

# Autonomous Localization of $1/R^2$ Sources Using an Aerial Platform

Eric T. Brewer

Thesis submitted to the Faculty of the  
Virginia Polytechnic Institute and State University  
in partial fulfillment of the requirements for the degree of

Master of Science  
in  
Mechanical Engineering

Kevin B. Kochersberger, Chair  
Steve C. Southward  
Cornel Sultan

December 9, 2009  
Blacksburg, Virginia

Keywords: Source Localization, Autonomous, VTOL, UAV  
Copyright 2009, Eric T. Brewer

# Autonomous Localization of $1/R^2$ Sources Using an Aerial Platform

Eric T. Brewer

## ABSTRACT

Unmanned vehicles are often used in time-critical missions such as reconnaissance or search and rescue. To this end, this thesis provides autonomous localization and mapping tools for  $1/R^2$  sources. A “ $1/R^2$ ” source is one in which the received intensity of the source is inversely proportional to the square of the distance from the source. An autonomous localization algorithm is developed which utilizes a particle swarm particle filtering method to recursively estimate the location of a source.

To implement the localization algorithm experimentally, a command interface with Virginia Tech’s autonomous helicopter was developed. The interface accepts state information from the helicopter, and returns command inputs to drive the helicopter autonomously to the source. To make the use of the system more intuitive, a graphical user interface was developed which provides localization functionality as well as a waypoint navigation outer-loop controller for the helicopter. This assists in positioning the helicopter and returning it home after the the algorithm is completed.

An autonomous mapping mission with a radioactive source is presented, along with a localization experiment utilizing simulated sensor readings.

This work is the first phase of an on-going project at the Unmanned Systems Lab. Accordingly, this thesis also provides a framework for its continuation in the next phase of the project.

# Acknowledgments

I would like to extend my sincere thanks to everyone who helped to make this thesis possible.

I would first like to thank all of my committee members for their support and suggestions throughout the process. I would especially like to thank Dr. Kochersberger for the opportunity to work in the Unmanned Systems Lab on this project. He was never too busy to discuss ways to improve my work, and his support and dedication have been a huge help.

This work would not have been possible if it weren't for the project sponsors (who shall remain unnamed). I also owe a special thanks to Jim Curry and Steve Martin from Sandia National Labs. They were a big help in my understanding of radiation detection and mapping technology.

I would also like to thank all of my fellow graduate students at the Unmanned Systems Lab. Everyone was a big help in one way or another; whether it was assisting with my research, or just taking a break to play football or jump R/C cars.

I would like to extend a very special thanks to my parents, who made college possible for me and always provided constant love, support, and encouragement throughout my life.

Finally, I would like to thank my fiancé Beth for all her love and support. I have had very little spare time over the past year and a half, and I really appreciate her patience as I completed my graduate work.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Original Contributions . . . . .	2
<b>2</b>	<b>Literature Review</b>	<b>3</b>
2.1	Source Types . . . . .	3
2.2	Localization Techniques . . . . .	4
<b>3</b>	<b>Area Radiation Mapping Using an Autonomous Helicopter</b>	<b>8</b>
3.1	Detector Technology Overview . . . . .	8
3.2	Scan Pattern . . . . .	11
3.3	Aerial Platform . . . . .	14
3.4	Flight Operations at the Unmanned Systems Laboratory . . . . .	17
3.5	Experimental Setup . . . . .	19
3.6	Scan Results . . . . .	20
<b>4</b>	<b>Autonomous Localization Algorithm</b>	<b>22</b>
4.1	Scenario . . . . .	22
4.2	Algorithm Selection . . . . .	23
4.2.1	Recursive Bayesian Estimation . . . . .	23
4.2.2	Representing the Belief Space . . . . .	24
4.2.3	Computational Considerations . . . . .	25



4.3	The Particle Filter . . . . .	27
4.3.1	Background . . . . .	27
4.3.2	1-D Example . . . . .	27
4.4	Adapting the Particle Filter For this Application . . . . .	31
4.4.1	Defining the Weight Update Equation . . . . .	31
4.4.2	Resampling . . . . .	34
4.5	Control Strategy . . . . .	37
<b>5</b>	<b>Helicopter Software Interface Development</b>	<b>41</b>
5.1	Hardware Setup . . . . .	41
5.2	Navigation Solution . . . . .	43
5.3	Control Algorithms . . . . .	44
5.4	Proposed Scan Techniques . . . . .	46
5.5	Graphical User Interface . . . . .	46
<b>6</b>	<b>Experiment with Simulated Sensor Data</b>	<b>48</b>
6.1	Simulation Results . . . . .	48
6.2	Experimental Setup and Results . . . . .	50
<b>7</b>	<b>Future Considerations</b>	<b>53</b>
7.1	Point Source Experiment . . . . .	53
7.2	Sources of Unknown Intensity . . . . .	54
7.3	Proposed Scanning Strategies . . . . .	54
7.4	Multiple Source Localization . . . . .	55
<b>8</b>	<b>Conclusions</b>	<b>57</b>
	<b>Bibliography</b>	<b>59</b>

# List of Algorithms

1	Particle Filter . . . . .	28
2	Particle Movement Algorithm . . . . .	30
3	Resampling Algorithm . . . . .	35

# List of Figures

2.1	Sound Inverse Square Law . . . . .	4
2.2	Ghost Measurements . . . . .	7
3.1	Detector Model . . . . .	9
3.2	Static Detector Experiments . . . . .	10
3.3	Raster Scan Pattern . . . . .	11
3.4	Scan Angle . . . . .	12
3.5	Adapted Raster Scan Pattern . . . . .	13
3.6	Intensity Readings at Various Altitudes . . . . .	14
3.7	Virginia Tech’s RMAX Helicopter . . . . .	15
3.8	Yamaha RMAX with Radiation Detector Pod . . . . .	16
3.9	WePilot Flight Controller . . . . .	16
3.10	The WePilot Ground Station Software . . . . .	17
3.11	RMAX Helicopter Before Flight Test . . . . .	18
3.12	Ground Command Station . . . . .	18
3.13	Radiation Mapping Communications Architecture . . . . .	20
3.14	Scan Results . . . . .	21
4.1	Recursive Bayesian Estimation . . . . .	24
4.2	Gaussian Belief Distribution . . . . .	25
4.3	Non-Gaussian Belief Distribution . . . . .	25
4.4	Belief Representations . . . . .	26

4.5	Recursive Bayesian Estimation Algorithms . . . . .	27
4.6	1-D Particle Filter Example . . . . .	29
4.7	1-D Particle Filter Example . . . . .	32
4.8	Particle Belief Distributions . . . . .	33
4.9	Unsuccessful Resampling Algorithm . . . . .	35
4.10	Successful Resampling Algorithm . . . . .	36
4.11	Particle Filtering Algorithm . . . . .	37
4.12	Control Strategies . . . . .	38
4.13	Control Toward Estimate . . . . .	39
4.14	Control Tangent To Estimate . . . . .	40
5.1	Communications Architecture Flow Chart . . . . .	41
5.2	Communications Architecture . . . . .	42
5.3	Ground-based Error Testing . . . . .	44
5.4	Google Maps Image . . . . .	44
5.5	Ground Control Software . . . . .	47
6.1	Source Locations for Testing . . . . .	49
6.2	Experiment With Simulated Sensor Data . . . . .	51
7.1	Directionality of the Detector . . . . .	55
7.2	Directional Detector Technology . . . . .	56

# List of Tables

3.1	Scan Study . . . . .	13
3.2	RMAX Specifications . . . . .	15
6.1	Localization Statistics Static Source Location . . . . .	49
6.2	Localization Statistics Varying Source Locations . . . . .	50

# Chapter 1

## Introduction

In recent years, unmanned vehicles have taken on increasingly diverse roles. This thesis explores the use of an unmanned aerial vehicle to assist in mapping and localizing  $1/R^2$  sources. A  $1/R^2$  source is one in which the received intensity at a distance  $R$  is inversely proportional to the square of the distance from the source. In other words, for an intensity  $I$  at a distance  $R$ , the received intensity would be one-fourth  $I$  at a distance  $2R$ , and one-ninth  $I$  at a distance  $3R$ . Accordingly, the localization of these types of sources can be difficult as the intensity may be very small when the distance becomes large. Some example sources that follow this spreading model include sound, light, and gamma radiation. The experiments presented in this thesis deal primarily with radioactive sources, although the presented algorithms can be used for any  $1/R^2$  source.

### 1.1 Motivation

As with any type of robot, autonomous vehicles are often utilized for dirty, dull, or dangerous missions. They also offer greater precision and efficiency than may be possible by a human operator. Accordingly, the presented algorithms are intended to aid in time-critical missions

such as mapping, target localization, and search-and-rescue. In the case of radioactive source localization, if the levels are considered unsafe for humans, an autonomous vehicle can be utilized to protect human operators.

## **1.2 Original Contributions**

This thesis presents the development of an autonomous localization algorithm for a single unshielded point source. The design process included the development of a particle swarm particle filtering algorithm for autonomous localization as well as a software interface with VT's autonomous helicopter. The interface allowed for the experimental implementation of the algorithm. Additionally, a graphical user interface was developed to make the algorithm execution more intuitive. Autonomous mapping experiments as well as static detector testing with a radiation detector provided by a national laboratory were performed. The final experimental contribution of this thesis is an autonomous localization experiment using simulated radiation data.

As this work is the first phase of a long-term project, the thesis concludes with a chapter on future research directions. This section outlines future developments to extend the algorithms for multiple sources, multiple distributions, and sources of unknown intensity.

# Chapter 2

## Literature Review

There has been significant research in the field of autonomous localization of sources. This chapter reviews some prior work.

### 2.1 Source Types

As mentioned above, this thesis focuses primarily on  $1/R^2$  sources. To elaborate on the brief definition provided in Chapter 1, the term  $1/R^2$  refers to the intensity of the free-space spreading model for the source at a distance  $R$ . For example, the equation for intensity of a wireless radio at distance  $R$  is given by Equation 2.1.

$$S_r(r) = \frac{P_t G_t G_r \lambda}{(4\pi)^2 R^2 L} = k \frac{P_t}{R^2} \quad (2.1)$$

where  $P_t$  is the emitting intensity,  $S_r(r)$  represents the received intensity, and  $G_t, G_r, L$ , and  $\lambda$  are all constants [1]. Notice that the received intensity  $S_r$  is inversely proportional to the square of the distance  $R$  from the source. The reason for this is that models predict the intensity level in a given area on a sphere surrounding the source. Therefore doubling the



area would reduce the intensity by a factor of four. Figure 2.1 displays this concept. Other sources that follow this model include sound, light, and radioactive materials. Each has a similar propagation equation to Equation 2.1.

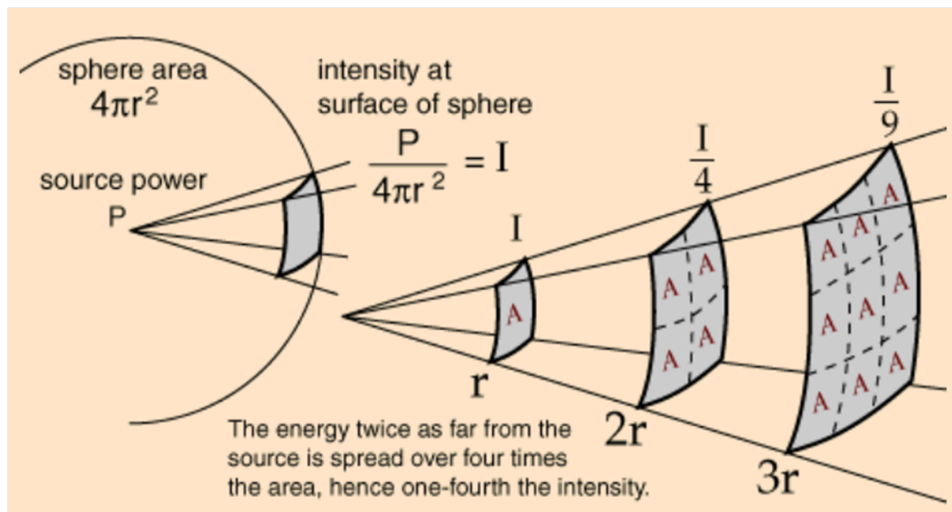


Figure 2.1: Sound Inverse Square Law

<http://hyperphysics.phy-astr.gsu.edu/HBASE/Acoustic/invsqs.html>, with Permission

## 2.2 Localization Techniques

There are several different approaches to source localization. If a directional detector is available, then direction-of-arrival (DoA) or angle-of-arrival (AoA) techniques can be used[2]. Localization techniques become more difficult when a non-directional detector is utilized. Time-of-arrival (ToA) and time-difference-of-arrival (TDoA) are techniques which take advantage of time of propagation. These particular techniques may be difficult to implement for sources that travel at the speed of light, as the time differences would be very small and hence great precision would be required.

Another common method for localization is using the received signal strength. In [3], a unique approach is presented that utilizes four arbitrary points to estimate the location of a radioactive point source. The authors include similar work in [4], which presents three algorithms for localizing a single radiological point source. The paper presents a Cramer-Rao

Bound (CRB) analysis to quantify the maximum accuracy with which it is possible to localize the source and estimate its intensity. Three localization algorithms were applied to trial data: the maximum likelihood estimator (MLE), the Extended Kalman Filter (EKF), and the Unscented Kalman Filter (UKF). The MLE provided the best results, but is computationally intensive, and likely would be difficult to run in real-time for complex multi-source scenarios. The UKF results were not as accurate as the MLE, and the EKF was divergent.

The authors of [5] present an automated sequential search algorithm for locating weak point sources. The authors suggest that “existing techniques for autonomous mapping and searching that are based on gradient following will fail”, as there may be no statistical significance to the measurements of weak point sources. Instead, the proposed method is to travel at some maximum scanning speed until an increased number of emissions is observed, and then the robot will decelerate to a slower speed such that a definitive conclusion can be drawn about the existence of a source. Thus, the motion of the robot is controlled by the acquired radiation measurements. The search area is divided into several cells, and each cell is searched until a statistically definitive conclusion can be drawn determining whether a source is present in the cell. Simulation results are provided which verify the efficacy of the approach. The authors follow with successful experimental results using a CsI crystal [6].

The authors of [7] present a method of localizing a point source of radiation by using four sensors. In this case, the sensors are stationary and the source is moving. They use a least-squares approach for the estimation. The authors state that least squares estimates work even when the model is nonlinear, while the EKF will break down due to discontinuities in the first derivative of the sensor model.

A particle filter is utilized in [8] to estimate the position of 5 wireless radios. An autonomous helicopter is used as a beacon that transmits a signal to each of the 5 wireless nodes. An RSSI (Received Signal Strength Indication) reading is then used in a particle filtering algorithm to determine the location of the nodes. The algorithm is successful in

localizing the radios within about 5 meters in a  $360\text{ m}^2$  area.

