Cooperative Searching and Geomagnetic Surveying Using Teams of Autonomous Agents

Christopher Lum

A research proposal submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

University of Washington

2006

Program Authorized to Offer Degree: Aeronautics & Astronautics

University of Washington

Abstract

Cooperative Searching and Geomagnetic Surveying
Using Teams of Autonomous Agents

by Christopher Lum

Chair of Supervisory Committee:

Assistant Professor Rolf Rysdyk
Aeronautics & Astronautics

Search missions are often complex and difficult operations due to dynamics in the environment. Targets may not be stationary and observations become less reliable as time progresses. In addition, searches are often initiated with only a rough idea of the target's location. This work considers using a team of heterogeneous agents to search for targets based on their magnetic signatures. The system maintains a world model which includes the estimate of possible target states and state of the environment. The issue of compelling agents to converge on possibly moving targets and continuing to search new regions is formulated as a model predictive control problem. The world model is propagated forward in time and autonomous strategic decisions are made based on the predicted future state of the world. Agents formulate control decisions for a fixed number of time steps by optimizing a team based objective function which allows for control and timing constraints.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Background and Motivation

Concurrent operation of multiple vehicles is currently limited by the number of operators required and the associated operator workload. Lack of autonomy of the vehicles in performing tasks such as path planning and target identification in the face of rapidly changing conditions severely increases operator workload. In many situations, the size of the crew required to operate these "unmanned" systems exceed that of traditional manned systems. Adaptive real-time mission planning algorithms onboard miniature unmanned aerial vehicle (UAV) platforms give the agents greater autonomy, thereby significantly reducing operator requirements. Algorithms for solving real-time task and path planning (TPP) problems were developed at the University of Washington under Defense Advanced Research Projects Agency (DARPA) funding. Target tracking algorithms have also been developed under UW RRF funding.

The specific mission of interest for demonstration is cooperative magnetic tracking of submarines in shallow waters. The functionality of this system will be extendible to a wide variety of intelligence, surveillance, and reconnaissance (ISR) missions and UAV configurations (e.g. forestry patrol, pipeline patrol, border patrol, maritime search, fishery patrol). In the Homeland Security Presidential Directive (HSPD)-5, President Bush directed the development of a National Response Plan for response to domestic incident management of radiological or hazardous chemical or biological release incidents. The algorithms developed for the magnetic anomaly detection and tracking have direct application to hazardous substance detection and tracking. The Environmental Protection Agency (EPA) and Department of Energy (DOE) have indicated interest in evaluation of UAVs for these applications.

The development of these autonomous algorithms will increase the performance of these agents, thus providing a direct outlet for military and commercial market demands requiring multiple vehicles. Currently, these types of operations are restricted to the experimental robotics research community.

1.2 Problem Statement

The overall objective of this work is to develop a robust and scalable control laws to apply to groups of agents so that they operate in a cooperative fashion to achieve a common goal. The specific goal in this application is coordinating a group of agents to search a two dimensional space for a target based on its magnetic signature. Although this goal may seem narrow in scope, many of the technologies developed for this application are easily adaptable to other more general tasks encountered in autonomous systems.

The autonomous algorithms developed in this endeavor fall into many different categories. The algorithms are differentiated by their complexity and level of autonomy involved. The three levels of autonomy are shown below in Figure 1.1.

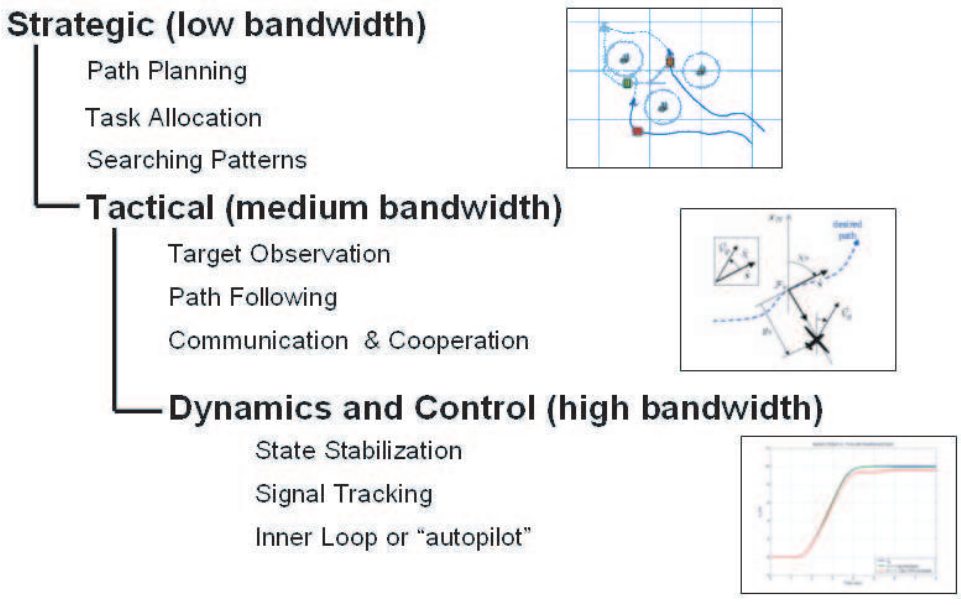


Figure 1.1: Different levels of autonomy.

Figure 1.1 shows the first level of autonomy as the strategic level. These algorithms often operate at a low bandwidth and may be in charge of tasks such as mission planning. They are sometimes referred to as "high level control schemes". Next, the tactical level of control operates underneath the strategic level. This level of control is concerned with more specialized tasks such as path following or orbit coordination. Lastly, the dynamics and control level consists of the classical control problems such as state stabilization or signal tracking. These are the most specialized and "narrow-minded" and operate at a high bandwidth. A successful autonomous system uses algorithms from all different levels that work together to achieve a common goal.

1.3 Document Layout

This proposal is partitioned into two main sections. Chapter 2 covers previous developments and research performed while Chapter 3 discusses future endeavors.

In Section 2.1, other pertinent work is reviewed including other academic contributions towards this research. Section 2.2 details a feasibility study of this problem is conducted to ensure that this problem is reasonable and possible to solve using currently existing technology. The target identification algorithm is described in Section 2.3. A genetic algorithm which is used to solve a subset of standard optimization problems is then covered in Section 2.4. The concept of an occupancy map and its uses in the algorithm is introduced in Section 2.5. A simple example of a multi-agent search mission using these method is described in Section 2.6. This simple behavior is expanded to a more sophisticated search strategy for a single vehicle in Section 2.7. Finally, a list of publications based on previous research is presented in Section 2.8.

Chapter 3 details where the project is headed in terms of ideas to pursue and develop. Section 3.1 describes how some of the previous developments regarding the searching system can be improved to increase functionality or efficiency of the system. Section 3.2 specifies how the world model can be updated to take into account time varying effects. Section 3.3 covers how an aeromagnetic survey could be carried out autonomously using a team of agents. Section 3.4 covers the difficulty in obtaining true magnetic signature of a target

and how this can be accomplished. Section 3.5 talks about the possibility of using game theory and emergent behavior ideas in order to have single agents solve simple problems which yield more complicated behavior of the overall system. Section 3.6 looks at some of the possible problems and solutions regarding communications in a large sensor network. Section 3.7 discusses how an actual flight test might be implemented along with some of the goals of such a test. Finally, Section 3.8 gives an expected time line for some of the major tasks.

Chapter 2

PREVIOUS WORK

2.1 Literature Review

The target identification algorithm utilizes a method known as a particle filter. This builds on the ideas presented by Fox et al. [6] where a particle filter is used to localize a vehicle in an environment. Other groups such as Durrant-Whyte et al. [2] studied the problem of searching for a target using a Bayesian probabilistic approach and investigated some of the communication issues involved in such a search. Polycarpou et al. [5] applied optimization techniques to generate search patterns over a finite number of steps. The search strategy presented here follows a similar approach and investigates the effect of incorporating a predictive world estimate to the problem of finding an optimal search pattern.

Previous work at the University of Washington established a framework for the integration of various tasks in an autonomous system. This involves classifying tasks as either a strategic, tactical, or dynamics and control problem. These correspond to low bandwidth tasks such as path planning [12], medium bandwidth tasks such as target identification [11], and high bandwidth tasks such as state stabilization, respectively. This work focuses on integrating the strategic task of coordinating multiple agents to search an area with the tactical task of performing target identification on unknown anomalies that are encountered during a search.

2.2 Feasibility Study

As stated previously, the goal of this work is to enable a team of agents to autonomously search and identify a target based on its magnetic signature. In order to do this, certain technologies must be in place to allow this to be feasible. Before devoting significant effort towards this goal, it is pertinent to perform a feasibility study to determine if this goal is

achievable with currently existing technology.

2.2.1 Magnetometer Capabilities

One of the most important pieces of equipment for this application is the sensor that is used to measure magnetic field strength. In this work, each agent is assumed to be equipped with a cesium-vapor scalar magnetometer and a fluxgate vectored magnetometer. These sensors measure the absolute field strength and the three component magnetic field vector, respectively. A typical fluxgate vectored magnetometer can measure fields in the range of 0 to 2,000,000nT with a resolution of better than 1nT. Furthermore, the measurement signal is expected to have a noise profile of 0.1nT.

A reasonable model of the magnetic field measured by an agent is given by the linear superposition of various sources [4].

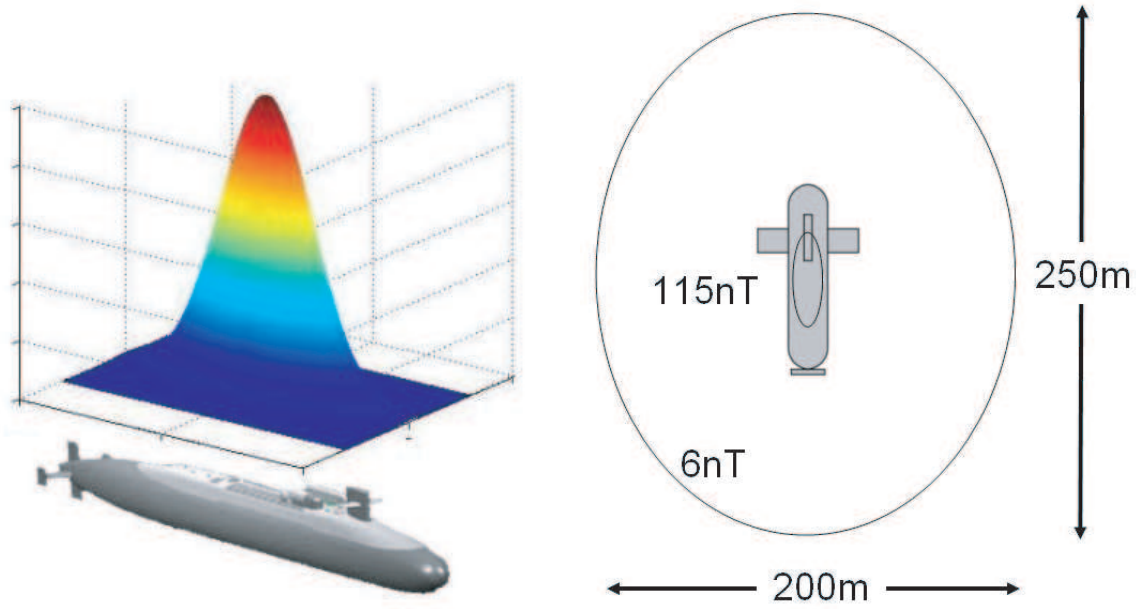
$$\overline{B} = \overline{b}_{tgt} + \overline{b}_{m} + \overline{b}_{uav} + \overline{b}_{var} \tag{2.1}$$

The magnetic field at the sensor, $\overline{B}$, is a linear combination of several different magnetic fields. The magnetic field predicted by an analytical model such as the International Geomagnetic Reference Field (IGRF) or the 2000 World Magnetic Model (WMM-2000) is given by $\overline{b}_m$; the magnetic field created by the target is given by $\overline{b}_{tgt}$; the magnetic field induced by the agent itself is given by $\overline{b}_{uav}$; and unaccountable magnetic field variations are given by $\overline{b}_{var}$. One of the main goals of this work is to be able to discern the magnetic signature of the target, $\overline{b}_{tgt}$ from the other sources. In order to do this, each term must be analyzed and compared to the capabilities of the sensors.

2.2.2 Target Magnetic Signature

The magnetic field intensity created by the target is the most interest and also is the signature which is the most difficult to model. Modeling the submarine as an ellipsoid 80m long, 10m in diameter, at depth of 25 meters, and the sensor at an altitude of 25m above the surface, the expected peak magnetic anomaly is approximately 115nT [8]. The profile falls off quickly and at 100m to either side of the submarine, the magnetic anomaly is reduced to

roughly 6nT. An example profile with the expected level sets is shown below in Figure 2.1.



(a) Magnetic intensity of sub $||\overline{b}_{tgt}||$ (b) Level sets of magnetic profile

Figure 2.1: Magnetic signature of submarine.

Currently, the submarine signature is only a function of the (x, y) coordinate ($f : \Re^2 \rightarrow \Re$). A more sophisticated model would take depth and altitude into account. The function would then be a function of four parameters. This formulation is easily taken into account in the target identification algorithm described in Section 2.3.

2.2.3 Mean Magnetic Intensity

The magnetic field at all points on the earth is modeled fairly accurately using analytical models such as the IGRF or WMM-2000. These models provide the three component estimated magnetic vector at any coordinate on the planet. The mean magnetic intensity of the magnetic field varies from roughly 30,000nT to 65,000nT. An example of the output of one of these models at the Boardman, OR bombing range (coordinates 47.780°N, 119.708°W) is shown below in Figure 2.2.

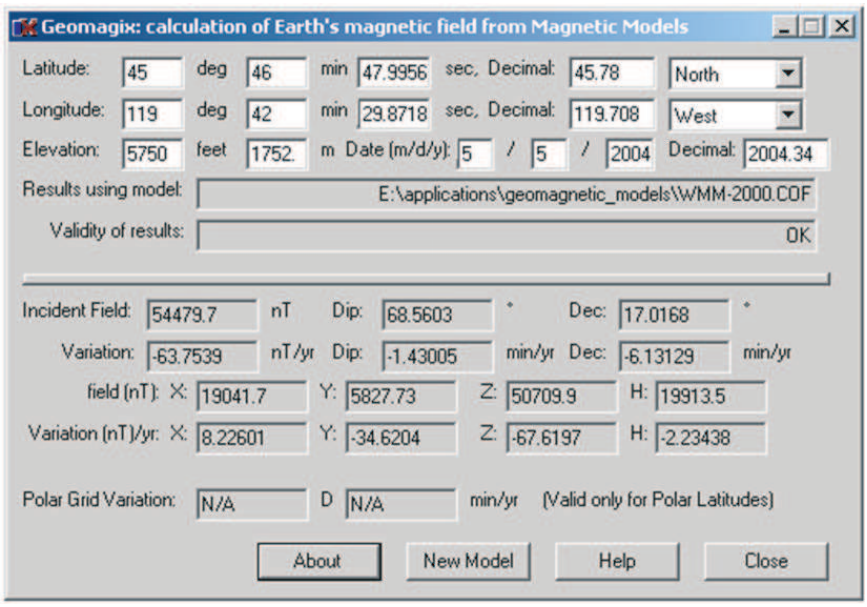


Figure 2.2: Predicted magnetic field at the Boardman, OR bombing range.

