A*-Based Path Planning for an Unmanned Aerial and Ground Vehicle Team in a Radio Repeating Operation

Bryan Michael Krawiec

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

Master of Science
in
Mechanical Engineering

Kevin B. Kochersberger, Chair
David C. Conner
Mazen Farhood

May 2, 2012
Blacksburg, Virginia

Keywords: A*, Radio Repeating, Path Planning, UAV, Roadway Detection

A*-Based Path Planning for an Unmanned Aerial and Ground Vehicle Team in a Radio Repeating Operation

Bryan Michael Krawiec

ABSTRACT

In the event of a disaster, first responders must rapidly gain situational awareness about the environment in order to plan effective response operations. Unmanned ground vehicles are well suited for this task but often require a strong communication link to a remote ground station to effectively relay information. When considering an obstacle-rich environment, non-line-of-sight conditions and naïve navigation strategies can cause substantial degradations in radio link quality. Therefore, this thesis incorporates an unmanned aerial vehicle as a radio repeating node and presents a path planning strategy to cooperatively navigate the vehicle team so that radio link health is maintained.

This navigation technique is formulated as an A*-based search and this thesis presents the formulation of this path planner as well as an investigation into strategies that provide computational efficiency to the search process. The path planner uses predictions of radio signal health at different vehicle configurations to effectively navigate the vehicles and simulations have shown that the path planner produces favorable results in comparison to several conceivable naïve radio repeating variants. The results also show that the radio repeating path planner has outperformed the naïve variants in both simulated environments and in field testing where a Yamaha RMAX unmanned helicopter and a ground vehicle were used as the vehicle team.

Since A* is a general search process, this thesis also presents a roadway detection algorithm using A* and edge detection image processing techniques. This algorithm can supplement unmanned vehicle operations and has shown favorable performance for images with well-defined roadways.

Acknowledgments

While the thesis presented here may serve as one of my most significant pieces of academic work to date, this document is not simply one standalone accomplishment. Rather, it is both the end and culmination of a progression of experiences that have shaped my academic career and have built the foundation for my professional life as an engineer to come. When taking this point of view, it would be both egocentric and erroneous for me to claim that I am the only one to credit for my accomplishments. Instead, while they are indeed partly the result of my efforts, they are also the result of the efforts of all those who have acted as mentors, role models, and teachers to me. I have been fortunate enough to have had so much support from family, friends, and colleagues throughout my academic career and I would like to take the time to thank some of them here.

First and foremost, I would like to thank my parents for all of the love and mentorship that they have given me over the years. They have provided me with endless encouragement and have taught me countless lessons throughout my life. While an exhaustive list of these lessons would likely take me an eternity to transcribe, learning to have a strong work ethic and a thorough attention to detail were two of the most important factors in my academic journey. For these lessons and for so many others, I thank them wholeheartedly. I would also like to thank my identical twin brother Jeff who has always and continues to provide me with support, even if we still bicker like brothers from time to time.

I would next like to thank Dr. Kevin Kochersberger for his guidance throughout my graduate education. Aside from simply recruiting me to the Unmanned Systems Lab, he has given me a great deal of advice and has allowed me to spend two years in an environment where I have been able to continuously develop both intellectually and professionally. I would also like to thank Dr. David Conner and Dr. Mazen Farhood for their expertise and insights on path planning, controls, and optimization.

Additionally, I would like to thank my fellow USLers for providing me with technical advice and always challenging me to strive for success. More so than just being colleagues though, the members of the Unmanned Systems Lab have been incredible friends that I hope to keep

for a lifetime. Despite thanking them for the technical information that they have contributed to portions of my work, I would also like to thank them for the fun-filled distractions that they have often provided. They have surely made my graduate school experience both enriching and unforgettable.

Finally, since it is infeasible for me to thank everyone who has had a hand in my academic career individually, I would just like to take a moment to acknowledge all of my family, friends, mentors, and colleagues who have not been specifically mentioned here. No matter how large or how small of a part each one of them has played in my life, I truly appreciate the experiences that I have shared with them for they have shaped my life into what it is now and what it can be in the future that awaits.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

Acronyms

BLOS	Beyond-Line-of-Sight
COFDM	Coded Orthogonal Frequency-Division Multiplexing
DLL	Dynamic-Link Library
FIFO	First In First Out
GCS	Ground Control Station
GPS	Global Positioning System
GS	Ground Station
GV	Ground Vehicle
LIFO	Last In First Out
LOS	Line-of-Sight
NLOS	Non-Line-of-Sight
RC	Radio Controlled
TCP	Transmission Control Protocol
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
USB	Universal Serial Bus
USL	Unmanned Systems Laboratory

Chapter 1

Introduction

In the event of a disaster in a populated area, whether it is of natural, accidental, or deliberate origin, first responders need to quickly evaluate the post-disaster environment in order to assess damages and plan effective response operations. Unfortunately, post-disaster environments typically pose a number of dangers to the responders. These dangers may consist of collapsed buildings that block planned ingress or egress routes, structures that may be on the verge of collapsing, or harmful particulates in the air that may be hazardous to the health of the responders. Without the proper situational awareness about the presence of these dangers, response operations would be ill-informed and the lives of both civilians and the responders themselves may be put at risk. On the other hand, unmanned vehicles provide the distinct advantages that they do not suffer from the same health limitations as humans and they are relatively expendable when put in comparison to a human life. These characteristics, combined with the fact that unmanned vehicles can be equipped with a variety of sensing and communication equipment, make them ideal candidates for performing initial investigations of the potentially dangerous post-disaster environments. By taking advantage of both unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs), responders can survey the environment and use the gathered video, imagery, and data to plan effective response operations. In this way, when first responders execute response operations and enter the post-disaster environment themselves, they will have a more well-informed concept of their surroundings thereby leading to safer and more efficient response efforts.

With this motivation, the Unmanned Systems Lab (USL) at Virginia Tech has been focusing its research efforts on developing an unmanned vehicle system that can facilitate this post-disaster investigative process. While this project is grand in scope and contains a variety of components and operations, the topic of discussion in this thesis is a radio repeating operation that seeks to provide extended and enhanced communication between a ground vehicle and a fixed ground control station (GCS). In this work, a UAV is used as a radio repeating node for the communication link and by intelligently navigating the ground vehicle and UAV cooperatively, the ground vehicle can reach a desired goal location while a healthy communication link to the GCS is maintained.

1.1 Motivation

Unmanned ground vehicles have seen a number of advances throughout their history. Vehicle design, guidance and navigation algorithms, and sensor suites have all improved significantly in recent years causing them to have increased utility for investigative tasks. Despite these developments though, unmanned ground vehicles still suffer from one major flaw: They typically require communication with a ground control station in order to effectively accomplish a mission. More so, many missions involving UGVs are carried out in complex environments where buildings, obstacles, or hills in the terrain exist between the ground station and the UGV. Since radio communication links can degrade substantially in non-line-of-sight scenarios between the ground control station and the UGV, a naïve approach to the navigation of the vehicle could result in data packet errors or even a loss of the communication link entirely. For a semi-autonomous ground vehicle where an operator must send control commands to the UGV, this flaw is obvious. If the radio link is lost, control commands can no longer be sent to the vehicle and the mission cannot be completed. In the case of a fully autonomous ground vehicle, a mission usually consists of having the vehicle investigate an environment instead of response personnel themselves. In this setting, data in the form of GPS locations, video streams, and imagery can be sent to the ground control station throughout the mission. If a radio link is degraded, the data received at the ground station may be degraded as well and if the radio link is lost, the operators can no longer take advantage of the situational awareness provided by the UGV. With these scenarios as motivation, it is abundantly clear that the maintenance of this communication link is critical in unmanned ground vehicle operations.

With this communication link limitation, the question naturally arises of how to effectively navigate the ground vehicle while also maintaining a strong communication link with the ground station. Of course the ground vehicle could be navigated in line-of-sight conditions or an operator could be conscious of the radio signal strengths as a mission progresses and attempt to correct the vehicle's path when signal strengths start to become low. However, neither of these solutions are ideal in that the former vastly limits the regions that a ground vehicle can navigate to and the latter adds the potential for human error into the operation. A better alternative would be to add a UAV to the operation thereby creating an unmanned air and ground vehicle team and extra degrees of freedom in the navigation process. By incorporating an aerial component to the operation, communications can be relayed from the ground station, through the aerial vehicle, and to the ground vehicle. This allows the aerial vehicle to act as an antenna in the sky, which can increase the effective communication range of the ground vehicle to regions beyond-line-of-sight from the ground station. This provides increased flexibility in navigation of the ground vehicle.

However, even if this concept is employed, a naïve choice for the position of the aerial vehicle could still result in a number of non-line-of-sight scenarios between the aerial vehicle and the ground vehicle which can provide similarly degraded communication links. With this in mind, this work seeks to address this issue by formulating this radio repeating operation as a path planning problem. Rather than simply planning a path for the ground vehicle and arbitrarily choosing a location to place the aerial vehicle, this thesis discusses a path planning algorithm to cooperatively navigate the aerial and ground vehicles together. The algorithm is formulated so that the resulting paths guide a ground vehicle to a desired goal location while maintaining healthy radio links between the ground vehicle, aerial vehicle, and ground station in the process.

1.2 Proposed Method and Organization

In order to accomplish the desired radio repeating task, this scenario is treated as a path planning problem whose solution is the set of resulting paths for the aerial and ground vehicles. For this, the A* path planning algorithm was chosen as the foundation for this work and the bulk of the thesis discusses details about the algorithm itself and its use in the radio repeating operation.

With this, the remainder of the thesis is organized in the following way. First, Chapter 2 presents a literature review that summarizes relevant work to the radio repeating scenario. While extensive literature exists in the path planning, radio modeling, and unmanned vehicle areas, the discussion that is presented has been focused to include only work in research areas that are relevant to the radio repeating scenario. As such, Chapter 2 includes details relating to A*-based path planning, well-accepted means of quantifying radio link performance, and work in the area of communication-aware path planning. Additionally, since evaluating a cost function for the radio repeating scenario is dependent upon the geometry of the search environment, Chapter 2 also includes details on relevant work in roadway detection from aerial imagery that serves as a foundation for the work of Chapter 6.

Chapter 3 then presents the A* path planning method and a number of theoretical details on its implementation. First, the general concept of a discrete search process is discussed and the A* algorithm is introduced. Then, since a completely optimal A* implementation could be too computationally complex for online calculation even with the use of efficient data structures, methods are discussed to sacrifice optimality slightly in an attempt to speed up the search process to a more computationally efficient state.

After this foundation has been set, Chapter 4 provides an in depth discussion on the formulation of the radio repeating scenario as a path planning problem. It starts by providing results from radio testing and discusses various features for modeling radio link health quantitatively. Chapter 4 then continues by presenting the state space, cost function, and simulation results from testing the radio repeating path planner.

Chapter 5 then briefly discusses the implementation of this path planner on actual vehicle platforms. For the purpose of testing, a Yamaha RMAX helicopter was used as the aerial vehicle and a generic truck was used as the ground vehicle. An overview is given of the system components and field testing results are discussed.

After this, Chapter 6 branches in a separate but related direction. As alluded to above, in order to effectively apply the path planning to a specific environment of interest, a terrain map is required as an input to the path planner. For the Unmanned Systems Lab, this requirement is not too limiting since the lab also possesses a stereovision boom that can facilitate the construction of a 3D terrain map when it is flown over an area of interest. For a group without such stereovision capabilities though, the generation of a terrain map may be difficult and therefore, Chapter 6 investigates a technique to use the A* algorithm to perform roadway detection on aerial imagery. While this is by no means a substitute for a terrain map, it is an initial attempt to potentially augment the unmanned vehicle operations and is worth discussion.

Finally, Chapter 7 presents a conclusion and discusses recommendations for potential future work.

Chapter 2

Literature Search

The development of a system capable of performing a radio repeating operation involves a number of aspects. Not only does planning desired trajectories for the vehicles in the operation require consideration, but sufficient modeling and quantification of the radio links' performance needs to be considered as well. Therefore, this chapter presents a literature review that describes relevant prior work in both the path planning and radio modeling areas of research. Coupled with these descriptions, discussions are also given to provide justification for the thesis work and to describe how the thesis work relates to published methods in the literature. Additionally, for completeness, a number of other efforts in communication-aware robotic operations are presented to provide perspective on the subject. Lastly, since Chapter 6 deals with the topic of image-based roadway detection, a brief overview of work in the roadway detection area of research is presented as well. By considering these different areas of research, this chapter aims to provide insight into currently available methodologies that relate to this work and also to set the stage for the work described in the remainder of this thesis.

2.1 Path Planning Methods

Path planning for vehicle navigation is a deeply studied field of research and numerous methods exist for generating desirable motion trajectories for these vehicles. Depending on whether the path planning problem is formulated as discrete or continuous, whether optimal or just feasible paths are required, and whether there is uncertainty inherent in the problem, different path planning techniques exist to handle the various scenarios. As an example, Steve LaValle [30] has written an entire textbook dedicated to the subject of path planning and the author provides an extensive discussion of many of the types of path planning techniques which are applicable for a number of the mentioned scenarios. However, as Ferguson points out [19], while the two most popular distinctions for planning come in the form of heuristic-based algorithms and randomized algorithms, when the dimension of the planning problem is low, the heuristic-based approaches are usually preferred due to their ability to provide optimality or suboptimality guarantees on the resulting paths. Considering the radio repeating scenario in this work, it is treated as a relatively low dimensional path planning problem and since there is great interest to optimize the quality of the path with respect to the performance of the radio links between the vehicles, the heuristic-based methods were chosen for this work. More so, due to the fact that GPS paths are desired for the vehicles, which can easily be represented discretely, it was chosen to use discrete methods as well. Additionally, the A* algorithm was chosen specifically due to its optimality guarantees [21] and therefore, the rest of the literature review on path planning focuses on this particular method and its extensions.

The A* algorithm is considered to be a general graph search method from a start state to a goal state where transitions between the states are quantified by a cost function [21]. Throughout the path planning process, this algorithm seeks to minimize the cost function as it progresses from the start to the goal. This allows the A* algorithm great versatility since the cost function can be customized to fit a specific path planning scenario. Because of this and the fact that it can be easily implemented in grid-based environments, the A* algorithm has seen widespread use in a number of scenarios. To provide a few examples, the A* algorithm has seen uses in the planning of trajectories for multiple degree of freedom robotic arms [31], sliding-tile puzzles [21], and planning through grid-based maps [11]. Also, in related work, Rose [41] employs the A* algorithm for ground robot navigation over irregular terrain where the cost function is formulated from distance measures that combine the actual distance traveled by the robot as well as equivalent distance measures to penalize turns and climbs that the robot has to make throughout the search.

Due to its widespread popularity, not only has a significant amount of research gone into the basic A* algorithm itself, but also into different modifications and enhancements to the algorithm to provide more efficient operation in complex scenarios. For example, Hansen and Zhou [21] discuss an inflated heuristic variant of the A* algorithm, known as Weighted A* (WA*), where inflating the heuristic for the search provides significantly enhanced computational efficiency at the expense of optimality. Although the true optimality of the resulting paths are sacrificed in this WA* search, the resulting paths are guaranteed to be ϵ-suboptimal for a heuristic that is inflated by a factor of ϵ where the resulting paths are at worst ϵ times the cost of the optimal path [19]. Also, as Hansen and Zhou [21] point out, this method is well suited for planning problems where a number of close-to-optimal paths exist since the WA* algorithm can find these close-to-optimal paths in only a tiny fraction of the time that it takes to generate an optimal path.

Furthermore, Hansen and Zhou also describe an Anytime A* (AA*) [21] algorithm that focuses on using a highly inflated WA* search to provide an initial quickly available suboptimal path from the planner to be used immediately and then as time permits, the search is refined to provide more optimal results at a later time. Likhachev [31] takes this work a step further with his Anytime Repairing A* (ARA*) that provides a more efficient anytime implementation. With the computational benefits gained by the modifications of the basic A* algorithm to the WA*, AA*, and ARA*, Likhachev has been able to demonstrate their applicability on up to a 20 degree of freedom robot arm [31], which is generally a very high-dimensional path planning problem for heuristic-based searches.

In a similar but related vein, other researchers have also produced papers discussing Dynamic A* or D* algorithms. These algorithms, which include D* [44], D* Lite [25], and Focussed D* [45], seek to provide efficient means of incrementally repairing the solutions of the A* algorithm as new information about an uncertain environment is detected. While these variants are powerful individually, Likhachev has combined the re-planning ability of the D* variants with the anytime performance of the Anytime algorithms to provide an algorithm with qualities of both. This is called the Anytime D* (AD*) [31] and this particular variant has been implemented and tested on the Boss vehicle which was victorious in the DARPA Urban Challenge [18]. As an additional note, it is also interesting that [18] describes a method of performing an initial two dimensional search to build up a more accurate heuristic and to create a more well-informed higher-dimensional search [18]. This, coupled with similar observations in [19], form the foundation for the discussion of a heuristic look-up table in section 3.6.

While these A* variants are useful in situations with unknown environments or where time is of the most absolute essence in planning operations, this work instead chooses to focus on the simpler WA* algorithm only. This choice is supported by the fact that in a radio repeating scenario, typically a path can be generated either prior to sending the vehicles on the operation or while the aerial vehicle is taking off. Either of these times provide sufficient time for a WA* planning operation. Additionally, re-planning may only be needed sparsely throughout the operation. Therefore, when weighing the computational benefits of the ARA* algorithm versus the repeated execution of the WA* algorithm [31], [19], the WA* is preferred for its simplicity. Also, since this work is predicated on a successful stereovision terrain map being produced [26], it is assumed that the terrain map that is generated is an accurate representation of the planning environment. Therefore, D* performance is not necessary for this work. Despite this rationale, as noted by [19], if changes or updates to the environment occur at locations that are not close to a vehicle being planned for, the D* Lite algorithm may be less efficient than the A* algorithm itself. Although not explicitly considered in this work, if the helicopter could update maps as the radio repeating operation progressed, it is definitely possible for new environment information to be obtained away from the ground vehicle and therefore the A* or WA* algorithms may be just as efficient as their D* or AD* counterparts. This provides further justification that even though many advanced planners exist, the WA* algorithm is sufficient for the remainder of this work.

2.2 Radio Link Quantification

As noted in the previous section, the generic A*-based path planning techniques can be catered to a specific path planning scenario by the proper formulation of a cost function for the search process. This cost function effectively quantifies the transitions between different cells or states in the search. Therefore, since in the radio repeating scenario the health of the radio links between the vehicles would like to be optimized, it is necessary to form a cost function that involves a moderately accurate quantification of the radio links' performance. Since radio and antenna modeling are very important concepts in the engineering community, much literature is devoted to studying the properties and mechanisms involved in radio communication. In fact, a number of texts, including [5], [24], and [40] are devoted to this subject.

Despite the fact that many of the aspects of radio communication are documented and agreed upon, the subject of path losses still remains an actively studied area of research. These path losses define the losses that occur through electromagnetic propagation through an environment between the transmitting and receiving antennas. As one might expect, this is a complicated factor to quantify and two distinct schools of thought exist for determining path losses in a radio link.

The first school of thought in modeling electromagnetic propagation for radio communications is the deterministic one. This area of research deals with accurately predicting the propagation characteristics for different types of environments. For example, Toporkov et al. [46] have performed studies and simulations of scattering from random rough one-dimensional surfaces, Brown [7] solves the case of a coherent wave propagating through a sparse concentration of particles, and Brown and Curry [8] present their work on modeling wave propagation through foliage. Each of these cases could be characteristic of a post-disaster environment where rubble may be littered on the ground, particulates may be floating in the air, or other obstacles may occur between the transmitter and receiver. Therefore, each of these scenarios, and likely others as well, would require analysis for a general post-disaster environment. Unfortunately, while the scope and results of each of these papers differ, the common thread between them is that accurate modeling of electromagnetic propagation involves solving complicated equations that are based on parameters that are very specific to the propagation environment being studied. While these deterministic methods are important in the radio community, they are infeasible for path planning for two primary reasons. First, in order to perform the path planning, a cost function must be evaluated hundreds or thousands of times in a very short time frame and therefore must be simple. Secondly, for the purposes of disaster response, the path planning will likely be performed in areas where the propagation environment is poorly known and therefore deterministic calculations would be impossible to accurately perform. This reasoning is complimented by Neskovic et al. [32], who note that the accurate prediction of radio signal strengths is a very complex task. The authors continue to note that the implementation of deterministic models tend to require extensive databases of environment characteristics which are often impossible or overly time consuming to obtain. This coupled with the computational complexity of the deterministic methods makes them infeasible for rapid vehicle path planning.

