2008

A Multi-objective No-regret Decision Making Model With Bayesian Learning For Autonomous Unmanned Systems

Matthew Howard
University of Central Florida

A MULTI-OBJECTIVE NO-REGRET DECISION MAKING MODEL WITH BAYESIAN LEARNING FOR AUTONOMOUS UNMANNED SYSTEMS

by

MATTHEW DAVID HOWARD

B.S. University of Florida, 2005

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in the School of Electrical Engineering and Computer Science in the College of Engineering and Computer Science at the University of Central Florida Orlando, Florida

Fall Term

2008

ABSTRACT

The development of a multi-objective decision making and learning model for the use in unmanned systems is the focus of this project. Starting with traditional game theory and psychological learning theories developed in the past, a new model for machine learning is developed. This model incorporates a no-regret decision making model with a Bayesian learning process which has the ability to adapt to errors found in preconceived costs associated with each objective. This learning ability is what sets this model apart from many others.

By creating a model based on previously developed human learning models, hundreds of years of experience in these fields can be applied to the recently developing field of machine learning. This also allows for operators to more comfortably adapt to the machine's learning process in order to better understand how to take advantage of its features.

One of the main purposes of this system is to incorporate multiple objectives into a decision making process. This feature can better allow its users to clearly define objectives and prioritize these objectives allowing the system to calculate the best approach for completing the mission. For instance, if an operator is given objectives such as obstacle avoidance, safety, and limiting resource usage, the operator would traditionally be required to decide how to meet all of these objectives. The use of a multi-objective decision making process such as the one designed in this project, allows the operator to input the objectives and their priorities and receive an output of the calculated optimal compromise.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 INTRODUCTION

One of the most important fields in the autonomy of mobile robots is the decision making process. Many of the advances in this field have come from adaptations from the learning theory associated with the fields of psychology and game theory. One such theory is Reinforcement Learning which states that the likelihood of an action being played in the future is a direct result of the action's payoff in the past [15]. This is one of the most basic theories in that it does not take into account any information about the opponent nor does it take into consideration any regret from not having played other actions. No-regret Learning is similar to Reinforcement Learning, however, No-regret Learning [7, 15] takes into consideration the payoffs that would have been incurred if each of the other actions was taken in the previous play. Another learning theory, presented by George W. Brown, in 1951 called Fictitious Play [3], considers a hypothesis on the opponent's course of action and plays accordingly. Another learning theory was presented by Thomas Bayes in 1763 [13, 4, 1]. Bayesian theory is a probability model used to estimate the chances of an action occurring based on all prior assumptions and histories. These, among many other learning theories are the backbone of the work toward the autonomy of robots.

There has been work toward converting these theories into algorithms that can be applied to machine learning. One example is Geoffrey Gordon's Lagrangian Hedging algorithms which incorporates No-regret Learning [7]. Also, in Vijaykumar Gullapalli's dissertation [8], Reinforcement Learning is adapted to work as a learning tool for control systems both with known and unknown action spaces. Another example is Tobias Karlsson's Masters Thesis where Bayesian theory is used to aid in the navigation of submersible vehicles [9].

In the proposed platform, shown in Figure 1, we will create a Multi-Objective Decision Making and Learning Model (DMLM) that will take, as inputs, a set of paths from the Finite State Model (FSM) and determine which path will yield the best results based on the set of objectives derived from information found in the Mission Planner (MP). The determination

of which path to traverse will be based on a No-Regret Decision Making Model that will assign a distribution of weights to these objectives. These weights will initially be determined by priorities set in the Mission Planner and updated by a Bayesian Learning Model based on the accuracy of the objectives given for each path.

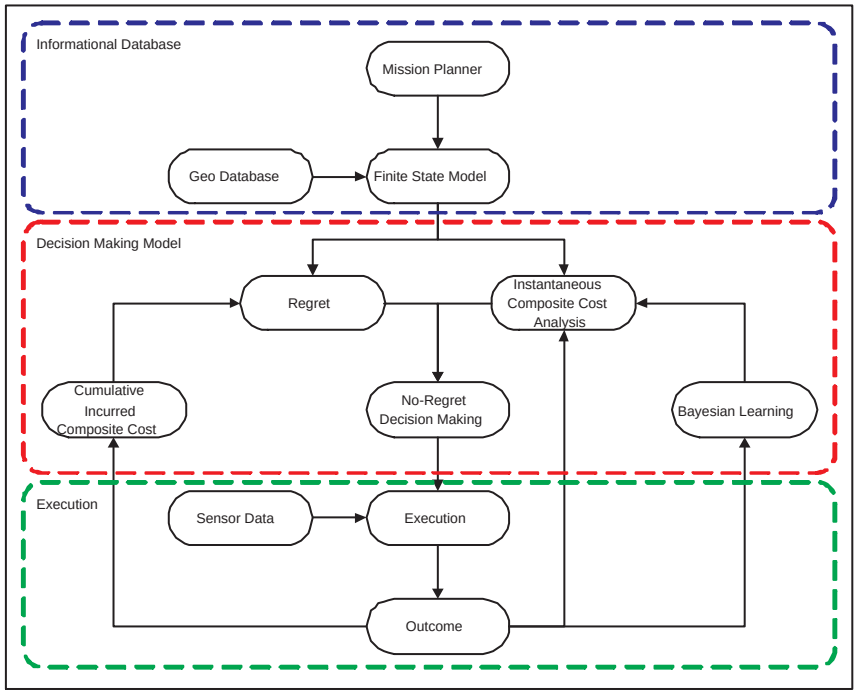


Figure 1: Platform

The set of paths that the DMLM can choose from is located in the Finite State Model (FSM) of the platform. The paths that populate the FSM will be derived from standard waypoint based path-planning algorithms. Also provided in the FSM is a set of objectives that characterize each of the proposed paths. All of the information needed by the path planner, as well as the additional information for each path, is located in the Geo Database (GD).

Once the FSM is populated, it is the job of the DMLM to iteratively determine the optimal path for the vehicle to traverse. These iterations will be performed at every waypoint where new information will be analyzed and applied to future decisions. The decision making is based on the regrets for not having chosen each of the other paths in the action space

and the estimated costs of each path in the future. Combining these two, the choice of the optimal path is determined by selecting the path with the least cost from the set of paths available to the vehicle. This is discussed in further detail in Section 2.2.

Once the DMLM has determined the optimal path, the execution of the decision is performed and the outcome is collected. The outcome and sensor data are returned to the DMLM in order to assess how accurate the expected values were for each objective. Based on this error between the expected and actual values for the objectives, the distribution of weights assigned to the objectives is adjusted according to a Bayesian Learning scheme, which is described in Section 2.3.

1.1 No-Regret Learning (Game Theory)

The theory of No-Regret learning consists of a style of play where not only the payoff for the action played is considered, but also the lost payoff if the player had played any of the other actions [15, 7]. Given a two person game between Player and Opponent. Each of the two players is allowed a finite set of actions from which they are allowed to choose from during each play of the game. Let $p \in P$ be the actions played by the Player and $o \in O$ be the actions played by Opponent.

At the end of one round of play, Player chose action p and Opponent chose action o. It is assumed that Opponent's actions are observable. Let us now define the payoff for Player to be $u(p_t, o_t)$ where p_t and o_t represent the actions chosen by Player and Opponent at time t, respectively. In order to determine the regret from Player choosing action p_t when Opponent played action o_t the following equation is used

$$r_p(t) = \sum_{i=1}^{t} u(p, o_i) - \sum_{k=1}^{t} i(p_k, o_k). \tag{1}$$

This equation states that the regret from not having played action p is the sum of the payoffs

that would have occurred if action p was played minus the actual incurred payoff up to t.

Once the regrets for each $p \in P$ are calculated, Player can determine which action would have been optimal and decide to put more emphasis on this action in the future. In a game setting, typically the player's goal is to maximize the total payoff. If this is the case, then Player can determine which actions would have yielded better results by determining which $p \in P$ satisfies the equation $r_p > 0$.

1.2 Bayesian Learning (Game Theory)

Thomas Bayes (1702-1761) was a mathematician and Presbyterian clergyman who developed Bayes' theorem which was not published until after his death in 1763 by Richard Price. In the letter written by Richard Price, *An Essay Towards solving a Problem in the Doctrine of Chances* [2], Thomas Bayes states his approach to solving the problem of inverse probability.