2.2.4 Self Induced Magnetic Field

The agent creates its own magnetic field which is measured by the sensor. The static effect of the agent can easily be calibrated and subtracted from the sensor readings. It is more difficult to characterize the dynamic effects of actions such as changing engine RPM or servo actuation on the magnetic field. These effects are shown below in Figure 2.3. In this figure, the effects of different actions on the magnetic field are shown for different sensor locations. The red line denotes data collected when the magnetometer was closest to the agent's core whereas the purple line corresponds to moving the magnetometer as far away as possible on the agent. Due to proprietary concerns, the scale of these readings are not shown; but this suffices to show that this self-induced magnetic field is present and may have an effect on the measured magnetic field. If an agent is optimized to make magnetic readings, these dynamic effects can be minimized.

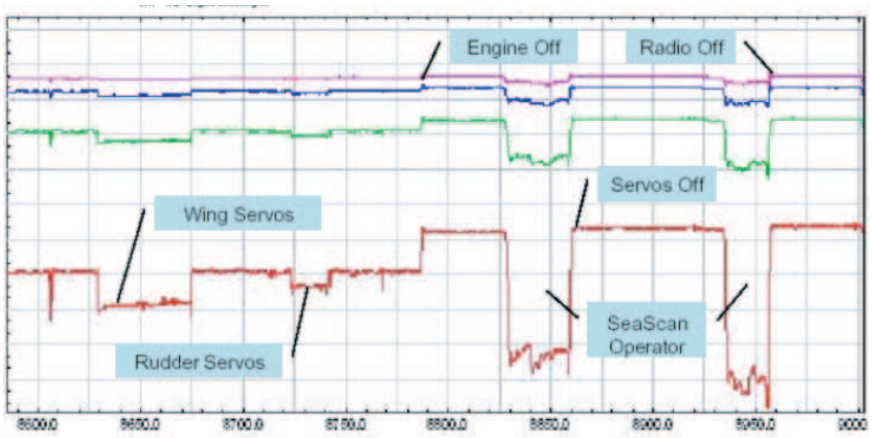


Figure 2.3: Self induced magnetic field effects.

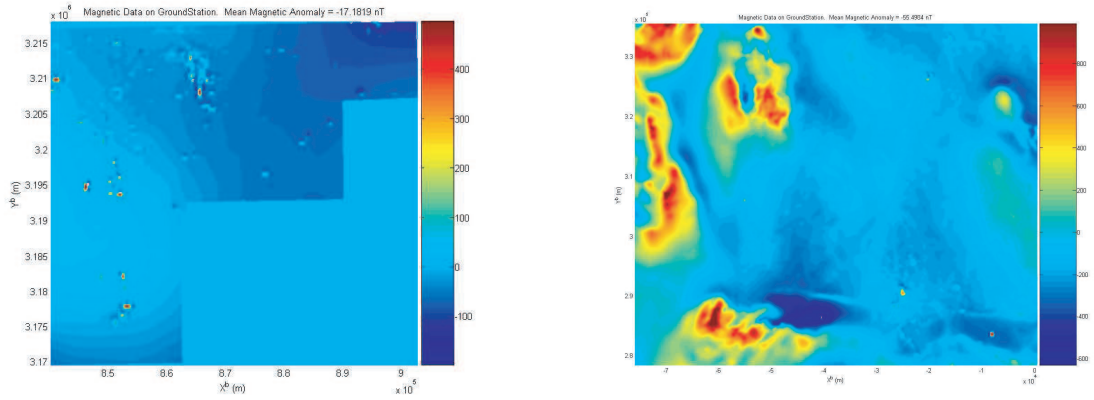
2.2.5 Magnetic Field Variation

The magnetic field also has variations due to other unaccountable factors. These can include effects such as short period magnetic fluctuations (1-2nT), long term drift (60nT/year), and unaccounted magnetic signatures of structures or land formations. The effect of the short period magnetic fluctuations can be satisfactorily modeled as white noise with amplitude 1-2nT. The long term drift is addressed in Section 3.3. The unaccounted magnetic signatures of structures and land formations are characterized by a total magnetic intensity (TMI) map. This map is function which maps an (x, y) coordinate to an scalar which represents the deviation from the predicted magnetic field strength ($T : \Re^2 \rightarrow \Re$). The graph of this function is essentially the resultant field after correcting the observed field for a regional gradient field (i.e. IGRF or WMM-2000).

$$T(x) = ||\overline{B}_{observed}(x) - \overline{b}_m(x)|| \tag{2.2}$$

Two examples of actual TMI maps are shown below in Figure 2.4. Since these surveys are obtained using specialized vehicles, the self-induced magnetic field is negligible ($\overline{b}_{uav} \approx 0$) and there are assumed to be no targets present during the survey ($\overline{b}_{tgt} = 0$) so the TMI map gives and measure of accuracy of the predictive models. The areas of light blue are where the magnetic anomaly is near zero. This implies that the measured magnetic field measured

is very close to the the field predicted by a regional model. However there are many regions where the measured field is significantly different than the predicted field. In Figure 2.4(b), the magnetic anomaly ranges from roughly -600nT to 900nT. Since the peak anomaly of the target is expected to be around 115nT, the predictive regional models cannot be relied upon and an actual survey of an area must be obtained in order to discern the $\overline{b}_{tgt}$ from $\overline{b}_{var}$.



(a) 76.4km by 57km map near the Gulf of Mexico

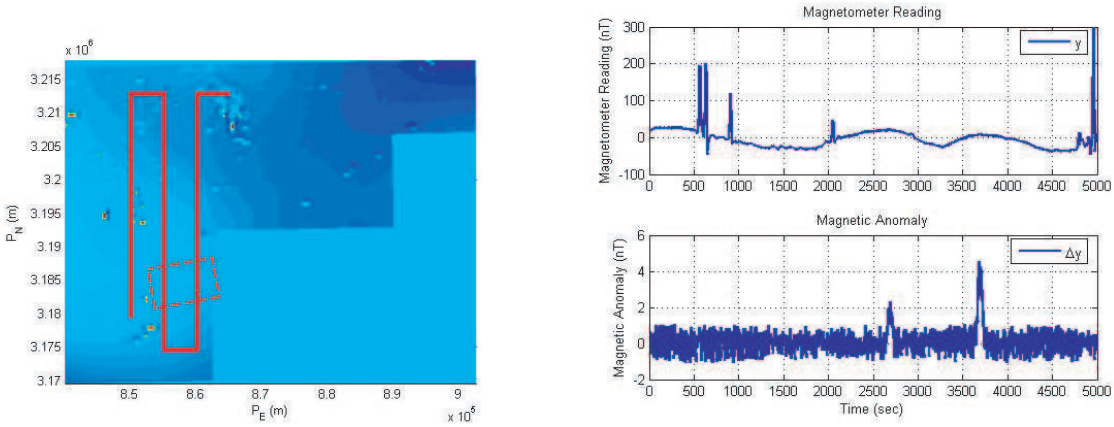
(b) 62.5km by 48.2km map over Puget Sound

Figure 2.4: Total magnetic intensity maps.

Once again, this map is only a function of two variables (namely the (x, y) coordinates). This forces the agents to fly at the same altitude that the original survey vehicle flew at when the map was acquired. This map would need to be corrected for altitude variations or multiple maps at different altitudes would need to be obtained. This makes this map more complicated (a function of three or more variables) but would increase the functionality and versatility of the system.

2.3 Target Identification

Magnetic anomalies can be caused by many factors such as temporal variations in the magnetic field or false targets encounters (i.e. boats/vessels). Once a magnetic anomaly is encountered, it must be identified and classified.



(a) Total magnetic intensity map.

(b) Associated magnetic traces.

Figure 2.5: The total magnetic intensity map and trajectory over area with corresponding magnetometer readings.

When an actual search is executed, differences between the ground station map of the magnetic field and the actual magnetic field will appear as magnetic anomalies. A TMI map of a region in the Gulf of Mexico and a simple grid search trajectory are shown below in Figure 2.5(a). Here, the data is acquired in an approximate 60x50 km grid. The regions of uniform color denote areas where survey data is not available, creating the "staircase" appearance. Assuming that there are only permanent fixtures in the region when the map is acquired, this map (in combination with the predictive magnetic model used to correct it) now constitutes the reference set of data.

If the agent does not fly over any objects, the magnetic anomaly should be near zero. Small non-zero anomaly encounters can be attributed to temporal variations in magnetic field and sensor noise. A simple grid search pattern is shown in Figure 2.5(a). The location of the target is shown as a dashed red box and the trajectory of the agent is shown in the solid red line (starting in the lower left corner). The associated total magnetic intensity trace and differential measurement trace is displayed in Figure 2.5(b). The total magnetic intensity reading as the agent flies over this trajectory is shown in the upper trace and the differential measurement is shown in the lower trace. As the agent flies this search trajectory, the sensor measurement is constantly compared to the reference data set to generate a differential

measurement. As Figure 2.5(b) demonstrates, given the differential magnetometer reading, it is obvious how to detect where the anomaly occurred (two spikes at approximately 2700 and 3700 seconds) even though the actual range of absolute measurements may be large.

The goal now is to either classify the anomaly as the desired target or a false anomaly. Obviously, it would be simple to identify the anomaly if the entire magnetic signature of the anomaly is obtained (i.e. the UAV flies over the entire boxed region in Figure 2.5(a)). However, this requires many passes over a potential target, and significant time to make the necessary measurements. If the anomaly is moving or evading, this may not be feasible. The question now becomes: given only one or two passes over the target, is it possible to correctly identify or provide a probability that this anomaly is indeed the target in question? To address this issue, a particle filter method is used.

A particle filter is a recursive, non-parametric Bayes filter technique which estimates the states of a system using a finite number of state hypotheses [16]. In this situation, the state vector that is being estimated is the position of the agent with respect to the target, expressed in the target's frame of reference and the relative heading of the agent with respect to the target.

$$\overline{x}_t^{[m]} = \begin{bmatrix} x_{uav/tgt}^{tgt} \\ y_{uav/tgt}^{tgt} \\ \psi_{uav/tgt} \end{bmatrix} \tag{2.3}$$

Each individual state hypothesis, $\overline{x}_t^{[m]}$, is referred to as a particle, and together they make up the particle filter set, χ_t.

$$\chi_t = \bigcup_M \overline{x}_t^{[m]} = \left\{ \overline{x}_t^{[1]}, \overline{x}_t^{[2]}, \ldots, \overline{x}_t^{[M]} \right\} \tag{2.4}$$

GPS allows the position of the agent in the earth frame to be computed, but the target location and orientation in the earth frame is not always known. The goal of the particle filter is to estimate the state of the agent (position and orientation with respect to the target, expressed in the target's frame of reference). The true location of the agent with respect to the target expressed in the target's frame of reference at a time t is denoted as $\overline{x}_t^*$.

The particle filter performs this estimate using two main steps, a prediction and correction step.

2.3.1 Prediction

In the prediction step, each particle is propagated forward in time using a motion model of the individual agent.

$$\overline{x}_t^{[m]} = g\left(p\left(\overline{x}_t^{[m]}|\overline{u}_t^*, \overline{x}_{t-1}^{[m]}\right)\right) \tag{2.5}$$

In Eq. 2.5, g is a sampling function which simply chooses a sample from a probability density function. Each new particle is created from the old particle and the current control (applied to transition particle at time $t-1$ to time t). The term $p\left(\overline{x}_t^{[m]}|\overline{u}_t^*, \overline{x}_{t-1}^{[m]}\right)$ is a multi-dimensional probability density function of the new state given the old state and current control. Notice that in this formulation, the state transition is not a deterministic process. This stochastic aspect actually has important implications regarding the robustness of the particle filter [16].

Although $p\left(\overline{x}_t^{[m]}|\overline{u}_t^*, \overline{x}_{t-1}^{[m]}\right)$ may be difficult to compute analytically, Eq. 2.5 is implemented in simulation by simply adding noise to the control and then propagating the state forward using a deterministic motion model (a simple kinematic model in this case). The control vector for the model is simply

$$\overline{u}_t = \begin{bmatrix} V_a \\ \Delta\psi \end{bmatrix} \tag{2.6}$$

In simulation, the noise added to each element of the control vector is obtained by sampling from a normal, Gaussian distribution with a variable standard deviation, σ. The standard deviation is a function of the actual control applied to the agent, $\overline{u}_t^*$. In effect, as $||\overline{u}_t^*||$ increases, so does σ. Physically, this translates into a model whose state transition becomes more uncertain as the agent moves faster or executes larger heading changes.

In addition to the control input at each time step, the actual sensor measurement observed by the agent, z_t^*, is made available to the particle filter. Each particle is then assigned

a weight, $w_t^{[m]}$, based on how likely it is to generate the same sensor measurement at its current state.

$$w_t^{[m]} = p\left(z_t^* | \overline{x}_t^{[m]}\right) \tag{2.7}$$

In effect, this assigns high weights to particles whose states are close to the actual state, $\overline{x}_t^*$. Notice that Eq. 2.7 does not require a sampling function like Eq. 2.5 because z_t^* and $\overline{x}_t^{[m]}$ are known at this point. Eq. 2.7 describes the sensor model of the agent. It allows for the fact that even though a particle's state may be vastly different than the true state of the agent, if the sensor is poor or unreliable, it has the possibility of still making the same sensor reading as the agent.

The sensor model used in simulation calculates $w_t^{[m]}$ by creating an error between the particle sensor measurement and the true sensor measurement and then using this as the argument of a Gaussian distribution with zero mean and a standard deviation σ.

$$w_t^{[m]} = f\left(z_t^* - z_t^{[m]}\right) \tag{2.8}$$

In Eq. 2.8, $z_t^{[m]}$ is the predicted sensor measurement made by particle m. In simulation, $z_t^{[m]}$ is only a function of the first two states and is generated using the target's magnetic signature function (Figure 2.1(a)) to obtain $z_t^{[m]} = h(x_{uav/tgt}^{tgt}, y_{uav/tgt}^{tgt})$. As stated previously f is a Gaussian distribution with zero mean and standard deviation σ which be adjusted based on the sensor model. A larger σ implies an unreliable sensor; therefore, particles that do not make the same measurement as the true agent still receive high weights. Note that the weights are not probabilities; however this still achieves the goal of assigning high weights to particles that are more likely to have states which are similar to the true agent state.

The majority of this section has discussed the state estimation problem. Historically, particle filters have been employed in this manner to perform tasks such as localization [10] and state estimation [6]. These are certainly important tasks in this problem; however, to perform target identification, a closer look at the weights, $w_t^{[m]}$, is warranted.

A scalar quantity which collectively measures the overall accuracy of the particle filter

can be obtained by simply summing all the weights. If most of the particles are in locations that are similar to the true state, then the sum of the overall weights should be large. Traces of C_t for two different situations are shown later in Figure 2.8.

$$C_t = \sum_{m=1}^{M} w_t^{[m]} \tag{2.9}$$

This trace of a C_t vs. t might be considered a side-effect of estimating x_t^*, but as will be shown later, this is the main piece of information that will be used to address the target identification problem.

2.3.2 Correction

Now that each particle has been propagated forward and assigned a weight, it becomes necessary to correct the particle filter set so that it converges to the actual state of the agent. This process is known as resampling.