Rather than this, another more simplified method of predicting radio path losses is to use empirical methods. As noted in [32], these models are based on extensive measurements in different propagation environments. Because of this, all of the complicating factors involved with the deterministic calculations are implicitly taken into account. While the authors note that the accuracy of these models is surely based on the similarities between the environment of interest and where the measurements were taken, they serve as computationally simple tools for estimating path losses. A paper by Hata in 1980 [22] is one of the earliest and most widely cited papers on empirical path loss models. In this paper, Hata uses the data gathered by Okumura [38], which consists of extensive tests of path losses in various environments in Japan, to formulate simple algebraic equations that can be used to predict path losses. These equations are based on simple and readily available quantities like ground antenna height, aerial antenna height, frequency of communication, and distance between the antennas. Due to the fact that these parameters are generally available, the empirical models formulated by Hata are easily implementable in a vehicle planning scenario. More so, different empirical models are given for different types of environments such as a large urban environment, a medium urban environment, a suburban environment, and an open environment. By having a variety of different models based off of experimental data, the Hata models have become popular in estimating the performance of a radio link.

Also, in a similar mentality, Walfisch and Bertoni [47] published a theoretical model for path losses in an urban environment by treating the urban environment as a series of cylindrical rows and treating diffraction as the primary propagation mechanism. These authors compared their model with two data sets, one being the Okumura data from Tokyo [38], and found reasonable agreement. Shown in Figure 2.1 is a plot of some of the path loss models described above and as can be seen, they are all logarithmic-based functions that just have different rates of increase or vertical offsets than the typical Free Space Path Loss model [40].

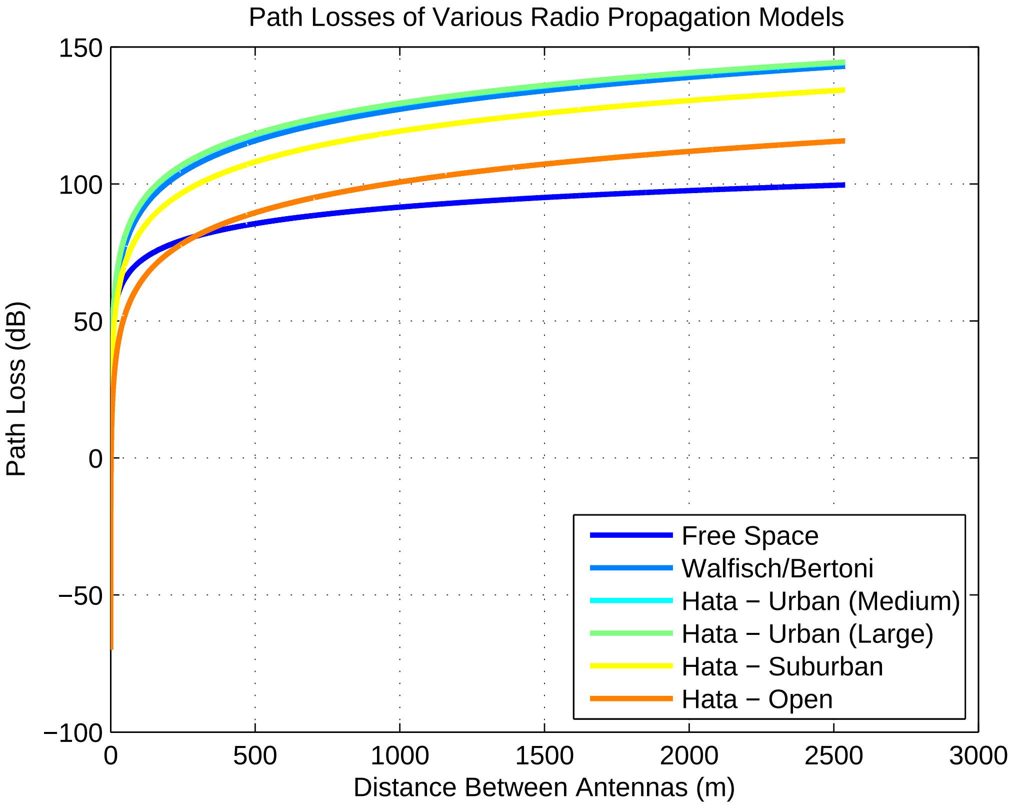


Figure 2.1: Path losses for various empirical path loss models [22], [40], [47]

As can be expected, with the simplicity of these empirical models, they and others have been studied and evaluated to determine their applicability to different propagation environments and also for different frequencies of communication than the ranges that they were originally tested in. For example, in 1988, the IEEE Vehicular Technology Society Committee on Radio Propagation published a paper [2] reviewing the numerous models including the Free Space, Okumura (Hata), and a number of other models like the Egli, Carey, Longley-Rice, and Lee. After their analysis, they claimed that the Hata models are attractive due to their simplicity, but they suggested that the Hata models should be used with caution due to the facts that the qualitative descriptors for the environment types, like urban and suburban, differ between Japan and the United States and also because environment specific factors have been abstracted away in the models resulting in averaged path loss predictions. This thesis adopts the Hata models as a means to predict average path losses but testing has been done to determine which models were most applicable to the radios and the testing environment. Lastly, it should be noted that studies continue to be performed as in [1] to both validate and extend the applicability of empirical models to different frequency ranges and environment types.

2.3 Communication-Aware Robotic Operations

While the above sections describe the main body of literature that is most relevant to the radio repeating operation, this section provides a brief overview of some other efforts found in the literature that are centered around communication-aware robotic operations. Although the literature discussed here is not strictly related to the radio repeating operation, this section intends to provide perspective by presenting additional work that involves radio communications and robotics.

First, as Hsieh and Kumar discuss in a 2004 paper [23], much of the literature on multi-robot operations up to that date focused on areas like control and planning, while naïvely assuming that the robots can freely communicate with each other regardless of their configuration. Therefore, these authors discuss a method of formulating an exploration problem in order to create a radio signal strength map that can be used in subsequent planning operations. This work was extended in [10] to be implemented with a multi-robot team. In this work, each robot was assigned a list of waypoints and measurements of radio signal strengths were taken between the pairs of robots. These measurements then formed a radio connectivity map that the authors envision being used to plan paths for the multi-robot team in other operations. Additionally, [3] discusses very preliminary work to perform radio mapping with robots and to understand the relationship between sources of interference and changes in radio signal strengths. While it is true that generating radio maps would be beneficial for a planning operation, many times in response operations time is of the essence and it is impractical to waste precious time generating these radio maps. Although this mapping work is used as inspiration, the work of this thesis seeks to perform communication-aware path planning without necessarily having a complete radio map of the area and therefore, the incorporation of a potential online radio mapping capability is only discussed as a future recommendation for this work.

Additionally, in a recent 2010 publication, the GRASP lab extended its work [20] to use this radio mapping concept along with Gaussian processes to localize a radio signal source in an environment. While this localization work is outside of the scope of this thesis, the paper points out several important concepts worth noting. The first is that received radio signal strength is an important parameter in maintaining a healthy communication link and that it has been heavily correlated to the bit error rate of a communication link. Therefore, this will be used as the primary metric for radio signal health in the radio repeating work. Also, it shows that relevant work is being done in the area of communication-aware path planning and this supports the importance of the radio repeating work discussed in this thesis.

The idea of using repeater links to augment a ground robot operation is also seen in the literature as well. The author of [37] discusses the utility of using a UAV as a communications relay for warfighters. In this paper, important observations are made that setting up a communication infrastructure in a region is difficult and that often times ground units can quickly get out of the range of a static communications system. The author asserts that using a UAV as a communication node can abate these issues and the idea serves as a motivation for the radio repeating work. Another example of work that relates to this repeater concept is seen in [16] and [15], where a controller for a fixed wing UAV was developed to allow the UAV to maintain either constant signal-to-noise ratio orbits from radio sources or follow a gradient path to the best signal-to-noise ratio location. Additionally, another example of this aerial repeater concept comes from [39] and [4] where this aerial communications relay concept is extended to multiple UAVs. While the idea of swarming concepts is indeed a natural extension of the above concepts, they will not be pursued in this thesis. The reason for this decision is that a network of UAVs would provide a more area-based coverage while the work of this thesis focuses on point-to-point communications between vehicles.

Lastly, considering just ground-based applications, another application of the radio repeating concept comes from [33], [34], [35], and [36], which are members of a series of papers from 2002 to 2004 where a group of slave ground robots are used as repeater nodes as a lead ground robot investigates an indoor environment. Essentially, the slave robots are deployed as a bread crumb trail, per se, to enhance the radio link between a remote operator and the lead robot. The first paper in the set discusses initial work while the rest progress to discussing this concept with varying levels of sophistication. While this work is limited to ground vehicle operations, its philosophy of using additional vehicles to enhance radio communications is central to the radio repeating work of this thesis.

2.4 Roadway Detection

Although not applicable to the radio repeating operation directly, Chapter 6 discusses a use of the A* path planning technique for roadway detection that could be useful in generating traversability maps for a similar operation. Therefore, this section provides a somewhat brief overview of recently published methods for roadway detection.

Roadway detection is a popular field of research and a number of different methods exist for attempting to extract roadways from aerial images. The authors of [48] for example use thresholding, dilation, and filling operations to determine potential roadway estimates. Additionally, after these operations are performed, the most likely roadway candidate region in the filtered image is chosen as the one with the smallest square root of area to perimeter ratio and a snaking method is used to trace out the selected roadway candidate. While there are a number of image processing techniques used in [48] that could each be discussed on its own, one of the most important details to note from this paper is that in order to effectively detect a roadway in an image, an image processing algorithm must employ defining features that are unique to that roadway region. In [48], by using the square root of area to perimeter ratio, the algorithm essentially determines the most long and slender of the regions in the image as a roadway. This provides favorable results for the algorithm although the processing times of the complete algorithm are relatively large. As an example, [48] shows computation times ranging from 2.04 seconds for small images (142 by 129 pixels) and up to 24.98 seconds for larger images (600 by 348 pixels).

Another paper [12] also takes a geometric approach to roadway detection. This method first converts the image from its basic color representation to an intensity representation where each pixel is defined by one scalar value as opposed to a vector of values. After this conversion, the authors apply a gradient filter to the image and employ a line detector to gather line segments that could potentially be roadway edges. Then, in a similar vein as the above paper, this paper uses an opposite gradient feature as its defining roadway characteristic and deletes all detected edges that do not exhibit this trait. This opposite gradient concept is based on the idea that in an image, a roadway is tyically surrounded on both of its sides by a relatively uniform background of a different average intensity than the roadway area itself. Because of this, when considering gradients, one side of the road will have a gradient from the background to the road in one direction and the other will have a gradient from the road to the background which will be in the opposite direction. While the resulting extracted roadway networks produced from this paper are not completely well-connected, the concept of defining a roadway based upon opposite gradient information at either side of the roadway is a moderately unique feature of a roadway that is exploited in the work of Chapter 6.

In addition to the two geometric approaches described above, color content of pixels in an image can also be used for detecting roadways in images. For example, [43] discusses a method of extracting roadways by comparing the colors of image pixels to determine which pixels in the image are a portion of a roadway. The comparisons are made between the image pixels and a set of expected roadway colors that are obtained from an initial semi-automatic training process. For this method, the authors use color as their defining roadway characteristic and assume that the roadways are of relatively uniform color throughout the image. Additionally, this paper makes use of Canny edge detection to produce the final detected roadway results once the color comparisons are complete [9]. As noted by the results of this paper, relatively strong results were obtained from testing this algorithm on a trial set of imagery. In these tests, 81.64% of the roadway regions in the trial imagery were positively detected and only 3.94% of pixels were wrongly detected as a roadway. This provides confidence that color can be a strong identifier for roadway regions in an image and this will be taken advantage of in Chapter 6.

Lastly, another method worth noting is the Hough transform. This method is a means for finding straight lines in images and is a widely used and well-accepted line detection method [17]. While this method is mentioned in papers relating to roadway detection, as in [43] for example, the Hough transform is not quite as useful when roadways have a curvature to them or do not persist as completely straight lines in an image. Therefore, this method, while worth discussion, is not adopted in this thesis. Rather, details of the aforementioned techniques serve as inspiration for the roadway detection method discussed in Chapter 6.

Chapter 3

A* Path Planning

As noted in the introduction and literature review, path planning is the concept of searching through a state space to find a path from a start state to a desired goal state. While many path planning techniques exist, this thesis will primarily deal with graph search techniques like the A* algorithm. When applying this search to a grid, which is done for the rest of this thesis, this methodology essentially represents a state space as an orderly grid and systematically searches through that state space for an optimal path to a goal state. The benefit of this method is that it is orderly, systematic, easily implemented with array based data structures, and can handle a general class of nonlinear problems, which is typically the case for navigation in obstacle-rich environments. The remainder of this chapter describes the foundations of the A* Algorithm and sequentially steps through improvements made to its basic implementation that offer computational benefits for the search process. By the end of the chapter, the formulation of the search is altered in such a way that makes it amenable to rapid execution which is a desired result for implementation with a vehicle team.

3.1 Discrete Forward Search

Starting off with the basic details that are relevant to the foundation of the A* search, a general continuous time nonlinear dynamic system can be modeled in the form below.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

Here $\mathbf{x}$ is the state vector for the system that contains information regarding the dynamic configuration of the system, $\mathbf{u}$ is an input vector which describes the inputs or actions that can be applied to the system, $\mathbf{f}(\cdot, \cdot)$ is a vector valued function that describes how $\mathbf{x}$ and $\mathbf{u}$ affect the dynamics of the system, and $\dot{\mathbf{x}}$ is the derivative of the state vector $\mathbf{x}$. More so, the set of all possible states is known as a state space, denoted as X, and the set of all possible inputs to the system is known as the action space, denoted as $U(\mathbf{x})$. Note that for a continuous system, the state space is a continuous set and exhibits the property that at any instant in time, the state of the dynamic system is contained within the state space. That is, $\mathbf{x} \in X \ \forall t$. The action space also exhibits the property that $\mathbf{u} \in U(\mathbf{x}) \ \forall t$. However, it must be noted that for a general system, the actions that can be performed on the system may be dependent on the current state of the system. Therefore, the action space contains an $\mathbf{x}$ dependency. Using this mathematical form, a dynamic system that can be described in terms of ordinary differential equations can be compactly written in this way [30].

While this continuous time model provides a convenient and compact representation of a system, planning a path through its state space would be rather difficult due to the continuous nature of the spaces involved, the potentially infinite inputs $\mathbf{u} \in U(\mathbf{x})$ that could be applied to the system at any time, and the integrations that must be performed for each state transition. Therefore, instead of representing a dynamic system in a continuous domain, the system can instead be represented in a discrete form as shown below.

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}, \mathbf{u})$$

Here, $\mathbf{x}$ still represents the state of a system and is contained within the collection of all possible states X. Also, $\mathbf{u}$ still represents an action that can be applied to a system and is in the set of all possible actions $U(\mathbf{x})$. Additionally, $\mathbf{f}(\cdot, \cdot)$ still represents a function that describes how the current state $\mathbf{x}$ and current input $\mathbf{u}$ affects the dynamics of the system. However, rather than having an $\dot{\mathbf{x}}$ on the left hand side of the equation that represents the derivative of the state vector, now the system model has a $\mathbf{x}^{'}$ on the left hand side of the equation which represents a successor to the current state under the action of the current input. That is, $\mathbf{x}^{'}$ is the next state in time and also belongs to the state space X. With this mathematical form of our system, a few key features can be noted. First, since the system is represented in this discrete manner, the state space X consists of a discrete set of states and is either finite in size, if there are boundaries placed on the domain of the problem, or countably infinite, if no such boundaries are in place. The set of possible actions at each state $U(\mathbf{x})$ are usually quantized as well into a finite set of possibilities. Additionally, instead of needing to perform integrations on $\dot{\mathbf{x}}$ to find the state transitions, only $\mathbf{f}(\cdot, \cdot)$ must be evaluated to determine the next state in the sequence of states. By looking at these advantages, this discrete formulation of a dynamic system is favorable in terms of computational effectiveness and will form the foundation of the path planning algorithms used throughout the thesis [30].

With this formulation in place, it is important to define a state space X and an action space $U(\mathbf{x})$ for the system of interest. In general, the state space can be of any form as long as it contains all of the states of the system. Similarly, the action state $U(\mathbf{x})$ can contain any actions on the system that are appropriate for the planning process. For example, in the Traveling Salesman Problem, which is a classical NP-hard optimization problem that asks how to find the shortest path between a set of cities while only visiting each city once, the state space X can be the set of cities that can be visited and the action space $U(\mathbf{x})$ can be the potential routes that the salesman can take from the city he or she is currently at to the next city [30]. Therefore, referring to the Figure 3.1, the state space would be $X = \{1, 2, 3, 4\}$ and the action space at $\mathbf{x} = 1$ for example would be $U(1) = \{R_1, R_2\}$.

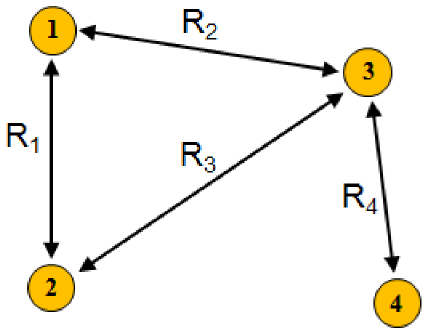


Figure 3.1: Example of a traveling salesman problem

Looking at the traveling salesman example, the state space X is just a collection of locations and the action space $U(\mathbf{x})$ is very dependent upon the current state x. From a computational perspective, this would be tedious to implement since a different set of actions would have to be stored for each state $\mathbf{x} \in X$. For a small state space, this is simple enough, but it would be cumbersome in a large state space. It would be much more convenient if an action space could be independent of $\mathbf{x}$ and one common set of actions could be applied at each state. From a computational perspective, this is easier to implement since only one set of input actions U needs to be stored for the entire state space.

Considering the above arguments, there is good motivation to the formulate the state space as an orderly grid and the action space as being independent of $\mathbf{x}$. With this grid-based formulation, the state space can be easily stored in an array and only one set of actions needs to be stored as well. Therefore, even though the A* algorithm is applicable to an arbitrary discrete state space and action space, the grid-based formulation will be strictly incorporated throughout the remainder of this thesis. In a grid-based format, the state space can now be conveniently defined as

$$X = \{\mathbf{x} \in \mathbb{R}^n |\ |\mathbf{x}^{'} - \mathbf{x}| = \mathbf{r}, \mathbf{x}^{'} \in X\} \ \ , \ \ where \ \ r_i = \begin{cases} 0 \ , \ if \ x_i = x_i^{'} \\ \Delta x_i \ , \ if \ x_i \neq x_i^{'} \end{cases} \ , \ i = 1, 2, .., n$$

Here x_i and r_i represents the i^{th} components of $\mathbf{x}$ and $\mathbf{r}$ respectively, and Δx_i represents the resolution of the grid for the i^{th} coordinate. Similarly, the action space can be defined as the movements to each neighboring state in the grid. That is,

$$U = \{\mathbf{u} \in \mathbb{R}^n |\ u_i \in \{-\Delta x_i, 0, \Delta x_i\}\ ,\ i = 1, 2, ..., n\} - \{\mathbf{0}\}$$

Shown below in Figure 3.2 is an example of a simple 3 by 3 grid where the state space and action space are shown.

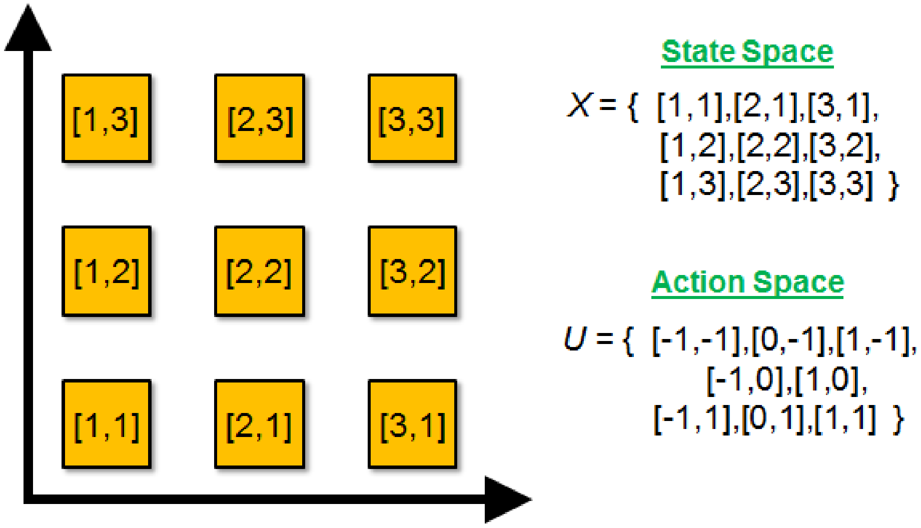


Figure 3.2: Example of a grid-based state space and action space

With this grid-based state space and action space in place, the problem of planning a path through the state space can now be discussed. The idea of path planning involves trying to find either a feasible or an optimal trajectory through the state space from a start state (or node) to a goal state. Written mathematically, it is desired to find a feasible or optimal path from an initial state $x_I \in X$ to a goal state $x_G \in X$ or a goal set $X_G \subset X$. There are many different ways that this task can be accomplished and many search algorithms have been created and analyzed to assess their ability at performing this task. Regardless of this fact, many of these algorithms act in a general forward search manner and are different only in the way that they decide to handle various features of the search. As noted in [30], a generic forward search functions according to the following algorithm.