Let us begin the discussion of Bayesian Learning with the following definitions. Let A be the explanation of a new observation, B be the new observation, and let C represent a summary of all prior assumptions and experiences. Bayes' rule is stated as follows

$$P(A|BC) = \frac{P(A|C)P(B|AC)}{P(B|C)}. \tag{2}$$

In this equation, $P(A|BC)$ denotes the probability that A is true given that B and C are true. Therefore, $P(A|BC)$ denotes the posterior probability, $P(B|AC)$ denotes the prior probability, and $P(A|C)$ and $P(B|C)$ are probabilities for A being true and B being true given that the history is true, respectively.

Equation 2 describes how a player should learn once new data has become available, therefore B must be known a priori. However, in order to predict the posterior probability prior to the observation being observable, the principle of marginalization must be employed

[13]. The principle of marginalization is described as

$$P(B|C) = \sum_i P(A_i|C)P(B|A_iC). \tag{3}$$

Equation 2 can now be derived to predict the posterior probability of the new observation as follows

$$P(A_i|BC) = \frac{P(A_i|C)P(B|A_iC)}{\sum_j P(A_j|C)P(B|A_jC)}. \tag{4}$$

This equation can be further converted into a form that can be used for utilizing a likelihood function instead of a probability function as follows

$$P(A_i|BC) = \frac{L(C|A_i)P(B|A_iC)}{\sum_j L(C|A_i)P(B|A_jC)} \tag{5}$$

and the purpose of using the likelihood function over the probability function is found in Section 2.3.

CHAPTER 2 PROPOSED SYSTEM

The proposed system consists of a No-Regret decision making process coupled with a Bayesian learning process. The couplng of these two systems, along with the modifications made to the original theories found in Sections 1.1 and 1.2, produces a Decision Making and Learning Model (DMLM) that determines an optimal action based on past regrets and future expected costs as well as an ability to learn from errors found in these expected costs. By basing the proposed models on traditional decision making and learning models found in psychology and game theory, much of the research previously preformed in these fields can be applied to the up and coming field of machine learning.

The process flow for this system can be found in Figure 2. Many of these steps may not

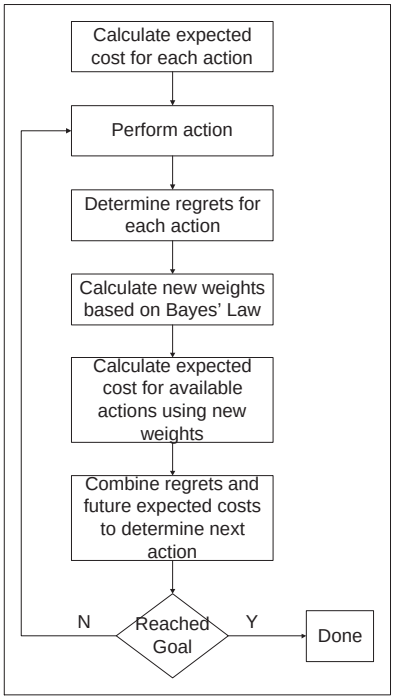


Figure 2: Process Flow Diagram

be apparent at this point, but they will become clear as the following sections are read. To summarize the chart, the process begins with a least cost analysis where the costs for each action are derived from the knowledge known by the system a priori and from priorities set

for each of the multiple objectives. The least costing action is determined and that action is performed. The results of that action are then analyzed and any differences between the expected costs and the actual costs for that action are fed into the Bayesian learning model. The Bayesian learning model then makes adjustments to the priorities of each of the objectives based on the accuracy of the expected costs. Then the least cost analysis is perform on the next set of actions and the least costing action is performed. This process continues until the goal is reached.

In Section 2.1 the system is defined and some of the assumptions are put forth. The following two sections describe in detail the implementation of the DMLM which consists of two parts. The first part, Section 2.2, is the no-regret decision making process that determines which action is optimal based on a least cost analysis. The second part, Section 2.3, describes the learning process that accounts for errors found in the expected costs. In the last section, Section 2.4, a simple example is worked through in order to better show the process used by the DMLM system.

2.1 System Setup

The proposed system assumes the existence of a finite set of actions from which the decision making process determines the optimal solution. The determination of the optimal action is based on all of the information known a priori as well as current data which may be provided by the sensor network. Let the *Universal Set* of actions be defined as

$$\Omega = \{A_1, A_2, ..., A_{N_d}\} \tag{6}$$

where A_i is the description of the action and N_d is the total number of actions in the *Universal Set*.

The determination of the optimal action is dictated by the feasible choices and is selected

by determining the action whose total expected cost is the least. The feasibility for each of the actions is determined by the current state of the system and is necessary because not all of the actions are available at all of the states. Let us define Ω_k to be the set of actions available at instance k and the complement, Ω_k^c, to be the set of actions not available at instance k.

Each action in the *Universal Set* of actions contains N_k iterations at which the decisions are evaluated. This segmenting of the actions allows for the system to update its decision process and choose a different course of actions if the situation requires such a change. Let us define these steps to be

$$\mathcal{K} = \{1, 2, ..., N_k\}. \tag{7}$$

Each iteration of each action is assigned multiple objectives whose values represent the cost of that objective associated with each iteration. These objective costs are initially assigned estimated values represented by

$$\mathcal{O}_i^e = \{f_1^e, f_2^e, ..., f_{N_o}^e\} \tag{8}$$

where f_j^e is defined to be a vector containing the expected cost values for each of the iterations, namely

$$f_j^e = [f_j^e(1), f_j^e(2), ..., f_j^e(N_k)]. \tag{9}$$

Once an iteration of an action is performed, an actual value for each objective can be determined. These actual values are represented by

$$\mathcal{O}_i^a = \{f_1^a, f_2^a, ..., f_{N_o}^a\} \tag{10}$$

where $i \in \{1, 2, ..., N_d\}$ and f_j^a is defined similarly to f_j^e.

This set of objectives is supplemented by a probability distribution of weights,

$$\mathcal{W} = \{w_1, w_2, ..., w_{N_o}\}, \tag{11}$$

that are initially assigned values determined by priorities found in the mission planner, but are later adjusted based on how accurate the estimates are compared to the actual values for the objective costs. These weights are assigned lower bounds, $\lfloor w_i \rfloor$, dictated by the mission planner, in order to ensure that no objective is completely removed from the decision making process.

2.2 No-Regret Decision Making

The decision making process is a derivative of the No-Regret learning process taken from gaming theory, described in section 1.1, with two main differences. The first change proposed is the use of a cost minimization function as opposed to a payoff maximization function to determine the optimal action. Given the action space, Ω, the cost minimization function at instance k is given by

$$A^*_\beta(k): \quad r_{A^*_\beta}(k-1) + \sum_{l=k}^{N_k} C^e_{A^*_\beta}(l) = \min_{\beta \in \Omega_k} \left[r_\beta(k-1) + \sum_{l=k}^{N_k} C^e_\beta(l) \right]. \tag{12}$$

The second change is the inclusion of the future expected values in the decision making process as given by $\sum_{l=k}^{N_k} C^e_\beta(l)$ in Equation 12. This element is calculated as the sum of the instantaneous expected composite costs of action β starting from iteration k. The instantaneous expected composite cost is defined as

$$C^e_\beta(k) = \sum_{j=1}^{N_o} w_j(k) f^e_j(k). \tag{13}$$

This equation shows the use of the weighted sum to determine the total costs associated with

each of the actions. One caveat to this weighted sum is that all of the objective values must be normalized to equal values such that $f_i \in (0, 1)$. One way to accomplish this normalization is to determine the mean of each of the objectives and then normalize these values to 0.5.

The other element in the minimization function, given in Equation 12, is the no-regret function. This function calculates the regret for not having chosen each of the other actions in the *Universal Set* of actions. The regret for not having chosen action β up to time k is

$$r_\beta(k) = \sum_{l=1}^{k} C_\beta(l) - C^*(k-1) \tag{14}$$

where $C_\beta(l)$ is the instantaneous cost associated with action β at time l and C^* is the total incurred cost up to time k defined as

$$C^*(k) = \sum_{l=1}^{k} C_{\beta_l^*}(l) \tag{15}$$

where β_l^* represents the iteration chosen at instance l.

From Equation 14, the instantaneous composite cost for action β at iteration l is defined to be either the expected composite cost if the iteration has not previously been chosen, or the actual composite cost associated with the iteration if it has been chosen in the past. The instantaneous composite cost function is defined as

$$C_\beta(l) = \begin{cases} \sum_{j=1}^{N_o} w_j(l) f_j^a(l) & \text{if } \beta = \beta_l^* \\ \sum_{j=1}^{N_o} w_j(l) f_j^e(l) & \text{if } \beta \neq \beta_l^* \end{cases} . \tag{16}$$

2.3 Bayesian Learning

Included in the decision making process is a value for the estimated cost associated with each action for each of the objectives. These estimated costs are calculated from data that may or may not be erroneous. Some possible factors producing erroneous estimated cost values are sensor noise, the use of long-range inaccurate sensor networks, or changes in the environment such as weather conditions or unforeseen obstacles. More accurate, or actual, cost values for the objectives are determined for each of the iterations that have been chosen to be optimal in the past. These cost values may be calculated from data provided by shorter-range more accurate sensors.

