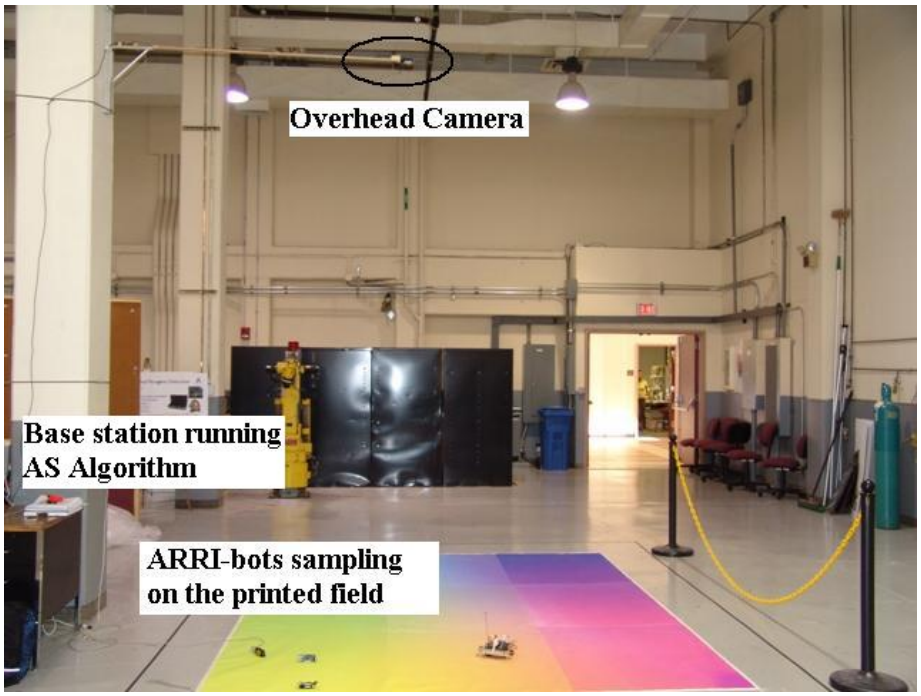


Figure 6.1 Illustration of the experimental setup being used to validate the AS algorithm

6.1.1 Color field

Within this initial testbed, a simple linear field was printed on the lab floor. The field was physically distributed over a search space of 3.15 m x 2.25m. The camera at a height of 10.5 m encompasses the entire area in its field of view. Three separate linear field models, one for each primary color (RGB), are used to generate a color field, with numerical values:

$$R = r_0 + r_1 x + r_2 y, G = g_0 + g_1 x + g_2 y, B = b_0 + b_1 x + b_2 y \ , \tag{6.1}$$

$$r_0 = 0.2307, r_1 = 0.0012, r_2 = -0.00048, g_0 = 0, g_1 = 0.0002,$$
$$g_2 = 0.0018, b_0 = 1.0, \ b_1 = -0.00078, b_2 = -0.001$$

The sampling mission is for mobile robots equipped with a color sensor to recover the unknown (to the robot) field coefficients by sampling.

6.1.2 ARRI-Bots

ARRI-Bots are inexpensive MWSN units built at DIAL that are depicted in the schematic diagram in Figures 6.2-6.3. The ARRIBot energy harvester consists of a solar panel and associated electronics for outdoor operation and a piezoelectric energy harvester that augments an on-board Ni-CAD battery pack.

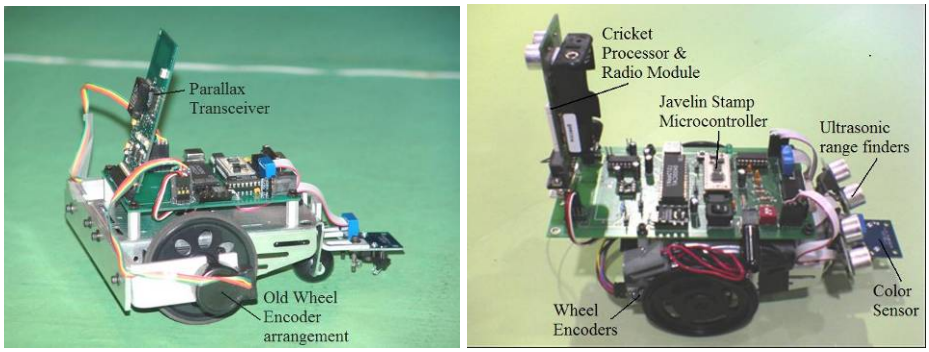


Figure 6.2 Inexpensive ARRI-Bots V-1 (left) and V-2 (right)

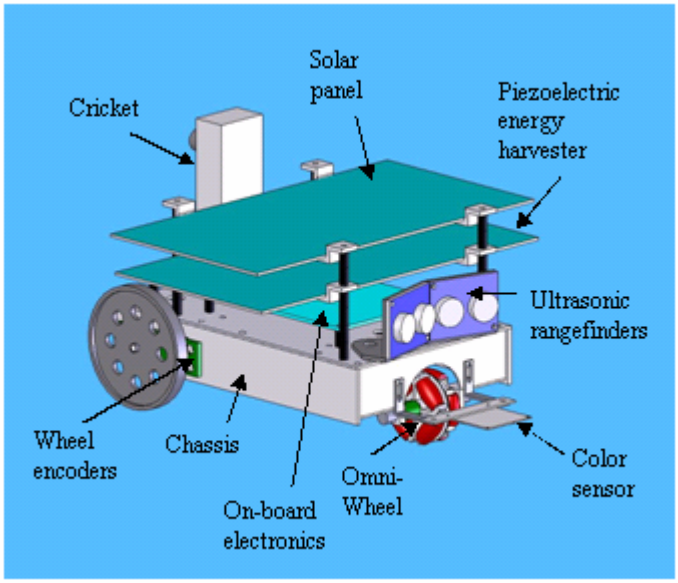


Figure 6.3 Schematic diagram of the ARRI-Bot V-2

The subsystems of the ARRI-Bot include:

(i) Mechanical subsystem, consisting of an aluminum frame chassis, a differential drive and a front omni-directional wheel.

(ii) On-board sensor systems, consisting of wheel encoders for dead-reckoning localization, ultrasonic rangefinders for obstacle detection, and a color sensor for detecting color printed on the lab floor.

(iii) Wireless sensor node, consisting of a Parallax Transceiver (for ARRI-Bot V-1) or a Cricket® processor/radio module (for ARRI-Bot V-2), operating in the 433 MHz frequency band. Each Mote is equipped with a default sensor pack (light, temperature, etc). The Cricket unit carries ultrasonic transducers for range finding and is used for both communication and localization.

(iv) ARRI-Bot electronics, consisting of a custom-made PCB board containing Javelin stamp CPU, and glue electronics for interfacing to all the sensor and wireless subsystems. In addition, the electronics board contains power management and energy harvesting circuitry.

(v) The ARRI-Bot power system, consisting of a Ni-MH battery pack, a solar panel for out-door recharging and operation, and a piezoelectric cantilever array for battery augmentation.

The intelligence of ARRI-Bot to carry out various maneuvers is regulated by on- board processing unit, a *Javelin Stamp* micro-controller. The motivation behind using a simple Javelin Stamp was to demonstrate that our AS algorithms can be implemented on a cost effective, and not very powerful computing platform. Cost-

effectiveness considerations are important for deploying large numbers of simple robot nodes to sample the environment. For the work presented in this Thesis, our original goal was only partially accomplished, since the EKF field calculations were performed on a base-station PC running MATLAB. However, at the time this dissertation was written, many inexpensive mobile robot units, with more powerful processors have become available on the market, and they have a similar price target as the ARRI-Bot V-2. In the future, the AS algorithms should be validated on such hardware, but it is clear that the results can only improve beyond the results presented here.

ARRI-Bot maneuvers are accomplished by means of easy, structured and object oriented Java based programming. Further detail about the design of the ARRI-bot can be found in the MS thesis of DIAL team members Sreenath [141] and Ghadigaonkar [142]. Figure 6.4 shows a screen capture of the Javelin programming interface environment, while Figure 6.5 depicts the list of simple wireless commands used to position the ARRI-Bots.

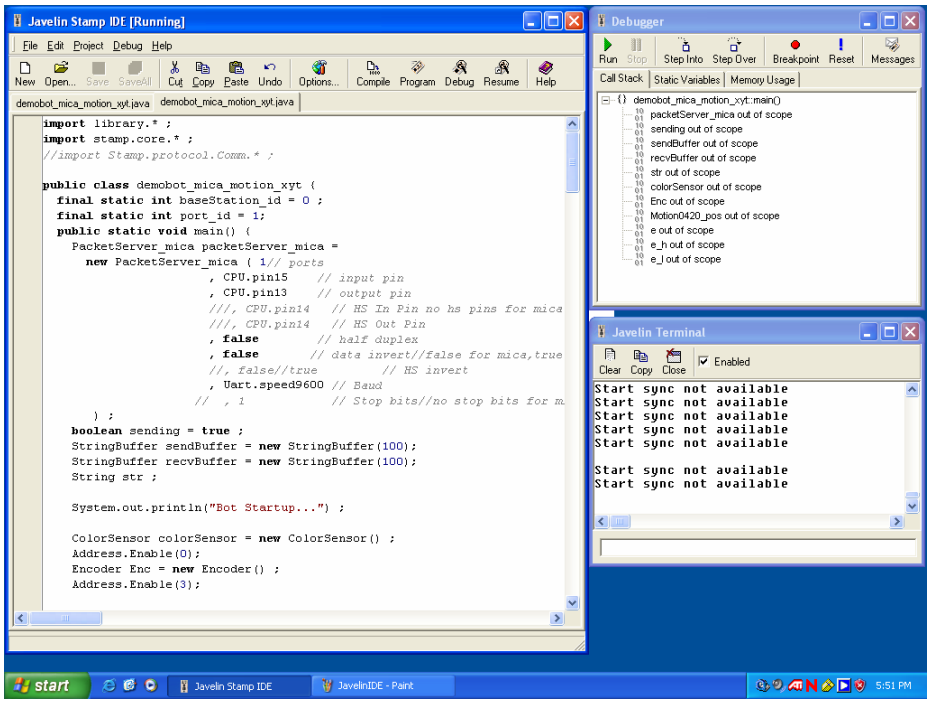


Figure 6.4 Snapshot of Javelin Stamp IDE

Commands	Interpretation
'F'	Move forward infinitely
'M250'	Move forward by the specified number of encoder values for e.g. move for 250 encoder value, where 0< encoder value≤32767
'B'	Move Backward infinitely
'M-250'	Move Backward by the specified of encoder value e.g. move backward for 250 encoder value, where -32767≤enc value<0
'S'	Stop
'L'	Turn Left infinitely
'T20'	Turn Left for the specified encoder value 0< encoder value≤32767
'R'	Turn Right infinitely
'T-20'	Turn Right for the specified encoder value -32767≤encoder value<0
'C'	Take a color sample
'J'	Switch the crickets mode from radio to Beacon for 15 sec and then back to radio
'Z'	Switch the crickets mode from radio to listener for 10 sec and then back to radio

Figure 6.5 Simple robot commands for ARRI-Bot

6.1.3 Robot model and dead-reckoning location estimation

A differential kinematic robot model with uncertainty in wheel radii and unequal wheel base was used in conjunction with a dead-reckoning localization algorithm. In our simulations presented in Chapter 5, a systematic error is injected into the system to account for navigational errors that arise due to inaccuracies in construction and mechanical assembly, ad depicted in Figure 6.6.

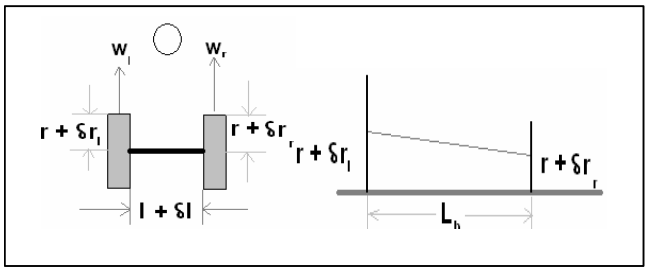


Figure 6.6 Differential-drive mobile robot with uncertainty in wheel radii and axle length

The robot runs an onboard position estimator using encoder data and nominal physical dimensions (lengths, wheel radii, etc) by dead-reckoning. The states of the robot are the position and orientation $X = \begin{bmatrix} x & y & \theta \end{bmatrix}^T$. A discrete time position estimator using the encoder counts $\Delta\phi_L, \Delta\phi_R$ is shown in equation (6.2). K_{drv}, K_{turn} are the actuator drive constants for either actuator in terms of distance, turn per drive count:

$$\begin{aligned}
\hat{X}_1(k) &= \hat{X}_1(k-1) + \frac{K_{drv-R}\Delta\phi_R r_R + K_{drv-L}\Delta\phi_L r_L}{2}\cos\left(\hat{X}_3(k-1)\right) \\
\hat{X}_2(k) &= \hat{X}_2(k-1) + \frac{K_{drv-R}\Delta\phi_R r_R + K_{drv-L}\Delta\phi_L r_L}{2}\sin\left(\hat{X}_3(k-1)\right) \\
\hat{X}_3(k) &= \hat{X}_3(k-1) + \frac{K_{turn-R}\Delta\phi_R r_R - K_{turn-L}\Delta\phi_L r_L}{L_b} \\
X &= [\hat{X}_1 \;\; \hat{X}_2 \;\; \hat{X}_3] = [x \;\; y \;\; \theta]
\end{aligned} \tag{6.2}$$

To implement (6.2), a series of Monte-Carlo experiments were performed to find the constants K_m and K_t in equations (6.3) and (6.4). The ARRI-Bot is a non-holonomic robot. DIAL team member Talati [156] considered path planning navigation algorithms for this robot, but his planner was not used in our work. Instead, we used a "quasi-holonomic" path planner to validate the AS algorithms. With this planner, navigating the wheeled mobile robot from (x_k, y_k) to (x_{k+1}, y_{k+1}) involves a *turn* (e.g. pointing the robot to the target) and a *move* (e.g. moving the robot in a straight line to the target). Both of these commands involve simple $(\Delta\phi_t, \Delta\phi_m)$ encoder turns.

6.1.3.1 Servo Motors & Optical Encoder

Futaba S148 RC servo motors are used on ARRI-bots. The servo module has a built in motor, gearbox and the controlling electronics. The direction and speed of the motors is controlled through PWM pulses generated by a PIC 12F508 micro-controller.

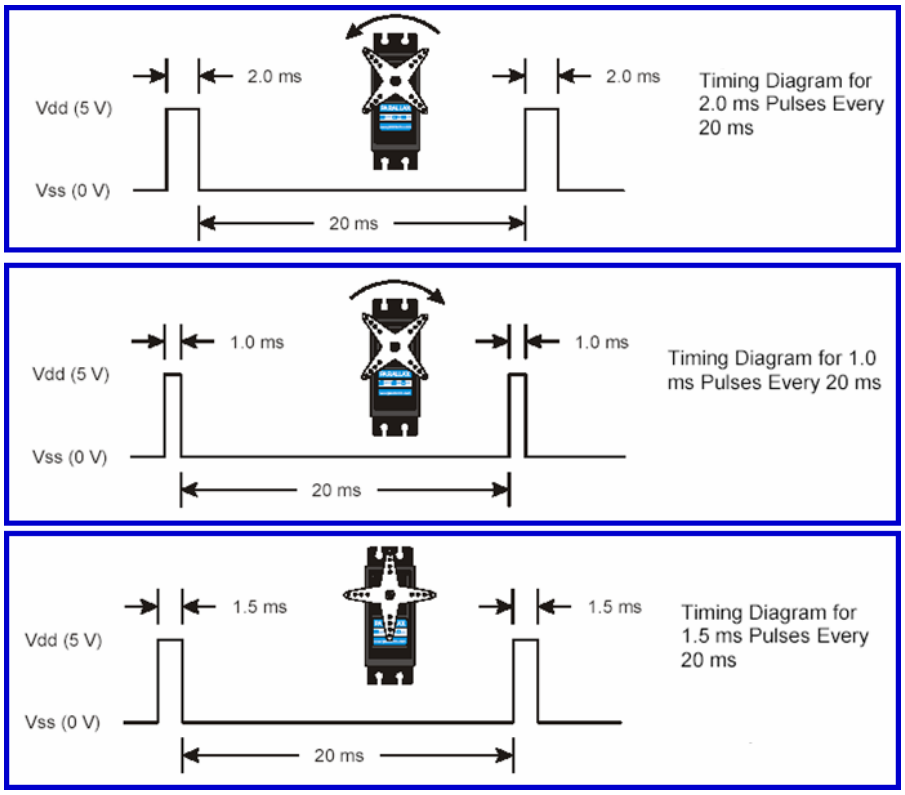


Figure 6.7 RC servo motors PWM signals [139]

The pulse width determines the direction and speed of the motor, as shown in Figure 6.7. In order for the robot to move straight, the speed of both motors should be identical. For negotiating a turn in place maneuver, the speed of both motors should be equal, but in opposite directions. Ideally, a pulse width of 1.5 ms stops the motors, and similarly, a pulse width of 1.2 ms should run both motors at the same speed. However, due to the low cost of the hardware, after continuous use, the motors start drifting. An auto-calibration code was written to apply different PWM signals to both wheels and check for the time it takes for the motors to reach certain distance. The program then compares the two times to adjust appropriate PWM widths for both motors. This procedure is illustrated in Figure 6.8. Furthermore, the Javelin-Stamp performs only

fixed-point calculations for localization while navigating, making the implementation of closed-loop control and KF estimation more difficult to program.

Wheel encoders (WW-01) from Nubotics were mounted on the servos to measure the angular displacement of the motors. In comparison to the old encoders 600EN-128-CBL on the first generation ARRI-Bot V-1, the WW-01 units are easy to mount, can be directly interfaced with the processor and are less expensive. Highly precise maneuverability of resolution less than 3 mm was achieved through the use of the servo motors and optical encoders. The list below summarizes some of the specifications for the ARRI-Bot V-2 hardware:

- Encoder Resolution = 64 pulses per rotation
- Wheel Diameter = 2.75 inches
- Wheel Circumference = 8.639 inches
- Distance traveled in one rotation ~ 21.9 cm
- Resolution Achieved ~ 3mm

By amplifying the encoder resolution to 128 pulses per rotation, a small motion up to 1.7 mm can be captured.

		Timer Value	Time (in sec)	Gain for left Wheel						Timer Value	Time (in sec)	Gain for Right Wheel			
	BW	675876	5.9	131	675745	5.9	131		FW	698642	6.1	131	700974	6.1	131
		679734	5.9	132	675814	5.9	132			698698	6.1	132	701055	6.1	132
		679585	5.9	133	675806	5.9	133			698690	6.1	133	700957	6.1	133
		679709	5.9	134	675857	5.9	134			698590	6.1	134	705250	6.1	134
		679629	5.9	135	679558	5.9	135			706154	6.1	135	709492	6.2	135
		683429	5.9	136	687181	6.0	136			710034	6.2	136	717574	6.2	136
Increasing Speed		691061	6.0	137	690965	6.0	137			655793	5.7	137	660226	5.7	137
▲		694782	6.0	138	698521	6.1	138			728898	6.3	138	734031	6.4	138
▲		706153	6.1	139	709933	6.2	139			744078	6.5	139	750507	6.5	139
▲		655745	5.7	140	655814	5.7	140			763046	6.6	140	771006	6.7	140
▲		740310	6.4	141	744141	6.5	141			723938	6.3	141	734338	6.4	141
▲		766833	6.7	142	770661	6.7	142			831238	7.2	142	845074	7.3	142
▲		746714	6.5	143	750497	6.5	143			830198	7.2	143	849669	7.4	143
▲		811173	7.0	144	822553	7.1	144			870589	7.6	144	895532	7.8	144
▲		862397	7.5	145	870557	7.6	145			990854	8.6	145	1019346	8.8	145
▲		1001989	8.7	146	1017310	8.8	146			1204739	10.5	146	1196824	10.4	146
▲		1207112	10.5	147	1233489	10.7	147			1553138	13.5	147	1611829	14.0	147
▲		1590727	13.8	148	1631431	14.2	148			2358151	20.5	148	2445430	21.2	148
		-	-	149	-	-	149			-	-	149	-	-	149
Stop		-	-	150	-	-	150			-	-	150	-	-	150
		-	-	151	-	-	151			-	-	151	-	-	151
▼	FW	2558130	22.2	152	2682331	23.3	152		BW	6500174	56.4	152	7295582	63.3	152
▼		6528506	56.7	153	6682970	58.0	153			3610595	31.3	153	3848894	33.4	153
▼		2304978	20.0	154	2316534	20.1	154			1872009	16.2	154	1890647	16.4	154
▼		1534915	13.3	155	1546742	13.4	155			1365097	11.8	155	1374494	11.9	155
▼		1132457	9.8	156	1140438	9.9	156			1086629	9.4	156	1095281	9.5	156
▼		969971	8.4	157	977615	8.5	157			931763	8.1	157	945179	8.2	157
▼		846151	7.3	158	850130	7.4	158			881121	7.6	158	890094	7.7	158
▼		741850	6.4	159	745830	6.5	159			842429	7.3	159	790757	6.9	159
▼		784033	6.8	160	722442	6.3	160			753562	6.5	160	830993	7.2	160
▼		699158	6.1	161	768451	6.7	161			737990	6.4	161	748433	6.5	161
▼		687330	6.0	162	691346	6.0	162			726390	6.3	162	735725	6.4	162
▼		679766	5.9	163	683550	5.9	163			784123	6.8	163	727249	6.3	163
Increasing Speed		675618	5.9	164	679518	5.9	164			710825	6.2	164	784257	6.8	164
		671839	5.8	165	675495	5.9	165			710785	6.2	165	714465	6.2	165
		668039	5.8	166	671858	5.8	166			714660	6.2	166	718703	6.2	166
		668266	5.8	167	671954	5.8	167			710794	6.2	167	714598	6.2	167
		668071	5.8	168	671859	5.8	168			706933	6.1	168	710183	6.2	168
		667883	5.8	169	668154	5.8	169			706966	6.1	169	710222	6.2	169
		667936	5.8	170	671674	5.8	170			706933	6.1	170	714579	6.2	170

Figure 6.8 Left and right wheels motor calibration

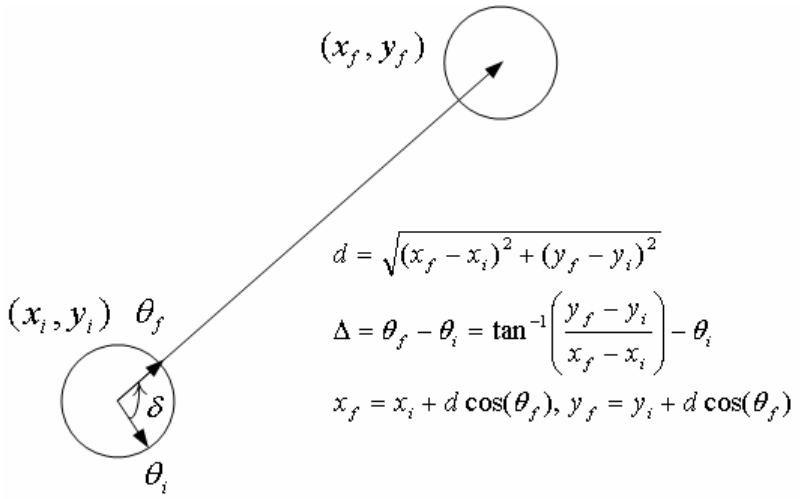


Figure 6.9 Differential drive robot's navigation from $(x_k, y_k) = (x_i, y_i)$ to $(x_{k+1}, y_{k+1}) = (x_f, y_f)$

In order to navigate with the differential drive ARRI-Bot, a simple point and turn maneuver was implemented, as depicted in Figure 6.9. This maneuver consists of a "turn in place" and point to target, and a "forward move" to the target. To achieve left/right turn in place, the robot states evolve according to:

$$\begin{aligned} x_{k+1} &= x_k \\ y_{k+1} &= y_k \\ \theta_{k+1} &= \theta_k + K_t \Delta\phi_t \\ \Delta\phi_t &= \frac{\delta}{K_t} \end{aligned} \quad , \tag{6.3}$$

where K_t is the angle turn per encoder count (left and right encoder counts are equal in magnitude but with opposite sign), $\Delta\phi_t$ is change in encoder counts and δ is the in angle turned.