Algorithm 1 Generic Forward Search Algorithm, [30]

```
1:  $Q.Insert(x_I)$ and mark $x_I$ as visited
2:  while $Q$ not empty do
3:      $x \leftarrow Q.ChooseState()$
4:      if $x \in X_G$ then
5:          return SUCCESS
6:      end if
7:      for all $u \in U$ do
8:          $x^{'} \leftarrow f(x, u)$
9:          if $x^{'}$ not visited then
10:             Mark $x^{'}$ as visited
11:             $Q.Insert(x^{'})$
12:         else
13:             Resolve duplicate $x^{'}$
14:         end if
15:     end for
16: end while
17: return FAILURE
```

As can be seen by inspecting the algorithm above, the forward search algorithm follows a systematic process in order to determine a feasible path from the initial state x_I to the goal set X_G. In words, this algorithm acts according to the following process below.

1. Initialize priority queue Q of states and start the main loop (Lines 1 and 2)

2. Choose a state based on some priority from the queue Q (Line 3)

3. Check if selected state is in the goal set X_G and end the search if it is (Line 4)

4. Calculate the successors of x and add them to the queue Q (Line 7)

5. Repeat the process of choosing states from Q and calculating successors until a goal state is reached

Although this process outlines a forward search, a number of items need to be clearly defined before this method can be directly implemented. First, Q defines a priority queue in which states are stored according to some metric. Two natural questions arise from this. The first is a question of how the priority queue deals with its elements. For example, if the priority queue uses a First-In-First-Out (FIFO) [30] scheme, states that enter the queue first with $Q.Insert$ will be the first ones to exit the queue when $Q.ChooseState$ is called. This will result in a much more breadth first type of search that will expand radially outward from the initial state x_I. On the other hand, if Q uses a Last-In-First-Out (LIFO) scheme, more recently inserted states will be pulled out of the queue and evaluated. This results in a depth first type of search where the planning effort becomes very focused and narrow. The Figures 3.3 and 3.4 below give graphical representations of the searches that could result from these two types of queue mechanisms. It should be noted though that while the depiction of a LIFO scheme seems to be much more direct than the FIFO scheme, if dead ends exist in the environment, the LIFO scheme can easily mislead the search into those dead ends causing it to not be as direct as it initially may seem.

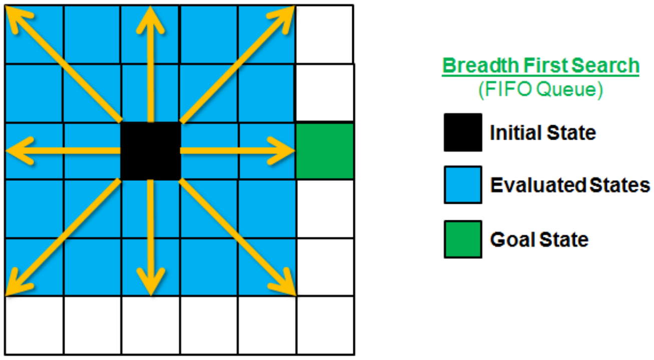


Figure 3.3: Example of a breadth first search produced from a FIFO queue

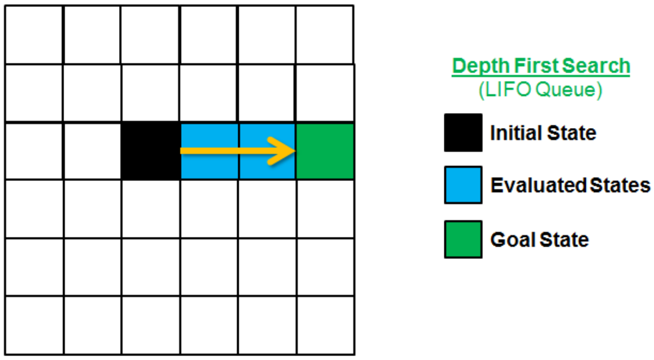


Figure 3.4: Example of a depth first search produced from a LIFO queue

In addition to these the FIFO and LIFO schemes, instead of rating a state based on when it was encountered in the search, states can also be rated based upon a cost associated with the transitions between states and can be stored in Q according to their costs. This idea defines the second question that needs to be resolved and this question of how to define the transition costs is one that is very dependent upon the objectives of the search being performed. In general terms, a transition cost, denoted as $c(x, x^{'})$ is defined for the transition from state x to state $x^{'}$ and a states are stored in the priority queue Q based on the total cost of transitioning from the start state x_I to the state $x^{'}$ [21]. By rating states in this manner, the search can be guided to not just find a feasible path through a state space but instead optimize some criterion for the search. In this way, these forward searches can be catered to a specific objective and this idea is what the A* algorithm and the remaining work of this thesis attempt to do.

3.2 Dijkstra's Search and the A* Algorithm

The discrete forward search discussed in the previous section was described as a general means of searching through a grid-based environment. In this section, Dijkstra's search will be presented as a specific case of the generic forward search and this section will discuss theoretical and practical details about its performance. Additionally, this section will extend the theory of Dijsktra's search to the A* algorithm which provides the same optimality guarantees as Dijsktra's search but is more computationally efficient [30].

As noted in the previous section, there are a number of ways to store states in the priority queue Q. Dijkstra's search operates by assigning each state in the queue a cost and the queue stores the costs in order of minimum cost. The cost assigned to a specific state is denoted as $g(x^{'})$ and this cost is formed recursively as the sum of the cost of its parent node $g(x)$ and a state transition cost $c(x, x^{'})$. That is, the cost of a node is defined as $g(x^{'}) = g(x) + c(x, x^{'})$. When a state is pulled from the queue, it is the minimum cost node that is pulled. This allows the search to progress in a a way that seeks to minimize the total cost from the initial state x_I to the goal state x_G or set X_G and this provides optimality guarantees for the search [30].

Also, in order to assess the issue of resolving duplicates of states throughout the search, Dijkstra's search stores states that have been encountered into two lists. A list $OPEN$ is used as the priority queue Q and stores states that have been encountered in the search process. Once a state has been selected as having the minimum cost in the $OPEN$ list, it is moved to a second list named $CLOSED$ [21]. If a duplicate is found for a state in the $OPEN$ list, its cost is updated if the duplicate has a lower cost associated with it. However, if a duplicate is found for a state in the $CLOSED$ list, the duplicate is neglected. This is due to the fact that once a state is removed from $OPEN$ and inserted into $CLOSED$, it has been declared as the lowest cost node of the states discovered so far. Due to the breadth first nature in terms of cost that Dijkstra's search exhibits, when a state is found as the minimum cost state and is moved to $CLOSED$, there is no other path to that node that could possibly result in a lower cost. Therefore, any duplicate of a state in $CLOSED$ that is found must be of a higher cost and is therefore neglected since the search seeks the minimum cost path to the goal [30]. Lastly, as a note on terminology, when a state is moved from $OPEN$ to $CLOSED$, this process will be referred to as expansion of the state. This terminology follows from the idea that when a state is pulled from $OPEN$, its neighbors are found and their costs are evaluated [21].

This technique is desirable due to its systematic nature, but a problem with the search algorithm exists in that since it expands states based upon a cost which is usually dependent in some way on the geometry of the search environment, the search tends to be quite breadth-first in nature. Because of this, a large number of states are typically expanded in the search process and this becomes a computational burden for either very large or high-dimensional state spaces. The A* algorithm attempts to alleviate this concern by incorporating an additional heuristic in the cost function to more effectively guide the search process [21]. The addition of this heuristic generally results in fewer state expansions than in Dijkstra's search while having the same optimality guarantee as long as certain conditions are placed on the heuristic $h(x)$.

The general A* algorithm is presented in Algorithm 2 and is formulated as a specific variant of the discrete forward search. Now the states are expanded according to a total cost function $f(x) = g(x) + h(x)$, where $g(x)$ is the same recursive cost accrued from the start state to the current state as in Dijkstra's search and $h(x)$ is the guiding heuristic [21]. In order to effectively guide the search process, this heuristic is an estimate of the least cost path to the goal. In an intuitive sense, this essentially has the effect of attracting the search in the direction of the goal and thereby causing fewer states to be expanded before the path to the goal is found. As discussed in [21], in order for the search to find the optimal path based on the cost function g and for the search to never expand duplicate states, the heuristic must be both *consistent* and *admissible*. Consistency implies that the heuristic at a particular state underestimates the cost of transitioning to a neighboring state plus its heuristic value. As [21] points out, this condition is equivalent to $h(x) \leq c(x, x^{'}) + h(x^{'})$. Admissibility implies that the heuristic does not overestimate the optimal cost to the goal at any state. However, since a heuristic being consistent implies that it is admissible, the only condition that needs placed on the heuristic is given below.

$$h(x) \leq c(x, x^{'}) + h(x^{'})$$

Algorithm 2 A* Search Algorithm, [41]

```
$g(x_I) \leftarrow 0$
$h(x_I) \leftarrow \textsc{dist}(x_I, X_G)$
$f(x_I) \leftarrow g(x_I) + h(x_I)$
$x_{I,parent} \leftarrow 0$
$OPEN \leftarrow x_I$
$CLOSED \leftarrow \{\}$
while $OPEN \neq \{\}$ do
    $x \leftarrow \arg\min_{x \in OPEN} f(x)$
    $OPEN \leftarrow OPEN \backslash \{x\}$
    $CLOSED \leftarrow CLOSED \cup \{x\}$
    if $x \in X_G$ then
        ReconstructPath()
        return SUCCESS
    end if
    for all $u \in U$ do
        $x' \leftarrow$ CalculateSuccessor$(x, u)$
        if $x' \in CLOSED$ then
            continue
        else
            if $x' \notin OPEN$ then
                $g(x') \leftarrow g(x) + c(x, x')$
                $h(x') \leftarrow \textsc{dist}(x, x')$
                $f(x') \leftarrow g(x') + h(x')$
                $x'_{parent} \leftarrow x$
                $OPEN \leftarrow OPEN \cup \{x'\}$
            else                                          $\{x' \in OPEN\}$
                $g\_temp(x') \leftarrow g(x) + c(x, x')$
                if $g\_temp(x') < g(x')$ then
                    $g(x') \leftarrow g\_temp(x')$
                    $f(x') \leftarrow g(x') + h(x')$
                    $x'_{parent} \leftarrow x$
                end if
            end if
        end if
    end for
end while
return FAILURE
```

Table 3.1: Comparison of A* and Dijkstra's search for a simple 2D map

	Closed Nodes	Open Nodes	Path Cost
A* Search	981	3923	1385.929
Dijkstra's Search	995697	3995	1385.929

If this is the case, then the A^* algorithm produces optimal paths based on the cost g but expands fewer states than Dijkstra's algorithm does. This makes the A^* search less computationally expensive than Dijkstra's search while producing the same results. Table 3.1 shows a simple comparison for these two searches where the state space is simply a 1000 by 1000 grid free of obstacles and where the start state is placed arbitrarily at the point (10,10) and the goal state is placed at the point (990,990). The cost function g is the distance traveled from the start state to the current state and the heuristic h is the straight line Euclidean distance from the current state to the goal state. As can be seen, the costs of the optimal paths are the same for both algorithms but the A* algorithm expands significantly fewer states. This results in faster computation time and less memory storage for the A* algorithm since fewer states need to be evaluated and stored in the $OPEN$ and $CLOSED$ lists throughout the search.

3.3 Weighted A*

Even with the large improvement that A* provides over Dijkstra's search in terms of computational efficiency, it is still relatively slow as far as search algorithms are concerned. This is especially the case as the state spaces for the searches get very large in size, increase in dimension, and become cluttered with obstacles that may make determining a path through an environment difficult. Since many interesting path planning problems are high dimensional, are desired to be implemented on a high resolution grid, and are employed in environments with significant clutter, the A* algorithm is not usually computationally efficient enough for these situations. As will be discussed in the following chapter of this thesis, the radio repeating path planning problem is formulated as a 4-dimensional search through an obstacle-rich environment. Therefore, it is necessary to speed up the A* algorithm in order to obtain rapid performance.

The computational burden of the A* algorithm primarily results from the fact that it produces optimal results. Many times, optimality is difficult to determine especially in highly nonlinear state spaces. The A* algorithm produces the optimal path from the start state to the goal and because of this, many states need to be expanded in the process to ensure that the path found is indeed optimal. However, often times in path planning scenarios, optimal paths are not always necessary and typically a feasible path is acceptable for many planning operations. This is usually predicated on three ideas. The first is that often times a feasible but suboptimal path is a workable solution for the planning operation. In many vehicle route planning scenarios, the planning environments are not adverse enough that only the optimal path is acceptable. Secondly, proving optimality is difficult and as [21] discusses, a suboptimal path can be found in a few state expansions whereas an optimal path requires a large number of additional expansions to prove optimality. This causes a large computational burden where the benefits of finding the optimal path rarely outweigh the computational costs required in determining optimality. Lastly, unless the planning problem is very highly nonlinear, as [21] discusses, suboptimal paths will have costs that are not substantially different than that of the optimal path and are therefore near optimal themselves.

These factors and other similar arguments have driven the quest for a faster suboptimal variant of the A* search. The Weighted A* (WA*) search is one such variant [21] and it operates by inflating the heuristic of the A* search so that it is inadmissible and therefore generates a stronger attractive force of the search towards the goal than the normal A* search would. The resulting total cost function for the WA* search is therefore

$$f(x) = g(x) + \epsilon h(x)$$

where $\epsilon > 1$ is an inflation factor for the heuristic. This makes the path planning operation much more depth first in nature especially at highly inflated heuristics which tends to decrease the number of states expanded substantially. More so, as [19] and [21] describe, the costs of the paths generated by the WA* search can be no greater than ϵ times the cost of the optimal path to the goal state. Therefore, even though the resulting paths are suboptimal, they are bounded in their suboptimality. Because of this, if the resulting costs need to be bounded for some reason, a trade-off can be performed between suboptimality and computational speed by adjusting the ϵ value accordingly.

For large or high dimension searches, this variant has the ability to speed up the search process substantially thereby making the search much more amenable to complex path planning problems. Simulation results for this variant are discussed in the following section as a way to validate this claim.

3.4 2D Simulation Results

The WA* algorithm discussed above makes use of an inflated heuristic in order to reduce the number of states that need to be expanded in order to determine a path from the start state to the goal state. As noted in the section above, these paths are suboptimal and have upper bounds on their path costs of ϵ times the optimal path cost. The performance of the WA* algorithm with respect to these upper bounds is based on the quality of the heuristic used throughout the search so testing has been performed for both informative and uninformative heuristics. Testing has shown though that even for a relatively uninformed heuristic, the paths that are determined have costs that are close to optimal even at high inflation factors ϵ. Therefore, this WA* algorithm provides rapid execution and produces near optimal paths.

In order to assess the performance of this algorithm in comparison to its A* counterpart, 50 random grid environments were constructed and paths were planned through these environments with different inflation factors ϵ. The results from the 50 simulations at each ϵ

value were then averaged to form the tabulated results. The environments were 300 by 400 grids with 30 randomly placed obstacles that were of a maximum size of 50 by 50 grid cells. These obstacles were intended to simulate buildings that would be typical of an urban environment that a typical unmanned vehicle mission might be performed in. Additionally, the start location was always at a fixed location of (10,10) and the goal was at a fixed location of (290,390). In the first batch of simulations, the cost function for the search was simply the distance traveled from the start state to the current state and the heuristic used was the straight line Euclidean distance from the current node to the goal. With this cost function and heuristic, the A* search is properly formulated and the heuristic is both consistent and admissible. Figure 3.5 shows an example of a state space used in the simulations and an optimal path planned through it. Note that the start position is denoted as a blue dot in the upper-left portion of the figure, the goal as a green dot in the lower-right portion of the figure, and impassable obstacles as red regions in the figure. Also, Figure 3.6 shows the states that were expanded with the states contained in the closed list in green and the states contained in the open list in blue. This is shown to provide some insight into the computational complexity of the A* search in an obstacle-rich environment.

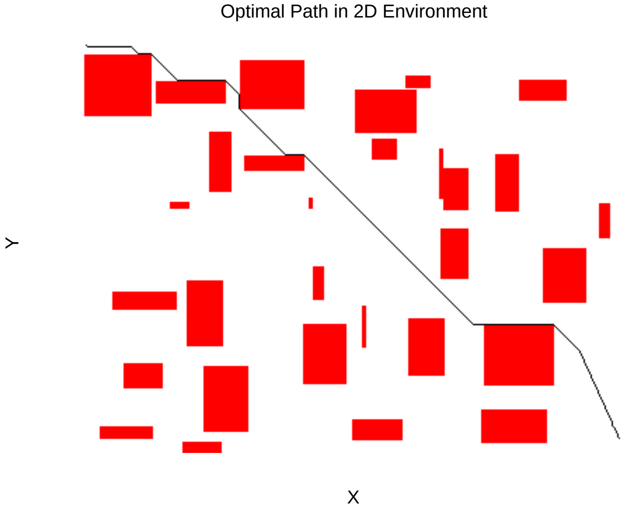


Figure 3.5: Example of an A* optimal path in a simulated 2D environment

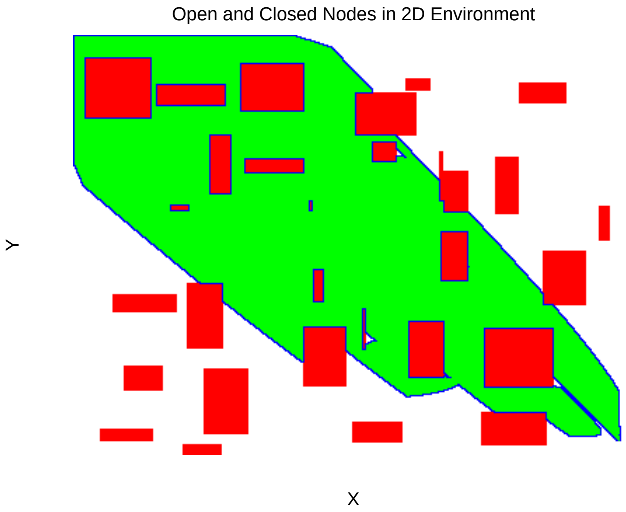


Figure 3.6: Example of the A* open and closed states searched in a simulated 2D environment

Additionally, Figures 3.7 and 3.8 show an example of the suboptimal path and the states in the open and closed lists for a WA* search with an inflation factor of $\epsilon = 1.2$. As can be seen in the figure, even at a very low inflation factor, the number of states that require expansion are much less than the optimal A* ($\epsilon = 1$) counterpart. This is to be expected though since in this set of simulations, the heuristic is considered to be very informative. With no additional uncertainty or costs associated with the path planning aside from strict distance traveled, the straight line Euclidean heuristic has a very well-informed idea of how to guide the search. Therefore, with an informed heuristic like this case, the computational efficiency of the WA* algorithm is clear.

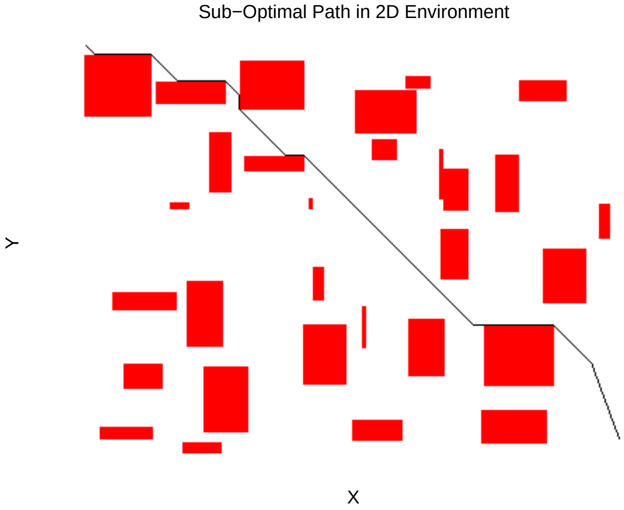


Figure 3.7: Example of an WA* suboptimal path ($\epsilon = 1.2$) in a simulated 2D environment

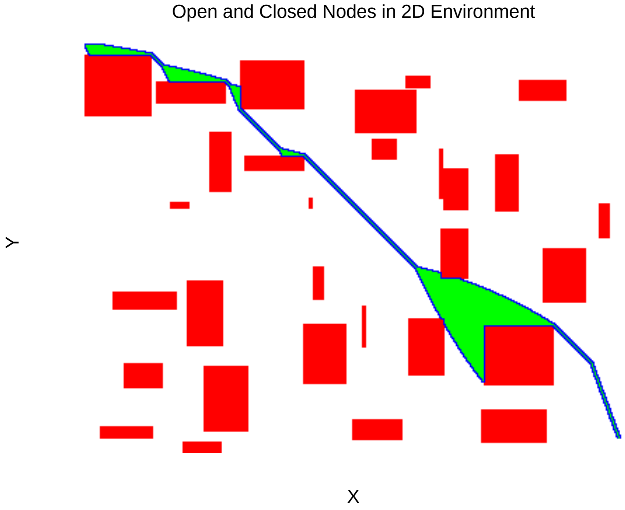


Figure 3.8: Example of the WA* open and closed states ($\epsilon = 1.2$) searched in a simulated 2D environment

Also shown is Table 3.2 where the results from the 50 simulations that have been performed at each ϵ value have been averaged. While 50 simulations is not overly extensive, the behavior of the searches does not change drastically for different environments of the same obstacle count so therefore 50 was deemed sufficient to understand the performance between searches

Table 3.2: 2D Weighted A* simulation data averaged over 50 trials - For loop implementation

Epsilon	Closed States	Open States	Path Cost	Computation Time (s)
100.0	454	1344	523.341	0.118
50.0	454	1343	523.395	0.118
20.0	457	1343	523.202	0.118
15.0	459	1343	523.150	0.119
10.0	464	1343	523.023	0.120
5.0	476	1345	521.110	0.123
4.0	479	1349	519.068	0.124
3.0	479	1347	518.682	0.123
2.0	501	1352	514.711	0.129
1.5	604	1359	510.589	0.156
1.4	675	1366	509.313	0.174
1.3	822	1372	507.537	0.220
1.2	1462	1395	506.484	0.505
1.1	4331	1484	506.442	2.793
1.0	32726	2346	504.872	86.301

of different inflation factors. Lastly, it is pertinent to note that in the implementation of the search algorithms, a naïve for loop technique was used to perform all of the insertion, deletion, and checking for the open and closed lists.