These errors are accounted for in such a way as to discredit the data used in future decisions. However the way in which to discredit the data and the amount in which to discredit the data is one of the most important models of this proposed system. Based on the game theory found in Section 1.2, Bayesian learning has been proposed to solve this problem. Traditional Bayesian learning uses a probability function to measure the conditional probability, $P(B|A)$, which estimates how often one action will occur. In the proposed system, however, the accuracy of the data in the set of actions is what is needed, therefore, a likelihood function is used in place of the probability function [6]. The likelihood function derived for this system is

$$L_j(k) = 1 - |f_j^a(k) - f_j^e(k)| \tag{17}$$

with $j \in [1, 2, ..., N_o]$ representing the number of objectives. This likelihood represents the probability that objective j will be true at instance k.

In Section 1.2, the Bayesian learning method from game theory was discussed and Equation 5 was found. This equation must be transformed into a form that can be used in the

proposed system. This transformed function is defined as

$$w_j(k+1) = \begin{cases} \frac{L_j(k)w_j(k)}{\sum_{z=1}^{N_o} L_z(k)w_z(k)}, & \text{if } w_j(k+1) > \lfloor w_j \rfloor \\ \lfloor w_j \rfloor, & \text{otherwise} \end{cases} \tag{18}$$

where L_j is defined in Equation 17 and $\lfloor w_j \rfloor$ is the floor of the j^{th} weight. This equation only takes into consideration the values for the objectives from the previous iteration, however, it can be transformed into a running average if the changes in w_j are found to be too sporadic.

2.4 Numerical Example

Following from the process flow diagram shown in Figure 2, the following example illustrates the process involved with the proposed DMLM. Let us begin with the course shown in Figure 3. This course consists of four possible paths, $N_d = 4$, each with two segments yielding three waypoints, $N_k = 2$. Each of the segments is arbitrarily assigned two values representing two objectives, shown in blue type.

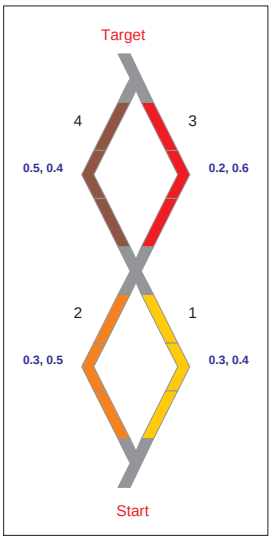


Figure 3: Numerical Example - Course

The first step in the process is to determine the best of the four paths and to traverse

that path segment. At this point in the process, there is not regret, thus $r_\beta(k-1)$ from Equation 12 is equal to zero. Therefore, the minimization function is solely based on the expected future costs of each of the path segments. Calculating these costs while assuming that the weight distribution is even, we get the following equations for the expected costs.

$$\sum_{l=1}^{2} C^e_{1-3}(l) = [0.5(0.3) + 0.5(0.4)] + [0.5(0.2) + 0.5(0.6)] = 0.75$$
$$\sum_{l=1}^{2} C^e_{1-4}(l) = [0.5(0.3) + 0.5(0.4)] + [0.5(0.5) + 0.5(0.4)] = 0.80$$
$$\sum_{l=1}^{2} C^e_{2-3}(l) = [0.5(0.3) + 0.5(0.5)] + [0.5(0.2) + 0.5(0.6)] = 0.80$$
$$\sum_{l=1}^{2} C^e_{2-4}(l) = [0.5(0.3) + 0.5(0.5)] + [0.5(0.5) + 0.5(0.4)] = 0.85$$

From the above calculations, path 1-3 is determined to be the least costing path. Therefore, path segment 1 is traversed. Once this path segment is traversed, the actual values associated with the two objectives are determined. These values are shown in green type in Figure 4.

Once the first path segment has been traversed, the regrets from not having taken each of the other paths is calculated.

$$r_{1-3}(1) = [0.5(0.7) + 0.5(0.4)] - 0.55 = 0.00$$
$$r_{1-4}(1) = [0.5(0.7) + 0.5(0.4)] - 0.55 = 0.00$$
$$r_{2-3}(1) = [0.5(0.3) + 0.5(0.5)] - 0.55 = -0.15$$
$$r_{2-4}(1) = [0.5(0.3) + 0.5(0.5)] - 0.55 = -0.15$$

These values indicate that there exists more regret for having chosen path segment 1 than there would have been if we had chosen path segment 2. This is due to the fact that we now

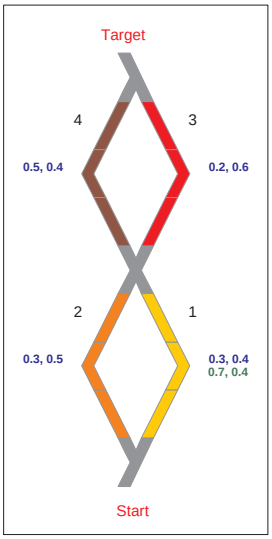


Figure 4: Numerical Example - First Iteration

know what the actual values are for path segment 1. However, because we do not know any more information about path segment 2, no conclusions can be made about the validity of the path chosen from this information alone.

The next step in the decision making process is to feed the actual values into the Bayesian learning algorithm in order to make up for any errors. From Equation 17, the likelihood functions for the two weights are

$$L_1(1) = 1 - |0.7 - 0.3| = 0.6$$
$$L_2(1) = 1 - |0.4 - 0.4| = 1.0$$

which says that there is a 60% likelihood that objective 1 will be correct and a 100% likelihood that objective 2 will be accurate. These values are fed into Equation 2 and the following shows the calculations for determining the next weight distribution.

$$w_1(2) = \frac{(0.6)(0.5)}{(0.6)(0.5) + (1.0)(0.5)} = 0.3750$$
$$w_2(2) = \frac{(1.0)(0.5)}{(0.6)(0.5) + (1.0)(0.5)} = 0.6250$$

Now that the new weight distribution has been calculated, the expected future costs for each of the available paths must be calculated. At this point in the course, there are only two paths available, path 1-3 and path 1-4, therefore the expected future costs for each of these paths must be determined using the new weight distribution.

$$\sum_{l=2}^{2} C^{e}_{1-3}(l) = [0.375(0.2) + 0.625(0.6)] = 0.4500$$
$$\sum_{l=2}^{2} C^{e}_{1-4}(l) = [0.375(0.5) + 0.625(0.4)] = 0.4375$$

and the minimization function shown in Equation 12 becomes

$$\min_{\beta \in \Omega_k} [0.0 + 0.45, 0.0 + 0.4375]$$

yielding path 1-4 as the next optimal path to traverse. After traversing path segment 4, the vehicle has reached its destination, therefore, no more calculations are needed. However, for due diligence in case the vehicle is assigned more possible waypoints once it has reached this goal, let us recalculate the regrets and the new weight distribution based on the actual values for path segment 4 found in Figure 5.

The regret from not having traversed each path is again calculated to determine if the correct course of action has been taken. To this end, the regret calculations are presented as follows

$$r_{1-3}(2) = [0.5(0.7) + 0.5(0.4) + 0.3750(0.2) + 0.5(0.6)] - 0.9875 = 0.0125$$
$$r_{1-4}(2) = [0.5(0.7) + 0.5(0.4) + 0.3750(0.5) + 0.5(0.4)] - 0.9875 = 0.0000$$
$$r_{2-3}(2) = [0.5(0.3) + 0.5(0.5) + 0.3750(0.2) + 0.5(0.6)] - 0.9875 = -0.1375$$
$$r_{2-4}(2) = [0.5(0.3) + 0.5(0.5) + 0.3750(0.5) + 0.5(0.4)] - 0.9875 = -0.1500$$

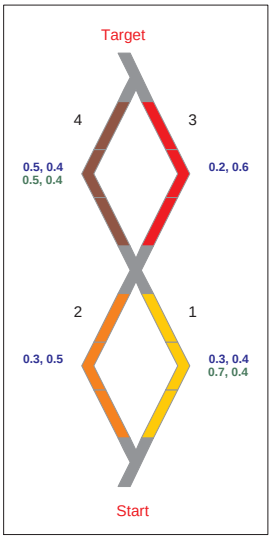


Figure 5: Numerical Example - Second Iteration

and the new likelihood calculations are

$$L_1(2) = 1 - |0.5 - 0.5| = 1.0$$

$$L_2(2) = 1 - |0.4 - 0.4| = 1.0.$$

The next weight distribution to be used if the current position is not the final goal are to remain unchanged as is shown by the following calculations

$$w_1(3) = \frac{(1.0)(0.375)}{(1.0)(0.375) + (1.0)(0.625)} = 0.3750$$
$$w_2(3) = \frac{(1.0)(0.625)}{(1.0)(0.375) + (1.0)(0.625)} = 0.6250$$

and is intuitive due to the fact that the expected and actual values were equal.