Similarly, to accomplish a "forward" move, the robot states evolve according to:

$$
\begin{aligned}
x_{k+1} &= x_k + K_m \Delta\phi_m \cos(\theta_k) \\
y_{k+1} &= y_k + K_m \Delta\phi_m \sin(\theta_k) \\
\theta_{k+1} &= \theta_k \\
\Delta\phi_m &= \frac{d}{K_m}
\end{aligned}
\quad , \tag{6.4}
$$

where K_m is the distance covered per encoder count (left and right encoder counts are the same).

6.1.3.2 Accuracy measure of DR location

The UMBmark test [80, 112], which is a bi-directional square path test, was performed for measuring the dead-reckoning error of the ARRI-Bot. The results are shown in Figure 6.10 and they clearly indicate that the true robot position deviates more from the desired path than the dead-reckoning position. The former involve both systematic as well non-systematic influences, while later can only capture systematic errors.

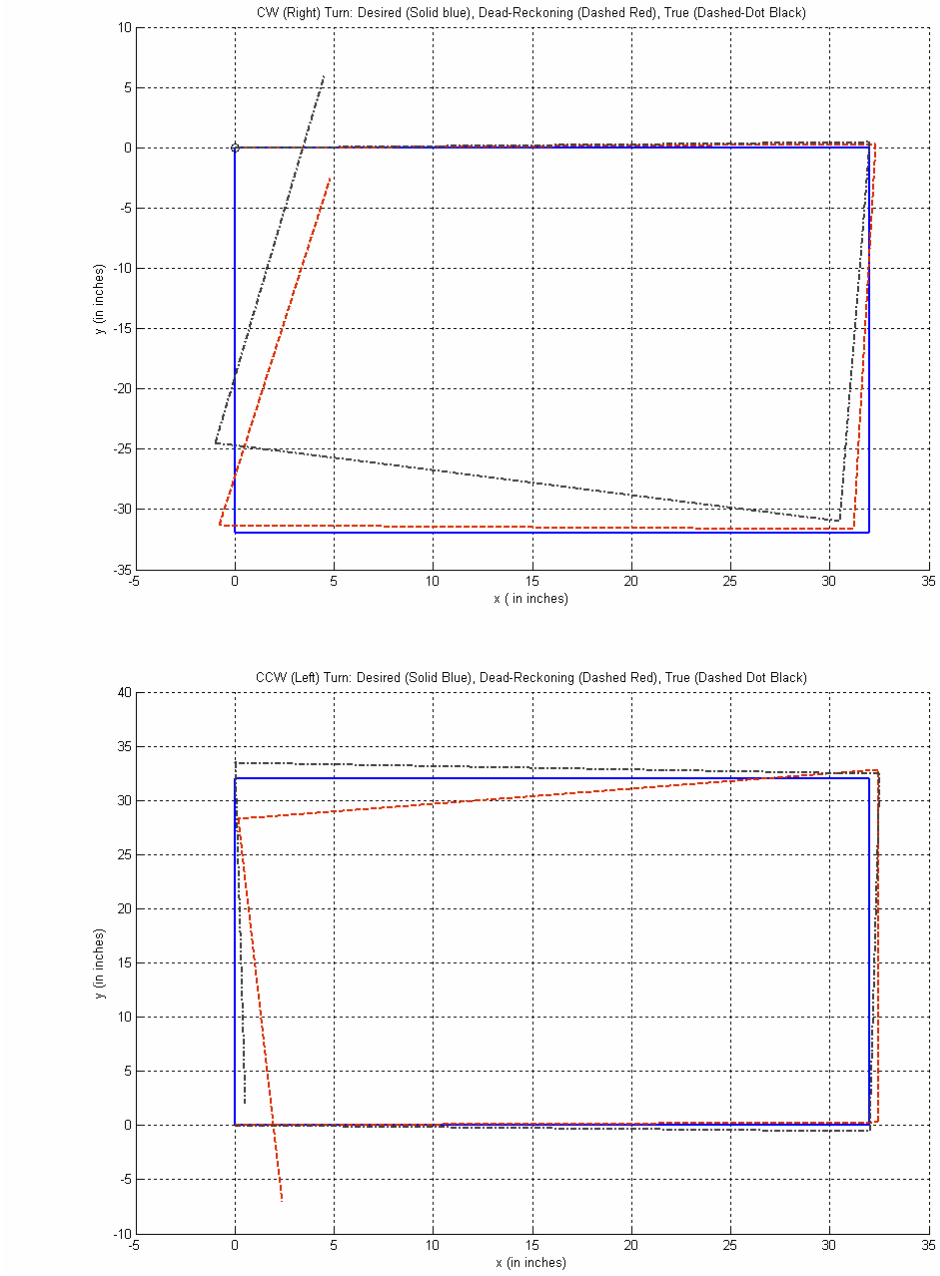


Figure 6.10 Results of UMBMark (square path) test for measuring dead-reckoning error. Box of 32x32 square inches along which robot navigates in clockwise (top) and counter-clockwise (bottom) direction

6.1.3.3 Fixed-point calculations

Since the Javelin stamp processor doesn't support floating point operations, a fixed-point math library was written to perform 32-bit addition, subtraction multiplication, division and trigonometric operations. This makes the implementation of calculation such as (6.3) and (6.4) possible. However, in the experimental results reported in the next section, the AS EKF field estimation calculations run in MATLAB on the ARRI-Bot base-station and not on the robot.

6.1.4 Color sensor measurement

ARRI-Bot first and second generations are equipped with a TAOS TCS230 color sensor used to validate the adaptive sampling algorithm based on the color measurement obtained from a linear parametric field, printed color field on the floor. The color sensor module is comprised of a color detector including a RGB sensor chip, white LED illuminator and a collimator lens (Figure 6.11). It has an array of photo detectors each with a red, green, or blue filter or no filter (clear). The filters are distributed evenly throughout the array to eliminate location bias among the colors.

Figure 6.11 TAOS Color Sensor

The proportion of R, G, and B at a sampled point varies from 0 to 255, which are then normalized to values between 0 and 1. A series of Monte Carlo experiments were performed to find the repeatability, accuracy and resolution errors of color sensor measurement at different locations on the printed color field with slightly varying ambient light to decide for appropriate value of measurement error covariance *R*. These steps were taken also to minimize the discrepancy in the printed color field. A reasonable error of 10% was reached in all experiments for RGB measurement values.

6.1.5 Absolute localization using over-head camera

The overhead camera system serves as a positioning system with low sampling rate in order to correct dead-reckoning navigation errors. Unlike the dead-reckoning error, localization using vision doesn't grow with time.

Robot localization using vision is a simple object recognition problem for the lab floor image. MATLAB Image Acquisition & Image Processing toolboxes running on a base-station computer are used to process the camera images to get correct robots' position and orientation ("surrogate GPS"). The orientation of the robots can be estimated by using the robot triangular shape, and calculating properties such as the centroid, major axis, minor axis and area of the object, and finding the farthest point from the centroid that lies on the object. A rectangular grid array was used to correct the wide-angle lens distortion, and a GUI shown in Figure 6.12 was used for manual wireless control of the rovers, as well as field estimation calculations and display. The characteristics of the image processing algorithm for our testebed were:

- Image Resolution=3.64 pixels/inch

- Accuracy of robot localization (~1”,~1”,<5°)
- Repeatability of robot localization (<0.1”,<0.1”,<1°)

The following image processing steps are taken after the subtracted image is acquired using a custom MATLAB code written for localizing ARRI-Bots:

- ‘Filling’ to convert the robots pixels to solid objects.
- ‘Boundary Tracing’ for the objects.
- Ignoring objects other than the robot objects by reading ‘MajorAxisLength’, ‘MinorAxisLength’ and ‘Area’ of the objects.
- Calculate the ‘Centroid’ of the objects.
- Finding the farthest point on the robot object from the centroid
- Finding the robots orientations based on the line joining the centroid and the farthest point.
- Acquire the position and orientation of the robot in pixel co-ordinates system.
- Convert the position and orientation in pixel coordinate system to world coordinate system by incorporating the grid distortion factors.

Results of image processing with our code are shown in Figures 6.12 and 6.13.

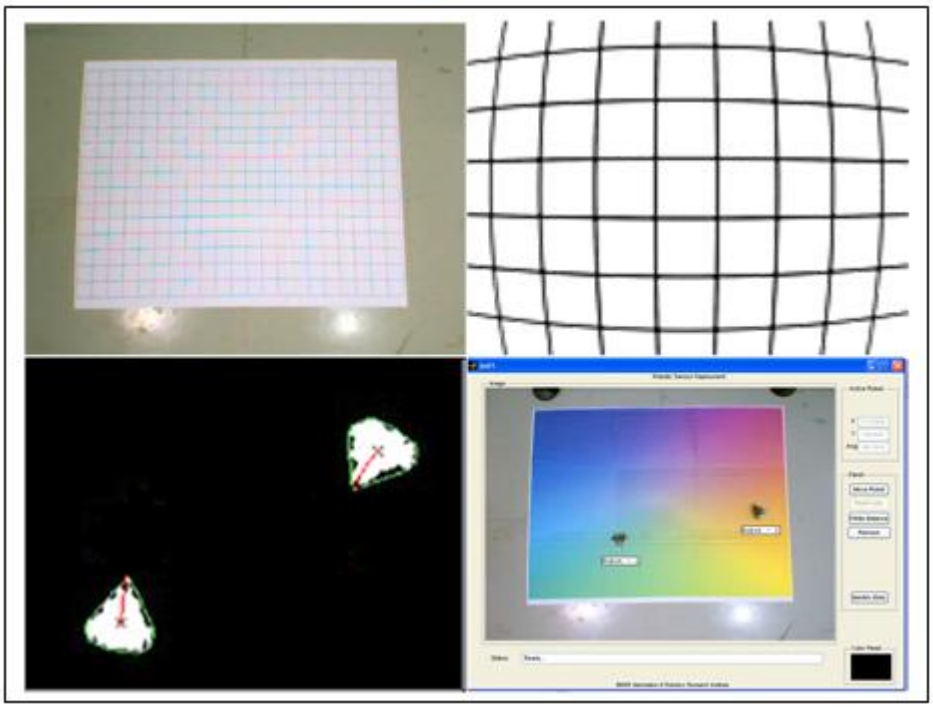


Figure 6.12 Grid (top left), Distortion in the grid image (top right), Image Segmentation as seen from the MATLAB image (bottom left) and MATLAB GUI (bottom right) to determine robot position and orientation

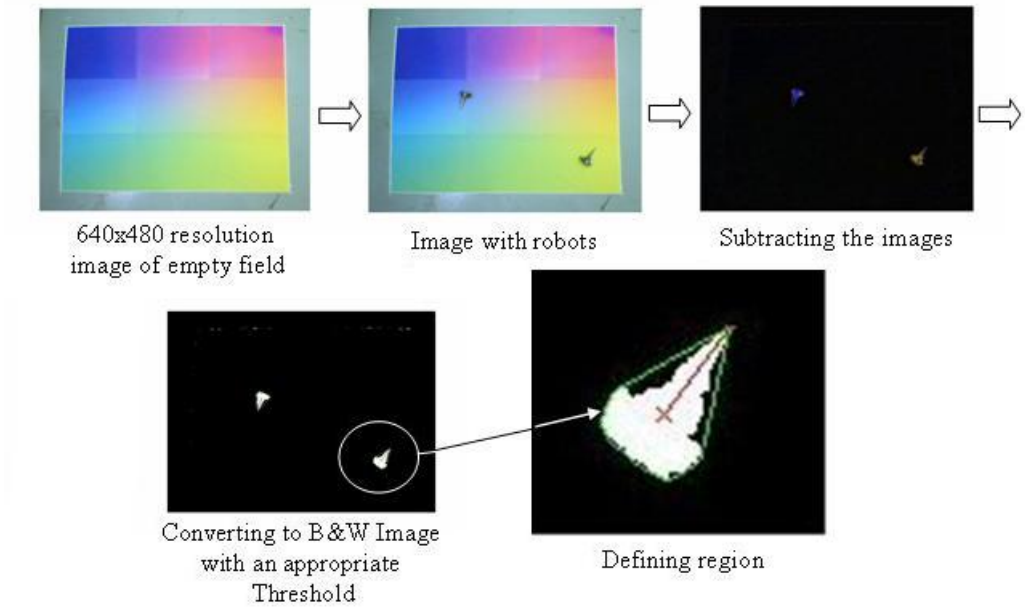


Figure 6.13 Steps involved in image processing for robot localization

6.1.6 Absolute localization using Cricket beacons

The most common way to use Cricket® Motes is to deploy actively transmitting beacons on walls or ceilings and attach listener to host devices whose location needs to be obtained. In the case of our testbed, the Crickets are mounted vertically on the robots as shown in Figure 6.14 and the robot that needs to be localized is in "listener" mode. Each beacon periodically broadcasts its space identifier and position coordinates on a RF frequency channel which listener within radio range can receive. Each beacons also broadcast an ultrasonic pulse at the same time as the RF message. Listeners that have line of sight connectivity to the beacon and are within the ultrasonic range receive this pulse. Because RF travels about 10^6 times faster than ultrasound, the listener can use the time difference of arrival between the start of the RF message from a beacon and the corresponding ultrasonic pulse to infer its distance from the beacon. If the listener robot can get this distance from three static nodes with known location it can easily triangulate to find its relative location.

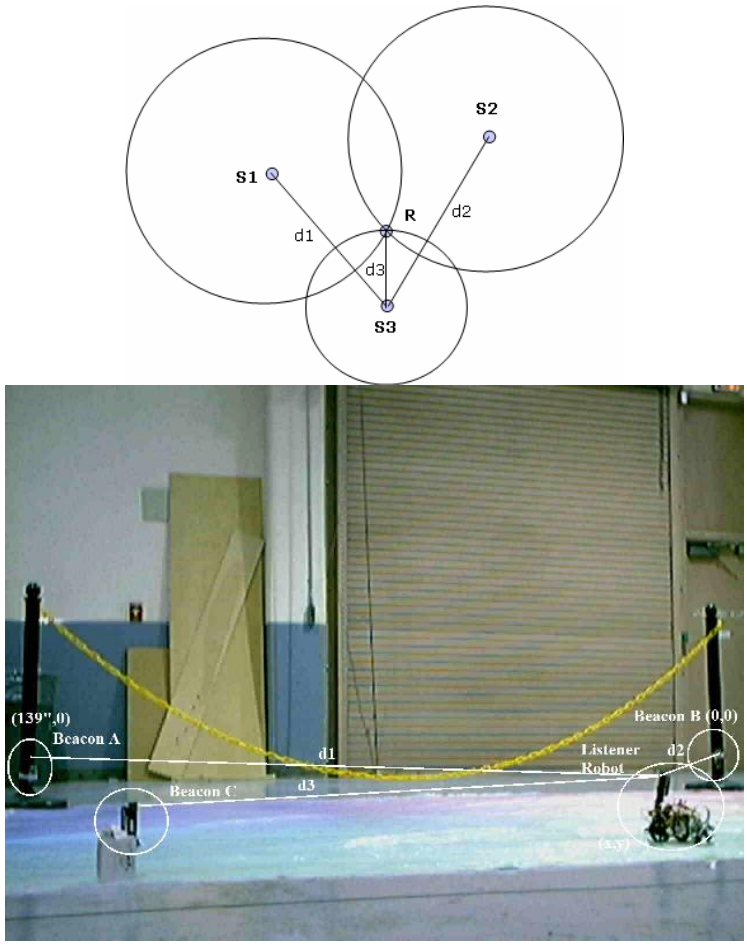


Figure 6.14 Robot localization from static beacons by triangulation

For an output power of 9 dbm a linear increase in error is observed in the actual distance and the measured distance from cricket, which is compensated through a simple correction equation: $d_{actual} = 1.1141.d_{measured} - 2.1150$, plotted in Figure 6.15.

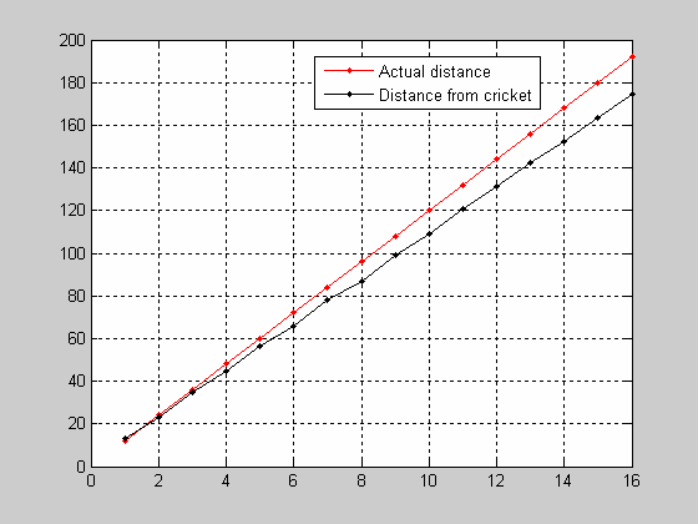


Figure 6.15 Difference in actual distance and measured distance for localization by triangulation

6.1.7 Camera-projector system

In order to test the sampling algorithms for complex, time-varying, non-linear fields, the printed color field on the lab floor is no longer appropriate, and instead we installed a camera-projector system at DIAL, as shown in Figure 6.16. The hardware includes a projector mounted 22 ft high on the ceiling. The projected image size is 124 x 93 inches and the image resolution is 1024 x 768. The projector is of 3000 ANSI lumen brightness so that it can perform satisfactorily in bright light in the lab. The projector is connected to the base-station which runs MATLAB to generate any complex field that can be projected on the lab floor.

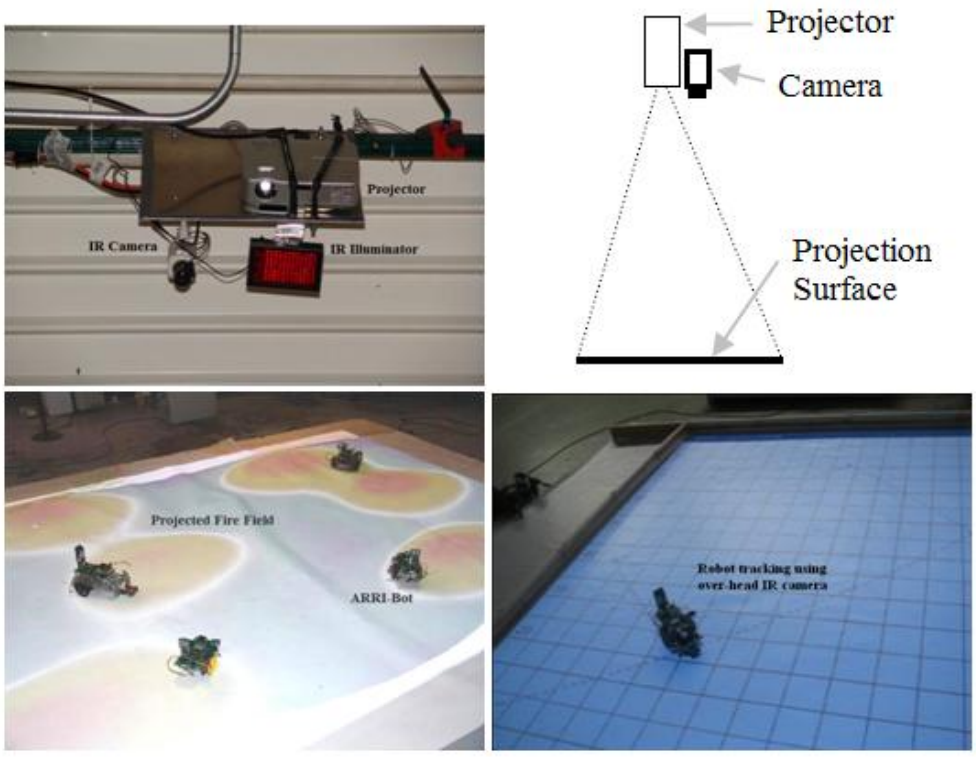


Figure 6.16 Testbed with simulated fire field projected on the floor from a projector. ARRI-Bots are shown sampling at various locations for estimating field parameters. Over-head IR camera is used to aid in localization and for validating the accuracy of estimated location

6.2 Experimental validation of AS algorithms for linear parametric field

In a first set of experiments, the ARRI-Bot takes color sensor measurements and sends it to the base-station computer, which runs adaptive sampling algorithms. The algorithm uses the color sensor and dead-reckoning position measurement from the ARRI-Bot, and location information from the overhead camera, in order to run the EKF estimator. The adaptive sampling algorithm then find the next sampling location based on the minimum variance criterion with the 2-norm, as described in Chapter 3. The flow chart of the robot operation is shown in Figures 6.17 and 6.18.