From Table 3.2, it is shown that even at very slight inflation factors, the number of closed states for the WA* searches are substantially less than the number of closed states for the normal A* search. Since the computation time increases as the sizes of the open and closed lists go up, it can be seen that the computation time decreases substantially when using the WA* search instead of the A* search. More so, it can be noted that even at very high inflation factors, $\epsilon = 100$ for example, the path cost is not substantially worse than that of the optimal path. In fact, even at the highly inflated $\epsilon = 100$ search, the path cost is only about 103% of the optimal path cost. This provides confidence for the claim that in implementation with a well-informed heuristic, even high inflation factors generate searches that are near optimal. Therefore, by using this WA* search, path planning can be hopefully implemented with major computational savings and with little effect on the path quality.

Table 3.3: Comparison of 2D Weighted A* simulation data with A* simulation data - For loop implementation

Epsilon	Fraction Optimal Closed States	Fraction Optimal Path Cost	Fraction Optimal Computation Time (s)
100.0	0.014	1.037	0.001
50.0	0.014	1.037	0.001
20.0	0.014	1.036	0.001
15.0	0.014	1.036	0.001
10.0	0.014	1.036	0.001
5.0	0.015	1.032	0.001
4.0	0.015	1.028	0.001
3.0	0.015	1.027	0.001
2.0	0.015	1.020	0.002
1.5	0.019	1.011	0.002
1.4	0.021	1.009	0.002
1.3	0.025	1.005	0.003
1.2	0.045	1.003	0.006
1.1	0.132	1.003	0.032
1.0	1.000	1.000	1.000

As an additional note, it should be noted that these simulations were performed on a Dell Latitude D630 laptop computer and although Windows might interfere slightly with the processing of the path planning, attempts were made to minimize the processor impact of other applications during the simulations. Therefore, although the computation time should be indicative of the performance of the path planning algorithm, better metrics to use for the computational expense of the planning process are the sizes of the open and closed lists.

To accentuate the effect of applying the inflation factor ϵ to the heuristic in the WA* search, Table 3.3 is provided which compares the number of closed states, path cost, and computation time for each of the WA* searches with those of the optimal A* search. These numbers were generated from the averaged results in Table 3.2.

Another very important point to note is that since the A* and WA* path planning algorithms are based on inserting, removing, and updating states in open and closed lists, the data structures used in the search process are very important to the speed at which the planning operations can be executed. In the naïve for loop implementation of the path planning above, states were stored sequentially in an array as they were added to the open and closed lists. This provided $O(1)$ insertion time since a new state could be simply appended to the end of one of the lists. On the other hand though, this implementation provided a worst case $O(n)$ time for searching to check whether an element is in the open or closed lists and a worst case $O(n)$ time for deleting a state if it is at the top of the list. Here n is the length of either of the lists. Since it is critical for the A* and WA* algorithms to check whether states are newly encountered, in the open list already, or in the closed list already, this $O(n)$ search time becomes very computationally expensive as the sizes of the open and closed lists get larger and larger.

A more clever implementation of these path planning algorithms makes use of a binary heap and a hash table for storing the open and closed lists. With these data structures, the path planning process can be implemented with an $O(\log_2(n))$ time for insertion and deletion, and an $O(1)$ time for searching whether a state is already in the open and closed lists. This computational benefit is quite substantial, especially when the sizes of the open and closed lists get large. Due to the computational benefit gained from using these data structures, the binary heap and hash table implementation will be used throughout the remainder of the thesis. Table 3.4 shows averaged data from 50 WA* simulations for each ϵ value which had the same parameters, costs, and environments as in the simulations that produced Table 3.2. The column on the right compares the computation time for the binary heap and hash table implementation with that of the for loop implementation. As is shown, there are substantial computational savings by implementing these data structures especially when the sizes of the open and closed lists get large. Therefore, these data structures, coupled with the WA* inflated heuristic makes the WA* path planning process much more attractive for real-time or near real-time implementation.

A last but very important detail with regards to the WA* algorithm is that while theoretical upper bounds are provided for the path costs of the resulting paths, the performance of the WA* algorithm with respect to these bounds is dependent on how well-informed the heuristic

Table 3.4: 2D Weighted A* simulation data averaged over 50 trials - Heap/Hash table implementation

Epsilon	Closed States	Open States	Path Cost	Computation Time (s)	Fraction For Loop Computation Time
100.0	454	1344	523.341	0.053	0.449
50.0	454	1343	523.395	0.047	0.398
20.0	457	1343	523.202	0.047	0.398
15.0	459	1343	523.150	0.047	0.395
10.0	464	1343	523.023	0.047	0.392
5.0	476	1345	521.110	0.048	0.390
4.0	479	1349	519.068	0.048	0.387
3.0	479	1347	518.682	0.048	0.390
2.0	501	1352	514.711	0.049	0.380
1.5	604	1359	510.589	0.053	0.340
1.4	675	1366	509.313	0.056	0.322
1.3	822	1372	507.537	0.062	0.281
1.2	1462	1395	506.484	0.090	0.178
1.1	4331	1484	506.442	0.214	0.077
1.0	32709	2354	504.872	1.476	0.017

used in the search is. As can be expected, if a very ill-informed heuristic is inflated by a factor ϵ, the resulting paths might be farther from optimal than the paths generated from the well-informed heuristic above. In order to address this, another batch of simulations was performed on the same set of 50 environments and with the same inflation values and conditions as in the previous simulations. In this batch though, a certain level of randomness was added to the environments by adding a random cost associated with each grid cell transition of 1-64 in value. This costs were generated prior to the simulations and therefore was common over all simulations at different ϵ values. By summing these costs with the distance traveled costs and by using the same straight line Euclidean distance heuristic, the simulations resulted in less ideal behavior. This is due to the fact that with the added random costs, the heuristic is not as informative as in prior simulations and therefore does not as effectively guide the search. Figures 3.9 and 3.10 display the optimal path and searched states in one of the environments while Figures 3.11 and 3.12 display the optimal path and

Table 3.5: 2D Weighted A* simulation data averaged over 50 trials - Random terrain cost

Epsilon	Closed States	Open States	Path Cost	Computation Time (s)
100.0	474	1416	12181.830	0.052
50.0	520	1418	10574.441	0.053
20.0	1703	1850	7738.606	0.107
15.0	9478	3096	6975.268	0.443
10.0	70459	4416	6653.174	2.779
5.0	95063	4543	6595.405	3.508
4.0	96483	4492	6592.443	3.519
3.0	97004	4428	6590.443	3.494
2.0	97140	4416	6590.128	3.470
1.5	97180	4411	6590.121	3.452
1.4	97186	4410	6590.121	3.447
1.3	97193	4410	6590.121	3.461
1.2	97199	4410	6590.121	3.446
1.1	97205	4410	6590.121	3.455
1.0	97211	4409	6590.121	3.438

searched states for the $\epsilon = 20$ variant in the same environment. This ϵ variant was chosen to accentuate the difference between the A* and WA* results since, as is discussed below, the $\epsilon = 1.2$ variant does not show considerable improvement from the A* ($\epsilon = 1$) case. Additionally, Table 3.5 displays the averaged results from the simulations and Table 3.6 shows comparisons between the optimal and suboptimal results.

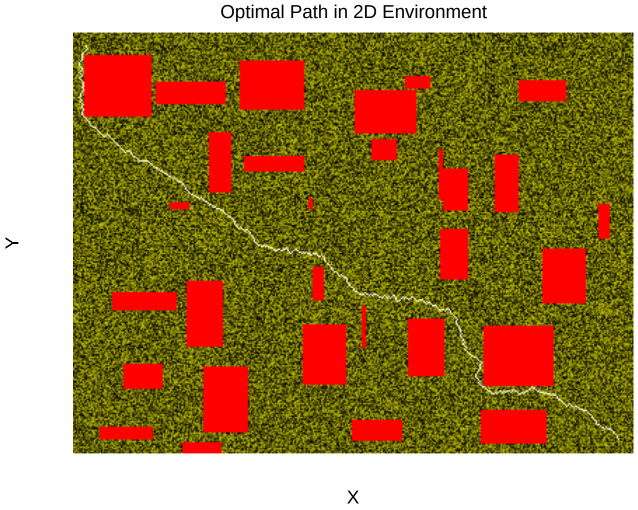


Figure 3.9: Example of an A* optimal path in a simulated 2D environment with random terrain costs

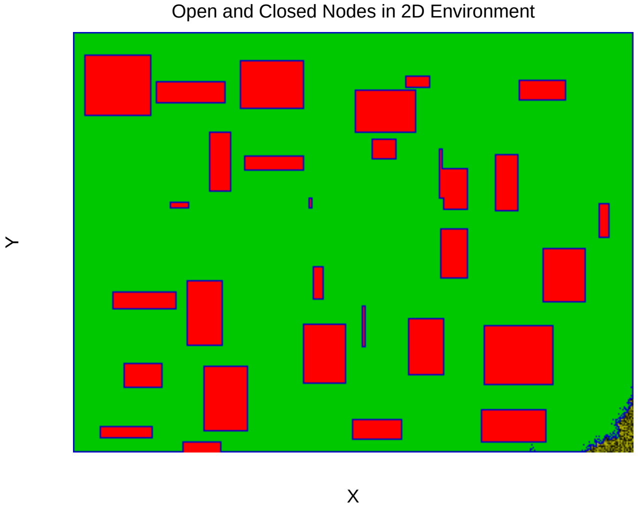


Figure 3.10: Example of the A* open and closed states searched in a simulated 2D environment with random terrain costs

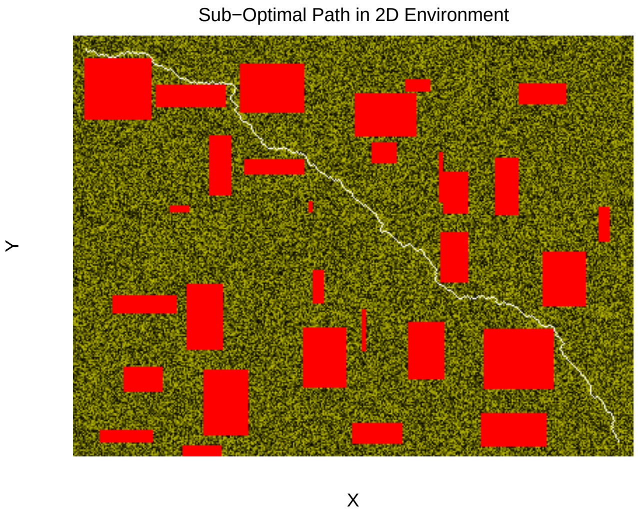


Figure 3.11: Example of an WA* suboptimal path ($\epsilon = 20$) in a simulated 2D environment with random terrain costs

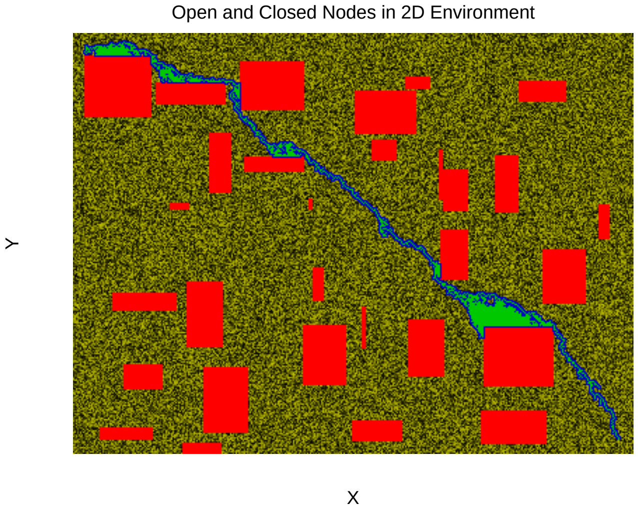


Figure 3.12: Example of the WA* open and closed states ($\epsilon = 20$) searched in a simulated 2D environment with random terrain costs

A number of interesting details can be noticed from these simulation results. The first is that while a well-informed heuristic generated very near optimal paths even at high inflation factors, this is not quite the case for the ill-informed heuristic. For example, at an inflation

Table 3.6: Comparison of 2D Weighted A* simulation data with A* simulation data - Random terrain cost

Epsilon	Fraction Optimal Closed States	Fraction Optimal Path Cost	Fraction Optimal Computation Time (s)
100.0	0.005	1.849	0.015
50.0	0.005	1.605	0.015
20.0	0.018	1.174	0.031
15.0	0.098	1.058	0.129
10.0	0.725	1.010	0.808
5.0	0.978	1.001	1.020
4.0	0.993	1.000	1.024
3.0	0.998	1.000	1.016
2.0	0.999	1.000	1.010
1.5	1.000	1.000	1.004
1.4	1.000	1.000	1.003
1.3	1.000	1.000	1.007
1.2	1.000	1.000	1.002
1.1	1.000	1.000	1.005
1.0	1.000	1.000	1.000

factor of $\epsilon = 100$, the average resultant path cost is approximately 1.8 times that of the optimal path. While this is less optimal than the paths from the well-informed heuristic, given that the theoretical upper bound on path cost is 100 times that of the optimal path, the WA* search still produces excellent paths even at high inflation factors. The other large detail to note is that while the well-informed heuristic caused far lower open and closed list sizes even for small inflation factors, this is not quite the case for the ill-informed heuristic. It is not until $\epsilon = 20$ or greater that the amount of closed states considered decreased substantially. This is likely an artifact of two phenomena. The first is that since the terrain costs (1-64) are generally higher than the distance costs in the search, it takes a higher inflation factor to allow the heuristic to become the dominant guiding force in the search than it would with out these terrain costs. Secondly, since the ill-informed heuristic does not guide the search as effectively as a well-informed one, more states are considered throughout the search inherently thereby causing it to not be as depth first as it could be.

The main point to be gathered from these results is that in order to experience the full benefit of the WA* search in terms of its computational efficiency and path quality, a well-informed heuristic should be used for the search process. A well-informed heuristic would address both of the phenomena that caused increased search times in the ill-informed heuristic simulations by having a more accurate concept of the terrain costs associated with the environment. Additionally, while this would be beneficial to lower dimensional searches, the true power of a well-informed heuristic would be when extending searches to higher dimensions. This provides motivation for section 3.6 which discusses the generation of a more accurate heuristic in the form of a heuristic look-up table.

3.5 3D Simulation Results

While the simulations and WA* formulations discussed thus far have been in a two dimensional search environment, the A* and WA* processes are generic and are not limited in terms of dimensionality. They can be applied to a state space of arbitrary size and dimension which makes them versatile for a variety of path planning scenarios as long as the state space, action space, cost function, and heuristic can be formulated properly.

Therefore, to provide some perspective of the extension of this algorithm to higher dimensions, simulations were performed in a three dimensional search environment and these results are given below. For the sake of memory requirements in the use of the hash table, the state space was chosen as being a grid of size 150 by 200 by 150. Also, given this smaller state space, obstacles were limited to a peak size of 25 by 25 by 25 grid cells and there were 30 obstacles placed within the state space. The start position was arbitrarily placed at the (10,10,10) coordinate and the goal location was arbitrarily placed at a fixed location of (145,195,145). Also, as with some of the previous two-dimensional cases, the cost function was formulated as the path length traveled from the start node to the current node and the heuristic was a straight line Euclidean distance from the current node to the goal. The results are given in 3.7 and 3.8 and were generated from averaging the data from running searches for each ϵ value on 50 environments with the characteristics described above. More so, Figures 3.13 and 3.14 show an example of the optimal path found and an example of the open and closed lists for the search. As with the previous results, the closed list states are represented by green grid cells and the open list states (which surround the closed list states) are represented by blue ones.

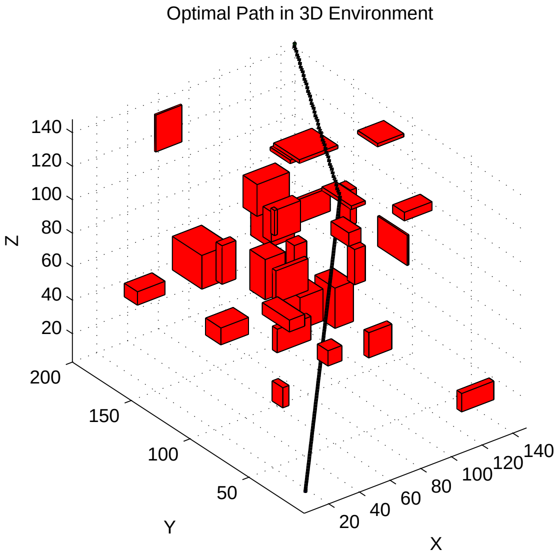


Figure 3.13: Example of an A* optimal path in a simulated 3D environment

Table 3.7: 3D Weighted A* simulation data averaged over 50 trials - Heap/Hash table implementation

Epsilon	Closed States 3D	Open States 3D	Path Cost	Computation Time (s)	Closed States 2D	Open States 2D
100.0	187	2857	284.871	0.472	232	664
50.0	187	2857	284.871	0.529	233	664
20.0	187	2857	284.871	0.525	234	663
15.0	187	2857	284.871	0.526	235	663
10.0	187	2857	284.873	0.524	236	662
5.0	187	2857	284.867	0.526	240	662
4.0	187	2858	284.877	0.524	242	663
3.0	187	2859	284.879	0.525	249	665
2.0	189	2863	284.918	0.524	275	671
1.5	190	2868	284.728	0.524	322	673
1.4	194	2875	284.747	0.524	357	680
1.3	195	2878	284.558	0.524	388	682
1.2	211	2905	284.538	0.525	591	698
1.1	4553	7743	284.498	0.847	1663	750
1.0	192094	53285	284.391	12.806	8292	1160

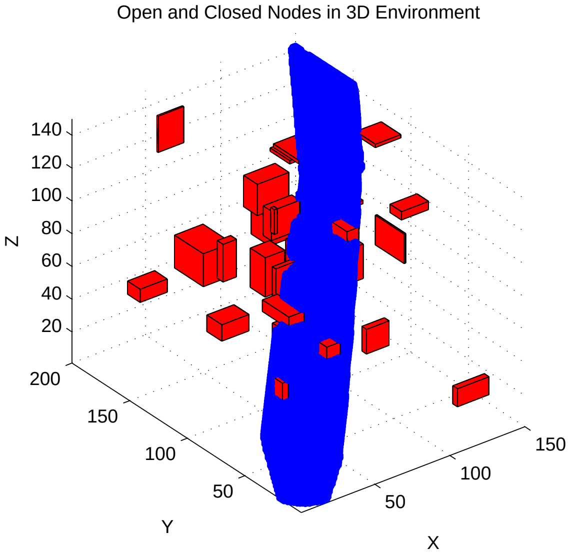


Figure 3.14: Example of the A* open and closed nodes searched in a simulated 3D environment

Table 3.8: Comparison of 3D Weighted A* simulation data with A* simulation data - Heap/Hash Table Implementation

Epsilon	Fraction Optimal Closed States	Fraction Optimal Path Cost	Fraction Optimal Computation Time (s)
100.0	0.001	1.002	0.037
50.0	0.001	1.002	0.041
20.0	0.001	1.002	0.041
15.0	0.001	1.002	0.041
10.0	0.001	1.002	0.041
5.0	0.001	1.002	0.041
4.0	0.001	1.002	0.041
3.0	0.001	1.002	0.041
2.0	0.001	1.002	0.041
1.5	0.001	1.001	0.041
1.4	0.001	1.001	0.041
1.3	0.001	1.001	0.041
1.2	0.001	1.001	0.041
1.1	0.024	1.000	0.066
1.0	1.000	1.000	1.000

Looking at the open and closed lists both visually as in Figure 3.14 and numerically in Table 3.7, there tends to be a larger amount of states investigated throughout the search process than in a similarly sized 2D state space as shown also in Table 3.7. This shows the “curse of dimensionality” of path planning operations and further shows the necessity of speeding up the search processes in high dimensional searches even with a well-informed heuristic.