CHAPTER 3 APPLICATIONS

The proposed system can be utilized for many applications requiring a multi-objective decision making algorithm with a learning capability. This system was originally designed as a path selection algorithm to determine the optimal paths for an unmanned autonomous ground vehicle. The platform for this system is being developed jointly by the University of Central Florida and Lockheed Martin as a total package for mission planning and navigation for unmanned autonomous ground vehicles. The system architecture for the vehicle navigation model can be seen in Figure 1.

A second application for the proposed DMLM, with slight modifications to the platform, is its use in the updating of sensor fidelity range values. Assume an unmanned ground vehicle has a sensor network that has high precision in the short range, and low precision in the long range. The vehicle can be assigned ranges of sensor fidelity and each of these ranges is assigned a probability of accuracy. This probability is then updated using a Bayesian approach as developed in Section 2.3.

3.1 Vehicle Navigation Model

The Vehicle Navigation Model is made of three subsystems shown in Figure 1. The first subsystem is an Informational Database which consists of all of the data needed by the Decision Making Model. This data is organized into a Finite State Model which contains a waypoint based path-planning algorithm in order to determine the possible paths for the DMM to choose from. Also contained in the FSM is the data required for calculating the objective costs for each of the paths and their segments. This data originates from a Geo Database which is populated by either long-range sensor networks or data known a prior.

The second subsystems is the Decision Making and Learning Model which decides which paths, supplied by the FSM, will yield the greatest results based on the mission objectives

found in the mission planner. This DMLM is based on the proposed system defined in Section 2 of this document. The DMLM receives data from two sources, the FSM and feedback from the Execution subsystem.

The third subsystem in this model is the Execution of the actions determined by the DMLM. This subsystem consists of the hardware and software needed for the vehicle to traverse the specified path. This subsystem also feeds back sensor data to the DMLM. This sensor data is then used to determine the accuracy of the data found in the FSM.

In order to better quantify possible objectives for each path segment, a list of path characteristics has been formulated. This list of characteristics is not necessarily complete, but shows many of the road characteristics that may be taken into consideration when the objective costs are calculated for each path segment. These characteristics originate from both Human Driver Models [5, 10, 11] created for the purpose of traffic safety as well as mentally simulating the driving process and the steps taken by an individual to determine a best route. The Human Driver Models typically include both behavioral models and cognitive models. The behavioral models represent how the typical driver reacts to certain road conditions or events whereas the cognitive model attempts to describe why drivers react in a particular way to these conditions or events. The path characteristics are divided into groups as shown in the following sections.

3.1.1 Path Specifics

Data pertaining to each of the paths will be provided to the DMM by the waypoint based path-planning algorithm found in the FSM. This data includes the information needed for a vehicle to traverse the paths as if there are no unknown variables such as weather, dynamic obstacles, etc. This list includes the following characteristics.

- **Distance between waypoints:** This is the measured distance between all of the waypoints for a given path.

- **Bearing between waypoints:** The measured compass bearing between each of the waypoints.
- **Total distance of path:** A simple measure calculated by summing all of the distances between all of the waypoints for a given path.
- **Duration of traversal:** The distances between the waypoints is used in combination with the maximum allowable speed for the respective path segment to calculate an estimated total time to traverse the path.

3.1.2 Path Conditions

More information is needed about each of the paths in order for the Decision Making Model to make informed decisions. This information would ideally be provided to the DMM prior to sending out any vehicles. This unfortunately will not always be the case since some of these characteristics change in real-time. Therefore, this data will be represented to the best of the Geo Database's ability, but then updated as the vehicle traverses through the paths. This additional information includes the following list of categories.

- **Climate:** The climate for each of the paths will typically be similar, however, if there are differences, these differences can produce an effect on the decision of whether or not to traverse this particular path. More specific conditions are listed as follows.
  - **Rain/Snow:** This can effect visibility of the sensors as well as cause problems for vehicles that are not resistant to moisture.
  - **Wind Speed:** For larger vehicles, the wind speed can alter both the course and speed of the vehicle.
  - **Temperature:** The ambient temperature can affect the vehicles in such manners as motor temperature, sensor range/accuracy, etc.

  - **Extreme:** This category is included for extreme weather circumstances such as tornadoes, hail, etc. that may cause physical damage to the vehicles.

- **Known obstacles:** These are obstacles that the mission planner has prior knowledge of. This knowledge most likely comes from either other vehicles or from terrain maps. Such obstacles can include bridges, pedestrian walkways, etc.

- **Condition of roadway:** This category describes the condition of the roads that the vehicle will travel. More specific conditions are listed as follows.

  - **Paved:** Whether or not the roadway is a manicured dirt road, a paved road, or a trail.

  - **Disrepair:** Determines how well the roadway is maintained; are there potholes, ruts, speed bumps, etc that may damage the vehicle.

- **Unknown obstacles:** There is an inherent risk in all of the paths for some unforeseen obstacles that may hinder the progress of the vehicle. This measure is the probability of this occurring on this path.

  - **Traffic negotiation:** This is a value used to estimate the probability that the vehicle will incur delays due to traffic if this path is taken. If traffic is encountered, the vehicle will likely have to slow down and convert from an obstacles avoidance strategy to a leader-follower strategy for reaching its destination.

- **Maximum speed:** Typically, roadways have speed limits that reflect the safest speed allowed for safe travel. These speed limits are based on the number of lanes, side streets, etc.

3.1.3 Vehicle Specifics

Information about the vehicle will be needed to make proper decisions. The mission could include vehicles of differing types. Therefore, a model for each vehicle must be taken into account in the final decision of which path(s) to take for that vehicle. The information that must be known about each vehicle is listed as follows.

- **Sensor Range:** This is the maximum distance that the sensors for the vehicle can accurately sense data in an ideal environment.
- **Dimensions:** Measurements of the vehicle used to determine whether there are any size restraints for the vehicle. For example, is the vehicle short enough to pass under all of the bridges, or is the vehicle narrow enough to allow for on-coming traffic to pass.
- **Turning Radius:** The maximum turning radius is needed to ensure that the vehicle can successfully negotiate all of the turns needed in the path.
- **Speed:** This is the maximum speed that the vehicle can travel.
- **Safe repulsive field:** This is the smallest repulsive field required to encompass the entire vehicle in order to be sure that the vehicle remains safe.
- **Current Status:** Each vehicle will have a current status report that will inform the decision maker if there is a problem with the vehicle or if it in good working order.

3.1.4 Objectives

Each of the objectives in the DMM will be a combination of the aforementioned characteristics to determine estimated costs associated with each of the path segments. This section presents a few examples of possible objectives used by the DMM.

- **Maximum Time:** This defines a maximum time allowed to traverse from the starting point to the ending point.

- **Vehicle Safety:** This is used to define the size of the repulsive field that surrounds the vehicle in order to keep the vehicle safe. With a larger field, the vehicle may need to travel at a slower speed or even take a different route in order to avoid all obstacles.

- **Obstacle Safety:** This is similar to Vehicle Safety except that it applies to the obstacles instead of the vehicle.

- **Fuel/Energy consumption:** Most vehicles have a limited traveling range before they need to be refueled. This measure gives the maximum allowed consumption during the traversal of the path.

- **Vehicle Separation:** The maximum amount of distance separating the vehicles can be prescribed. This objective can be used, for example, to set how long it takes for every vehicle to reach the goal after the first vehicle has reached it. Another example might be to ensure that communication between all of the vehicles is maintained.

3.2 Sensor Fidelity Ranges

The second application shows how the DMLM can be used to adjust a sensor network's fidelity ranges based on historic accuracy. To this end, let us assume we are using a vehicle that has three sensor fidelity ranges, 100%, 50%, and 25%, as shown in Figure 6. When an obstacle passes into the next lowest fidelity range, it can be determined with better accuracy whether or not the obstacle exists. If the obstacle enters the 100% fidelity range, then the obstacle is known to exist. The percentages associated with each of the fidelity ranges must be adjusted based on what the vehicle actually encounters.