As stated before, the particle filter's estimate of the state is represented by the distribution of all the particles. Currently, the particle filter set contains particles which have both high and low weights. As more and more sensor measurements are acquired, it is desired that high scoring particles are replicated and retained in the next generation population whereas low scoring particles are discarded. The important feature in this evolutionary process is that the particles are resampled with replacement so that the total number of particles remains constant at each cycle. Any type of evolutionary scheme, such as survival of the fittest, can be used to evolve the current population to the next.

In simulation, a roulette wheel method is used. In this method, M bins are created (one for each particle). The size of each bin is directly proportional to the weight of the associated particle. The bins are placed next to each other and a random number is generated. The bin in which the random number falls has its associated particle included in the next population. This process is repeated M times and is synonymous to spinning a roulette wheel M times where the number and size of the slots on the wheel are directly proportional to M and the weights, respectively.

Using the roulette wheel method yields resampling proportional to the weights. This

allows for a particle to be copied multiple times in the next generation. This also generates a small probability that particles with low weights have the possibility to survive to the next generation as well.

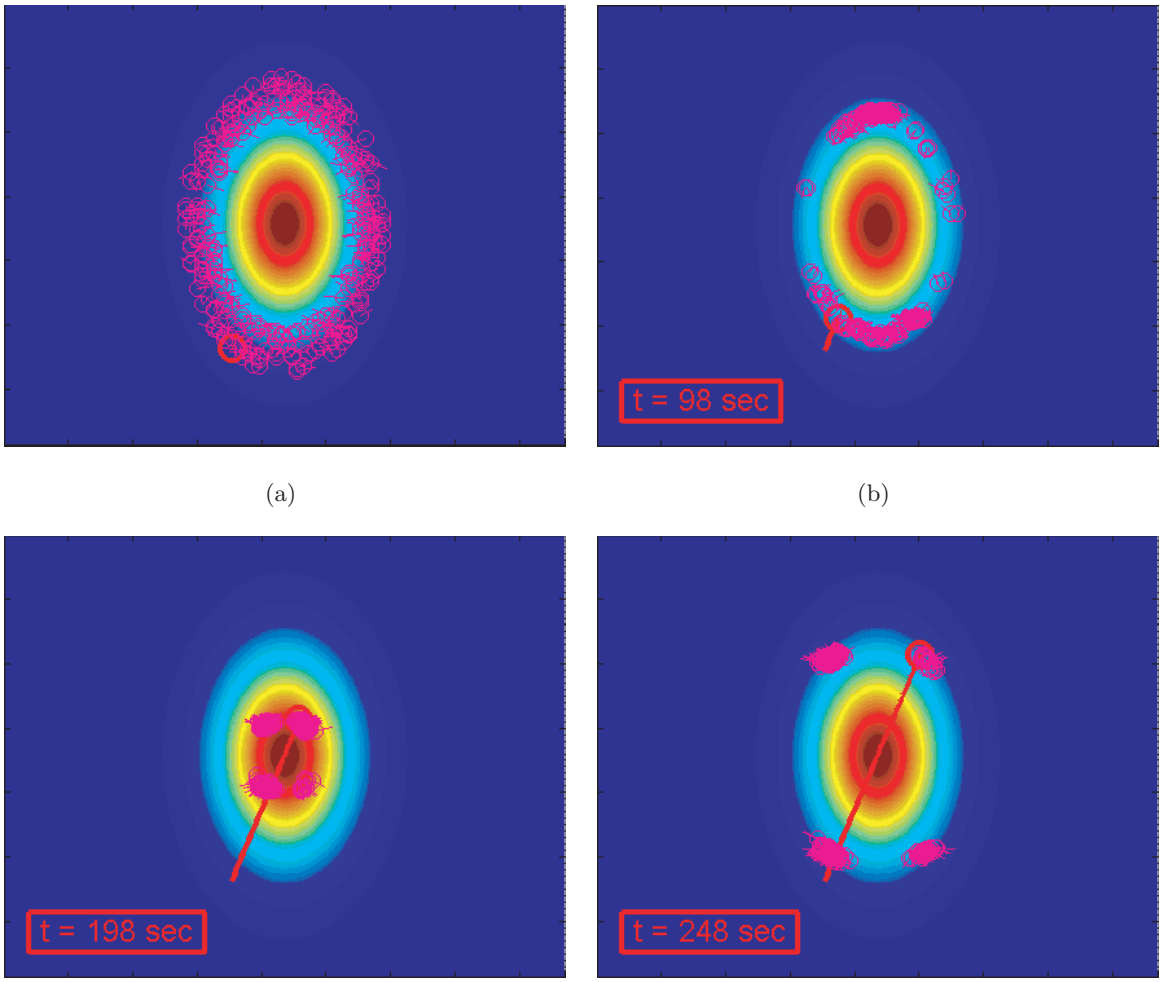
One important feature of the particle filter is the ability to use different motion and sensor models. This allows for a team of agents to be comprised of different types of vehicles and sensors. This simply requires modifying the motion and sensor models of each particle filter for each member of the heterogeneous team.

2.3.3 Execution

When an agent encounters an anomaly whose magnitude exceeds the noise threshold (approximately 1 nT in this case), the particle filter is initiated in an attempt to estimate the state of the agent with respect to the target. In addition, recall that the trace of C_t vs. t is the true end-product of the particle filter that is used for target identification. The particle filter's progression as the agent flies diagonally over the target is displayed over a top down view of the target signature (Figure 2.1(a)) and is shown below in Figure 2.6.

In this sequence, the large red circle represents the actual location of the agent and the solid red line represents the agent's trajectory over the target. The smaller purple dots represent the particle filter's many different hypotheses of the possible state of the agent (position north, position east, and heading). The actual agent crosses over the target starting in the lower left corner and flies over it to the upper right corner. Also note that the initial distribution of particles is not simply random over the domain. Since the algorithm is recursive, the number of iterations before convergence is based on its initial condition. Incorporating a priori knowledge that the particle filter is started when the anomaly magnitude exceeds 1 nT suggests that the particles be clustered along the level curves where the target signature is 1 nT.

As the agent obtains more and more sensor measurements (at a simulated rate of 1 Hz), the particle filter is able to eliminate particles which are inconsistent with the current measurement and resample these particles to regions which have a higher probability of producing the actual sensor reading, z_t^*. This is why as time progresses, the particles



(a) (b)

(c) (d)

Figure 2.6: Particle filter progression during a target encounter. The solid line indicates actual aircraft position relative to target signature, while the particles concentrate about possible positions.

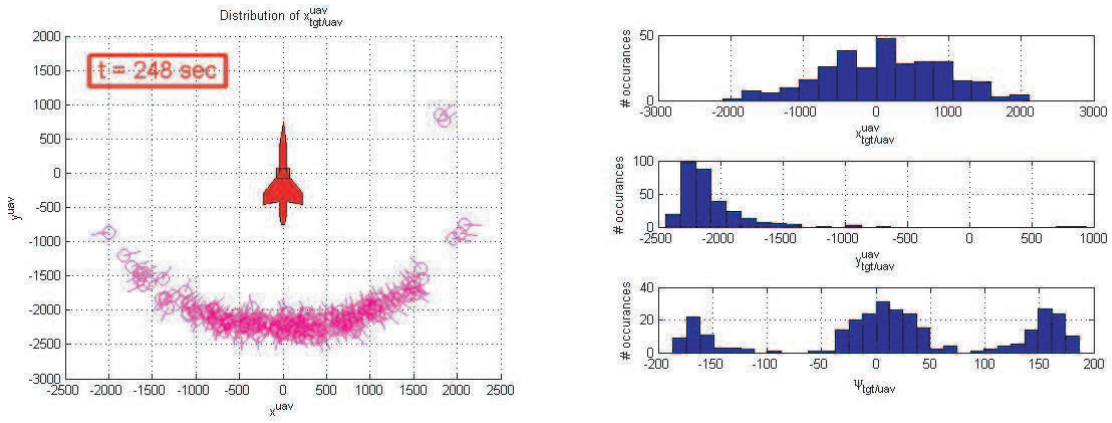
become concentrated around the actual UAV location. Near the end of the simulation, there are four distinct groups of particles. This is due to the symmetry of the underlying target signature. Each of these four groups of particles are equally likely because each group would produce the correct actual sensor readings (a trajectory from SW to NE looks the same as a trajectory from NE to SW which looks the same as a trajectory form NW to SE, etc.). In effect, $z_t^{[m]} \approx z_t^* \, \forall m$. Because of this symmetry, the particle filter is not able to

uniquely identify the position of the agent with respect to the target. This would require multiple passes over the target and more sensor measurements.

Although the goal of the particle filter is to estimate the position of the agent with respect to the target in the target frame of reference, in the larger picture, the location of the target with respect to the agent in the agent frame of reference is more useful because it then becomes simple to locate the target in the earth frame of reference (agent's position and orientation in the earth frame of reference is known from GPS). Each particle can be transformed using Eq. 2.10.

$$\begin{bmatrix} x^{uav}_{tgt/uav} \\ y^{uav}_{tgt/uav} \\ \psi_{tgt/uav} \end{bmatrix} = \begin{bmatrix} -\cos(\psi_{uav/tgt}) & \sin(\psi_{uav/tgt}) & 0 \\ -\sin(\psi_{uav/tgt}) & -\cos(\psi_{uav/tgt}) & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x^{tgt}_{uav/tgt} \\ y^{tgt}_{uav/tgt} \\ \psi_{uav/tgt} \end{bmatrix} \tag{2.10}$$

When each particle is transformed in this fashion, the distribution of the target location with respect to the agent in the agent's frame of reference is shown in Figure 2.7.



(a) Distribution of transformed particles

(b) Histogram distribution of $x^{uav}_{tgt/uav}$, $y^{uav}_{tgt/uav}$, and $\psi_{tgt/uav}$

Figure 2.7: Particles now represent position and orientation of target with respect to the agent in the agent's frame of reference.

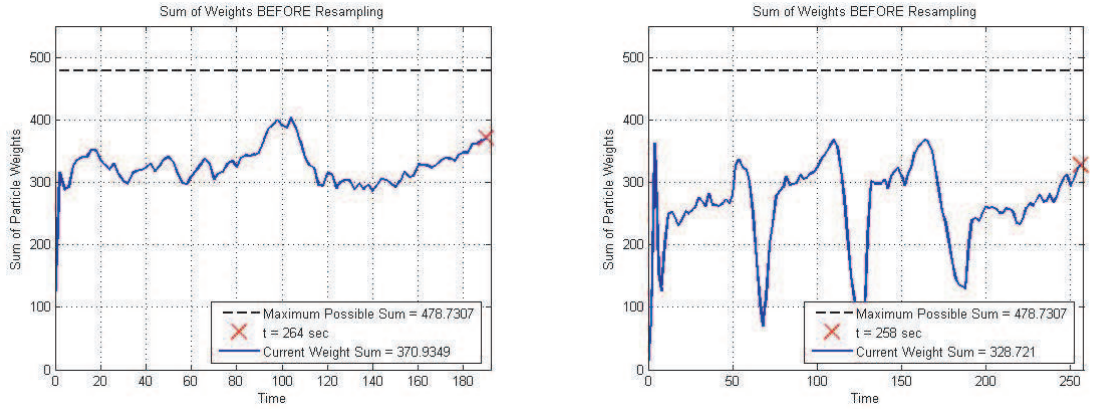
As shown in the first two plots in Figure 2.7(b), it appears that the particle filter now has a somewhat unique estimate of the location of the target relative to the agent as shown

by an approximate unimodal distribution in $x^{uav}_{tgt/uav}$ and $y^{uav}_{tgt/uav}$ centered at approximately 0 and -2250, respectively. However, notice that the distribution of $\psi_{tgt/uav}$ is obviously a multimodal distribution. This distribution is actually the sum of four peaks which should ideally be centered at ± 11.3 degrees and ± 168.7 degrees. Since the number of particles was not large enough and since the motion and sensor models of the particle filter were not highly accurate, the two peaks centered at ± 11.3 degrees appears as a single peak at 0 degrees.

This multimodal distribution in $\psi_{tgt/uav}$ reflects the four distinct state hypotheses shown previously in Figure 2.6(d). However, if the orientation of the target is not desired, then by transforming the particles, it is possible to obtain a unique estimate of simply $x^{uav}_{tgt/uav}$ and $y^{uav}_{tgt/uav}$. Note that this is only the case when the agent happens to fly directly over the target (as shown in this example). In a more general case, agents may pass over the target off-centered. In this case, even with the transformation of the particles, the location of the target cannot be determined uniquely (but the number of possible locations may be reduced).

The previously described algorithm will perform regardless if the anomaly encountered is the actual target or a false anomaly. A method to identify the target is now required. The sum of all the particle weights, C_t, provides a qualitative measure of how confident the particle filter is that the anomaly encountered is the actual target. If all or most of the particles are resampled to areas which are near the actual state of the agent, then most of the weights will be fairly high. The sum of the particle weights for an encounter with the actual target and an encounter with a false anomaly are shown below in Figure 2.8.

In Figure 2.8, the difference between a true target encounter and a false anomaly encounter is fairly clear. In the situation where the agent encounters the true target, the confidence measure increases initially as the particles are quickly resampled to locations which are consistent with the actual sensor measurements and then stays fairly constant. However, in the case where the agent encounters a false anomaly, the particle filter regularly "loses confidence" as inconsistent sensor measurements are obtained. This is characterized by the sharp drops in the sum of the particle weights. Current research is directed towards training a neural net to recognize these features and thus provide a qualitative measure to



(a) True target encounter

(b) False anomaly encounter

Figure 2.8: Sum of all particle weights during a true target encounter and a false anomaly encounter.

the target identification problem. In the end, the particle filter will provide the trace of the sum of the weights over time (Figure 2.8) and the neural net will process this trace. In combination, the particle filter and neural network provide a mapping from magnetic sensor measurements to a single scalar value which represents a measure of how confident the particle filter is that the encountered anomaly is the desired target.

2.4 Optimization Using Probability Collectives

The searching method is described in Section 2.5. Some of the ideas that are used in the searching strategy are described here. The method described here is an optimization method and is presented in its most general form. The specialization of this technique to the searching application is described in Section 2.7.2.

The algorithm proposed here attempts to solve a class of optimization problems using ideas of probability collectives [9] and particle filters [6]. This provides a algorithm which attempts to find a minimizer to the following problem

$$(\wp) \quad \underset{x \in X = \text{a box}}{\text{minimize}} \; f_0(x) \tag{2.11}$$

The set X is a box. Recall that a box is defined by each element x_k of $x \in X \subseteq \Re^n$ being in a certain interval $I_k = [l_k, u_k]$.

$$X = \left\{ x \left| \begin{array}{c} x_1 \in I_1 = [l_1, u_1] \\ x_2 \in I_2 = [l_2, u_2] \\ \vdots \\ x_n \in I_n = [l_n, u_n] \end{array} \right. \right\} \tag{2.12}$$

The difficultly in solving Eq. 2.11 arises from the fact that the objective function may not be well behaved (i.e. non-convex, non-differentiable, etc.). This is especially true in our case. It may be difficult or impossible to find an optimal solution. An algorithm to find a quasi-optimal, feasible solution is proposed below.