In [9], wireless local area network (LAN) is used to localize a Personal Digital Assistant (PDA). Three wireless access points (WAPs) are positioned around the PDA, and its position is determined by measuring the signal strength at each WAP. The approach presented is to create a database of signal strengths at each location, and then locate the user by correlating measured signal strengths to the database. This method is suitable for situations where the source distribution is known a priori.

In [10], an RoSD or Ratio of Square Distances approach is used to localize weak radioactive sources. The presented approach is to use a large number of static sensors and fuse the data from each. Because a relatively weak point source was localized, the sensors had to be no more than 4 feet from the source. The sensor of choice was a Cadmium Zinc Telluride (CZT) detector. They also implement an iterative pruning algorithm to deal with “ghost” measurements. Ghost measurements arise because a non-directional detector is utilized. Figure 2.2 illustrates this concept. If the helicopter flies the along the dotted line, the received intensities would be identical for the ghost source and the true source. Using the techniques presented, the authors are able to achieve an accuracy of 32.5 meters using a sensor density of about 1 per  $1100\text{ m}^2$ .

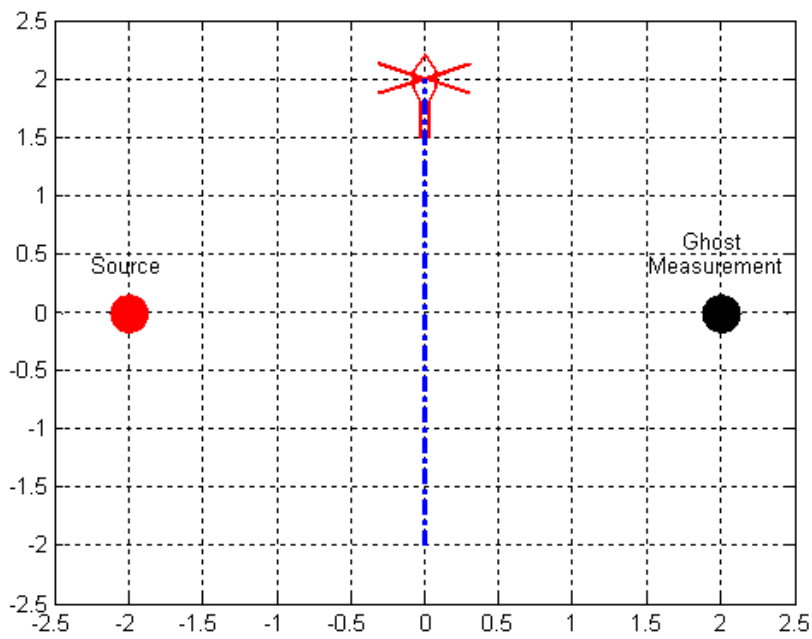


Figure 2.2: Ghost Measurements

# Chapter 3

## Area Radiation Mapping Using an Autonomous Helicopter

This chapter details the development of tools to be used for autonomous radiation mapping. The maps generated with these tools will be useful in identifying areas of interest for the lower-altitude localization algorithm, as well as provide insight into the overall radiation levels in the given area.

### 3.1 Detector Technology Overview

The radiation detector used for the mapping flights was developed by a national laboratory. The detector is a cylindrical crystal which converts incoming radiation into faint flashes of light. The light is then amplified using a photomultiplier tube. The end measurement provided by the detector is “counts” integrated over a one-second interval. The integration occurs because it is difficult to distinguish between individual pulses, thus the total energy is averaged over the time interval. The number of counts corresponds to the amount of energy received by the detector over the interval. Because the scanning is an integration of received

counts, slower scans will provide better resolution in the area of interest. A more complete overview of detector technology is provided in [11].

In addition to providing the detector, the national laboratory also developed a model of the expected detector response. The detector model predicts received count rate by projecting the area of the detector onto a sphere surrounding the point source. The number of received counts is proportional to the area that the detector covers on that sphere. Figure 3.1 shows the parameters used for the radiation model.

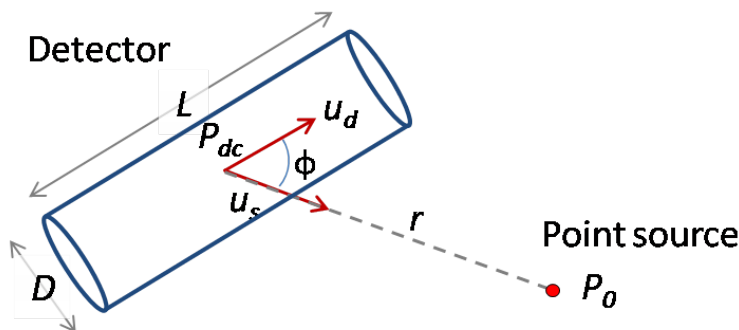


Figure 3.1: Detector Model

Figure courtesy of Steve Martin, Sandia National Labs (SNL), With Permission

Equation 3.1 is used to calculate the expected count rate (counts/second) at a distance  $r$  and angle  $\phi$  from the source.

$$R_{det} = \sum_m N(m) e^{\alpha_m r} \left( \frac{DLF(E_m) |\sin(\phi)| + \pi \left(\frac{D}{2}\right)^2 |\cos(\phi)|}{4\pi r^2} \right) \quad (3.1)$$

where  $N(m)$  is the source gamma emission rate at energy  $E_m$ ,  $F(E_m)$  is the fraction of incident gammas absorbed,  $\alpha_m$  is the air attenuation, and all other parameters are defined in Figure 3.1. Notice that the angle  $\phi$  determines how much of each surface of the detector is exposed to the radiation. The angle determines how much of the side of the cylinder is facing the source versus how much of each end is exposed.

To test the accuracy of the model, static tests were performed with a weak collimated source provided by Virginia Tech. Figure 3.2 shows a plot of detector count rate versus

distance from the source. The detector was placed in the back of a vehicle oriented with the longitudinal axis of the cylinder aligned with the source. The theoretical counts for this orientation are shown in blue. The predicted counts for the side aligned with the source are shown in black. As the figure shows, the measured counts were between the two models. The most likely cause of the discrepancy is ground scatter, which can cause a significant increase in counts received. The collimation of the source can cause additional scattering effects. This effect is not expected to be nearly as severe for mapping and localization flights because the scattering effects are significantly decreased with altitude. Further testing on the helicopter with an altitude measurement device is planned to characterize the model accuracy.

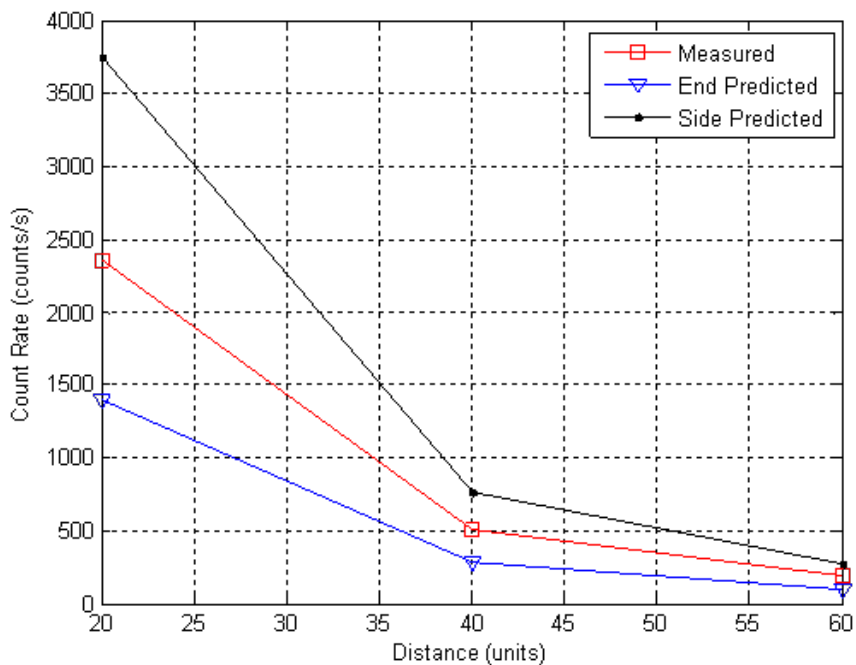


Figure 3.2: Static Detector Experiments

For the scenarios presented in this thesis, only the gross count rate from the detector is used. The detector also provides spectral data which breaks the received counts into separate energy bins. The spectral data can be used to determine what type of source is emitting the radiation. The use of this technology is beyond the scope of this thesis.

## 3.2 Scan Pattern

For the high-level mapping flight, the selection of an appropriate scan pattern can greatly influence the efficiency and effectiveness of the map generation. A raster scan pattern (Figure 3.3) covers the entire area of interest by doing a line-by-line scan. The pattern is rather time intensive but offers the advantage of complete coverage of the search region.

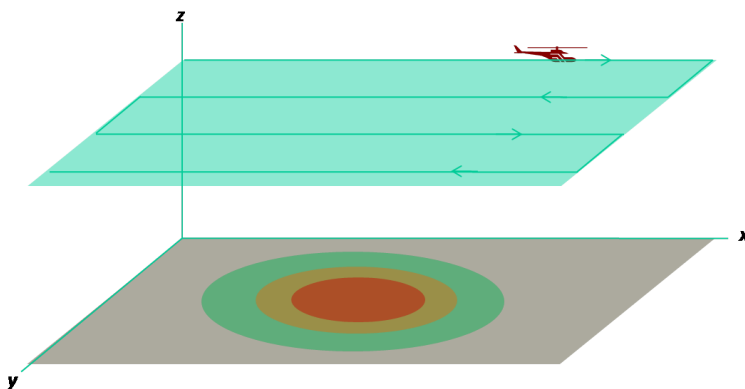


Figure 3.3: Raster Scan Pattern  
Figure Courtesy of Steven Martin, SNL, with Permission

One parameter to adjust for this type of scan is the width of the scan lines. The trade off is between time of the scan and resolution of the produced map. Increasing the distance between the scan lines will obviously reduce the scan time to cover the area, but the scan may not provide complete coverage of the area. The idea of “complete coverage” can be difficult to quantify. One possible model is to assume that the detector will receive a  $90^\circ$  cone of radiation. Figure 3.4 shows this concept. For this type of scan, the width of the scan lines is twice the scan altitude. As the figure indicates, increasing the altitude will also increase the blurring in the map.

The Unmanned Systems Lab performed testing of the raster-scan pattern in the WePi-*lot*’s simulation mode, and it was determined that stopping the helicopter at each waypoint would be very time consuming, as the helicopter acceleration is limited to  $0.5 \text{ m/s}^2$ . Instead, an oval-shaped “Zamboni” scan pattern (Figure 3.5) was adopted. This scan path avoids the starting and stopping, but also can potentially introduce a new problem. Figure 3.5



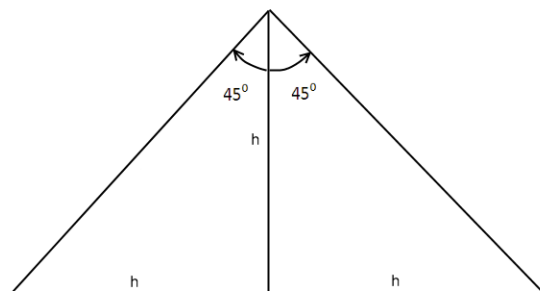


Figure 3.4: Scan Angle

shows that if the turning radius is not sufficient, some scan lines will not intersect the search area. The easiest way to avoid this is to decrease the scanning speed, and hence reduce the required turning radius.

Scanning speed affects both the resolution of the map and the total scan time. Given the time-criticality of the mission, the Zamboni scanning technique aims to reduce the total scan time. Table 3.1 shows the trade-off between scanning speed and total time of flight for an example scan area of 255 m x 130 m. The required turning radius is a function of velocity, and is a conservative estimate based upon WePilot simulations. The helicopter may be capable of a tighter turning radius; however, Virginia Tech does not have access to the performance parameters of the flight controller. As the table shows, a scanning velocity of 8 m/s provided the least distance traveled as well as the shortest flight time for this example. Future work includes generating an optimization routine to generate the optimal scan pattern for a given search area.

In addition to adjusting the scanning speed, map resolution can also be increased through the use of spatial deconvolution. Deconvolution is a signal processing technique which can be used to reduce “blurring” in the map. Convolution is defined as the integral

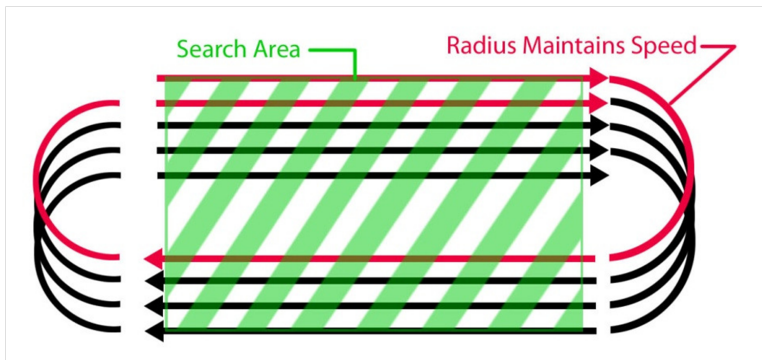


Figure 3.5: Adapted Raster Scan Pattern

Table 3.1: Scan Study

Velocity (m/s)	2	4	6	8	10
Required Turning Radius	10	20	30	40	50
Total Distance (m)	4030	2598	5181	2303	2879
Total Time (min.)	33.6	10.8	14.4	4.8	4.8

of the product of two functions after one is reversed and shifted [12]. Equation 3.2 shows the convolution equation for two functions  $f$  and  $g$ . The blurring that occurs in the map generation can be modeled as a convolution, therefore the goal of the deconvolution algorithm is to recreate the signal as it existed before the convolution took place [13]. Thus, the resolution of the scan can be increased without changing the flight pattern.

$$(f \star g)(t) = \int_{-\text{inf}}^{\text{inf}} f(\tau)g(t - \tau)d\tau \quad (3.2)$$

The scan altitude also has a distinct effect on the map generation. Figure 3.6 shows received signal strength for a simulated fly-over at various altitudes. As the figure shows, there is a pronounced peak at a relatively low altitude  $h$  (Figure 3.6(a)). As the altitude is increased, the peak becomes less and less clear. As Figure 3.6(d) shows, the source location is almost indistinguishable at an altitude of  $8h$ .