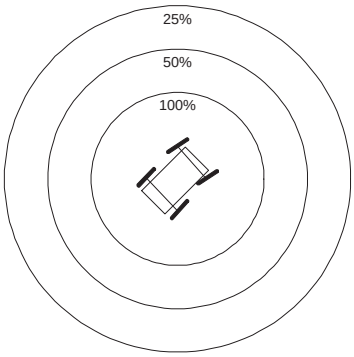


Figure 6: Fidelity Simulation - Vehicle Diagram

The adjustments to the fidelity ranges can be determined by the same Bayesian method developed in Section 2.3, with some stipulations. The first stipulation is that the Bayesian method developed in Section 2.3 requires that the weight distribution among the objectives be a probability distribution. This mean that all of the weights in the distribution must sum to one. Therefore, each fidelity range must be considered separately and since at least two weights must be present for any changes to occur, each fidelity range is assigned a fictitious value for a second weight. In the two weight case, the following equation is adapted from Equation 18

$$w_1(k+1) = \frac{L_1(k)w_1(k)}{L_1(k)w_1(k) + L_2(k)w_2(k)} \tag{19}$$

where $w_1(k+1)$ is the updated fidelity range, $w_1(k)$ is the previous fidelity range, $L_2(k) = 1$ is the fixed likelihood for the fictitious weight, and $w_2(k) = 1 - w_1(k)$ is the fictitious weight. From this equation we derive the following

$$w_r(k+1) = \frac{L_r(k)w_r(k)}{L_r(k)w_r(k) + 1.00[1 - w_r(k)]} \tag{20}$$

where r is the index of the fidelity range being updated.

In order to define the likelihood function for each weight, the system must keep track of the existence of each of the obstacles that pass into the next lowest fidelity range. To that end, let us define $G_r(k)$ to be the number of obstacles that have existed in both ranges r

and $r-1$ up to time k. Let us also define $B_r(k)$ to be the number of obstacles that existed in r and should have existed in $r-1$, but did not. The likelihood function for range r is then defined as

$$L_r(k) = 1 - \left[w_r(k) - \frac{G_r(k)}{G_r(k) + B_r(k)} \right]. \tag{21}$$

Using these equation, a finite number of fidelity ranges can be assigned to a vehicle. These fidelity ranges can then be updated as the vehicle traverses closer to the obstacles according to the accuracy of the fidelity ranges as the obstacles pass into higher accuracy ranges.

CHAPTER 4 SIMULATIONS

The proposed Multi-Objective Decision Making and Learning Model (DMLM) is simulated using Matlab in four separate sets of simulations. The first set of simulations, Section 4.1, uses the proposed platform in Figure 1 to traverse through a field simulating an urban environment. This urban field consists of many static obstacles which form a grid-like pattern of streets. Section 4.2 discusses two simulations performed in an open field environment where fewer static obstacles are randomly placed. The third simulation, Section 4.3, discusses a simulation performing a sensor fidelity updating process described in Section 3.2. The last simulation combines the sensor fidelity updating process with the open field environment to show the adaptability of the system to multiple applications.

4.1 Urban Field Simulations

The first set of simulations are performed using the proposed DMLM and consists of a simulation field that is representative of a real-world application. This simulation field represents an urban environment consisting of a grid-like array of streets with surrounding obstacles. This urban field can be seen in Figure 7.

The urban field consists of a map of an urban environment that is assumed to be know a priori. However, only the static obstacles shown in Figure 7 are known. This leaves the possibility for the presence of unknown static and dynamic obstacles located within the paths. Some examples of these unknown obstacles may include vehicular traffic, pedestrian traffic, poor road conditions, poor weather conditions, etc. These unknown obstacles are what the proposed DMLM uses to determine an optimal path. To this end, the presence of unknown obstacles must be estimated for each of the path segments. For example, typical traffic densities can be estimated using historical information that may be found in the Geo Database.

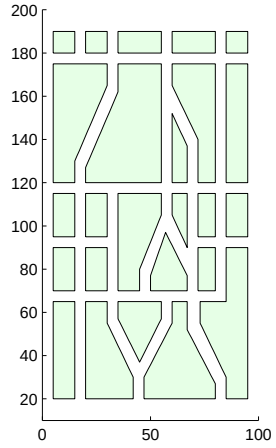


Figure 7: Urban Simulation - Field

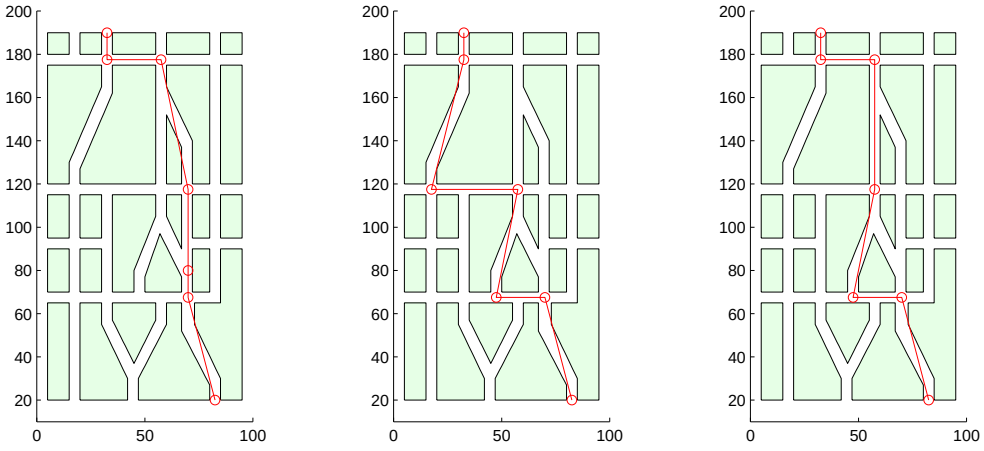
This urban field consists of a large finite number of paths for the DMLM to selected from, however, due to computation restraints, let us assume that we are given a set of only five paths to choose from, $N_d = 5$. These five paths can be seen in Figure 8. Each of these paths is divided into six segments which yields seven waypoints, $N_k = 6$.

The three simulations performed using this urban field collectively show how unknown obstacles, in this case vehicular traffic, can change the selection of the optimal path. The first simulation shows how the DMLM performs with only one objective. The second simulation shows what happens when traffic is introduced to the decision making process. Lastly, the third simulation shows how a mission objective of avoiding traffic can be implemented.

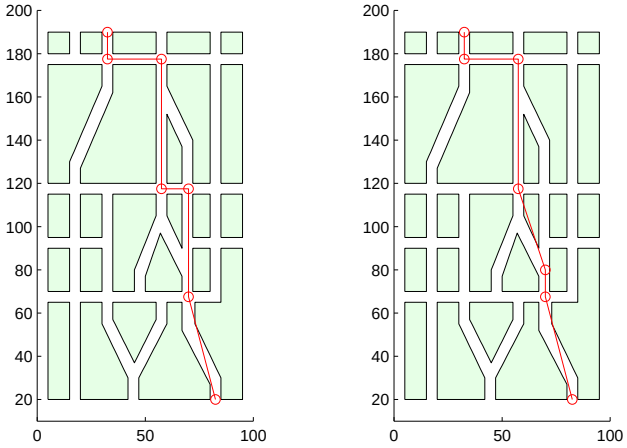
4.1.1 Urban Simulation - Shortest Distance

The first simulation to be performed on this field consists of one objective, $N_o = 1$. This single objective is to minimize the total amount of distance traveled from the starting point at the bottom right to the ending point at the top left. The straight line distances between the selected waypoints can be seen in Table 1.

The value in italics, namely the total straight line distance for path one, is used to



(a) Path 01 (b) Path 02 (c) Path 03



(d) Path 04 (e) Path 05

Figure 8: Urban Simulation - All Available Paths

Table 1: Urban Simulation - Estimated Path Distances

Segment	1	2	3	4	5	6	Total
Path 01	49.12	12.50	37.50	61.29	25.00	12.50	*197.91*
Path 02	49.12	22.50	50.99	40.00	61.85	12.50	236.91
Path 03	49.12	22.50	50.99	60.00	25.00	12.50	220.11
Path 04	49.12	50.00	12.50	60.00	25.00	12.50	209.12
Path 05	49.12	12.50	39.53	60.00	25.00	12.50	198.65

determine that the optimal path is path one. Assuming no obstacle avoidance path planning algorithm is used to avoid the static obstacles surrounding the paths leads to the assumption that no errors exist in the distances traveled after each path segment is traversed. This assumption guarantees that no change in the selection of the optimal path is present, and therefore the path traversal is shown in Figure 8(a).

