Cooperative Autonomous Tracking and Prosecution of Targets Using Range-Only Sensors

by

Arthur D. Anderson

B.S., Engineering Science and Mechanics, The Pennsylvania State University, 2006

Submitted to the Department of Mechanical Engineering in partial fulfillment of the requirements for the degrees of

Naval Engineer

and

Master of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2013

Author ....
Department of Mechanical Engineering
June 2013

Certified by ....
John J. Leonard
Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Certified by ....
Michael R. Benjamin
Research Scientist, Department of Mechanical Engineering
Thesis Supervisor

Accepted by ....
David E. Hardt
Chairman, Department Committee on Graduate Students

Cooperative Autonomous Tracking and Prosecution of Targets Using Range-Only Sensors

by

Arthur D. Anderson

Submitted to the Department of Mechanical Engineering
on June 2013, in partial fulfillment of the
requirements for the degrees of
Naval Engineer
and
Master of Science in Mechanical Engineering

Abstract

Autonomous platforms and systems are becoming ever more prevalent. They have become smaller, cheaper, have longer duration times, and now more than ever, more capable of processing large amounts of information. Despite these significant technological advances, there is still a level of distrust for the public autonomous systems. In marine and underwater vehicles, autonomy is particularly important being that communications to and from those vehicles are limited, either due to the length of the mission, the distance from their human operators, the sheer number of vehicles being used, or the data transfer rate available from a remote operator to an underwater vehicle through acoustics. The premise for this research is to use the MOOS-IvP code architecture, developed at MIT, to promote and advance marine vehicle autonomy collective knowledge through a project called Hunter-Prey. In this scenario, two or more surface vehicles attempt to cooperatively track an evading underwater target using range-only sensors, and ultimately maneuver into position for a "kill" using a simulated depth charge. This scenario will be distributed to the public through academic institutions and interested parties, who will submit code for the vehicles to compete against one another. The goal for this project is to create and foster an open-source environment where parties can compete and cooperate toward a common goal, the advancement of marine vehicle autonomy. In this paper, the Hunter-Prey scenario is developed, a nominal solution is created, and the parameters for the scenario are analyzed using regression testing through simulation and statistical analysis.

Thesis Supervisor: John J. Leonard
Title: Professor of Mechanical and Ocean Engineering

Thesis Supervisor: Michael R. Benjamin
Title: Research Scientist, Department of Mechanical Engineering

Acknowledgments

I would like to thank the many people who have helped encourage me and support me in completing this thesis. Nothing worth doing is done alone, and this paper is certainly no exception.

Foremost, I'd like to thank Alon Yaari, who has been very patient in teaching me and working with me over the past year, and given me much advice and help through the various steps of this research. Thank you to Mario Bollini, who has been a major contributor to the regression testing code. I'd also like to thank Michael Benjamin, for his reading and editing and advice in the direction of this thesis, for his detailed notes and edits, and his enthusiastic support. And I'd like to thank my thesis adviser, John Leonard, who despite an extremely busy schedule, has found the time to advise and guide me throughout this whole process.

Last and most of all, I'd like to thank my colleagues, friends, and family, who have endured me being away for many hours at a time to work on my studies, and always supported me in more ways than I can mention here, so that I could always be my best. Without their support, none of this would have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Background

1.1 The Need for Autonomy

The focus of this research is Marine Vehicle Autonomy, Communication, and Cooperation. Autonomous platforms and systems, in both the military and the commercial worlds, are becoming ever more prevalent. They have become smaller, cheaper, have longer duration times, and now more than ever more capable of large amounts of information. Ships are being designed with less and less manning, and unmanned vehicles, either in the air or in the water, are being used for numerous applications today. All these are trends toward systems with greater amounts of autonomy with less human input [2].

This progress is due to a number of different reasons. Advances in battery technologies have allowed autonomous platforms to stay out for longer periods of time. Sensors, such as GPS and sonar, are becoming smaller, cheaper and more capable. And computing power, which used to be a highly limiting factor in marine autonomous systems, can get the same and better performance for significantly smaller size and less electrical power. Acoustic communications have also made significant advances in recent years. Figure 1-1 depicts these key components necessary for fielding autonomous marine vehicles, while Figure 1-2 shows how these different technologies, which are still very much in the process of developing and improving, have matured over the past 18 years. These advances in the critical components are what drive the

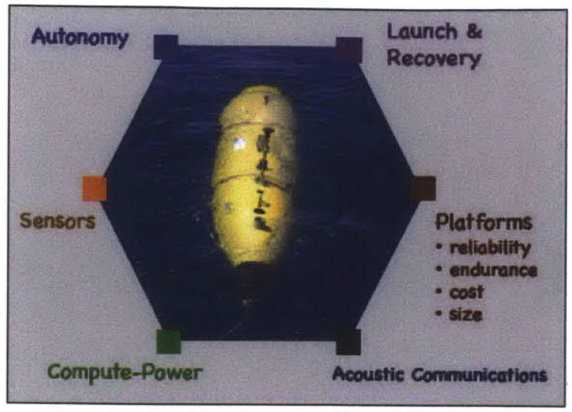


Figure 1-1: There are several key technology components that must mature for effective unmanned marine systems to be developed [1].

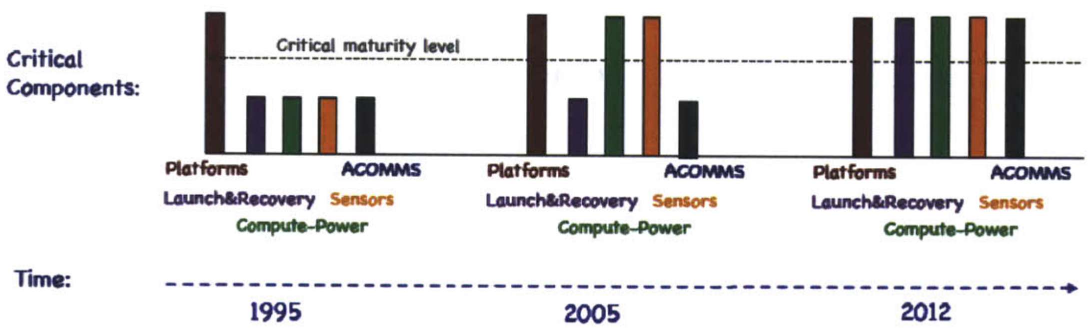


Figure 1-2: Over the past 18 years, the technology capability of the key components for unmanned marine systems has improved significantly. The advances in the other key components drive what is expected from autonomy [2].

expectations of marine autonomy [2].

Despite these significant technological advances, there is still a level of distrust for human operators in autonomous systems, as they are often seen as unreliable or incapable of making important decisions without human input. Autonomy, however, is particularly important especially in the case of marine and underwater vehicles. Communication from those vehicles is often limited, either due to the length of the mission, their distance from human operators, the sheer number of vehicles being used, or the data transfer rate available from a remote operator to an underwater vehicle through acoustics [2].

This gap of trust must be crossed if we are to continue the path of fielding more autonomous vehicles. Therefore, the goal of autonomy research is ultimately to increase the self-reliance of these autonomous systems, and to facilitate a greater trust and understanding for both military and industry in using autonomous vehicles to accomplish their tasks.

1.2 The Hunter-Prey Project

The premise for the Hunter-Prey project is as follows: using the MOOS-IvP code architecture developed at MIT for autonomous marine vehicles, a set of rules will be created for a hunter-prey type scenario, in a way that two or more surface vehicles attempt to track an underwater target using range only sensors. The vehicles will attempt to maneuver in such a way that maximizes their sensing capability of the underwater vehicle and also maneuvers the vehicles into a position for a "kill", using a simulated depth charge, explained later in Chapter 3. The vehicles in play will have limited communication between themselves and a remote human operator.

This scenario and set of rules will then be distributed to a number of different academic institutions and interested parties, who will submit ideas and algorithms to dictate the vehicles behavior, whose performance will be numerically graded and analyzed (also explained further in Chapter 3). These algorithms will be judged based on their ability to track and find an underwater target, as well as the ease of their operator interface with the autonomous system.

The overall project will attempt to accomplish three main goals. First, it will create and foster an open-source environment where many parties can compete and cooperate toward a common goal, which may be useful when more realistic scenarios could be developed and require solutions. Second, it will allow us an in-depth look as to what sort human input is optimal in an environment where human input and communication is limited, and third, how solutions should be shaped in the future. Finally, it will help contribute to the open-source MOOS-IvP code already developed in depth for future potential research and applications.

1.3 Goals of Thesis Research

This goal of the research in this paper is to create baseline upon which the Hunter-Prey project can build. More specifically, this thesis will seek to: 1) define the rules, guidelines, and set-up for the Hunter-Prey scenario, 2) develop a "straw-man" or basic solution to the problem, and 3) run this solution through regression testing to determine which factors, such as sensor capabilities or vehicle speed, affect the problem, and by how much. Lastly, 4) this project will discuss the ways to move forward with the project as it moves toward becoming an open competition. This will allow for a greater understanding of how the parameters affect the problem so we can better set them for the competitors for the more general Hunter-Prey project. The method presented for scoring in this research will also provide a framework for understanding how to measure the success rates of more complex problems, such as including actual acoustic signatures, multiple path returns, and tracking multiple targets, as they are developed.

The solution to the Hunter-Prey problem presented in Chapter 3 is not an optimized solution, but is intended to be a baseline solution that can be used by participants in the Hunter-Prey project for their submissions. Where specific improvements can be made to the algorithms for the vehicles are discussed in Section 4.7, but the searching for a more optimal solution is not the goal of this research.

1.4 Current Literature Review and Comparison

The major difference between this paper and papers that attempt tracking problems is the measure by which success is determined. Other research papers, when evaluating the effectiveness of a range-only tracker or solution, use a least squares measurement of the detected target track against the actual target track. For example, in 2006, a paper by Donald P. Eickstadt and Michael R. Benjamin explore using bearing only sensors to track a target vehicle [7]. In this study, two tracking vehicles do loitering circles, while a third vehicle, the one being tracked, passes between them,

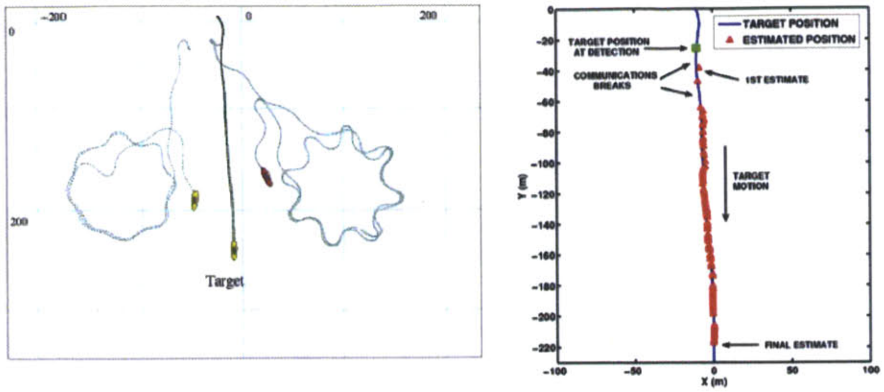


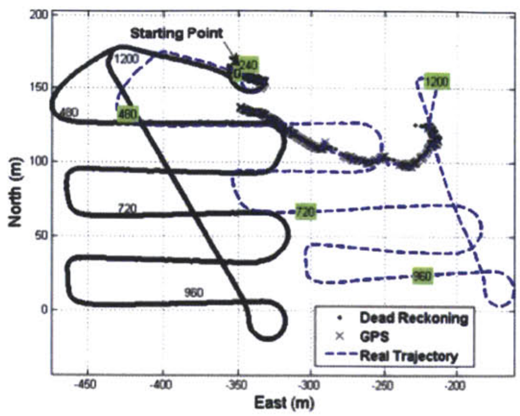
Figure 1-3: Two tracking vehicles conduct loiter circles while a target vehicle tracks between them. The tracking vehicles are using bearing only sensors to attempt to localize the target [7].

as the tracking vehicles attempt to use their sensors to localize the third vehicle as illustrated in Figure 1-3. As we can see from the figures, the accuracy of the tracking algorithm is measured through a least squares estimation.

This is illustrated in another study. In 2010, Gao Rui and Mandar Chitre wrote a paper in which one autonomous vehicle with the ability to locate itself with a high degree of accuracy using GPS, and another vehicle could only find it's position through less accurate dead reckoning [14]. Using range-only measurements between the two vehicles, the vehicle with dead-reckoning tracking is able to obtain a significantly higher degree of accuracy in it's own position, as shown in Figure 1-4. Again, the method for determining success was using a least squares measurement to determine error.

In this paper and the overall Hunter-Prey project, however, the defining principle is not a measurement of positional error, but the measure of overall mission success. This is done because the Hunter-Prey concept is more complex than these developed scenarios and therefore positional error would be more difficult to analyze, and also, more importantly, because the goal of this research is to look at how the range-only tracking problem posed affects the success of the mission, which ultimately is the goal for fielding autonomous vehicles.

It should also be noted that previous work has been done on improving particle filter and other tracking algorithms. For example, in 2013, Guoquan P. Huang and



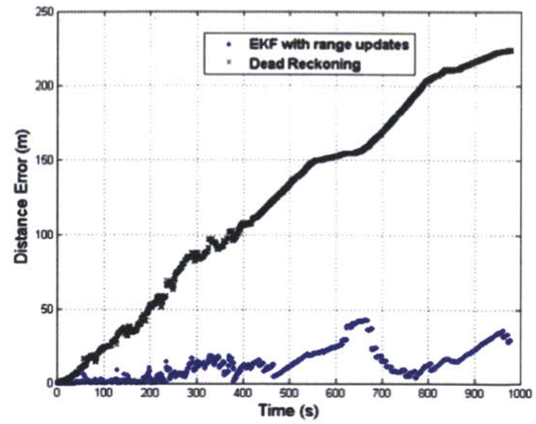


Figure 1-4: Two vehicles, one with good navigational data and the other not, communicate using only range data between them to help localize the vehicle with inferior navigation [14].

Stergios I. Roumeliotis wrote a paper which looks at using a Gaussian mixture based approximation proposal distribution, allowing for slower depletion of particles [9]. The range-only tracking in this solution is solved using a particle filter code developed for MOOS-IvP [13]. This a basic particle filter, and this research doesn't seek to find methods to improve it's performance. In this way, the research can focus on the creation of the project and analysis, instead of researching better particle filter algorithms. Better particle filters can be worked out by competitors making submissions for the Hunter-Prey Project, but is outside the scope of this thesis.

1.5 MOOS-IvP: The Code Architecture

The code architecture to be used for the Hunter-Prey project is based on an open source project called MOOS-IvP. Launched originally at MIT in 2005, MOOS-IvP includes more than 150,000 lines of code and 30+ core applications dedicated to controlling marine vehicles, mission planning, debugging, and post-mission analysis. This software has been run on over a dozen different platforms and is being used at the Office of Naval Research (ONR), the Defense Advanced Research Projects Agency (DARPA), and the National Science Foundation programs at MIT. MOOS-IvP can be used for a simulated environment, or for fielding the vehicles in a real environment.