3.6 Heuristic Look-Up Table

While the WA* algorithm does an excellent job at speeding up the path planning process, a complication arises if a heuristic is formulated in a search that does not effectively guide the search to the goal state. In the example immediately above, the state space was three dimensional and the heuristic was also three dimensional in that it guided the search to a single goal state in the three dimensional space. However, in some searches, especially in the radio repeating problem, a heuristic that guides the search to a specific state in the state space is not always available. For example, a situation can be imagined where the goal for a search is actually an entire region or subset of the state space.

Using the radio repeating problem as an example, there are two vehicles that require planning cooperatively. There is the ground vehicle and the helicopter. In this search, the ground vehicle has a well defined goal state but the helicopter is just meant to be placed in a following mode where it moves to maintain the health of the radio link. Therefore, there is no one distinct goal location for the helicopter that can be predicted a priori since its movements are very much based on those of the ground vehicle throughout the search. Because of this, in the state space, there is a region of goal states where the ground vehicle is at a fixed goal location but the helicopter can be anywhere that it ends up in the state space. Therefore, a guiding heuristic can only really be used for the ground vehicle. Although the search is coupled, this provides sort of a well-guided search for the ground vehicle but an unguided one for the helicopter. This results in an unavoidable large amount of state expansions as the helicopter searches through its possible movements.

In order to show this problem visually, a state space has been formulated where a three dimensional search is performed. In this search, the scenario assumes a freely moving aerial vehicle at the start location (blue dot) and it is desired to get to a goal (x,y) location (green line). For the purposes of this example scenario, the aerial vehicle can stop at any altitude (z location) in the space but throughout the search it must maintain a line-of-sight condition with the start point. This is done to mimic the idea of a strong radio signal needing to be maintained with a ground station at the start point.

In this scenario, the helicopter is guided by a heuristic that attracts it to the goal (x,y) position but there is no simple way to formulate a heuristic that is known a priori to guide it to a z location. Therefore, the aerial vehicle is relatively unguided in the z direction except for the line-of-sight constraint. As shown in Figures 3.15 and 3.16, this can result in a large amount of states expanded in the search which can slow down the search process greatly.

For each (x,y) pair that the aerial vehicle investigates, it must investigate a number of z locations before continuing the search.

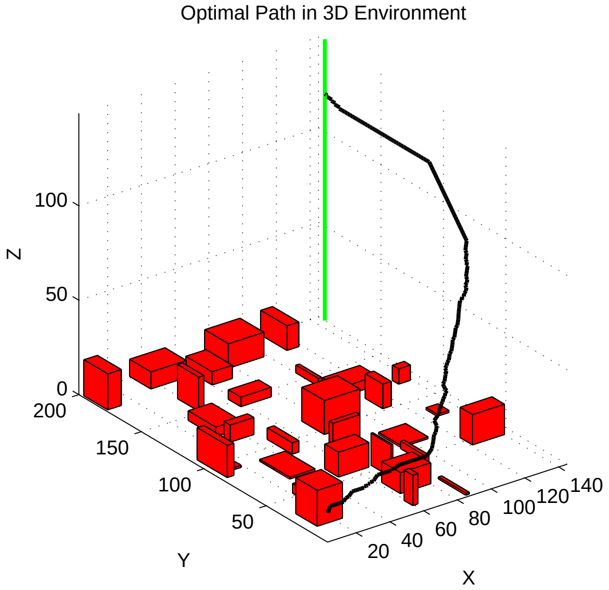


Figure 3.15: Optimal path for poorly guiding heuristic search

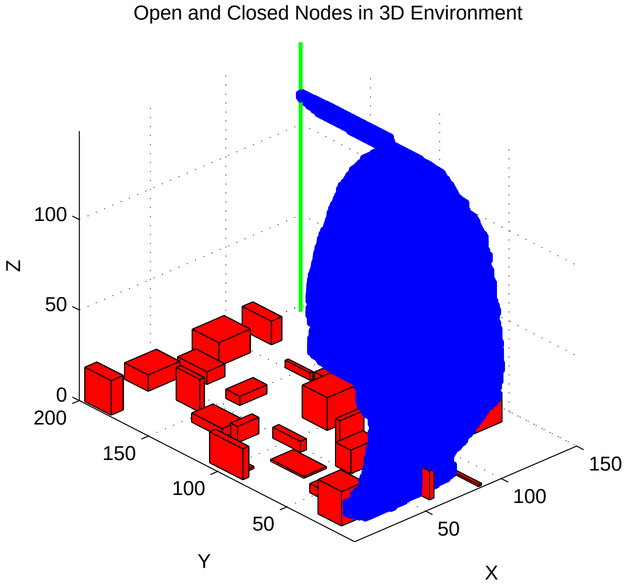


Figure 3.16: Open and closed states for poorly guiding heuristic search

While this problem cannot always be eliminated in every path planning formulation, the effects can be minimized by using a more accurate heuristic to guide the (x,y) movements of the search. This way, less (x,y) locations are searched and, consequently, less z locations are searched. This helps to speed the search up in scenarios where a well-guiding heuristic is not available for all dimensions of the search.

As discussed by [18] and [19], a better heuristic can be formulated that caters to the specific environment of the search by performing an initial lower dimensional search through the environment to act a pre-filter for the full path planning process. For the the radio repeating example, this equates to first performing a backwards unguided search from the goal to the start for the two dimensions that the ground vehicle travels in. The distances traveled to each state from the goal in this two-dimensional search are then stored in a look-up table which serves as the heuristic for the higher dimensional search. This heuristic look-up table much more effectively navigates the ground vehicle to its goal location than a standard straight line Euclidean distance heuristic would. This heuristic look-up table is effective for the case described here and for higher dimensional searches with random terrain costs as discussed in section 3.4 for example.

Figures 3.17 and 3.18 visually show example heuristic values for the straight line Euclidean distance heuristic and the heuristic look-up table, respectively. Here an environment similar to Kentland Farms is used except that additional barriers have been added to accentuate the effect of the look-up table. Higher heuristic values are shown in red while lower values are shown in blue. As shown in these figures, the straight line Euclidean distance heuristic just expands radially outward from the goal location which is not effective at guiding the search in a situation where many obstacles or dead ends exist. On the other hand, the heuristic look-up table provides a heuristic that more effectively guides the search around the buildings and towards the goal.

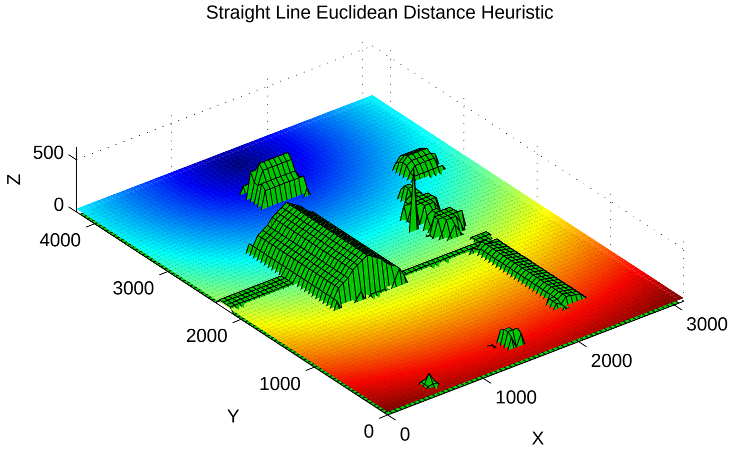


Figure 3.17: Plot of Euclidean heuristic in Kentland Farms style environment

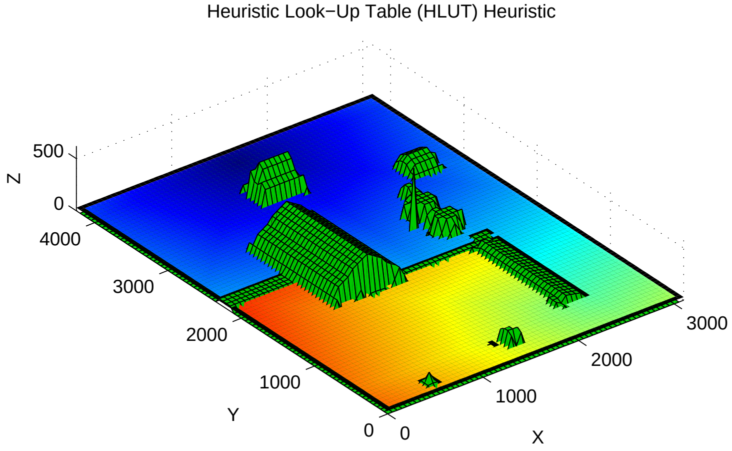


Figure 3.18: Plot of heuristic look-up table in Kentland Farms style environment

By incorporating the various details from this chapter into a path planning search, many path planning problems can be formulated and effectively solved even in spite of large state spaces, high dimensional state spaces, and poorly guiding heuristics. Most notably, by using the Weighted A* algorithm, by taking advantage of efficient data structures like binary heaps and hash tables, and by using a heuristic look-up table, path planning operations can be performed quickly even for complicated scenarios. Even more so, this path planning process is orderly, systematic, and gives suboptimality guarantees for the resultant path. Because of these advantages, the remainder of the thesis uses the details of this chapter as the foundation for the path planning efforts.

Chapter 4

Path Planning for Radio Repeating

As noted in the A* section above, the A* algorithm is a general grid-based search technique that can be applied to a variety of path planning scenarios. In this chapter, the general algorithm for A*, and more specifically WA*, is catered to the specific application of radio repeating. In this scenario, a multiple vehicle team is used to allow for a ground robot to navigate through an obstacle-filled environment while ensuring that strong and healthy radio repeating links are maintained between the vehicles in the system and the ground station. In order to adapt the path planning to these objectives, not only do the costs associated with the trajectory of the ground vehicle need to be taken into account but the performance of the radio links between must be assessed as well. Therefore, this chapter is split into two primary sections. The first discusses the quantification and modeling of the radio links between the radios used in the mission. This section discusses the principle of radio link budgeting and how various factors in the link budgeting equation are affected by the geometry of the radios and their locations in a search environment. Additionally, it presents experimental data taken from testing the radios and discusses Fresnel ellipsoids which are commonly considered in the radio community for radio network planning [40]. With this modeling in place, the second section discusses the formulation of the path planning problem and how the cost function and heuristic for the search were generated. Simulated testing of the radio repeating path planning algorithm is also documented and the path planner's performance is discussed. Finally, since a natural extension to this work would be to incorporate this concept with multiple aerial vehicles, a brief discussion is given about the applicability of the path planner to situations involving two and three aerial vehicles in the radio repeating operation.

4.1 Radio Modeling

In order to create a path planner that seeks to provide healthy radio repeating links between the vehicles used in the operation, the links between the radios in the system needed to be quantified. In order to accomplish this task, the received signal strength of a radio link was used as the primary metric by which the overall health of that link was determined. The rationale for this, as discussed in the literature search, is that [20] noted that the received signal strength is highly correlated with the bit error rate of a communication link. Since a higher bit error rate is indicative of a weaker communication link, this decision to use received signal strength as a metric is well founded. Therefore, this section begins by discussing the radio link budget equation which relates the received signal strength of a radio link to the various gains and losses incurred in the transmission and receiving processes. When considering the link budget, the two primary factors that will likely vary throughout the search process are the antenna gains and the path loss values. Because of this, a distinct section is also devoted to the modeling of each one of these link budget terms. Lastly, as can be seen in section 4.1.3, there is a very large degradation in performance of the radio link when line-of-sight behavior is not observed. Therefore, the last section in the radio modeling portion of this chapter discusses the concept of Fresnel Ellipsoids which are used in the cost function to tend the path planning towards more line-of-sight conditions.

4.1.1 Link Budgeting

In order to quantify the radio links between the vehicles and the ground station in the radio repeating scenario, a link budgeting technique is used. Link budgeting is a widely used and systematic method of predicting the received signal strength of one radio based on a given transmitted power from the other radio in the pair. This link budget takes into account many considerations that cause gains and losses in the radio signal. These include effects like directionality of the antennas used in the system, losses produced by the cables and connectors in the system, propagation effects, and any other gains or losses that influence the received signal strength of the receiving radio. This section gives an overview of the link budgeting equation and some of the factors that play a role in the calculation of received radio signal strength.

In a simple radio link, there are two radios that are present. The first radio is the transmitting radio and the other is the receiving radio. In a real radio link, both radios transmit and receive but for the sake of exposition, one will be considered the transmitter and the other the receiver. Starting off by considering the transmitting radio, a certain input power P_t is delivered to its antenna which causes electromagnetic radiation to be emitted spherically from the antenna. Using the derivation of the Friis Transmission Equation from [5], the isotropic power spectral density of the transmitted power is simply

$$\rho = e_t \frac{P_t}{4\pi R^2}$$

Here the transmitted power P_t is spread over the area of a sphere of radius R as it radiates outward from the antenna. Additionally, an efficiency factor e_t is included to account for inefficiencies or losses in this transmission process. This isotropic power spectral density represents the power per unit area that would be transmitted if the antenna could radiate equally in all directions. Real antennas however are not isotropic and possess a certain directionality to them. This is known as directivity and is the property that describes how the radiated power is concentrated or focused in a particular direction [24]. Mathematically, it is defined as the radiation intensity in a single direction over the average radiation intensity of the antenna averaged over all directions. That is, it is the scaling factor of the radiation intensity in a particular direction as compared to an isotropic counterpart. With this, following [5], we can write the power spectral density in a particular direction as

$$\rho = e_t \frac{P_t D_t(\theta_t, \phi_t)}{4\pi R^2}$$

Here, $D_t(\theta_t, \phi_t)$ is the directivity of the antenna. Once the power is transmitted from the transmitting radio and propagated through an environment, the receiving radio must collect some of the electromagnetic radiation to complete the radio link. In order to quantify this, an effective area A_r is defined for the receiving antenna. Therefore the power received by the receiving radio is the power spectral density multiplied by this effective area [5].

$$P_r = A_r \rho$$

By using the Lorentz Reciprocity Theorem [24], an argument can be made that the radiation pattern of the antenna, which is the plot of the directivity in all directions, is the same when the antenna is used in both transmitting and receiving modes. As noted by [5], this result is general provided that linear materials are used in the antenna electronics and that the propagation medium is linear as well. For the radios used in this work and using air as a propagation medium, this is a valid approximation. This allows the effective area A_r to be related to the directivity of the receiving antenna by the following equation [5].

$$A_r = e_r D_r(\theta_r, \phi_r) \left(\frac{\lambda^2}{4\pi} \right)$$

Here an efficiency factor e_r was included to account for any inefficiencies in the receiving antenna and λ is the wavelength associated with the frequency of transmission. With this relation, the received power can be written as follows [5].

$$P_r = e_t e_r P_t D_t(\theta_t, \phi_t) D_r(\theta_r, \phi_r) \left(\frac{\lambda}{4\pi R}\right)^2$$

At this point, the efficiencies e_t and e_r have just been left as general factors to account for inefficiencies in the system. However, these are commonly separated out into a number of terms that account for specific inefficiencies in the system. Shown below, the efficiencies are typically split into the following factors.

$$e_t e_r = e_{cdt} e_{cdr} (1 - |\Gamma_t|^2)(1 - |\Gamma_r|^2) |\hat{\rho}_{\mathbf{t}} \cdot \hat{\rho}_{\mathbf{r}}|^2$$

Here, e_{cdt} and e_{cdr} account for conduction and dielectric losses in the system, $(1 - |\Gamma_t|^2)$ and $(1 - |\Gamma_r|^2)$ account for reflection losses or, equivalently, mismatches in impedance in the antenna circuitry, and $|\hat{\rho}_{\mathbf{t}} \cdot \hat{\rho}_{\mathbf{r}}|^2$ accounts for polarization mismatches between the transmitting and receiving antennas. These polarization mismatches occur when the electric fields associated with the transmitting and receiving antennas are not aligned. By parsing the efficiencies out in this way, the general form of the Friis Transmission Equation [5] is as follows.

$$\frac{P_r}{P_t} = e_{cdt} e_{cdr} (1 - |\Gamma_t|^2)(1 - |\Gamma_r|^2) \left(\frac{\lambda}{4\pi R}\right)^2 D_t(\theta_t, \phi_t) D_r(\theta_r, \phi_r) |\hat{\rho}_{\mathbf{t}} \cdot \hat{\rho}_{\mathbf{r}}|^2$$

For the case of the radios used in this work, impedance mismatches can be neglected ($\Gamma_t = \Gamma_r = 0$). Also, both the transmitting antenna and the receiving antenna are the same model with the same polarization. Since this is the case and since they have been mounted in every test and experiment to have their polarizations matched, there are no losses from polarization mismatch ($\hat{\rho}_{\mathbf{t}} \cdot \hat{\rho}_{\mathbf{r}} = 1$). Therefore, the Friis Transmission Equation can be written in the more simplified form below.

$$\frac{P_r}{P_t} = e_{cdt} e_{cdr} \left(\frac{\lambda}{4\pi R}\right)^2 D_t(\theta_t, \phi_t) D_r(\theta_r, \phi_r)$$

Lastly, combining the conduction and dielectric inefficiencies with the directivities of the antennas, the gains G_t and G_r can be formed which are the signal amplifications that the antennas produce in reality. Therefore, the final simplified form of the Friis Transmission Equation can be written as

$$\frac{P_r}{P_t} = \left(\frac{\lambda}{4\pi R}\right)^2 G_t(\theta_t, \phi_t) G_r(\theta_r, \phi_r)$$

With this equation, the received signal strength can be calculated based on the transmitted power, the gains of the antennas, and the loss in free space which is represented by the $\left(\frac{\lambda}{4\pi R}\right)^2$ term. However, since the gain and loss factors can be quite large in radio communications, it is common practice to convert the Friis Transmission Equation to a decibel equivalent and to use the resulting equation for link budgeting. Here, since power quantities are being dealt with, the the conversion to decibels uses dB$= 10\log_{10}(G)$ where G is a gain. This is in contrast to the decibel conversion that uses a factor of 20 which is common when dealing with positions, velocities, and field strengths for example. By making this conversion and using the property of logarithms that $\log_{10}(ab) = \log_{10} a + \log_{10} b$, the link budgeting equation becomes

$$P_r = P_t + G_t - L_{Path} + G_r$$

All terms in the above equation are represented in terms of some form of decibels. For example, the transmitted power P_t and the received power P_r are represented in terms of dBm which is a decibel scale that has a reference based on the milliwatt. In this scale, 1 mW= 0 dBm and 1 W= 30 dBm for example. The rest of the quantities are given as relative gains which are specified in dB for the path losses and dBi for the antenna gains. Here, dBi is simply a decibel scale with a reference to a theoretical isotropic radiator but since an isotropic radiator radiates equally in all directions and produces a unity gain, a gain in dBi can simply be added directly to a gain in dB. Therefore the units are all consistent.

It should also be noted that here the path loss L_{Path} is the decibel equivalent of the $\left(\frac{\lambda}{4\pi R}\right)^2$ term from the Friis Transmission Equation. This is known as the Free Space Path Loss and it is the quantification of the losses from propagation in free space which is given on a decibel scale as [40]

$$L_{Path} = 20\log_{10}(f) + 20\log_{10}(R) - 147.55$$

Here, the conversion $\lambda f = c$, where c is the speed of light, has been used to obtain an equation in terms of frequency f since this is a more commonly available value in radio communications. Also, f is in units of Hz, R is in units of meters, and the sign of the equation has been modified to be consistent with the above link budgeting equation.