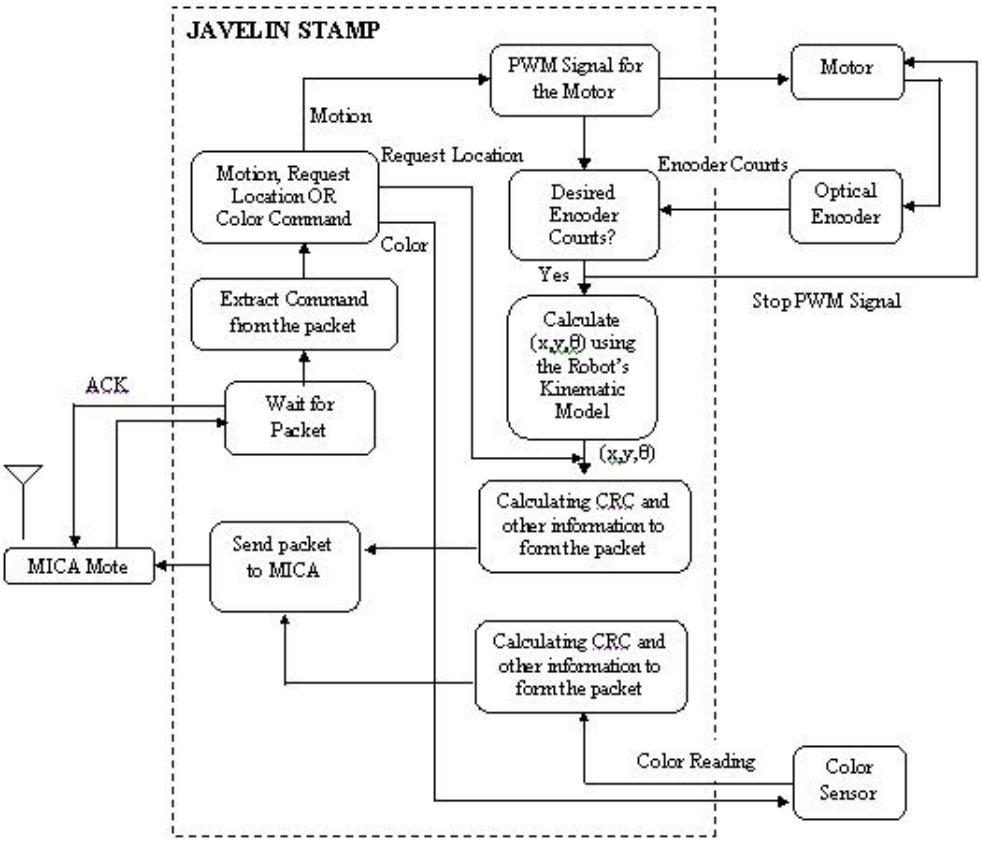


Figure 6.17 Flow of operations performed on ARRI-Bot

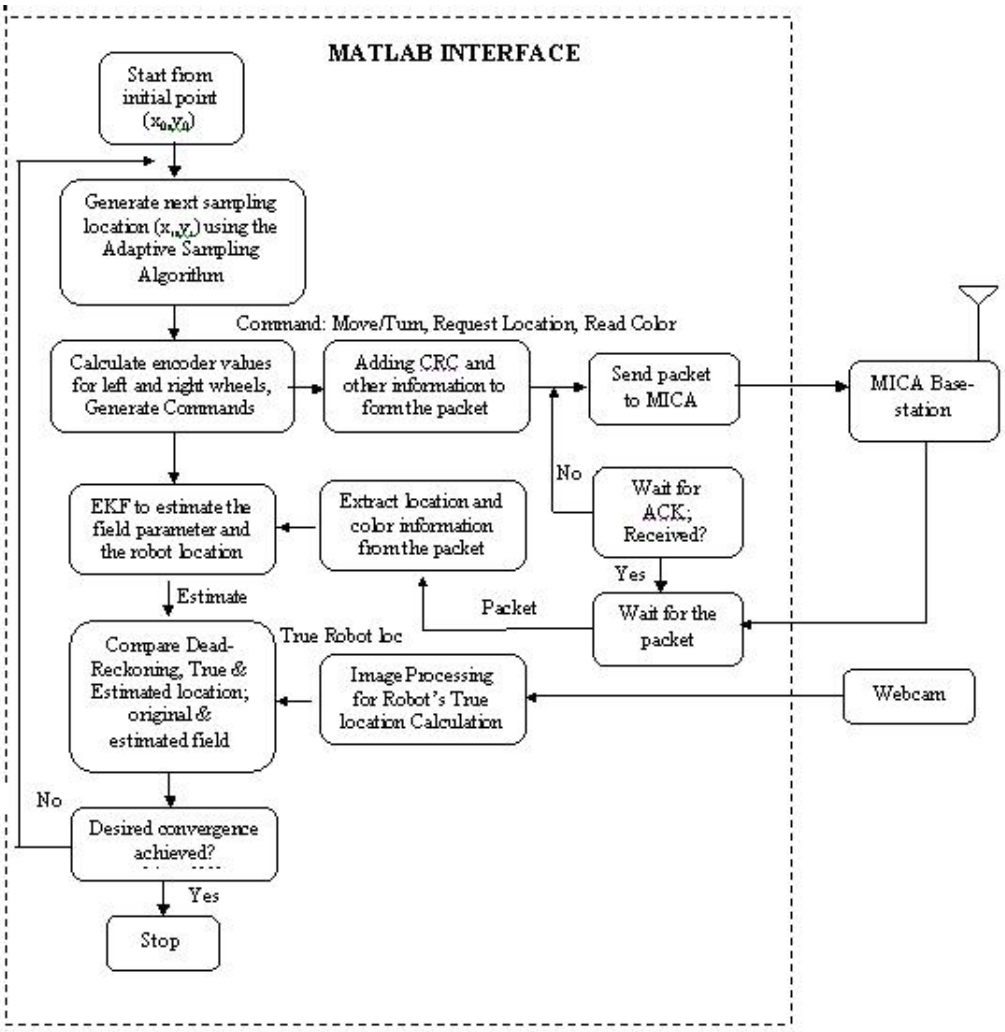


Figure 6.18 Flow of EKF-AS algorithm running on the base-station

6.2.1 KF estimation for linear color field with no uncertainty in localization through camera

Here we present the experimental validation of the case discussed in Section (3.2.2), when uncertainty in camera measurement is ignored. In this case, we are recovering a linear color field with 9 parameters, therefore $X_1 = x, X_2 = y, g_1(X_1) = x, g_2(X_2) = y$. The base station runs the adaptive sampling algorithm and commands the mobile sensor to sample at a particular location. The

mobile sensor navigates to that location by dead reckoning and thus will end up at a location different than the commanded one. The process and measurement model and updates equations of the sampling algorithm are:

- System Model:

$$A_{k+1} = A_k = \begin{bmatrix} r_0 & r_1 & r_2 & g_0 & g_1 & g_2 & b_0 & b_1 & b_2 \end{bmatrix}^T \tag{6.5}$$

- Measurement Model:

$$\begin{aligned} \tilde{z}_k &= h_k(A_k) + v_k \\ \begin{bmatrix} \tilde{r}_k \\ \tilde{g}_k \\ \tilde{b}_k \end{bmatrix} &= \begin{bmatrix} r_{0k} + r_{1k}x + r_{2k}y \\ g_{0k} + g_{1k}x + g_{2k}y \\ b_{0k} + b_{1k}x + b_{2k}y \end{bmatrix} + v_k \end{aligned} \tag{6.6}$$

$$Q = 0,\ v_k \sim N(0, R),\ x_0 \sim N(\bar{x}_0, P_0)$$

- Effect of System Dynamics:

$$\begin{aligned} \hat{A}_{k+1}^- &= \hat{A}_k \\ P_{k+1}^- &= P_k \end{aligned} \tag{6.7}$$

- Effect of Measurement:

$$H_{k+1} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x & y & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & y \end{bmatrix}$$

$$K_{k+1} = P_{k+1}^- H_{k+1}^T (H_{k+1} P_{k+1}^- H_{k+1}^T + R_{k+1})^{-1}$$

$$P_{k+1} = (I_{12} - K_{k+1} H_{k+1}) P_{k+1}^-$$

$$\hat{A}_{k+1} = \hat{A}_{k+1}^- + K_{k+1}\left\{\tilde{z}_{k+1} - \hat{h}_{k+1}\right\} = \hat{A}_{k+1}^- + K_{k+1}\left\{\begin{bmatrix} \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{r}_{0k+1}^- + \hat{r}_{1k+1}^- x + \hat{r}_{2k+1}^- y \\ \hat{g}_{0k+1}^- + \hat{g}_{1k+1}^- x + \hat{g}_{2k+1}^- y \\ \hat{b}_{0k+1}^- + \hat{b}_{1k+1}^- x + \hat{b}_{2k+1}^- y \end{bmatrix}\right\} \tag{6.8}$$

6.2.1.1 Simulation based on experimental setup

We first performed a simulation using similar uncertainty values as in the experimental setup. We assumed the following true values for the unknown field parameters, as they were used to print the lab floor image:

$$R = r_0 + r_1 x + r_2 y, G = g_0 + g_1 x + g_2 y, B = b_0 + b_1 x + b_2 y \tag{6.9}$$

$$r_0 = 0.2307, r_1 = 0.0012, r_2 = -0.00048, g_0 = 0, g_1 = 0.0002,$$
$$g_2 = 0.0018, b_0 = 1.0, b_1 = -0.00078, b_2 = -0.001$$

In Figures 6.19 and 6.20, a comparison of Raster-Scan sampling and AS is presented. The simulation runs until the 2-norm of error in coefficients reduces to small values. The simulations also show the convergence of all the parameters to their desired value. Initially it was assumed that the robot samples in a grid of 1 sq inch (6.55 cm^2) which leads to a search space of 126 x 90 points for the AS algorithm. In order to decrease the run time for the algorithm, the grid size was increased to 91.7 cm^2, and only the center of the cell was considered for sampling.

By using dead-reckoning, the error in position and orientation for the robot increases as the robot travels more distance. Furthermore, the AS simulation results in the robot taking long-jumps, thus increasing sampling time. As a result, we also compare the results with a simple Greedy adaptive sampling strategy in Figure 6.21. This scheme has a 1-step horizon search for the next sampling location.

The above three algorithms are compared in the Table 6.1 with respect to estimation error in the unknown coefficients below 0.2. It can be seen that Greedy Adaptive Sampling lies somewhere in between raster scan and simple adaptive

sampling in terms of number of samples, but takes the least time to reconstruct the field. As expected, the standard Adaptive Sampling algorithm takes the least number of samples. At the start of all simulations, the EKF assumes zero values for R, G, and B.

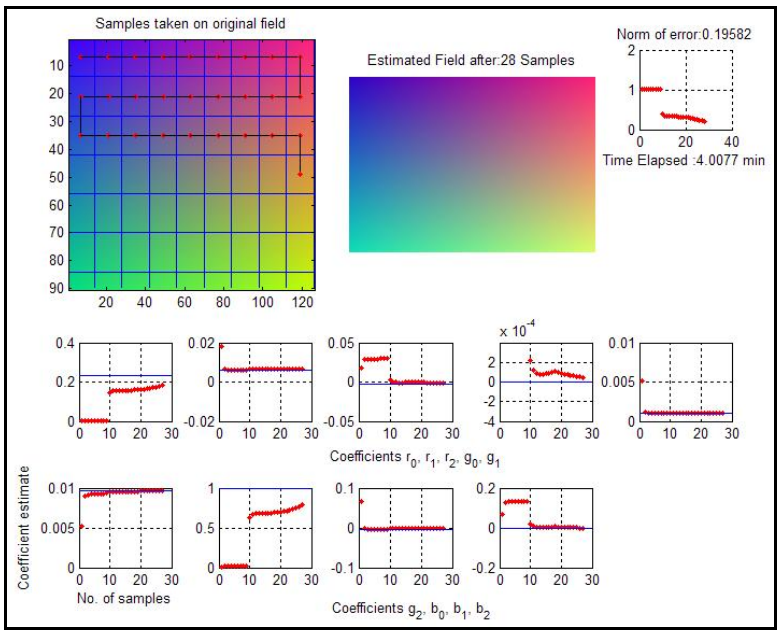


Figure 6.19 Simulation results based on experimental setup for raster scanning of a linear field. Simulation stops when norm of error covariance of field parameters drops below 0.2. It takes 28 samples when Raster scanning is done

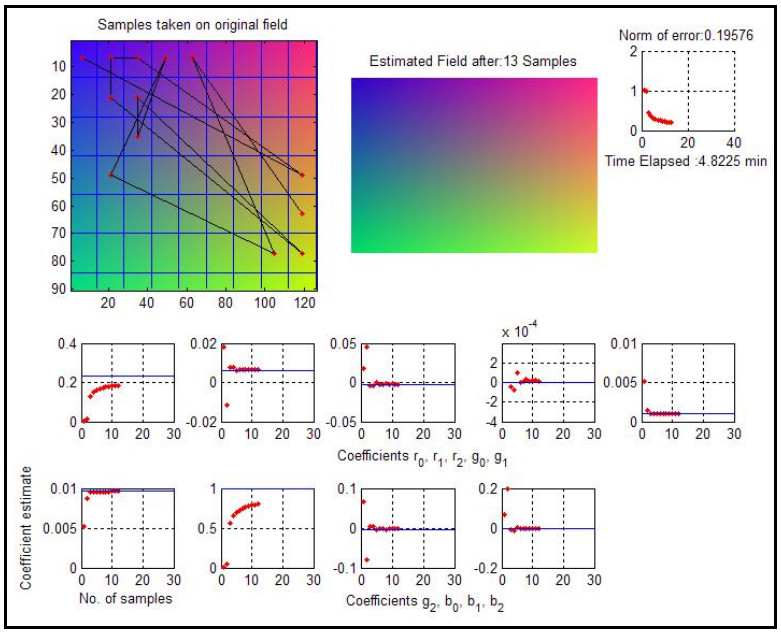


Figure 6.20 Adaptive sampling algorithm takes 13 samples but more time to reconstruct the field

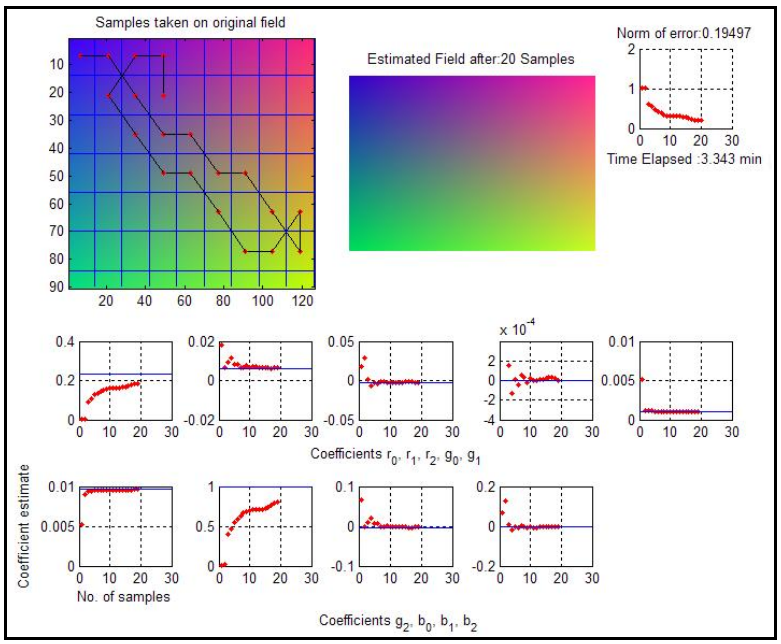


Figure 6.21 Greedy adaptive sampling algorithm is the best trade-off between time and number of samples as it takes 20 samples but least time to reconstruct the field

Table 6.1 Comparison of standard AS, Greedy AS & Raster scanning for linear color field (Simulation Results)

	No. of samples	Time Taken	norm(P)
Raster Scan Sampling	28	4.0077 min	0.19582
Adaptive Sampling	13	4.8225 min	0.19576
Greedy Adaptive Sampling	20	3.343 min	0.19497

6.2.1.2 Experimental results

Experimental results in Figures (6.22)-(6.24) are now presented to confirm that the Greedy AS algorithm recovers the color field in minimum time as shown in Table 6.2. The results illustrate the convergence of the field parameters to values close to nominal after successive samples are taken. The rate of convergence of the three algorithms can be compared through the speed of convergence for r_0, g_0 and b_0. The figures also show a camera snapshot of the actual field, estimated field and norm of the error in these coefficients that decreases with the number of samples. The reason for the slight discrepancy is not color measurement or estimation errors, but differences between screen and printer colors on the lab floor. In these experiments, the robot localization is obtained through the overhead camera image, as opposed to Dead-Reckoning.

Table 6.2 Comparison of standard AS, Greedy AS & Raster scanning for linear color field (Experimental Results)

Algorithms	No. of Samples (n)	2-norm of "error in coefficients" after 'n' samples
Raster Scan	50	0.378
Adaptive Sampling	15	0.365
Greedy AS	30	0.499

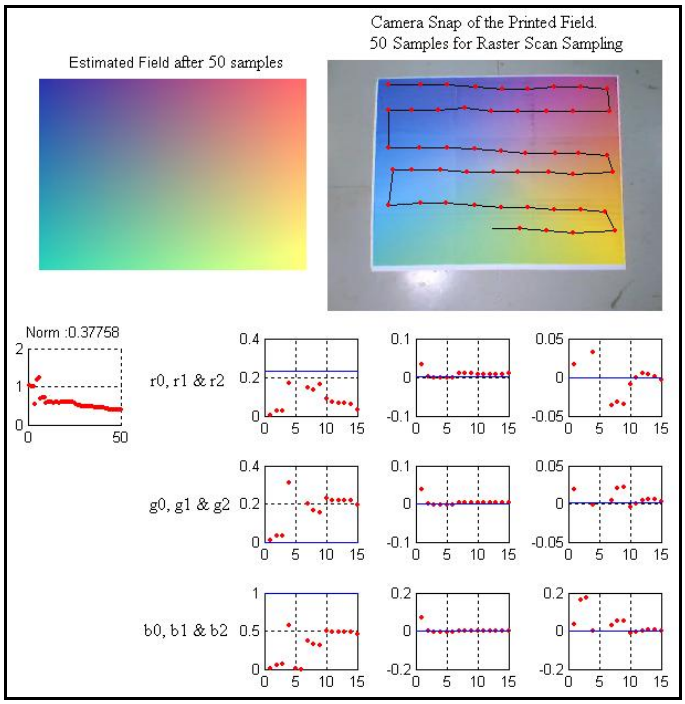


Figure 6.22 Experimental results for Raster Scan Sampling of linear color field shows estimated field after 50 samples

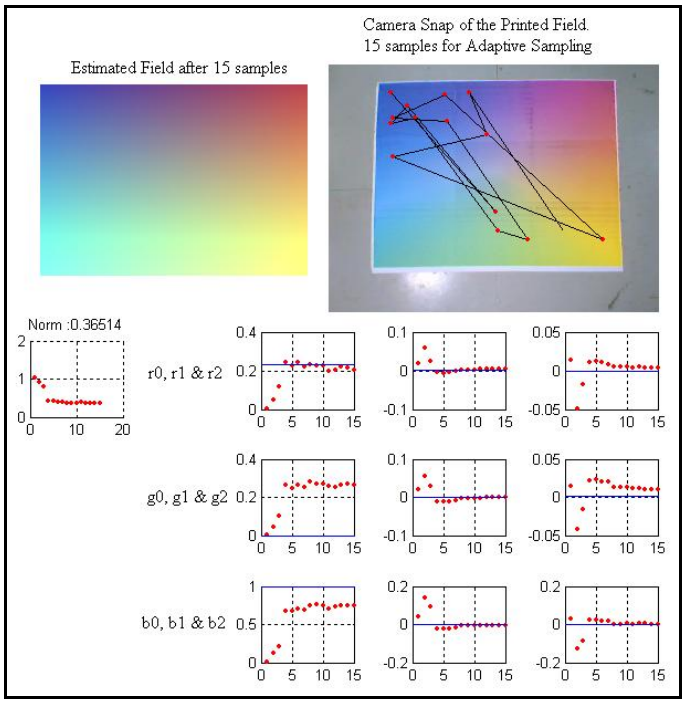


Figure 6.23 Experimental results for AS shows estimated field after 15 samples

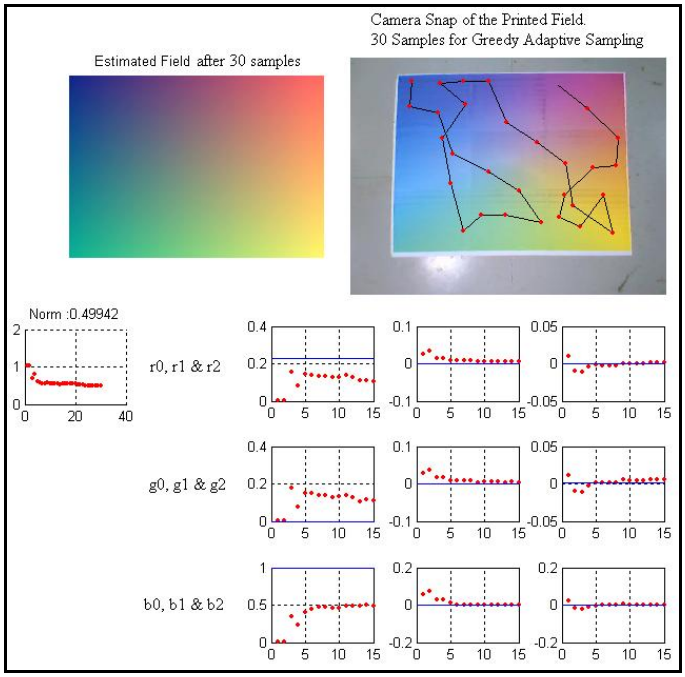


Figure 6.24 Experimental results for GAS shows estimated field after 30 samples

6.2.2 KF estimation for linear color field with uncertainty in localization through camera

In this section, we present experimental validation of the case discussed in (3.2.3) ,when uncertainty in camera measurement is added to the model. Hence there are now two measurements, namely Y_k for localization and Z_k for field measurement. The EKF equations consist of:

- System Model:

$$\begin{pmatrix} X_{k+1} \\ A_{k+1} \end{pmatrix} = \begin{pmatrix} X_k \\ A_k \end{pmatrix} + \begin{pmatrix} B_k \\ 0 \end{pmatrix} U_k + \begin{pmatrix} w_k \\ 0 \end{pmatrix}$$

$$\begin{bmatrix} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}_{k+1} = \begin{bmatrix} x \\ y \\ \theta \\ r_0 \\ r_1 \\ r_2 \\ g_0 \\ g_1 \\ g_2 \\ b_0 \\ b_1 \\ b_2 \end{bmatrix}_k + \begin{bmatrix} \frac{K_R r_R}{2}\cos(\theta_k) & \frac{K_L r_L}{2}\cos(\theta_k) \\ \frac{K_R r_R}{2}\sin(\theta_k) & \frac{K_L r_L}{2}\sin(\theta_k) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\phi_R \\ \Delta\phi_L \end{bmatrix}_k + \begin{bmatrix} w_k \\ 0 \end{bmatrix} \tag{6.10}$$

$$w_k \sim N(0, Q), \quad \xi_k \sim N(0, R_1), \quad v_k \sim N(0, R_2), \quad x_0 \sim N(\bar{x}_0, P_0)$$

- Measurement Model:

$$\begin{bmatrix} \tilde{Y}_k \\ \tilde{Z}_k \end{bmatrix} = \begin{bmatrix} \tilde{x}_k \\ \tilde{y}_k \\ \tilde{\theta}_k \\ \tilde{r}_k \\ \tilde{g}_k \\ \tilde{b}_k \end{bmatrix} = \begin{bmatrix} x_k \\ y_k \\ z_k \\ r_{0k} + r_{1k} x_k + r_{2k} y_k \\ g_{0k} + g_{1k} x_k + g_{2k} y_k \\ b_{0k} + b_{1k} x_k + b_{2k} y_k \end{bmatrix} + \begin{pmatrix} \xi_k \\ \nu_k \end{pmatrix} \tag{6.11}$$

- Effect of System Dynamics:

$$A = I_{12}$$

$$P_{k+1}^{-} = P_k + Q_k$$

$$\hat{X}_{k+1}^{-} = \hat{X}_k + \hat{B}_k U_k$$

$$\hat{A}_{k+1}^{-} = \hat{A}_k$$

$$\begin{bmatrix} \hat{x} \\ y \\ \hat{\theta} \\ \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{bmatrix}_{k+1}^{-} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{\theta} \\ \hat{r}_0 \\ \hat{r}_1 \\ \hat{r}_2 \\ \hat{g}_0 \\ \hat{g}_1 \\ \hat{g}_2 \\ \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{bmatrix}_k + \begin{bmatrix} \frac{K_R r_R}{2}\cos\left(\hat{\theta}_k\right) & \frac{K_L r_L}{2}\cos\left(\hat{\theta}_k\right) \\ \frac{K_R r_R}{2}\sin\left(\hat{\theta}_k\right) & \frac{K_L r_L}{2}\sin\left(\hat{\theta}_k\right) \\ \frac{K_R r_R}{2L_b} & -\frac{K_L r_L}{2L_b} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\phi_R \\ \Delta\phi_L \end{bmatrix}_k \tag{6.12}$$

- Effect of Measurement:

$$\widehat{H}_{k+1} = \frac{\partial \widehat{h}_{k+1}}{\partial(\hat{X}_{k+1}^{-}, \hat{A}_{k+1}^{-})} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{r}_{1k+1}^{-} & \hat{r}_{2k+1}^{-} & 0 & 1 & \hat{x}_{k+1}^{-} & \hat{y}_{k+1}^{-} & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{g}_{1k+1}^{-} & \hat{g}_{2k+1}^{-} & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^{-} & \hat{y}_{k+1}^{-} & 0 & 0 & 0 \\ \hat{b}_{1k+1}^{-} & \hat{b}_{2k+1}^{-} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^{-} & \hat{y}_{k+1}^{-} \end{bmatrix}$$

$$K_{k+1} = P_{k+1}^{-}\widehat{H}_{k+1}^{T}(\widehat{H}_{k+1}P_{k+1}^{-}\widehat{H}_{k+1}^{T} + R_{k+1})^{-1}$$

$$P_{k+1} = (I_{12} - K_{k+1}\widehat{H}_{k+1})P_{k+1}^{-}$$

$$\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}_{k+1}^{-} \\ \hat{A}_{k+1}^{-} \end{pmatrix} + K_{k+1}\left\{ \begin{bmatrix} \tilde{Y}_{k+1} \\ \tilde{Z}_{k+1} \end{bmatrix} - \widehat{h}_{k+1} \right\} = \begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} + K_{k+1}\left\{ \begin{bmatrix} \tilde{x}_{k+1} \\ \tilde{y}_{k+1} \\ \tilde{\theta}_{k+1} \\ \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{x}_{k+1}^{-} \\ \hat{y}_{k+1}^{-} \\ \hat{\theta}_{k+1}^{-} \\ \hat{r}_{0k+1}^{-} + \hat{r}_{1k+1}^{-}\hat{x}_{k+1}^{-} + \hat{r}_{2k+1}^{-}\hat{y}_{k+1}^{-} \\ \hat{g}_{0k+1}^{-} + \hat{g}_{1k+1}^{-}\hat{x}_{k+1}^{-} + \hat{g}_{2k+1}^{-}\hat{y}_{k+1}^{-} \\ \hat{b}_{0k+1}^{-} + \hat{b}_{1k+1}^{-}\hat{x}_{k+1}^{-} + \hat{b}_{2k+1}^{-}\hat{y}_{k+1}^{-} \end{bmatrix} \right\}$$

(6.13)

Through Monte-Carlo experiments, the following uncertainties are determined and added to the EKF model:

(i) Camera accuracy error of zero mean and $(2^{"}, 2^{"}, 2^{\circ})$ variance, meaning that our surrogate GPS measurements are accurate within 2 inches and 2 degrees.