1. Generate M particles (instances of $x \in X$) distributed over X in some fashion.

2. Assign weights to each particle based on its objective function value.

3. Resample the particles proportional to the weights.

4. Repeat step 2 and 3 until some stopping criterion is reached.

Let us examine each step in detail.

2.4.1 Initial Particle Distribution

To find a quasi-optimal minimizer of $f_0(x)$, a finite set of possible minimizers are used. Each individual guess of a minimizer, $x^{[m]}(t)$ is called a particle and together the particles make up the particle set, $\chi(t)$.

$$\chi(t) = \bigcup_M x^{[m]}(t) = \left\{ x^{[1]}(t), x^{[2]}(t), \ldots, x^{[M]}(t) \right\} \tag{2.13}$$

To initialize the algorithm, we need to assign actual values to the initial particle set. Since there is no a priori knowledge regarding the minimizer of $f_0(x)$, the initial distribution of the particles is chosen as a uniform distribution over the set X

$$x_k^{[m]}(0) = \text{rand}(u_k, l_k) \quad \text{for} \begin{matrix} m = 1, \ldots, M \\ k = 1, \ldots, n \end{matrix} \tag{2.14}$$

2.4.2 Assign Particle Weights

We now assign a weight to each particle.

$$w^{[m]}(t) = -f_0(x^{[m]}(t)) \quad \text{for } m = 1, \ldots, M \tag{2.15}$$

Note that this formulation assigns a higher weight to particles which yields a smaller objective function value.

2.4.3 Resample Particles

In order to generate the next particle set, we sample from the current particle set proportional to the weights.

$$\tilde{x}^{[m]}(t) = g(\chi(t), w(t)) \quad \text{for } m = 1, \ldots, M \tag{2.16}$$

Here, g is a sampling function which samples elements from $\chi(t)$ proportional to the weights $w(t)$. One popular sampling method is the roulette wheel method. In this method a roulette wheel with M slots is created. The weights are normalized so that they sum to 1. Each normalized weight then represents the angular percentage that this particle occupies on the roulette wheel. The wheel is spun and depending on where it lands, the corresponding particle $x^{[m]}(t) \in \chi(t)$ is selected as $\tilde{x}^{[m]}(t)$. This process is repeated M times.

As with many genetic algorithms, a mutation process must be included when evolving one population to another. This is true here as well and the mutation operation is represented by simply adding noise to each sample $\tilde{x}^{[m]}$. Feasibility requires that each particle satisfy $x^{[m]}(t) \in X \; \forall t$. Care must be taken so that the noise added does not "push the particle out of X". The noise must therefore be in the interval

$$n^{[m]}(t) \in [l - \tilde{x}^{[m]}(t), u - \tilde{x}^{[m]}(t)] \quad \text{for } m = 1, \ldots, M \tag{2.17}$$

Finally, the new particle set is determined by

$$x^{[m]}(t+1) = \tilde{x}^{[m]}(t) + n^{[m]}(t) \quad \text{for } m = 1, \ldots, M \tag{2.18}$$

This formulation guarantees that each particle $x^{[m]}(t) \in X \; \forall \; t$ (each particle represents a feasible solution to $(\wp)$). It has the feature that as the particle set evolves from generation to generation, the particles with a higher weight (i.e. lower objective function value) are more likely to continue on to the next population.

2.4.4 Stopping Criterion

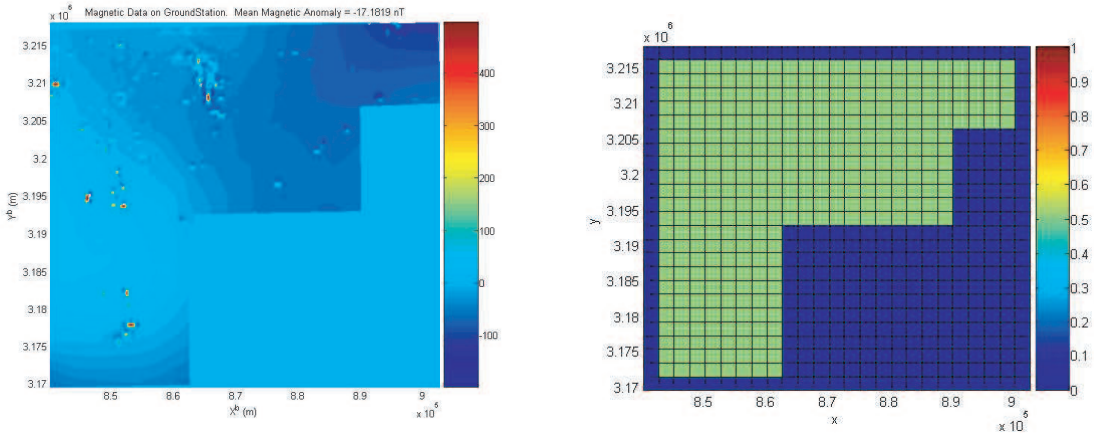
Step 2 and 3 are repeated until some stopping criterion is reached. One possible criterion is to stop when the variance of the particles is reduced below some threshold. Another simple method is to simply repeat for T steps. The quasi-optimal minimizer is computed from the average of the final particle set.

$$\bar{x}^{\star} = \frac{1}{M} \sum_{m=1}^{M} x^{[m]}(T) \tag{2.19}$$

2.5 Occupancy Map Based Searching

In order to effectively search a two dimensional domain for a target, the system must keep track of the state of the world in terms of possible target locations. To do this, an occupancy based map is employed. In this scheme, the search domain is discretized into rectangular cells. Each cell is assigned a score based on the probability that the target is located in that grid. This is similar to a two-dimensional, discretized probability density function [3]. This centralized occupancy based map is shared and updated by all agents involved in the search. At each time step, guidance decisions for each agent are chosen based on this map. An example is shown below in Figure 2.9.

In Figure 2.9(b), the blue sections represent areas with zero scores whereas the green represents scores of 1/2. This is the initial state of the occupancy based map. It represents having no a priori knowledge of the targets location, other than it cannot be in a section where no real data exists (i.e. the sections of uniform blue in Figure 2.9(a)).



(a) Total magnetic intensity map of region

(b) Initial Occupancy Based Map

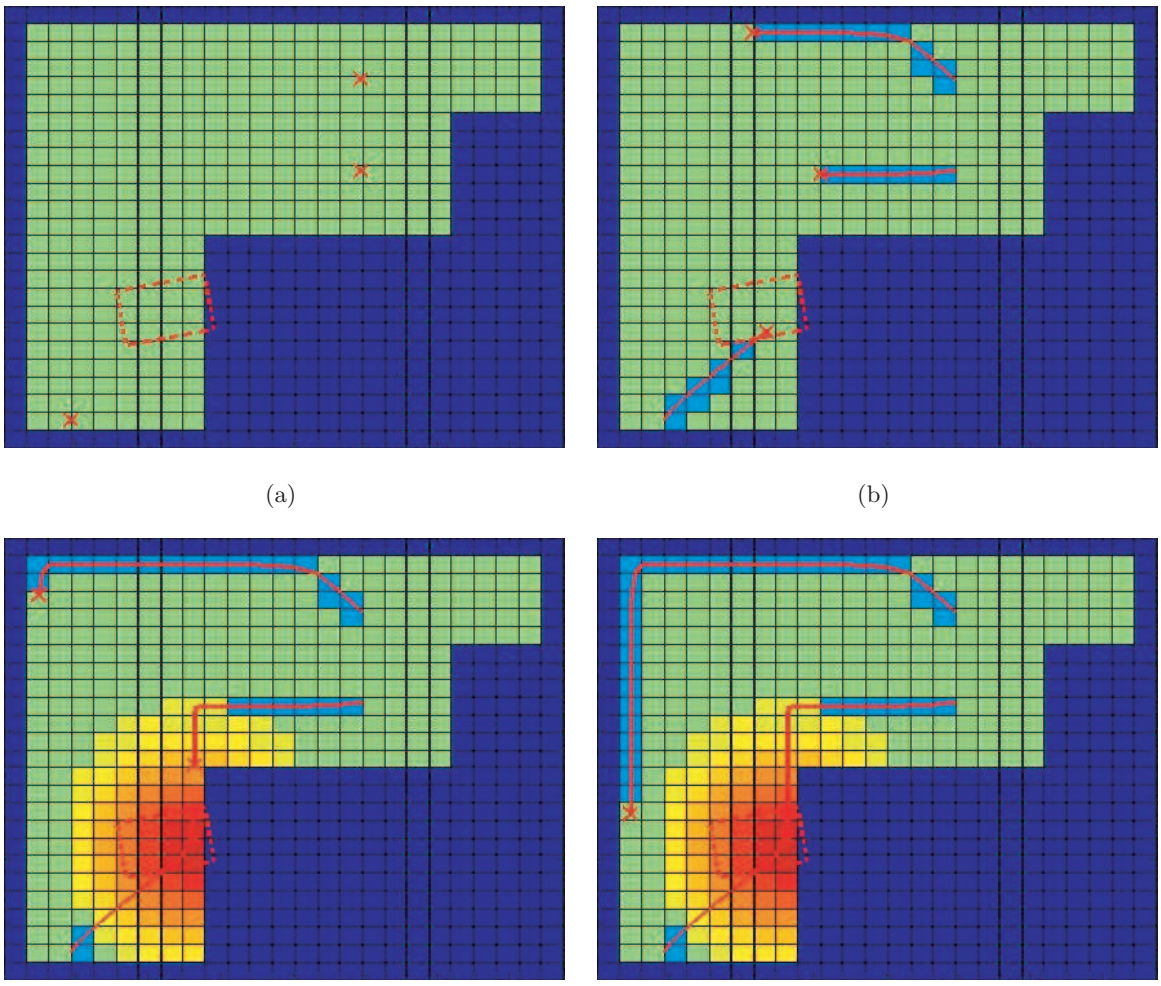
Figure 2.9: Discretization of search region into an occupancy based map.

There are several advantages to representing the target location belief in this fashion. One advantage is that it is simple to add time varying behavior into this map. This is addressed in Section 3.2.

2.6 Greedy Occupancy Map Based Search

Simulation using this greedy algorithm ($d = 1$) and a planar kinematic plant model where the agents only look 1 step ahead is shown below in Figure 2.10.

In this situation, the target location is shown as the dashed, red box and the agents are represented by red x's. Figure 2.10(a) shows the initial location of the agents relative to the target. In Figure 2.10(b), one of the agents is about to encounter an anomaly. In this case, the anomaly happens to be the target and therefore the particle filter is able to identify and classify this anomaly as the target and the agent makes a positive ID at its current cell. This updates the nearby occupancy map cells with increased scores. This causes the second agent to converge and investigate this location as shown in Figure 2.10(c). However, the third agent continues searching the other regions of the map as seen in Figure 2.10(d). This can be later compared to the system using the optimizing algorithm with the predictive world model.



(a) (b)

(c) (d)

Figure 2.10: Occupancy based map search with three agents.

2.7 Single Agent Search Strategy

This section does not focus on how the occupancy based map is updated but instead concentrates on how to find a optimal cell to search using an occupancy based map. The overall goal for the agent is to attempt to converge on regions of high scores (a high probability that the target is located there). Of course, the problem of finding an (x, y) coordinate which maximizes $x_w(t)$ may not be simple. From an optimization standpoint, $x_w(t)$ is in general non-convex, discontinuous, and has gradient equal to either zero or infinity. We now

propose a method which yields a quasi-optimal solution which is feasible and is formulated as a convex optimization problem.

The overall flow of the system is shown below in Figure 2.11. The process is comprised of three main problems which are referred to as ($\wp_1$), ($\wp_2$), and ($\wp_3$).

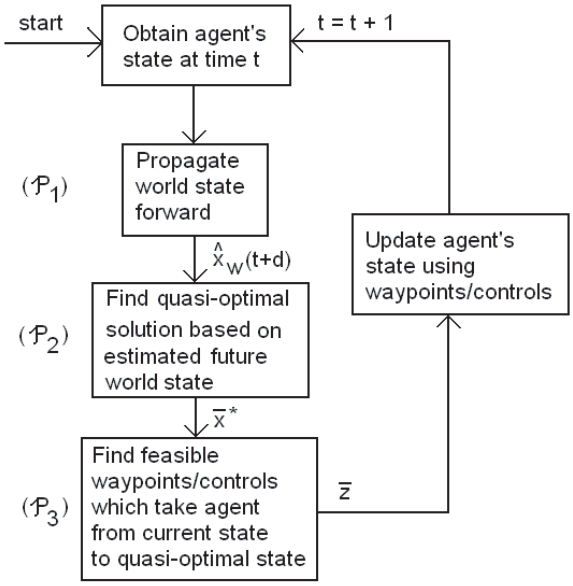


Figure 2.11: Flow diagram for optimization process.

The process starts by finding the agent's state at the current time, $z_{uav}(t)$. Next, ($\wp_1$) is solved to obtain the estimated world state at time $t + d$. Next, ($\wp_2$) is solved to find the coordinates of the cell with the maximum score in the reachable cells (these are the cells that the agent can reach in d steps). Once this quasi-optimal solution, $\overline{z}^*$, is found, ($\wp_3$) consists of finding an optimal set of waypoints/controls, $\overline{z}$, which will take the agent from its current state to the quasi-optimal state. This is formulated as a convex optimization problem.

2.7.1 ($\wp_1$) *Predictive World Model*

The first problem ($\wp_1$) involves creating an estimate of the world state at a given time and then projecting this estimate forward in time to obtain the estimated state of the world at

time $t + d$. The block diagram for the world estimator is shown below in Figure 2.12.

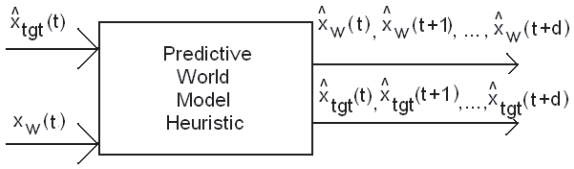


Figure 2.12: Block diagram for world estimator.