Finally, although the link budgeting equation provides a way to tabulate the gains and losses associated with the radio communication, in a real system other effects need to be considered. For example, both cabling and connectors have losses associated with them and can be factored into the link budget. More so, free space conditions are not always valid so the L_{Path} equation above may not be an accurate predictor of the losses throughout the space between the transmitting and receiving antennas. As will be shown, empirical models for path losses typically make modifications to the L_{Path} equation in an attempt to more accurately quantify the effects of electromagnetic propagation in a variety of environments. For this reason, this value is kept general in the link budgeting equation for now. Including the cable and connector losses, the final link budgeting equation that will be used throughout the rest of this work is given below. Figure 4.1 is also included to summarize the various gains and losses discussed throughout this section. Note that the blue lobe shapes on the antennas are intended to depict common omnidirectional antenna radiation patterns.

$$P_r = P_t - L_{Conn_t} - L_{Cable_t} + G_t - L_{Path} + G_r - L_{Conn_r} - L_{Cable_r}$$

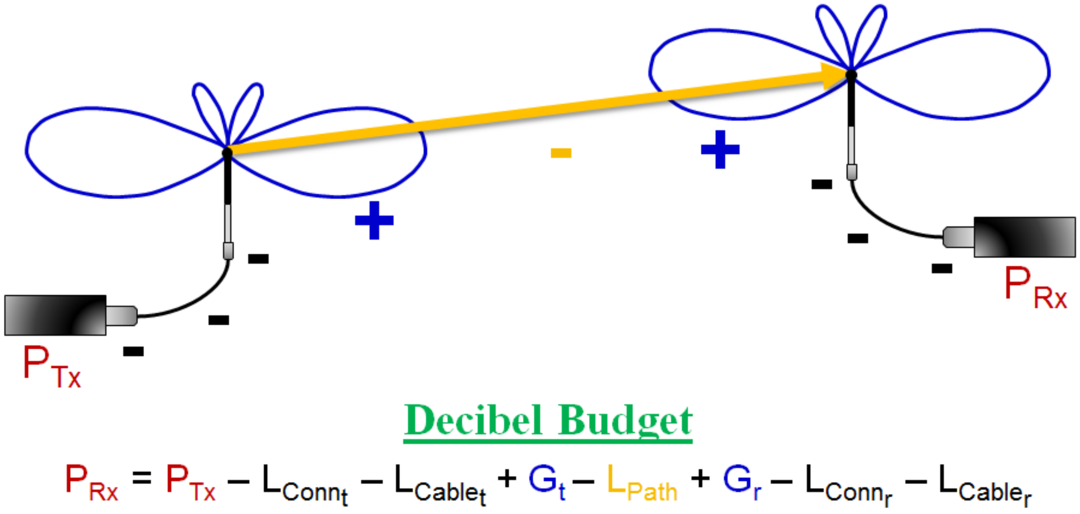

+
-
+
-
-
-
-
-
-
P_{Rx}
P_{Tx}
Decibel Budget
$P_{Rx} = P_{Tx} - L_{Conn_t} - L_{Cable_t} + G_t - L_{Path} + G_r - L_{Conn_r} - L_{Cable_r}$

Figure 4.1: Link budget diagram for a typical radio link

4.1.2 Antenna Lobe Testing

With the overarching link budgeting equation determined, the remaining steps in effectively modeling the radio performance involved characterizing how each term in the budget affects the received signal strength prediction. Looking at the link budget equation, only two major losses require experimental characterization. These are the antenna gains G_t and G_r, as well as the path loss value L_{Path}. The rest of the terms are either known quantities or estimates that are published by manufacturers. For example, the transmitted power, P_t is a known value and is typically input to the radio as a configuration setting. Also, the connector losses, L_{Conn_t} and L_{Conn_r}, and the cable losses, L_{Cable_t} and L_{Cable_r} are usually provided on data sheets for the connectors and cables respectively. Therefore, in order to characterize the rest of the terms in the link budget, this section will cover the experimental setup and results for determining the antenna gains and the next section will describe a similar experiment for determining the path loss values.

The antenna gains are an important parameter for characterizing the health of a radio link and knowing their geometric dependencies is critical for path planning operations. As described in the Link Budgeting section above, real antennas are not perfectly isotropic and do not radiate electromagnetic energy equally in all directions. Real antennas also have efficiencies that may cause extra losses in the transmitting and receiving processes. These two factors when combined together give the gain of the antenna, $G(\theta, \phi)$, which includes a directional dependence on both the azimuth angle θ and the elevation angle ϕ between the transmitting and receiving antennas. By taking measurements of the antenna gain values at each (θ,ϕ) pair, the antenna radiation pattern can be determined, which is commonly represented as a three dimensional surface plot in spherical coordinates. In this plot, the distance of any point on the surface from the origin corresponds to a gain for the antenna in the corresponding direction of that point. Often the scale is offset where the origin does not strictly represent 0 dB but regardless, these plots very clearly show the directional dependence of a particular antenna. Figure 4.2 below shows a typical radiation pattern plot for an omnidirectional antenna at the left hand side of the figure.

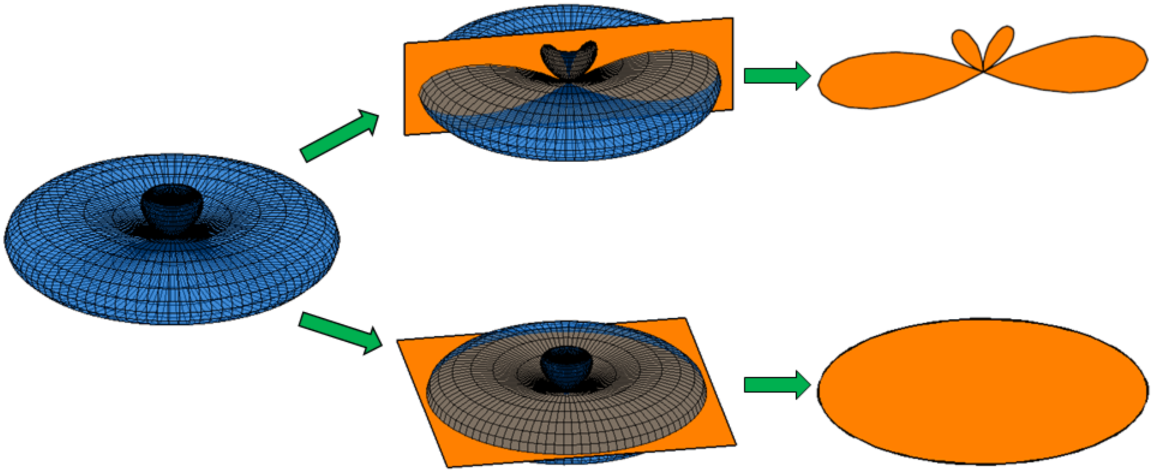


Figure 4.2: Typical omnidirectional antenna radiation pattern and relevant cross sections

As might be expected by the three dimensional nature of this radiation pattern plot, it would be a tiresome procedure to accurately measure the gain of an antenna at all possible azimuth and elevation angles. Fortunately, many antennas, especially omnidirectional ones, have radiation patterns that are symmetric to some degree. Therefore, in lieu of taking exhaustive measurements at every θ and ϕ value for the antenna, only a single cross section in the vertical and horizontal directions are usually measured and published. Figure 4.2 shows these cross sections with the vertical cross section depicted at the top of the figure and the horizontal cross section depicted at the bottom of the figure. If an antenna pattern was not symmetric, information would be lost in this process but when dealing with omnidirectional antennas where there is little to no azimuth angle dependency in the radiation pattern, measuring the gains for these cross sections is sufficient for reconstructing the radiation pattern plot for the antenna. Therefore, since omnidirectional antennas were used in this work, this cross section method is the method that was used in the antenna radiation pattern experiment described below.

Since determining the gains associated with the antennas in the radio system is critical to characterizing the health of a radio link, an experiment was performed to determine the directional dependence of the gains. In this experiment, two Cobham COFDM radios were set up at a known distance apart and radio signal strength measurements were successively taken as one of the antennas was rotated in its elevation and azimuth directions. By using estimates of the losses associated with the cabling and connectors, by setting the radios up in a location where free space path loss conditions apply, and by assuming that the antennas were identical due to the fact that they were the same model and make, taking received signal strength measurements at different elevation and azimuth angles were enough to determine the antenna gains. Figure 4.3 and Figure 4.4 show the test setup for the antenna radiation pattern experiment. Figure 4.3 shows the transmitting and receiving antennas mounted on tripods and positioned along a road that is devoid of obstacles. More so, the road that was used in the test setup was chosen not only because there was little traffic and no obstacles present, but also because the road was hilly enough that both antennas could be placed at the top of hills with a valley in between them. As will be discussed in section 4.1.4, a setup like this allows the first Fresnel zone for the radio wave propagation to be clear and this allows free space path loss to be a valid approximation [40]. Figure 4.4 shows the transmitting antenna at an elevation angle of 90 degrees in order to show how the tripod was used to change the angle of the antenna as measurements were taken. The tripods that were used had a range of motion from -50 to 90 degrees so these are the angles that were measured for the antenna gains. As it turns out, the region from -50 to -90 degrees is a null zone for the antennas so it was not critical that these angles were measured.

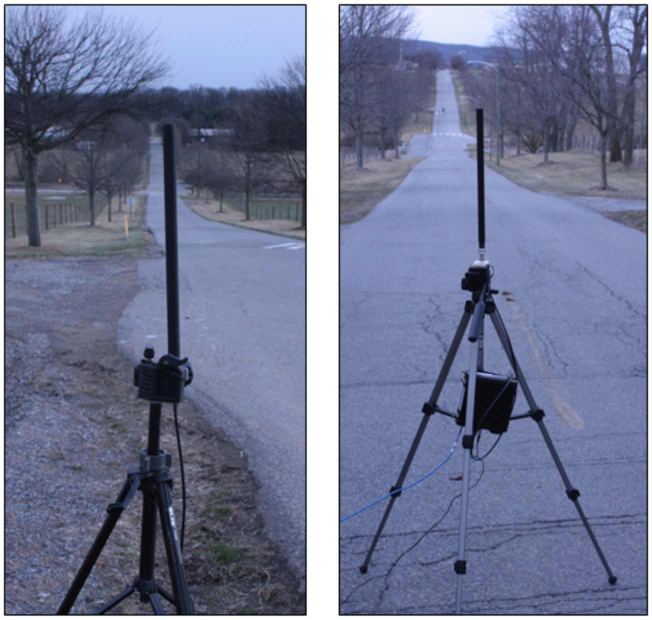


Figure 4.3: Setup of antennas for the radiation pattern experiment - 434 meters between antennas

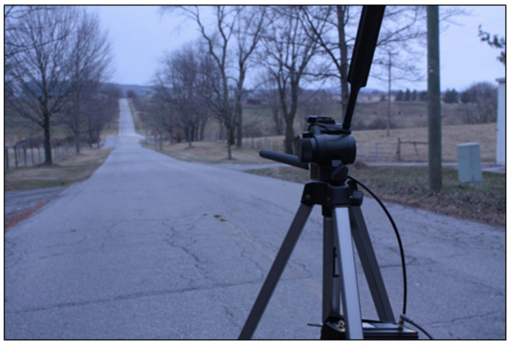


Figure 4.4: Transmitting antenna at a 90 degree elevation angle

Although the experiment, including its setup and results, could be described in a large amount of detail, a brief list of the experimental procedure and the results are given below to explain the essential details from the experiment.

1. The two antennas were set up on camera tripods that had a range of motion from −50 to 90 degrees. These tripods contained a useful locking mechanism so that the antennas could be held stationary at any desired angle within this range.

2. The transmit power for the Cobham COFDM transmitting radio was set to 1 W, or equivalently 30 dBm. This power value is considered to be a known and constant value for the radio communication.

3. The two Cobham COFDM radios were connected to their respective antennas via a 4 foot cable. This cable was an L-com CA-195R cable that had an SMA connector (L-com ASM-1714) at the radio end and a TNC connector (L-com ATF-3700) at the antenna end. Product information for the cables and connectors that give estimates for their decibel losses can be found in [27], [28], [29]. For the given length of cable used (8 feet total), a cable loss of 1.486 dB could be assumed and for all of the connectors, a total loss of 0.80 dB could be assumed.

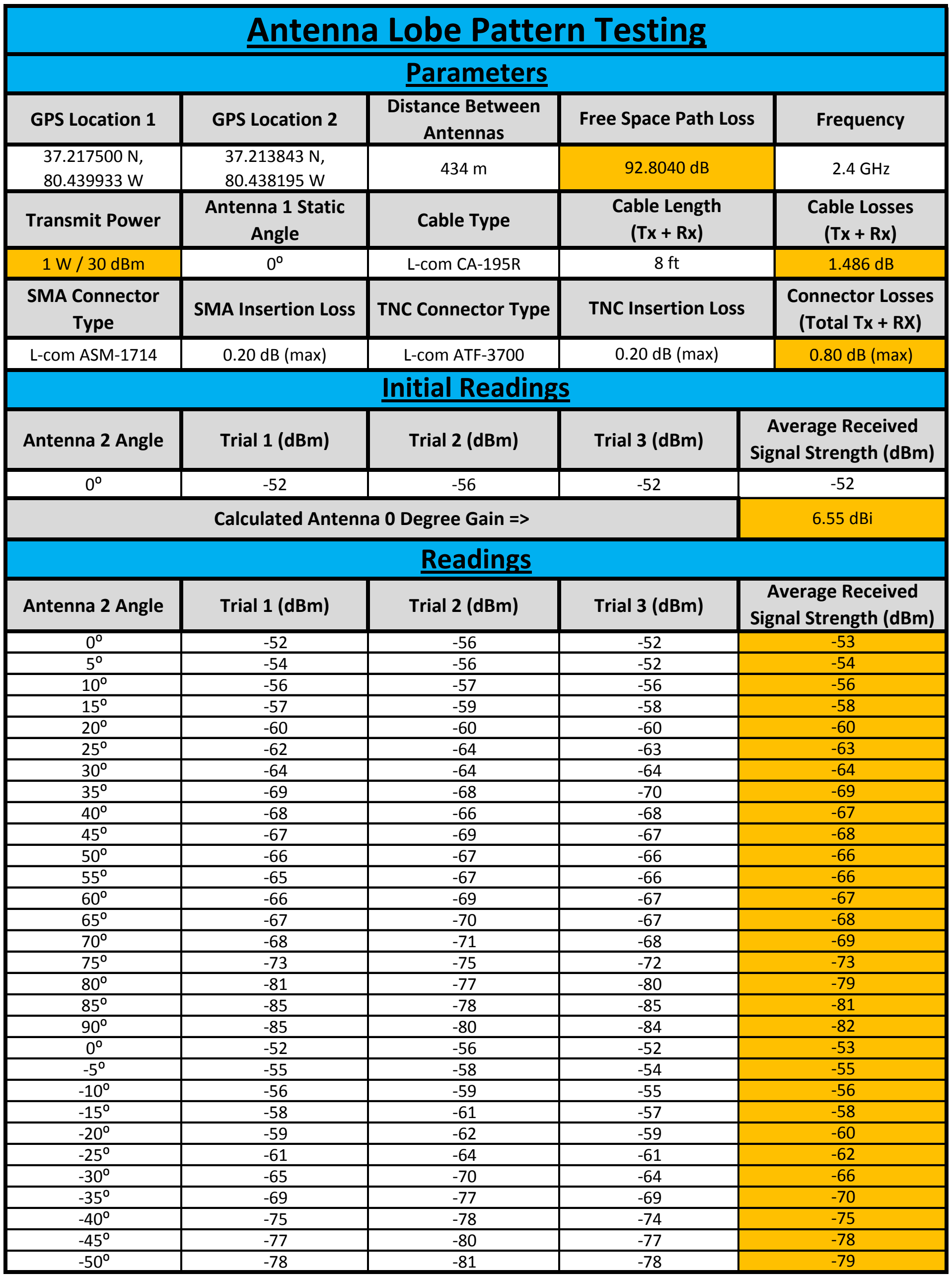
4. The tripods with the transmitting and receiving radios and antennas were powered on and were placed at a known distance of 434 meters apart. More so, the antennas were placed along a road that was free of obstacles and had a hilly profile so that the antennas could be placed at the same height with a valley between them. This ensured that the first Fresnel zone was clear and that free space path loss conditions could be assumed. Under these assumptions, the path loss was approximated to be 92.8 dB.

5. At this point, when a received signal strength measurement was taken, the only two variables left were the gains of the transmitting and receiving antennas. In order to eliminate the receiving antenna gain as a variable, an intial reading of the received signal strength was taken where both antennas were set at a 0 degree elevation angle.

By assuming that both antennas have the same radiation patterns since they are of the same make and model and by noting that the gains are for the antennas are no different in transmitting or receiving mode based on the reciprocity argument from [24], the antenna gains for the transmitting and receiving antennas were assumed to be the same for this reading. This allowed the receiving antenna gain to be calculated as a constant value of 6.55 dBi for the experiment.

6. Next, received signal strength readings were taken at various elevation angles and these values could be used to calculate the gain of the transmitting antenna at different elevation angles. These measurements were taken at 5 degree increments over the full range of motion of the tripod and their values were recorded. Additionally, although their values were not recorded, measurements of the received signal strength were observed for the transmitting antenna at a 0 degree elevation angle and a variety of azimuth angles. This was done to ensure that the horizontal cross section was circular as depicted in 4.2 above and indeed it was to a close enough approximation.

7. Three trials of these measurements were performed and they were averaged on a linear, as opposed to a logarithmic, scale. The gains of the transmitting antenna were calculated and a simple curve fit was performed for the antenna patterns. Figure 4.5 shows the data sheet from the experiment where the values important to the link budget are highlighted in orange and Figure 4.6 shows the antenna radiation pattern data for the vertical cross section plotted. As described above, this vertical cross section and the fact that the antenna is omnidirectional in its horizontal cross section is enough to reconstruct the full radiation pattern for the transmitting antenna. Also, based on the reciprocity argument from [24] the radiation pattern for the receiving antenna is assumed to be the same as the transmitting antenna. Therefore, both antenna radiation patterns have been determined and this concludes the experiment.



Antenna Lobe Pattern Testing

Parameters

GPS Location 1	GPS Location 2	Distance Between Antennas	Free Space Path Loss	Frequency
37.217500 N, 80.439933 W	37.213843 N, 80.438195 W	434 m	92.8040 dB	2.4 GHz
Transmit Power	**Antenna 1 Static Angle**	**Cable Type**	**Cable Length (Tx + Rx)**	**Cable Losses (Tx + Rx)**
1 W / 30 dBm	0°	L-com CA-195R	8 ft	1.486 dB
SMA Connector Type	**SMA Insertion Loss**	**TNC Connector Type**	**TNC Insertion Loss**	**Connector Losses (Total Tx + RX)**
L-com ASM-1714	0.20 dB (max)	L-com ATF-3700	0.20 dB (max)	0.80 dB (max)

Initial Readings

Antenna 2 Angle	Trial 1 (dBm)	Trial 2 (dBm)	Trial 3 (dBm)	Average Received Signal Strength (dBm)
0°	-52	-56	-52	-52
Calculated Antenna 0 Degree Gain =>				6.55 dBi

Readings

Antenna 2 Angle	Trial 1 (dBm)	Trial 2 (dBm)	Trial 3 (dBm)	Average Received Signal Strength (dBm)
0°	-52	-56	-52	-53
5°	-54	-56	-52	-54
10°	-56	-57	-56	-56
15°	-57	-59	-58	-58
20°	-60	-60	-60	-60
25°	-62	-64	-63	-63
30°	-64	-64	-64	-64
35°	-69	-68	-70	-69
40°	-68	-66	-68	-67
45°	-67	-69	-67	-68
50°	-66	-67	-66	-66
55°	-65	-67	-66	-66
60°	-66	-69	-67	-67
65°	-67	-70	-67	-68
70°	-68	-71	-68	-69
75°	-73	-75	-72	-73
80°	-81	-77	-80	-79
85°	-85	-78	-85	-81
90°	-85	-80	-84	-82
0°	-52	-56	-52	-53
-5°	-55	-58	-54	-55
-10°	-56	-59	-55	-56
-15°	-58	-61	-57	-58
-20°	-59	-62	-59	-60
-25°	-61	-64	-61	-62
-30°	-65	-70	-64	-66
-35°	-69	-77	-69	-70
-40°	-75	-78	-74	-75
-45°	-77	-80	-77	-78
-50°	-78	-81	-78	-79

Figure 4.5: Data sheet from antenna radiation pattern experiment

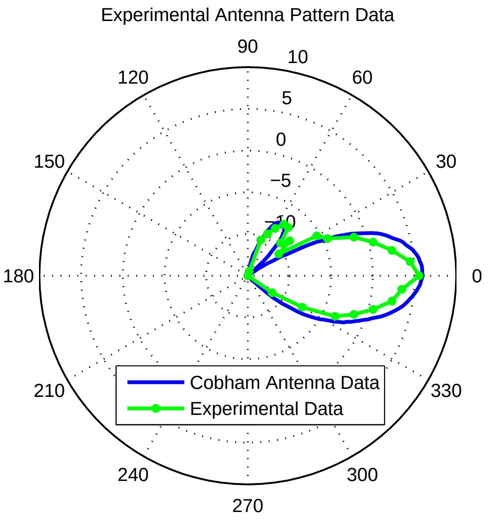


Figure 4.6: Experimental antenna radiation pattern data and Cobham data for antenna vertical cross section [42]

Looking at Figure 4.6, a few details can be noted. First, there is a relatively good agreement between the published Cobham antenna values [42] and between the experimental data taken. Both the main lobe and the side lobe are clearly defined in the experimental data and this helps to validate the published results. However, it should be noted that by inspecting the plot, there are some minor disagreements between the experimental and published data. For example, throughout a majority of the main lobe the experimental data displays gain values that are lower than the published result. This is not unexpected though and could be due to the fact that multipath signals may have caused destructive interference in the radio link even though the testing environment was relatively free of obstacles. Also, there were some measurements taken in the null zone between the main lobe and side lobe that exhibited a noticeable gain. This is most likely due to measurement error and the fact that only 5 degree increments of elevation angle were used in the experiment. Regardless of these few inconsistencies, the data generally agrees well and validates the published results for the antenna radiation patterns.