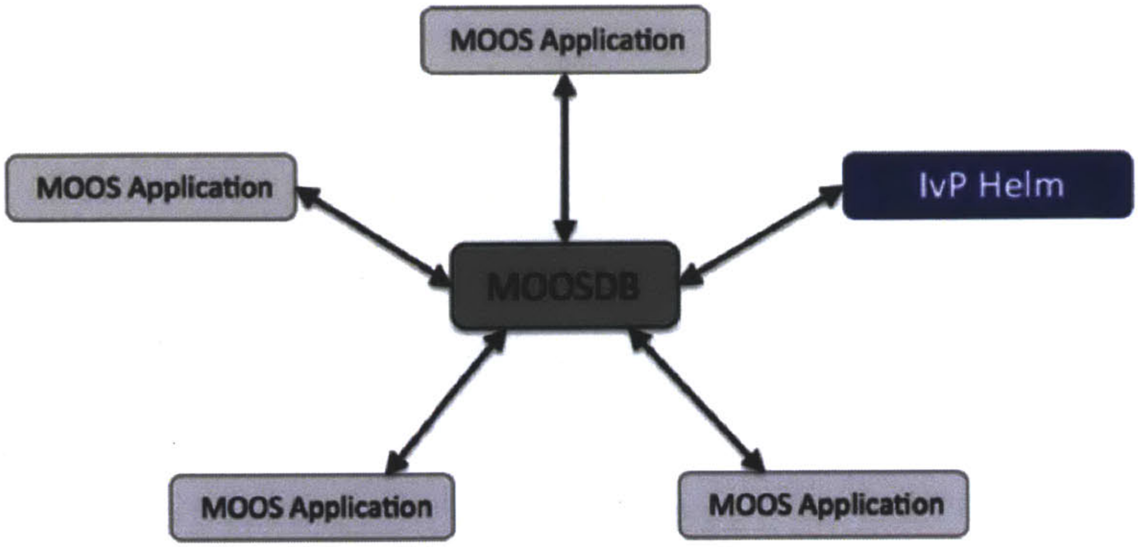


Figure 1-5: A MOOS community is a collection of MOOS applications, each publishing and subscribing to variables published to the MOOSDB. A MOOS community typically operates on a single vehicle or computer. [2].

The MOOS portion of MOOS-IvP stands for "Mission Oriented Operating Suite", and contains a core set of modules that provide a middleware capability based on a publish-subscribe architecture. Processes in the MOOS database are defined by what messages they subscribe to (publications), and what messages they consume (subscriptions). The key idea for MOOS is that it allows for applications that are mostly independent, and that any application can be easily replaced or upgraded with an improved version with the requirement that only its interface match [2]. Figure 1-5 shows a MOOS community, which typically runs on a single machine, and the structure of processes. MOOS communities set-up on different vehicles are also capable of communicating with one another.

The IvP portion stands for Interval Programming, and is a single MOOS application that runs inside the MOOS database. IvP uses a behavior-based architecture for implementing autonomy. These behaviors are distinct modules that are each dedicated to a specific aspect of autonomy, for example, following a set of waypoints or collision avoidance. If multiple behaviors are active, the IvP uses a solver to reconcile the desires of each behavior using an objective function, or IvP function [2]. Figure 1-6 shows how the IvP system structure is structured.

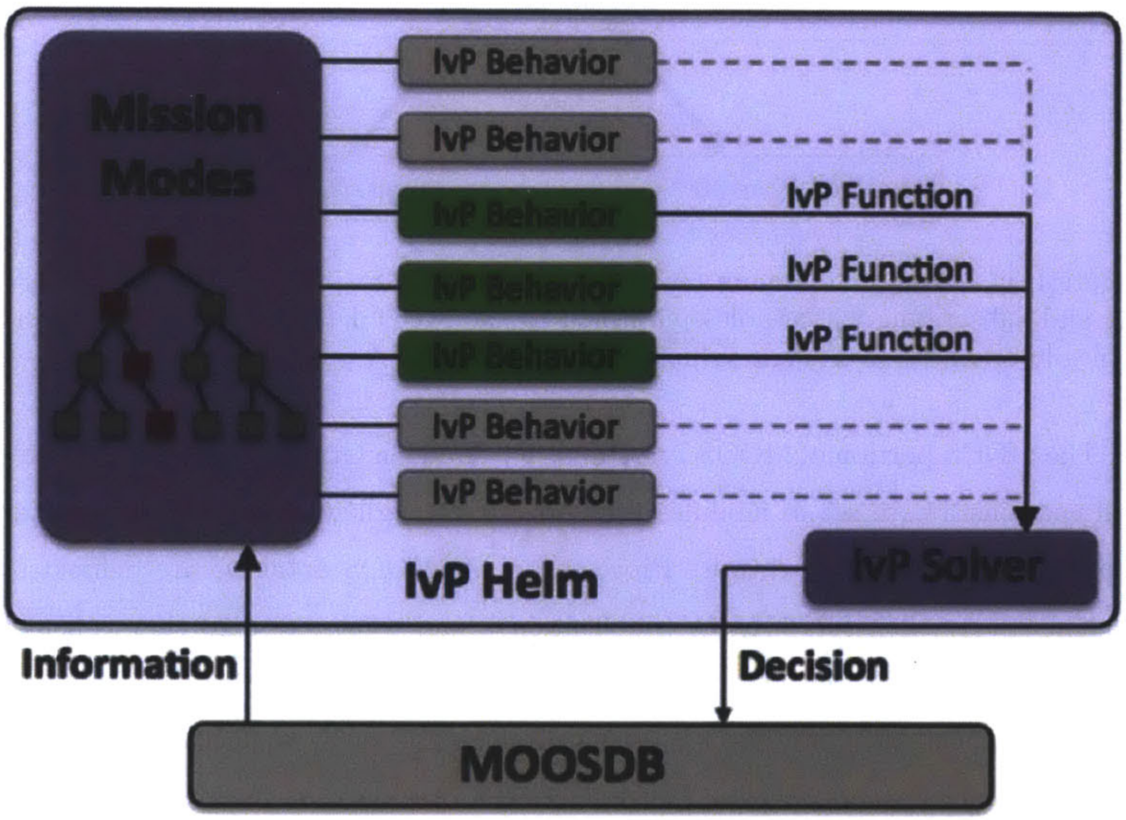


Figure 1-6: The IvP helm is a single MOOS application. It uses a behavior-based architecture in which uses a mode structure to determine which behaviors are active. Each of the active behaviors are then reconciled using a multi-objective optimization solver, or the IvP solver. The resulting decision is then published to the MOOS database [2].

Many existing behaviors already exist as part of the open MOOS-IvP software available. These behaviors a fully leveraged in this project: all behaviors used in this research have already been created and documented in the available MOOS-IvP documentation [2]. The specific behaviors used are discussed in Chapter 3.

Chapter 2

Tracking with a Particle Filter

2.1 Overview

Tracking with a sensor that gives you only ranges can be a difficult problem, although it is certainly not a novel one, and many methods have been used to explore this problem.

This problem is solved using a particle filter, also known as a Sequential Monte Carlo (SMC) method, which is a means of developing target solution using observations from sensors. In the case of the Hunter-Prey problem, this data is in the form of range-only information. A PF simulates possible solutions, or particles, that fit the range observations made, and then readjusts and updates as more information is obtained. If enough particles are generated, the distribution of particles can represent a continuous probability distribution function (pdf) of the target's position. A simple way to think of a PF can be thought of as a method for developing a track solution based on range information, as illustrated in Figure 2-1.

A PF has 4 steps. The first step, called initialization, generates all the particles, and occurs when the first range measurement is received. Each particle can be thought of as a guess as to the position and velocity of the target track. When the next range observation of the target is received, each of the particles, of track guesses, are evaluated based on the new information, and new, more accurate particles can be evaluated if needed. More specifically, upon receiving a new range measurement

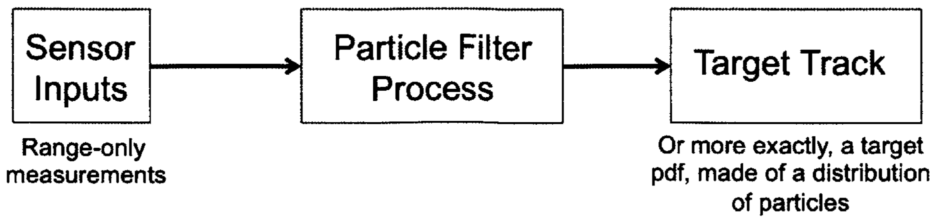


Figure 2-1: The particle filter takes a set of measurements or inputs, and produces a target track.

after the first one, the PF advances the particles to where they would be based on their previous states (step 2), assigns each of the particles weights (step 3), and then checks to see if enough particles have degraded to the point where they need to be resampled (step 4). After each new range measurement is received, steps 2, 3, and 4 are repeated. The following sections describe in greater detail each of these steps and how they are completed.

2.1.1 Step 1: Initialization

When the first range measurement is received, N particles are generated, all at the received range. Each particle is given a Cartesian coordinate:

$$\mathbf{x_i} = [x_i, y_i, z_i]^T \tag{2.1}$$

such that

$$\begin{aligned} x_i &= r_t \sin(\phi_i)\cos(\theta_i) \\ y_i &= r_t \sin(\phi_i)\sin(\theta_i) \\ z_i &= r_t \cos(\phi_i) \end{aligned} \qquad \begin{aligned} &\text{for a random:} \\ &-\pi/2 <= \phi_i <= 0 \\ &0 <= \theta_i <= 2\pi \end{aligned}$$

This creates $i = 1, 2, ..., N$ particles such that they are randomly distributed in a hemisphere below the source (because the target will either be at or below the depth of the vehicle) at the range r_t received by the range sensor at time t. Here, ϕ represents the elevation angle from the sensor to the target and θ represents the bearing to the target. In addition to position, each of the particle filters are also described by a velocity:

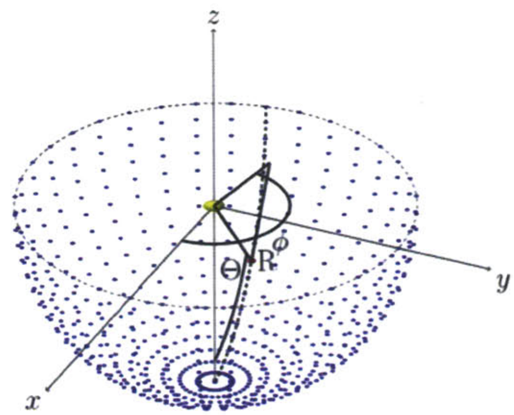


Figure 2-2: A set of N particles randomly distributed at a range r_t from the sensor. This is what the particles distribution will look like after initialization [13].

$$\dot{\mathbf{x}}_\mathbf{i} = [\dot{x}_i, \dot{y}_i, \dot{z}_i]^T \tag{2.2}$$

such that:

$$\begin{array}{ll} & \text{for a random:} \\ \dot{x}_i = v_i \sin(\Phi_i)\cos(\Theta_i) & 0 <= v_i <= v_{max} \\ \dot{y}_i = v_i \sin(\Phi_i)\sin(\Theta_i) & 0 <= \Phi_i <= 2\pi \\ \dot{z}_i = v_i \cos(\Phi_i) & -\pi/2 - \Phi_{max} <= \Phi_i <= \pi/2 + \Phi_{max} \end{array}$$

In the above equations, Φ represents the elevation angle of the target, Θ represents its heading, and v_t its velocity magnitude. v_{max} represents the maximum expected speed of the target track, and Φ_{max} is the maximum elevation angle that the target track can manage. Given the positions and velocities of each particle from Equations 2.1 and 2.2, the state of the particle can be defined as:

$$\zeta_i^t = \begin{bmatrix} \mathbf{x}_\mathbf{i} \\ \dot{\mathbf{x}}_\mathbf{i} \end{bmatrix} \tag{2.3}$$

This equation shows the state of the particle. Stepping back, we see that a number of particles have been generated randomly in a hemisphere below the point source, as shown in Figure 2-2.

2.1.2 Step 2: Prediction

Once the particles have been initialized and then a second range measurement or observation has been received, the particles must be advanced to the new positions. The equations by which the particles advance is simply to use each of the particles positions and velocities from the state Equation 2.3, and the time Δt difference between this report and the last report.

$$
\begin{aligned}
x_i^t &= x_i^{t-1} + \dot{x}_i^{t-1}\Delta t \\
y_i^t &= y_i^{t-1} + \dot{y}_i^{t-1}\Delta t \\
z_i^t &= z_i^{t-1} + \dot{z}_i^{t-1}\Delta t
\end{aligned}
$$

After we advance the particles, in preparation for the next step, noise must be inserted into each of the particles' velocities. This is done for two reasons. First, as we will see in Section 2.1.4, because we will draw new particles from old ones as some particles become degenerate, we want to create some variation such that all the particles are not the same particle. Secondly, if the target track changes heading, elevation angle, or velocity magnitude, we want the particles to be able to follow the target, and this can only be done of some particles are allowed to deviate from their previous velocity state.

How much noise needs to be added is a careful consideration when using a PF. Adding too much noise will cause the particles to go off in many directions, and make it less able to follow a target going in a straight line and constant speed. On the other hand, if not enough noise is added, it may take several time steps before a single particle is able follow a target track going through a sudden speed or velocity change. For the purposes of this paper, the values heuristically determined in Andrew Privette's paper were used [13]. These values are listed in Table 2.1. One solution to the problem of not enough noise to follow a track is the use of reserve particles, discussed in Section 2.2.2

The following is the process for adding noise. First, separate the velocity vector $\dot{\mathbf{x}}_i$ into velocity, heading and speed:

$$v_i^{t-1} = \sqrt{\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2}$$
$$\Theta_i^{t-1} = \tan^{-1}\left(\frac{\dot{x}_i}{\dot{y}_i}\right)$$
$$\Phi_i^{t-1} = \tan^{-1}\left(\frac{\dot{z}_i}{\sqrt{\dot{x}_i^2 + \dot{y}_i^2}}\right)$$

Next, add noise to each of the parameters:

$$v_i^t = v_i^{t-1} + v_{noise}$$
$$\Theta_i^t = \Theta_i^{t-1} + \Theta_{noise}$$
$$\Phi_i^t = \Phi_i^{t-1} + \Phi_{noise}$$

Finally, translate these back into the original velocity vector to describe a new state:

$$\dot{x}_i = v_i \sin(\Phi_i)\cos(\Theta_i)$$
$$\dot{y}_i = v_i \sin(\Phi_i)\sin(\Theta_i)$$
$$\dot{z}_i = v_i \cos(\Phi_i)$$

This new velocity vector is used to describe the particles at the next time step t, This, along with the new position vector from Equation 2.1, fully describes the new state of the particles.

2.1.3 Step 3: Weight Calculation

When the particles have been advanced, we then compare their positions to the range r_t measured from the sensor. We do this by by using an importance factor, or weights, where the weight of a particle represents how likely the particle might be the actual target track given the range observations made. The equation used to determine the weights is:

$$w_i^t = w_i^{t-1}\frac{p(r_t|\zeta_i^t)p(\zeta_i^t|\zeta_i^{t-1})}{q(\zeta_i^t|\zeta_i^{0:t}, r_t)} \tag{2.4}$$

here, $p(\zeta_i^t|\zeta_i^{t-1})$ is known as the transition prior and $q(\zeta_i^t|\zeta_i^{0:t}, r_t)$ is called the importance function, and for convenience we set them equal [11]. This allows us to simplify the weight equation to:

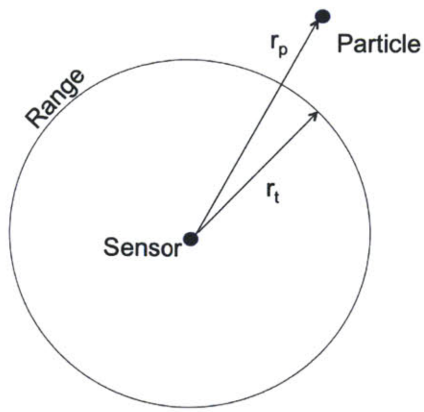


Figure 2-3: The variables measured geometrically for the determination of weights.

$$w_i^t = w_i^{t-1} p(r_t | \zeta_i^t) \tag{2.5}$$

From this equation, we find that the weight of each of the particles is based on the weight from the prior time step, and the probability of the state of the particle r_t given ζ_i^t. We approximate this distribution as a normal distribution.

2.1.4 Step 4: Resampling

As new range information comes in, it will become apparent that many of the particles will be less and less likely. Many particles will be found to just completely incorrect by being completely off from the new range, while some may fall directly on it. If enough accurate information comes in, it may be likely that only one or two of the particles are the most likely solution. This is called particle degeneration, and in order to prevent this, we resample.