(ii) Color sensor uncertainty of zero mean and $(0.01, 0.01, 0.01)$ variance, meaning that the color sensor accuracy is 1% RGB.

In Figures 6.25 the lines in Cyan color shows the dead-reckoning robot location based on the encoder counts. It is observed that the dead-reckoning error increases as the distance covered increases. At every sample, a color measurement is taken by the robot, and the overhead camera finds the robot location, which is shown in Red color. Location information and color sensor information are fused in the EKF model in

(3.2.3). The estimated robot locations are shown in Black color. After taking 25 samples, error-covariance in (x, y, θ) drops from $(10^4, 10^4, 10^4)$ to $(0.8239, 0.8257, 3.5078)$. The error covariance for the color states drops to:

$$(0.0034, 9.43*10^{-7}, 1.58*10^{-6}, 0.0034, 9.44*10^{-7},$$
$$1.58*10^{-6}, 0.0034, 9.40*10^{-7}, 1.58*10^{-6})$$

The total norm of error in coefficients after 25 samples is 0.8741. Qualitatively, the AS algorithm recovers the printed image on the lab floor very well.

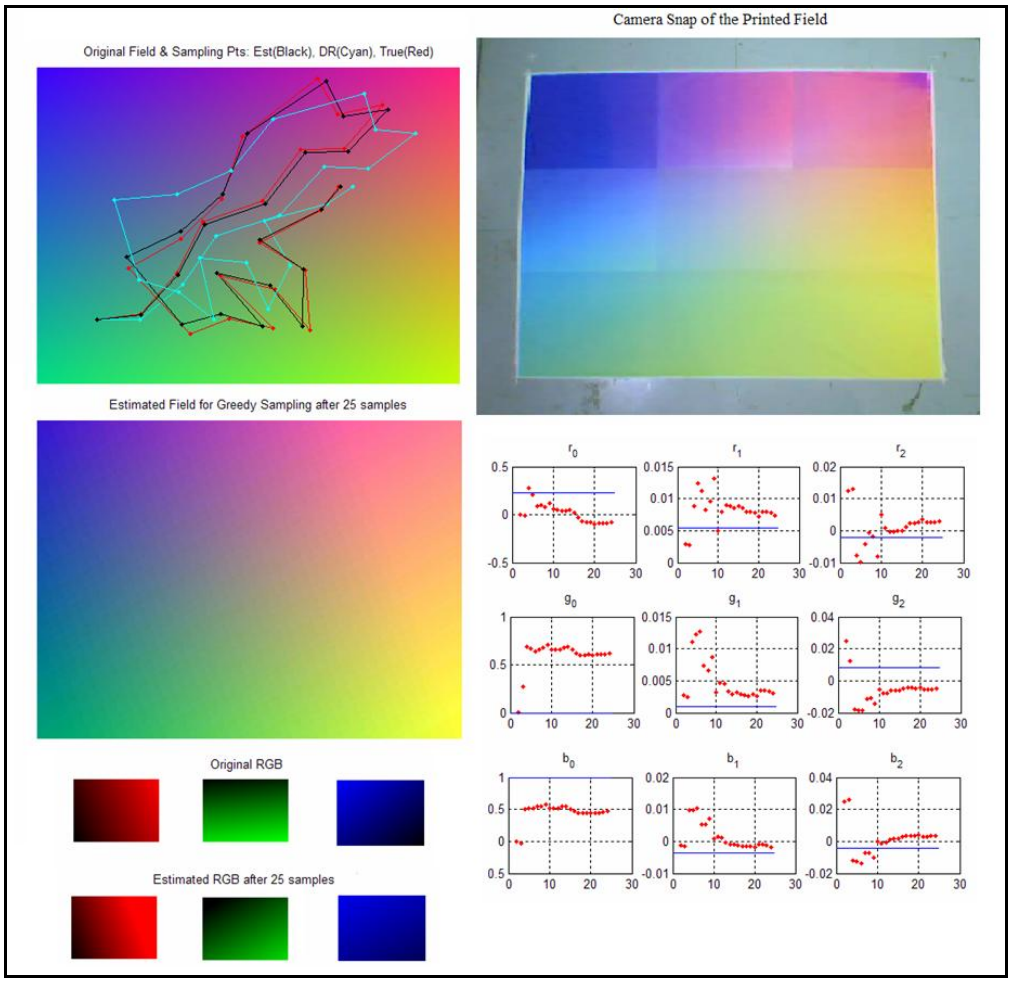


Figure 6.25 Experimental results for GAS when location information from camera and color measurement is fused in the EKF. Figure shows the field convergence after 25 samples considering the camera uncertainty

6.2.3 KF estimation for linear color field with no camera for localization

In this experimental validation for the case presented in section (3.2.4), only the color measurement Z_k is available. In this case, the color sensor measurement is used for field estimation as well as for the robot location estimation. In other words, we are using the color measurements to aid in the localization of the robot. The system model and update equations are the same as the previous case, but measurement equation are now:

- Measurement Model:

$$\begin{aligned}
&\tilde{z}_k = h_k\left(X_k, A_k\right) + v_k \\
&\begin{bmatrix} \tilde{r}_k \\ \tilde{g}_k \\ \tilde{b}_k \end{bmatrix} = \begin{bmatrix} r_{0k} + r_{1k}x_k + r_{2k}y_k \\ g_{0k} + g_{1k}x_k + g_{2k}y_k \\ b_{0k} + b_{1k}x_k + b_{2k}y_k \end{bmatrix} + v_k
\end{aligned} \tag{6.14}$$

- Effect of Measurement:

$$\begin{aligned}
&\hat{H}_{k+1} = \frac{\partial \hat{h}_{k+1}}{\partial(\hat{X}_{k+1}^-, \hat{A}_{k+1}^-)} = \begin{bmatrix} \hat{r}_{1k+1}^- & \hat{r}_{2k+1}^- & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{g}_{1k+1}^- & \hat{g}_{2k+1}^- & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- & 0 & 0 & 0 \\ \hat{b}_{1k+1}^- & \hat{b}_{2k+1}^- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \hat{x}_{k+1}^- & \hat{y}_{k+1}^- \end{bmatrix} \\
&K_{k+1} = P_{k+1}^- \hat{H}_{k+1}^T (\hat{H}_{k+1} P_{k+1}^- \hat{H}_{k+1}^T + R_{k+1})^{-1} \\
&P_{k+1} = (I_{12} - K_{k+1}\hat{H}_{k+1}) P_{k+1}^- \\
&\begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} = \begin{pmatrix} \hat{X}_{k+1}^- \\ \hat{A}_{k+1}^- \end{pmatrix} + K_{k+1}\left\{\tilde{z}_{k+1} - \hat{h}_{k+1}\right\} = \begin{pmatrix} \hat{X}_{k+1} \\ \hat{A}_{k+1} \end{pmatrix} + K_{k+1}\left\{ \begin{bmatrix} \tilde{r}_{k+1} \\ \tilde{g}_{k+1} \\ \tilde{b}_{k+1} \end{bmatrix} - \begin{bmatrix} \hat{r}_{0k+1}^- + \hat{r}_{1k+1}^- \hat{x}_{k+1}^- + \hat{r}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{g}_{0k+1}^- + \hat{g}_{1k+1}^- \hat{x}_{k+1}^- + \hat{g}_{2k+1}^- \hat{y}_{k+1}^- \\ \hat{b}_{0k+1}^- + \hat{b}_{1k+1}^- \hat{x}_{k+1}^- + \hat{b}_{2k+1}^- \hat{y}_{k+1}^- \end{bmatrix} \right\}
\end{aligned} \tag{6.15}$$

Results indicate that the field parameters converge to values close to the actual values for 10 samples, after which the norm of error start to increase rather than

decrease, due to the accumulation of dead-reckoning errors. Hence it is a good idea to estimate the field for a few samples, then stop the field estimation and switch instead to improving the localization using the estimated field. Here the location estimate stays close to the dead-reckoning estimate instead of getting closer to the true robot location (Figure 6.26). After taking 25 samples, the error-covariance in (x, y, θ) drops from $(10^4, 10^4, 10^4)$ to $(90.5653,\ 65.9856,\ 0.9902)$. The error covariance for the color states drops to:

$$(0.0058,\ 1.7359*10^{-6}, 2.7066*10^{-6},\ 0.0079,\ 3.0633*10^{-6},$$
$$4.1337*10^{-6}, 0.0066,\ 1.4988*10^{-6},\ 2.3830*10^{-6})$$

The total norm of error in coefficients after 25 samples is 1.3568. In this case again, the AS algorithm again recovers the printed image on the lab floor as shown in Figure 6.26.

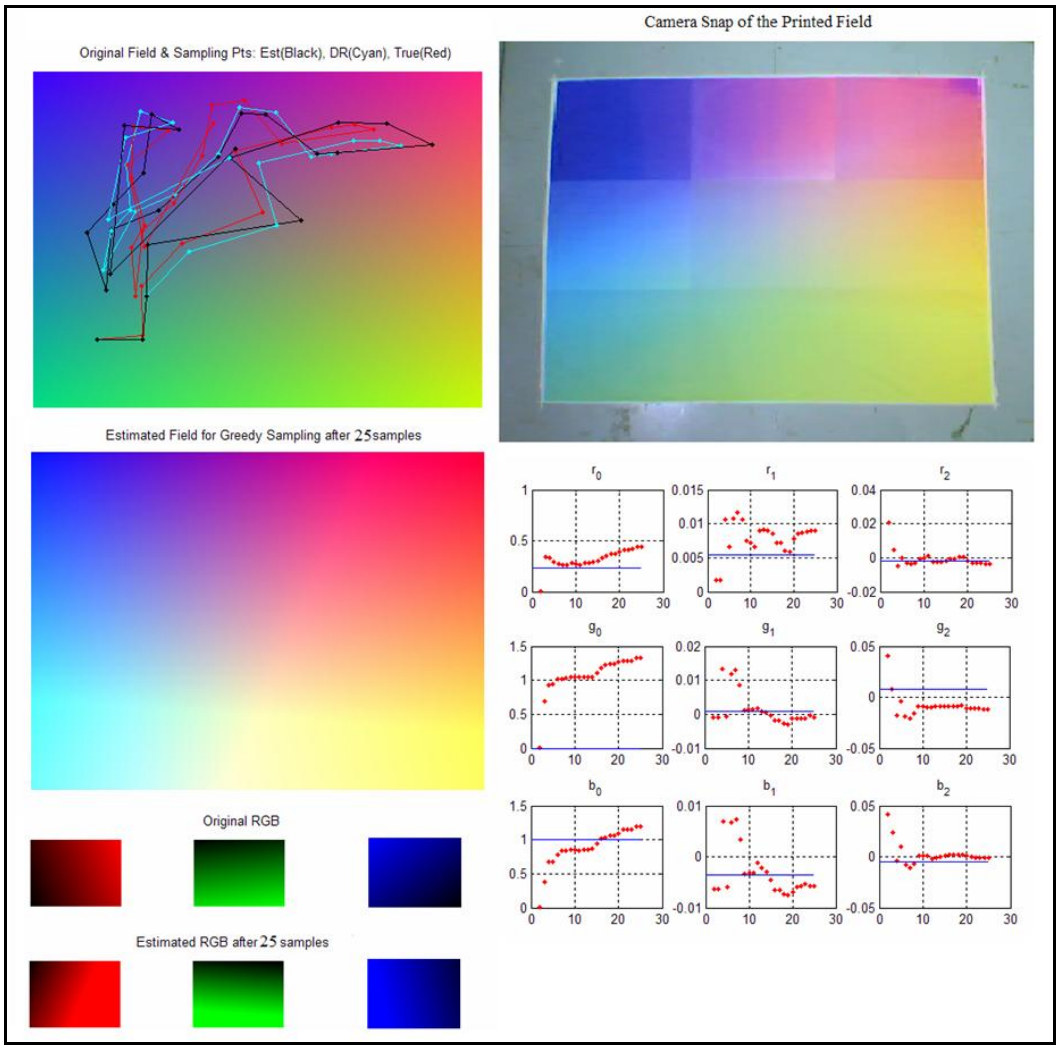


Figure 6.26 Experimental results for GAS with location estimation from color sensor measurement when location information from camera is not available. Figure shows the field convergence after 25 samples

6.2.3.1 Simulations showing the effect of field measurement error on localization and field estimates

As expected, since only the field sensor measurement is used for localization, the increase in sensor noise adversely affects both the estimation of the field, as well as the localization of the robot.

In Figures 6.27-6.30, the lines in cyan color show the dead-reckoning robot location calculated from a differential drive kinematics without using geometric and encoder uncertainties. The location estimate based on the color sensor measurement EKF is shown in yellow, while the "true" robot position calculated from differential drive kinematics with added geometric and encoder uncertainties is plotted in red. The "true" sampling locations correspond to measurements of position using the overhead camera. As expected, the dead-reckoning error increases as the robot covers more distance.

By comparing figures 6.27-6.31 it is clear that the location estimate is close to the true location for small errors in color sensor measurement. Figure 6.31 shows that large discrepancies in both the estimated field and robot position result for large measurement uncertainties.

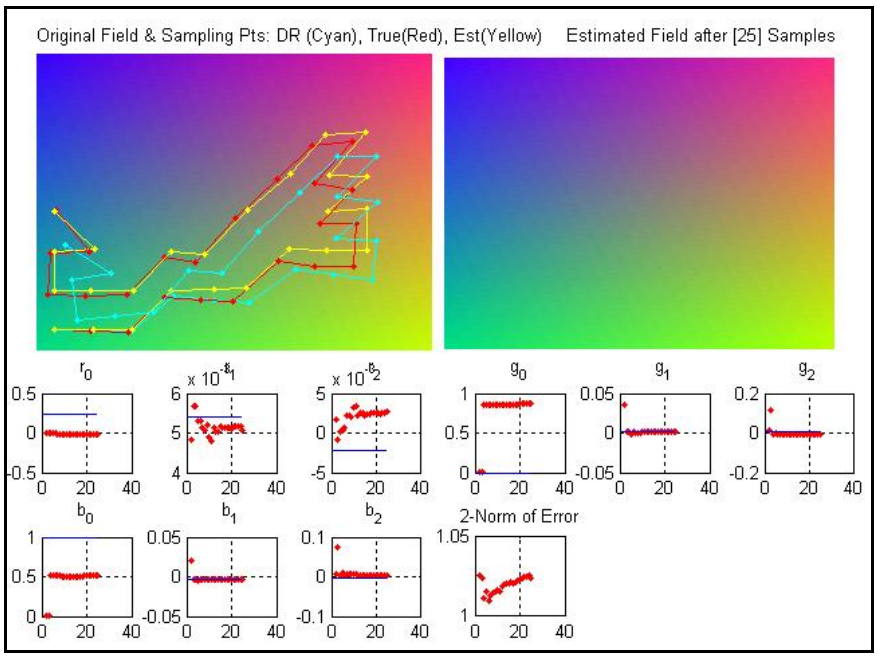


Figure 6.27 Uncertainty of (0.01, 0.01, 0.01) in color sensor measurement: Field estimate converges and localization error is very small (after 25 samples)

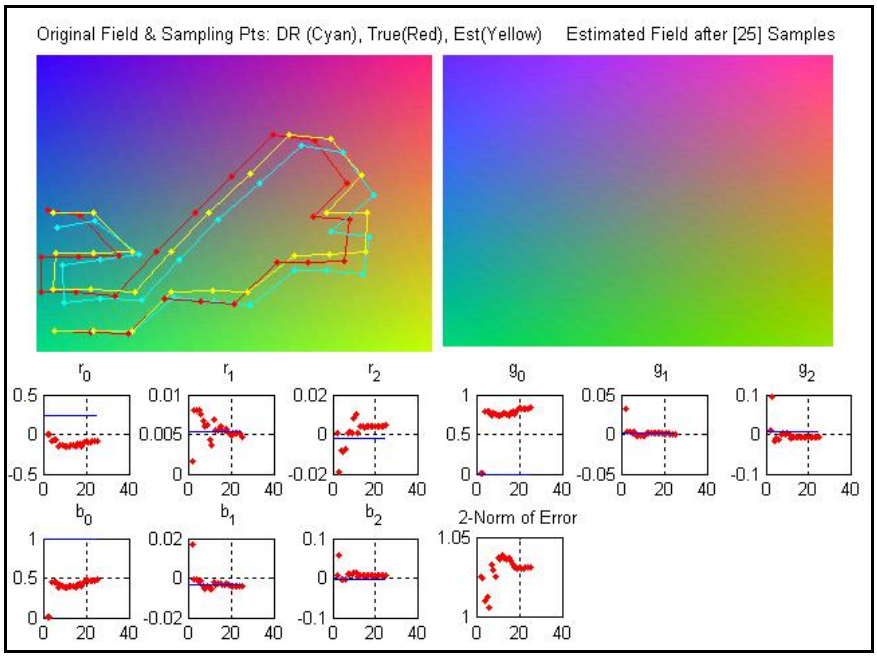


Figure 6.28 Uncertainty of (0.1, 0.1, 0.1) in color sensor measurement: Field estimate converges and localization error is very small (after 25 samples)

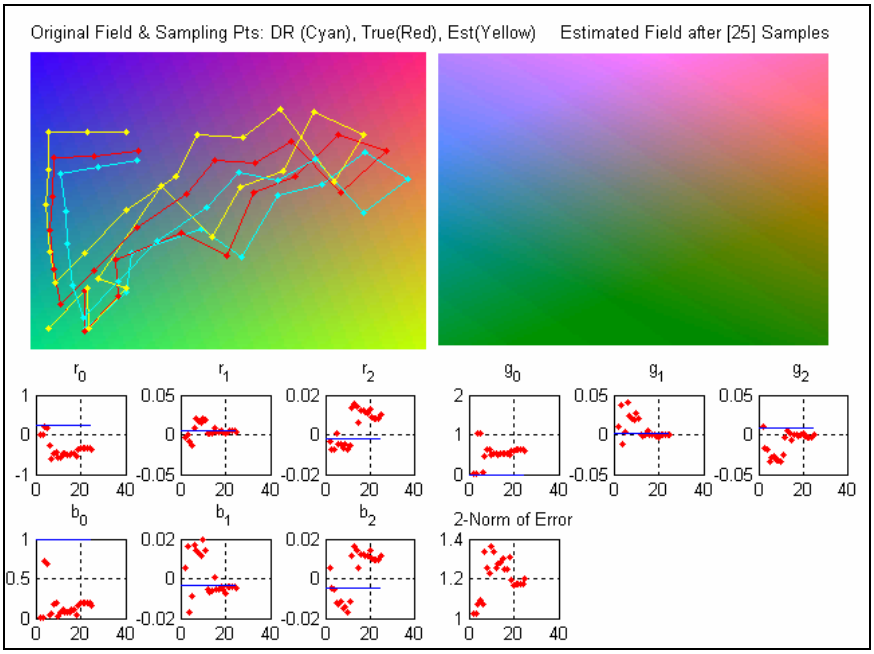


Figure 6.29 Uncertainty of (0.35, 0.35, 0.35) in color sensor measurement: Field estimate converges but with a marginal error and localization error is large

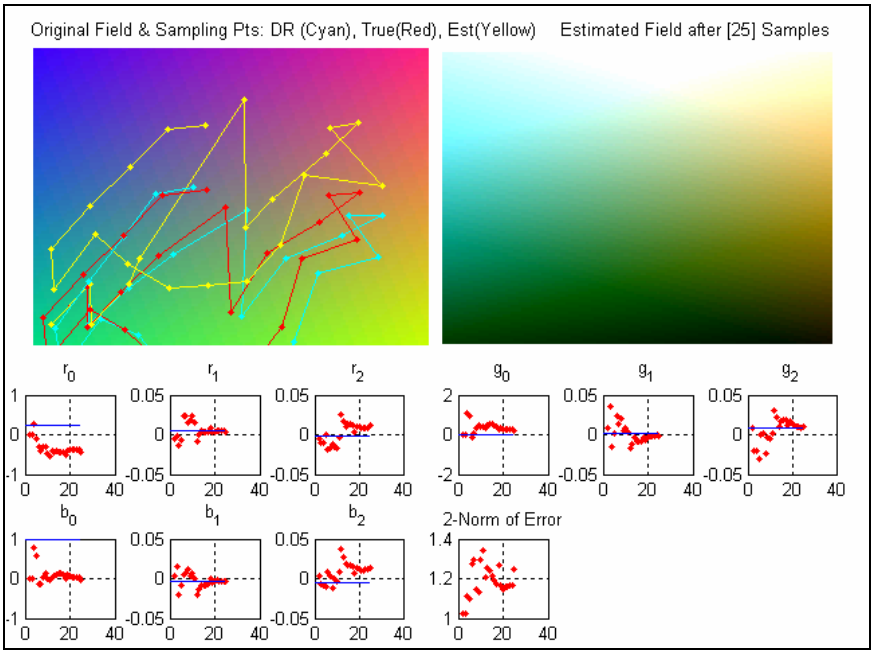


Figure 6.30 Uncertainty of (0.45, 0.45, 0.45) in color sensor measurement: Field estimate diverges and localization error is unacceptable (after 25 samples)