4.1.2 Urban Simulation - Shortest Distance with Equal Traffic

The second simulation performed on this field includes the addition of a second objective. This second objective is to minimize the amount of traffic encountered by the vehicle. To this end, each of the path segments must be assigned an expected objective cost associated with the expected traffic density of that path segment. For the sake of showing how an objective's weight changes based only on its accuracy, let us assume that no traffic exists on any of the path segments. The addition of this second objective is necessary so that a probability distribution can be formed across more than one weight. This allows for the weight associated with objective one to fluctuate depending on its accuracy. Let us also assume that initially the weight distribution across these two objectives is equal.

Based on the fact that there exists no traffic on any of the path segments, the choice of the optimal path is entirely based on the first objective. Therefore, the DMLM determines that path one, just as in the first simulation, is the optimal path at the first iteration. After the first segment of path one is traversed, let us arbitrarily assign an actual traveled distance of 58.50 to objective one. This error is then used to shift the weight distribution to favor the second objective since it was found to be more accurate than the first objective. This change can be seen in Figure 9.

With this new weight distribution, the DMLM determines that the optimal path is no longer path one. Instead, the next optimal path is calculated to be path five. This intuitively makes sense because by adding 9.38 to the distance of path one, it is no longer the shortest

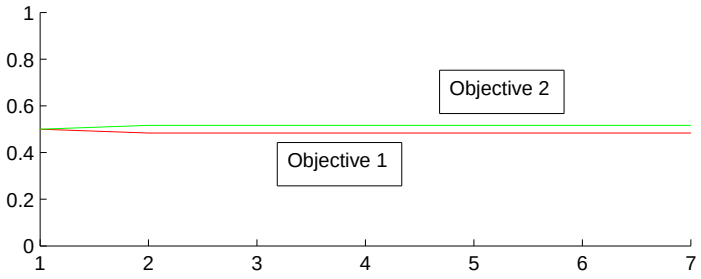


Figure 9: Urban Simulation - Weight Distribution with no Traffic

path and therefore, the next shortest path, path five, is chosen. Since there are no other perturbations, the DMLM calculates that path five is the optimal path for the rest of the path segments. The trajectory of this path looks the same as Figure 8(e) since the first segment of all of the paths is the same.

4.1.3 Urban Simulation - Shortest Distance with Least Traffic

The third simulation performed on the urban field is a continuation of the first two simulation sets presented in this chapter. Let us assign values to the estimated traffic densities for each of the path segments. These percentages are randomly chosen and are shown in Table 2.

Table 2: Urban Simulation - Estimated Traffic Densities

Segment	1	2	3	4	5	6	Total
Path 01	18.97	15.09	59.36	81.80	53.41	37.04	265.67
Path 02	18.97	69.79	49.66	19.34	30.93	37.04	225.73
Path 03	18.97	69.79	49.66	34.20	53.41	37.04	263.07
Path 04	18.97	86.00	82.16	34.20	53.41	37.04	311.78
Path 05	18.97	15.09	64.50	34.20	53.41	37.04	223.21

If it is assumed that the initial weight distribution is equal and the objectives remain the same as before, then path five is selected to be the optimal path. If a distance error of 17 is found for segment one of path five, then the weight distribution shifts, but the optimal path

remains path five. The change in weight distribution can be seen in Figure 10 from iteration one to iteration two.

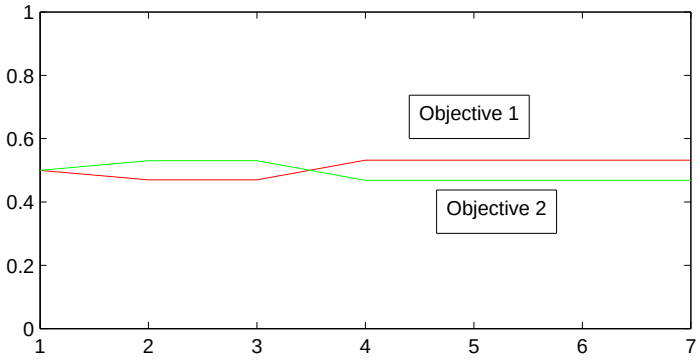


Figure 10: Urban Simulation - Weight Distribution with Traffic

Next, let us assign an error to the amount of estimated traffic located in segment three of path five. If an increase in traffic density of 22% is encountered, then the DMLM chooses path two as the optimal path. The trajectory of this simulation can be seen in Figure 11 where path five is shown in blue and path two is shown in red.

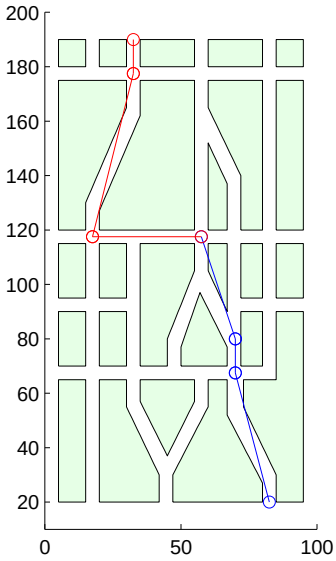


Figure 11: Urban Simulation - Trajectory with Traffic

This simulation shows the possibility for the changing of a decision in real-time due to errors encountered in the costs associated with the objectives during the traversal of each path segment.

4.2 Open Field Simulations

This next set of simulations are performed on an open terrain field where the obstacles are randomly placed. This field is divided into four sections with the starting point on the left and the ending point on the right. Three sets of three waypoints are placed in the field offering a total of twenty-seven paths for the DMLM to choose from. This can be seen in Figure 12 where the filled circles represent the obstacles and the open circles represent the waypoints.

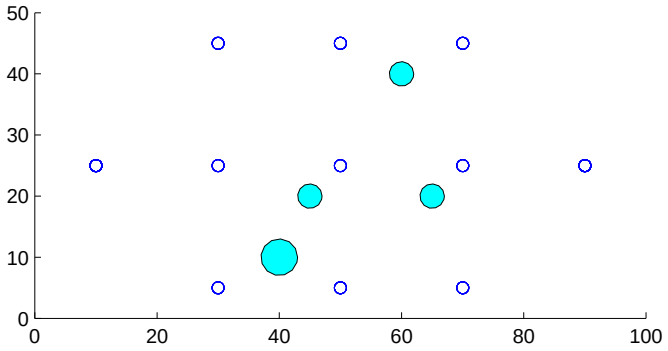


Figure 12: Open Simulation - Field

The two simulations presented in this section show how the changing of the initial weight distribution can allow for the execution of different mission goals. Each of these simulations will be given the same two objectives. The first objective is to minimize the total distance traveled to the goal and the second objective is to minimize the distances from each of the

path segments to each of the obstacles.

$$\begin{aligned} w_1 f_1 &\rightarrow \text{minimize distance to goal} \\ w_2 f_2 &\rightarrow \text{minimize distances to obstacles} \end{aligned} \tag{22}$$

Also included in these simulations is a dynamic obstacle avoidance path planning algorithm [12] previously developed by Lockheed Martin and the University of Central Florida. This path planning algorithm is used to more precisely simulate the scenarios by taking into consideration actual values for distances traveled between the waypoints. The path planning algorithm takes as inputs a starting point and heading, an ending point and heading, and the static and dynamic obstacles present between the waypoints. Although the path planning algorithm supports both static and dynamic obstacles, as well as hard and soft obstacles [14], only hard static obstacles will be considered.

4.2.1 Open Simulation - Exploratory

This first simulation shows how a vehicle can perform an exploratory role while maintaining a short distance between the starting point and the goal. In order to accomplish this task, the weight distribution is manipulated such that there is a 75% weighting on the objective of minimum distance to each obstacle and a 25% weighting on the objective to travel the shortest distance. The result of this simulation can be seen in Figure 13.

$$\begin{aligned} 0.25 f_1 &\rightarrow \text{minimize distance to goal} \\ 0.75 f_2 &\rightarrow \text{minimize distances to obstacles} \end{aligned} \tag{23}$$

Using the path planning algorithm discussed previously, the actual distances traveled can be measured and are fed back into the Bayesian updating process. The results obtained from

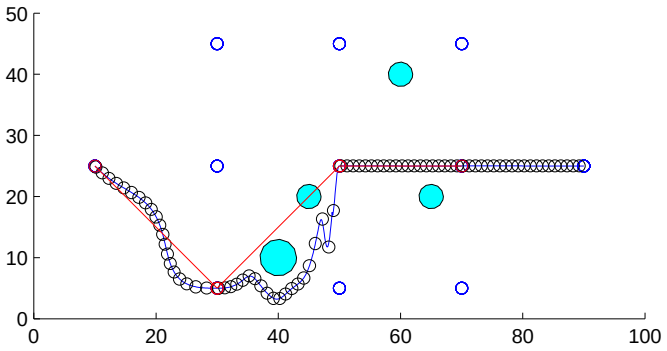


Figure 13: Open Simulation - Exploratory Trajectory

using the path planning algorithm show that each of the distances traversed is either equal to or more than the expected values, which intuitively makes sense. For this simulation, the obstacles are assumed to be stationary and that no errors occur in their expected locations. Therefore, there are no errors found between the actual values and the expected values for f_2.