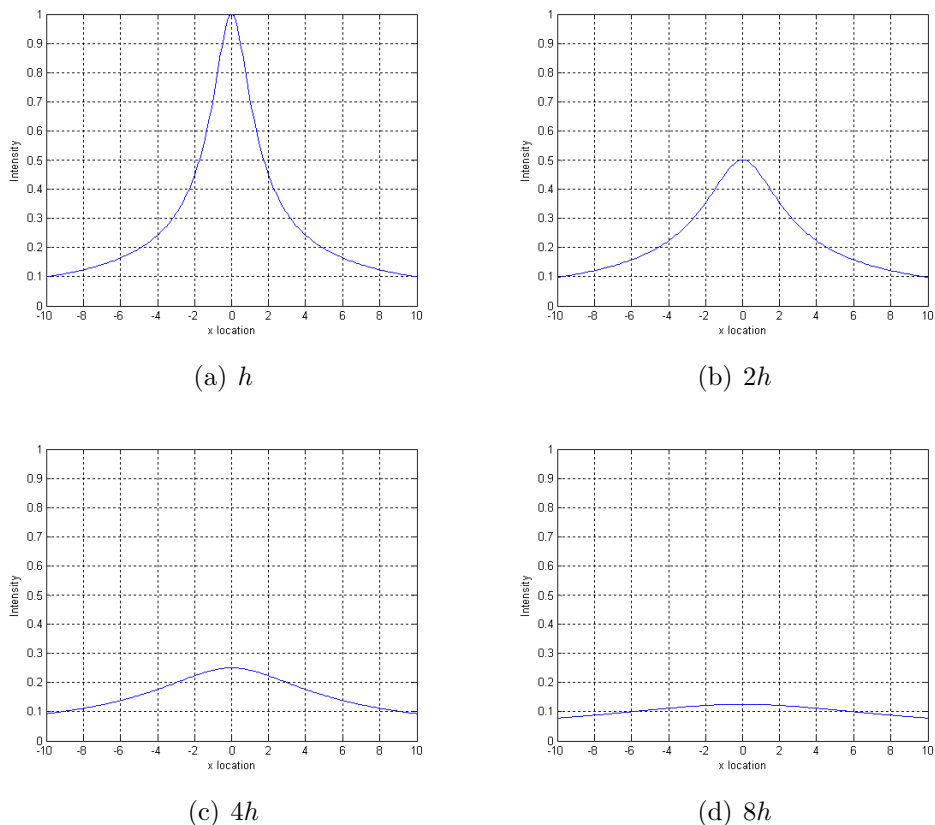


Figure 3.6: Intensity Readings at Various Altitudes

### 3.3 Aerial Platform

The Yamaha RMAX, shown in Figure 3.7, is the primary test platform for the Unmanned Systems Lab at Virginia Tech. Table 3.2 shows the specifications for the aircraft. The payload capacity and endurance are sufficient for carrying the radiation detector for 45 minutes. The payload capacity and flight time are limiting factors, making efficient source localization all the more important.

The radiation detector weighs 8.6 kg. (19 lbs.), and was designed to fit in the payload bay of the RMAX. To protect the detector, Virginia Tech housed the unit in a trapezoidal pod which is shock-mounted on the RMAX. Figure 3.8 shows the RMAX in flight with the rad pod in the payload bay.



Figure 3.7: Virginia Tech’s RMAX Helicopter

Table 3.2: RMAX Specifications [14]

<b>Max Payload</b>	28 kg.
<b>Gross Weight</b>	94 kg.
<b>Flight Time (with Payload)</b>	45 min.
<b>Main Rotor Diameter</b>	3.1 m.
<b>Overall Length</b>	3.6 m.
<b>Overall Height</b>	1.1 m.

To provide autonomy, Virginia Tech’s RMAX is equipped with the WePilot flight controller (Figure 3.9). The WePilot utilizes  $H_\infty$  control laws for attitude (inner-loop) stability, and PID control for guidance and navigation (outer loop). The maximum velocity for the helicopter is  $10 \text{ m/s}$  ( $22.4 \text{ mi/hr}$ ).

The WePilot is equipped with a Digi XTend 900 MHz wireless radio for communications with the ground station. A simple and intuitive user interface is provided for status moni-



Figure 3.8: Yamaha RMAX with Radiation Detector Pod

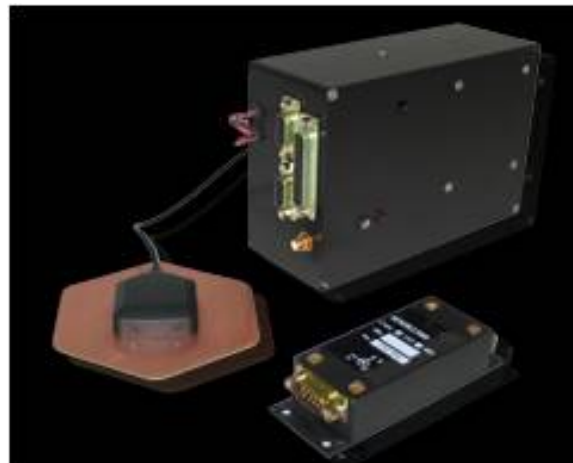


Figure 3.9: WePilot Flight Controller  
([http://www.vikingaero.com/uav\\_autopilot.html](http://www.vikingaero.com/uav_autopilot.html)  
With Permission )

toring and sending mission-level commands (such as waypoint navigation) to the helicopter.

Figure 3.10 shows the user interface.

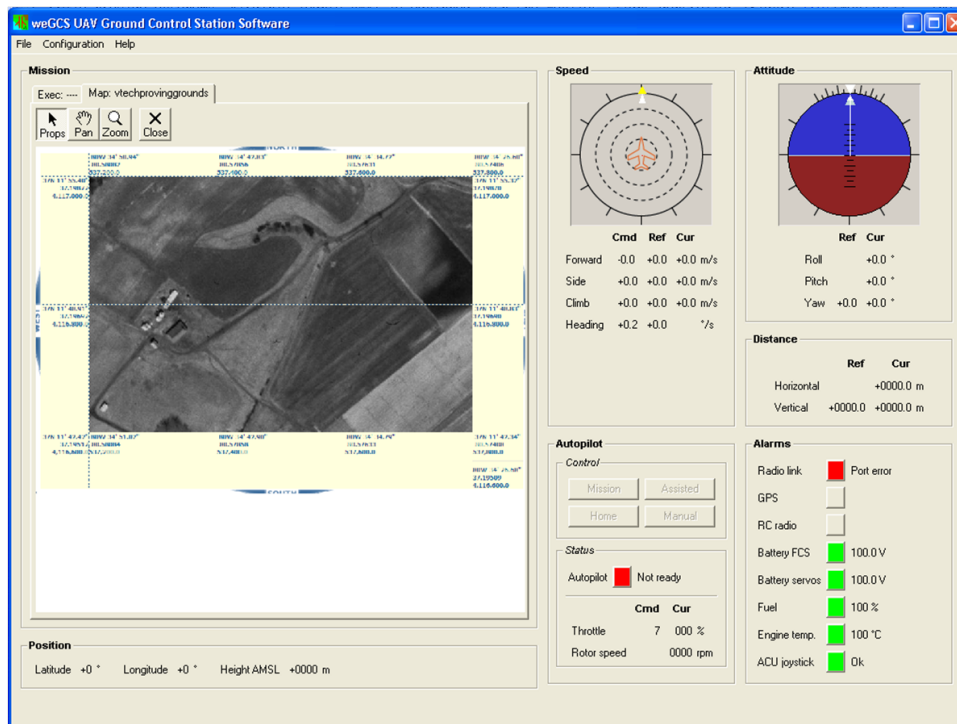


Figure 3.10: The WePilot Ground Station Software

### 3.4 Flight Operations at the Unmanned Systems Laboratory

The Unmanned Systems Lab (USL) performs all aerial experimentation at Kentland Farms, a Virginia Tech research facility located approximately ten miles from campus. The test site is a farming research facility which offers ample space for flight testing. Figure 3.11 shows the RMAX and the USL ground command vehicle at the Kentland Farms test site.

The ground command vehicle houses the ground station computers and serves as the primary command center for autonomous missions. Figure 3.12 shows the command center during a typical mission.

To ensure smooth flight operations, the Unmanned Systems Laboratory at Virginia Tech has developed a flight-testing procedure that is followed before each flight. The procedure consists of the following:





Figure 3.11: RMAX Helicopter Before Flight Test



Figure 3.12: Ground Command Station

1. Perform a Flight Readiness Review (24 hours before flight)
  - Mechanical Check
  - Systems Check
2. Perform Final Readiness Review (Directly before flight)

- Mechanical Preflight Checklist
- Systems Preflight Checklist

A flight readiness review is performed 24 hours prior to testing to ensure that the helicopter and each sub-system are ready for the flight. Preparing a day in advance allows extra debugging time for any problems that may arise. The final readiness review consists of reviewing the mechanical preflight checklist and the systems preflight checklist. Each subsystem has its own individual checklist.

Each flight at Virginia Tech requires at least three personnel: a safety pilot, a ground station operator, and an observer. The safety pilot is responsible for manual takeoff and landing of the helicopter, as well as crash avoidance in the case of an in-flight emergency. The ground station operator is responsible for monitoring the status of the helicopter, as well as executing mission-level commands (e.g. sending the helicopter to waypoints etc.). The observer is required to satisfy FAA flight regulations.

## 3.5 Experimental Setup

In order to evaluate radiation scanning concepts, a relatively weak collimated radiation source was provided by Virginia Tech. The collimated source emitted less than a 13° cone of radiation.

The hardware for the test consisted of the Virginia Tech's RMAX equipped with the WePilot flight control system, the radiation pod with the detector housed inside, and a two Dell Latitude laptop computers. Figure 3.13 shows the communications architecture between the helicopter and radiation display ground station.

The radiation detector calculates all of the spectral data and count rates on-board the helicopter. The detector also provides its location via an on-board GPS receiver. This information is then relayed to the ground station through a Microhard Systems wireless



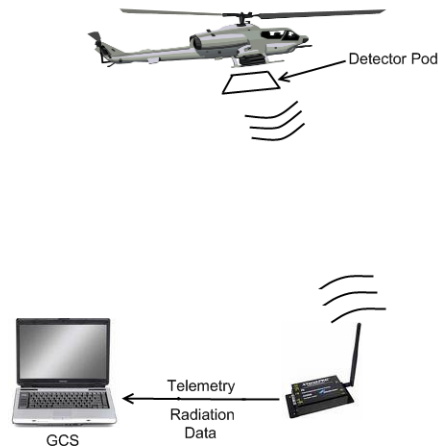


Figure 3.13: Radiation Mapping Communications Architecture

radio. The ground station uses the radiation and telemetry data to build a radiation map and display it in real-time. The experiment required three people: a safety pilot, a ground station operator, and a radiation computer operator.

### 3.6 Scan Results

Figure 3.14(a) shows the GPS coordinates of the scan pattern. Figure 3.14(b) shows the resulting counts received by the detector, which shows that the source is only “visible” when the helicopter is almost directly over it. In reality, the source will be unshielded. The scan pattern would be expected to return a bell-shaped graph of the  $1/R^2$  intensity falloff for the unshielded case.

Because there was not a reliable and accurate method to measure the distance from the source to the helicopter, the accuracy of the detector model could not be determined by this test. The scan does, however, verify the functionality of the detector.

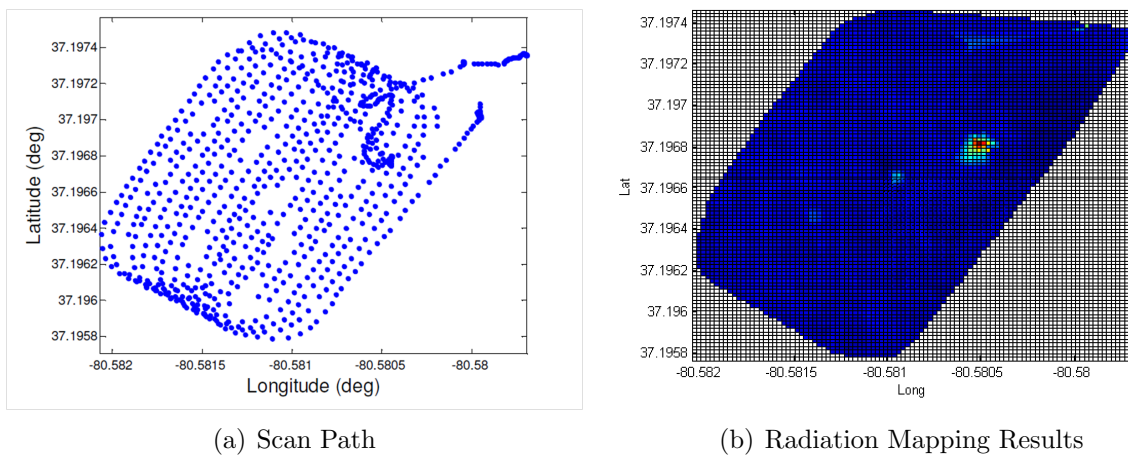


Figure 3.14: Scan Results

# Chapter 4

## Autonomous Localization Algorithm

This section describes the development of an autonomous localization algorithm to locate a single point source.

### 4.1 Scenario

The localization scenario examined by this thesis is the autonomous localization of a single un-collimated point source. The intensity of the point source is known to follow the  $1/R^2$  relationship where  $R$  is distance from the source. A non-directional sensor which measures intensity will be the only means of measuring source location. It is known a priori that the source is within a defined search region. Additionally, a model of expected intensity at a given distance is available.

The localization of a point source in 3-dimensional space is essentially a 4-dimensional problem, as the source location will provide 3 dimensions, and the source intensity the fourth. To simplify the problem, the algorithm presented considers a source of known intensity, which reduces the problem to three dimensions. The problem is further simplified by constraining the helicopter to fly at a fixed altitude relative to the source. Although the algorithm uses

these simplifications, it is designed such that extending its capabilities to more complex scenarios should be trivial. These extensions will be discussed further in Chapter 7.

## 4.2 Algorithm Selection

### 4.2.1 Recursive Bayesian Estimation

There are several localization methods listed in Chapter 2. Recursive Bayesian estimation algorithms offer an excellent combination of real-time computational feasibility and localization accuracy. They also offer the advantage of adaptability to virtually any source distribution. Additionally, Bayesian algorithms can be written to use information theory to calculate inputs that maximize information gain [15], which can be particularly useful given the unique mobility provided by a helicopter. With these advantages in mind, this thesis adopts a Bayesian approach to source localization. Several variations are available, and choosing the correct one for a given application requires an understanding of how each works.

Almost every recursive Bayesian estimation (RBE) algorithm operates on the same basic principle. The idea is to use a process model to predict the behavior of a system, and then use measurements to correct the prediction. If the prediction matches the measurement closely, then the filter will converge. If the measurements are too noisy, or the process is not modeled correctly, then the filter will diverge from the true state. The recursive algorithm requires a motion model and a sensor model to provide the predictions and corrections. Figure 4.1 shows the basic filter operation. The filter must be initialized to some initial state  $x_0$ . The prediction step updates the belief with the process model, and the correction step uses measurements to correct the prediction, and the process repeats. One of the main advantages of RBE is that unlike algorithms such as the Maximum Likelihood Estimator (MLE), Bayesian algorithms represent previous measurements by a posterior belief distribu-

tion rather than storing all of the previous measurements. This is particularly useful as the number of measurements becomes large.