Table 6.3 Effect of sensor measurement error on estimated field parameters

Sensor measurement error (R)	Norm of Error in estimated parameters (after 25 samples)
0.01	1.0240
0.1	1.0316
0.3	1.0911
0.35	1.1995
0.45	1.2462

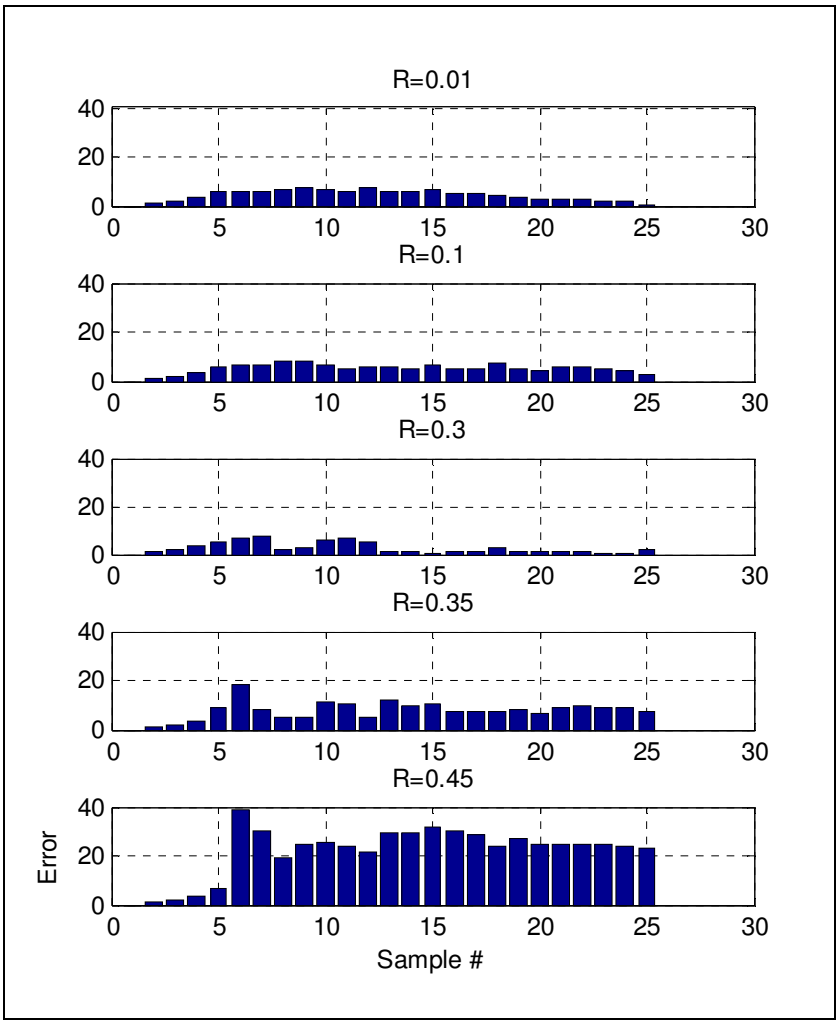


Figure 6.31 Difference between the True and Estimated robot location versus the number of samples for different sensor measurement errors

In this Chapter, experimental results of sampling with simple ARRI-Bot wireless mobile robots are presented for sampling of a linear color field. We used a single robot in several scenarios related to the type of sensor measurements that are available to localize the robot. Different methods to localize the robot were employed, including relative localization using optical encoders, and absolute localization using Crickets and overhead camera. Results clearly demonstrate that in the absence of relative location information, the joint-EKF and the field distribution measurements can

help improve the location estimation of the robot beyond dead-reckoning. Furthermore, the performance of sampling algorithms AS, GAS and RS is similar to the simulation results, with GAS showing the shortest time to recover the color field.

CHAPTER 7

MULTI-ROBOT ADAPTIVE SAMPLING

If the environment is large, it may be impractical for a single robot to navigate to many sampling locations, even in the presence of efficient sampling algorithms. Using multiple robots, the sampling area can be divided into smaller regions, thus reducing the navigation time. Furthermore, since complex fields are represented by hundreds of parameters, it is computationally cumbersome for a single robot to compute and store all parameter estimates and the uncertainty measures. It also quickly becomes unfeasible for individual robots to run a large EKF-AS algorithm, and share large covariance matrices wirelessly. Furthermore, with multi-robot sampling, the resources can be allocated efficiently if some resources are busy or not available.

If we somehow can distribute the filter computation among multiple robots, the number of computations performed by all the robots will be greater than the processing by just one robot doing sampling. However, we expect that the speed of convergence and reduction in complexity that will be gained is significant. With a single robot, the total field estimation time includes the time necessary for navigation, sensing and computations of the estimate (as there is no communication). With multiple robots, the

field estimation time includes the time taken for sensing, computation, communication, and final fusion to recover the field. We expect that the speed of convergence increases using multiple robots simply because of sampling in parallel, and the navigation time reduces significantly compared to modest increases in computation, communication, and fusion.

Our proposed parametric adaptive sampling algorithm, EKF-NN-GAS, represents a complex field with sum of overlapping Gaussians, which means that each sampling instance in a region gains information about the parameters which have dominant effect in that region. Therefore, in order to distribute computations, we need to fuse the parameter estimates and construct the map of density distribution.

This problem is similar to reformulating the algorithm from a conventional single sensor, single processor system to a multi-sensor, multi-processor system. Distributed algorithms have been used before in many applications, and the degree of parallelism varies from algorithm to algorithm. Example includes target location estimation using several sensors, and fusing the measurement either at the central station or at each sensor depending upon the degree of parallelism of the multi-sensor fusion algorithm [35]. Another example is formation control for multiple robots [157].

As discussed in Chapter 2, decentralized and distributed KF are two different problems. In a decentralized algorithm, the filter is full-order, which means that every local filter carries partial information about all parameters, and the information is shared to reach consensus. The objective of distributed algorithms is to efficiently decompose the full-order filter into several reduced-order filters, in order to reduce the

computational complexity and communication overhead, and hence improve the scalability. It can be said that decentralization is the first step towards efficient distribution. Decentralized approaches are good enough for applications involving a small number of states such as tracking of objects, etc. But problems such as parametric sampling involve hundreds of parameters, and hence distributing the filter is very important.

This Chapter focuses on examining completely and partially centralized, decentralized and distributed multi-robot computations, and formulating a sampling scheme which is efficient for running the proposed multi-robot adaptive sampling algorithm. Section 7.1-7.3 summarizes the existing approach for completely centralized, completely decentralized, and partially centralized federated filters respectively. Section 7.4 presents the distributed federated Kalman filter along with description of partitioning of sample space, and the distribution of computations and reduction of communication overhead. Finally in Section 7.5 and 7.6 we discuss the simulation and experimental results respectively to demonstrate the effectiveness of the distributed AS algorithm.

7.1 Completely centralized filter

In a completely centralized sampling approach, each robot j takes sensor measurement $Z_{j,k+1}$ and transmits it to the central processor, which then calculates parameter estimates $\hat{A}_{k+1}$ and error covariances P_{k+1}. The central processor computes the estimated error covariance and parameter estimation using an equation similar to

equation (3.7), with the difference that here it fuses multiple measurements from N robots:

$$P_{k+1} = \left[\left(P_{k+1}^{-}\right)^{-1} + G_{k+1}^{T} R_{k+1}^{-1} G_{k+1}\right]^{-1} = \left[\left(P_{k+1}^{-}\right)^{-1} + \sum_{j=1}^{N} G_{j,k+1}^{T} R_{j,k+1}^{-1} G_{j,k+1}\right]^{-1}$$

$$\hat{A}_{k+1} = \hat{A}_{k+1}^{-} + P_{k+1} G_{k+1}^{T} R_{k+1}^{-1} \left[\begin{pmatrix} Z_{1,k+1} \\ Z_{2,k+1} \\ \vdots \\ Z_{N,k+1} \end{pmatrix} - \begin{pmatrix} g_{k+1}(\hat{A}_{k+1}^{-}) \\ g_{k+1}(\hat{A}_{k+1}^{-}) \\ \vdots \\ g_{k+1}(\hat{A}_{k+1}^{-}) \end{pmatrix}\right] \quad . \qquad (7.1)$$

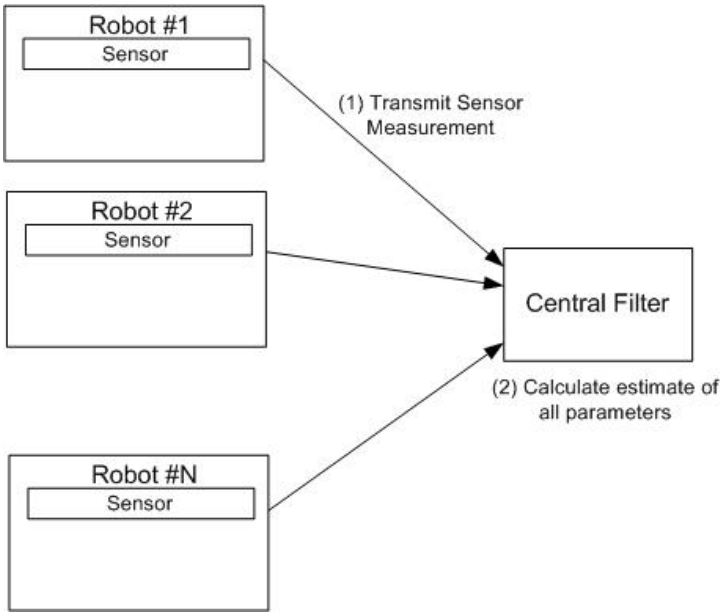


Figure 7.1 Completely centralized AS algorithm

Figure 7.1 illustrates the completely centralized approach, in which all robots transmit their sensor measurement to the central filter, which then calculates the field estimate using equation (7.1). This type of scheme is simple, as there is little communication involved and no redundant computations. But the disadvantage is that the sensing robots do not carry any information. Hence, they do not know where to

sample next until told to do so by the central processor, which is running a large algorithm receiving asynchronous data.

7.2 Completely decentralized filter

For a completely decentralized filter implementation, each robot runs the AS algorithm, and generates new sampling locations within the vicinity of its current position. The robots take new measurements and calculate only partial estimates of the field parameters and error covariance. After a few samples, the robots communicate and share their field estimate information. The parameter estimate and the error covariance are the two terms each robot needs to transmit to others. Each robot assimilates the received information using for instance a decentralized EKF scheme similar to [35].

If a completely decentralized approach considered, then an AS algorithm running on each robot carries the information about all the field parameters, and there is no data fusion filter. Hence each robot i can calculate the local field estimate $\hat{A}_{i,k+1,LE}$ and $P_{i,k+1,LE}$ using the following equations:

$$\begin{aligned} P_{i,k+1,LE} &= \left[\left(P_{i,k+1}^{-}\right)^{-1} + G_{i,k+1}^{T} R_{i,k+1}^{-1} G_{i,k+1}\right]^{-1} \\ \hat{A}_{i,k+1,LE} &= \hat{A}_{i,k+1}^{-} + P_{i,k+1,LE} G_{i,k+1}^{T} (R_{i,k+1})^{-1}\left[Z_{i,k+1} - g_{k+1}(X_{k+1}, \hat{A}_{k+1}^{-})\right]. \end{aligned} \tag{7.2}$$

Robot i acquires the local estimate $\hat{A}_{j,k+1,LE}$ and $P_{j,k+1,LE}$ from the other $(N-1)$ robots and assimilates it to get the complete information $P_{i,k+1}$ and $\hat{A}_{i,k+1}$. The following assimilation equation need to run on robot i:

$$(P_{i,k+1})^{-1} = (P^{-}_{i,k+1})^{-1} + \sum_{j=1}^{N}\left[(P_{j,k+1,LE})^{-1} - (P^{-}_{j,k+1})^{-1}\right]$$
$$\hat{A}_{i,k+1} = P_{i,k+1}\left[(P^{-}_{i,k+1})^{-1}\hat{A}^{-}_{i,k+1} + \sum_{j=1}^{N}\left[(P_{j,k+1,LE})^{-1}\hat{A}_{j,k+1,LE} - (P^{-}_{j,k+1})^{-1}\hat{A}^{-}_{j,k+1}\right]\right]. \quad (7.3)$$

Figure 7.2 illustrates the completely decentralized filter in which each robot has a local filter to compute local estimates and a global filter for assimilating the estimates acquired from other nodes and generate the global field estimate.

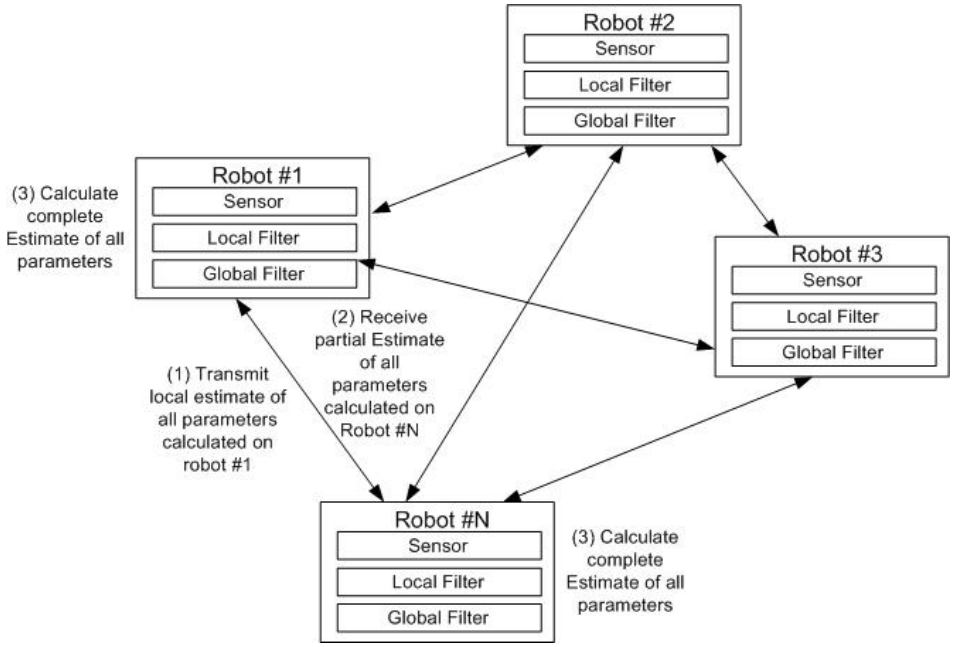


Figure 7.2 Completely decentralized AS algorithm

The advantage of this approach is that it does not involve any approximations, and there is no dependence on a central filter, at least for computing the local estimate. Also the objective of sampling in parallel can be successfully achieved. The disadvantage of the algorithm is that it is inefficient in terms of communication and computational requirements. The network has to be fully connected and there is

excessive communication. This type of algorithm is good for applications such as target tracking, because the problem involves estimating only few parameters (such as location, speed, etc of the target). Hence, the distribution of computations may not be that important if the number of parameters is small. For an adaptive sampling algorithm running such that each robot has a designated sampling region, different robots carry better information about different parameters. Hence, a loss of information from only a few robots will make a significant difference to the field estimation result. This is not the case for target tracking, where a loss of information from one or two nodes does not significantly impact the accuracy of estimate, if enough nodes can still track the target.

For an adaptive sampling scenario, if, for example, a field is represented by 100 parameters, running this decentralized algorithm would require each robot to calculate the local estimate of 100 parameters, and to wirelessly transmit an error covariance matrix of size 100x100, and a parameter estimate vector of size 100x1 to every other robot. Clearly, such a scheme would be very inefficient and not scalable.

7.3 Partially centralized federated filter

A tradeoff approach can be adopted for sampling, in which each robot i takes sensor measurements, estimates local error covariances and field parameters, and transmits this information to the global filter for assimilation, in a similar fashion to the approach proposed in [137]. Each robot runs equation (7.2), but the fusion is done only at the fusion filter using equation (7.3).

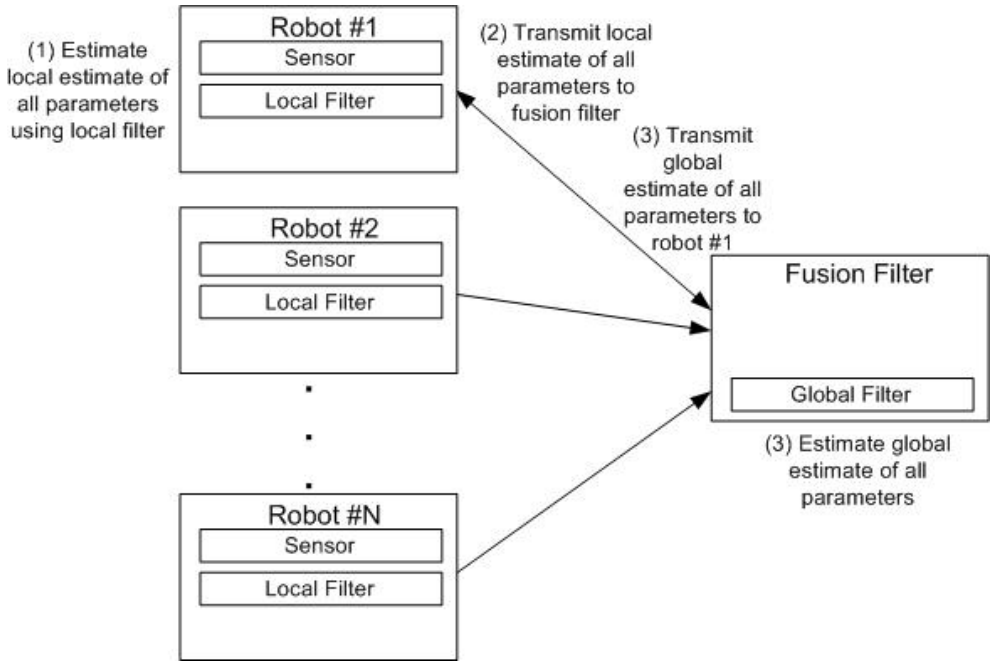


Figure 7.3 Partially centralized federated AS algorithm

Figures 7.3 illustrate the centralized federated filter in which each robot calculate a local estimate and transmits it to the fusion filter, which then computes the global field estimate. The advantage of this approach is that there is less communication compared to the completely decentralized case. Although none of the robots carries complete information about all parameters, this approach will be more efficient than the centralized implementation, because adaptive sampling algorithms can run based on the knowledge about local parameter estimates. The formulation of the filter can be implemented in a general approach by assuming that each robot carries some information about all the parameters. For multi-robot sampling performed in such away that sampling area for robots is assigned apriori, it is quite reasonable to assume that every robot carries information about some parameters and no information about others.

This assumption can be used to distribute the filter equations, as described in the following section.

7.4 Distributed Federated Kalman Filter (DFKF)

In this section we describe an algorithm to distribute the computations of the field parameter estimates among multiple robots, and we show that the scheme is efficient in terms of communication and computation overhead. The sampling area is divided, e.g. each robot is responsible for sampling certain regions and sets of parameters. A block diagram of the distribution scheme is shown in Figure 7.4. Compared to the approach presented in previous section, such a distributed scheme involves the transmission and computation of only certain parameters.

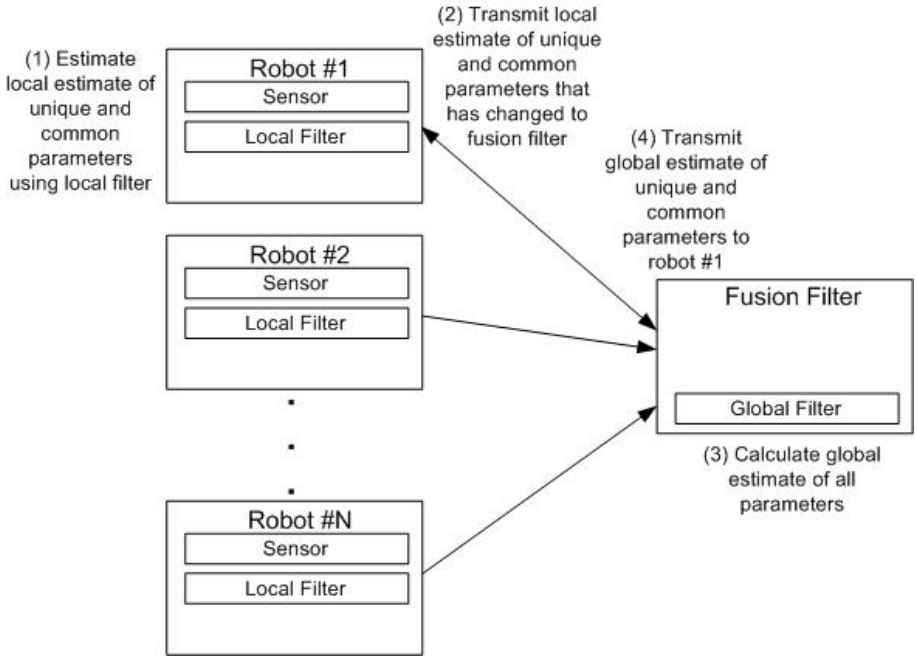


Figure 7.4 Distributed federated AS algorithm

7.4.1 Partitioning of sampling area

A method is clearly needed to efficiently divide the sampling area into clusters, in order to run a parallel AS algorithm with multiple robots. Fuzzy *c*-means clustering (FCM) has been used in many applications for classification of numerical data [51]. The basics of FCM were presented in Chapter 2. Moreover, Centroidal Voronoi Tessellation (CVT) diagrams [50], have been recently proposed for forming non-uniform size grids to better explore high variance areas for non-parametric distributions [15]. Here a scheme is proposed in this thesis to efficiently divide the sampling area for parametric distributions using both FCM and CVT. In this approach, the FCM algorithm clusters based on estimated location of the center of approximating Gaussians.

The implementation of the CVT diagram is based on Lloyd's algorithm [50], and uses the centroid locations acquired by fuzzy clustering to fuse all points in discrete space that are closest to the centroid as a single group. Mathematically, a point p on the field is part of the cluster r if:

$$\left|p - c_r\right| \le \left|p - c_s\right|, \quad s = 1,...C, s \ne r. \tag{7.4}$$

As a result, more Gaussians will overlap in those areas where there are large field variations. The use of fuzzy *c*-means clustering and the CVT diagram for area classification results in regions which have more variations to be as small as it is required in order to sample them thoroughly. The area with less variation, though large, requires less samples, since it is represented by only a few parameters. The idea is illustrated in the simulation of figure 7.5, where we partitioned the sampling space of a non-uniform distribution represented by L=100 Gaussians into 8 regions.