The first stage in resampling is to determine whether or not resampling is necessary. To do this, the effective number of particles is calculated using Equation 2.6. If the effective number of particles N_{eff} is lower than a set threshold $N_{threshold}$, then a resampling is performed. $N_{threshold}$ is generally set to half the number of particles $N/2$.

$$N_{eff} = \frac{1}{\sum_{i=1}^{N}(w_i^2)} \quad (2.6)$$

where

$$N_{threshold} = \frac{N}{2}$$

Resampling is then performed based on the weights of the particles. N new particles are drawn from the old set, but particles with higher weights are more likely to be drawn from than particles with low weights. In other words, if a particle has a high weight, then many of the particles of the new set will have the same velocity as the old, higher weighted particle from the previous set.

This idea is illustrated well in Figure 2-4, for two particles. At time $t-1$ each of the particles will have a very high weight because each particle falls on almost the exact range measured by the sensor. However, when new range information is received at time t, Particle 1 is found to be off the range, will be calculated to have a lower weight, and will much less likely to be resampled if resampling will occur at this step. Particle 2 however, will have a high weight at time steps $t-1$ and t, and therefore will be much more likely to be drawn from in the next resampling process.

After N new particles have been drawn, the old set is dropped, and each of the new particles is given an equal weight of $1/N$. These are the particles that will be evaluated at the next time step, when the next new range measurement is received, and the steps are repeated.

2.2 Other Considerations

2.2.1 More Advanced Particle Filters

It should also be known that more extensive particle filtering methods have been developed than he basic particle filter developed for this problem. For example Huang and Roumeliotis in their paper build the probability density function of particles

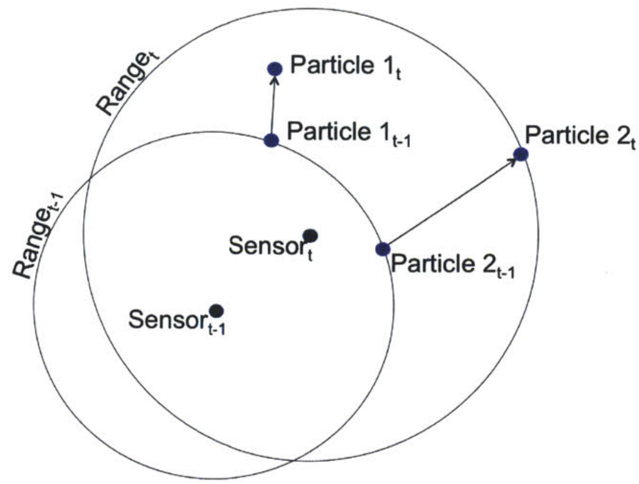


Figure 2-4: Two particle moving being evaluated at initial time $t-1$, and then again at time t.

based on an analytically determined Gaussian pdf rather than an assumed one, which helps reduces the rate of particle depletion [9]. The particle filter could also be improved by analyzing the values used for variance, noise, particle count, reserve particle count, and other parameters. The numbers used in this algorithm were determined empirically, however, using more optimal numbers, or even writing an algorithm that would allow the vehicle itself to calibrate these values autonomously could be highly beneficial [13].

These solutions, while they would certainly lead to better and more efficient tracking, are not used for the purpose of this research, as the goal is only to create a simple "straw-man" solution and then analyze it as discussed in section 1.3. Better filters and algorithms should be the subject of further research, and for submissions to the Hunter-Prey competition.

2.2.2 Reserve Particles

One additional tool used in PF Tracking is the use of reserve particles. When all the particles are resampled, the new particles have the velocity vectors of the old particles from which they are drawn, with some random noise added so that new particles which are drawn from the same old particle are not exactly the same. But if the contact decides to make a sharp turn, this could be a problem for particles attempting to track that turn. For example, is the maximum turning noise was 15 degrees, and the target being tracked turned 60 degrees, it would take the particles at the end of that turning spectrum at least 4 time steps to get to the correct bearing, and by that time, the particles will have made a wider turn than the vehicle, and now need to speed up to catch it.

In order to resolve this problem, reserve particles are used. During the resampling process, these particles are drawn from the the old set similarly to how the previous set was drawn, however, now these vehicles are given a random velocity vector. This way, when the target makes a sharp turn, some of the reserve particles are likely to closely reflect that turn, and are able to track it. These particles should be a minority compared to the other particles, and how many to use exactly is something to be considered. Too many reserve particles means more noise will be generated in tracking a vehicle moving in a straight line, but not enough reserve particles will make it more difficult to catch sharp turns.

2.2.3 2D Tracking in the Hunter-Prey Problem

There are several other factors that come into play when dealing with the specific Hunter-Prey problem. The first is that the problem, as described in greater detail in Chapter 3, is generally presented and solved in 2 dimensions only. This significantly simplifies the particle filter task, in that only X and Y position and velocity components need to be generated. This also means that fewer particles need to be generated in order to achieve an accurate solution.

2.3 PF Parameters

A number of parameters have been mentioned over the course of this chapter, such as the number of particles and the random noise, all which may be tweaked and adjusted in tests. for maximum effectiveness. While a rigorous test was not performed on the PF for the Hunter-Prey project, a series of informal trial and error tests were performed to obtain a good, working solution. Table 2.1 shows the parameters used for tracking in this particular problem:

PF Parameter Values		
N	=	2000
$N_{threshold}$	=	$N/2$
Range Variance	=	30
Speed Noise	=	0.1
Course Noise	=	40
Reserve Particles	=	300

Table 2.1: A list of the parameters used and their values for the particle filter for this research in the Hunter-Prey scenario.

Chapter 3

The Hunter-Prey Scenario

With the understanding of the particle filter and how a vehicle is able to track with range-only information, the next step is to address the full Hunter-Prey scenario scenario. This chapter will discuss how the scenario rules are established, and also provide a set of logics or algorithms that demonstrates a basic solution of how this scenario could be solved from both sides of the problem. This is the solution that will be tested and analyzed in Chapter 4, with an exploration into how each of the parameters set for the problem affect the outcome of the scenario.

3.1 Mission Environment and Vehicles

The Hunter-Prey Scenario has been designed so that it will work inside an operating box (or op-box) within the wi-fi coverage area of the MIT sailing pavilion on the Charles River in Boston. Figure 3-1 shows a satellite picture of how the op-box is situated within the wi-fi area. This is the facility from where MIT's vehicles are launched, and has sufficient area to conduct the full mission. The op-box area within the wi-fi area was chosen to be large enough to conduct the mission, but no so large as to interfere with traffic on the South side of the river.

While the underwater vehicle in the scenario will be submerged, the Hunter-Prey problem is being treated as two dimensional. All participating vehicles and ranges are given locations only in the X-Y plane. This not an unreasonable assumption

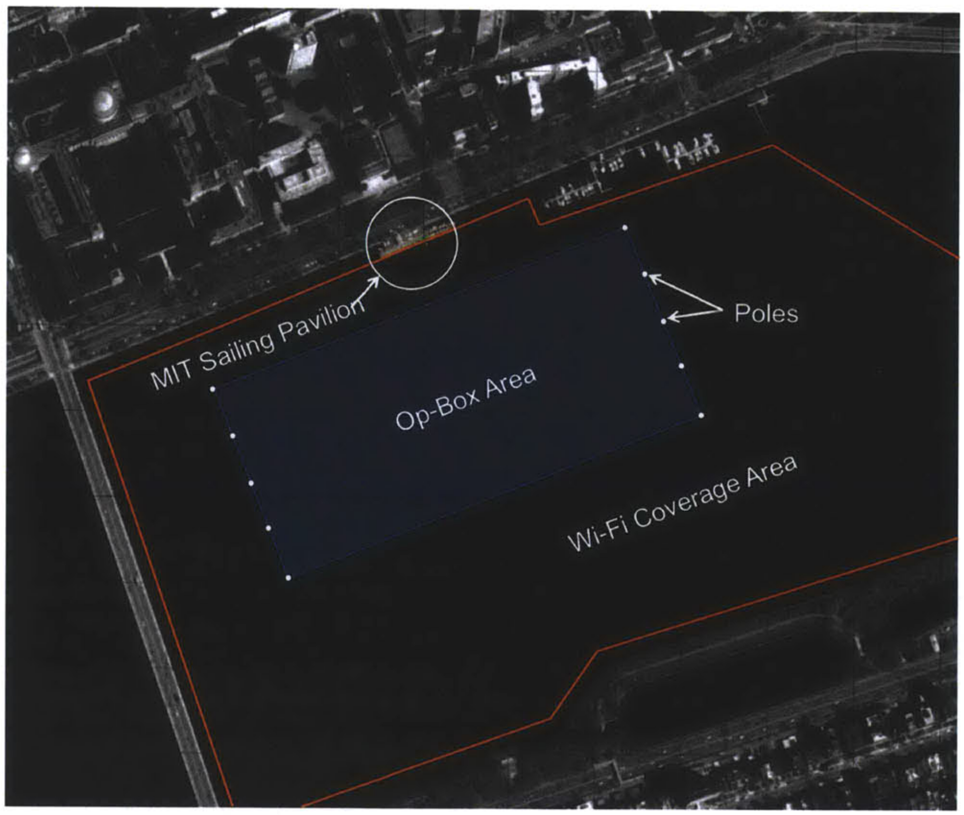


Figure 3-1: The op-box for the Hunter-Prey scenario on the Charles River by the MIT Sailing Pavilion. The orange area represents the wi-fi coverage area, while the blue box represents the op-box. The white dots represent virtual poles, which mark starting positions and waypoints for the vehicles.

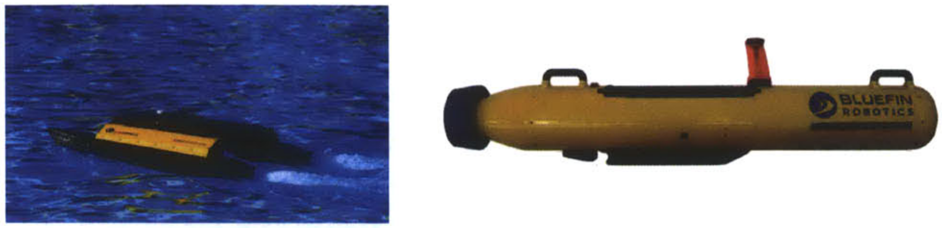


Figure 3-2: *Left:* A Kingfisher M200 Unmanned Surface Vehicle (USV) and *Right:* a Bluefin-9 Unmanned Underwater Vehicle (UUV).

because the Charles River is not particularly deep, the maximum depth being only 12m [10]. And during operation, the UUV will only operate a few feet below the surface. Under this set of rules, a surface vehicle may occupy the same X-Y position as an underwater vehicle, but two surface vehicles may not.

The surface vehicles being used for this mission are Kingfisher M200 USV's. These vehicles are made by Clearpath Robotics, and are the primary research surface vehicles used at MIT. They are relatively inexpensive, are driven by a ducted water jet propulsion system to a maximum speed of 2.0 m/s, and at 64 lbs, are easily launch-able by a single person. Most importantly, they are "autonomy-ready" and can be governed by software developed within the MOOS-IvP architecture. All these features make them an ideal candidate for testing in the Hunter-Prey scenario [6].

The UUV for this mission is a Bluefin-9, which is a lightweight, two-man-portable autonomous underwater vehicle equipped with a side scan sonar and camera. It has multiple navigational sensors, including GPS, a DVL, a CT sensor and a compass, that allows for less than 0.3% error for the distance traveled underwater. Like the M200 USV's, the Bluefin-9's maximum speed is 2.0 m/s, and most importantly the bluefin-9 is capable of accepting a number of different autonomy architectures, including MOOS-IvP [5].

3.2 Description and Rules

3.2.1 General Overview

In this scenario there are three vehicles, two USV's, which are named *Archie* and *Betty*, and the UUV, which is named *Jackal.* The objective of the scenario for Jackal is to start at one of the 5 virtual poles at the west end of the box, travel to one of the poles at the east side of the op-box, and then return to the finish, again, at one of the poles on the west side. The poles are waypoints on the edge of the op-box area, and are illustrated in Figure 3-1. It must do this while trying to avoid the USV's which are attempting to detect and "kill" Jackal using a simulated depth charge.

The goal for Archie and Betty is to prevent Jackal from completing its traversal. To do this, they have two tool at their disposal: each have a range-only sensor and a number of simulated depth charges. In order to stop Jackal, the USV's must drop a depth charge on top of Jackal. The depth charges, once dropped, have a set time delay before they "explode". If at the time of the explosion Jackal is within the range of the depth charge, then Archie and Betty have completed their goal, and the mission ends. The following sections discuss the rules and guidelines for how this scenario is set up.

3.2.2 Initial Set-Up

After all the vehicles are launched form the MIT sailing pavilion, and connect with the MOOS database, a deploy signal is sent from the shoreside computer, which orders the vehicles to travel to their starting positions. Note that there are 5 'poles' labeled on either side of the op-box in Figure 3-1. Upon receiving this command, Jackal submerges and traverses to any of the 5 poles on the west-side, whichever the vehicle so chooses, while Archie and Betty traverse to the east side. Archie's starting position is at the North-East corner of the op-box, or the top East pole, while Betty's is at the South-East corner, or the bottom East pole. All the vehicles then wait at their starting positions until the end of a designated time-period, at the end of which will signal mission start. Figure 3-1 shows the corners of the op-box, as well as the positions for each of the poles.

At the mission start the USV's may begin to search for Jackal, while Jackal may begin its traverse. Jackal must start at one of the 5 poles on the west side of the op-box, pass through one of the poles on the east side, and then finish at any of the poles back on the west side again. During the entire scenario, Jackal is confined to operate only inside the op-box. Archie and Betty have somewhat more free range, and may travel outside the op-box, although they must stay well within the confines of the wi-fi coverage area. They must also at all times never close within 10 meters of each other for safety purposes.

3.2.3 USV Range Sensor Rules

In order to locate the Jackal, Archie and Betty each have a range sensor, which give the range between the vehicle with the sensor and the target. For simulation, the range sensor is simulated on the shoreside by a MOOS application called *uFldContactRangeSensor* [4]. In order to use the range sensor, Archie and Betty must send a request to the shoreside computer for a range to jackal. The proper configuration for this message request is as follows:

CRS_RANGE_REQUEST = name=archie,target=jackal

This request will be received by the *uFldContactRangeSensor* application, which will determine if enough time has passed since the last request, as specified by the mission configuration parameters, and if the target is within range of the requesting vehicle's sensor. If both these conditions are met, the *uFldContactRangeSensor* application will pass back the jackal's range from the shoreside (simulated sensor) to the requesting vehicle in the following format:

CRS_RANGE_REPORT_ARCHIE = vname=archie,range=30,
target=jackal, time=68162

Because the Hunter-Prey problem is two dimensional, ranges from the requesting vehicle to the target are given in the x and y planes only (depth is not considered in the range).

For the purposes of this project, a modified *uFldContactRangeSensor* application has been created that allows for some chance in the sensing, as well as the ability to limit the sensor to certain sectors around the the vehicle.

As configured normally, the *uFldContactRangeSensor* looks at the range between the sensor and the target, and determines if this is less than the *pull distance* plus the *push distance*, and if it is, returns the range. However, in order to create another element of probability into this scenario, and also to more closely simulate an acoustic environment, a modified version of the application was used. In this scenario, the range sensor looks at the *pull distance* plus the *push distance*, adds the two together to create a maximum sensor distance, and then uses the following equation to determine the probability of detection:

$$Probability = e^{\left(\frac{3(Max-d)}{Max}\right)} \tag{3.1}$$

Where Max is the *pull distance* plus the *push distance*, and d is the distance from the sensor to the target. Beyond the Max distance, the probability of detection decays exponentially. This equation can be shown graphically in Figure 3-3 for a Max of 50 meters.