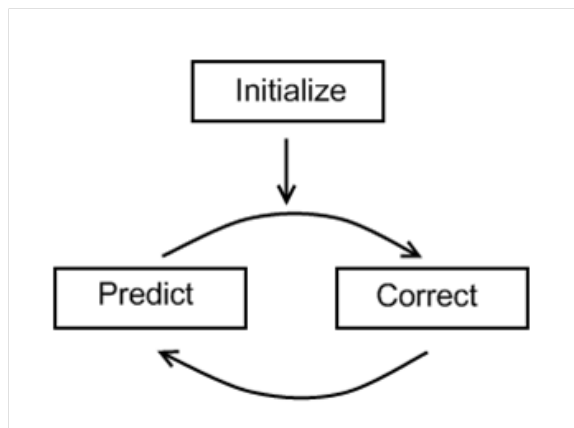


Figure 4.1: Recursive Bayesian Estimation

## 4.2.2 Representing the Belief Space

The main difference between the Bayesian algorithms is how each represents the belief space. The Kalman filter is able to represent the belief space with only two parameters, the mean and covariance. As a result, the filter is very computationally simple, although the shape of the belief distribution is constrained to be Gaussian (see Figure 4.2).

The belief space of a system often cannot be captured by a simple Gaussian distribution. For example, consider measurements from a static sensor measuring signal intensity from a point source. In a two-dimensional space, these measurements could not be used to calculate a source location directly. Rather, the belief space would provide a “ring” in which the source may lie. Figure 4.3 illustrates this situation. As the figure shows, this type of distribution may be difficult to model using a Gaussian approximation. The particle filter, grid-based method, and element-based method all provide more accurate representations of this distribution (see Figure 4.4). A surface is fitted over the particles in Figure 4.4(a) for ease of visualization. Note that the element-based method is an extension of the grid-based method, and hence its distribution is omitted.

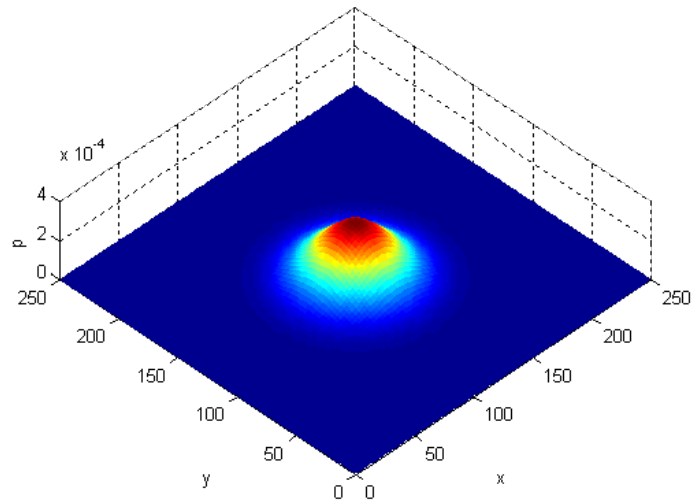


Figure 4.2: Gaussian Belief Distribution

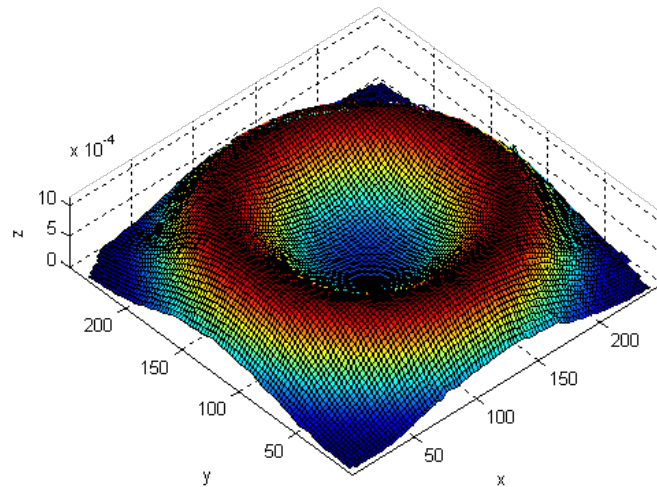


Figure 4.3: Non-Gaussian Belief Distribution

### 4.2.3 Computational Considerations

Figure 4.5 shows several recursive Bayesian algorithms plotted on axes of efficiency, ability to handle non-Gaussian distributions, and ability to handle nonlinear systems. The Kalman Filter (KF) is the simplest and hence the most efficient of the algorithms, but as mentioned

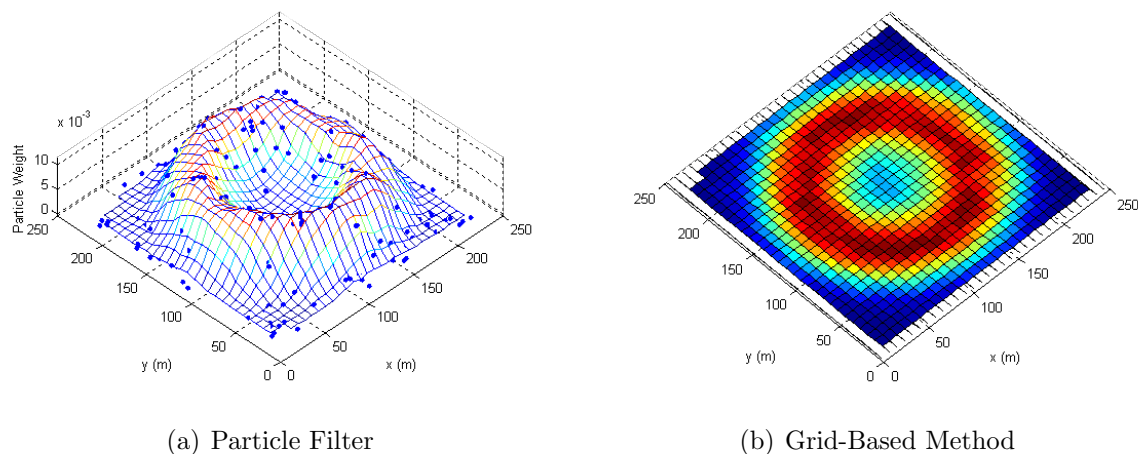


Figure 4.4: Belief Representations

above it is very poor in representing non-Gaussian systems. The process model used for the KF is also constrained to be linear. The Extended Kalman Filter (EKF) relaxes the linearity constraint by linearizing the process model about the operating point via the Jacobian matrix. The Unscented Kalman Filter (UKF) further extends the functionality of the basic Kalman Filter to allow some non-Gaussian distributions, but still often struggles with highly non-Gaussian and nonlinear systems ([4],[7]). The accuracy and computational efficiency of the Grid-Based method (GM) and Element-Based method (EM) are highly dependent on the number of elements or how fine of a grid mesh is selected. When a fine-grained grid is selected, it may be difficult to execute the algorithms in real-time. The main reason for this is the convolution that is used for the motion and measurement update. For example, the motion update for a 3-D grid would require a 6-D operation [16]. As the Figure 4.5 shows, the particle filter (PF) provides the best compromise between efficiency and ability to handle non-linear and non-Gaussian systems, and hence it was selected for the search algorithms.

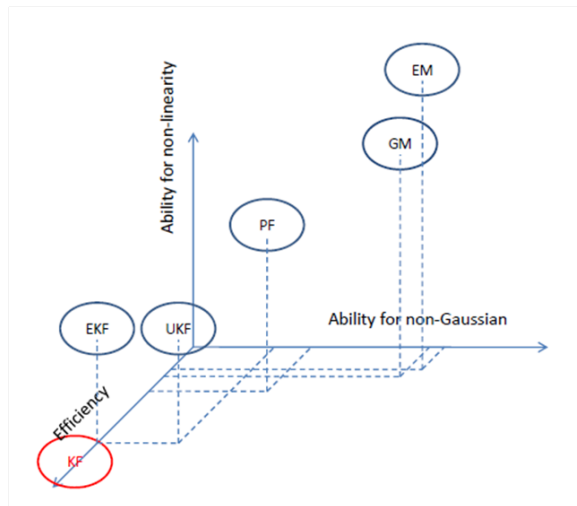


Figure 4.5: Recursive Bayesian Estimation Algorithms

Figure courtesy of Tomo Furukawa, Virginia Tech, with Permission

## 4.3 The Particle Filter

### 4.3.1 Background

The particle filter is a Monte-Carlo simulation method that uses a set of random state samples, or particles, to represent the posterior belief. In other words, a particle is a hypothesis as to what the true world state may be at time  $t$  [16]. Each particle then has an associated weight which represents how accurate the hypothesis is based upon state measurements.

### 4.3.2 1-D Example

To explain the basic operation of the particle filter, a 1-D localization example is given. The source is located at  $x = -3$  and a helicopter with a sensor moves along the  $x$ -axis. As a starting point, an algorithm similar to the one given in [8] is used.

Algorithm 1 describes the filter operation. Lines 1 and 2 initialize the particles and weights, where the particles are drawn from a random distribution, and the weights initialized to  $1/N$ , where  $N$  is the number of particles. This step puts particles at random locations



throughout the space, each with an equal weight. This is because initially we have no idea where the source is, so there is an equal chance that it could be at any given location. Line 4 is the belief update step, where  $p(x_k|x_{k-1})$  gives  $x$  at the  $k^{th}$  time step given  $x$  at the previous time step. Step 4 will be discussed in detail later in this example.

Lines 5 and 6 calculate normalized errors based upon the sensor measurement. In this example, the sensor measures distance from the source directly, thus the  $i^{th}$  predicted distance is simply the distance from particle  $i$  to the helicopter. The errors are then normalized in line 6 to lie between -1 and 1.

Lines 7-9 are the weight update equations. The weights are updated by multiplying the weight at the previous time step by a number drawn at the point  $error^{(i)}$  from the normal distribution  $\mathcal{N}$  with mean  $\mu$  and variance  $\sigma$ . The weights are then normalized so that they sum to one in line 10. The source estimate is calculated in line 11, where  $x^{(i)}$  is the  $i^{th}$  particle. Lines 12 and 13 update the weights and particles for use at the next time step.

---

**Algorithm 1** Particle Filter
 

---

```

1: Initialize Particles:  $x_0 = rand(N, 1)$ 
2: Initialize Weights:  $w_0 = 1/N$ 
3: while 1 do
4:   sample  $x_k \sim p(x_k|x_{k-1})$ 
5:    $error^{(i)} = prediction^{(i)} - measured^{(i)}$  Errors Between Prediction and Measurement
6:    $error = \frac{error}{max(|error|)}$  Normalize the Errors
7:   for  $i = 1$  to  $N$  do
8:     Update Weights  $w_k^{(i)} = w_{k-1}^{(i)} \mathcal{N}(error^{(i)}, \mu, \sigma)$ 
9:   end for
10:  Normalize Weights  $w = \frac{w}{\sum_{i=1}^N w^{(i)}}$ 
11:   $\hat{x} = \sum_{i=0}^N w^{(i)} x^{(i)}$  Calculate State Estimate
12:   $x_{k+1} = x_k$ 
13:   $w_{k+1} = w_k$ 
14: end while

```

---

Figure 4.6 shows a time line of the algorithm where the helicopter moves from left to right. The source is the red dot, and the helicopter represents the sensor location. The blue dots are the particles. The first time step (Figure (a)) shows the initial distribution of

particles. Notice that after two time steps there are two possible locations for the source. This is because the sensor measures distance only, so the source could be on either side of the helicopter. The inaccurate source location is sometimes called a “ghost” measurement[10]. As the helicopter moves to the right, the distribution shifts toward the true source location (Figure (c)). By the eighth time step, the source location is quite clear from the distribution. One problem, however, is that there are only a few particles near the source location, and many of the (lower-weight) particles are taking up computational time while adding little value to the distribution.

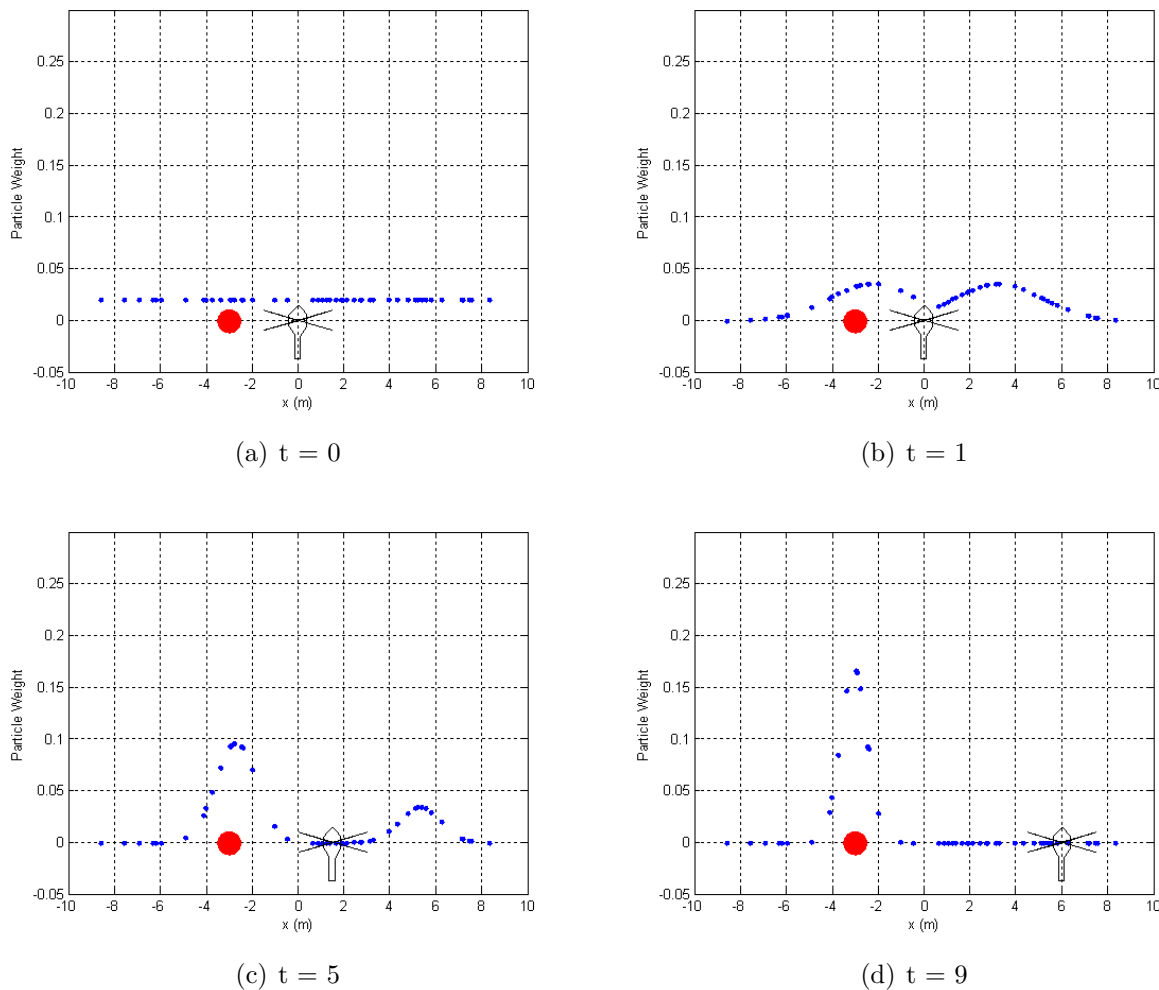


Figure 4.6: 1-D Particle Filter Example

The reason for the inefficient use of particles is the way that the particles were drawn from the posterior distribution (Algorithm 1 line 4). For the example given in Figure 4.6, the particles were not updated in this step. The specific implementation of step 4 depends upon the type of particle filter used. In the case of an sequential importance sampling (SIS) particle filter, this step would draw a new random set of particles from the posterior distribution at each time step[17]. This requires that it is possible to draw from the posterior distribution, which is not always easily represented.