Figure 14 shows the changes in the weight distribution as the vehicle progresses through the course. It can be seen that since the actual traveled distance and the expected travel distance are not equal, the weight distribution tends toward the more accurate values of the second objective.

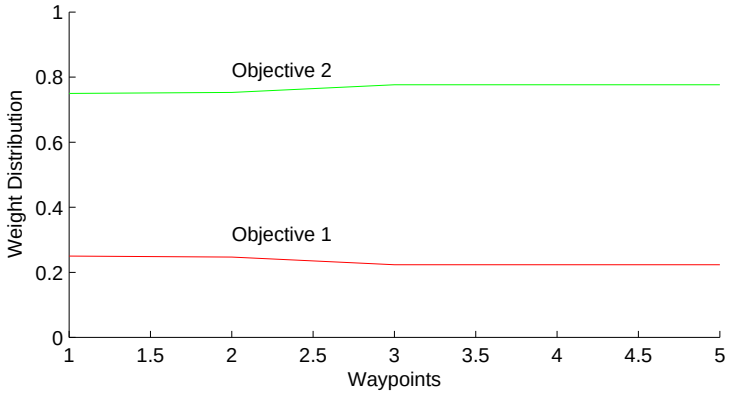


Figure 14: Open Simulation - Exploratory Weight Distribution

4.2.2 Open Simulation - Shortest Duration

The second simulation shows how the weight distribution can be manipulated to accommodate a mission plan of reaching the target in the shortest distance. This is accomplished by changing the initial weight distribution to favor the first objective.

$$\begin{aligned} 0.60 f_1 &\rightarrow \text{minimize distance to goal} \\ 0.40 f_2 &\rightarrow \text{minimize distances to obstacles} \end{aligned} \tag{24}$$

The traversed path can be seen in Figure 15.

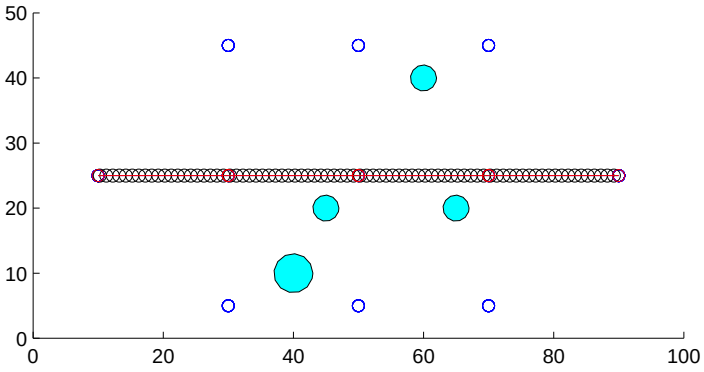


Figure 15: Open Simulation - Shortest Distance Trajectory

The weight distribution for this simulation does not incur significant changes from erroneous data. This is due to the path planner being able to traverse closely to the straight line estimations between the waypoints. The weight distribution can be seen in Figure 16.

4.3 Sensor Fidelity Updating Simulation

A simple simulation consisting of only fidelity range updating is presented to show the functionality of the updating of sensor fidelity range values as described in Section 3.2. Let

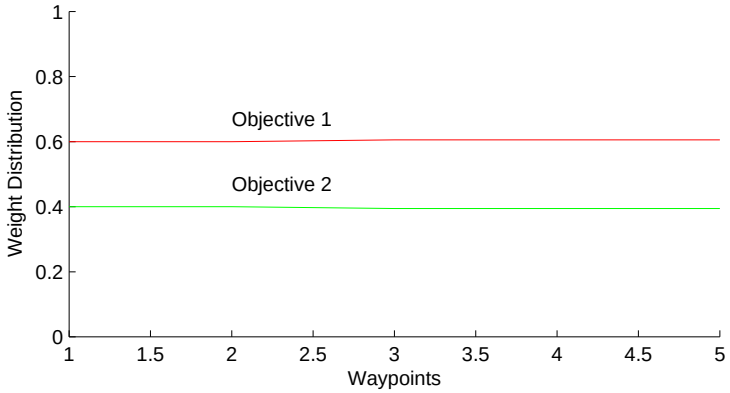


Figure 16: Open Simulation - Shortest Distance Weights

us assume there are three ranges as shown in Figure 6 and given initial values of $w_1 = 1.00$, $w_2 = 0.50$, and $w_3 = 0.25$. Figure 17 shows the field of obstacles as well as the vehicle shown in its starting position with the three circles representing the sensor fidelity levels.

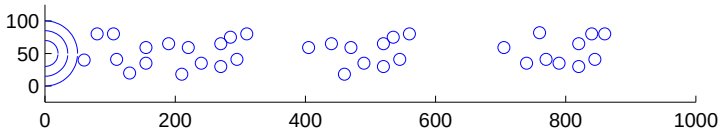
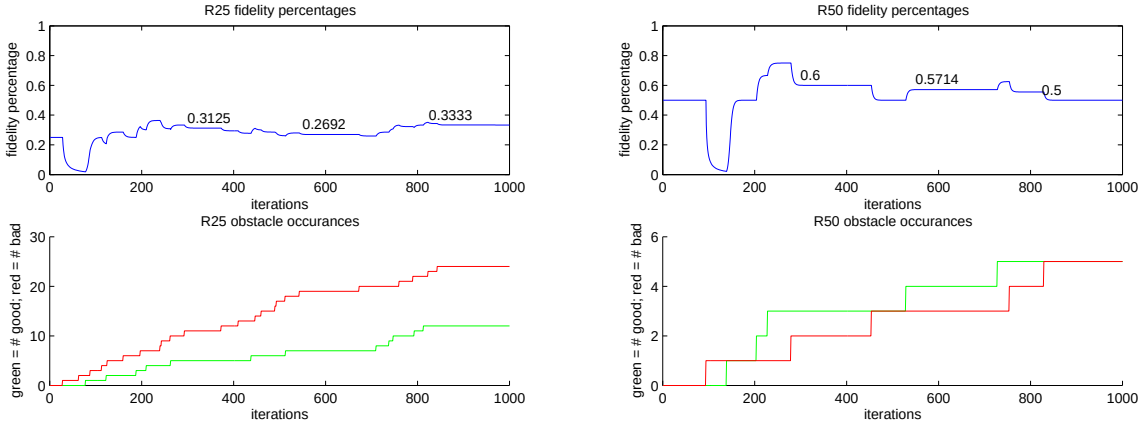


Figure 17: Sensor Fidelity Simulation - Field

None of the obstacles shown are known to the vehicle until the obstacle is located within one of the fidelity ranges. Also, if the vehicle traverses in a straight line along the x-axis, all of the obstacles will be avoided, which is not apparent from the figure. The field is divided into three sections with x-axis values of $1 - 320$, $320 - 600$ and $600 - 1000$. Each of these sections contains a different ratio of good and bad obstacles. The first section contains a ratio of obstacles which allows for both the 50% and the 25% ranges to increase. The second section's ratio shows how both ranges can decrease while the third section illustrates how the 50% range can decrease while the 25% range increases. Figure 18 shows the changes in the range values as well as the associated values for G_r and B_r.



(a) 25% Fidelity Range

(b) 50% Fidelity Range

Figure 18: Sensor Fidelity Simulation - Fidelity Ranges

Taking a closer look at the numbers, we can see that at the 320 mark on the x-axis, the 50% range values for G_{50} and B_{50} are 3 and 2, repsectively. If we logically think about the success rate for this section we can see that 3 out of 5 times the 50% range was accurate. This yields a probability of $3/5 = 0.60$ or 60% that the sensor network will accurately detect an obstacle within this range. This is the same values listed in Figure 18(b) at the 320 mark. Table 3 shows the results if the same procedure is applied at the end of all three segments for each of the two ranges.