3.2.4 Depth Charge Rules

Depth charges are simulated by the *uFldDepthChargeManager* application run on the shoreside. Each vehicle is given a certain number of depth charges as specified by the mission parameters. In order for a specific vehicle to drop a depth charge, that vehicle must send a message in the following format:

DEPTH_CHARGE_LAUNCH = vname=betty,delay=20

where vname is the name of the requesting vehicle, and and delay is the requested delay for the depth charge. Upon receiving this message, the *uFldDepthChargeManager* application will check to see if at least 5 seconds have passed since the last

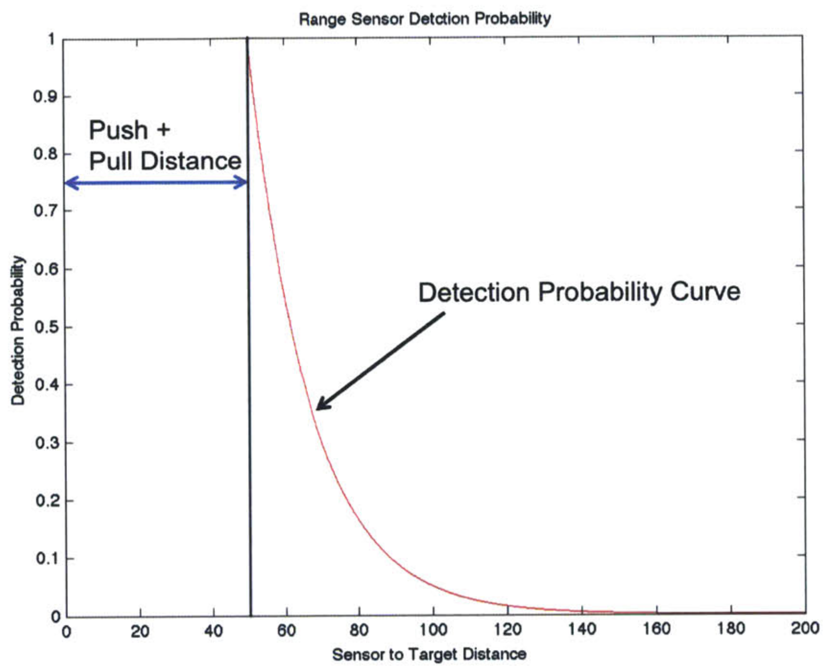


Figure 3-3: This graph shows the probability of receiving a range report on the target given the distance between the sensor and the target. Below the *push* plus *pull distance*, or *Max*, The probability is 100%. At greater ranges, the probability decays exponentially.

charge, if the delay requested for the depth charge is at least the minimum required by the mission parameters, and finally if the vehicles still has any depth charges. If both these conditions are met, a simulated depth charge is be generated, with the delay specified by the user, and a blast radius as specified by the mission parameters. This Hunter-Prey problem is two dimensional, so the the simulated explosion radius is a circle on the horizontal plane surrounding the drop location.

If a surface vehicle wishes to see how many depth charges it has left, and the status of the depth charges it has already launched, it may send the following request to the *uFldDepthChargeManager* application:

DEPTH_CHARGE_STATUS_REQ = vname=archie

The *uFldDepthChargeManager* will receive this request and send a reply in the following format:

DEPTH_CHARGE_STATUS_ARCHIE = name=archie,amt=3,range=25,
launches_ever=2,
launches_now=1,hits=2

If the vehicle has used up all it's depth charges and wishes to get more, it must return to the MIT Sailing Pavilion, marked by the point X = 0, and Y = 0, and then send a request to *uFldDepthChargeMgr* in the following format:

DEPTH_CHARGE_REFILL_REQ = vname=betty

If the vehicle is within 20 meters of the refill range, a counter will begin, and after a certain amount of time has passed (mission parameter $RefillTime$) with the vehicle remaining in range, the vehicles Depth Charge Supply will refill to the maximum amount the vehicle started with. During the refill period, a message will be passed in one of the following formats, depending on the status of the refill:

REFILL_STATUS_ARCHIE = vname=archie,status=refilling,
time_remaining=45.37

REFILL_STATUS_ARCHIE = vname=archie,status=complete,
time_completed=6532

REFILL_STATUS_ARCHIE = vname=archie,status=FAILED,
reason=moved_out_of_range

The first message of the three options above occurs if the vehicle has requested a refill, is in range, but has not yet been near the MIT Sailing Pavilion long enough to receive the depth charges. The second is posted when the vehicle successfully receives the refill, and the third occurs if the vehicle's refill failed because either the vehicle was out of range when the request was sent, or moved out of range between the requested time and the successful completion of the refill.

3.2.5 USV Communication

For this scenario, communication is unlimited, and mail may be passed back and forth between Archie and Betty using the *uFldNodeComms* application. Jackal does not communicate with the other vehicles.

3.2.6 Mission Parameters

The previous couple of sections mentioned "mission parameters" These are variables, such as the speed of the USV, used to describe how the mission is played out. The following is a list of the mission parameters that may change or be defined differently for a given runs.

1. **Sensor Range:** The maximum range at which sensor can sense Jackal 100% of the time.
2. **Sensor Frequency:** The time alloed between range sensor pings.
3. **USV Speed:** The maximum speed for Archie and Betty.
4. **UUV Speed:** The maximum speed for Jackal.

5. **Depth Charge Range:** The explosion radius (2D) of te simulated depth charges.

6. **Depth Charge Amount:** The starting number of depth charges for Archie and Betty, and maximum amount allowed to be held for the duration of the scenario.

7. **Depth Charge Refill Time:** The time vehicle must remain within 20 meters of the refill point in order to refill depth charges to the amount established by Depth Charge Amount.

8. **Depth Charge Delay:** The time following the depth charge drop, before the depth charge explodes.

9. **Start Time:** The time specified between when the vehicles are given the deploy command, and when the mission starts, giving the vehicles time to pre-position themselves at the start.

These mission parameters can be changed for different missions, and the first seven of these will be varied during the regression testing of the Hunter-Prey to determine what values of the parameters will be used for the Hunter-Prey Competition. Because many of these variables are simulated even during real water testing (such as depth charges), they are not limited by constraints. The exception, however, are the USV Speed and UUV Speed parameters, which are limited by the maximum speeds of the vehicles being used in the water. Both USV Speed and UUV Speed can be set no higher than 2 m/s.

3.2.7 Scoring System

In order to create a simple point around which we'll optimize the system, a grading system was created. Basically, the system starts at 200 points, and then users are penalized as time passes, and for each missed depth charge that is dropped. The following is the scoring equation:

$$Score = 200 - C_1(Misses) - C_2((TimeToHit) - T_{Max}) \tag{3.2}$$

where C_1 is the miss multiplier, C_2 is the time multiplier, and T_{Max} refers to the time when the time penalty begins to be applied. These variables are configured by the scorer. The miss multiplier is the number of points lost in the scenario for a given depth charge miss, and the time multiplier is the number of points lost per second after the specified T_{Max} constant. $Misses$ refers to the number of depth charges dropped that detonate and do not hit the UUV, and only refers to misses prior to the hit. After the UUV is hit, the mission is over and all the vehicles are returned to the MIT Dock. T_{Max} refers to the time between mission start and the first depth charge that detonates within the range of the UUV, measured in seconds. The score is capped at 200. The values used for these constants for this research are as follows:

$$\begin{aligned} C_1 &= 5 \\ C_2 &= 0.1 \\ T_{Max} &= 150 \end{aligned} \tag{3.3}$$

Plugging these values into Equation 3.2, and we have the general scoring equation used for the Hunter-Prey scenario:

$$Score = 200 - 5(Misses) - 0.1((TimeToHit) - 150) \tag{3.4}$$

For the purposes of this research, Equation 3.4 will give a single value which can be evaluated given the mission parameters. In the general Hunter-Prey project that will become an open competition, this will will help to provide weights based on the penalties of each. For example, with the constant values specified in the equation above, one depth charge miss is equivalent to 50 extra seconds of mission time in terms of mission score. This is important when writing autonomy code, because the vehicle will need to decide how much time it will want to take to make sure it has enough accuracy and good enough position to drop the depth charge, or if it wants to drop depth charges at will in the hopes that one will hit. How these constants are defined can greatly change the motivation of the autonomy decision making process.

3.3 Nominal Solution

In order to test the different Mission Parameters, a nominal solution was developed to the problem, which was subjected to regression testing. The following sections described the basic solution to the problem - the logics and algorithms that were applied to the vehicles to compete against one another based on te rules set forth in Section 3.2. This solution is not optimized, but was only developed to test the difficulty given a certain set of mission parameters.

3.3.1 UUV Logic

The UUV at this time has no knowledge of where the USVs search for it are. So, the solution was developed using the waypoint behavior, which allows a vehicle to traverse along a set of randomly generated waypoints. Jackal randomly determines which of the 5 poles on the left hand side to start from, and then steers towards that point after the deploy command is given. At the mission start, Jackal begins to traverse the op-box via the generated set of waypoints. The waypoints on the way to the east side are placed at the quarter, half, and three-quarter marks in the East-West direction across the op-box, but are chosen randomly on the North-South. Then a pole is randomly selected for the east side of the op-box for the next waypoint, and then the vehicles uses a similar method for the return. A sample set of waypoints is shown in Figure 3-4.

For future iterations of the Hunter-Prey project, Jackal will have a method to detect the USV's, for example, by knowing the position of the surface vehicles when they ping. This could create more interesting, and more complex, scenarios where the UUV can attempt to maneuver away from the surface vehicles, and where one of the surface vehicles may desire to stop pinging, or "go quiet", in order to mask their position from Jackal. However, for the purposes of keeping this solution and project within a reasonable time-frame, the current waypoint logic will be used without an avoidance behavior.

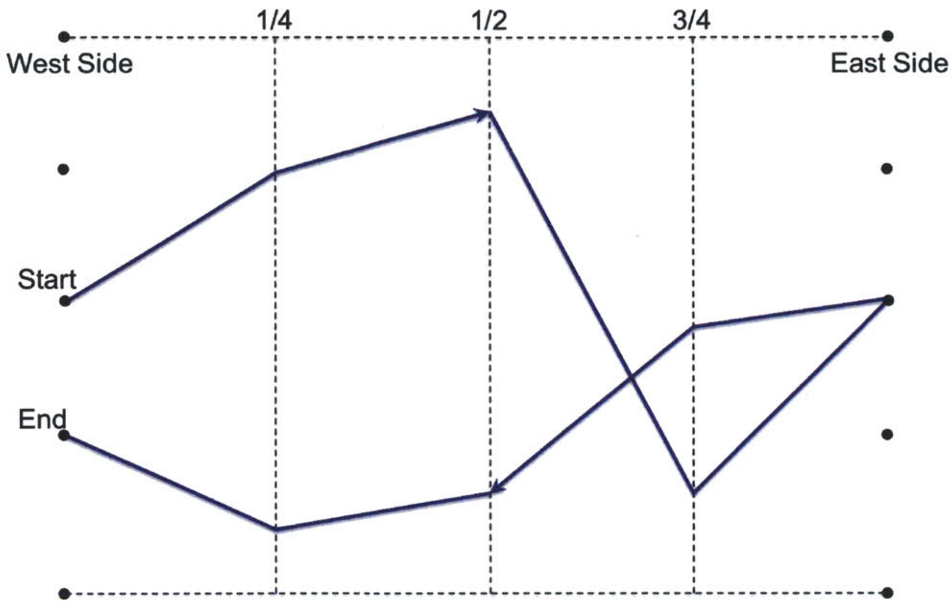


Figure 3-4: An example randomly generated path for Jackal.

3.3.2 USV Logic

Archie and Betty have a somewhat more complex set or rules than Jackal. While in operation, the two vehicles have five basic modes in which they operate: Start, Resupply, Search, Track, and Prosecute. Each of these modes utilize a specific behavior from the standard set of behaviors from the MOOS-IvP tree, which will be described in the following paragraphs. In addition to these behaviors, a collision avoidance behavior is also always active in order to ensure the two vehicles do not collide with one another. The IvP solver, described in Section 1.5, uses its multi-objective optimization algorithm to decide which direction to go. A brief description of each of the modes, as well as the set of conditions required for those modes to be activated is shown in Figure 3-5. The following paragraphs describe in greater detail each of these modes and the conditions required for them to be met.

Start Mode

The Start mode utilizes the station keep behavior as defined in the MOOS-IvP manual [3]. When the initial deploy command is given to the vehicles as they are launched from the dock, they enter Start mode and traverse to their respective starting poles,

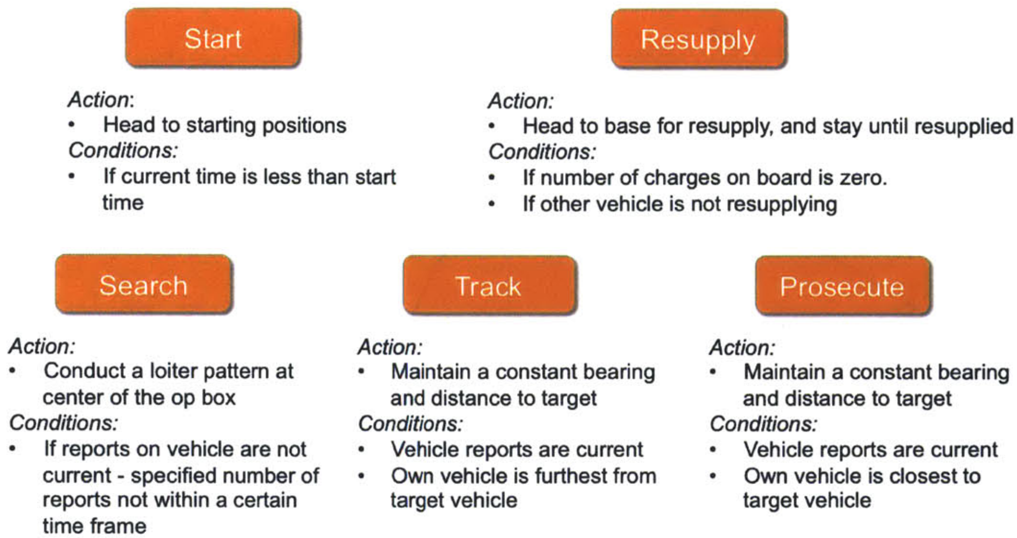


Figure 3-5: The five behaviors which govern the movement of the USV's, their descriptions, and the the conditions that must be met for them to be activated.

the top and bottom poles on the east side of the op-box, and station-keep there until the mission start. The mission start is a pre-determined time specified in the launch file for both the USV and the UUVs after the initial deployment command is given. After mission start, the vehicles exit this mode, and use the others for the remainder of the mission.

Search Mode

Typically, the Search mode is the first mode the vehicles enter following Start. Search mode is used when the vehicles do not currently know where the UUV is, so they search the op-box while pinging their range sensors searching for Jackal. The condition for this behavior is that the vehicles don't have current track information on Jackal. More specifically, if between both surface vehicles, less than three range reports have been received on Jackal during the previous 60 seconds, the reports will be considered "not current", and the vehicle will be in Search mode.

There are many ways this searching could be optimized, however because only a "straw-man" solution is being developed for this research, the idea relatively simple. Within the op-box, the vehicles move in circles (or more specifically polygons), using

Figure 3-6: The position of the loiter circles for Archie and Betty when in "Search" mode.

the loiter behavior as defined in the MOOS-IvP manual [3]. The specific location of those circles is illustrated in Figure 3-6. While the vehicles are in Search mode, the vehicles will continue to traverse these loiter circles until enough reports have been received to have the track be considered "current".

Prosecute Mode

The Prosecute mode, along with Track mode, are the two modes the surface vehicles will enter (one in each mode) when the vehicles are not in Start or Resupply mode, and if range reports and are considered "current", meaning more than 3 range reports have been received within the last 60 seconds between the two vehicles. Based on the output of the particle filter, the vehicle closest to the target track will enter Prosecute mode, while the vehicle furthest from the target track will enter Track mode.

The job of a vehicle in Prosecute mode is to attempt to maneuver in front of Jackal and drop a depth charge. It uses the CutRange behavior [3] to accomplish this. The prosecuting USV will attempt to move in front of the target track, based on its perceived velocity by a distance determined by the product of the speed of the target track and the delay on the depth charge, as shown in Equation 3.5. This position is called the drop point.

$$Distance = V_T T_{Delay} \tag{3.5}$$

where V_T is the speed of the the target and T_{Delay} is the time delay on the Depth Charge. When a USV in Prosecute mode is within 15 meters of the drop point, a drop timer begins. If the prosecuting vehicle can stay within 15 meters for the duration of the drop timer, the vehicle sends a message to drop a depth charge as described in Section 3.2.4. The drop timer for the prosecuting vehicle is reset if the distance between the prosecuting vehicle and target track becomes more than 15 meters, or if the vehicle drops a depth charge.