One last detail to note is that in order to effectively use these results in a path planning setting, curve fits of these antenna patterns would be useful. These curve fits have been calculated for the main lobe and the side lobe and are shown in Figure 4.7 below. The main lobe was curve fitted with a fourth order polynomial and the side lobe was curve fitted with a third order polynomial. These curve fits have norms of their residuals of 2.76 and 1.63 respectively which indicate strong fits with the Cobham data. These curve fits are given below and are used throughout the rest of the work to characterize the omnidirectional antenna gains in the radio system.

$$-16.93\phi^4 - 63.24\phi^3 - 68.07\phi^2 - 3.19\phi + 6.68 \;\; \forall\phi \in \left[\frac{-90\pi}{180}, \frac{25\pi}{180}\right] \;, \;\forall\theta \in [0, 2\pi]$$

$$19.00\phi^3 - 75.80\phi^2 + 88.81\phi - 41.22 \;\; \forall\phi \in \left[\frac{25\pi}{180}, \frac{90\pi}{180}\right] \;, \;\forall\theta \in [0, 2\pi]$$

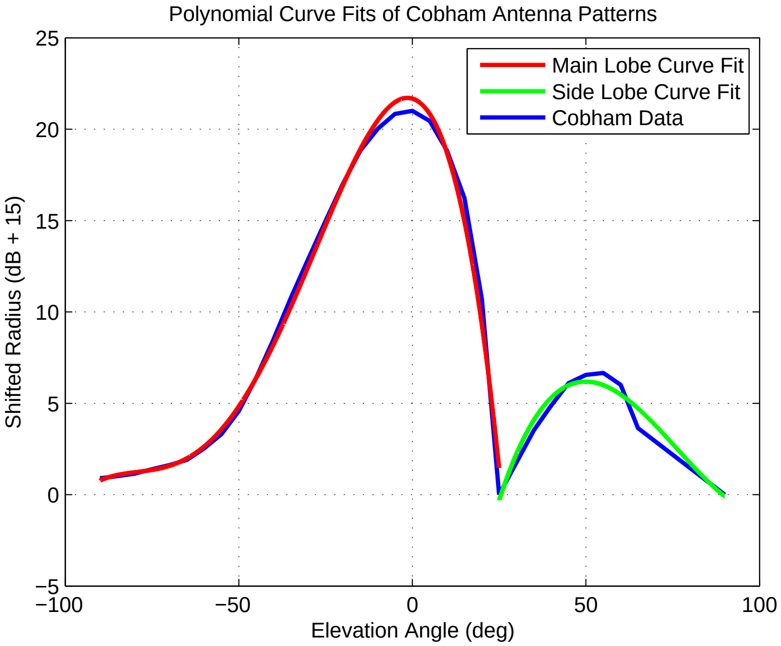


Figure 4.7: Curve fits for main lobe and side lobe of antenna radiation pattern vertical cross section, [42]

4.1.3 Path Loss Testing

With the radio testing for the antenna radiation patterns complete, the remaining variable that required characterization was the path loss L_{Path}. Section 4.1.1 provided a theoretical equation that represents the path loss for electromagnetic propagation in free space conditions. However, while this equation was valid for the scenario of the antenna pattern testing, path losses generally do not follow according to free space predictions. This is due to the fact that in an urban, suburban, or hilly environment, many obstacles exist that the radio waves can reflect or propagate off of. Because of this, often times many waves other than the direct line-of-sight ones can intersect the receiving antenna at the same time. Also, because the path lengths of the reflected signals are longer than the line-of-sight path, there is often a phase difference between the two incoming signals. This phenomenon known as multipath can substantially degrade the strength of the signal received by the receiving radio.

In order to quantify the effects of multipath, a great deal of work has been done in the radio community. As noted in the literature search, some studies have attempted to predict the received signal strengths in multipath environments by deterministic means [46], [7], [8]. These techniques are often very computationally intensive and are not well suited for path planning operations where calculations of received signal strength may need to be calculated thousands of times in a fraction of a second. Additionally, these methods often require very accurate knowledge of the propagation environment which is generally a poor assumption for disaster response scenarios [32].

Other studies have taken more of an empirical approach where experimental data has been used to generate curve fits of the path losses in a particular type of environment and these averaged results are then used as predictors for path losses in similar environments [22], [47]. Although care must be taken in using these empirical models since the environment must be very similar to the one in which the data was taken in, these relations are simply calculated and are well suited for path planning operations. Therefore, this work makes use of empirical formulations for the path planning operations.

Since path losses and multipath are extensively studied in the radio community, many empirical formulas exist for predicting path losses in different environments and there is still debate as to which are best or most appropriate [1], [2]. Additionally, since these relations are often given in qualitative terms like “urban”, “suburban”, “open” [22], and such, it is often difficult to assess which model is appropriate to which situation. Therefore, an experiment was performed to measure the path losses of the radio link in the environment where flight tests were to be performed and this data was used to select the most appropriate empirical model for the path planning process.

In order to quantify the approximate path loss that would be expected as a function of distance between the antennas, the path loss experiment simply consisted of setting one stationary antenna at a fixed location and measuring received signal strength values with a mobile antenna at a variety of other points throughout the environment. By using the link budgeting equation, the estimates for the cable and connector losses, the curve fits for the antenna radiation patterns, and the GPS positions and altitudes of the stationary and mobile antennas, the approximate path losses could be calculated. This path loss information was then used to determine the appropriate empirical model to predict the path losses at the experiment site.

Additionally, there are a few other details about the experiment that are worth mentioning. First, the experiment was performed at Kentland Farms in Blacksburg, Virginia. This site is commonly used as a test field for autonomous vehicle operations and is the only current location where the Unmanned Systems Lab is permitted to fly its unmanned aerial vehicles. Therefore, in order to most accurately model the radio path losses for the path planning, this site was used as the testing location for the experiment. Second, the stationary antenna was positioned at the top of a hill at Kentland Farms that was at a higher altitude than any of the other mobile antenna locations. This was done to mimic the idea of the helicopter being at an altitude higher than the ground vehicle or ground station in an actual radio repeating operation. Lastly, there were a number of measurements taken at different locations throughout Kentland Farms where some were in line-of-sight conditions with the stationary antenna and where some were not in line-of-sight conditions. This was done to see the effect of obstacles in the path losses. Figure 4.8 shows the locations where measurements were taken with green markers denoting line-of-sight conditions and red markers denoting non-line-of-sight conditions.

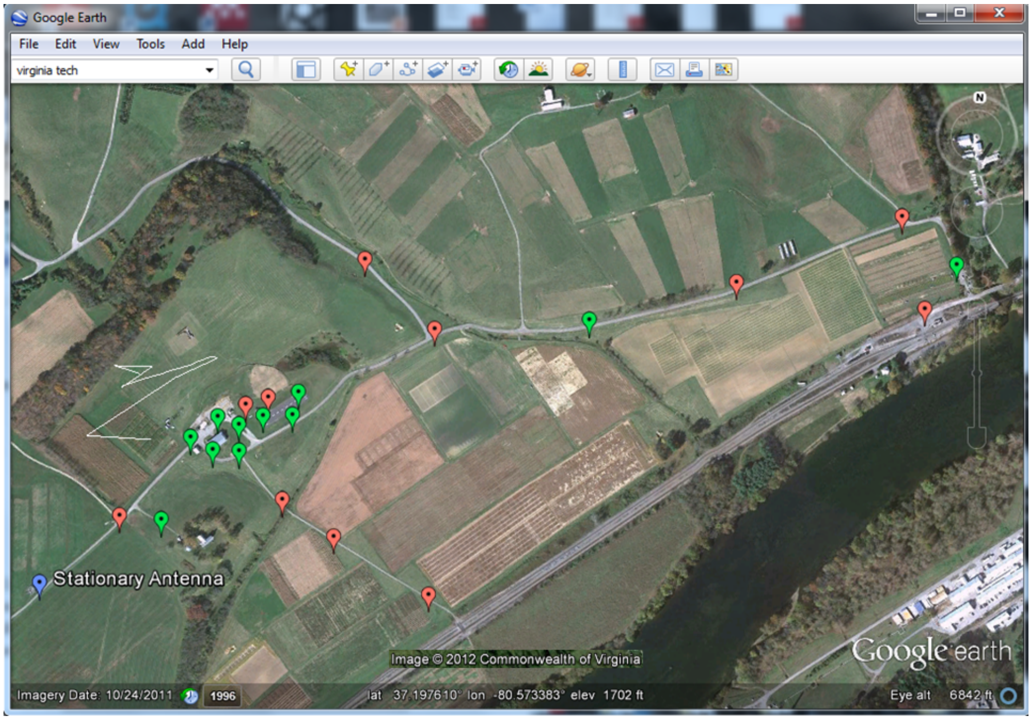

Google Earth
File Edit View Tools Add Help
virginia tech
N
Stationary Antenna
Image © 2012 Commonwealth of Virginia
Google earth
Imagery Date: 10/24/2011
1996
lat 37.197610° lon -80.573383° elev 1702 ft
Eye alt 6842 ft

Figure 4.8: Mobile antenna locations from radio testing at Kentland Farms overlaid in Google Earth (© 2012 Google) - Green markers indicate mobile antenna locations where line-of-sight conditions were available, red markers indicate mobile antenna locations where only non-line-of-sight conditions were available, and the blue marker indicates the stationary antenna

Figure 4.9 shows the data that was obtained from the experiment with some additional overlays showing three of the empirical models. As noted in the legend of the figure, the green data represents the path loss values from the line-of-sight locations and the red data represents the path loss values from the non-line-of-sight locations. Also, the blue curve represents the model of the free space path losses discussed in the Link Budgeting section and the orange and yellow curves represent the Hata Open and Hata Suburban empirical models, respectively. These Hata models are two commonly used empirical models for radio link planning [22] and are the two models of those considered that most closely fit the data obtained from the experiment. As can be seen, the free space path loss model does not seem to be a good predictor of the path losses seen in the experiment. Also, the line-of-sight data and the non-line-of-sight data tend to exhibit very different path loss trends and are moderately well predicted by the Hata Open and Suburban models, respectively. This claim can be validated by inspecting Table 4.1 which tabulates statistics relating to the residuals of the path loss data with respect to the various empirical models. As can be seen, by considering all of the data points as being predicted by a single model, the means of the residuals are relatively high in all three cases. However, when separating the data into two sets, one for line-of-sight (LOS) conditions and the other for non-line-of-sight (NLOS) conditions, and when using the Hata Open model as a predictor for the LOS points and the Hata Suburban for the NLOS points, the residual means and standard deviations are drastically improved. More so than this, by making use of two-tailed unpaired samples hypothesis tests, it was determined that there is a statistically significant change in the population means for the LOS and NLOS residuals [14] when comparing them to a single empirical model. This test was performed once to compare the residuals of the LOS and NLOS data to the Hata Open model and again to compare the residuals of the LOS and NLOS data to the Hata Suburban model. Both tests, as expected, resulted in the conclusion that there is a statistically significant difference between the means of the residuals for the LOS and NLOS data to a 99% confidence level ($p<0.01$). Also, for completeness, since the empirical models are linear when plotted against a logarithmic distance scale, the correlation coefficients for the LOS and NLOS data were also calculated to show that, on a logarithmic scale, they exhibited linear behavior as well. The correlation coefficient for the LOS data is $r_{xy} = 0.94$ and the correlation coefficient for the NLOS data is $r_{xy} = 0.79$. Comparing these to critical values of correlation coefficients as discussed in [13], both of these data sets exhibit a trend to the standard engineering 95% confidence level.

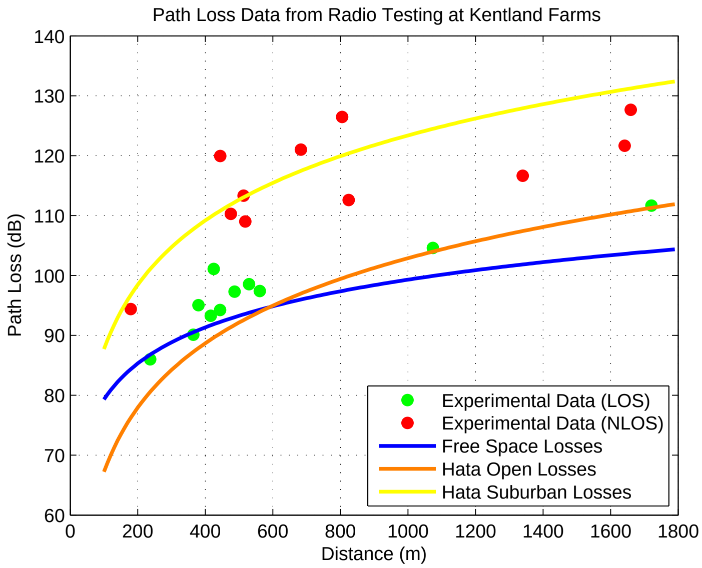


Figure 4.9: Path loss data from radio testing at Kentland Farms

Table 4.1: Means and standard deviations for path loss model residuals

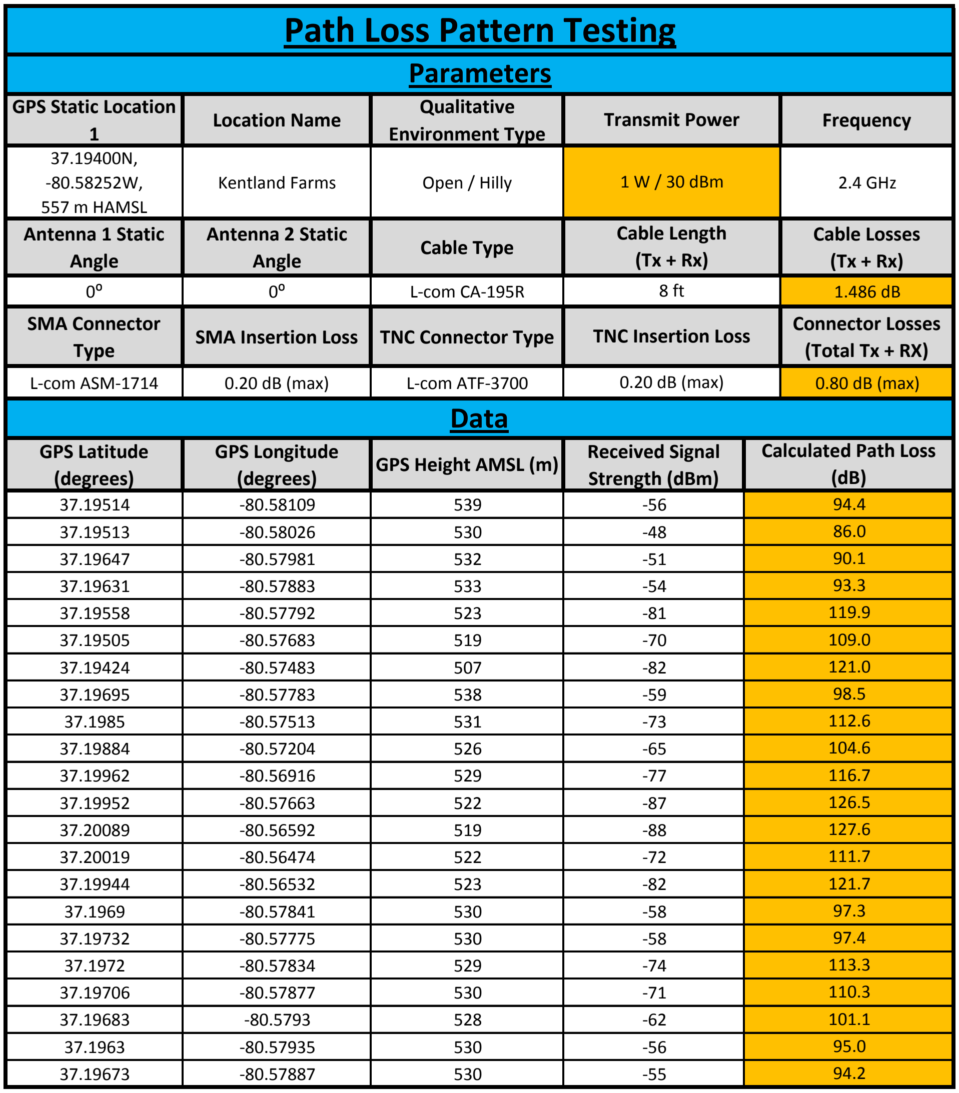
Residual Type	Mean	Std. Dev.
Free Space	11.66	9.42
Hata Open	12.37	8.70
Hata Suburban	-8.12	8.70
Hata Open (LOS)	5.33	3.03
Hata Suburban (NLOS)	-1.08	6.37

Therefore, based on these observations and calculations, the Hata Open model will be used to calculate the path losses for line-of-sight conditions and the Hata Suburban model will be used to calculate path losses for non-line-of-sight conditions. As noted in Figure 4.9, this is obviously contingent on the condition that the radios are separated by reasonable distances since the path loss models were only tested for data points of approximately 200 meter distances or greater. Many practical environments fall into this category and all of the simulations were performed on environments where this was the case so this is not an unreasonable assumption. For environments where the distances considered are less than this, it is recommended that the free space path loss model be used instead since the roll-off of the Hata models are too severe at low distance values. Lastly, for the sake of completeness, the experiment data sheet is provided in Figure 4.10 where the useful decibel values are shaded in orange.

4.1.4 Fresnel Ellipsoids

With all of the terms in the link budget sufficiently modeled for the given antennas, cabling, connectors, and propagation environment, this link budget can be used in a path planning setting to, on average, predict the anticipated received signal strength of a radio link. Because of this, a path planning scenario can be formulated where this radio budget can be used to predict the costs incurred by moving the vehicles into different configurations. However, as noted by inspecting the data plots of section 4.1.3, there is a significant path loss increase when the radios are configured in a non-line-of-sight setting as opposed to a line-of-sight setting. This results in a significant degradation of the received signal strength which is an undesirable result. Therefore, this radio modeling section concludes by discussing the concept of Fresnel ellipsoids, or Fresnel zones, which are used to tend the path planner towards line-of-sight behavior as the planning is performed.

As discussed in the path loss experiment section, due to the presence of obstacles in an environment, multipath propagation may occur which has the potential to destructively interfere with a received signal and may cause path losses in excess of free space path losses. More so, since electromagnetic propagation is based on a $\frac{1}{R^2}$ relationship, where R is the distance between the transmitting and receiving antenna, waves that travel shorter distances are stronger than waves that travel longer distances. Because of this, obstacles that are near the direct line between the transmitter and receiver can cause strong waves to reflect and cause more destructive interference than reflections that occur farther away from the direct line between the transmitter and receiver. Therefore, it is common practice among the radio community to ensure that the region near the direct line-of-sight between the antennas is free of obstacles [40]. This concept defines the Fresnel ellipsoids which are the surfaces



Path Loss Pattern Testing

Parameters

GPS Static Location 1	Location Name	Qualitative Environment Type	Transmit Power	Frequency
37.19400N, -80.58252W, 557 m HAMSL	Kentland Farms	Open / Hilly	1 W / 30 dBm	2.4 GHz
Antenna 1 Static Angle	**Antenna 2 Static Angle**	**Cable Type**	**Cable Length (Tx + Rx)**	**Cable Losses (Tx + Rx)**
0°	0°	L-com CA-195R	8 ft	1.486 dB
SMA Connector Type	**SMA Insertion Loss**	**TNC Connector Type**	**TNC Insertion Loss**	**Connector Losses (Total Tx + RX)**
L-com ASM-1714	0.20 dB (max)	L-com ATF-3700	0.20 dB (max)	0.80 dB (max)

Data

GPS Latitude (degrees)	GPS Longitude (degrees)	GPS Height AMSL (m)	Received Signal Strength (dBm)	Calculated Path Loss (dB)
37.19514	-80.58109	539	-56	94.4
37.19513	-80.58026	530	-48	86.0
37.19647	-80.57981	532	-51	90.1
37.19631	-80.57883	533	-54	93.3
37.19558	-80.57792	523	-81	119.9
37.19505	-80.57683	519	-70	109.0
37.19424	-80.57483	507	-82	121.0
37.19695	-80.57783	538	-59	98.5
37.1985	-80.57513	531	-73	112.6
37.19884	-80.57204	526	-65	104.6
37.19962	-80.56916	529	-77	116.7
37.19952	-80.57663	522	-87	126.5
37.20089	-80.56592	519	-88	127.6
37.20019	-80.56474	522	-72	111.7
37.19944	-80.56532	523	-82	121.7
37.1969	-80.57841	530	-58	97.3
37.19732	-80.57775	530	-58	97.4
37.1972	-80.57834	529	-74	113.3
37.19706	-80.57877	530	-71	110.3
37.19683	-80.5793	528	-62	101.1
37.1963	-80.57935	530	-56	95.0
37.19673	-80.57887	530	-55	94.2

Figure 4.10: Data sheet from path loss experiment

where waves reflecting off of an obstacle at that surface and towards the receiver are a half wavelength multiple out of phase with the direct line-of-sight path. For example, the first Fresnel ellipsoid is the surface where if a radio wave propagates off of an obstacle at the surface, it will be a half of a wave length out of phase with the direct line-of-sight wave when it approaches the receiving antenna. Appendix A contains a derivation of the first Fresnel ellipsoid and provides more detail about this concept. As a result of this derivation, the first Fresnel ellipsoid is defined by [40]

$$1^{st}\ Fresnel\ Ellipsoid\ \rightarrow\ \frac{x^2}{\left(\frac{D}{2}\right)^2}+\frac{y^2}{r_{max}^2}+\frac{z^2}{r_{max}^2}=1$$

where r_{max} is defined by

$$r_{max}=8.66\sqrt{\frac{D}{f}}\ \ ,\ \ where\ \ f\ [=]\ MHz\ ,\ D\ [=]\ m$$

It is a common rule of thumb [40] that if the first Fresnel ellipsoid is free of obstacles then free space conditions can be approximated for the path losses. While this is an approximation, it provides the insight that in the path planning operations, if the first Fresnel ellipsoid is kept free of obstacles, then the received signal strength will typically be higher than it would be otherwise. Therefore this is taken into consideration in the path planning process.