Another method that can be implemented for this step is a particle swarm optimization technique. Particle swarm optimization (PSO) is a computational technique based upon the social behavior metaphor [18]. The foundation is similar to the particle filter in that the algorithm is initialized with random particles, each representing a possible solution. Each particle is then assigned a velocity and moved toward the particle of “best fitness”. The best fitness is the best value achieved by any particle thus far. Combining PSO with the particle filter allows for a more efficient use of particles without the requirement of drawing from the posterior distribution.

Implementing this technique directly did not yield acceptable results, as the “best fitness” value was not necessarily the source location. This is evident in Figure 4.6(b). Accordingly, a new algorithm was implemented which uses a weighted measure of distance and particle weight to adjust the particle location. Pseudo code is provided in Algorithm 2.

---

#### Algorithm 2 Particle Movement Algorithm

---

```

1: for  $i = 1$  to  $N$  do
2:   for  $j = 1$  to  $N$  do
3:      $distance(i, j) = \|part^{(i)} - part^{(j)}\|$  calculate distance between  $i^{th}$  and  $j^{th}$  particle
4:      $fitness(i, j) = \frac{w(i)^n}{distance(i, j)}$  calculate weighted measure of distance and particle weight
5:   end for
6: end for
7: for  $i = 1$  to  $N$  do
8:    $p_{max}^{(i)} = find(max(fitness(:, i)))$  Find the maximum fitness for each particle
9:    $part(i) = part(i) + k * direction$  Move Particle  $i$  toward  $p_{max}$ 
10: end for

```

---

Lines 1-6 calculate the “fitness” of each particle with respect to all of the other particles. This fitness is a measure of how close particle  $i$  is to particle  $j$  as well as how high the weight of particle  $j$  is. In other words, the fitness is used to select the best particle  $j$  to move particle  $i$  towards given the metric listed on line 4. The parameter  $n$  in line four controls how much the particles will “clump”. The weights will always be less than one, so smaller values of  $n$  will result in more emphasis placed on the weight of the particles as opposed to the distance. Lines 7-10 move each particle toward the particle of highest fitness. The parameter  $k$  in line 9 is used to control how rapidly the particles converge, and is expressed as a percentage of distance. For example, a  $k$  value of 0.2 will send particle  $i$  20% closer to  $p_{max}^{(i)}$ .

The updated particle filter algorithm is shown in Figure 4.7. This version of the algorithm is successful in concentrating the particles in areas of interest; however, as Figure (d) shows, the problem of “degeneracy” is accentuated in this case. Degeneracy is a common problem with particle filter algorithms in which most particles will have a weight close to zero, while a few highly-weighted particles will dominate the distribution. Resampling is the common solution to this problem, and a resampling algorithm is presented in the next section.

## 4.4 Adapting the Particle Filter For this Application

### 4.4.1 Defining the Weight Update Equation

One key component of the particle filter algorithm is updating the particle weights. One possible update equation (shown in Equation 4.1) was to use the error in the measured counts from the predicted counts.

$$error^{(i)} = counts - counts_{pred}^{(i)} \quad (4.1)$$

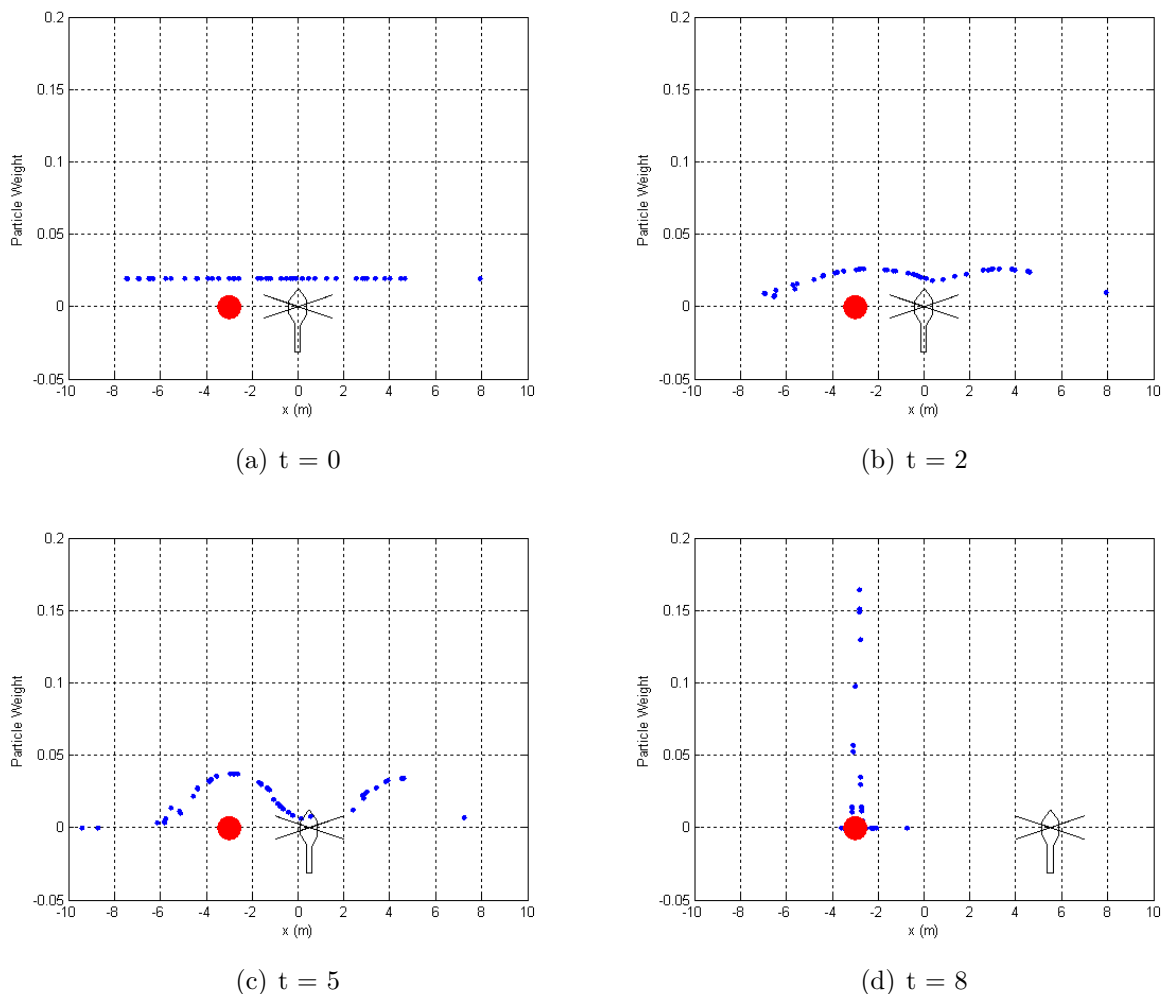


Figure 4.7: 1-D Particle Filter Example

where  $counts_{pred}^{(i)}$  is the predicted counts at particle  $i$  obtained from a detector model provided by Sandia National Labs (see Section 3.1). The errors were then normalized to lie between -1 and 1, and the weights were updated using the weight update equation from Algorithm 1. The problem with using this error calculation is that the error of the  $i^{th}$  particle is inversely proportional to the square of the distance from the source. This means that particles very close to the helicopter would predict a very large number of counts. As such, these particles had errors that were orders of magnitude greater than the other particles. Figure 4.8(a) shows that the large errors dominated the distribution.

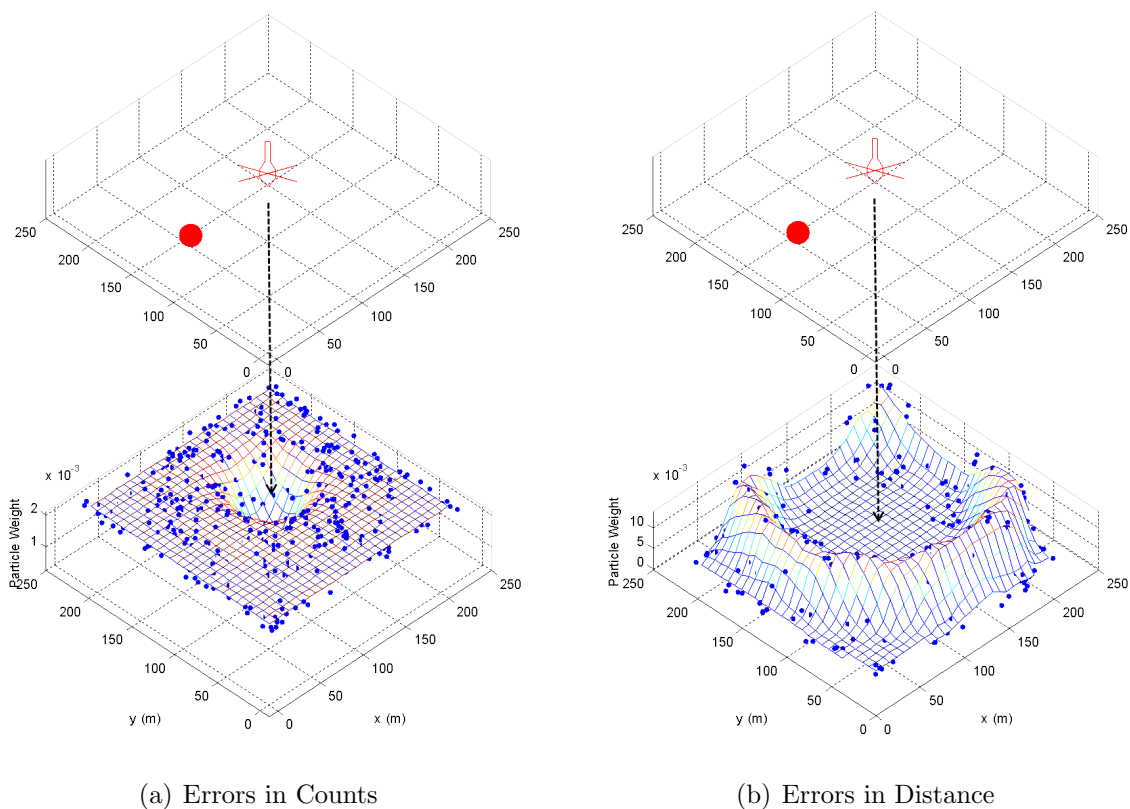


Figure 4.8: Particle Belief Distributions

With this in mind, an error in distance (which would be linear) instead of counts was used. Equation 4.2 was used to calculate the errors.

$$error^{(i)} = d - d_{pred}^{(i)} \quad (4.2)$$

where  $d$  is the measured distance and  $d_{pred}^{(i)}$  is the predicted distance for particle  $i$ . This provided the desired linear error distribution. One problem with this solution was that although the characteristics of the count distribution are dominated by the inverse square law, there are other characteristics present such as air attenuation and fall-off due to solid angle of detector with respect to the source. For this reason, an analytical expression for distance from measured counts would have been very difficult to develop. Instead, MATLAB's optimization

toolbox was used to back-solve the count prediction equation iteratively. The result, shown in Figure 4.8(b), was a much more representative distribution.

#### 4.4.2 Resampling

As mentioned in Section 4.3.2, one problem with particle filtering is the problem of degeneracy. To combat this problem, a resampling algorithm was developed. Resampling reconcentrates particles to areas of interest, as well as adds “randomness” to the particles in order to prevent a degenerate distribution. Equations (4.3 - 4.4) were used as a starting point for resampling. Equation 4.3 replaces the weights with a uniform distribution.

$$w_k^{(i)} = \mathcal{U}(1/N) \quad (4.3)$$

where  $\mathcal{U}$  is the uniform distribution and  $N$  is the number of particles. Equation 4.4 redistributes the particles around the particle of highest weight.

$$x_k = x_{k-1}(\max_w) + mag * rand \quad (4.4)$$

where  $\max_w$  is the index of the particle with the highest weight,  $mag$  is the magnitude of noise, and  $rand$  is random noise drawn from a uniform distribution on the interval  $[-1, 1]$ .

This method is inadequate, as sometimes the resampling causes the loss of valuable data. Resampling with this algorithm results in only a small block of particles near the highest weighted particle. Sometimes a noisy measurement coincides with an incorrect particle, and hence that (incorrect) particle will have the highest weight. This means that if the resampling occurred at this instant, the resampled particles would be at an incorrect location. The sequence shown in Figure 4.9 illustrates this concept.

Instead, a new resampling algorithm was developed that only replaced particles of

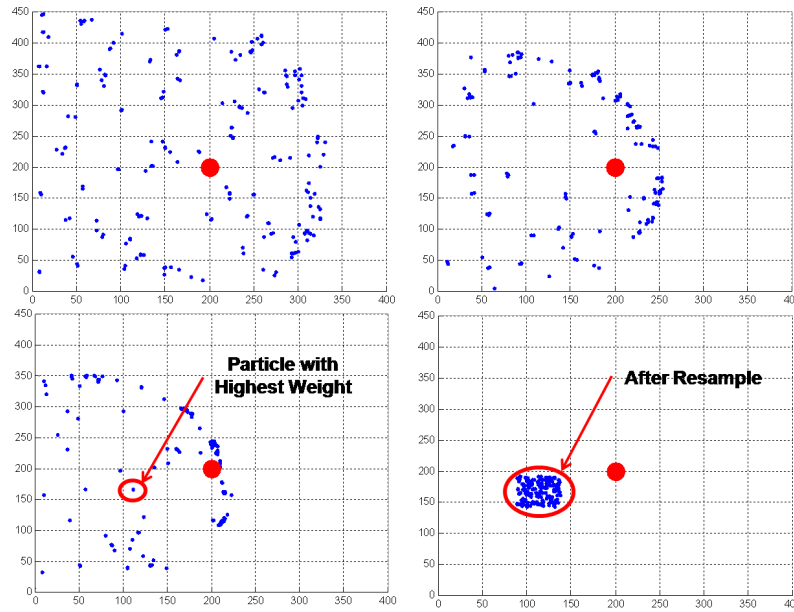


Figure 4.9: Unsuccessful Resampling Algorithm

relatively low weight (see Algorithm 3). Note that the weights greater than the threshold ( $w_{thresh}$ ) and their corresponding particles remain unchanged, while the weights less than the threshold are replaced from the uniform distribution (lines 2-7). The particles corresponding to the weights less than the threshold are replaced with remaining particles plus random noise (lines 9-11). As a result, much more useful information from the distribution is retained. Figure 4.10 shows a time line of the updated resampling algorithm.