Track Mode

The Track mode is based on the Trail behavior from the standard MOOS-IvP library [3]. When in Track mode, a vehicle will attempt to maneuver itself south relative to the target position at a range of 50 meters. From here, the surface vehicle's goal is to be close enough to receive range reports, but also not so close that it interferes with the other vehicles, which will be attempting to prosecute Jackal. The vehicle will follow the Track behavior if range reports on the target are "current" (more than 3 reports in the last 60 seconds between the two vehicles), and the vehicle is not in Resupply or Start mode.

Resupply Mode

The Resupply mode is designed to work to resupply Archie and Betty with depth charges after they've run out. It uses the waypoint behavior [3] to do this. When in Resupply mode, a vehicle will steer towards the refill point at the end of the MIT sailing pavilion dock, and then remain there until the vehicle is refilled by the

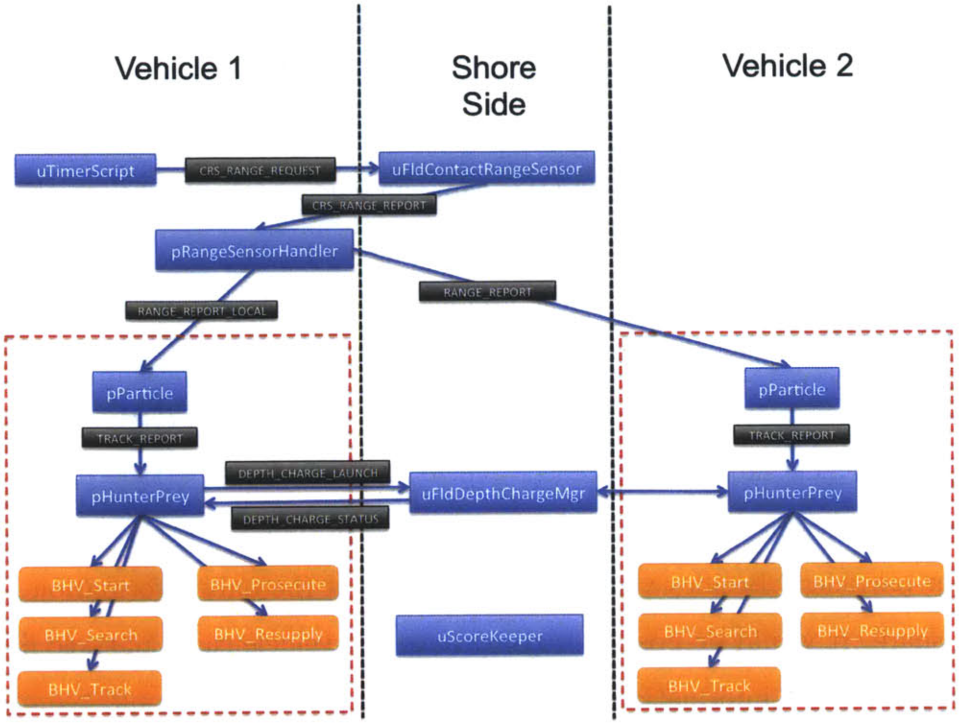


Figure 3-7: A basic diagram of how the USV's interact between applications and between vehicles in a simulated environment.

uFldDepthChargeMgr application, described in Section 3.2.4. The conditions for this behavior are three-fold: 1) the vehicle is not in Start mode, 2) the vehicle has zero depth charges left, and 3) the other surface vehicle is not currently also in Resupply mode. Once a vehicles entera into Resupply mode, it will remain in this mode until a refill complete message is received.

3.3.3 USV Code Architecture

In order for surface vehicles to work in the manner described in the previous Section 3.3, an architecture of applications was set-up set up to work within the MOOS-IvP data structure. Figure 3-7 shows a simplified model of each of the processes developed, how each of the processes communicate with on another, and also how information is passed between each of the vehicles. The process is designed so as in the spirit of keeping communication between the vehicles to a minimum.

The process is simple. If the vehicle is actively conducting the mission, the *uTimerScript* application will generate the range request, and send it to the range sensor on board the vehicle, in this case simulated by *uFldContactRangeSensor*. Upon receipt of the message *uFldContactRangeSensor* will determine if the sensor can detect the UUV (see Section 3.2.3), and if it can, will send a message back to the surface vehicle declaring the range.

This message is then received by the *pRangeSensorHandler* application, which was written for two purposes: 1) to reformat the message in a way that can be absorbed by the *pParticle* application, and 2) to send this message also to the *pParticle* application on the other vehicle. In this way, each vehicle has a separate particle filter running on board, but uses information from both vehicles in order to generate and update their own particles. Besides vehicles position, this is the only information the two vehicles share. It would be possible to only have one vehicle run a particle filter and then send the track information to the other vehicle, but this would require more communication, and would require one vehicle to rely entirely on the other.

The application *pParticle* then performs its function (see Chapter 2) and outputs a best guess on the track on the UUV, which is then sent to the *pHunterPrey* application. Based on the information it receives, *pHunterPrey* will dictate which of the 5 designated modes to follow (see Section 3.3.2), and will also send messages to drop Depth Charges to the *uFldDepthChargeMgr* application (see Section 3.2.4), which simulates the depth charge.

While all this is happening, another application, *uScoreKeeper*, is keeping a running tally of the overall score of the mission (Section 3.2.7), and outputting a visual representation for users to follow throughout the mission. *uScoreKeeper* also reports the final score when a hit is recorded to be used by the regression tester, described in Chapter 4.

Chapter 4

Regression Testing and Analysis

4.1 Goals of Testing

In order to determine the validity of the parameters being varies, a series of regression tests were run to find out which parameters affected the problem the most. In order to accomplish this, a design of experiments (DOE) was put together, where each of the parameters were varied, and then studied to see their effects on the overall score of the mission. This is important because by understanding how the parameters affect the problem, we can set up the a problem that is neither to easy, nor too hard for any of the vehicles.

4.2 Variables and Description

The variables for this experiment are listed in Section 3.2.6, and for the rest of this chapter, will be called effects on the solution, be denoted by capital letters. The following list shows the variables and their associated letters.

The list also shows the baseline values used for each of these variables. These baseline values were determined using the boundaries of the real world (such as vehicle speeds) and by using a heuristic approach to determine which values made the scenario difficult, but not so much that the surface vehicles never find Jackal.

Each of these variables are considered independent variables, X, of the experiment

		Parameter Designator	Value	
A	-	Sensor Range	50	meters
B	-	Sensor Frequency	10	seconds
C	-	USV Speed	1.75	m/s
D	-	UUV Speed	1	m/s
E	-	Depth Charge Range	20	meters
F	-	Depth Charge Amount	4	charges
G	-	Depth Charge Refill Time	60	seconds

Table 4.1: The letter designators for each of the parameters, as well as the heuristically obtained baseline values.

or simulation, while the the score achieved during simulation or testing as described in Section 3.2.7 is a dependent variable, Y. The goal of this testing is to use this set of experiments to develop a model which predicts the dependent variable Y as a function of the dependent variables X, and the randomness of the simulation described by the unknown variables β as in Equation 4.1.

$$Y = f\{X|\beta\} \tag{4.1}$$

In order to keep the model simple, a linear regression model was used to determine each of the effects. A sample linear regression for a single dependent variable is as follows in Equation 4.2, where the dependent variable y is a function of the parameters β_i, the independent variables x_i and the random noise ϵ associated with the simulation. The model for the Hunter-Prey problem is the same, but with seven β independent variables instead of two.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \epsilon \tag{4.2}$$

In order to determine the model, we take the baseline values determined heuristically above, and vary them positively and negatively to see how they affects the overall score.

In the example Equation 4.2 above, if one wanted to see the effect of the varying the sensor range (A), sensor frequency (B), and their combination effects (AB), two

Expt #	A	B	AB
1	+	+	+
2	+	-	-
3	-	+	-
4	-	-	+

Table 4.2: A sample Design of Experiments (DOE) for two parameters.

values would be chosen, one above and one below 50 meters (in this experiment 25 and 75 meters for the sensor range and 5 seconds and 15 seconds for the sensor frequency), and then the scores for each of those experiments would be compared. The higher values would each map to the independent variable value of $x = 1$, while the lower values would map to the value of $x_i = -1$. By running a full factorial experiment of four simulations, or 2^2, as shown in Table 4.2, the values of the four parameters for β_i can be determined, and the overall model can be developed. Note that a value of $+$ refers to a $x_i = 1$ and $-$ is $x_i = -1$. Also, the combination, or the multiplication of the A and B values, is also checked.

With seven variables as in the Hunter-Prey scenario, the picture is more complicated, as we look at the effects of each of the variables along with the pairwise combinations of all of them, which comes to a total of 28 possible effects, which are measured by a full factorial design of 2^7 or 128 experiments. Also, in order to get better results (and to help reduce the significant noise that will be associated with the score) 4 regression tests total were performed for a total of 512 experiments.

4.2.1 Assumptions

There are several assumptions that need to be addressed. First, this model assumes the problem is linear or that a change to one of the independent variables, or a combination of the variables will correspond to a proportional linear change in the dependent variable. While this is not the exact model, it can be considered a good approximation for within the boundaries set in the high and low values chosen. It will give a good understanding as to the general understanding of the problem.

Another assumption is that the variance in the noise remains constant at all levels of the parameters. This is also not an exact picture of the score results. For example, if a certain set of parameters almost guarantees complete success or complete failure of the USV's, they will continue to receive the lowest or highest scores, resulting in less variation than if all the results fell somewhere in the middle. However, because this constant variation is only important in the selection of the important variables, it is an acceptable assumption to make and good judgment principles about the end result can be applied.

4.3 Determining the Main Effects

The effect of a single variable or combination of variables on the score can be determined by adding the values of the experiments when the parameter being tested is positive, subtracting the experimental values form the experiment is negative, and then normalizing the result [16]. For example, in a 2^3 factorial experiment, when the effect or parameter being evaluated is positive in the 1st, 3rd, 5th, and 7th, experiments, but negative in the 2nd, 4th, 6th, and 8th, the total effect can be calculated by the following equation:

$$\beta = d_+ - d_- = \frac{1}{4}[(d_1 + d_3 + d_5 + d_7) - (d_2 + d_4 + d_6 + d_8)] \tag{4.3}$$

In the above example, d_i refers to the ith experiment, and β is the effect of the variable. The MATLAB code developed for this research is capable of determining whether an effect is positive or negative for each experiment, and calculating each effect accordingly. After the effect is calculated, this can now be plugged back into and equation similar to Equation 4.2, for the entire model. While the effects of each problem can be calculated, it is also important to understand if the effect is statistically significant. This issue is addressed in Section 4.4.

4.4 ANOVA Testing Background

When all the data had been collected for each of the experiments for each of the regression runs, an analysis of variance or ANOVA test was performed in order to determine which effects were statistically significant [16]. Before this is done, however, some terms need to be defined. A *treatment* is a set of parameters that are all the same. There are 128 total treatment combinations for this DOE, and a set of simulations with one of each treatment is a regression. Each run or experiment or simulation for a given treatment is called an *observation.* Now, let y_{ti} be the score obtained for the ith observation of the tth treatment. The total sum of the squares *within the tth treatment* is given by the following equation:

$$S_t = \sum_{i=1}^{n_t} (y_{ti} - \bar{y}_t)^2 \tag{4.4}$$

where n is the total number of observations for a given treatment, and $\bar{y}_t$ is the mean of all the observations within a treatment. The sum of the squares *within an effect* is given as follows:

$$S_R = \sum_{t=1}^{k} S_t = \sum_{t=1}^{k} \sum_{i=1}^{n_t} (y_{ti} - \bar{y}_t)^2 \tag{4.5}$$

where k is the total number of treatments. This gives the residual, or error, sum of squares. The *between treatments* sum of the squares is given by:

$$S_T = \sum_{i=1}^{n_t} (\bar{y}_t - \bar{y})^2 \tag{4.6}$$

and the *total* sum of the squares about the grand average is:

$$S_D = \sum_{t=1}^{k} \sum_{i=1}^{n} (y_{ti} - \bar{y})^2 \tag{4.7}$$

where N is the total number of simulations. We then determine the respective degrees of freedom for each of the sum of the squares values as follows:

$$\begin{aligned} \nu_R &= N - k \\ \nu_T &= k - 1 \\ \nu_D &= N - 1 \end{aligned} \tag{4.8}$$

and in order to complete the analysis is the estimate of the variance for each of these values, we finally find the *mean square*, or the total *within treatment*, *between treatments*, and *total* mean squares with the following set of equations, respectively:

$$\begin{aligned} s_R^2 = \frac{S_R}{\nu_R} &= \frac{\sum_{t=1}^{k}\sum_{i=1}^{n_t}(y_{ti} - \bar{y}_t)^2}{N - k} \\ s_T^2 = \frac{S_T}{\nu_T} &= \frac{\sum_{i=1}^{n_t}(\bar{y}_t - \bar{y})^2}{k - 1} \\ s_T^2 = \frac{S_D}{\nu_D} &= \frac{\sum_{t=1}^{k}\sum_{i=1}^{n}(y_{ti} - \bar{y})^2}{N - 1} \end{aligned} \tag{4.9}$$

We then calculate each of these values, and then perform an F-test [16] to determine if these values are having a significant or negligible effect on the dependent variable, or the average end score for a given treatment or set of parameters. The F-test involves comparing the mean square of the between treatments to the within treatments (or residuals) to test to what degree the null hypothesis, which states that there is no correlation between the treatments, is true. The F-Ratio is calculated as in Equation 4.10:

$$F\_Ratio = \frac{s_R^2}{s_T^2} \tag{4.10}$$

With the assumptions given in Section 4.2.1, the F-Ratio will have an distribution function $g(F)$ such that:

$$g(F|\nu_1,\nu_2) = \frac{\Gamma\left(\frac{\nu_1+\nu_2}{2}\right)\left(\frac{\nu_1}{\nu_2}\right)^{u/2}}{\Gamma\left(\frac{\nu_1}{2}\right)\Gamma\left(\frac{\nu_2}{2}\right)} \frac{F^{\nu_1/2-1}}{\left[\left(\frac{\nu_1}{2}\right)F+1\right]^{(\nu_1+\nu_2)/2}} \tag{4.11}$$

where Γ is the Gamma function [16] and ν_1 and ν_2 in the Hunter Prey case are the the degrees of freedom for ν_T and ν_R, respectively as defined in Equation 4.9. Because the F-Ratio will follow this distribution if the null hypothesis is true, it is therefore possible to calculate the probability of the accept the null hypothesis with Equation 4.12:

$$P\_Value = \int_F^{\infty} g(F|\nu_T,\nu_R)dx \tag{4.12}$$

By convention and for the purposes of this study, the null hypothesis can be rejected if the P-value is less than 0.05, or 5%. This method will be used to analyze the regression tests of the simulations to determine which effects are important and which are not, and also estimate how much of an effect they have on the score. The calculations were done using the MATLAB programming language.

4.5 Testing Results

The 7 variables mentioned were tested at higher and lower values than listed in Section 4.2. Table 4.3 shows each of the variables that were tested. The Hunter-Prey simulation was run through these tests with every possible combination of these variables for a total of $2^7 = 128$ simulations per regression, for a total of 512 simulations or experiments. The first step was to determine the effects of each of the parameters. Table 4.4 shows the main effects calculated.

From the table, one can conclude that some effects have a much greater effect on the outcome of the score than others. For example, on average, the difference between a range setting of 25 meters and 75 meters will result in a difference in scores of about 48, while the depth charge amount (the max number of depth charges that can be carried by the vehicles) only effects the problem by about 6 points.