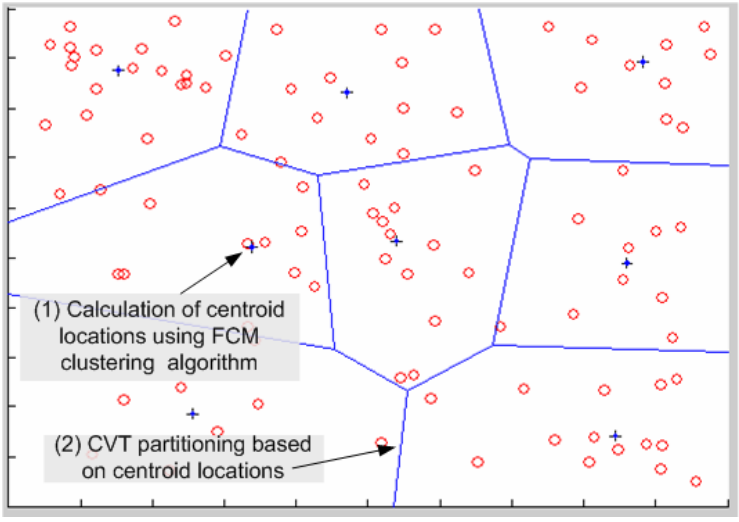


Figure 7.5 Sampling area with Gaussian field centers partitioning performed in two steps using FCM and CVT

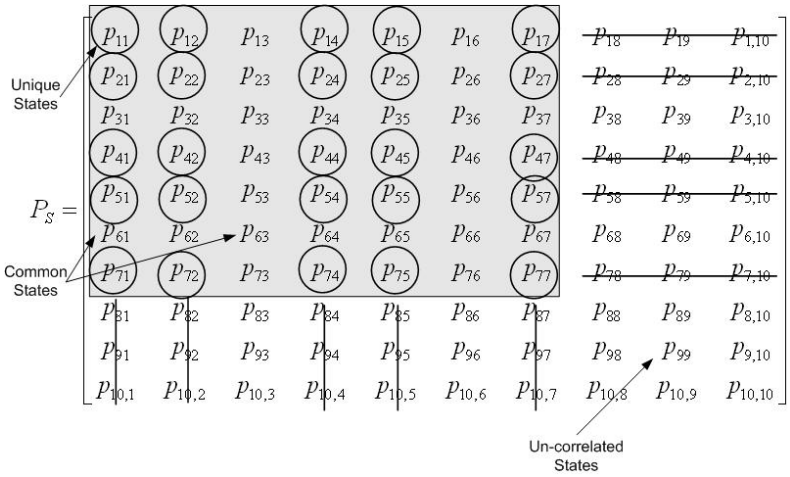


Figure 7.6 Three categories of parameters (*Unique*, *Common* and *Uncorrelated*) can be formed if DFKF approach is used along with partitioning of sampling area for each robot

Individual robots are only interested in the parameters that are either unique to it or common with other robots. These parameters fall under three categories of obvious significance, referred here as: *common, unique, and uncorrelated*. Common parameters are most crucial to the field estimate, as they exist near the border of the sampling areas.

The sampling information from multiple robots contributes to the estimation of common parameters. As shown in figure 7.4, only the estimate of common and unique parameters that have changed need to be transmitted to the fusion filter. Furthermore, the cross-covariance terms of unique states with uncorrelated states can be assumed to be zero. Similarly, some of the *common* states are not correlated with the *uncorrelated* states.

7.4.2 Distributed computations and communications

The objective of the work presented in this section is to modify the formulation of a completely decentralized federated scheme, in order to reduce communication and computational load. This formulation is new because it considers the cross-covariance terms of neighboring Gaussians and ignores the ones which are far from each other as a tradeoff between accuracy and computational complexity. Accurate DKF is not possible in this adaptive sampling problem because local measurement models are not available. Furthermore, the use of global measurement models on each node requires the estimate of all parameters, which will contradict the motivation behind the implementation of DKF. There are other schemes that handle the error covariance terms "very lightly" such as Kalman Consensus schemes, which take the average of error covariance of parameter estimate in order to implement the DKF with only neighboring nodes communication [29].

EKF has $O(n^3)$ computational complexity if each sample updates all the S parameters of the n-dimensional parametric field. However it can be assumed that a single sample affects only neighboring parameters. With this assumption, the algorithm

can run in distributed fashion, and sampling nodes computation complexity can be reduced. Only the fusion filter's complexity remains of order O(n^3), because it needs to combine information about all the parameters. However, this central field parameter fusion process occurs less frequently.

Table 7.1 Comparison of computational complexity and communication overhead for centralized, decentralized, federated and distributed federated filter

	Computations			Communication
	Robot	Fusion Center	Combined	
Centralized Filter	-	$O(Nmn^3)$	$O(Nmn^3)$	$O(Nm)$
Decentralized Filter	$O\left(mn^3+(N-1)pn^3\right)$	-	$O\left(Nmn^3+N(N-1)pn^3\right)$	$O\left(N(N-1)pn^3\right)$
Decentralized Federated Filter	$O(mn^3)$	$O(pn^3)$	$O(Nmn^3+pn^3)$	$O(Npn^3)$
Distributed Filter	$O\left(\frac{mn^3}{N^3}\right)$	$O(pn^3)$	$O\left(\frac{mn^3}{N^2}+pn^3\right)$	$O\left(\frac{pn^3}{N^2}\right)$

Table 7.1 illustrates a comparison of computations and communication complexity for centralized, completely decentralized, federated decentralized and distributed filter. Let N be the number of sampling robots, m is the number of sensor measurements per robot, n is the number of field parameters, and p is the number of times robots communicate to share their information.

For the centralized filter, the sensing robots do not perform any computation. Hence, the computational and communication complexity are $O(Nmn^3)$ and $O(Nm)$ respectively.

For a completely decentralized filter, the computational complexity to calculate the local estimate on each robot is $O(mn^3)$, and to calculate the global estimate on each robot is $O((N-1)pn^3)$, after taking estimates from $(N-1)$ robots at a frequency of p. Hence the combined computational complexity becomes $O(Nmn^3 + N(N-1)pn^3)$. At the same time, the communication complexity is $O(N(N-1)pn^3)$.

In order to reduce the communication overhead and computational complexity, a federated filter calculates the global estimate on the fusion filter, which reduces the computational complexity to $O(mn^3 + pn^3)$, and the communication complexity to $O(Npn^3)$.

Finally, for the proposed distributed version of federated decentralized filter, instead of calculating n states on a single robot, we simply calculate the estimate of n/N states on a single robot. This approach reduces the computational and communication complexity to $O(mn^3/N^2 + pn^3)$ and $O(pn^3/N^2)$, respectively.

Let's now assume that the field classification includes a large cluster for a continuous field distribution. A *large cluster* is defined as the one whose parameters are independent of parameters outside that cluster. This means that the Gaussians representing this large cluster do not overlap with other Gaussians outside the cluster. Furthermore, let's assume that there are *L* parameters in a large cluster, and that a cluster head robot is in-charge of carrying the updated parameters estimates and the error covariance of all *L* parameters.

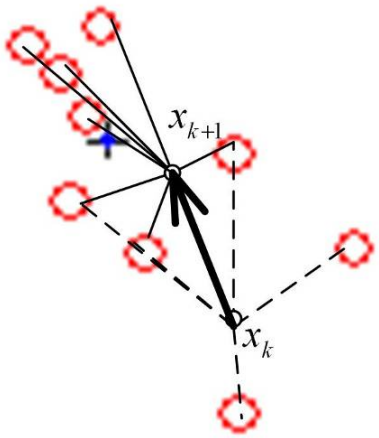


Figure 7.7 Centers of Gaussians that are close to sampling locations x_k and x_{k+1}

The large cluster will contains smaller clusters. A *small cluster* is defined as one whose parameters are estimated by the same robot for a span of time until the tessellation changes. The participant parameters, and hence the shape of a small cluster might change after some samples are taken and as the parameter estimates change. Parameters of one small cluster can be dependent on parameters of another small cluster around it. Furthermore, let's assume that there are *S* parameters in or around a particular small cluster that are expected to change because of sampling for next few samples by a particular robot. Assume *M* parameters are expected to change by sampling at a particular location. In our scheme, instead of computing the error covariance and parameter estimate of size S, each sample updates only M parameters.

Summarizing the clustering parameters, and referring to the diagram in Figure 7.8, let

N = Number of sampling robots

L = Parameters in large cluster

S = Parameters in or around small cluster

M= Parameters that are expected to change by sampling a particular location

C= Parameters whose estimates have change so far since last update from cluster head

P_L =Error covariance of L parameters which includes unique, common and uncorrelated parameters

P_S =Error covariance of S parameters which are either unique or common

P_M =Error covariance of M parameters

P_C =Error covariance of C parameters

$\hat{A}_L$ =Estimate of L parameters

$\hat{A}_S$ =Estimate of S parameters

$\hat{A}_M$ =Estimate of M parameters

$\hat{A}_C$ =Estimate of C parameters

$P_L, \hat{A}_L$ are carried by the cluster head

$M \subset C \subset S \subset L$

$g_L(X_0, A_{L,0})$ =sum of L Gaussians

$g_S(X_0, A_{S,0})$ =sum of S Gaussians

$g_M(X_0, A_{M,0})$ =sum of M Gaussians

Sampling in region M, the field measurements are given by:

$$Z_{k+1} = g(X_{k+1}, A_{L,k+1}) + \nu_{i,k+1} \approx g(X_{k+1}, A_{S,k+1}) + \nu_{i,k+1} \approx g(X_{k+1}, A_{M,k+1}) + \nu_{i,k+1} \quad (7.5)$$

$$g(X_{k+1}, \hat{A}_{M,k+1}) = \sum_{u=1}^{M} g(X_{k+1}, \hat{A}_{u,k+1}) \quad , \tag{7.6}$$

where M Gaussians are the ones whose centers are closest to the currently sampled location $X_{i,k+1}$ by robot i, and

$$G_{M,k+1} = \frac{\partial g(X_{k+1}, A_{M,k+1})}{\partial \hat{A}_{M,k+1}} \in \Re^{1xM}, \; G_{S,k+1} = \frac{\partial g(X_{k+1}, A_{S,k+1})}{\partial \hat{A}_{S,k+1}} \in \Re^{1xS}. \tag{7.7}$$

$g(X_{k+1}, A_{L,k+1})$ is sum of L Gaussians, but for a particular sampling location it will only affect M Gaussians considerably, and to a lesser extent other Gaussians. These M Gaussians can be assumed to be the ones close to the currently sampling location. The dimension of the matrices is reduced by binary transformation matrices U that act as masks over the actual dense EKF matrices. The communication between robots is reduced by transmitting only the parameter values that changed since last update, instead of transmitting the information about all the parameters every time. The transformation matrix is also sent along with the estimates to the fusion filter in order for the cluster head to know which parameters have changed since the last update. The proposed algorithm consists of the following steps:

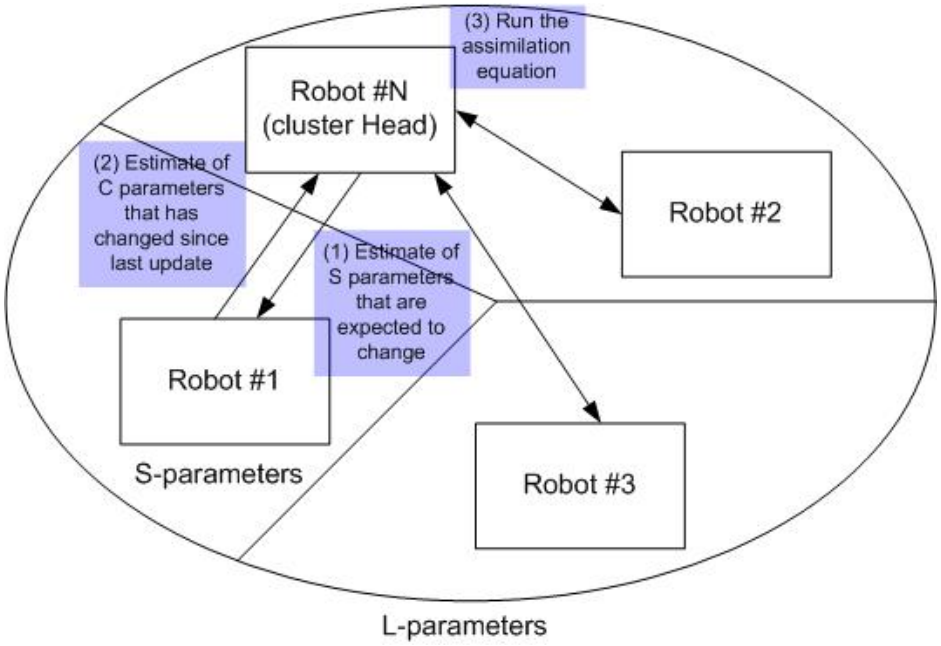


Figure 7.8 Partially centralized AS algorithm with distributed computations

Distributed Federated Kalman Filter (DFKF)

Step 1: At the Cluster head transformation from $(P_{L,k}, \hat{A}_{L,k})$ to $(P_{S,k}, \hat{A}_{S,k})$

The cluster head evaluates the initial estimate of $(P_{S,k}, \hat{A}_{S,k})$ by first generating the transformation matrix $U_{LS,k}$, and transmitting $(P_{S,k}, \hat{A}_{S,k})$ to robot 1. The matrix $U_{LS,k} = U_{SL,k}^T$ is kept in memory by the cluster head for the final assimilation stage.

$$\begin{aligned} P_{S,k} &= U_{LS,k}^T P_{L,k} U_{LS,k} \\ \hat{A}_{S,k} &= U_{LS,k}^T \hat{A}_{L,k} \\ \hat{A}_{S,k} &\in \Re^{Sx1}, P_{S,k} \in \Re^{SxS}, \hat{A}_{L,k} \in \Re^{Lx1}, P_{L,k} \in \Re^{LxL}, U_{LS,k} \in \Re^{LxS} \end{aligned} \tag{7.8}$$

Step 2: Transmit estimate of *S* parameters $(P_{S,k}, \hat{A}_{S,k})$ to Robot #1

Step 3: Take measurement sample and calculate estimate of *M* parameters $(P_{M,k+1}, \hat{A}_{M,k+1})$ via:

$$P_{M,k+1} = \left[\left(U_{SM,k+1}^T P_{S,k} U_{SM,k+1}\right)^{-1} + G_{M,k+1}^T R_{k+1}^{-1} G_{M,k+1}\right]^{-1} \in \Re^{MxM}$$

$$\hat{A}_{M,k+1} = U_{SM,k+1}^T \hat{A}_{S,k} + P_{M,k+1} G_{M,k}^T (R_{k+1})^{-1}\left[Z_{k+1} - g(X_k, \hat{A}_{M,k})\right] \in \Re^{Mx1} \quad (7.9)$$

$$G_{M,k+1} = \frac{\partial g(X_k, \hat{A}_{M,k+1}^-)}{\partial \hat{A}_{M,k+1}^-} \in \Re^{1xM}$$

$$\hat{A}_{M,k+1} \in \Re^{Mx1}, P_{M,k+1} \in \Re^{MxM}, \hat{A}_{S,k+1} \in \Re^{Sx1}, P_{S,k+1} \in \Re^{SxS}, U_{SM,k+1} \in \Re^{SxM}$$

Step 4: Transformation from $(P_{M,k+1}, \hat{A}_{M,k+1})$ to $(P_{C,k+1}, \hat{A}_{C,k+1})$ using:

$$\begin{aligned} P_{C,k+1} &= U_{C01}^T P_{C,k} U_{C01} + U_{MC,k+1}^T (P_{M,k+1} - U_{SM,k+1}^T P_{S,k} U_{SM,k+1}) U_{MC,k+1} \\ \hat{A}_{C,k+1} &= U_{C01}^T \hat{A}_{C,k} + U_{MC,k+1}^T (\hat{A}_{M,k+1} - U_{SM,k+1}^T \hat{A}_{S,k}) \end{aligned} \quad (7.10)$$

$$\hat{A}_{C,k+1} \in \Re^{Cx1}, P_{C,k+1} \in \Re^{CxC}, \hat{A}_{C,k} \in \Re^{C_0 x1}, P_{C,k} \in \Re^{C_0 xC_0},$$
$$U_{MC,k+1} \in \Re^{MxC}, U_{C01} \in \Re^{C_0 xC}$$

Step 5: Transmit estimate of $(P_{C,k+1}, \hat{A}_{C,k+1}, U_{CS,k+1})$ parameters to cluster head when requested

Step 6: <u>At Robot 1:</u> Transformation from $(P_{M,k+1}, \hat{A}_{M,k+1})$ to $(P_{S,k+1}, \hat{A}_{S,k+1})$ for further sampling. Go to step 3

$$\begin{aligned} P_{S,k+1} &= P_{S,k} + U_{MS,k+1}^T (P_{M,k+1} - U_{SM,k+1}^T P_{S,k} U_{SM,k+1}) U_{MS,k+1} \\ &= P_{S,k} + U_{MS,k+1}^T G_{M,k+1}^T R_{k+1}^{-1} G_{M,k+1} U_{MS,k+1} \\ \hat{A}_{S,k+1} &= \hat{A}_{S,k} + U_{MS,k+1}^T (\hat{A}_{M,k+1} - U_{SM,k+1}^T \hat{A}_{M,k}) \end{aligned} \quad (7.11)$$

$$\hat{A}_{S,k+1} \in \Re^{Sx1}, P_{S,k+1} \in \Re^{SxS}, \hat{A}_{M,k+1} \in \Re^{Mx1}, P_{M,k+1} \in \Re^{MxM}, U_{MS,k+1} \in \Re^{MxS}$$
$$U_{MS,k+1} = U_{SM,k+1}^T$$

Step 7: <u>At Cluster head:</u> Run the assimilation equations after *n* steps:

$$(P_{L,k+n})^{-1} = (P_{L,k})^{-1} + \sum_{j=1}^{N}\left[(P_{L,k+n})^{-1} - (P_{L,k})^{-1}\right]_j$$
$$\hat{A}_{L,k+n} = P_{L,k+n}\left[(P_{L,k})^{-1}\hat{A}_{L,k} + \sum_{j=1}^{N}\left[(P_{L,k+n})^{-1}\hat{A}_{L,k+n} - (P_{L,k})^{-1}\hat{A}_{L,k}\right]\right]_j \quad (7.12)$$

where,

$$\left[P_{L,k+n}\right]_j = \left[P_{L,k} + U^T_{SL,k+n}(U^T_{CS,k+n}P_{C,k+n}U_{CS,k+n} - U^T_{CS,k}P_{C,k}U_{CS,k})U_{SL,k+n}\right]_j \quad (7.13)$$
$$\left[\hat{A}_{L,k+n}\right]_j = \left[\hat{A}_{L,k} + U^T_{SL,k+n}(U^T_{CS,k+n}\hat{A}_{C,k+n} - U^T_{CS,k}\hat{A}_{C,k})\right]_j$$

Let's now consider an example for the proposed algorithm:

Assume that there are 10 parameters in a small cluster S, to be calculated by a robot. The error covariance $P_{S,k}$ and parameter estimate $A_{S,k}$ matrices can be written as:

$$P_{S,k} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{15} & p_{16} & p_{17} & p_{18} & p_{19} & p_{1,10} \\ p_{21} & p_{22} & p_{23} & p_{24} & p_{25} & p_{26} & p_{27} & p_{28} & p_{29} & p_{2,10} \\ p_{31} & p_{32} & p_{33} & p_{34} & p_{35} & p_{36} & p_{37} & p_{38} & p_{39} & p_{3,10} \\ p_{41} & p_{42} & p_{43} & p_{44} & p_{45} & p_{46} & p_{47} & p_{48} & p_{49} & p_{4,10} \\ p_{51} & p_{52} & p_{53} & p_{54} & p_{55} & p_{56} & p_{57} & p_{58} & p_{59} & p_{5,10} \\ p_{61} & p_{62} & p_{63} & p_{64} & p_{65} & p_{66} & p_{67} & p_{68} & p_{69} & p_{6,10} \\ p_{71} & p_{72} & p_{73} & p_{74} & p_{75} & p_{76} & p_{77} & p_{78} & p_{79} & p_{7,10} \\ p_{81} & p_{82} & p_{83} & p_{84} & p_{85} & p_{86} & p_{87} & p_{88} & p_{89} & p_{8,10} \\ p_{91} & p_{92} & p_{93} & p_{94} & p_{95} & p_{96} & p_{97} & p_{98} & p_{99} & p_{9,10} \\ p_{10,1} & p_{10,2} & p_{10,3} & p_{10,4} & p_{10,5} & p_{10,6} & p_{10,7} & p_{10,8} & p_{10,9} & p_{10,10} \end{bmatrix}, A_{S,k} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \\ a_9 \\ a_{10} \end{bmatrix}.$$

Let the sample taken at time $(k+1), (k+2)$ and $(k+3)$ respectively estimate the parameters (2,4,7), (3,4,6) and (1,2,4,9). Then,

$$A_{M,k+1} = \begin{bmatrix} a_2 \\ a_4 \\ a_7 \end{bmatrix}, A_{M,k+2} = \begin{bmatrix} a_3 \\ a_4 \\ a_6 \end{bmatrix}, A_{M,k+3} = \begin{bmatrix} a_1 \\ a_2 \\ a_4 \\ a_9 \end{bmatrix}, \text{ and}$$

$$P_{M,k+1} = \begin{bmatrix} p_{22} & p_{24} & p_{27} \\ p_{42} & p_{44} & p_{47} \\ p_{72} & p_{74} & p_{77} \end{bmatrix}, P_{M,k+2} = \begin{bmatrix} p_{33} & p_{34} & p_{36} \\ p_{43} & p_{44} & p_{46} \\ p_{63} & p_{64} & p_{66} \end{bmatrix},$$

$$P_{M,k+3} = \begin{bmatrix} p_{11} & p_{12} & p_{14} & p_{19} \\ p_{21} & p_{22} & p_{24} & p_{29} \\ p_{41} & p_{42} & p_{44} & p_{49} \\ p_{91} & p_{92} & p_{94} & p_{99} \end{bmatrix}$$

The transformation matrices at these instances will be:

$$U_{SM,k+1} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, U_{SM,k+2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, U_{SM,k+3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The error covariance calculated at instant $(k+2)$ is:

$$U_{C01} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, U_{C01}^{T} P_{C,k+1} U_{C01} = \begin{bmatrix} p_{22} & 0 & p_{24} & 0 & p_{27} \\ 0 & 0 & 0 & 0 & 0 \\ p_{42} & 0 & p_{44} & 0 & p_{47} \\ 0 & 0 & 0 & 0 & 0 \\ p_{72} & 0 & p_{74} & 0 & p_{77} \end{bmatrix}$$

$$U_{MC,k+2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, P_{C,k+2} = \begin{bmatrix} p_{22} & 0 & p_{24} & 0 & p_{27} \\ 0 & p_{33} & p_{34} & p_{36} & 0 \\ p_{42} & p_{43} & p_{44} - \bar{p_{44}} & p_{46} & p_{47} \\ 0 & p_{63} & p_{64} & p_{66} & 0 \\ p_{72} & 0 & p_{74} & 0 & p_{77} \end{bmatrix}$$

$$U_{CS,k+2} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}.$$

7.5 Simulation results

7.5.1 Sampling of complex field with centralized AS algorithm using 4 robots along with partitioning of sampling area

A complex fire field of size 124x90 pixels generated using a CA simulation is to be reconstructed in parallel by sampling using 4 robots. An initial approximation is acquired by training the RBF neural network with 30 neurons with spread factor of 25. The numerical covariance assumptions are:

$$\begin{aligned} &P_{b0} = 5, P_{a0} = 5, P_{\sigma 0} = 10^{-4},\ P_{x00} = 5,\ P_{y00} = 5 \\ &R = 0.01,\ n = 15,\ p = 5 \end{aligned} \tag{7.14}$$

Figures 7.9 (b) and (c) respectively show the initial field, and final estimated field after 172 samples. Figure 7.9(e) indicates the locations where the 4 robots took samples, represented by different symbols. It also shows the 4 regions formed using fuzzy *c*-means clustering algorithm and the Voronoi Tessellation Diagram, where different robots sample. Regions are updated after every 20 samples. The partitioning algorithm is implemented with *fcm* and *voronoi* functions of MATLAB. As shown in figure 7.9 (d) and (g), the parametric error drops from 47.7 to 14.83, and the non-parametric error drops from 24.34 to 10.62 by taking 172 samples. The estimate of 120 parameters of 30 Gaussians are updated using the centralized AS algorithm.