The inputs are the estimated state of the target (position and velocity) at the current time, $\hat{x}(t)$ and the current state of the occupancy based map, $x_w(t)$. In order to propagate an estimate of the target state, we assume simple dynamics of the form

$$\hat{x}_{tgt}(t+1) = A_{tgt}\hat{x}_{tgt}(t) + B_{tgt}\hat{u}_{tgt}(t) \tag{2.20}$$

The world estimate at time $t+p$ is then a function of the estimated target state at time $t + p$ and the world state at the original time t.

$$\hat{x}_w(t+p) = h(\hat{x}_{tgt}(t+p), x_w(t)) \text{ for } p = 0, ..., d \tag{2.21}$$

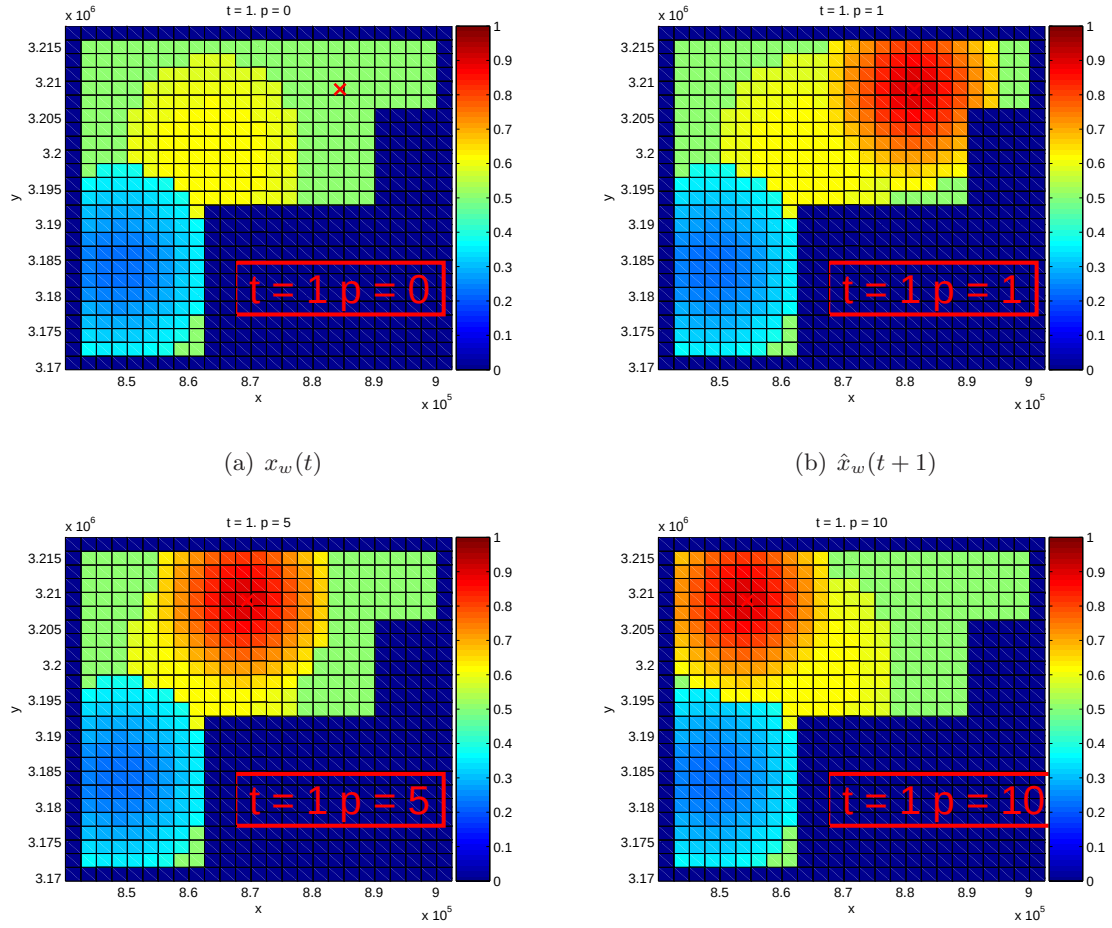
In our example, the function $h()$ simply adds a two-dimensional Gaussian centered about $\hat{x}_{tgt}(t+d)$ to $x_w(t)$. An example of this is shown below when the estimated target state is observed to be moving to the left at a constant velocity.

Now that the state of the world can be estimated at time $t + d$, we can attempt to find the coordinates of a cell with a high score in the reachable cells, $\bar{z}^\star$. This is addressed in $(\wp_2)$.

2.7.2 $(\wp_2)$ *Find Feasible Area of High Score*

Each agent can only reach certain cells in d time steps. These cells are known as the "reachable cells" for each agent.

We can apply the above method of finding a quasi-optimal solution to $(\wp_2)$. We define this problem as



(a) $x_w(t)$ (b) $\hat{x}_w(t+1)$

(c) $\hat{x}_w(t+5)$ (d) $\hat{x}_w(t+10)$

Figure 2.13: Estimated world states at different times for estimated target moving to the left.

$$(\wp_2) \underset{z \in Z}{\text{minimize}} \; f_0(z) \tag{2.22}$$

Parameterizing X

Here, the set Z is comprised of all the locations the agent is able to reach in d steps (reachable states).

$$Z = \left\{ z \,\middle|\, z = z_{uav} + r \begin{pmatrix} cos(\pi/2 - \psi) \\ sin(\pi/2 - \psi) \end{pmatrix}, \begin{matrix} r \in I_r \\ \psi \in I_\psi \end{matrix} \right\} \tag{2.23}$$

In Eq. 2.23, the intervals describe the maximum radius and heading angle that the agent can achieve. Since we assume a simple model, we have $I_r = [0, d \cdot \Delta T \cdot V_{max}]$ and $I_\psi = [0, 2\pi]$. An example of this set Z is shown below in Figure 2.14.

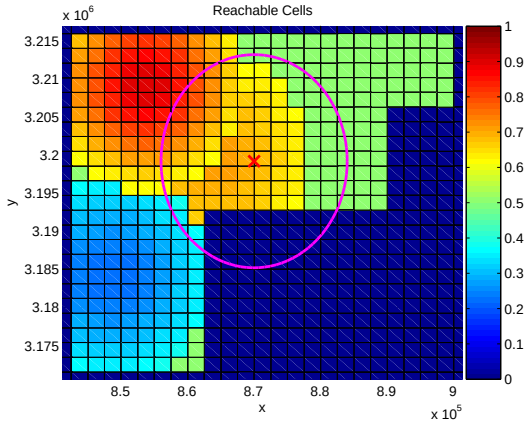


Figure 2.14: Reachable locations Z shown inside purple circle

Note that the set Z is not a box. So to utilize the method above, the reachable cells are parameterized using r and ψ, namely

$$x = \begin{pmatrix} r \\ \psi \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{2.24}$$

So the set X is simply

$$X = \left\{ x \,\middle|\, \begin{matrix} x_1 \in I_r = [0, d \cdot \Delta T \cdot V_{max}], \\ x_2 \in I_\psi = [0, 2\pi] \end{matrix} \right\} \tag{2.25}$$

Using Eq. 2.23, we see that the conversion between $x \in X$ and $z \in Z$ is simply

$$z = z_{uav} + x_1 \begin{pmatrix} cos(\pi/2 - x_2) \\ sin(\pi/2 - x_2) \end{pmatrix} \tag{2.26}$$

Define $f_0(x)$

The objective function $f_0(x)$ is now defined. The scores are a function of the position z. The maximum score is equivalent to the minimum of the negative scores. Furthermore, in the context of model predictive control, instead of minimizing over the current world state at time t, the objective function is minimized over the projected world state at time $t + d$. Eq. 2.26 can be used to convert between z and x, so the objective function can be defined as

$$f_0(x) := -\hat{x}_w(t + d) \tag{2.27}$$

Combining Eq. 2.25 and 2.27, it can be seen that $(\wp_2)$ is a specialized version of $(\wp)$. Therefore, the methods described above can be used to find a quasi-optimal minimizer $\bar{x}^\star$ which can be converted into $\bar{z}^\star$. A block diagram showing inputs and outputs for the probability collective minimization routine is shown below in Figure 2.15.

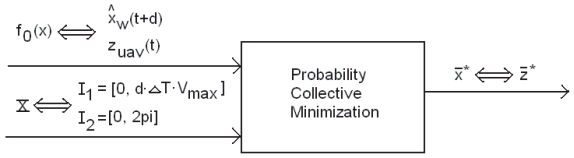


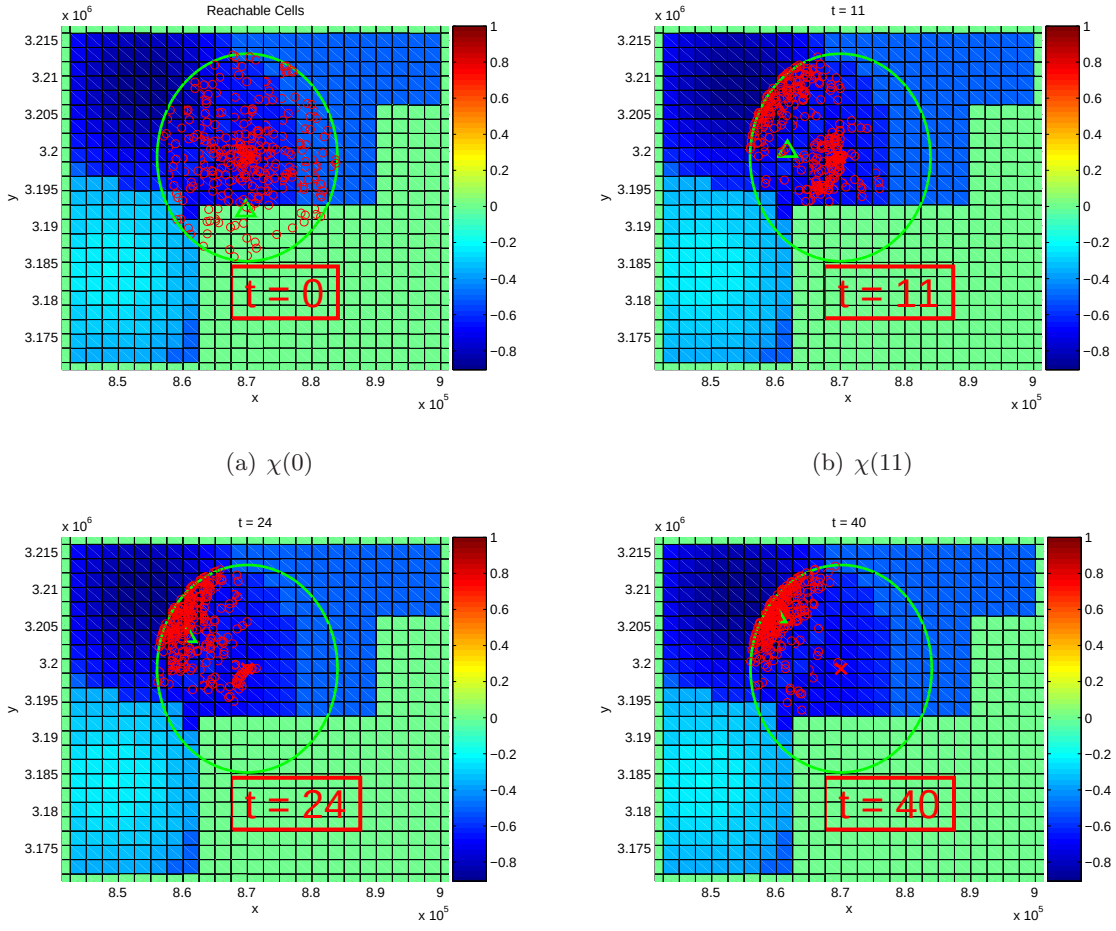
Figure 2.15: Block diagram for Probability Collective minimization.

An example of progression of this process is shown below in Figure 2.16.

The particles are shown as red circles. The centroid of the particles (green triangle) eventually centers near the true optimal solution. Note that it does not achieve the true optimal but it does achieve a feasible solution.

Computing Average (Accounting for Wrap Around)

In this context, Eq. 2.19 cannot be used to compute the centroid of the particles. This is because the second element of the state vector is an angle with the property that the angle wraps around from 0 to 2π. Eq. 2.19 simply computes the average of the particles. If



(a) $\chi(0)$

(b) $\chi(11)$

(c) $\chi(24)$

(d) $\chi(40)$

Figure 2.16: Progression of probability collective process. The true minimum is located in the upper left corner of the reachable cells.

there were only two particles and they had angles of 1 and 359 degrees, the correct optimal solution should be 0 degrees but Eq. 2.19 would return an optimal solution of 180 degrees. This wrap around is taken into account by instead transforming from r and ψ back to x and y and then applying Eq. 2.19. This yields the correct quasi-optimal solution.

2.7.3 ($\wp_3$) Convex Formulation

The final problem ($\wp_3$) is now addressed. This concerns finding feasible waypoints which take the agent from the current state $z_0 = z_{uav}$ to the quasi-optimal state found in ($\wp_2$). From an optimization standpoint, the corresponding decision vector $z \in \Re^{2 \cdot d}$ is

$$z = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ \vdots \\ z_{2 \cdot d-1} \\ z_{2 \cdot d} \end{pmatrix} = \begin{pmatrix} z_1(1) \\ z_2(1) \\ z_1(2) \\ z_2(2) \\ \vdots \\ z_1(d) \\ z_2(d) \end{pmatrix} \tag{2.28}$$

Using Eq. 2.26, the quasi-optimal solution $\bar{x}^\star$ obtained by the probability collectives method can be converted to an (x, y) coordinate $\bar{z}^\star$. The objective function for ($\wp_3$) can then be formulated as

$$f_0(z) := \sum_{t=1}^{d} ||z(t) - \bar{z}^\star||_2^2 \tag{2.29}$$

Here, each coordinate $z(k)$ is a waypoint that dictates where the agent should be located at time k. Obviously, the minimum of this function is zero. This corresponds to all the waypoints being placed at the quasi-optimal solution $\bar{z}^\star$. However this would not be feasible in general. The agent can only travel a distance $r_{max} = \Delta T \cdot V_{max}$ in a single step. So in order for the waypoints to be feasible, constraints are introduced of the form

$$f_1(z) := ||z(1) - z_0||_2^2 - r_{max}^2 \leq 0 \tag{2.30}$$

$$f_i(z) := ||z(i) - z(i-1)||_2^2 - r_{max}^2 \leq 0 \text{ for } i = 2, \ldots, d \tag{2.31}$$

Physically, these constraints enforce that each waypoint must be within a distance r_{max} of the previous waypoint. This ensures that the waypoints generated are feasible ones.

The problem can now be formally stated as

$$(\wp_3) \text{ minimize } f_0(z) \text{ over } z \in \Re^{2 \cdot d} \qquad (2.32)$$
$$\text{subject to} f_i(z) \leq 0 \text{ for } i = 1, \ldots, d$$

In order to analyze what type of optimization problem this is, a closer look at the objective function and constraint functions is required. The objective function can be written as

$$f_0(z) = \frac{1}{2} z^T H z + f^T z + r \qquad (2.33)$$

$$\begin{array}{rrcl} \text{where} & H & = & 2I_{d \times d} \\ & f^T & = & 2\left(\bar{z}_1^\star \quad \bar{z}_2^\star \quad \bar{z}_1^\star \quad \bar{z}_2^\star \quad \ldots\right) \\ & r & = & d||\bar{z}^\star||_2^2 \end{array}$$

This is a strictly convex function since it is in a quadratic form and the Hessian is equal to H which is positive definite (all eigenvalues are equal to 2).

The constraint functions can be analyzed in a similar fashion. The first constraint $f_1(z)$ can be written as

$$f_1(z) = \frac{1}{2} z^T H_1 z + f_1^T z + r_1 \qquad (2.34)$$

$$\begin{array}{rrcl} \text{where} & H_1 & = & \text{diag}(2I_{2\times 2}, 0_{d-2\times d-2}) \\ & f_1^T & = & \left(-2z_{1,0} \quad -2z_{2,0} \quad 0 \quad \ldots \quad 0\right) \\ & r_1 & = & -r_{max}^2 + ||z_0||_2^2 \end{array}$$

And the constraint functions $f_2(z)$ through $f_d(z)$ can be written as

$$f_i(z) = \frac{1}{2} z^T H_i z + f_i^T z + r_i \text{ for } i = 2, \ldots, d \qquad (2.35)$$

$$
\begin{aligned}
\text{where} \quad H_i &= \text{diag}(N_i, A, M_i) \\
f_i^T &= \begin{pmatrix} 0 & \dots & 0 \end{pmatrix} \\
r_i &= -r_{max}^2 \\
N_i &= \text{zeros}(2(i-2)) \\
M_i &= \text{zeros}(2d - 4 - 2(i-2)) \\
A &= \begin{pmatrix} 2 & 0 & -2 & 0 \\ 0 & 2 & 0 & -2 \\ -2 & 0 & 2 & 0 \\ 0 & -2 & 0 & 2 \end{pmatrix}
\end{aligned}
$$

In Eq. 2.34 and 2.35, $\text{diag}(x, y)$ represents a block diagonal matrix with submatrix x in the upper left block and submatrix y in the lower right corner. Similarly, $\text{zeros}(p)$ represents a square zero matrix of size $p \times p$. Although this appears to be a complicated formulation, note that only H_i is a function of i. Therefore H_2 is a block diagonal matrix with A in the upper left corner and zeros elsewhere. H_3 is a block diagonal matrix where the submatrix A moves two columns to the right and two rows down. This process of moving the A matrix by 2 rows and columns with each i is described by the N_i and M_i submatrices.