Table 3: Sensor Fidelity Simulation - Verification of Sensor Ranges

	400	700	1000
G_{50}	3	4	5
B_{50}	2	3	5
$Probability_{50}$	60.00%	57.14%	50.00%
G_{25}	5	7	12
B_{25}	11	19	24
$Probability_{25}$	31.25%	26.92%	33.33%

4.4 Sensor Fidelity in Open Field Simulation

The last simulation presented in this thesis combines the open field simulations from Section 4.2 with the sensor fidelity simulations presented in Section 4.3. This combination creates a path selection algorithm that takes into account the accuracy of the sensor network located on the vehicle. Unlike the other simulations presented in this chapter, the Bayesian updating process will not be applied to the multiple objectives. Instead, the Bayesian updating process is used to update the sensor fidelity ranges of the vehicle as new obstacles are encountered.

The use of multiple objectives is still applied with the each of the objectives assigned a priority, just as in previous simulations, however, these priorities will not be changed. The first objective used in this simulation is the same as in all of the others, which is to minimize the total distance traveled by the vehicle. The second objective used in this simulation is similar to the second objective from the open field simulation. The main difference between this simulation and the previous simulation is in the way f_2 is calculated. In the previous simulation, f_2 is calculated by simply determining how far each of the obstacles was from the straight line estimation of each path segment. In this simulation, f_2 is calculated from only the obstacles that fall into one of the fidelity ranges. Also in the calculation of f_2, the probability of accuracy for the sensor fidelity range from which each obstacle is located will be used to weight the distances.

The simulation field is nearly the same as that of the open field simulation, however, it is elongated and populated with eight obstacles instead of four. The field can be seen in Figure 19. One caveat to this simulation is the way in which these obstacles are handled. Each of the eight obstacles are assigned thirteen obstacles which are located around its perimeter. This is a requirement of the path planning algorithm being used to traverse between the waypoints. This method of handling the obstacles does not have any direct effect on the ability of the proposed algorithm, in fact, to some it may improve it. The caveat is that

if an obstacle does not completely fall into one of the fidelity ranges, then not all of that obstacle's thirteen obstacles will be counted.

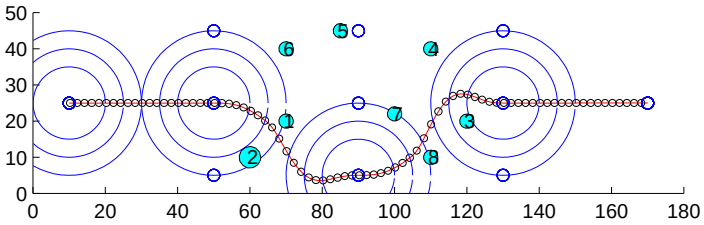
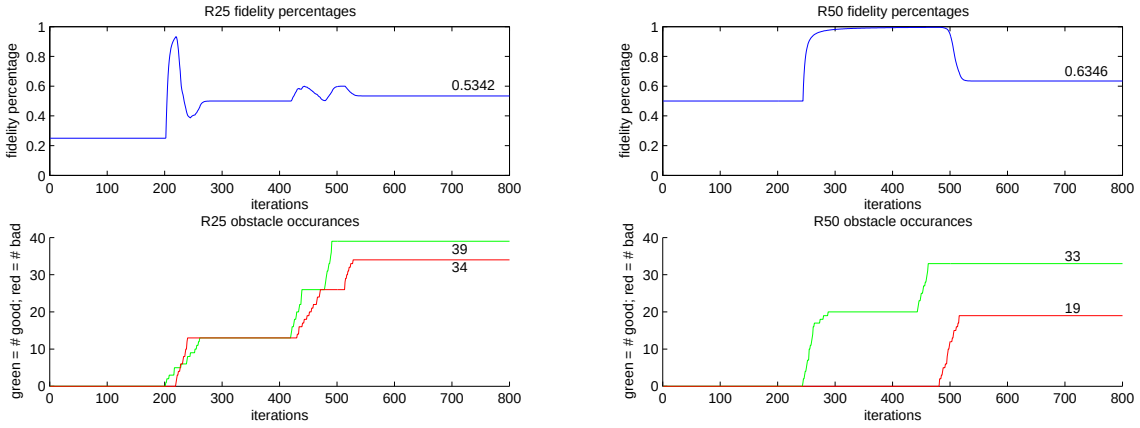


Figure 19: Sensor Fidelity in Open Field Simulation - Field

For instance, obstacle six only falls halfway into the outer sensor fidelity range. This means that if obstacle six were a "ghost" obstacle, then only six of its thirteen obstacles will be considered as "ghosts." The other seven will be unknown. In theory this makes sense because the vehicle has no knowledge of what lies outside of the range, but it makes it tricky to show that the vehicle is counting the obstacles correctly. For clarification, the number of good and bad obstacles have been included with the figures showing the sensor fidelity ranges, which are shown in Figure 20.



(a) 25% Fidelity Range

(b) 50% Fidelity Range

Figure 20: Sensor Fidelity in Open Field Simulation - Fidelity Ranges

It can be shown by calculations similar to those performed in the previous section that

the Bayesian updating process tends toward the probability of accuracy, which is what is expected. This simulation therefore shows how the proposed DMLM can be used not only to make decisions based on multiple objectives, but also to improve upon these decisions by updating its knowledge of the accuracy of its sensor network.

CHAPTER 5 CONCLUSIONS

The proposed Multi-Objective Decision Making and Learning Model presented in this thesis consists of two parts adapted from traditional game theory and psychological learning models. The first part of the model is a No-Regret Decision Making Model which is used to determine an optimal action based on a weighted sum of costs associated with each of the available actions. It turns out that this weighted sum of costs is the source of a problem. This problem is that each of the objective costs must be normalized, between zero and one, in such a way as to allow for the comparison of independent objective costs. This proves to be difficult and more research needs to be done in determining a standard way for doing this.

The second part of this model is a Bayesian Learning Model which can be used in multiple ways for different applications. In this thesis, two different ways for using this Bayesian Learning Model have been presented. The first application was to use the Bayesian Learning Model to update the weights associated with the objectives in order to account for errors present in the expected costs for each of these objectives. The second application for the Bayesian Learning Model was to update a set of sensor fidelity ranges that are used to calculated one of the objectives. Other applications exist, of course, and may even include using multiple Bayesian Learning processes for the calculation of multiple objectives.

Also included in this thesis is a Vehicle Navigation Model which incorporates both the No-Regret Decision Making and the Bayesian Learning into a tool used in path selection for Unmanned Ground Vehicles (UGVs). Within this model are examples of possible path characteristics which can be used to calculate the objectives for the UGVs. These characteristics have been adapted from Human Driver Models which are developed by transportation authorities for making automobiles safer for people.

REFERENCES

[1] Gordon Antelman, Albert Madansky, and Robert McCulloch. *Elementary Bayesian Statistics*. Cheltenham, UK: Edward Elgar, 1997.

[2] Rev. Thomas Bayes. An essay towards solving a problem in the doctrine of chances. 1763.

[3] Ulrich Berger. Brown's original fictitious play. *Journal of Economic Theory*, 127(1):572–578, July 2007. available at http://ideas.repec.org/a/eee/jetheo/v135y2007i1p572-578.html.

[4] G. Larry Bretthorst. *Bayesian Spectrum Analysis and Parameter Estimation*. Springer-Verlag, 1988.

[5] Delphine Delorme and Bongsob Song. Human driver model for smartahs. 2001.

[6] R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 1925.

[7] Geoffrey J. Gordon. No-regret algorithms for structured prediction problems. December 2005.

[8] Vijaykumar Gullapalli. *Reinforcement Learning and its Application to Control*. PhD thesis, University of Massachusetts, 1992.

[9] Tobias Karlsson. Terrain aided underwater navigation using bayesian statistics. Master's thesis, Linkoping University, 2002.

[10] K. Keith, M. Trentacoste, L. DePue, T. Granda, E. Huckaby, B. IlIbarguen, B. Kantowitz, W. Lum, and T. Wilson. Roadway human factors and behavioral safety in europe. 2006.

[11] Yiting Liu. Human driver model and driver decision making for intersection driving. In *in Proceedings of the IEEE Intelligent Vehicles Symposium, (Istanbul, Turkey)*, June 2007.

[12] Zhihua Qu, Jing Wang, and Clinton E. Plaisted. A new analytical solution to mobile robot trajectory generation in the presence of moving obstacles. *IEEE Transactions on Robotics*, vol.20(no.6):pp.978–993, 2004.

[13] Harri Valpola. *Bayesian Ensemble Learning for Nonlinear Factor Analysis*. PhD thesis, Helsinki University of Technology, Nov 2000.

[14] Jian Yang, Zhihua Qu, and Kevin L. Conrad. L2-norm optimal trajectory planning for mobile ground vehicles in a dynamically changing environment with both "hard" and "soft" obstacles. *submitted to Robotics and Autonomous Systems*, 2007.

[15] H. Peyton Young. *Strategic Learning and its limits*. Oxford University Press, 2004.