The use of 4 robots instead of 1 for sampling also reduces the time for field reconstruction from 6.7 to 1.2 minutes (more than 4 times reduction). The reason can be understood intuitively, because by sampling using 4 robots not only the number of samples per robot reduces, but also navigation time reduces because of a smaller sampling area allocated to each robot.

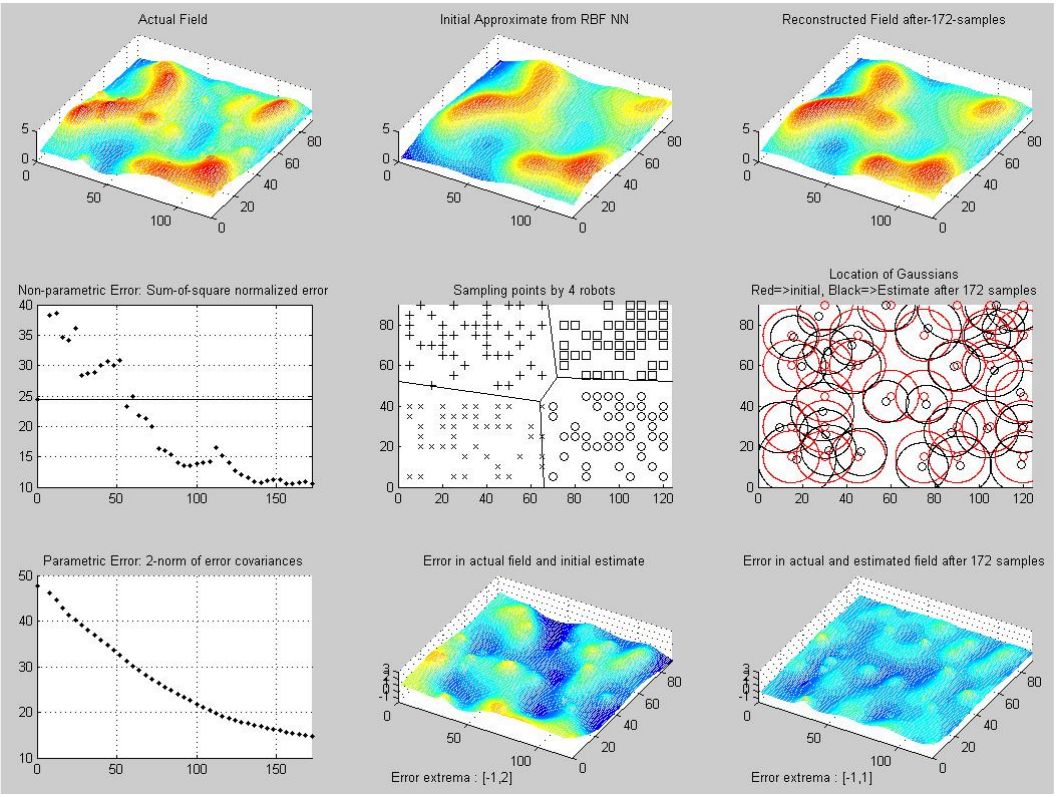


Figure 7.9 Sampling operation divided between 4 robots using GAS algorithm to reconstruct the field. The results plotted are after 172 samples

Table 7.2 illustrates the number of computations and communications involved in sampling of complex field represented with 30 Gaussian RBF, using 4 robot, and considering a decentralized and distributed federated filter. Since the decentralized filter is full-order, it includes estimates of all 120 parameters. The number of computations performed on the fusion filter is same for decentralized and distributed filter. The

combined computations for distributed filter are 675,840 compared to 8,631,840 for decentralized federated filter, e.g. approximately 12 times smaller. Similarly, the number of data communicated in case of distributed filter is 15 times smaller compared to the decentralized filter.

Table 7.2 Comparison of decentralized and distributed federated filter for sampling of a complex field represented with 30 Gaussian RBF using 4 robots

	Computations			**Communication**
	Robot	Fusion Center	Combined	
Decentralized Federated Filter	$p=16, m=10, n=120$ $p(2m^2n+2mn^2+m^3+n^3)$ $=2{,}041{,}000$	$p=16, n=120, N=4$ $p(2n^2+n^2N+nN)$ $=467{,}840$	8,631,840	$pN(n^2+n)$ $=929{,}280$
Distributed Filter	$p=16, m=10, n=30$ $p(2m^2n+2mn^2+m^3+n^3)$ $=52{,}000$	$p=16, n=120, N=4$ $p(2n^2+n^2N+mN)$ $=467{,}840$	675,840	$pN(n^2+n)$ $=59{,}520$

7.6 Experimental results

7.6.1 Sampling of linear color field with centralized AS algorithm using 2 robots

A linear RGB field represented by 9 parameters was considered in the experiments of Chapter 6. The same color field was reconstructed using 2 robots with an AS algorithm running on the central base-station. A star-network was formed, and the base-station runs the adaptive sampling algorithm and communicates with two ARRI-Bots. The base-station uses both robots to estimate the field parameters in half the time if data packet collisions are not encountered. If the sampling space is partitioned apriori, the results shown in Figure 7.10 indicate that the color field is again recovered with reasonable accuracy after 30 combined samples.

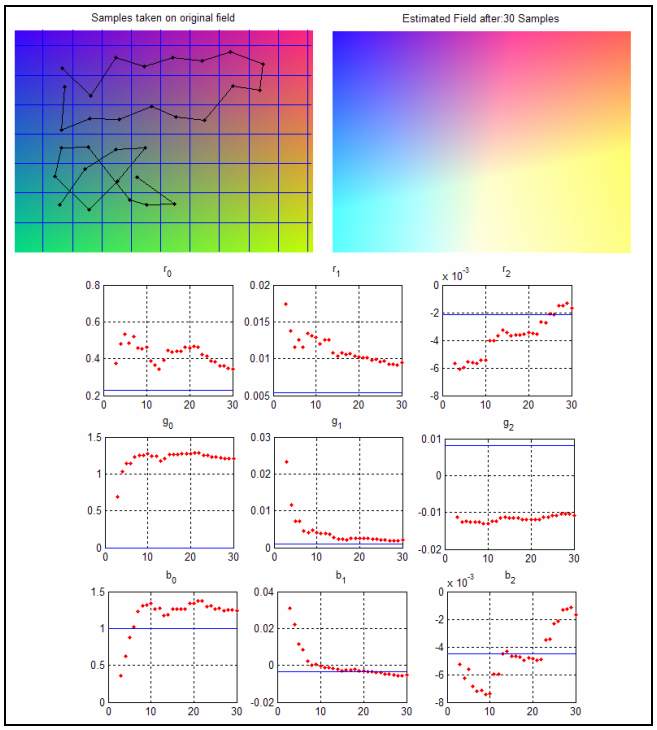


Figure 7.10 Experimental results for GAS with 2 robots showing the estimated field after 30 combined samples

7.6.2 Sampling of complex fire field with centralized AS algorithm using 2 robots

Experimental results are shown in figure 7.11 for a complex fire field projected in an area of 124x90 square inches. Figure 7.11(a) shows the field that is generated using the Cellular Automaton. A low-resolution grid size with n=20 gives 27 (3x3) samples for training the neural network. We use 20 neurons and a spread factor of 25 to approximate the field with 80 parameters (20x4). Two robots sample the field in parallel using the GAS algorithm with grid size p=5 and horizon size 3.

Figure 7.11 (b) and (c) respectively shows the initial field and final estimated field after 118 samples. Figure 7.11(e) indicates the locations by 'x' and 'o' symbols where the 2 robots took samples. As shown in figure 7.11 (d) and (g), the parametric

error drops from 37.6 to 13.91, and the non-parametric error drops from 39.05 to 15.86 by taking 118 samples. This final non-parametric error is close to what we would get if we sample data from an actual field assuming a grid size of 5 (which uses 446 samples) to train the neural network. The numerical assumptions for uncertainties were:

$$P_{b0} = 5, P_{a0} = 5, P_{\sigma 0} = 10^{-4},\ P_{x00} = 5,\ P_{y00} = 5$$
$$R = 0.01 \tag{7.15}$$

The use of 2 robots instead of 1 for sampling also reduces the time for reconstruction from 6.7 to 2.6 minutes (which is less than half). This can be understood intuitively because not only the number of samples per robot reduces, but also navigation time reduces in a smaller sampling area.

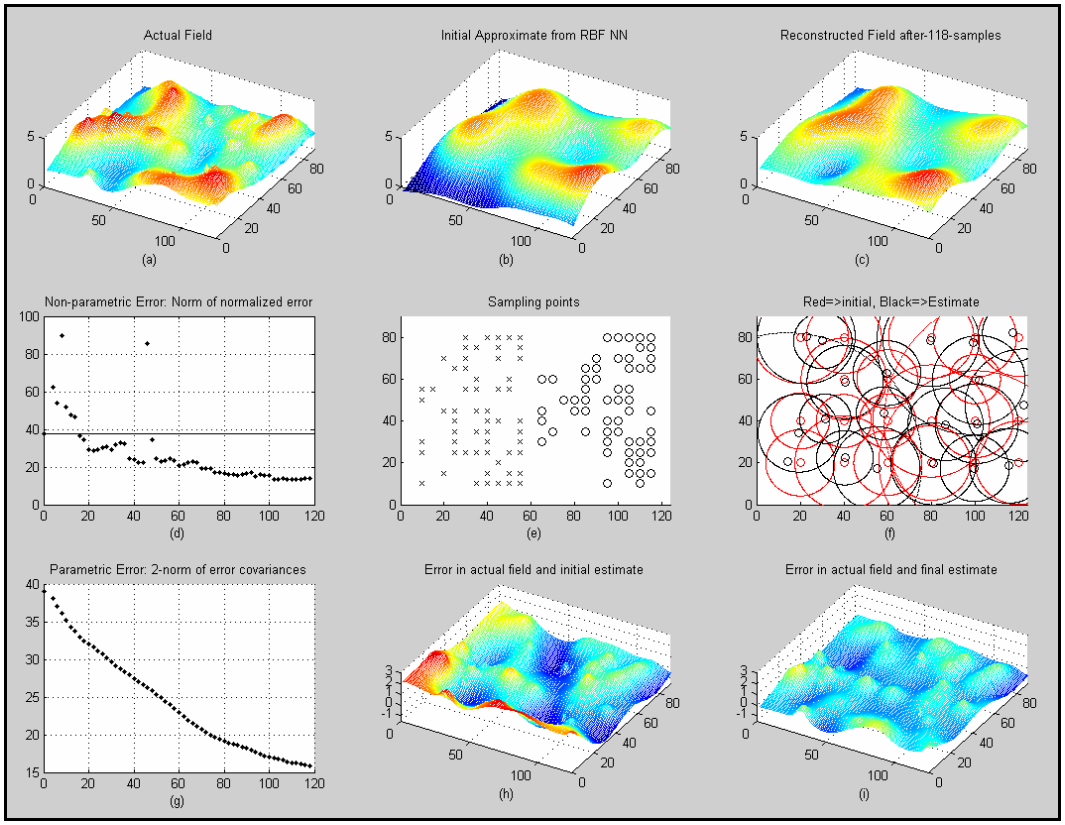


Figure 7.11 Experimental results for estimation of a complex field represented by 20 neurons RBF using 2 robots

In summary, in this Chapter we propose a distributed federated filter, and a clustering scheme for the distribution of computational and communication load among N robots performing sampling missions. Estimates of communication and computational load on the robots show that a dramatic reduction can be expected compared to a centralized sampling scheme, while a reduced sampling time in excess of N times can be expected. Further simulations and experiments are needed to verify the efficiency and convergence properties of the distributed filter scheme we propose.

CHAPTER 8

CONCLUSION & FUTURE WORK

This dissertation presents a Multi-scale Adaptive Sampling (AS) algorithm called "EKF-NN-GAS" for combining measurements from robotic sensors of different scales, rates and accuracies to reconstruct complex spatio-temporal fields. Adaptive sampling algorithms continuously adapt in response to real-time measurements, and direct the robots to locations most likely to yield maximum information about the sensed field. It is demonstrated through extensive simulations and experiments that the proposed AS algorithm provides field reconstruction in significantly smaller number of samples compared to a simple raster scan approach. The sampling time and computational complexity of the algorithm also improves by using a local search greedy adaptive sampling strategy.

In addition, the localization uncertainty of the robots is reduced by combining the location states and field parameters in a Joint-EKF. In the absence of absolute location information such as GPS, inertial localization information from internal sensors can still be used for a short while to build a parametric model of the environment. Later, the acquired model can be used to reduce the location uncertainty of the robots. The

algorithm has been experimentally validated on ARRI-Bots in DIAL Lab (Distributed Intelligence and Autonomy Lab) at ARRI.

Further analysis is needed to evaluate the performance of *non-uniform grid size sampling* and to quantify how they contribute to the objective of collecting more samples in dense regions of parametric field representations. Also, the multi-step GAS needs investigation in terms of how it improves local searching and to prevent it from being trapped in local minima.

It is demonstrated through extensive simulations that the proposed multi-scale sampling approach, along with self-organized clustering and RBF parameterization is an efficient way to accomplish wide-area. Furthermore, this RBF parameterization can model even the fields with sharp variations by selecting appropriate number of neurons and width of the Gaussians using heuristic approaches.

The multi-scale approach of adaptive sampling was also applied to map the forest fires. The proposed algorithm uses a multi-scale, multi-rate approach achieved by taking two types of measurements: a low-resolution (high-spatial) satellite measurement gives an initial estimate of field parameters, which is further refined by high-resolution (low-spatial) sampling. In this thesis, a single measurement of fire field intensity was considered, which, in practice, can be captured using thermal imaging equipment. Due to time-varying nature of the fire, additional measurements that contribute to the spread of fire should also be investigated, such as wind speed and direction, slope, air temperature and humidity, etc. Future work should include the implementation of the sampling algorithms using multi-scale robots equipped with thermal imaging

equipment, topographic mapping, wind, air temperature and other sensors for fire monitoring.

The proposed parametric sampling algorithm in this Thesis uses the standard Extended Kalman Filter. In the future, it can be modified to use an Ensemble Kalman filter (EnKF) which is suitable for problems with large number of variables. EnKF is commonly used in data assimilation algorithms such as weather prediction. EnKF uses sample (or mean) covariance of parameters rather than auto and cross covariance of all the parameters. This makes the filter simpler and easier to implement.

Since we used the Extended Kalman Filter, the proposed sampling schemes must be initialized sufficiently close to the actual field parameters. Overhead satellite imagery of the field provides a reasonable initial estimate, but the absolute algorithm convergence cannot be guaranteed, as widely discussed in the literature. In addition, the use of Heuristic Search methods in the algorithm can also lead to the presence of local minima. We can avoid such minima by restricting the search space so that we do not re-visit already sampled points, but such heuristics may not always work for time-varying fields. While studying convergence conditions for our algorithms was beyond the scope of our thesis, future work should investigate these limitations.

The use of multiple robots for sampling was also investigated in the thesis. It was inferred from simulations that the partitioning of the sampling area reduces the reconstruction time by a factor greater than the number of robots. Furthermore, a distributed multi-robot scheme was proposed for parametric adaptive sampling, resulting in reduction in computations and communication overhead. Every robot

focuses on certain parameters which are unique to itself, or common between itself and other robots. Hence, reaching a consensus only on certain parameters of interest will speed up the computational efficiency of the algorithm. If N is the number of robots used for sampling, then our approach reduces the number of computations and communication per robot by a factor of N^3, when compared to the decentralized approach.

A completely decentralized and distributed approach for adaptive sampling with multiple robots is the best choice. In this case, studying the required information flow through the network such that different robots can reach consensus about the desired parameters in minimum time, should also be addressed in the future. These aspects relate to multi-robot cooperation for AS problems, and include routing, coverage, connectivity, congestion control, communication bandwidth, node energy and mission planning.

Finally, future work should include the implementation of the proposed sampling algorithms in conjunction with information, communication, mission and energy-aware deployment schemes on low-altitude UAVs for fire mapping.

REFERENCES

[1] Anderson B., Moore J., “Optimal Filtering”, Englewood Cliffs, NJ: Prentice-Hall, 1979.

[2] Anderson D. H., Catchpole E. A., DeMestre N. J., Parkes T., “Modeling the spread of grass fires”. J. Austral. Math. Soc. (Ser. B.), vol. 23, p. 451-466, 1982.

[3] Cannell C. J., Stilwell D. J., "A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles", OCEANS, 2005. Proceedings of MTS/IEEE, vol. 2, p. 1514-1521, 2005.

[4] Christopoulos V. N., Roumeliotis S., “Adaptive Sensing for Instantaneous Gas Release Parameter Estimation”, International Conference on Robotics and Automation, 2005, ICRA 2005, Proceedings of the 2005 IEEE, p. 4450-4456, 18-22 Apr. 2005.

[5] Chuvieco E. (Ed.), “Wildland Fire Danger Estimation and Mapping: The Role of Remote Sensing Data”, World Scientific Publishing Company, 2004.

[6] Creed E. L., Glenn S. M., Chant, R. “Adaptive sampling experiment at LEO-15”, OCC 1998 Proceedings, Marine Technology Section, p. 576-579, Nov. 1998.

[7] Elfes A., "Using occupancy grids for mobile robot perception and navigation" Computer, vol. 22, no. 6, p. 46-57, Jun. 1989.

[8] Farrell J. A., Shuo P., Wei L., "Chemical plume tracing via an autonomous underwater vehicle", IEEE Journal of Oceanic Engineering, vol. 30, no. 2, p. 428-442, April 2005.

[9] Finney M. A., "FARSITE: Fire Area Simulator-model development and evaluation", Res. Pap. RMRS-RP-4, Ogden, UT: U.S. Department of Agriculture, Forest Service, Rocky Mountain Research Station. p. 47, 1998.

[10] Fox D., Burgard W., Kruppa H., Thrun S., "A probabilistic approach to collaborative multi-robot localization", Autonomous Robots, Special Issue on Heterogeneous Multi-Robot Systems, vol. 8, no. 3, p. 325-344, 2000.

[11] Ge S. S., Cui Y. J., "Dynamic Motion Planning for Mobile Robots Using Potential Field Method", International Journal of Autonomous. Robots, p. 207-222, Nov. 2002.

[12] Green D. G., "Shapes of simulated fires in discrete fuels". Ecol. Mod., vol. 20, p. 21-32, 1983.

[13] Green D. G., Gill A. M., Noble I. R., "Fire shapes and the adequacy of fire-spread models". Ecol. Mod., vol. 20, p. 33-45, 1983.

[14] Haykin S., "Neural Networks: a Comprehensive Foundation", 2nd. Prentice Hall PTR, 1998.

[15] Hombal V., Sanderson A., Blidberg R., "A Non-Parametric Iterative Algorithm For Adaptive Sampling And Robotic Vehicle Path Planning", International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ, p. 217-222, Oct. 2006.

[16] Howard A., Mataric M. J., Sukhatme G. S., "Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem" in Proc. of 6th International Symposium on Distributed Autonomous Robotic Systems, Fukuoka, Japan, p. 299-308, 2002.

[17] Jatmiko W., Sekiyama K., Fukuda T., "A Mobile Robots PSO-based for Odor Source Localization in Dynamic Advection-Diffusion Environment", International Conference on Intelligent Robots and Systems, 2006, IEEE/RSJ, p. 4527-4532, Oct. 2006.

[18] Johnson E. A., Miyanishi K. (Ed.), "Forest Fires - Behavior and Ecological Effects". Academic Press, San Diego, 2001.

[19] Karafyllidis I., Thanailakis A., "A model for predicting forest fire spreading using cellular automata", Ecol. Mod., vol. 99, p. 87-97, 1997.

[20] Khatib O., "Real-time obstacle avoidance for manipulators and mobile robots", in Proc. IEEE, International Conference on Robotics and Automation, p. 500-505, 25-28 Mar. 1985.

[21] Krogh B. H., "A generalized potential field approach to obstacle avoidance control", in Proc. of International Robotics Research Conference, p.1150-1156, Aug.1984.

[22] Latombe J., "Robot Motion Planning", Kluwer Academic Publishers: Boston, 1991.

[23] Lewis F. L., Xie L., Popa D. O., "Optimal and Robust Estimation: With an Introduction to Stochastic Control Theory", CRC Press, 2007.

[24] Li X., G. A., “Neural-network-based cellular-automata for simulating multiple land use changes using GIS”, International Journal of Geographic Information Science, vol.16, no.4, p.323–344, 2002.

[25] Lippmann R. P., "Pattern classification using neural networks", Communications Magazine, IEEE, vol. 27, no. 11, p. 47-50, 59-64, Nov 1989.

[26] Low K. H., Gordon G. J., Dolan J. M., Khosla P., "Adaptive Sampling for Multi-Robot Wide-Area Exploration", 2007 IEEE International Conference on Robotics and Automation, p. 755-760, 10-14 Apr.2007.

[27] Mandel J., Darema F. (Ed.), “Dynamic data driven wildfire modeling”, Dynamic Data Driven Applications Systems, Klumer Academic Publishers, 2004.

[28] Muzy A., Innocenti E., Aiello A., Santucci J., Wainer G., “Specification of Discrete Event Models for Fire Spreading”, Simulation, vol. 81, no. 2, p. 103-117, 2005.

[29] Olfati-Saber R., "Distributed Kalman Filter with Embedded Consensus Filters", 44th IEEE Conference Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05 p. 8179-8184, 12-15 Dec. 2005.

[30] Popa D. O., “Optimal sampling using singular value decomposition of the parameter variance space”, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, p. 3131- 3136, 2-6 Aug. 2005.

[31] Popa D. O., Mysorewala M. F., Lewis F. L., “EKF-based Adaptive Sampling with Mobile Robotic Sensor Nodes”, International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ, p. 2451-2456, Oct. 2006.

[32] Popa D. O., Mysorewala M. F., Lewis F. L., "Deployment Algorithms and In-Door Experimental Vehicles for Studying Mobile Wireless Sensor Networks", to appear in ACIS International Journal of Sensor Networks, 2009.

[33] Popa D. O., Stephanou H. E., Helm C., Sanderson A. C., “Robotic deployment of sensor networks using potential fields”, in the Proc. Of International Conference on Robotics and Automation, April-May 2004.

[34] Popa D.O, et. al., “Adaptive sampling algorithms for multiple autonomous underwater vehicles”, Autonomous Underwater Vehicles, 2004 IEEE/OES, p. 108- 118, 17-18 June 2004.

[35] Rao B. S., Durrant-Whyte H. F., "Fully decentralised algorithm for multisensor Kalman filtering", Control Theory and Applications, IEE Proceedings, vol. 138, no. 5, p. 413-420, Sep 1991.

[36] Rothermel R.C., “How to predict the spread and intensity of forest and range fires” Gen. Tech. Rep. INT-143. Ogden, UT: U.S. Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station, p. 161, 1983.

[37] Roumeliotis S. I., Bekey G. A., "Distributed Multi-Robot Localization", Distributed Autonomous Robotic Systems 4, Springer Verlag, p. 179-188, 2000.

[38] Russell R. A., Bab-Hadiashar A., Shepherd R. L., Wallace G. G., “A comparison of reactive robot chemotaxis algorithms”, Robotics and Autonomous Systems, vol. 45, no. 2, p. 83-97, 30 Nov. 2003.