---

**Algorithm 3** Resampling Algorithm
 

---

- 1: **for**  $i = 1$  to  $N$  **do**
  - 2:   **if**  $w^{(i)} > w_{thresh}$  **then**
  - 3:      $index_L = [index_L \ i]$  Store Index
  - 4:   **else**
  - 5:      $w^{(i)} = \mathcal{U}(1/N)$  Replace Weights from Uniform Distribution
  - 6:      $index_M = [index_M \ i]$  Store Index
  - 7:   **end if**
  - 8: **end for**
  - 9: **for**  $m = 1$  to  $length(index_L)$  **do**
  - 10:    $x_k^{(i)} = x_{k-1}(index_M(m)) + mag * rand$  Replace Particles With Small Weights
  - 11: **end for**
-



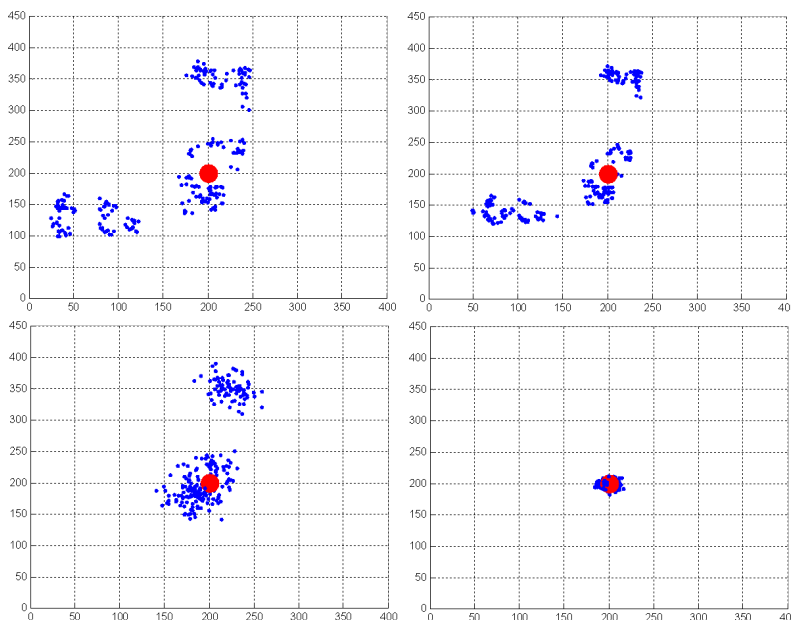


Figure 4.10: Successful Resampling Algorithm

One practical consideration is how often to resample. Resampling too often can result in loss of diversity, while not resampling enough causes many particles to be used in areas of low interest. The resampling logic for this application was chosen from [17], where the resampling is only performed if the number of effective particles ( $N_{eff}$ ) drops below some threshold ( $N_{th}$ ). The effective particles are defined in Equation 4.5. Note that when the weights are uniformly distributed,  $N_{eff}$  will be equal to  $N$ , while if only one particle has a non-zero weight, then  $N_{eff}$  will be one.

$$N_{eff} = \frac{1}{\sum (w_t^i)^2} < N_{th} \quad (4.5)$$

Figure 4.11 shows a flow chart of the particle filtering algorithm that was implemented.

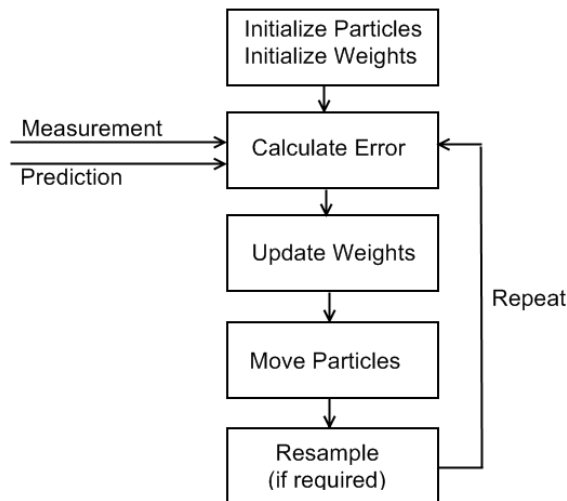


Figure 4.11: Particle Filtering Algorithm

## 4.5 Control Strategy

Because the localization algorithm presented utilizes a non-directional detector, the flight path can have a significant effect on algorithm convergence. If the helicopter is stationary, the sensor measurements will only describe a ring in which the source will lie. To localize the source, two control strategies were considered: one which directs the helicopter toward the source estimate, and one which flies “tangent” to the source estimate. Because the helicopter is constrained to fly in a plane of fixed altitude, tangent in this case refers to a tangent path relative to the projection of the source location on to the plane of the helicopter (see Figure 4.12).

The former was chosen as a first attempt. Intuitively, it seemed that traveling directly toward the source would minimize the flight time required to get to the source. In simulation, however, this did not hold true. As Figure 4.13 shows, the algorithm often struggled to find the source with this control strategy. The reason is not known for certain, although the author suspects the problem was the “ghost” source that can arise due to the non-directional

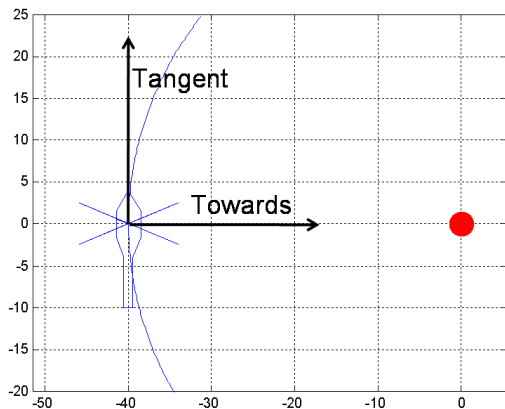


Figure 4.12: Control Strategies

measure of the detector. Figure 4.13(c) shows that the source estimate (the black dot) and actual source (the red dot) are symmetric about a line connecting the helicopter to the midpoint of the two dots. If the helicopter were to continue along that line, the measurements would be identical for the real source and the estimate. Once the algorithm has converged on the ghost point, it is difficult for the algorithm to find the source.

A path tangent to the estimated source was used to combat this problem. Moving along a tangent path would maximize the difference between the ghost source measurements and the actual measurements. To implement the tangent flight path, a vector from the helicopter to the source estimate ( $\vec{p}_{hs}$ ) was calculated using Equation 4.6.

$$\vec{p}_{hs} = \vec{p}_s - \vec{p}_h \quad (4.6)$$

The control input was then a vector tangent to  $\vec{p}_{hs}$ . To ensure that the inputs were the correct direction, the input was selected such that the cross product from the input vector to  $\vec{p}_{hs}$  was always positive, which meant that the helicopter would circle clockwise. Figure 4.14 shows the updated control strategy. After the variance of the past 15 source measurements is less than a specified threshold, the helicopter moves toward the source (Figure 4.14(d)). Future control developments are discussed in Chapter 7. A summary of

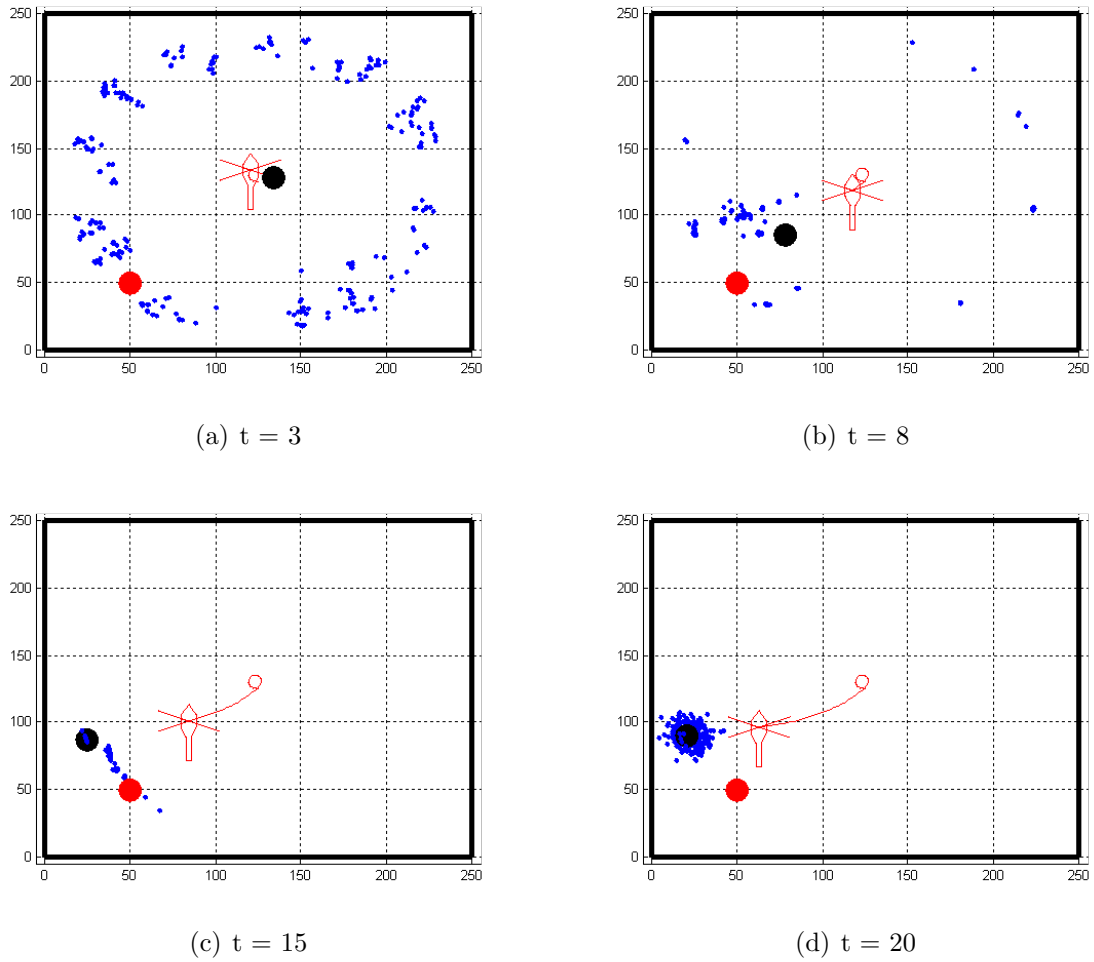
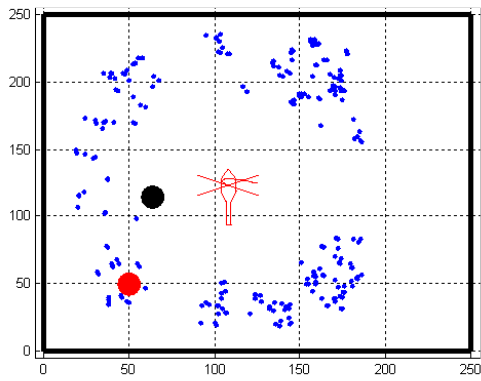
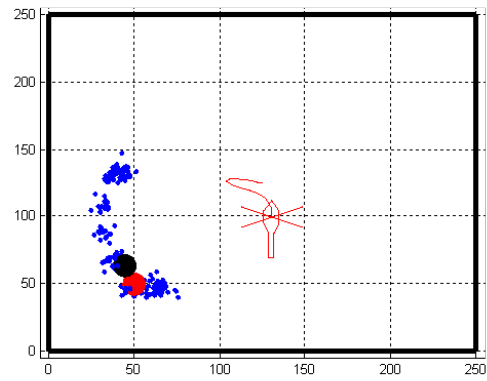


Figure 4.13: Control Toward Estimate

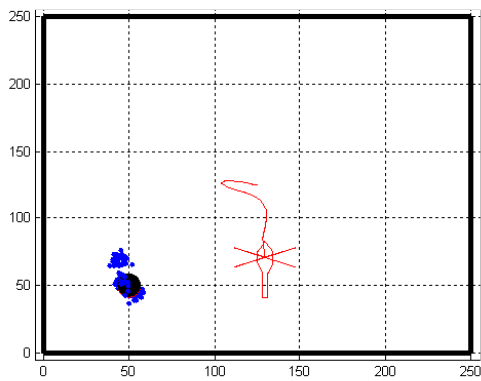
the simulation results are presented in Chapter 6.



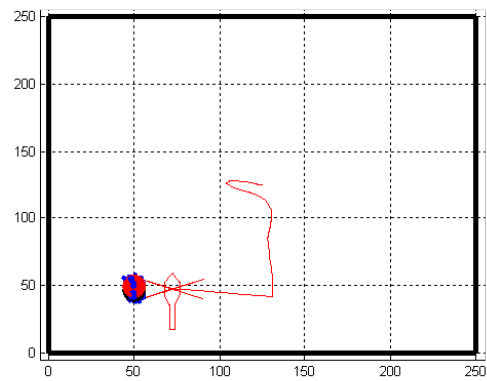
(a)  $t = 5$



(b)  $t = 15$



(c)  $t = 25$



(d)  $t = 40$

Figure 4.14: Control Tangent To Estimate

# Chapter 5

## Helicopter Software Interface Development

In order to implement the search algorithm on an autonomous helicopter, a software interface was developed. The basic architecture is shown in Figure 5.1. The interface accepts telemetry and radiation data from the helicopter, and sends control inputs back to the helicopter to drive the search algorithm. This chapter outlines the development of the software interface.

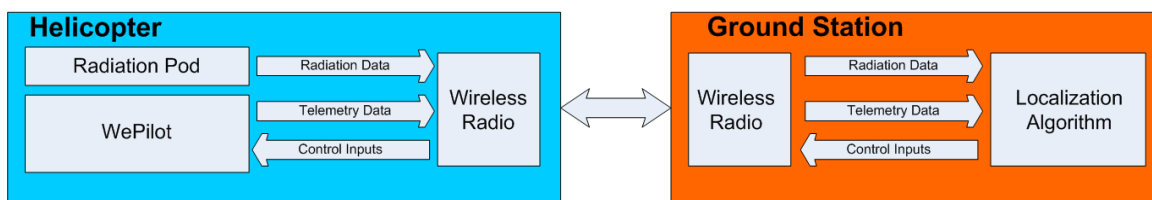


Figure 5.1: Communications Architecture Flow Chart

### 5.1 Hardware Setup

There are two primary ways to interface with the WePilot flight control system. One involves an on-board computer that sends commands to the WePilot via RS-232. The second is an

interface with the ground station software which sends velocity commands over a wireless link. The former would seem like the appropriate choice, as it would eliminate the delays and potential reliability issues presented by the wireless link. Unfortunately, at the time of this work, the on-board interface was not available. Accordingly, an interface with the ground station software was developed for short-term use.