Table 4.3: Regression Testing Parameter Values

	Variable	-	+	units
A	Sensor Range	25	75	meters
B	Sensor Frequency	5	15	seconds
C	USV Speed	1.5	2	seconds
D	UUV Speed	0.75	1.25	m/s
E	Depth Charge Range	15	25	meters
F	Depth Charge Amount	2	6	charges
G	Depth Charge Refill Time	30	120	seconds

Table 4.4: Effects' Names and Values

Desig.	Parameter	Effect
A	Sensor Range	48.1
B	Sensor Frequency	-43.3
C	USV Speed	35.3
D	UUV Speed	-87.9
E	Depth Charge Range	38.6
F	Depth Charge Amount	5.6
G	Depth Charge Refill Time	3.7

In order to be thorough, secondary effects were also determined. This involves looking at two variables to determine if there is an interaction between them. For n = 7 parameters, there are $n(n-1)/2 = 21$ interaction effects. These interaction effects were calculated by the regression analysis program and can be seen in Table 4.5.

After calculating the effects, it's important to see if the effects are large enough to be considered statistically significant. Some variables are more apparent, for example, the Speed of the UUV likely has a significant effect on the score, while the depth charge refill time probably does not. However, for some of the effects which have values around 10 or 15, it's unclear whether the variables really do have an effect, or if those effects are only due to simulation noise. This uncertainty can be determine by calculating the P-value, as described in Section 4.4. The following are the ANOVA tables for the main effects. The equations used for each of these tables are discussed in Section 4.4. The ANOVA tables for the interaction effects can be found in Appendix C.

Table 4.5: Interaction Effects

Interaction	Effect	Interaction	Effect
A	48.1	BD	10.1
B	-43.3	BE	-12.3
C	35.3	BF	-1.2
D	-87.9	BG	4.8
E	38.6	CD	4.2
F	5.6	CE	6.6
G	3.7	CF	1.3
AB	15.2	CG	-6.8
AC	8.2	DE	-7.3
AD	-12.3	DF	-3.1
AE	0.1	DG	-7.9
AF	-0.3	EF	2.6
AG	-7.1	EG	-0.3
BC	-4.3	FG	0.1

Figure 4-1: The full ANOVA tables for the 7 main effects. The values for these numbers are calculated via the methods presented in Section 4.4

Sensor Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	148343	1	148343	24.0
Residuals	3153035	510	6182	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.0000

Sensor Frequency

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	120412	1	120412	19.1
Residuals	3217770	510	6309	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.0000

USV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	79921	1	79921	12.4
Residuals	3280275	510	6432	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.0005

UUV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	494997	1	494997	102.2
Residuals	2470408	510	4844	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.0000

Depth Charge Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	95704	1	95704	15.0
Residuals	3250989	510	6374	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.0001

Depth Charge Amount

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	1975	1	1975	0.3
Residuals	3443379	510	6752	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.5888

```
Depth Charge Refill Time
-----------------------------------------------------
Source of          Sum of    Degrees of    Mean
Variation          Squares    Freedom      Square     F Ratio
-----------------------------------------------------
Between Effect         884        1           884        0.1
Residuals          3445721      510          6756
Total about the    3447145      511          6746
  grand average
-----------------------------------------------------

P-Value for F Ratio: 0.7177
```

If the P-value for the F-ratio is found to be less than 0.05, or in other words if the chance of achieving the calculated effect is due to random noise and not the effect being measured is less than 5%, the value is said to be statistically significant. Table 4.6 shows the P-values obtained for each of the effects, in order of importance.

Desig.	Effect	P-value	Desig.	Effect	P-value
D	-87.9	0.0000	CE	6.6	0.5048
A	48.1	0.0000	F	5.6	0.5888
B	-43.3	0.0000	BG	4.8	0.6413
E	38.6	0.0001	BC	-4.3	0.6403
C	35.3	0.0005	CD	4.2	0.6215
AB	15.2	0.1334	G	3.7	0.7177
AD	-12.3	0.2204	DF	-3.1	0.7645
BE	-12.3	0.2248	EF	2.6	0.8023
BD	10.1	0.3086	CF	1.3	0.9006
AC	8.2	0.4078	BF	-1.2	0.9070
DG	-7.9	0.4435	AF	-0.3	0.9669
DE	-7.3	0.4616	EG	-0.3	0.9787
AG	-7.1	0.4884	AE	0.1	0.8594
CG	-6.8	0.5059	FG	0.1	0.9908

Table 4.6: The calculated effects with P-values, for both main and interaction effects, in order of importance.

Another useful tool to determine the effects is to plot the main effects against a standard normal on a what is known as a QQ-plot. This is done in Figure 4-2. The red line represents the distribution as it would be given a standard normal distribution, while the plus signs represent the main and combination effects. The most significant effects are labeled.

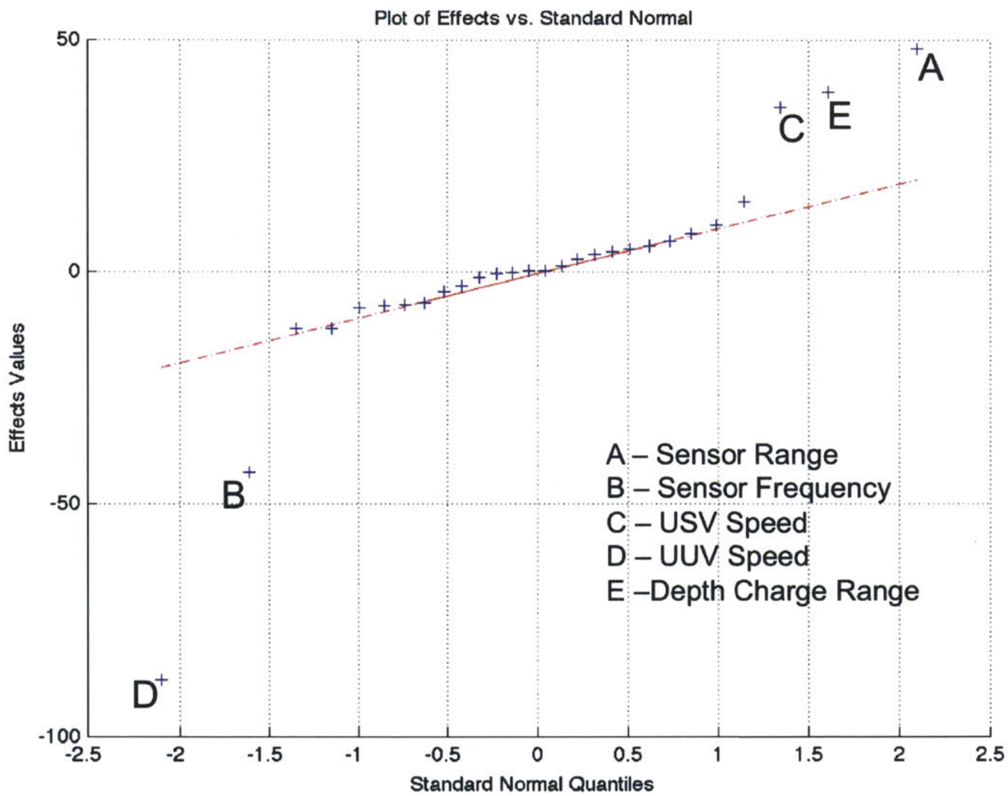


Figure 4-2: A plot of the calculated effects vs. a standard normal. Effects that fall well off the standard normal are considered the most statistically significant effects. This visual representation supplements and supports the P-value analysis.

4.6 Results Analysis

From the data presented in Section 4.5, three variables stand out from the others. The five values that have P-values well below 5% and also well lie well of the standard distribution as illustrated in Figure 4-2, in order of importance, are D, A, B, E, and C. None of the combination effects checked were found to be significant. The effects' names and values along with they're P-values are displayed in Table 4.7. The following sections will discuss each of these effects qualitatively, and provide some insight as to the importance and role played by each.

	Parameter	Effect
1	UUV Speed	-87.9
2	Sensor Range	48.1
3	Sensor Frequency	-43.3
4	Depth Charge Range	38.6
5	USV Speed	35.3

Table 4.7: The five effects calculated determined to be statistically significant.

UUV Speed

Not surprisingly, the speed of the UUV played the biggest role in determining the overall score with a total average score change of -83 for a UV speed of 0.75 vs. 1.25 m/s. The value of the effect is negative because an increase in the speed of the target results in a decrease in the score. There are several factors that cause the UUV Speed to change the score.

One observable factor affected by UUV Speed is the the UUV's time within range of the USV's sensors. More UUV Speed means less time for Archie and Betty to get current reports to switch from searching mode to tracking and prosecute modes and also less time to localize the particles to the correct position before the UUV moves out of position. So when Jackal is passing through Archie and Betty's search area, they have less time to get the requisite number of received pings to get an accurate enough track to begin pursuing.

Secondly, if Archie and Betty are in track and prosecute mode, they will have significantly more difficulty catching up to a track they know is there, particularly if they are on the low end of their speed range. Often when the UUV's speed was high and the USV's low, subjectively it seems the only likely position for the UUV to get caught is at the turn-around point at the east end of the op-box.

Sensor Range

The larger the USVs' sensor range, the better the probability they are able to find and prosecute Jackal earlier, and hence, a larger score. A greater sensor range allows the USV's a bigger window to pick up the UUV, and also much less missed reports once the USV's have current reports. Also, if the particle filter doesn't give a completely accurate result for a ping or two and the vehicles start to drive away from the UUV in confusion, the larger sensor size gives them more of an opportunity to re-localize Jackal's track before they move out of range.

Sensor Frequency

Sensor Frequency made a significant difference as well. The more pings that can be made per second, the greater the probability the USV's will be successful. The higher frequency means the particle filter will localize on the target faster, meaning the surface vehicles need to spend less time in sensor range of Jackal to both move to the track/prosecute mode and also for the particle filter to pin down the point of Jackal.

As discussed in Section 3.3.2, in order for Archie and Betty to switch to track/prosecute mode and have "current" reports, they need to have received at least 3 range reports within the last 60 seconds on Jackal. This could become a significant problem if pings do not occur at regular enough intervals. For instance, if the ping frequency was greater than once per 20 seconds, it would be impossible for one vehicle to get enough pings in the required time to switch out of searching mode. For future iterations, it would be a good idea to come up with some sort of equation that lessens the "current reports" requirement for lower ping frequencies. For this study, the current set of rules is sufficient, but not optimal, and this sort of solution could be more practical over a larger range of ping frequencies.

Depth Charge Range

A depth charge range of 25 meters vs. 15 meters was significantly different, with the larger range almost twice the distance of the smaller range. If the surface vehicles were able to track the UUV, they were much more likely to get hits (with fewer misses) with the larger range than the smaller range alternative, and fewer misses means a higher score. By the scoring metric in Section 3.2.7 each miss counts as a 5 point penalty. Table 4.8 shows the average number of misses for the scenarios run at the greater range vs. the smaller range. The smaller depth charge range results in almost 2.5 times as many misses as the larger.

DC Range	Avg. Misses
15 meters	1.496
25 meters	0.623

Table 4.8: The average number of misses for a given Depth Charge Range parameter value. A smaller range results in almost 2.5 times as many misses.

Like the ping frequency, the positional accuracy required for the drop, based on the current algorithms, did not scale with the explosion range of the depth charge. Per the current algorithm from Section 3.3.2, the vehicles need to be within 15 meters of a spot in front of the target track, for at least 10 seconds. Just as the "current" reports requirement could change based on the ping frequency, the requirements for a depth charge drop could change with the range of the depth charge. Future iterations and improvements to the solution should take into account the range of their depth charge in their algorithms, especially if being tested over a range of ping frequencies.

USV Speed

USV speed was the smallest statistically significant factor. This may be due to the fact that USV speed does not have as many advantages as some of the other parameters that have a larger effects. For example, it may allow the USV's to move into an acquired target track faster, it may also mean they move toward an incorrect target track faster, so they move away quicker the same as they move toward quicker. Also, while the USV's are in Search mode, the higher USV speed doesn't necessarily increase the amount of time Jackal will be within sensor range, as the Archie and Betty will only maneuver in circles faster, still covering the same area as they would with a slower speed. Overall however, as one might expect, USV speed is an advantage and does make a significant difference in the score.

4.7 Recommendations for Solution Improvements

There are a number of improvements that can be made to the Hunter Prey solution presented in this paper, some which have already been discussed. Many of the improvements can be made in the search algorithm. As a reminder, as described in Section 3.3.2, the vehicles enter the search mode when they do not have enough range reports to validate a "current reports" status, which is simply to follow a set of near-circular waypoints as identified in Figure 3-6. There are many alternatives that could be considered that would likely yield better results for the USV.

One idea, for example, instead of having fixed waypoint circles, the waypoint circles could be shifted east or west within the op-box as time moves on, in an attempt to always keep them on top of where the UUV might be at a given time. The position might also be based on where the last known (or assumed) position of the UUV was. Other ideas could be to use random waypoints, or weighted random waypoints based on again some method of position estimate based on time or last known position. Once these ideas are implemented, there are opportunities to build on these ideas as well.

Another significant area for improvement is optimizing the particle filter. The parameters used for the particle filter, listed in Section 2.3, first of all were determined heuristically [13], and may not be the optimal combination of parameters given the amount of computing power. Also, more particles will always make the particle filters more accurate. So an increase on computing power would give some ability to the PF to track more accurately, although for a given addition of computing power, this will bring diminishing returns.

Also, currently the vehicles track the highest weighted particle. One major problem that prevented the vehicles from being able to localize the target many times is that the highest weighted particle can jump from place to place as particles are re-sampled and re-weighted and make it difficult for the vehicles to follow and get close enough for a depth charge drop. If an algorithm were able to treat the set of particles more as a probability distribution function, or a distribution with some error of the track, there may be room for significant improvement in the tracking ability of the vehicles.

Lastly, as mentioned in some paragraphs in Section 4.6, the logic the vehicles use to enter into the track and prosecute mode, or to be close enough for a long enough time to drop a depth charge, doesn't change with the values set for the parameters such as sensor frequency or depth charge range. Writing new algorithms in to the code and doing some testing has the potential to show some improvement in the scores, specifically for this type of regression testing. This would not be as important if the parameters are fixed as they may be for the Hunter-Prey competition, however, even then, one could still do some exploration to find the optimal settings for whatever parameters are set.

4.8 The Future of Hunter-Prey

As the Hunter-Prey project will be ongoing, this section will address possible ideas to improve the problem as it's presented to a public audience, as well as ideas to make the problem more complex if solved too easily.

UUV Competitiveness

As the research and the set up stands now, most of the Hunter-Prey scenario focuses on the performance of the two USV's, Archie and Betty. This is reasonable for this research, as the tracking problem and algorithms required are significantly more challenging for the USV's and to get the solution working. However, as mentioned in Section 1.2, the goal of this project is to get competitors interested in both sides of the problem - the USV's and the UUV. While beyond the scope of this paper, in the future, this could be encouraged more, as the project moves forward.

One easy solution for this, is to allow Jackal some glimpse of the USV's. Jackal as of now has no mechanism by which to react to the USV's. That is because as the problem is designed now, the UUV has does not have any sensors that allow it to see the USV's. Equipping the UUV with its own range sensor, or even a full ability to see the USV's when they ping or drop depth charges could make the problem significantly more interesting for the UUV, which will have plenty of more options in trying to avoid getting caught. Furthermore, implementing a separate scoring system for Jackal would make the problem more interesting as well.

Sensor Arcs

There are also ways to make the problem more difficult for the USV's that done. One tool that was implemented early on the research but not used due to the added complexity to the problem, was the limiting of the range sensors only to certain arcs relative to the heading for the vehicle, so they are no longer able to see all 360 degrees around the vehicle. For instance, the vehicles may only be able to see in 90 degree arcs on the sides of the vehicle, or perhaps only a 120 degree arc on the front. These examples are shown in Figure 4-3 This would significantly change the problem and make it more interesting, making it more interesting as the vehicles would now need to manage their relative heading to the target in addition to all the other objectives it's attempting to accomplish.