One can now see that the constraint functions $f_i(z)$ for $i = 1, \dots, d$ are convex functions because they are quadratic forms and their respective Hessians are all positive semi-definite.

Therefore, $(\wp_3)$ consists of a strictly convex objective function over a convex set. The resulting optimization problem is a convex programming problem. It can be showed that this problem is well posed and the feasible set is non-empty, so a unique optimal solution exists. In a similar fashion to the previous two problems, $(\wp_3)$ can be packaged nicely into a system shown in Figure 2.17.

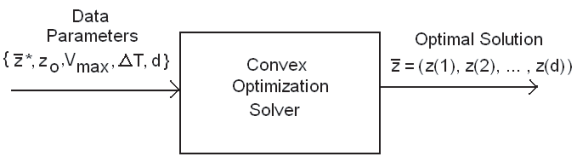


Figure 2.17: Block diagram for convex optimization solver.

An example of the solution for the situation with $d = 10$ is shown below in Figure 2.18.

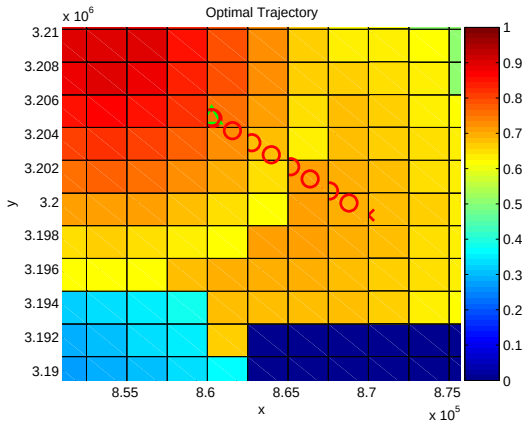


Figure 2.18: Optimal solution $\bar{z}$ to $(\wp_3)$ zoomed into area of interest with $d = 10$.

In Figure 2.18, the red x represents the location of the agent, the green triangle is the desired destination $\bar{z}^\star$, and the red circles represent waypoints $\bar{z}$. Notice that although $d = 10$, there are only 8 visible waypoints. This is because waypoints 8, 9, and 10 are overlapping and all are equal to $\bar{z}^\star$. Furthermore, constraints $f_1(z)$ through $f_7(z)$ are active. This shows that the formulation of the objective function yields waypoints which place the agent at the optimal solution in the shortest possible time. This is the desired behavior during a searching type application where it is desired that the agent locate and verify the target as soon as possible.

2.8 Publications

Much of this work has been converted into conference papers and presentations. Some of these include

- Lum, C.W., Rysdyk, R.T., and Pongpunwattana, A., "Autonomous Airborne Geomagnetic Surveying and Target Identification," *Proceedings of the 2005 Infotech@Aerospace Conference*, Arlington, VA, September 2005

- Lum, C.W., "A Single Agent Search of a Two Dimensional Space Using Probability Collectives and Convex Optimization", Tech. rep., Autonomous Flight Systems Laboratory, Seattle, WA 2005

- Lum, C.W., Rysdyk, R.T., and Pongpunwattana, A., "Occupancy Based Map Searches Using Heterogeneous Teams," *To be presented at the 2006 AIAA Guidance, Navigation, and Control Conference*, Keystone, CO, August 2006

Chapter 3

PROPOSED RESEARCH

3.1 Submarine Detection and Searching

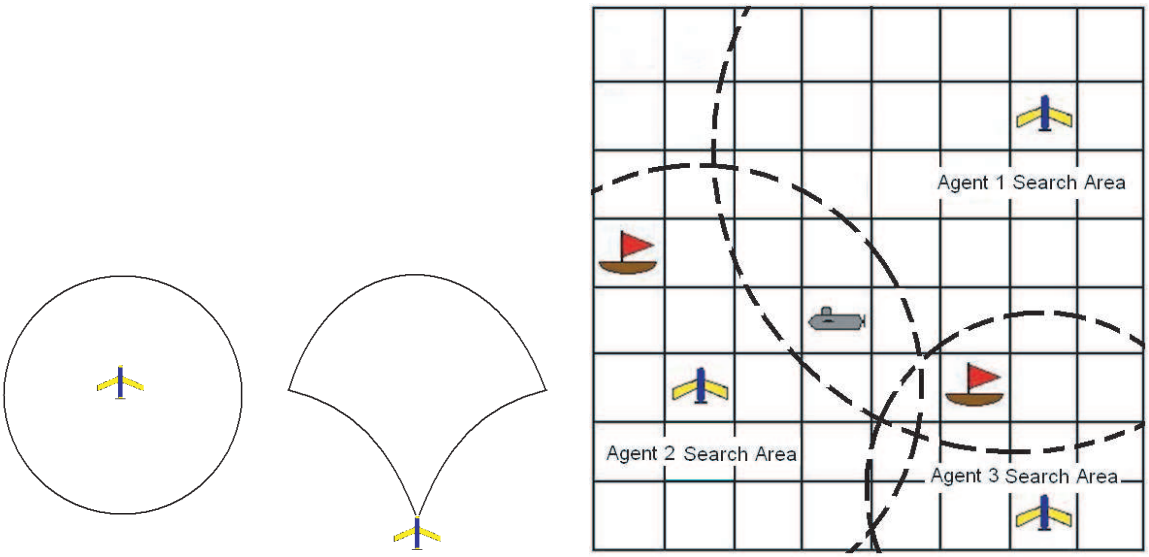
The main focus of this work is to develop efficient methods to search for and locate a target. Currently there are several improvements to the current scheme that are being considered.

3.1.1 Modifying Reachable Cells

Different agents might have different capabilities and are able to reach different regions at time $k + d$. This set of reachable cells might be based on the agent's capabilities such as saturation limits or environmental variables such as wind. For example, an agent with the ability to instantaneously change heading in no wind would have a reachable set of cells as a circle with radius $r = d * \Delta t * v$. However, an agent which has a severely limited turn rate would have more of a flower-shaped reachable set of cells. This is shown below in Figure 3.1(a). The reachable cells become patrol regions of sorts for each agent. These patrol regions could still overlap and would dictate which possible trajectories are feasible [7]. A set of overlapping reachable sets is illustrated below in Figure 3.1(b).

3.1.2 Benefits of Using a Finite Horizon Window

There are several situations where this type of predictive world model becomes useful. One example is in the situation of a moving target. Without the predictive world model, the agent will simply try to "catch" the target by trying to converge on its present location. This leads to a slow convergence. However, with the predictive world estimate, it would be possible to converge on likely future state of the target. This idea is illustrated below in Figure 3.2(a). In this situation, the curved trajectory represents what might happen if the agent did not have the capability to predict where the target would be at time $k + d$. It



(a) Different feasible set for different agents (b) Multi agent team with individual patrol areas

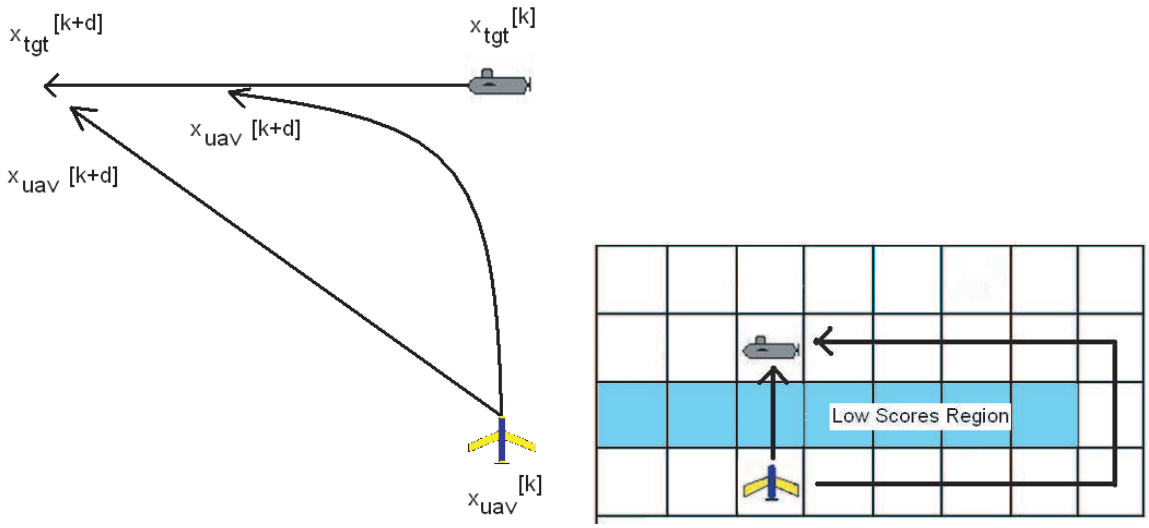
Figure 3.1: Reachable cells and patrol regions.

would simply try to do what was best for itself only 1 step ahead. However, the straight line represents what might occur if the optimization routine was able to access the predictive world model. The agent would then be planning trajectories which would benefit it d steps in the future rather than just focusing on the immediate future.

Another situation where this benefits the agent is if the target is hidden behind a "wall" of low scores as shown below in Figure 3.2(b). In this situation, with the ability to look ahead more than 1 step, the agent would choose a long and indirect trajectory to the target. But with the predictive world model, it would be possible for the agent to plan d steps ahead and then swiftly converge to the target.

3.1.3 Multi-Agent Search

All of these ideas presented previously would be scalable to teams of agents. Each agent would follow control laws previously described in Section 2.7. In this formulation, each agent does not have explicit knowledge of the other agents in the team but there is still coupling between the agents. The coordination is implicitly built into the algorithms and



(a) Agent converging to target

(b) Target hidden behind wall of low scores

Figure 3.2: Potential situations with and without predictive world estimate.

all agents are weakly coupled through the centralized occupancy map. More explicit team cooperation is discussed in Section 3.5.

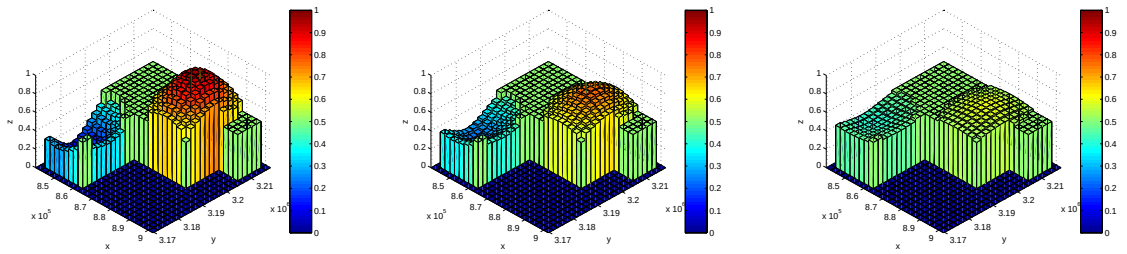
3.2 Time Varying World Model

The system maintains a current estimate of the world using the sensor measurements from each agent at each time step k. Since a connected communication topology is assumed, a group consensus concerning the current state of the world can be reached and the centralized occupancy based map can be updated. The scores are allowed to be time varying to model the fact that the target location estimate becomes more uncertain as time progresses. This is modeled as a simple linear dynamic model of the form

$$B_{ij}(k+1) = \tau B_{ij}(k) + B_{nom} - \tau B_{nom} \tag{3.1}$$

B_{nom} is the nominal score and τ is the time constant governing how fast the score decays back to the nominal. This effect is shown below in Figure 3.3. There is no estimate of target velocity so the regions of high probability do not translate in the x and y directions.

However, as time progresses, the estimates return to their nominal values to model the phenomenon that old measurements cannot be relied upon to judge if the target is located in a certain cell or not.



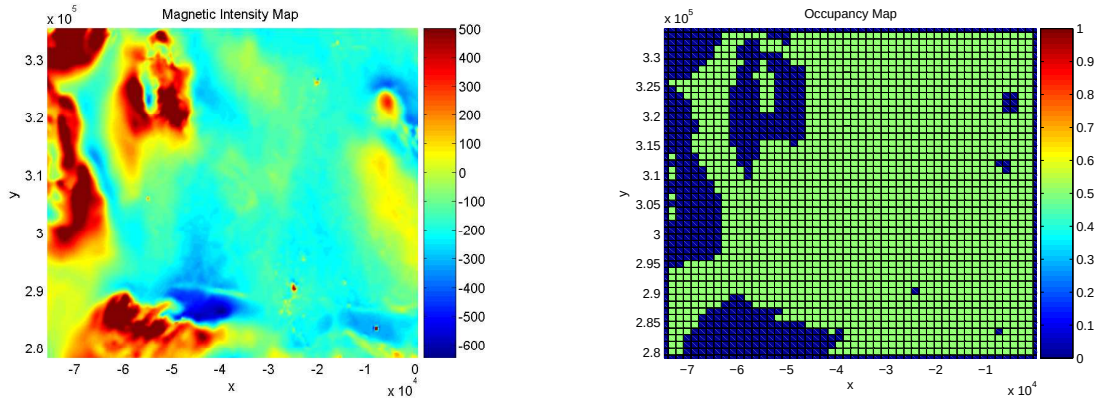
(a) $B(k)$ (b) $B(k+1)$ (c) $B(k+2)$

Figure 3.3: World estimate propagation.

3.3 Autonomous Geomagnetic Surveying and Mapping

As shown previously, the need to acquire the TMI map of an area is crucial to the execution of an accurate search-and-detect mission. The survey flown to obtain Figure 2.4(b) was flown at an altitude of 800 ft and along parallel lines spaced 0.25 miles apart [1]. The total distance flown was over 43,000 miles. Assuming the aircraft cruises at 100 MPH, obtaining this survey requires over 430 hours (almost 3 weeks) of continuous flight time to perform. Optimally, surveys which are used for target identification purposes would be flown much closer to the surface of the water (less than 75 ft). This task of long endurance, low altitude flight over water environments is a dull and dangerous task best suited for autonomous systems.

In the application of searching for a target in a marine environment, flights over land are not required and the survey time can be drastically reduced if the agents conducting the survey can be controlled in a fashion which directs them only over marine areas. The use of the occupancy map can be exploited in order to achieve this goal. The occupancy map can have scores set to zero over regions which are over land. This is shown below in Figure 3.4.



(a) Geographical map over survey area

(b) Occupancy map over survey area

Figure 3.4: Occupancy and Geographical map over survey areas with non-zero scores only over marine sections.