[39] Scott D., “Multivariate Density Estimation: Theory, Practice, and Visualization”, Wiley-Interscience, 1992.

[40] Silverman B., “Density Estimation for Statistics and Data Analysis”, Chapman & Hall / CRC, 1986.

[41] Singh A., et.al., “Multiscale Sensing: A new paradigm for actuated sensing of high frequency dynamic phenomena”, International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ, p. 328-335, Oct. 2006.

[42] Trunfio G. A., “Predicting Wildfire Spreading through a Hexagonal Cellular Automata Model”, ACRI, p. 385-394, 2004.

[43] Wagner V., “A simple fire growth model”, Forestry Chron. vol. 45, p. 103-104, 1969.

[44] www.nifc.gov

[45] Nelles O., “Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models”, Springer, 2000.

[46] Hall D. L., McMullen S. A., “Mathematical Techniques in Multisensor Data Fusion”, Artech House Publishers, 2nd edition, 2004.

[47] Abidi M. A., Gonzalez R. C., “Data Fusion in Robotics and Machine Intelligence”, Academic Press, 1992.

[48] Iyengar S. S. (Ed.), Brooks R. R. (Ed.), “Distributed Sensor Networks”, Chapman & Hall/CRC; 1st edition, 2004.

[49] Fedorov V. V., Studden W. J., Klimko E. M., “Theory of Optimal Experiments”, Academic Press Inc., U.S., 1972.

[50] Qiang D. et. al., “Centroidal Voronoi Tessellations: Applications and Algorithms”, SIAM Review, vol. 41, no. 4, pp. 637-676, 1999.

[51] Bezdek J. C., "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.

[52] Park J., Sandberg J. W., “Universal approximation using radial-basis-function networks”, Neural Computations, vol. 3, no. 2, pp. 246-257, 1991.

[53] N. Benoudjit, et. al., “Width optimization of the Gaussian kernels in Radial Basis Function Networks”, ESANN 2002, European Symposium on Artificial Neural Networks, Bruges, 2002, 425-432.

[54] http://en.wikipedia.org/wiki/California_wildfires_of_October_2007

[55] http://www.nasa.gov/vision/earth/lookingatearth/socal_wildfires_oct07.html

[56] Carlson N. A., "Federated square root filter for decentralized parallel processes", IEEE Transactions on Aerospace and Electronics Systems, vol. 26, no. 3, pp. 517-525, 1990.

[57] Rao B. S., Durrant-Whyte H. F., "Fully decentralized algorithm for multisensor Kalman filtering", IEE Proceedings-D, vol. 138, no. 5, pp. 413-420, 1991.

[58] Mutambara A. G., "Decentralized estimation and control for multisensor systems," CRC Press, Boca Raton, Chapters 2-3, pp. 19-79, 1998.

[59] Hashmipour H. R., Roy S., Laub A. J., "Decentralized structures for parallel Kalman filtering," IEEE Transactions on Automatic Control vol. 33, no. 1, pp. 88-93, 1988.

[60] Olfati-Saber R., "Distributed Kalman Filtering for Sensor Networks", Proceeding of the 46th IEEE Conference on Decision and Control, pp. 5492-5498, 12-14 Dec. 2007.

[61] Ren W., Beard R.W.; Kingston D. B., "Multi-agent Kalman consensus with relative uncertainty," Proceedings of the American Control Conference 2005, vol. 3, pp. 1865-1870, 8-10 June 2005.

[62] Tsai C., "A localization system of a mobile robot by fusing dead-reckoning and ultrasonic measurements", Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE, vol.1, pp.144-149, 18-21 May 1998.

[63] Roumeliotis S. I., Bekey G.A., “An extended kalman filter for frequent local and infrequent global sensor data fusion”, in Proceedings of the SPIE (Sensor Fusion and Decentralized Control in Autonomous Robotic Systems), pages 11-22, Oct. 1997.

[64] Kleeman L., "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning", Proceedings of the IEEE International Conference on Robotics and Automation, vol. 3, pp. 2582-2587, 12-14 May 1992.

[65] Tanaka K., Hasegawa T., Zha H, Kondo E., Okada N., "Mobile robot localization with an incomplete map in non-stationary environments", Proceedings of the IEEE International Conference on Robotics and Automation, vol.2, pp. 2848- 2853, 14-19 Sept. 2003.

[66] MourikisA. I., Roumeliotis S. I., "On the treatment of relative-pose measurements for mobile robot localization", Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2277- 2284, May 15-19, 2006.

[67] Goel P., Roumeliotis S. I., Sukhatme G. S., "Robust localization using relative and absolute position estimates", International Conference on Intelligent Robots and Systems, vol.2, pp.1134-1140, 1999.

[68] Durrant-Whyte H., Bailey T., "Simultaneous localization and mapping: part I", Robotics & Automation Magazine, IEEE, vol.13, no. 2, pp. 99-110, June 2006.

[69] Bailey T., Durrant-Whyte H., "Simultaneous localization and mapping (SLAM): part II", Robotics & Automation Magazine, IEEE, vol.13, no.3, pp. 108-117, Sept. 2006.

[70] Roumeliotis S. I., Bekey G. A., "Distributed multirobot localization", IEEE Transactions on Robotics and Automation, vol.18, no.5, pp. 781-795, Oct 2002.

[71] Martinelli A., Pont F., Siegwart R., "Multi-Robot Localization Using Relative Observations", Proceedings of the IEEE International Conference on Robotics and Automation, pp. 2797-2802, 18-22 April 2005.

[72] Roumeliotis, S.I.; Bekey, G.A., "Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots", Proceedings of the IEEE International Conference on Robotics and Automation, pp.2958-2965, 2000.

[73] Russell R.A., “Odour Sensing for Mobile Robots”, World Scientific, 1999.

[74] Christopoulos V.N., Roumeliotis S., "Adaptive Sensing for Instantaneous Gas Release Parameter Estimation", Proceedings of the IEEE International Conference on Robotics and Automation, pp. 4450- 4456, 18-22 April 2005.

[75] Cannell C. J., Stilwell D. J., "A comparison of two approaches for adaptive sampling of environmental processes using autonomous underwater vehicles", OCEANS, 2005, Proceedings of MTS/IEEE, vol. 2, pp. 1514- 1521, 2005.

[76] Rachkov M. Y., Marques L., Almeida A. T., “Multisensor Demining Robot”, Autonomous Robots vol. 18, no. 3, pp. 275-291, May 2005.

[77] Willett R., Martin A., Nowak, R., “Backcasting: adaptive sampling for sensor networks”, In Proceedings of the Third international Symposium on information Processing in Sensor (IPSN '04), ACM Press, New York, NY, pp. 124-133, April 26-27, 2004.

[78] Martinelli A., Tomatis N., Tapus A., Siegwart R., “Simultaneous localization and odometry calibration for mobile robot”, International Conference on Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ, vol. 2, pp. 1499-1504, 27-31 Oct. 2003.

[79] Martinelli A., “The odometry error of a mobile robot with a synchronous drive system”, IEEE Transactions on Robotics and Automation, vol.18, no. 3, pp. 399-405, Jun 2002.

[80] Borenstein J., Feng, L., "UMBmark A Method for Measuring, Comparing, and Correcting Dead-reckoning Errors in Mobile Robots", Technical Report, The University of Michigan UM-MEAM-94-22, December 1994.

[81] Howard A., Kitchen L., "Cooperative localisation and mapping," in International Conference on Field and Service Robotics (FSR99), pp. 92-97, 1999.

[82] Mysorewala M. F., Popa, D. O., Giordano, V., Lewis, F. L., “Deployment Algorithms and In-Door Experimental Vehicles for Studying Mobile Wireless Sensor Networks”, Software Engineering, International Conference on Artificial Intelligence, Networking, and Parallel/Distributed Computing, pp. 290- 298, 19-20 June 2006.

[83] Ham F. M., Kostanic I., “Principles of Neurocomputing for Science & Engineering”, McGraw-Hill Higher Education, 2000.

[84] Van der Heijden, F., Duin, R.P.W., de Ridder, D. and Tax, D.M.J. “Classification, parameter estimation and state estimation: an engineering approach using MATLAB”, John Wiley & Sons, London, UK, 2004.

[85] Rahimi M., Hansen M., Kaiser W. J., Sukhatme G.S., Estrin D., "Adaptive sampling for environmental field estimation using robotic sensors", International Conference on Intelligent Robots and Systems, 2005. (IROS 2005), IEEE/RSJ, pp. 3692- 3698, 2-6 Aug. 2005.

[86] Rahimi M., Pon R., Kaiser W. J., Sukhatme G. S., Estrin D., Srivastava M., "Adaptive sampling for environmental robotics", International Conference on Robotics and Automation, (ICRA '04), vol.4, pp. 3537-3544, April 26-May 1, 2004.

[87] Mupparapu S. S., Chappell S. G., Komerska R. J., Blidberg D. R., Nitzel R., Benton C., Popa, D. O., Sanderson A. C., "Autonomous systems monitoring and control (ASMAC) - an AUV fleet controller", Autonomous Underwater Vehicles, 2004 IEEE/OES, pp. 119-126, 17-18 June 2004.

[88] Carlson E. A., Beaujean P. P., An E., "Simulating communication during multiple AUV operations", Autonomous Underwater Vehicles, 2004 IEEE/OES, pp. 76-82, 17-18 June 2004.

[89] http://www.ausi.org/research/research.html

[90] Corke P., Peterson R., Rus D., “Networked Robots: Flying Robot Navigation using a Sensor Net”, In Proceedings of the Eleventh International Symposium of Robotics Research (ISRR), October, 2003.

[91] Burgard W., Fox D., Moors M, Simmons R., Thrun S., “Collaborative multi-robot exploration”, In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, 2000.

[92] Haas Z. J., "A new routing protocol for the reconfigurable wireless networks", IEEE 6th International Conference on Universal Personal Communications Record, 1997, vol. 2, pp. 562-566, 12-16 Oct 1997.

[93] Wen J. T., Arcak M., "A unifying passivity framework for network flow control", IEEE Transactions on Automatic Control, vol. 49, no. 2, pp. 162-174, Feb. 2004.

[94] Giordano V., Ballal P., Lewis F., Turchiano B., Zhang J. B., "Supervisory Control of Mobile Sensor Networks: Math Formulation, Simulation, and Implementation", IEEE Transactions on Systems, Man and Cybernetics, Part B, vol. 36, no. 4, pp. 806-819, Aug. 2006.

[95] Wilson A. D., “PlayAnywhere: a compact interactive tabletop projection-vision system”, In Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology, Seattle, WA, USA, October 23-26, 2005.

[96] Ashdown M., Flagg M., Sukthankar R., Rehg J, M., “A Flexible Projector-Camera System for Multi-Planar Displays”, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), Jun. 2004.

[97] Ishida H., Ushiku T., Toyama S., Taniguchi H., Moriizumi T., "Mobile robot path planning using vision and olfaction to search for a gas source", Sensors, 2005 IEEE, pp. 4-30, Nov. 2005.

[98] Jatmiko W., Ikemoto Y., Matsuno T., Fukuda T., Sekiyama K., "Distributed odor source localization in dynamic environment", Sensors, 2005 IEEE, 30 Oct.-3 Nov. 2005.

[99] Christopoulos V. N., Roumeliotis S., "Multi Robot Trajectory Generation for Single Source Explosion Parameter Estimation", IEEE International Conference on Robotics and Automation, 2005, ICRA 2005, pp. 2803-2809, 18-22 April 2005.

[100] Lizarralde F., Nunes E. V. L., Liu H., Wen J. T., "Mobile robot navigation using sensor fusion", IEEE International Conference on Robotics and Automation, ICRA '03, vol.1, pp. 458-463, 14-19 Sept. 2003.

[101] Kurazume R., Nagata S., “Cooperative positioning with multiple robots”, in IEEE ICRA, pp. 1250-1257, 1994.

[102] Kleeman L., "Optimal estimation of position and heading for mobile robots using ultrasonic beacons and dead-reckoning," IEEE International Conference on Robotics and Automation, vol. 3, pp.2582-2587, 12-14 May 1992.

[103] Curran A., Kyriakopoulos K. J., "Sensor-based self-localization for wheeled mobile robots", IEEE International Conference on Robotics and Automation, vol. 1, pp. 8-13, 2-6 May 1993.

[104] Leonard J. J., Durrant-Whyte H. F., "Mobile robot localization by tracking geometric beacons", IEEE Transactions on Robotics and Automation, vol. 7, no. 3, pp. 376-382, June 1991.

[105] Fabrizi E., Oriolo G., Panzieri S., Ulivi G., "A KF-Based Localization Algorithm for Nonholonomic Mobile Robots", 6th IEEE Mediterranean Conference on Control and Systems, Alghero, Italy, pp. 130-135, June 9-11, 1998.

[106] Fabrizi E., Oriolo G., Panzieri S., Ulivi G., "Mobile robot localization via fusion of ultrasonic and inertial sensor", 8th International Symposium on Robotics with Application, Maui, HI, 11-16 June, 2000.

[107] Georgiev A., Allen P. K., "Vision for mobile robot localization in urban environments", IEEE/RSJ International Conference on Intelligent Robots and System, 2002, vol. 1, pp. 472-477, 2002.

[108] Roumeliotis S. I., Bekey G. A., "Bayesian estimation and Kalman filtering: a unified framework for mobile robot localization", IEEE International Conference on Robotics and Automation, vol. 3, pp.2985-2992, 2000.

[109] Feng L., Borenstein J., Everett B., "Where am I? Sensors and Methods for Autonomous Mobile Robot Localization", Technical Report, The University of Michigan UM-MEAM-94-21, December 1994.

[110] Borenstein J., Feng L., "Measurement and Correction of Systematic Odometry Errors in Mobile Robots", IEEE Journal of Robotics and Automation, vol. 12, no 6, pp. 869-880, December 1996.

[111] Borenstein J., Feng L., "Correction of Systematic Odometry Errors in Mobile Robots", Proceedings of the 1995 International Conference on Intelligent Robots and Systems (lROS '95), Pittsburgh, Pennsylvania, pp. 569-574, August 5-9, 1995.

[112] Borenstein J., Feng L., "UMBmark: A Benchmark Test for Measuring Odometry Errors in Mobile Robots", Proceedings of the SPIE Conference on Mobile Robots, Philadelphia, 22-26 October, 1995.

[113] Borenstein J., Everett B., and Feng L., "Navigating Mobile Robots: Systems and Techniques", A. K. Peters, Ltd., Wellesley, MA, ISBN 1-56881-058-X, February 1996.

[114] Borenstein J., Everett H., Feng L., Wehe, D., “Mobile robot positioning: Sensors and techniques”, in Journal of Robotic Systems in special issue on Mobile Robots, vol, 14, no. 4, pp. 231-249, 1997.

[115] http://www-personal.umich.edu/~johannb/mypapers.htm

[116] von der Hardt H. J., Wolf D., Husson R., "The dead reckoning localization system of the wheeled mobile robot ROMANE", IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 603-610, 8-11 Dec 1996.

[117] Martinelli A., Tomatis N., Tapus A., Siegwart R., "Simultaneous localization and odometry calibration for mobile robot", IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 2, pp. 1499-1504, 27-31 Oct. 2003.

[118] Fenwick J. W., Newman P. M., Leonard J. J., "Cooperative concurrent mapping and localization", IEEE International Conference on Robotics and Automation, vol. 2, pp. 1810-1817, 2002.

[119] Howard A., Matark M. J., Sukhatme G. S., "Localization for mobile robot teams using maximum likelihood estimation", IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, pp. 434-439, 2002.

[120] Rekleitis I. M., Dudek G., Milios E. E., "Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy", IEEE/RSJ International Conference on Intelligent Robots and System, vol. 3, pp. 2690-2695, 2002.

[121] Roumeliotis S. I., Bekey G. A., "Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots", IEEE International Conference on Robotics and Automation, vol. 3, pp.2958-2965, 2000.

[122] Mourikis A. I., Roumeliotis S. I., "Analysis of positioning uncertainty in reconfigurable networks of heterogeneous mobile robots", International Conference Robotics and Automation, vol. 1, pp. 572-579, 26 April-1 May 2004.

[123] Roumeliotis S. I., Rekleitis I. M., "Analysis of multirobot localization uncertainty propagation", International Conference on Intelligent Robots and Systems, 2003, vol. 2, pp. 1763-1770, 27-31 Oct. 2003.

[124] Mourikis A. I., Roumeliotis, S.I., "Performance analysis of multirobot Cooperative localization", IEEE Transactions on Robotics, vol. 22, no. 4, pp. 666-681, Aug. 2006.

[125] Hidaka Y. S., Mourikis A. I., Roumeliotis S. I., "Optimal Formations for Cooperative Localization of Mobile Robots", IEEE International Conference on Robotics and Automation, pp. 4126-4131, 18-22 April 2005.

[126] Roumeliotis S. I., Bekey G. A., "Collective localization: a distributed Kalman filter approach to localization of groups of mobile robots", IEEE International Conference on Robotics and Automation, 2000, vol. 3, pp. 2958-2965, 2000.

[127] Martinelli A., Pont F., Siegwart R., "Multi-Robot Localization Using Relative Observations", International Conference on Robotics and Automation, pp. 2797-2802, 18-22 April 2005.

[128] Roumeliotis S. I., “Robust mobile robot localization: From single-robot uncertainties to multi-robot interdependencies”, Ph.D. dissertation, University of Southern California, 2000.

[129] Anousaki G.C., Kyriakopoulos K. J., "Simultaneous localization and map building for mobile robot navigation", IEEE Robotics & Automation Magazine, vol. 6, no. 3, pp.42-53, Sep 1999.

[130] Durrant-Whyte H., Bailey T., "Simultaneous localization and mapping: part I", IEEE Robotics & Automation Magazine, vol. 13, no. 2, pp. 99-110, June 2006.

[131] Csorba M., “Simultaneous Localisation and Mapping”, PhD thesis, University of Oxford, 1997.

[132] Betke M., Gurvits L., "Mobile robot localization using landmarks", IEEE Transactions on Robotics and Automation, vol. 13, no. 2, pp.251-263, Apr 1997.

[133] Armesto L., Ippoliti G., Longhi S., Tornero J., "FastSLAM 2.0: Least-Squares Approach", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.5013-5018, Oct. 2006.

[134] Bailey T., Nieto J., Nebot E., "Consistency of the FastSLAM algorithm", IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 424-429, 15-19 May 2006.

[135] Stachniss C., Hahnel D., Burgard W., "Exploration with active loop-closing for FastSLAM", IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2004, vol.2, pp. 1505-1510, 28 Sept.-2 Oct. 2004.

[136] Xiao W., Xie L., Lin J., Li J., "Multi-Sensor Scheduling for Reliable Target Tracking in Wireless Sensor Networks", 6th International Conference on ITS Telecommunications Proceedings, pp.996-1000, June 2006.

[137] Gao Y., Krakiwsky E. Y., Abousalem M. A., Mclellan J. F., “Comparison and analysis of centralized, decentralized, and federated filters”, Navigation. Vol. 40, no. 1, pp. 69-86. Spring 1993.

[138] Thompson S. K., “Sampling”, Wiley-Interscience, 2002.

[139] http://www.parallax.com

[140] Popa D. O., Sreenath K., Lewis F. L., "Robotic Deployment for Environmental Sampling Applications", in International Conference on Control and Automation, pp. 197-202, 2005.

[141] Sreenath K., “Adaptive Sampling with mobile WSN”, M.S. Thesis, The University of Texas at Arlington, December 2005.

[142] Ghadigaonkar J. S., “ARRI-Bot: An enabler for studying mobile wireless sensor network”, M.S. Thesis, The University of Texas at Arlington, August 2007.

[143] Priyantha N. B., Chakraborty A., Balakrishnan, H. “The Cricket location-support system”. In Proceedings of the 6th Annual international Conference on Mobile Computing and Networking, MobiCom '00, 06-11 August 2000.

[144] Arici T., Altunbasak Y., "Adaptive sensing for environment monitoring using wireless sensor networks," in Wireless Communications and Networking Conference, pp. 21-25, 2004.

[145] Petriu E. M., Whalen T. E., Abielmona R., Stewart A., "Robotic sensor agents: a new generation of intelligent agents for complex environment monitoring", IEEE Instrumentation & Measurement Magazine, vol. 7, pp. 46-51, September 2004.

[146] Zhao F., Shin J., Reich J., "Information-driven dynamic sensor collaboration", IEEE Signal Processing Magazine, vol. 19, pp. 61-72, March 2002.

[147] Zhao F., Liu J., Gibas L., Reich J., "Collaborative signal and information processing: an information-directed approach", Proceedings of the IEEE, vol. 91, pp. 1199-1209, August 2003.

[148] Majumdar S. J., Bishop C. H., Etherton B. J., Toth Z., "Adaptive sampling with the ensemble transform Kalman filter. II. Field program implementation", Mon. Weather Rev., vol. 130, pp. 1356-1369, May 2002.

[149] Bennett A. F., “Inverse Modeling of the Ocean and Atmosphere”, UK: Cambridge University Press, 2002.

[150] Casbeer D. W., Beard R. W., McLain T. W., Sai-Ming L., Mehra R. K., "Forest fire monitoring with multiple small UAVs", Proceedings of the American Control Conference, vol. 5, pp. 3530-3535, 8-10 June 2005.

[151] Goris M. J., Gray D. A., Mareels I. Y., "Reducing the computational load of a Kalman filter" Electronics Letters, vol. 33, no.18, pp.1539-1541, 28 Aug 1997.

[152] Huang S., Dissanayake G., "Convergence and Consistency Analysis for Extended Kalman Filter Based SLAM", Robotics, IEEE Transactions on Robotics, vol. 23, no. 5, pp.1036-1049, Oct. 2007.

[153] Mysorewala M., Popa D., Lewis F., "Multi-scale Adaptive Sampling with Mobile Agents for Mapping of Forest Fires", to appear in the Journal of Intelligent and Robotic Systems, 2008.

[154] Popa D., Mysorewala M., Lewis F., "Deployment Algorithms and In-Door Experimental Vehicles for Studying Mobile Wireless Sensor Networks", to appear in ACIS International Journal of Sensor Networks, 2009.

[155] Mysorewala M., Popa D., "Multi-scale Adaptive Sampling with Mobile Agents for Mapping of Forest Fires", accepted for the International Conference on Wireless Networks, Las Vegas, Nevada, 2008.

[156] Talati R., “A Potential field approach to multiple robot formation control”, M.S. Thesis, The University of Texas at Arlington, August 2007.

[157] Desai J., Ostrowski J., Kumar V., “Modeling and control of formations of non-

holonomic mobile robots", IEEE Transactions on Robotics and Automation, vol. 17, pp. 905-908, Dec. 2001.

BIOGRAPHICAL INFORMATION

Muhammad Faizan Mysorewala received his Bachelor degree in Electrical Engineering from NED University, Karachi, Pakistan in 1999. He worked as Design Engineer in Siemens Company in Power Engineering division for a year. He then joined the University of Texas at Arlington from where he received his Master of Science degree in Electrical Engineering focusing in Communications in 2002. He joined Automation and Robotics Research Institute for doing research in the area of Mobile Wireless Sensor Network. He received his Ph.D. degree in Electrical Engineering in 2008. His research interests are in the area of robotic deployment of sensor network for environmental mapping using statistical signal processing, system modeling and identification, time-series analysis, estimation and optimization. He also had several short term assignments of academic tutoring, and teaching assistantship for graduate and undergraduate level courses.