Additional Vehicles

One possible solution to make the Hunter-Prey problem more challenging on both sides is to increase the number of vehicles on the water. This could involve adding more UUV's, or more USV's. While this may prove difficult to do with real water

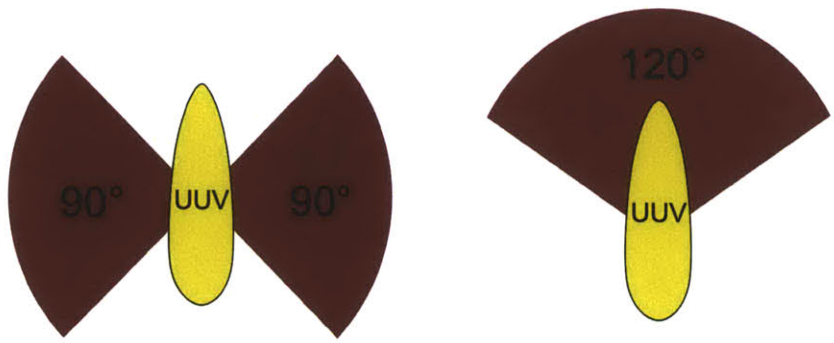


Figure 4-3: Possible arc configurations for the USV range-sensor for future iterations of the Hunter-Prey Project.

testing in the near term, it could still be run in simulation, and would also require competitors to come up with more scalable solutions for some number N vehicles, rather than just one or two. Putting more UUV's in the water would be a challenge as well, both for the particle filter side of things, which is now trying to track more than one target, and for the decision making on board the vehicles.

Other Suggestions

There are other ideas to increase the complexity of the problem as well, listed below:

1. Make the problem into a 3D tracking problem.

2. Change the sensor detection probability to be based on sound levels and attenuation in the water rather than strictly based on distance. Or make the vehicles use real sensors, in the water.

3. Change the problem to be longer term - over hours or days, in deeper, open water.

Chapter 5

Conclusions

Looking back at the original goals set forth for this thesis research in Section 1.3, the results presented in this paper completed what it set forth to do. The following paragraphs delineate each of the goals defined at the beginning of the paper, and how those goals were met.

1. *Define the rules, guidelines, and set-up for the Hunter-Prey scenario.* This was completed in Sections 3.1 and 3.2.1. Here, the vehicles, the op-box, the mission parameters, and the rules for the depth charges and range sensors and how they are used are all defined and explained. Communications and how the mission is scored is also defined and discussed in these sections.

2. *Develop a "straw-man" or basic solution to the problem.* This is addressed in Section 3.3, which lays out a full, non-optimized solution for both the UUV and the USV's to compete against one another in the Hunter-Prey scenario. The range-only sensing problem was explained and solved in Chapter 2, using a particle filter based on the pParticle application developed by Andrew Privette master's thesis [13], and the logic solutions developed worked adequately and were sufficient for testing. In addition, a number of different ideas for improvements on the current solution were discussed.

3. *Run this solution through regression testing to determine which factors significantly affect the problem, and by how much.* Chapter 4 presents an in-depth

testing process using regression analysis. The effects of the parameters, combinations of the parameters, and whether those effects were statistically significant, were analyzed and discussed in detail. The results give a picture of the most important effects, and quantitatively determine how important certain parameters are relative to others. Based on the variable ranges used, the speed of the UUV turned out to be by far the largest effect on the resulting mission score.

4. *Discuss the ways to move forward with the project as it moves toward becoming an open competition.* Future improvements as well as possible modifications to the Hunter-Prey project were discussed in Chapter 4. How the nominal or "straw-man" solution can be improved is discussed in Section 4.7, particularly with regards to the searching algorithm and particle filter. Also, Section 4.8 talks about a number of different ways the Hunter-Prey can move forward, including changing the op-area, making the problem more interesting for the UUV, and new possibilities for the environment, vehicles, and the sensors being used.

Overall, this research provides a solid starting framework for the Hunter-Prey project. The problems, solutions, and testing work provided here create an not only an effective start from which to work for this specific problem, but for future work in similar problems, in an effort to advance the field of autonomy and making it more available and attractive to a wider, more diverse audience.

Bibliography

[1] Michael R. Benjamin and Henrik Schmidt. Unmanned Marine Vehicle Autonomy, Sensing, and Communications. University Lecture, 2012.

[2] Michael R. Benjamin, Henrik Schmidt, Paul M. Newman, and John J. Leonard. Nested Autonomy for Unmanned Marine Vehicles with MOOS-IvP. *Journal of Field Robotics*, 27(6):834–875, November/December 2010.

[3] Michael R. Benjamin, Henrik Schmidt, Paul M. Newman, and John J. Leonard. An Overview of MOOS-IvP and a Users Guide to the IvP Helm - Release 13.5. May 2013.

[4] Michael R. Benjamin, Henrik Schmidt, Paul M. Newman, and John J. Leonard. MOOS-IvP Autonomy Tools Users Manual - Release 13.5. May 2013.

[5] Bluefin Robotics Corporation. Bluefin Robotics. `http://www.bluefinrobotics.com/products/bluefin-9/`, 2013.

[6] Clearpath Robotics Corporation. Clearpath Robotics. `http://www.bluefinrobotics.com/products/bluefin-9/`, March 2013.

[7] Donald P. Eickstedt and Michael R. Benjamin. Cooperative Target Tracking in a Distributed Autonomous Sensor Network. Technical report, Massachusetts Institute of Technology, 2006.

[8] Huang G.P. and Roumeliotis S.I. An Analytically-Guided Sampling Scheme for Particle Filtering and its Application to Target Tracking, 2011.

[9] Guoquan P. Huang and Stergios I. Roumeliotis. Analytically-Guided-Sampling Particle Filter Applied to Range-only Target Tracking. Technical report, Massachusetts Institute of Technology, 2013.

[10] Evan A. KARLIK. Remote Depth Survey of the Charles River Basin, 2007.

[11] Simon Maskell and Neil Gordon. A Tutorial on Particle Filters for On-line Nonlinear/Non-Gaussian Bayesian Tracking, September 2001.

[12] Samuel Prentice and Nicholas Roy. The Belief Roadmap: Efficient Planning in Belief Space by Factoring the Covariance. *The International Journal of Robotics Research*, 28(11-12):1446–1465, Oct 2009.

[13] Andrew J. Privette. Autonomous adaptation and collaboration of unmanned vehicles for tracking submerged contacts. Master's thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, June 2012.

[14] Gao Rui and Mandar Chitre. Cooperative positioning using range-only measurements between two AUVs. Technical report, Tropical Marine Science Institute at the National University of Singapore, 2010.

[15] Taek Lyul Song. Observability of Target Tracking with Range-Only Measurements. *IEEE Journal of Ocean Engineering*, 24(3), July 1999.

[16] Costas J. Spanos and Gary S. May. *Semiconductor Manufacturing and Process Control*. John Wiley and Sons, 2006.

Appendix A

Regression Testing Data

Table A.1: Parameter Values by Simulation

Expt. Number	Sensor Range	Sensor Freq	USV Speed	UUV Speed	Depth Charge Range	Depth Charge Amount	DC Refill Time
1	25	5	1.5	0.75	15	2	30
2	25	5	1.5	0.75	15	2	120
3	25	5	1.5	0.75	15	6	30
4	25	5	1.5	0.75	15	6	120
5	25	5	1.5	0.75	25	2	30
6	25	5	1.5	0.75	25	2	120
7	25	5	1.5	0.75	25	6	30
8	25	5	1.5	0.75	25	6	120
9	25	5	1.5	1.25	15	2	30
10	25	5	1.5	1.25	15	2	120
11	25	5	1.5	1.25	15	6	30
12	25	5	1.5	1.25	15	6	120
13	25	5	1.5	1.25	25	2	30
14	25	5	1.5	1.25	25	2	120
15	25	5	1.5	1.25	25	6	30
16	25	5	1.5	1.25	25	6	120
17	25	5	2	0.75	15	2	30
18	25	5	2	0.75	15	2	120
19	25	5	2	0.75	15	6	30
20	25	5	2	0.75	15	6	120
21	25	5	2	0.75	25	2	30
22	25	5	2	0.75	25	2	120

Expt. Number	Sensor Range	Sensor Freq	USV Speed	UUV Speed	Depth Charge Range	Depth Charge Amount	DC Refill Time
23	25	5	2	0.75	25	6	30
24	25	5	2	0.75	25	6	120
25	25	5	2	1.25	15	2	30
26	25	5	2	1.25	15	2	120
27	25	5	2	1.25	15	6	30
28	25	5	2	1.25	15	6	120
29	25	5	2	1.25	25	2	30
30	25	5	2	1.25	25	2	120
31	25	5	2	1.25	25	6	30
32	25	5	2	1.25	25	6	120
33	25	15	1.5	0.75	15	2	30
34	25	15	1.5	0.75	15	2	120
35	25	15	1.5	0.75	15	6	30
36	25	15	1.5	0.75	15	6	120
37	25	15	1.5	0.75	25	2	30
38	25	15	1.5	0.75	25	2	120
39	25	15	1.5	0.75	25	6	30
40	25	15	1.5	0.75	25	6	120
41	25	15	1.5	1.25	15	2	30
42	25	15	1.5	1.25	15	2	120
43	25	15	1.5	1.25	15	6	30
44	25	15	1.5	1.25	15	6	120
45	25	15	1.5	1.25	25	2	30
46	25	15	1.5	1.25	25	2	120
47	25	15	1.5	1.25	25	6	30
48	25	15	1.5	1.25	25	6	120
49	25	15	2	0.75	15	2	30
50	25	15	2	0.75	15	2	120
51	25	15	2	0.75	15	6	30
52	25	15	2	0.75	15	6	120
53	25	15	2	0.75	25	2	30
54	25	15	2	0.75	25	2	120
55	25	15	2	0.75	25	6	30
56	25	15	2	0.75	25	6	120
57	25	15	2	1.25	15	2	30
58	25	15	2	1.25	15	2	120

Expt. Number	Sensor Range	Sensor Freq	USV Speed	UUV Speed	Depth Charge Range	Depth Charge Amount	DC Refill Time
59	25	15	2	1.25	15	6	30
60	25	15	2	1.25	15	6	120
61	25	15	2	1.25	25	2	30
62	25	15	2	1.25	25	2	120
63	25	15	2	1.25	25	6	30
64	25	15	2	1.25	25	6	120
65	75	5	1.5	0.75	15	2	30
66	75	5	1.5	0.75	15	2	120
67	75	5	1.5	0.75	15	6	30
68	75	5	1.5	0.75	15	6	120
69	75	5	1.5	0.75	25	2	30
70	75	5	1.5	0.75	25	2	120
71	75	5	1.5	0.75	25	6	30
72	75	5	1.5	0.75	25	6	120
73	75	5	1.5	1.25	15	2	30
74	75	5	1.5	1.25	15	2	120
75	75	5	1.5	1.25	15	6	30
76	75	5	1.5	1.25	15	6	120
77	75	5	1.5	1.25	25	2	30
78	75	5	1.5	1.25	25	2	120
79	75	5	1.5	1.25	25	6	30
80	75	5	1.5	1.25	25	6	120
81	75	5	2	0.75	15	2	30
82	75	5	2	0.75	15	2	120
83	75	5	2	0.75	15	6	30
84	75	5	2	0.75	15	6	120
85	75	5	2	0.75	25	2	30
86	75	5	2	0.75	25	2	120
87	75	5	2	0.75	25	6	30
88	75	5	2	0.75	25	6	120
89	75	5	2	1.25	15	2	30
90	75	5	2	1.25	15	2	120
91	75	5	2	1.25	15	6	30
92	75	5	2	1.25	15	6	120
93	75	5	2	1.25	25	2	30
94	75	5	2	1.25	25	2	120

Expt. Number	Sensor Range	Sensor Freq	USV Speed	UUV Speed	Depth Charge Range	Depth Charge Amount	DC Refill Time
95	75	5	2	1.25	25	6	30
96	75	5	2	1.25	25	6	120
97	75	15	1.5	0.75	15	2	30
98	75	15	1.5	0.75	15	2	120
99	75	15	1.5	0.75	15	6	30
100	75	15	1.5	0.75	15	6	120
101	75	15	1.5	0.75	25	2	30
102	75	15	1.5	0.75	25	2	120
103	75	15	1.5	0.75	25	6	30
104	75	15	1.5	0.75	25	6	120
105	75	15	1.5	1.25	15	2	30
106	75	15	1.5	1.25	15	2	120
107	75	15	1.5	1.25	15	6	30
108	75	15	1.5	1.25	15	6	120
109	75	15	1.5	1.25	25	2	30
110	75	15	1.5	1.25	25	2	120
111	75	15	1.5	1.25	25	6	30
112	75	15	1.5	1.25	25	6	120
113	75	15	2	0.75	15	2	30
114	75	15	2	0.75	15	2	120
115	75	15	2	0.75	15	6	30
116	75	15	2	0.75	15	6	120
117	75	15	2	0.75	25	2	30
118	75	15	2	0.75	25	2	120
119	75	15	2	0.75	25	6	30
120	75	15	2	0.75	25	6	120
121	75	15	2	1.25	15	2	30
122	75	15	2	1.25	15	2	120
123	75	15	2	1.25	15	6	30
124	75	15	2	1.25	15	6	120
125	75	15	2	1.25	25	2	30
126	75	15	2	1.25	25	2	120
127	75	15	2	1.25	25	6	30
128	75	15	2	1.25	25	6	120

Appendix B

Regression Results

Table B.1: Results from Regressions 1 and 2

	Regression 1				Regression 2			
Expt	Hit?	Time	Misses	Score	Hit?	Time	Misses	Score
1	No	1265.1	2	0	No	1559.6	4	0
2	Yes	1188.6	2	86	Yes	1125.9	5	77
3	Yes	379.4	2	167	No	1253.0	4	0
4	Yes	1440.8	7	36	Yes	1371.2	5	53
5	Yes	436.0	1	166	Yes	532.5	2	152
6	Yes	436.0	0	171	Yes	557.5	0	159
7	Yes	249.2	0	190	Yes	331.9	0	182
8	Yes	397.2	0	175	Yes	345.8	1	175
9	No	999.9	2	0	No	927.9	2	0
10	No	1040.8	2	0	No	918.7	2	0
11	No	822.2	3	0	No	871.9	3	0
12	No	839.5	0	0	No	996.7	5	0
13	No	741.7	1	0	Yes	725.2	1	137
14	No	1058.7	3	0	No	812.2	0	0
15	No	860.3	0	0	No	852.1	6	0
16	No	800.9	0	0	Yes	900.4	1	120
17	No	1361.7	10	0	Yes	490.7	3	151
18	Yes	408.3	3	159	Yes	362.6	0	179
19	Yes	411.9	3	159	Yes	601.2	5	130
20	Yes	879.6	9	82	Yes	338.1	2	171
21	Yes	426.4	0	172	Yes	358.2	0	179
22	Yes	479.3	2	157	Yes	318.0	0	183
23	Yes	236.1	0	191	Yes	353.8	0	180

Expt	Regression 1				Regression 2			
	Hit?	Time	Misses	Score	Hit?	Time	Misses	Score
24	Yes	376.5	0	177	Yes	247.6	0	190
25	No	841.8	6	0	No	768.5	4	0
26	No	982.4	3	0	No	898.4	1	0
27	No	1076.1	0	0	No	1056.2	2	0
28	No	1021.1	5	0	No	825.0	6	0
29	Yes	374.0	0	178	No	772.2	0	0
30	No	869.8	4	0	Yes	890.4	3	111
31	No	873.4	4	0	No	850.6	2	0
32	Yes	229.8	4	172	Yes	367.2	0	178
33	No	1262.0	2	0	No	1706.6	5	0
34	No	1479.6	2	0	Yes	1058.8	3	94
35	No	1401.5	15	0	No	1736.3	0	0
36	Yes	323.1	0	183	Yes	1016.8	3	98
37	No	1583.9	1	0	No	1443.4	2	0
38	Yes	545.5	1	155	Yes	323.1	0	183
39	Yes	1174.5	4	78	No	1402.3	4	0
40	No	1233.5	6	0	No	1297.7	2	0
41	No	887.4	0	0	No	875.4	2	0
42	No	1024.1	1	0	No	832.5	1	0
43	No	890.4	4	0	No	783.4	1	0
44	No	861.6	5	0	No	807.0	0	0
45	No	838.7	3	0	No	851.1	0	0
46	No	949.9	1	0	No	849.8	2	0
47	No	836.3	3	0	No	850.2	1	0
48	No	883.1	0	0	No	1007.0	4	0
49	No	1253.7	4	0	No	1424.0	7	0
50	Yes	273.8	0	188	No	1328.6	2	0
51	No	2098.8	10	0	No	1328.6	9	0
52	Yes	1261.0	9	44	No	1500.3	5	0
53	No	1703.2	2	0	Yes	271.7	0	188
54	No	1452.1	4	0	Yes	334.9	1	177
55	No	1425.9	1	0	Yes	431.5	3	157
56	Yes	255.1	0	189	Yes	379.6	5	152
57	No	975.9	0	0	No	850.9	2	0
58	No	842.3	0	0	No	898.5	1	0