The control laws which guide a team of autonomous survey vehicles can use this occupancy map in order to guide the team to only survey the areas which have non-zero scores (i.e. over marine environments). The benefits of this become more evident over areas which have large sections of land and water mixed together. Areas such as ports and inlets are areas which can benefit significantly from this modified autonomous survey.

3.4 Realistic Magnetic Models

As stated in Section 2.2, accurate magnetic maps and target signatures are necessary for a successful target identification process. Based on our limited knowledge at the current time, the submarine signature is modeled as an oblong, two-dimensional Gaussian with peak of approximately 115nT. The function's level sets are roughly ellipsoids. The rate of decay from the peak can be tailored using the covariance matrix. Undoubtedly, this profile differs from a true submarine signature. Furthermore, this profile is a function of depth and sensor altitude. An accurate magnetic signature of the target is required in order to predict the target signature in different conditions. Some modeling has been done by engineers at Fugro Airborne surveys. One of the models is shown in Figure 3.5.

If modeling the complex target signature is not accurate enough or impractical, another

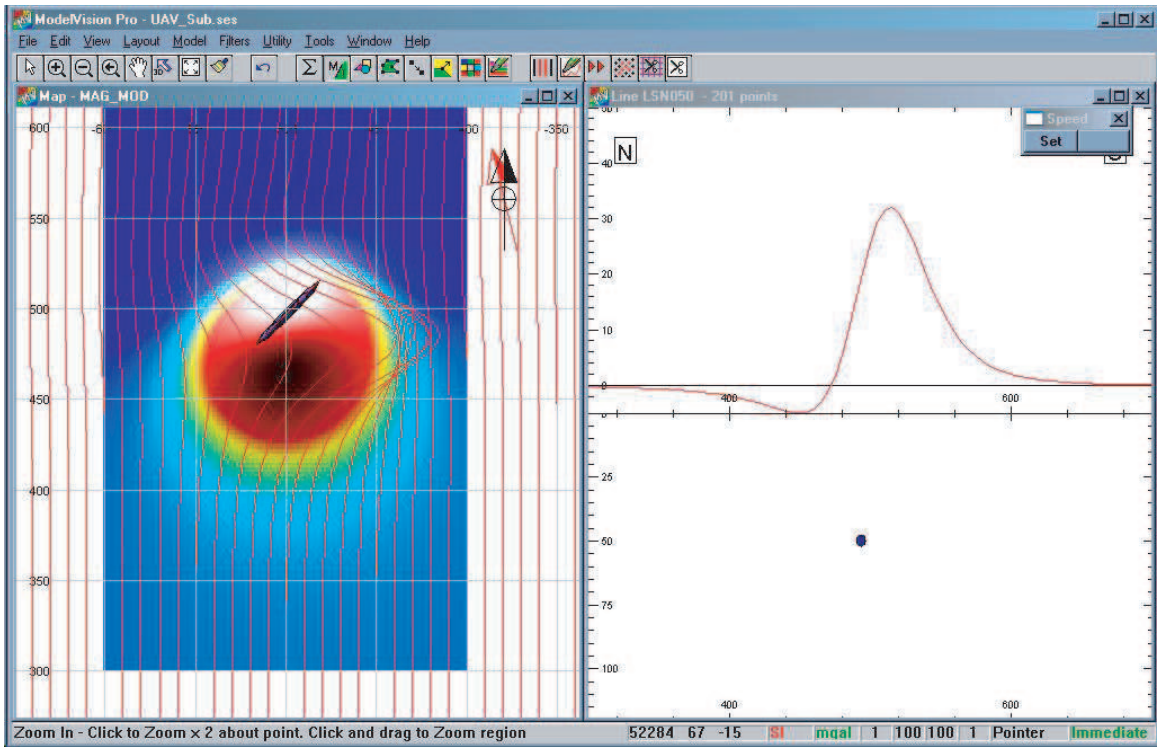


Figure 3.5: Submarine signature modeled with ModelVision Pro.

option is to obtain this data experimentally. This would involve first obtaining a TMI map of the test area without the target and then mapping the same area again with the target present in a known position and orientation. By Eq. 2.1, the difference of the two maps should be the magnetic signature of the target.

If obtaining this model or data is not practical, other options for demonstrating the target identification algorithms is discussed in section 3.7.2.

3.5 Multi-Agent Emergent Behavior

In the current setup, the agents are somewhat greedy; each agent executes its control laws and performs actions while oblivious to the existence of other agents in the environment. They perform actions which optimize their own objective function, but these actions may not be beneficial to the overall mission. The agents are weakly coupled because they all

share a centralized occupancy map. A more sophisticated and effective system would be a move towards a decentralized control scheme where there is explicit cooperation between agents.

The overall goal would be to have each agent solve their own simpler problems but have the overall system display more complex behavior. The agents would be more strongly coupled either through a centralized coordinator or through peer-to-peer communication (some communications issues are addressed in Section 3.6). This leads into the subjects of emergent behavior, decentralized control, and cooperative games.

3.5.1 Emergent Behavior

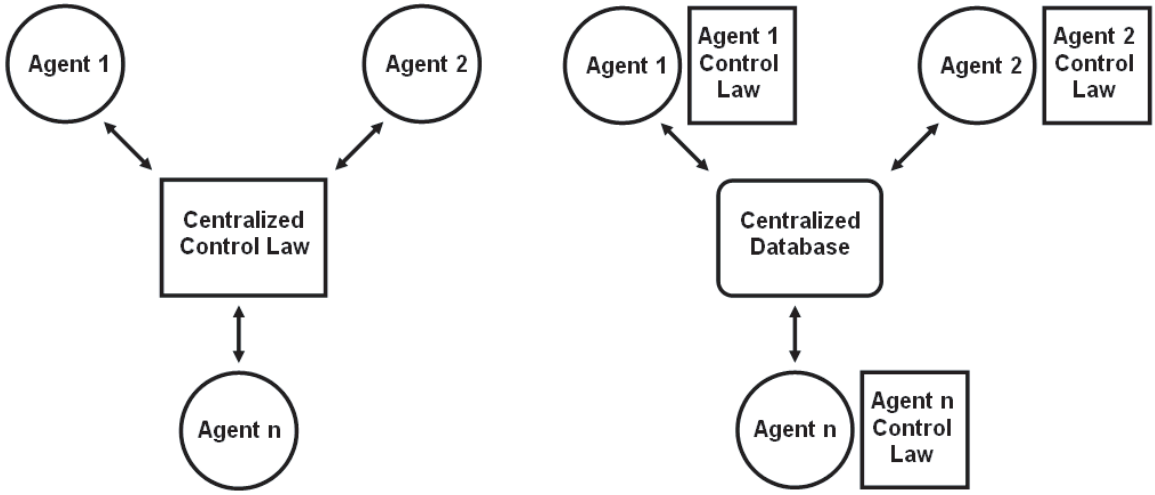
Systems displaying emergent behavior properties are ones where each individual in the system follows a simple set of control laws. Although the individual behavior of each agent is simple, the overall system may able to achieve more complicated goals. An advantage of this approach is that the individual control laws are usually somewhat simple and easy to analyze. The main drawback is that it is difficult to predict the behavior of the overall system.

Often, the agents are not coupled or weakly coupled with limited communication. The current setup can be classified as an emergent behavior algorithm since there is no explicit communication between the agents and the behavior of searching an area is achieve implicitly through the occupancy map rather than explicitly through cooperation or control laws.

3.5.2 Decentralized Control

In systems utilizing many agents, algorithms which govern the teams actions can be classified as either centralized or decentralized control schemes. A centralized scheme refers to an architecture where all agents are connected to a single node. This node may be a ground station or another agent. Agents may transmit information about their own state, but this centralized node does the computation of the actual control laws and dictates the actions to be taken by the other agents. A decentralized architecture implies that each agent computes its own control laws and actions. These agents may be connected to a centralized node or

to each other, but this is merely to obtain information which may increase performance. Each agent in a decentralized scheme is able to independently decide on its own actions regardless of the communication topology. A general architecture showing the difference between the centralized and decentralized control schemes is shown in Figure 3.6.



(a) Centralized architecture

(b) Decentralized architecture

Figure 3.6: Centralized and decentralized architecture.

Teams which rely on centralized control schemes enjoy many benefits such as increased performance and efficiency between its agents. However, these schemes are generally computationally intensive and not scalable. In order to address these shortcomings, decentralized control schemes are used. Although these types of algorithms are generally more complicated and may not produce globally optimal solutions, they have several notable advantages that make them more attractive to a team of autonomous systems. One of these advantages is that a team operating under a decentralized control scheme can be made robust to single points of failure in the system. In other words, it is possible for agents in the team to be removed from the mission and the team can adapt and continue [12]. Another main advantage is that these types of algorithms are scalable to larger systems.

In a searching mission over a large area, the communication topology of the system may change continuously (Section 3.6). Agents are able to communicate only with other nearby

agents. In general, this communication topography is not strongly connected (i.e. every agent is not connected to every other agent). Decentralized control algorithms have groups of connected agents working together to achieve a team specific mission. These groups are dynamic and vary as the communication topology changes. One decentralized method to coordinate multiple agents is a market based protocol which elects a coordinator agent to direct the actions of the other agents in the team [13]. Each group has a single coordinator agent and one or more vehicle agents. This formulation allows for groups to be formed depending on the current communication topology. An example of this is shown below in Figure 3.7.

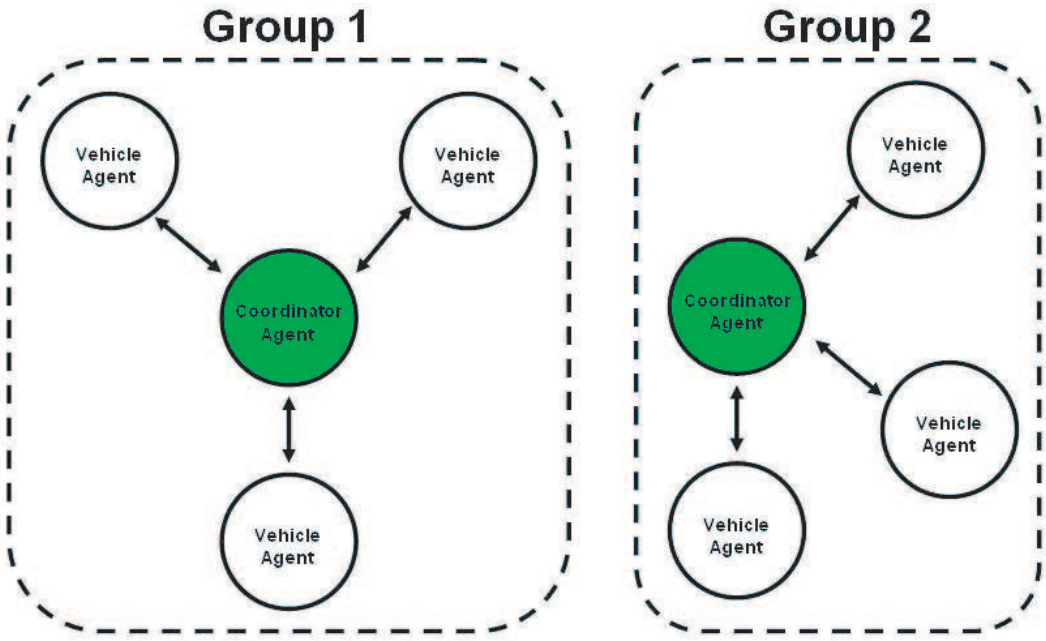


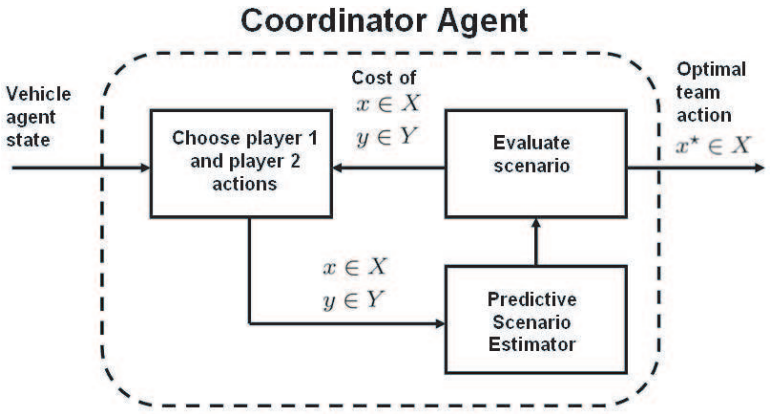
Figure 3.7: An example of forming two groups based on communication topology.

3.5.3 Cooperative Games

In addition to market based protocol, there are other methods to control groups of agents. One such method would be to formulate the problem as a special type of optimization problem. In this scenario, the team of agents could be seen as one player whereas the

environment could be seen as another player. The two players are playing a type of game against each other. This is the setting for a classic two player game. There has been significant research and developments in the field of game theory, particularly with two player games [15]. The problem consists of the team of agents selecting an action $x \in X$ which benefits itself, whereas the environment would select actions $y \in Y$ which would try to hinder the team. This is often analyzed as a type of worst-case scenario using minimax theory.

Evaluating the cost of choosing $x \in X$ and $y \in Y$ is not an easy task. There are complex predictive models that must be used in order to evaluate the consequences of these choices. The value of choosing $y_i \in Y$ and $x_j \in X$ make up the ij^{th} entry of a payoff matrix for this system. A possible architecture for formulating the payoff matrix and an example payoff matrix are shown below in Figure 3.8.



(a) Coordinator agent architecture

	x_1	x_2	x_3	x_4
y_1	3	2	-1	2
y_2	-5	-2	-1	-9
y_3	-4	2	-7	8

(b) Payoff matrix

Figure 3.8: Architecture for coordinator agent used to generate payoff matrix.

3.6 Communication Issues

In a system with heterogeneous agents spread out over large distances, communication becomes an issue of concern. For example, suppose an agent makes an observation of a target and wishes to relay that information back to the base of operations as shown below

in Figure 3.9.

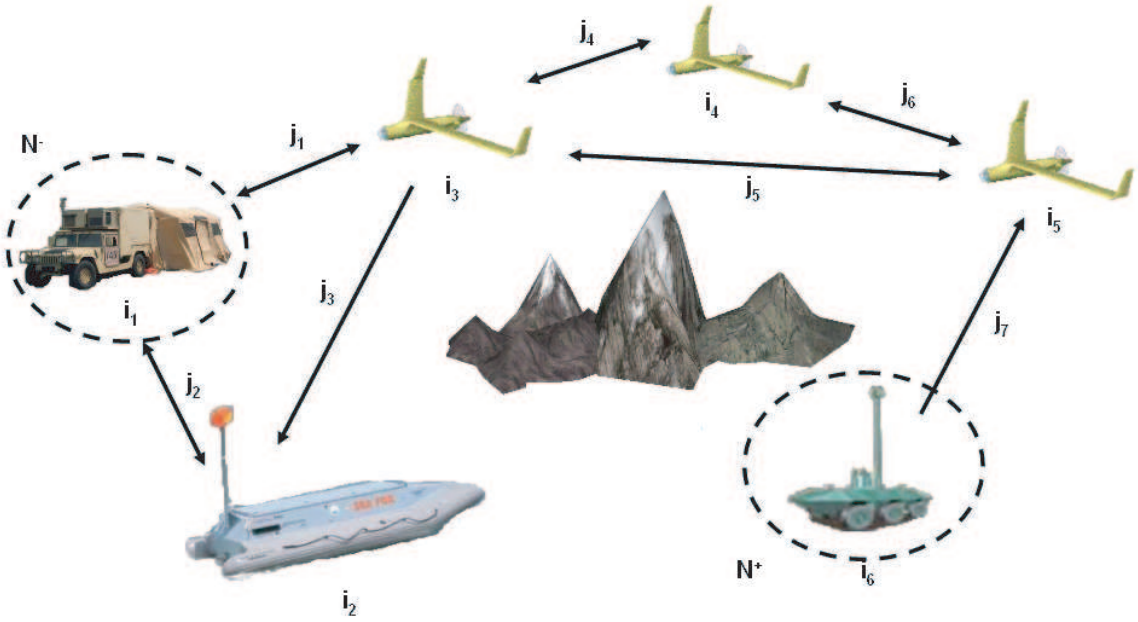


Figure 3.9: A potential situation showing different agents forming a directed graph.