Figure 5.1 shows the command architecture. Telemetry is sent from the helicopter to the ground station via a wireless link, and then a hardware serial splitter is used to relay the telemetry information to the localization ground computer. The localization computer sends control inputs to the WePilot ground station which then relays them to the helicopter. While this interface is not ideal, it has proved suitable for the experimentation performed in this thesis.

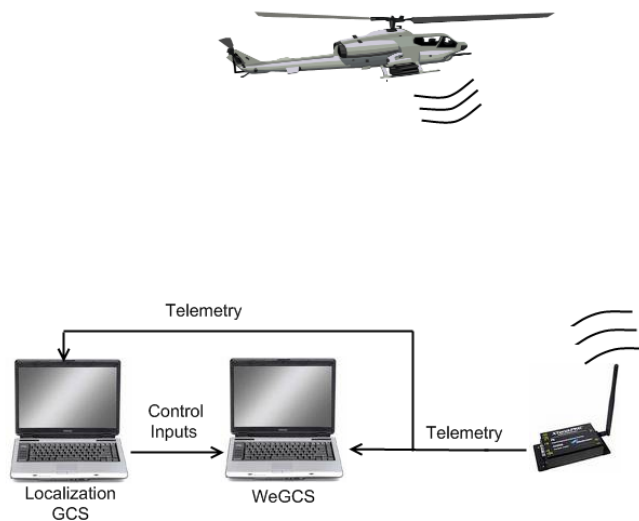


Figure 5.2: Communications Architecture

## 5.2 Navigation Solution

All telemetry for the experiment was provided by the WePilot flight controller. Its Extended Kalman filter (EKF) provides a full navigation solution including filtered GPS, INS, and North/East/Down (NED) position defined in an inertial coordinate frame from the takeoff point. The NED position proved to be the most useful for the algorithm testing, as all distances from the source were given from the inertial reference frame.

To test the accuracy of the position estimate, a simple ground-based experiment was performed. The experiment consisted of measuring a 61 m (200 ft.) x 30.5 m (100 ft.) rectangle, then manually wheeling the helicopter around to each of the four corners. Figure 5.3 shows the results of this experiment. Note that there is roughly a two meter error between the starting point of the rectangle and the ending point (near [0,0] in the figure). This can be attributed to GPS drift, as the start and end points were identical in reality. Two meters is a reasonable amount of error for the algorithm to operate with. In fact, the algorithm was tested with simulated noise with a variance of five meters, and the localization algorithm was still successful in localizing the source within 10 meters. The collected data is also considered a worst-case scenario, as between six and seven GPS satellites were available for this testing. Six is considered the minimum that is acceptable to fly with.

The distances provided by the WePilot were also reasonably accurate. The error was 8 ft. for a distance of 200 ft., which presents an error of 4%. This should be more than suitable for the experimentation. In addition, the estimate was compared with a Google Maps picture of the test site to ensure that the orientation was correct in the global frame. As Figure 5.4 shows, the results shows an overall agreement with the fly field.



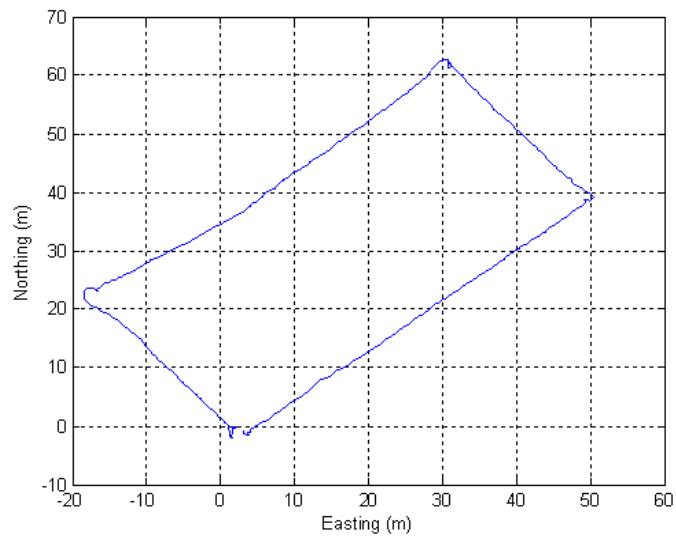


Figure 5.3: Ground-based Error Testing



Figure 5.4: Google Maps Image

### 5.3 Control Algorithms

Because the WePilot flight control system provides the inner-loop stability of the helicopter, the control algorithms presented are outer-loop guidance algorithms. The WePilot provides a

relative velocity input command, which allows the ground command station to send velocity commands in the inertial frame, rotated by the helicopter’s heading. In other words, if the helicopter is pointing North, a positive velocity command  $u$  would move the helicopter due North, while a positive velocity command  $v$  would send the helicopter due East, regardless of the attitude of the helicopter. Waypoint commands, however, are usually given either in GPS coordinates or in the NED frame. To convert the velocities from NED to the helicopter frame, a rotation matrix by angle  $\psi$  is used. This conversion is given in Equation 5.1.

$$\begin{bmatrix} u \\ v \end{bmatrix} = R_{N2B}(\psi) \begin{bmatrix} vel_N \\ vel_E \end{bmatrix} \quad (5.1)$$

where  $\psi$  is the heading angle,  $u$  and  $v$  are body-fixed forward and side velocities, respectively, and  $R_{N2B}(\psi)$  is the rotation matrix from the NED frame to the body-fixed frame.

The command interface offers the ability to fly to waypoints. This can be particularly useful in placement of the helicopter before a localization experiment, or after the experiment has ended and the helicopter needs to return home. Rather than relying on the pilot, the “go home” command sends the helicopter to the home waypoint autonomously. The velocity direction is calculated using Equation 5.2.

$$\vec{V}_{cmd} = \vec{P}_{des} - \vec{P}_{current} \quad (5.2)$$

where  $P_{des}$  is the desired position vector, and  $P_{current}$  is the current helicopter position vector. The magnitude is determined by a proportional control, where the final velocity output is clipped to be no more than  $V_{max}$  (see Equation 5.3). As mentioned previously, the detector integrates the number of counts over the sampling interval. As a result, traveling at slower velocities allows for a more precise measure of received counts. For this reason,  $V_{max}$  was limited to 2 m/s. Note that the helicopter was in “heading-hold” mode for these experiments. Normally, the heading would align with the velocity vector, but the heading-hold mode was

utilized for simplicity.

$$V = \min(K_p \|V_{cmd}\|_2, V_{max}) \frac{V_{cmd}}{\|V_{cmd}\|_2} \quad (5.3)$$

## 5.4 Proposed Scan Techniques

The localization algorithm presented in Chapter 4 can only be implemented when a source is known to be within the field-of-view of the detector. The proposed scan area for the mission may be much larger than the detectable region of the source. As such, the high-altitude radiation scan should be performed before the localization algorithm is implemented. If time permits the full raster scan pattern described in Chapter 3, the data provided by this scan should be sufficient for identifying areas of interest to run the localization algorithm. If there is not sufficient time for this scan, as may be the case, some different approaches could be implemented.

One such approach may be to start the helicopter in the middle of the search area, and perform a spiral outward. Once an increased number of counts is observed, the helicopter can abandon the spiral scan and begin the localization algorithm.

Another approach may be to use visual cues from the terrain and allow a human operator to make an educated guess about where the source may be. These techniques will be left for development in future work.

## 5.5 Graphical User Interface

In order to make the system more intuitive, a graphical user interface was developed using MATLAB's Guide graphical user interface development environment. This section outlines the functionality of the user interface shown in Figure 5.5.

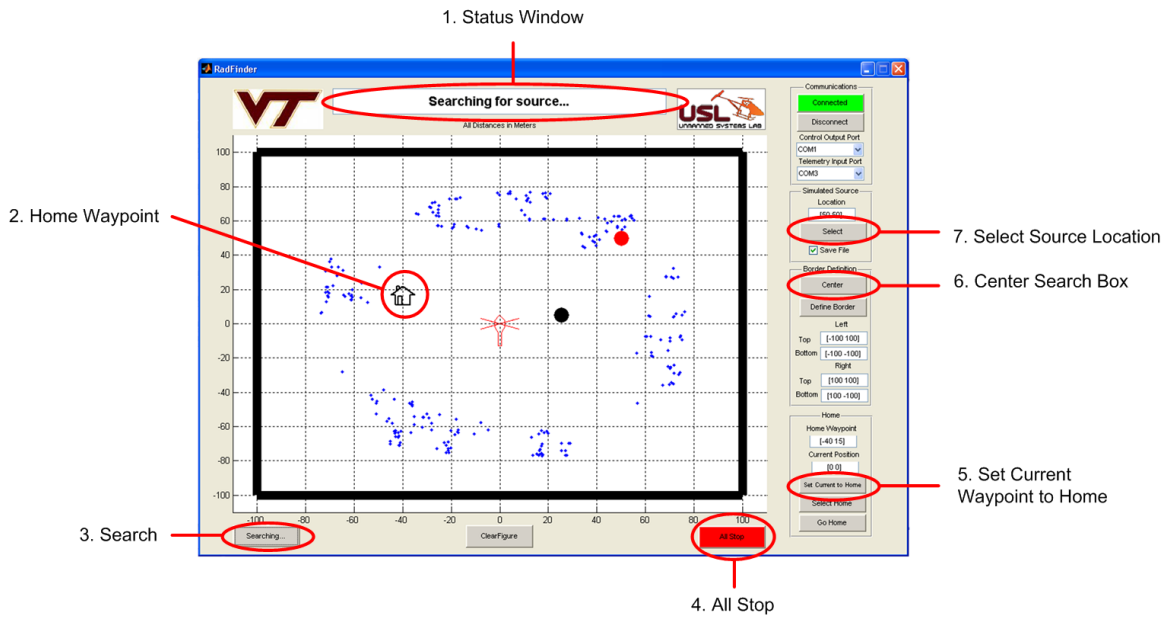


Figure 5.5: Ground Control Software

The status window (1 in the figure) displays the current state of the search algorithm, and also alerts the user if any problems arise during the search. The black border in the interface is a user-definable search box to be used for the localization algorithm. The helicopter will not leave this boundary during its search. The “Center Search Box” button centers the border around the helicopter’s current location. This feature is particularly convenient, as the inertial coordinates of the helicopter are always defined from the takeoff point, and hence change with each flight.

The user can also select the home waypoint and simulated source location by clicking points on the screen. The “Set Current to Home” button sets the helicopter’s current position as the home waypoint. The “Search” button begins the localization algorithm. In the event of an in flight emergency, the “ALL STOP” button sends a zero-velocity command to the helicopter.

# Chapter 6

## Experiment with Simulated Sensor Data

To test the tools developed in Chapters 4 and 5, a flight experiment was performed at Kentland Farms. Unfortunately, a suitable uncollimated radioactive source was not available at the time of the test. Instead, the detector counts were simulated using the analytical detector model. MATLAB’s “Poisrnd” command was then used to simulate the Poisson-distributed detector noise that could be expected in a real-world experiment. The following sections outline the simulation and experimental results.

### 6.1 Simulation Results

Before implementing the algorithm experimentally, several Hardware-in-the-Loop (HIL) simulations were performed to verify the robustness of the search algorithm. The WePilot is equipped with a HIL simulator which allows for mission execution with simulated GPS and telemetry data. The command architecture for the setup is identical to the system that was shown in the previous chapter (Figure 5.2).

Table 6.1 shows the results of 30 test simulations performed in a 250m x 250m search box. The helicopter flew at an altitude of 60 meters relative to the source, and the source was located 60 meters horizontally from the helicopter at the start of the simulation. An altitude of 60 meters was selected to decrease the effects of detector directionality. This concept is described further in Section 7.4. The average error of the localization was less than four meters, while the average time of localization was less than one minute.

Table 6.1: Localization Statistics Static Source Location

	Mean	Stand. Dev.	Min	Max
<b>Time (seconds)</b>	57	14	36	98
<b>Error (meters)</b>	3.95	1.35	1.75	7.38

To verify the robustness of the algorithm, the source location was varied around the search area. Figure 6.1 shows the locations used for the algorithm testing. Each “x” represents a test location. In each case, the helicopter began from the center of the search box (125,125).

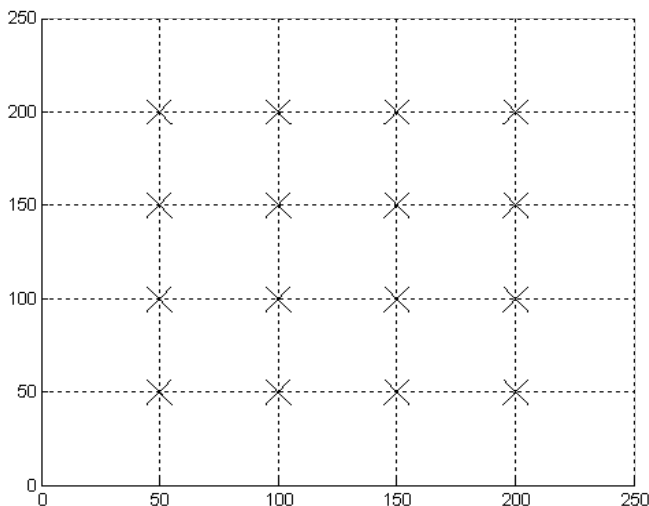


Figure 6.1: Source Locations for Testing

Table 6.2 shows the results of tests run at various source locations. The source was

localized successfully in each case. The average localization time was 45 seconds and the average error was 3.7 meters.

Table 6.2: Localization Statistics Varying Source Locations

	Mean	Stand. Dev.	Min	Max
<b>Time (seconds)</b>	45	10	26	69
<b>Error (meters)</b>	3.7	1.3	1.3	7.8

The results of these simulations show adequate capability to fly the experiment on the Lab’s RMAX helicopter. The tests prove the repeatability of the algorithm, as well as the ability to localize the source well within the 45-minute flight time provided by the RMAX. The next section describes the experimental setup and results.

## 6.2 Experimental Setup and Results

The localization experiment was performed at the Kentland Farms test site. The experimental setup followed the procedure detailed in Section 3.4. The only required equipment was the RMAX and two laptop computers. One computer ran the WePilot ground station software, and the other ran the localization algorithm. The laptops have no special requirements, as the algorithm is not computationally intensive. With the help of a computer engineer at the Unmanned Systems Lab, the algorithm was optimized to run in less than 0.01 seconds per iteration on an Intel Core 2 Duo with 2GB of ram.