Expt	Regression 1				Regression 2			
	Hit?	Time	Misses	Score	Hit?	Time	Misses	Score
59	No	870.9	2	0	No	985.8	1	0
60	No	1129.6	0	0	No	881.6	9	0
61	No	1002.2	1	0	No	790.9	4	0
62	No	887.9	4	0	No	964.7	3	0
63	No	769.9	1	0	No	912.6	1	0
64	No	791.0	0	0	No	785.3	2	0
65	Yes	1251.5	7	55	Yes	486.8	3	151
66	Yes	490.3	3	151	Yes	891.0	2	116
67	Yes	1029.9	16	32	Yes	343.0	2	171
68	Yes	460.1	7	134	Yes	497.6	4	145
69	Yes	463.7	0	169	Yes	275.4	0	187
70	Yes	384.2	1	172	Yes	364.8	0	179
71	Yes	272.6	0	188	Yes	273.7	0	188
72	Yes	276.4	0	187	Yes	379.3	1	172
73	No	907.9	1	0	No	1010.8	2	0
74	No	753.9	2	0	No	841.6	1	0
75	No	839.2	1	0	No	774.8	0	0
76	No	804.0	1	0	No	843.7	1	0
77	Yes	572.5	2	148	No	821.9	4	0
78	No	799.7	2	0	No	828.6	2	0
79	Yes	786.0	3	121	No	936.8	0	0
80	Yes	647.1	1	145	No	960.8	0	0
81	Yes	251.8	1	185	Yes	416.3	0	173
82	Yes	1267.7	9	43	Yes	547.7	3	145
83	Yes	247.1	1	185	Yes	328.6	2	172
84	Yes	441.8	9	126	Yes	824.0	7	98
85	Yes	232.8	0	192	Yes	349.7	0	180
86	Yes	314.3	0	184	Yes	261.8	0	189
87	Yes	282.2	0	187	Yes	253.8	0	190
88	Yes	317.6	0	183	Yes	224.5	0	193
89	Yes	289.0	1	181	No	820.7	1	0
90	Yes	382.9	2	167	No	837.8	2	0
91	Yes	427.7	3	157	No	898.5	1	0
92	Yes	832.4	6	102	Yes	653.2	1	145
93	Yes	294.6	1	181	Yes	316.7	2	173

	Regression 1				Regression 2			
Expt	Hit?	Time	Misses	Score	Hit?	Time	Misses	Score
94	No	867.8	3	0	No	829.0	3	0
95	Yes	385.2	0	176	No	929.3	1	0
96	Yes	257.1	0	189	Yes	466.8	1	163
97	Yes	410.6	1	169	Yes	561.9	1	154
98	Yes	437.1	0	171	Yes	1411.9	6	44
99	Yes	1224.1	14	23	Yes	596.6	3	140
100	Yes	699.2	5	120	No	1575.3	7	0
101	Yes	607.0	3	139	Yes	318.9	0	183
102	Yes	483.7	2	157	Yes	633.2	0	152
103	Yes	339.8	1	176	Yes	480.6	1	162
104	Yes	389.9	1	171	Yes	546.5	1	155
105	No	762.4	1	0	No	901.6	0	0
106	No	903.9	1	0	No	787.6	2	0
107	No	814.7	3	0	No	867.8	0	0
108	No	764.5	2	0	No	897.7	0	0
109	No	838.8	0	0	No	984.1	2	0
110	No	840.8	1	0	No	957.9	3	0
111	No	787.3	2	0	No	899.7	1	0
112	No	795.4	1	0	No	910.8	1	0
113	Yes	427.7	2	162	Yes	294.6	1	181
114	No	1487.4	8	0	Yes	371.9	3	163
115	Yes	367.3	5	153	Yes	862.5	12	69
116	Yes	413.8	3	159	Yes	942.4	13	56
117	Yes	1445.8	3	50	Yes	206.3	0	194
118	Yes	359.4	1	174	Yes	269.4	0	188
119	Yes	309.3	0	184	Yes	421.7	3	158
120	Yes	307.7	5	159	Yes	394.4	0	176
121	No	797.4	1	0	Yes	456.7	1	164
122	Yes	420.0	1	168	No	955.1	0	0
123	No	876.0	3	0	Yes	770.2	3	123
124	Yes	768.5	4	118	No	784.7	2	0
125	No	892.4	1	0	Yes	484.9	0	167
126	Yes	585.3	2	146	No	859.7	2	0
127	No	971.8	1	0	Yes	369.7	0	178
128	Yes	425.2	2	162	No	894.9	5	0

Table B.2: Results from Regressions 3 and 4

	Regression 3				Regression 4			
Expt	**Hit?**	**Time**	**Misses**	**Score**	**Hit?**	**Time**	**Misses**	**Score**
1	Yes	1330.8	5	57	Yes	947.3	2	110
2	Yes	1487.5	5	41	Yes	1166.7	5	73
3	Yes	437.9	1	166	Yes	1028.2	9	67
4	No	1584.9	7	0	Yes	445.9	3	155
5	Yes	507.7	1	159	Yes	568.3	0	158
6	Yes	1264.8	1	84	Yes	330.8	0	182
7	Yes	303.9	0	185	Yes	387.4	0	176
8	Yes	296.0	0	185	Yes	979.0	5	92
9	No	1091.4	2	0	No	786.5	0	0
10	No	917.2	0	0	No	868.3	2	0
11	No	904.4	7	0	No	789.1	1	0
12	No	809.2	0	0	No	874.4	5	0
13	No	799.2	2	0	No	835.7	0	0
14	No	833.9	0	0	Yes	607.1	2	144
15	No	930.5	1	0	No	867.8	5	0
16	No	801.1	5	0	No	791.7	2	0
17	Yes	347.6	1	175	Yes	374.5	0	178
18	Yes	374.5	1	173	Yes	343.0	1	176
19	Yes	356.7	2	169	Yes	439.4	2	161
20	Yes	430.0	2	162	Yes	352.7	0	180
21	Yes	266.0	0	188	Yes	332.6	1	177
22	Yes	458.0	2	159	Yes	386.5	1	171
23	Yes	326.7	0	182	Yes	357.9	1	174

Expt	Regression 3 Hit?	Time	Misses	Score	Regression 4 Hit?	Time	Misses	Score
24	Yes	234.9	0	192	Yes	374.2	0	178
25	No	847.3	5	0	No	850.3	4	0
26	Yes	719.0	0	143	No	846.3	3	0
27	No	1087.4	6	0	No	853.3	2	0
28	No	916.8	5	0	No	905.9	5	0
29	Yes	605.8	0	154	Yes	309.8	0	184
30	Yes	833.0	1	127	No	1019.1	0	0
31	Yes	464.9	1	164	Yes	344.2	1	176
32	No	931.3	1	0	No	823.1	1	0
33	No	1722.1	3	0	No	1314.2	0	0
34	No	1703.4	2	0	No	1586.4	4	0
35	No	1467.1	2	0	No	1299.4	1	0
36	No	1520.1	2	0	No	1372.1	3	0
37	No	1509.8	2	0	No	1468.6	3	0
38	Yes	952.8	0	120	No	1386.2	2	0
39	No	1268.5	3	0	Yes	1429.7	7	37
40	No	1444.7	4	0	No	1251.2	3	0
41	No	917.1	2	0	No	884.2	1	0
42	No	781.2	5	0	No	930.0	0	0
43	No	883.3	3	0	No	909.4	1	0
44	No	775.8	0	0	No	756.9	1	0
45	No	889.1	0	0	No	799.4	1	0
46	No	803.2	1	0	No	1038.6	0	0
47	No	922.1	4	0	No	828.0	0	0
48	Yes	749.7	3	125	No	862.9	2	0
49	No	1349.4	5	0	No	1437.0	4	0
50	No	1369.8	6	0	No	1397.5	6	0
51	Yes	1190.1	12	36	Yes	249.1	0	190
52	No	1449.5	4	0	No	1370.1	5	0
53	No	1331.3	2	0	No	1594.4	4	0
54	No	1318.1	4	0	Yes	205.8	0	194
55	Yes	233.9	0	192	Yes	1013.7	2	104
56	Yes	298.0	0	185	Yes	255.9	0	189
57	No	785.2	1	0	No	923.8	1	0
58	No	871.8	1	0	No	821.7	0	0

Expt	Regression 3				Regression 4			
	Hit?	Time	Misses	Score	Hit?	Time	Misses	Score
59	No	773.1	1	0	No	949.3	2	0
60	No	816.0	4	0	No	761.8	5	0
61	No	788.8	0	0	No	1061.2	0	0
62	No	823.0	2	0	No	904.9	0	0
63	No	873.7	3	0	No	822.6	2	0
64	No	754.1	2	0	No	956.0	7	0
65	Yes	298.0	0	185	Yes	490.4	3	151
66	Yes	1061.9	5	84	Yes	273.3	0	188
67	Yes	269.7	1	183	Yes	908.1	12	64
68	Yes	331.5	2	172	Yes	279.5	1	182
69	Yes	277.9	0	187	Yes	339.0	1	176
70	Yes	343.4	0	181	Yes	278.7	0	187
71	Yes	276.7	0	187	Yes	345.3	0	180
72	Yes	266.7	0	188	Yes	433.1	0	172
73	No	870.3	1	0	No	793.0	1	0
74	No	896.5	0	0	No	800.3	0	0
75	No	772.0	0	0	No	871.0	0	0
76	No	957.8	1	0	Yes	454.4	2	160
77	No	917.4	2	0	No	797.2	2	0
78	No	876.1	0	0	Yes	556.9	2	149
79	Yes	741.4	2	131	No	931.8	0	0
80	Yes	659.7	2	139	No	763.3	1	0
81	Yes	1213.2	11	39	Yes	1185.6	7	61
82	Yes	308.9	0	184	Yes	455.4	0	169
83	Yes	376.0	0	177	Yes	397.2	0	175
84	Yes	282.5	1	182	Yes	289.4	0	186
85	Yes	266.3	0	188	Yes	232.0	0	192
86	Yes	218.3	0	193	Yes	271.8	1	183
87	Yes	345.2	0	180	Yes	273.8	0	188
88	Yes	276.7	1	182	Yes	357.4	0	179
89	Yes	694.5	5	121	Yes	397.1	1	170
90	No	845.7	2	0	No	908.8	1	0
91	Yes	518.0	2	153	No	917.8	8	0
92	No	916.4	4	0	No	887.3	3	0
93	Yes	302.3	0	185	Yes	382.6	0	177

Expt	Regression 3 Hit?	Time	Misses	Score	Regression 4 Hit?	Time	Misses	Score
94	Yes	350	0	180	Yes	344.8	0	181
95	Yes	295	1	180	Yes	616.3	0	153
96	Yes	346	2	170	Yes	333.8	0	182
97	Yes	1202	4	75	Yes	817.3	4	113
98	Yes	783	0	137	Yes	415.6	1	168
99	Yes	703	4	125	Yes	1703.3	19	-50
100	Yes	589	2	146	Yes	526.3	4	142
101	Yes	636	0	151	Yes	539.4	2	151
102	Yes	655	0	150	Yes	654.7	0	150
103	Yes	468	2	158	Yes	510.0	0	164
104	Yes	399	0	175	Yes	864.2	2	119
105	No	795	1	0	No	840.4	0	0
106	No	833	0	0	No	0.0	0	130
107	Yes	417	0	173	No	849.1	0	0
108	No	1035	1	0	No	915.1	1	0
109	No	765	0	0	No	871.4	0	0
110	No	858	0	0	No	806.3	3	0
111	No	0	0	126	No	0.0	0	0
112	No	874	1	0	Yes	881.4	2	117
113	Yes	1122	8	63	Yes	351.8	0	180
114	Yes	367	1	173	Yes	438.6	3	156
115	Yes	252	1	185	Yes	432.9	2	162
116	Yes	341	2	171	Yes	504.7	2	155
117	Yes	385	2	167	Yes	361.7	0	179
118	Yes	317	1	178	Yes	316.4	1	178
119	Yes	251	1	185	Yes	299.9	0	185
120	Yes	399	0	175	Yes	270.7	0	188
121	No	803	5	0	Yes	897.3	4	105
122	Yes	514	2	154	No	791.0	2	0
123	No	961	6	0	Yes	508.1	3	149
124	No	803	2	0	No	840.9	1	0
125	Yes	494	0	166	No	960.6	0	0
126	No	0	0	132	No	869.2	2	0
127	No	802	4	0	Yes	379.7	0	177
128	Yes	761	2	129	No	755.6	3	0

Appendix C

Interaction ANOVA Tables

Table C.1: The full ANOVA tables for the 21 interaction effects. The values for these numbers are calculated via the methods presented in Section 4.4

Sensor Range with Sensor Freq

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	15107	1	15107	2.3
Residuals	3409390	510	6685	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.1334

Sensor Range with USV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	4635	1	4635	0.7
Residuals	3443876	510	6753	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.4078

Sensor Range with UUV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	10147	1	10147	1.5
Residuals	3437234	510	6740	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.2204

Sensor Range with DC Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	212	1	212	0.0
Residuals	3447635	510	6760	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.8594

Sensor Range with DC Amount

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	12	1	12	0.0
Residuals	3447150	510	6759	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.9669

Sensor Range with DC Refill Time

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	3245	1	3245	0.5
Residuals	3442221	510	6749	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.4884

Sensor Freq with USV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	1476	1	1476	0.2
Residuals	3442906	510	6751	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.6403

Sensor Freq with UUV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	6981	1	6981	1.0
Residuals	3428264	510	6722	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.3086

Sensor Freq with DC Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	9912	1	9912	1.5
Residuals	3422117	510	6710	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.2248

Sensor Freq with DC Amount

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	92	1	92	0.0
Residuals	3447020	510	6759	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.9070

Sensor Freq with DC Refill Time

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	1467	1	1467	0.2
Residuals	3443157	510	6751	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.6413

USV Speed with UUV Speed

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	1648	1	1648	0.2
Residuals	3442917	510	6751	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.6215

USV Speed with DC Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	3009	1	3009	0.4
Residuals	3445218	510	6755	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.5048

USV Speed with DC Amount

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	106	1	106	0.0
Residuals	3446902	510	6759	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.9006

USV Speed with DC Refill Time

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	2992	1	2992	0.4
Residuals	3442665	510	6750	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.5059

UUV Speed with DC Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	3659	1	3659	0.5
Residuals	3437066	510	6739	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.4616

UUV Speed with DC Range

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	3659	1	3659	0.5
Residuals	3437066	510	6739	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.4616

UUV Speed with DC Amount

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	607	1	607	0.1
Residuals	3446061	510	6757	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.7645

DC Range with DC Amount

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	424	1	424	0.1
Residuals	3446218	510	6757	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.8023

DC Range with DC Refill Time

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	5	1	5	0.0
Residuals	3447196	510	6759	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.9787

DC Amount with DC Refill Time

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	F Ratio
Between Effect	1	1	1	0.0
Residuals	3447154	510	6759	
Total about the grand average	3447145	511	6746	

P-Value for F Ratio: 0.9908