If the agent is far away from the base, direct transmission of data may not be possible. A feasible option would be to transmit data instead to another agent located closer to base and therefore have a better chance of successfully transmitting the data. If approached from a combinatorial standpoint, the problem becomes an NP-hard problem which becomes very difficult to solve in real time with a large number of agents. To exacerbate matters, the network changes continuously as some agents move out of range of others. As more agents become involved in this relaying process, a method of routing the data efficiently becomes necessary.

One such formulation would be to view the heterogeneous team of agents as a directed graph, or network. Each agent would be represented by a node $i \in N$, and each connection between agents would be represented by an arc $j \in A$. Formulating this problem as a network optimization problem allows many tools to be applied. For example, the well known min-path algorithm can be applied [14]. This requires the specification of the network structure which can be formulated as specified above. It also requires starting nodes, N^+

and ending nodes, N^-.

The length of an arc, $l(j)$, can be formulated to be inversely proportional to the probability of a successful packet transmission over the arc, p_k.

$$l(j_k) = 1/p_k \tag{3.2}$$

The min path algorithm solves

$$P^* = \arg\min_P \sum_{j \in P} l(j) \tag{3.3}$$

This formulation yields a path which has the shortest path and also the highest probability that the packet will be successfully routed from N^+ to N^-.

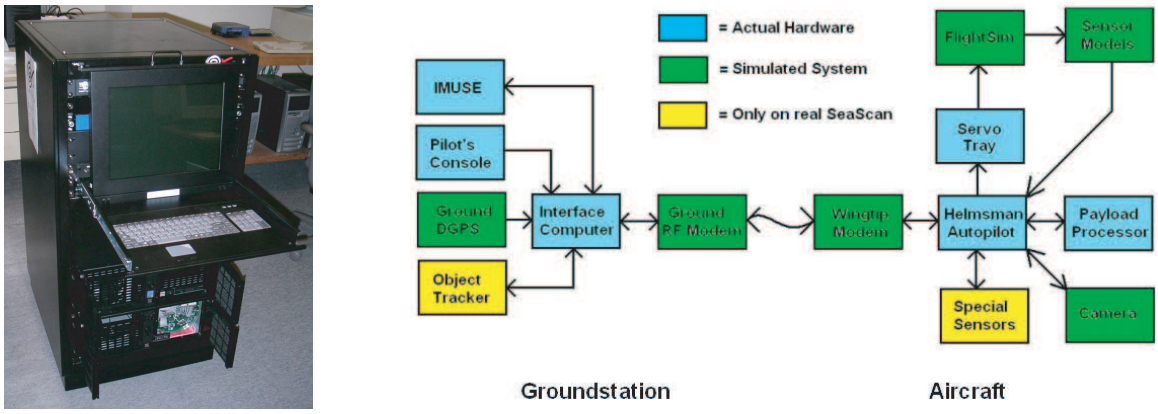
Eq. 3.2 can be modified in order to influence the routing if other goals are desired. This can be thought of as a tuning parameter in a cost function in an attempt to influence the solution P^*.

3.7 Implementation and Flight Tests

3.7.1 Hardware-in-the-Loop Simulation

The Insitu group has provided a high-fidelity Hardware-in-the-Loop (HiL) simulator. This simulator serves to bridge the gap between pure computer simulation and an actual flight test.

The HiL includes both a simulated and actual subsystem for an actual GeoRanger UAV as show in Figure 3.10(b). In addition, the HiL includes a payload processor expansion slot where an embedded processor can be attached to interface with UAV avionics and sensors. Previously developed searching and target identification algorithms can be loaded onto this embedded processor which will interface with the UAV avionics and sensors in order to provide high level guidance commands to the UAV's autopilot. This entire system will simulate an actual flight test of a single vehicle with a high degree of accuracy.



(a) HiL simulator

(b) Simulated and actual subsystems

Figure 3.10: Insitu provided HiL at the University of Washington AFSL.

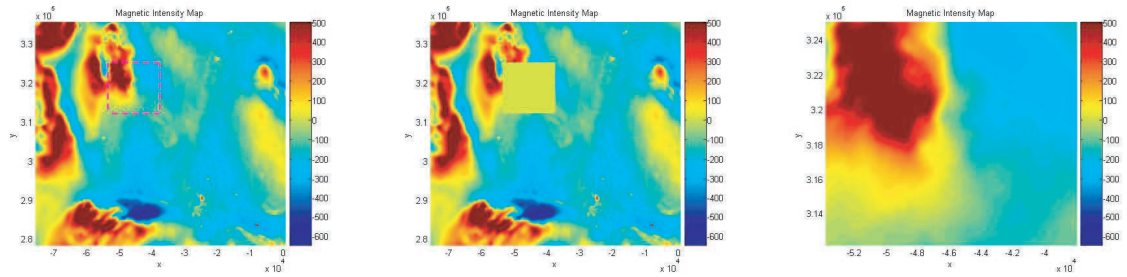
3.7.2 Flight Test

Once a successful HiL simulation is performed, the next step is to validate these algorithms using a flight test. The Insitu Group has access to a secure airspace arena in Boardman, OR where flight tests are regularly performed. The same embedded architecture used in the HiL simulation can be easily transferred to an actual UAV during an actual flight test. The primary purpose of the flight test would be to evaluate the algorithms while using an actual magnetometer instead of a simulated sensor. This allows an accurate characterization of the sensor noise profiles and accuracy. This has a significant effect on the previously described target identification algorithm.

For an initial flight test, it is not prudent to fly a searching scenario with an actual submarine. A more logistically feasible target would be a ground based target such as a car or structure. Its magnetic signature could be obtained by the same methods described in section 3.4 (flying a survey over area with no target and then flying same survey with target present and then looking at the difference).

Yet another possibility for a flight test would be to create an artificial target by simply removing a section of the magnetic map.

Figure 3.11(a) shows the true TMI map of a region where a flight test might take place.



(a) True TMI map of survey area (b) TMI map used by agents (c) Desired target signature

Figure 3.11: Images showing how an artificial target can be introduced into simulation.

The purple dashed lines indicate the areas where the target is to be located. To simulate the presence of the target, this section of the TMI map is removed (all values set to zero). Figure 3.11(b) shows the result of this removal and this is the TMI map that is provided to the agents during the mission. Finally, Figure 3.11(c) shows the target signature that the agents are now searching for.

The flight test would also serve to exercise both the task and path planning algorithms along with the search and target identification algorithms in a real environment for a single UAV. Due to safety concerns, simultaneous flight of multiple autonomous flight vehicles operating in close proximity requires significantly more design and preparation. Additional tasks to extend the flight testing towards demonstration of multi-vehicle flight are described in the next section.

3.7.3 Distributed Simulation

Due to their high cost, multiple vehicle simulation using multiple HiL systems is not practical. Therefore, simulations for teams of agents will be carried out using the UW AFSL Distributed Computing Facility (DCF). This test bed consists of multiple networked computers running software for simulating the distributed system environment for multiple vehicles and their intercommunication. It is meant to be a high-fidelity testing environment, accurately simulating the timing and data transfer required for cooperative planning. The DCF can be networked to run in parallel with the HiL and even with actual UAVs in

the field. Using the DCF, it becomes possible to simulate a scenario involving a team of autonomous agents. The team can be comprised of all GeoRanger UAVs or a combination of UAVs and other vehicles (such as autonomous ground vehicles). The architecture for such a distributed simulation is shown below in Figure 3.12.

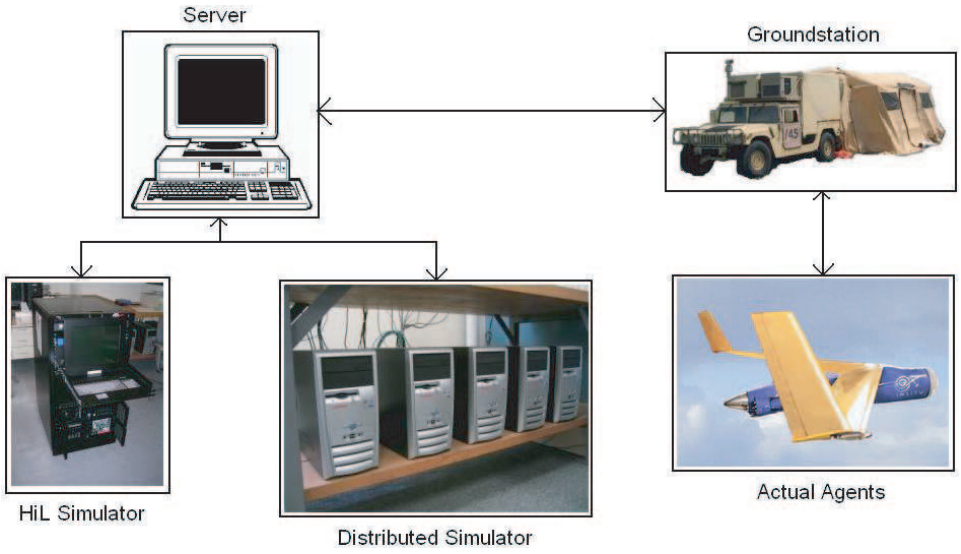


Figure 3.12: Multi-vehicle implementation using HiL and Distributed Computing Facility (DCF).

3.8 Proposed Timeline

A proposed timeline showing major milestones is shown below in Figure 3.13. This reflects work desired to be accomplished under WTC Phase II (extended) and Phase III funding.

Research Schedule, Phase III

TASKS	WHO	Months											
		1	2	3	4	5	6	7	8	9	10	11	12
1. Refinement of world model	L/R	■	■	■									
2. Integration of magnetic data	L/V/S		■	■									
3. Accurate model of sensor/world interface	L/R			■	■								
4. Development of multi-vehicle algorithms	L/P/R				■	■	■						
5. Hardware-in-the-Loop implementation	L/P					■	■	■					
6. Flight test setup	L/P/V/S				■	■	■	■	■	■			
7. Flight test	L/S/R						■	■	■	■	■	■	
8. Communication & Reporting	L/V/R											■	■

L = Chris Lum
R = Rolf Rysdyk
P = Anawat Pongpunwattana
S = Brad Schrick
V = Juris Vagners

Figure 3.13: Milestone chart showing projected major accomplishments.

Chapter 4

SUMMARY

In the most general sense, this work investigates methods to coordinate a group of autonomous systems to accomplish a common goal. Specifically, this goal is to search for a target in a region using the magnetic field of the target and the environment. This work has alternative uses in the fields of obscured vision, night-time operations, radiological hotspot detection, and many other useful applications. The occupancy based map provides the framework to represent probable target locations and also to coordinate the agents in the team. The particle filter method allows the magnetic anomalies to be detected and classified. Simulation has been completed using this framework an a greedy one-step ahead algorithm. These ideas are currently being extended to multi-agent systems with a finite receding horizon window. Although the framework for the system has been established, many more advances and developments are required in order to make this a viable system.

As mission demands become increasingly complicated, the requirement for greater levels of autonomy becomes necessary. It is generally not efficient to have large number of human operator or pilots perform dull, dirty, and dangerous tasks. These operations are better suited for unmanned systems which will require significant advances in autonomy in order to become practical. This project strives to make a contribution towards making teams of autonomous systems an efficient and practical solution to advance mission problems.

BIBLIOGRAPHY

[1] Puget sound aeromagnetic maps and data. Public Information. `http://pubs.usgs.gov/of/1999/of99-514/`.

[2] Frederic Bourgault and Hugh F. Durrant-Whyte. Communication in general decentralized filters and the coordinated search strategy. In *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, 2004. Australian Centre for Field Robotics.

[3] Frederic Bourgault, Tomonari Furukawa, and Hugh Durrant-Whyte. Coordinated decentralized search for a lost target in a bayesian world. In *Proceedings of the 2003 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, Las Vegas, NV, October 2003. Australian Centre for Field Robotics.

[4] Mark Dransfield, Asbjorn Christensen, and Guimin Liu. Airborne vector magnetics mapping of remanently magnetised banded iron-formations at rocklea, western australia. In *Proceedings of the ASEG 16th Geophysical Conference and Exhibition*, Adelaide, Australia, February 2003.

[5] Matthew Flint, Marios Polycarpou, and Emmanuel Fernandez-Gaucherand. Cooperative control for multiple autonomous uav's searching for targets. In *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, NV, 2004. University of Cincinnati.

[6] Dieter Fox, Jeffrey Hightower, Lin Liao, and Dirk Schulz. Bayesian filtering for location estimation. *IEEE Pervasive Computing*, pages 23–33, July 2003.

[7] Emilio Frazzoli and Francesco Bullo. Decentralized algorithms for vehicle routing in

a stochastic time-varying environment. In *Proceedings of the IEEE Conference on Decision and Control*, December 2004.

[8] Greg Hodges. Notes on magnetic model for submarine. Technical report, Fugro Airborne Surveys, Toronto, 2004.

[9] Chien-Feng Huang, David H. Wolpert, Stefan Bieniawski, and Charlie E.M. Strauss. A comparative study of probability collectives based multi-agent systems and genetic algorithms. In *Proceedings of the GECCO 2005 Conference*, 2005.

[10] Daniel J. Klein and Jonas O. Klink. Mobile robot localization. Technical report, University of Washington, Seattle, WA, 2005.

[11] Christopher W. Lum, Rolf T. Rysdyk, and Anawat Pongpunwattana. Autonomous airborne geomagnetic surveying and target identification. In *Proceedings of the 2005 Infotech@Aerospace Conference*, Arlington, VA, September 2005. Autonomous Flight Systems Laboratory.

[12] Anawat Pongpunwattana and Rolf T. Rysdyk. Real-time planning for multiple autonomous vehicles in dynamic uncertain environments. *AIAA Journal of Aerospace Computing, Information, and Communication*, pages 580–604, December 2004.

[13] Anawat Pongpunwattana, Rolf T. Rysdyk, Juris Vagners, and David Rathbun. Market-based co-evolution planning for multiple autonomous vehicles. In *Proceedings of the AIAA Unmanned Unlimited Conference*. Autonomous Flight Systems Laboratory, 2003.

[14] R.T. Rockafellar. *Network Flows and Monotropic Optimization*. Athena Scientific, Belmont, Mass, 1st edition, 1998.

[15] R.T. Rockafellar. Fundamentals of optimization. Technical report, University of Washington, Seattle, WA, 2006.

[16] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.