Figure 6.2 shows the experiment at various time steps. The RMAX began approximately 60 meters horizontally from the simulated source (shown as a red dot). Figure 6.2(a) shows that the particles are randomly distributed initially. Within nine seconds (6.2(b)), the algorithm has identified a “ring” in which the source may lie. By 25 seconds (6.2(d)), the particles have converged around the source location. The black dot represents the estimated source location. By 30 seconds, the algorithm is very certain of the source location. Part

6.2(f) shows the helicopter moving toward the source. Notice that the helicopter travels backward toward the source. This is because the helicopter is in “heading hold” mode. Normally, it would align the heading with the velocity vector.

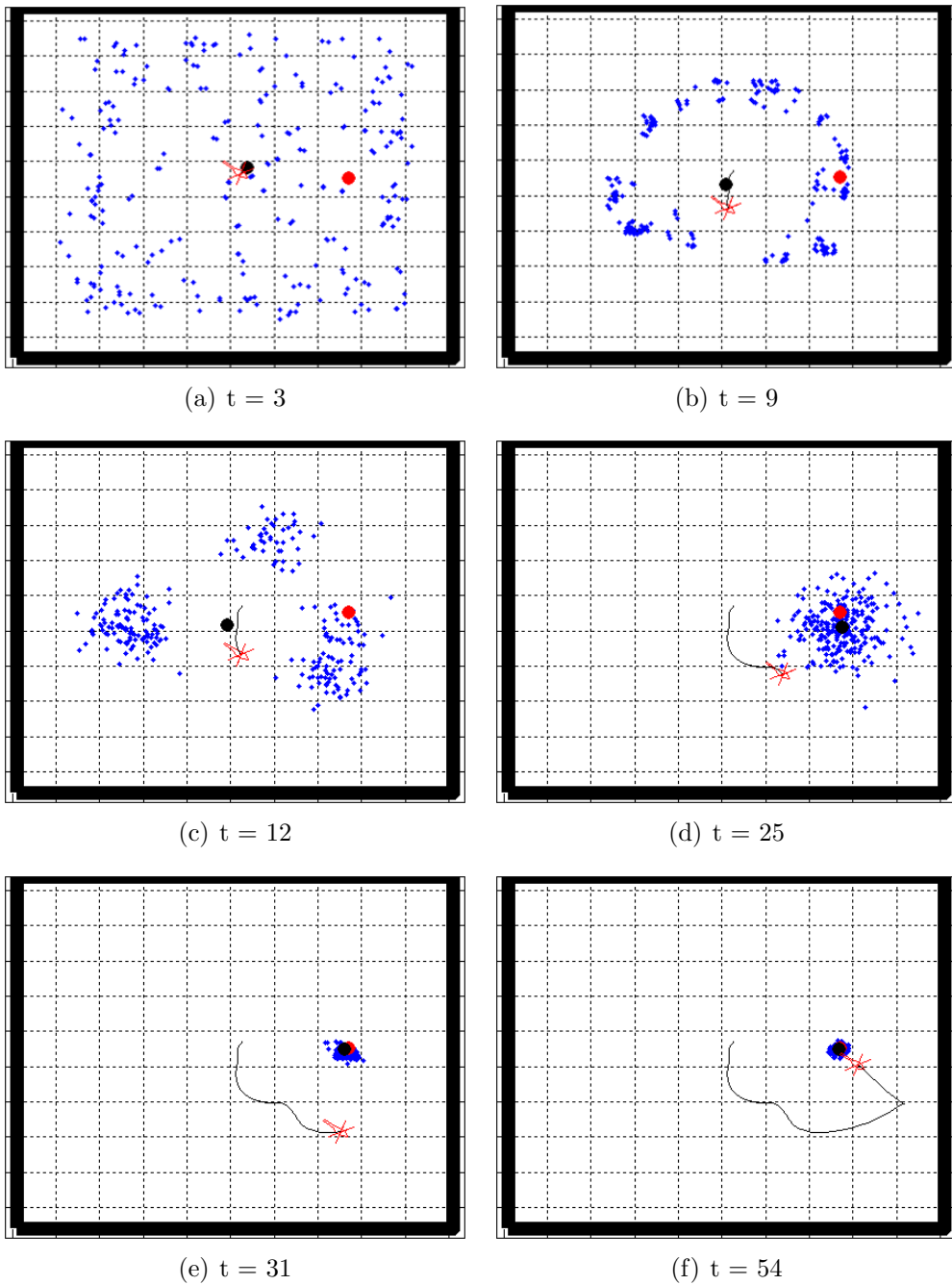


Figure 6.2: Experiment With Simulated Sensor Data



The source localization took less than 60 seconds from start to finish. The error in this case was less than one meter. The experiment was successful in verifying the functionality of control interface with the helicopter as well as the autonomous localization algorithm. While the algorithm was successful under simulated uncertainties, further testing will be required to verify its use with a real radioactive source.

# Chapter 7

## Future Considerations

The work presented in this thesis is the preliminary step in an ongoing project at Virginia Tech. The hope is that this research will provide a foundation for future developments in autonomous localization. This chapter outlines suggestions for enhancements and extensions to be added in future work.

### 7.1 Point Source Experiment

As mentioned in Chapter 3, further testing of the detector model will be necessary before performing localization experiments. Once the RMAX is equipped with an altitude measurement device, it is recommended that the detector be tested at various altitudes over a known source. The aerial testing should reduce the errors caused by ground scatter. The data can be used to characterize the detector response for accuracy and measurement variance. The variance can be used to tune  $\sigma$  in the particle filtering algorithm (Line 8 of Algorithm 1). The new characterizations can be added to the developed simulations for testing before experimentation.

## 7.2 Sources of Unknown Intensity

One relatively easy enhancement to the algorithm presented is to extend it to search for point-sources of unknown intensity. This would be particularly useful, as the type and size of the source likely will not be known. This would only add one dimension to the complexity of the algorithm presented. Instead of particles with only a spacial location, the particles would also include an intensity (e.g.  $x_k = \{x, y, I\}$ ). One potential problem may be “ghost” measurements caused by sources of different intensities. For example, a very strong source that is far away may predict the same counts as a weak source at a close distance. The author suspects that this problem can be resolved in a similar manner to the ghost measurement problem in Chapter 4.

## 7.3 Proposed Scanning Strategies

For the mission scenario presented, it is likely that a higher altitude mapping scan (see Chapter 3) will be performed before the localization algorithm is executed. This will provide a starting-point for the localization algorithm. The high-level scan can be incorporated as initial weights for the belief space. For example, if there is a “hot spot” in one part of the map, the belief space can be initialized with a higher density of particles and higher weights in that area, instead of using uniformly weighted random particles as a starting point. This should improve both the efficiency and accuracy of the localization algorithm.

As mentioned in Section 5.4, the time-criticality of the mission may not permit a high-altitude scan. In this case, a different approach should be developed to determine when to start the localization algorithm.

## 7.4 Multiple Source Localization

The algorithms presented here also have promise for use with multiple source distributions. A directional detector would make these algorithms much more efficient. Because of the cylindrical geometry of the current detector, it already possesses some (limited) directionality. Figure 7.1 shows this directionality at various altitudes. As the figure indicates, the directionality is heightened at lower altitudes. With this in mind, the localization experiment was performed at sufficient altitude to make these effects negligible. Lower altitude search algorithms may be able to take advantage of this property.

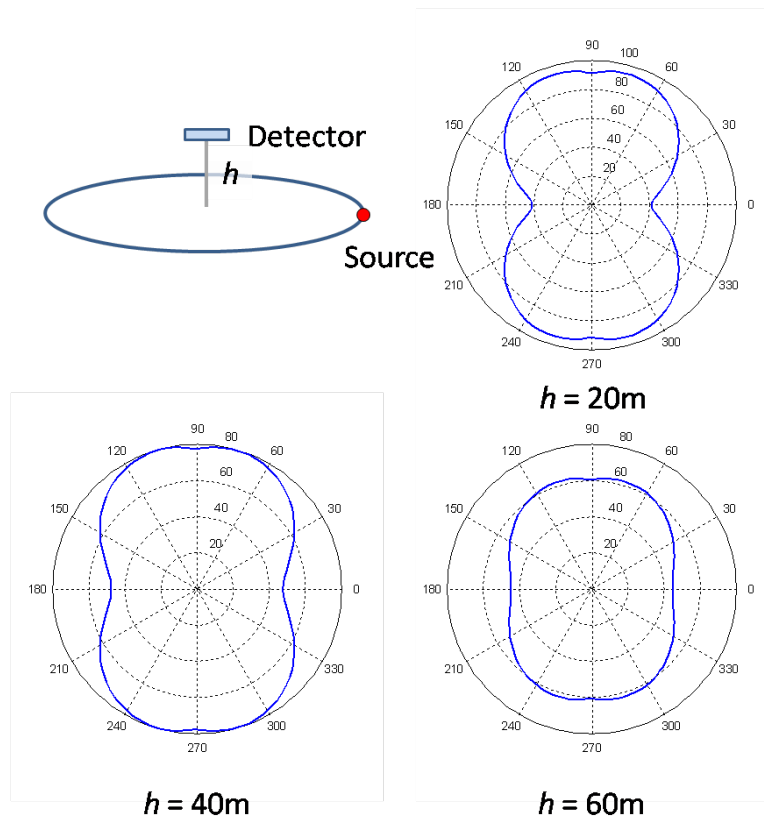


Figure 7.1: Directionality of the Detector  
 Figure Courtesy of Steve Martin, SNL

There are other detector technologies available that offer more distinct directional responses. One such design is shown in Figure 7.2. The design includes four detector crystals separated by tungsten shielding. Consider a source placed to the right of the detector. The

crystals on the right side would experience more radiation because the radiation must pass through the tungsten shielding before it hits the crystals on the left side. The array offers significant directionality, which could greatly enhance mapping and localization algorithms.

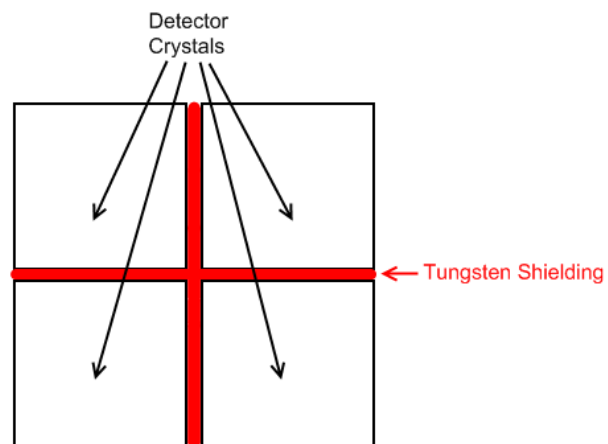


Figure 7.2: Directional Detector Technology

# Chapter 8

## Conclusions

This thesis has presented the research and development of tools to aid in the autonomous mapping and localization of  $1/R^2$  sources. A summary of the findings is presented below.

- An autonomous mapping experiment was performed with a radioactive source using a gamma radiation detector.
- A particle swarm particle filtering algorithm has been developed for the localization of a  $1/R^2$  point source.
- A software interface with Virginia Tech's autonomous helicopter has been developed for the experimental implementation of the algorithm.
- A graphical user interface has been developed to make the use of the system more intuitive.
- The localization algorithm was successfully implemented experimentally on the Yamaha RMAX using simulated sensor data.
- The helicopter began 60 meters from the point source and was able to autonomously localize the source within one meter in less than a minute.

These developments provide a solid foundation for future localization work at the Unmanned Systems Laboratory at Virginia Tech.

# Bibliography

- [1] Li, X., Wang, X., Li, Q., and Zhao, H., “Localization algorithm and error analysis for micro radio-localizer (Proceedings Paper),” .
- [2] Reed, J., *Approaches to Multiple-source Localization and Signal Classification*, Master’s thesis, Virginia Polytechnic Institute and State University, May 2009.
- [3] Gunatilaka, A., Ristic, B., and Gailis, R., “Radiological source localisation,” Tech. rep., Defence Science and Technology Organisation, Aeronautical and Maritime Research Laboratory, P. O. Box 4331, Melbourne, Victoria, 3001, Australia, 2007.
- [4] Gunatilaka, A., Ristic, B., and Gailis, R., “On localisation of a radiological point source,” *Information, Decision and Control, 2007. IDC’07*, 2007, pp. 236–241.
- [5] Kumar, A., Tanner, H., Klimenko, A., Borozdin, K., and Priedhorsky, W., “Automated sequential search for weak radiation sources,” *14th IEEE Mediterranean Conference on Control and Automation, Ancona, Italy*.
- [6] Cortez, R., Papageorgiou, X., Tanner, H., Klimenko, A., Borozdin, K., and Priedhorsky, W., “Experimental implementation of robotic sequential nuclear search,” *Proceedings of the IEEE 15th Mediterranean Conference on Control and Automation (June 2007)*.
- [7] Howse, J., Ticknor, L., and Muske, K., “Least squares estimation techniques for position tracking of radioactive sources,” *Automatica*, Vol. 37, No. 11, 2001, pp. 1727–1737.
- [8] Caballero, F., Merino, L., Maza, I., and Ollero, A., “A particle filtering method for wireless sensor network localization with an aerial robot beacon,” *Proc. IEEE ICRA*, 2008, pp. 596–601.
- [9] Idica, T. and Ohya, A., “Target user localization for a service robot using wireless LAN,” *SICE, 2007 Annual Conference*, 2007, pp. 351–355.
- [10] Chin, J., Yau, D., Rao, N., Yang, Y., Ma, C., and Shankar, M., “Accurate localization of low-level radioactive source under noise and measurement errors,” *Proceedings of the 6th ACM conference on Embedded network sensor systems*, ACM New York, NY, USA, 2008, pp. 183–196.



- [11] Knoll, G., *Radiation detection and measurement*, Wiley New York, 1989.
- [12] “Convolution,” Dec 2009, <http://en.wikipedia.org/wiki/Convolution>.
- [13] Smith, S., “The Scientist and Engineer’s Guide to Digital Signal Processing,” Dec 2009, <http://www.dspguide.com/ch17/2.htm>.
- [14] Bekey, G., *Autonomous robots: from biological inspiration to implementation and control*, The MIT Press, 2005.
- [15] Webb, S. and Furukawa, T., “Belief driven manipulator control for integrated searching and tracking,” *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006.
- [16] Thrun, S., *Probabilistic robotics*, ACM New York, NY, USA, 2002.
- [17] Arulampalam, M., Maskell, S., Gordon, N., and Clapp, T., “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” *IEEE Transactions on signal processing*, Vol. 50, No. 2, 2002, pp. 174–188.
- [18] Trelea, I., “The particle swarm optimization algorithm: convergence analysis and parameter selection,” *Information Processing Letters*, Vol. 85, No. 6, 2003, pp. 317–325.