Lastly, it should be noted that the antenna testing above was performed in a region where the first Fresnel zone was clear of obstacles and because of this, free space conditions could be assumed in the path loss calculations. For reference, the maximum radius of the first Fresnel zone in the antenna test was $r_{max} = 3.68$ m and the ellipsoid defined by this radius was definitely free of obstacles in the antenna test.

4.2 A* Radio Repeating Implementation

With the radio link sufficiently modeled, the radio repeating path planning scenario can now be discussed. As alluded to in the introduction of this chapter, the radio repeating mission consists of a ground vehicle, an autonomous helicopter, and a ground control station. The ground vehicle, whether autonomous, semi-autonomous, or manned, has a desired location in an urban environment that it would like to reach. For motivational purposes, this could be anything from a location that first responders would like to investigate in a disaster response setting, a rendezvous point for a convoy in a mission in a military setting, or even just a general point of interest for a particular operation. Regardless of the setting, the goal of this mission is to navigate the ground vehicle effectively to that desired goal in the environment.

The complicating factor for this mission is that in an urban environment where many obstacles and potential pathways exist or in a hilly environment with large elevation changes in the terrain, the ground vehicle will not typically remain in a line-of-sight setting with the ground control station. This beyond-line-of-sight scenario causes drastically increased radio path losses for the radio propagation and a very degraded radio link health. In an autonomous or semi-autonomous setting, it is absolutely critical for the ground vehicle to remain in communication with the ground station both for providing a ground station operator with situational awareness and for allowing tele-operation or supervisorial commands to be sent to the ground vehicle. More so, if the ground vehicle is sent in a situation where the radio signal degrades too severely, then there is a risk of losing the radio link and therefore the unmanned vehicle as well. Also, even in a manned setting, it is often essential that a manned vehicle be in contact with a ground station for information relaying and mission status updates. Therefore, maintaining a strong communication link in an urban setting is of paramount importance.

In order to address this concern, the Unmanned Systems Lab is using an autonomous helicopter to act as a radio repeating node for relaying the radio signal from the ground control station to the ground vehicle. With this though, the question remains of how to effectively navigate the ground vehicle and the helicopter to effectively maintain a strong radio link. This portion of the chapter combats this issue by taking advantage of the path planning developments of the previous chapter and by using the radio modeling developments of the previous section to generate a cost function for the path planning search. This portion of the chapter is divided up into a number of sections. The first discusses the state space and action space for the path planning problem and some of the practical considerations that were taken into account when deciding upon these spaces. The next section describes the total cost function in detail as well as the various terms that comprise it. Following this, a simulated implementation of the radio repeating algorithm is discussed and its performance is assessed. Finally, an extension of this algorithm to multiple unmanned aerial vehicle scenarios is briefly discussed to broaden this concept to more intricate situations.

4.2.1 State Space

In this radio repeating path planning scenario, the motions of the ground vehicle and the aerial vehicle are coupled. Since the health of the radio links involved between the ground station and the aerial vehicle, and the aerial vehicle and the ground vehicle, respectively, are dependent upon the positions of both mobile vehicles, the vehicles are virtually tethered together by the radio links between them. Therefore, since maintaining the health of the radio links is a coupled problem, then the path planning problem and consequently the state space for the system is coupled as well.

In a general vehicle control setting, an aerial vehicle is usually modeled with 12 states consisting of three positions (x, y, z), three velocities (u, v, w), three angular attitudes (θ, ϕ, ψ), and three angular velocities (p, q, r). Also, assuming that it is restricted to a planar surface, a ground vehicle can be modeled with 5 states consisting of its (x, y) positions, its forward velocity u, its yaw angle ψ, and its yaw rate r. If all of these states were taken into account, the planning problem would contain 17 states which would make the search 17-dimensional and would be far too unwieldy for real-time or pseudo-real-time implementation.

Fortunately, many autonomous helicopters, inclusive of the one that the Unmanned Systems Lab uses as its primary experimental aircraft, have attitude stabilizing control systems which allow the helicopters to be commanded to a hover state where the velocities, angular attitudes, and angular velocities are all maintained at zero values. This type of system also allows the aircraft to be commanded to various positions as well. Therefore, rather than dealing with all 12 helicopter states, the path planning can be simply performed on the 3 position states (x, y, z) of the helicopter and the autopilot can maintain control of the rest.

For the ground vehicle, a similar approximation is made that because the ground vehicle will likely be tele-operated, manned, or installed with a tracking controller, only the (x, y) positions of the ground robot need to be planned. This simplifies the unwieldy 17 dimensional planning problem to a much more tractable 5 dimensional problem when considering both vehicles. However, due mainly to practical considerations, the z state of the helicopter position is also eliminated making it a 4 dimensional state space. The reason for this elimination is primarily for safety reasons in that when actually flying the autonomous helicopter, it is much safer to set it at a fixed altitude and to not change the vertical position of the helicopter.

With these simplifications, the path planning problem is reduced to a four dimensional state space with a state vector that contains the positions of both the ground vehicle and helicopter. This state vector is as follows:

$$\mathbf{x} = \begin{bmatrix} x_G \\ y_G \\ x_H \\ y_H \end{bmatrix}$$

Also, since the A* path planning method is a discrete search technique, the state space is represented as a 4-dimensional grid and for practical considerations, the grid is bounded by a region where a terrain map of the environment is known. Written mathematically, this state space can be written as

$$X = \{\mathbf{x} \in V | \mathbf{0} \leq \mathbf{x} \leq [x_{max}, y_{max}, x_{max}, y_{max}]^T\}$$

where the inequality is component-wise. Here, V represents the discretized grid that the search is performed in and is given by

$$V = \{\mathbf{v} \in \mathbb{R}^4 | \; |\mathbf{v}^{'} - \mathbf{v}| = \mathbf{r}, \mathbf{v}^{'} \in V\} \;\; , \;\; where \;\; r_i = \begin{cases} 0 \;,\; if \; v_i = v_i^{'} \\ \Delta v_i \;,\; if \; v_i \neq v_i^{'} \end{cases} \;, \; i = 1, ..., 4$$

Written in words, this mathematical description says that the space V is a discrete grid with a resolution of Δv_i in each of its coordinates and that the state space is the set of members of that grid that are between the bounds given by 0, x_{max}, and y_{max}. This state space is depicted in Figure 4.11 below.

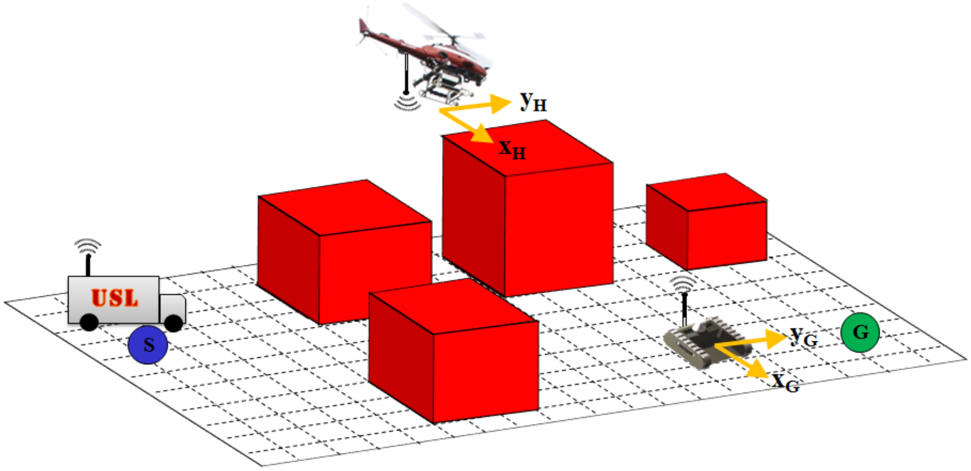


Figure 4.11: Diagram for radio repeating state space

The action space for this problem is simply the set movements to neighboring states of the current state being considered. For example, in a two-dimensional scenario, this would be movements to 8 neighbors, but for the four dimensional state space considered here, the action space contains the movements to the 80 neighboring states ($3^4 - 1 = 80$) of the state being considered. Again, because a waypoint path is desired and due to the simplicity of implementation of a neighbor-based action space, this action space is acceptable. Written mathematically, the action space is

$$U = \{\mathbf{u} \in \mathbb{R}^4 | u_i \in \{-\Delta v_i, 0, \Delta v_i\}\ ,\ i = 1, 2, 3, 4\} - \{\mathbf{0}\}$$

In summary, the state space is a bounded four-dimensional grid of resolution Δv_i in each coordinate and the action space is defined as the set of movements to the 80 neighboring states of a given state. With this formulation, not only has the state space been restricted to a region where a terrain map of an area is known, but because of the grid-based representation, it is easily implemented.

4.2.2 Cost Function

With the state space and action space clearly defined, the remaining component for the successful formulation of the radio repeating scenario as an A* path planning problem is to define a suitable cost function. This cost function essentially quantifies the performance of the search with respect to the radio repeating objective and penalizes behavior that is not conducive to achieving that objective. In this way, when the A* path planning is complete, the resulting paths will satisfy the radio repeating objective as defined by the cost function. As noted in the A* discussion above, the cost function is evaluated at every state transition and accrues as the search progresses. The cost function also primarily consists of two terms as shown below.

$$f = g + h$$

The g term is the main term that caters the search to the radio repeating objective, while the h term is simply a heuristic that effectively guides the search. Therefore, the g cost is the one that defines the radio repeating objective and the h heuristic essentially just increases the speed and efficiency of the search process by guiding the search toward its goal. With this in mind, the g cost for the radio repeating objective has been formulated as consisting of a variety of terms, each of which is responsible for a specific performance objective. These are given in a list format and are described in detail below.

1. *Ground Vehicle Distance Cost* - This cost represents the amount of distance that is traveled by the ground vehicle throughout the search. It accrues every time the search considers a state where the ground vehicle has moved from the previous state.

2. *Radio Link Budget Cost* - This cost represents the overall health of the radio repeating links. It accrues a value at each state transition based on the predicted received signal strengths of the radio repeating links. Because received signal strength is used as the metric for this cost, it takes into consideration the antenna gains, the path losses, and the other various factors contributing to the health of the radio links.

3. *Fresnel Ellipsoid Cost* - This cost penalizes intersections between various obstacles in the terrain map and the first Fresnel zones for the radio repeating links. This cost essentially attempts to tend the resulting paths to line-of-sight conditions since line-of-sight conditions have been shown to provide stronger links than non-line-of-sight conditions.

4. *Radio Link Difference Cost* - This cost also uses the received signal strength as a metric but slightly penalizes the search if there is a difference in received signal strengths between the radio repeating links. This cost attempts to ensure that the search does not bias towards keeping one link healthy while neglecting another.

5. *Out of Bounds Cost* - This cost highly penalizes the search if either the ground vehicle or the helicopter attempts to go outside of the state space defined by the terrain map of the area. It ensures that the search stays within a region that a map has been provided for.

6. *Wall Intersection Cost* - This cost highly penalizes the ground vehicle and the helicopter for intersecting a wall, building, or other obstacle. It ensures that neither vehicle

attempts to traverse through an obstacle in the terrain map.

Describing each of these costs in more detail and starting with the Ground Vehicle Distance Cost, this cost was implemented to keep the ground vehicle's path between its start and goal locations as short as possible. The overall objective of the search is to navigate the ground vehicle to a specific location while maintaining strong radio communications, so keeping the path length of the ground vehicle as short as possible is a definite objective. This cost operates by accruing the amount of distance traveled by the ground vehicle throughout the search. It is described mathematically below.

$$g_{GVDC} = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2$$

where $\mathbf{x}_i$ is a state being navigated to, $\mathbf{x}_{i-1}$ is the previous state or parent of $\mathbf{x}_i$, and n is the amount of states in the current path from the start state to the current state $\mathbf{x}_n$, exclusive of the start state itself.

Next, the Radio Link Budget Cost was implemented to predict and quantify the strength of the radio links used in the radio repeating scenario. As noted in section 4.1.1 above, this term consists of determining the received signal strengths of the links by adding and subtracting the various gains and losses associated with the ground station-to-helicopter and the helicopter-to-ground vehicle radio links. Then, the received signal strengths are used to accrue a radio link cost for every state transition. Starting with the radio link budget, this is calculated as follows.

$$P_r = P_t - L_C + G_t - L_{Path} + G_r$$

Here the transmit power was set to $P_t = 30$ dBm for the 1 W radios used, the L_C term was set to $L_C = 2.286$ dB to quantify the losses from the cabling and connectors on both the transmitting and receiving radios, and the antenna gains G_t and G_r were determined by using the antenna pattern curve fits from section 4.1.2 which are repeated here for convenience.

$$G_{t/r} = -16.93\phi^4 - 63.24\phi^3 - 68.07\phi^2 - 3.19\phi + 6.68 \ \ \forall\phi \in \left[\frac{-90\pi}{180}, \frac{25\pi}{180}\right] \ , \ \forall\theta \in [0, 2\pi]$$

$$G_{t/r} = 19.00\phi^3 - 75.80\phi^2 + 88.81\phi - 41.22 \ \ \forall\phi \in \left[\frac{25\pi}{180}, \frac{90\pi}{180}\right] \ , \ \forall\theta \in [0, 2\pi]$$

Also, as determined in section 4.1.3, line-of-sight path losses for the Kentland Farms testing area can be, on average, quantified by the Hata Open model and non-line-of-sight path losses for the Kentland Farms testing area can be, on average, quantified by the Hata Suburban model. These models are therefore used for the path loss term in the link budget depending on whether or not line-of-sight conditions are met. These models were obtained from [22] and are presented below. For line-of-sight conditions, the Hata Open model that can be used has a path loss prediction given by

$$L_{Path} = L_p - 4.78(\log_{10} f)^2 + 18.33\log_{10} f - 40.94$$

where f is the frequency of the radio communication in MHz and the L_p term is defined as

$$L_p = 69.55 + 26.16\log_{10} f - 13.82\log_{10} h_b - a_h + (44.9 - 6.55\log_{10} h_b)\log_{10} R$$

where h_b is the height of the helicopter's antenna in meters, R is the distance between antennas in kilometers, and a_h is defined as

$$a_h = (1.1\log_{10} f - 0.7)h_m - (1.56\log_{10} f - 0.8)$$

where h_m is the ground vehicle's antenna height in meters. For non-line-of-sight scenarios, the Hata Suburban model gives a path loss prediction of

$$L_{Path} = L_p - 2(\log_{10}\left(\frac{f}{28}\right))^2 - 5.4$$

where L_p is the same as defined above. With all of the terms for the radio link budget quantified, the Radio Link Budget Cost is written as

$$g_{RLBC} = \sum_{i=1}^{n} \|\mathbf{P}_{Received}\|_2$$

where

$$\mathbf{P}_{Received} = \begin{bmatrix} (P_t - P_r)_{GroundStation-to-Heli} \\ (P_t - P_r)_{Heli-to-GroundVehicle} \end{bmatrix}$$

Here, the vector $\mathbf{P}_{Received}$ contains terms for both of the links in the radio repeating scenario. Also, since a cost function is typically a non-negative quantity, the difference $P_t - P_r$ has been used to quantify the health of the link. As is expected, a lower value for the received signal strength, P_r, is representative of a less healthy link, which results in a larger value for $P_t - P_r$ and correspondingly a higher penalty in the cost function. With this cost term, less healthy links are penalized and therefore the overall strength of the radio signals are typically stronger by including this term in the cost function.

Thirdly, a cost is included in order to penalize obstacle intersections with the first Fresnel ellipsoid for each of the radio links. As noted in section 4.1.4, stronger radio links are obtained by not only keeping the line-of-sight path from the transmitter to the receiver clear of obstacles, but also by keeping the first Fresnel ellipsoid clear of obstacles as well. By incorporating a cost to penalize intersections with the first Fresnel ellipsoids of the radio links, the destructive effects of multipath propagation can be reduced in addition to also tending the search to more line-of-sight conditions than would be done otherwise. Since this Fresnel ellipsoid concept is a geometric one, the Fresnel Ellipsoid Cost is accrued by determining the first Fresnel ellipsoids at each state and checking for intersections with the terrain map. Briefly, this process is given as follows.

First, an ellipsoid is determined that has the appropriate radii as given by the equations of section 4.1.4.

$$\frac{x^2}{\left(\frac{D}{2}\right)^2} + \frac{y^2}{r_{max}^2} + \frac{z^2}{r_{max}^2} = 1$$

where r_{max} is defined by

$$r_{max} = 8.66\sqrt{\frac{D}{f}} \quad , \quad where \quad f \; [=] \; MHz \; , \; D \; [=] \; m$$

This ellipsoid is sampled at a number of points along four cross sections. The first is the vertical cross section, the second is the horizontal cross section, and the third and fourth are 45 degree cross sections. These cross sections are shown for a general ellipsoid in Figure 4.12.

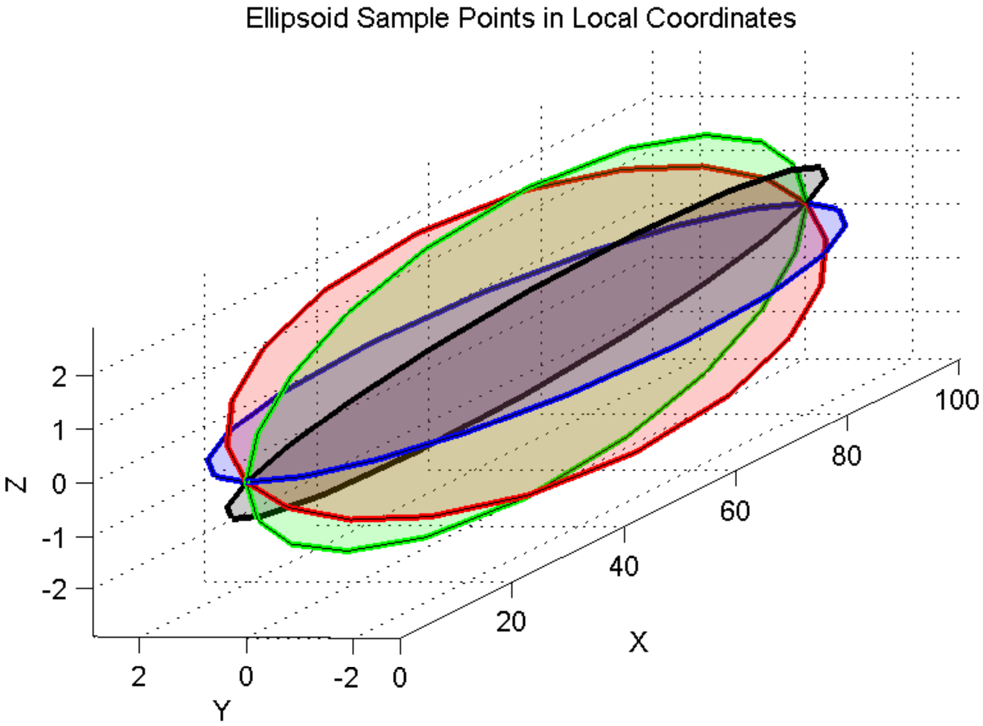


Figure 4.12: Example of first Fresnel Ellipsoid cross sections

After this is complete, the sampled cross sections are rotated and translated to their appropriate locations in the state space by applying the appropriate yaw rotation, pitch rotation, and translation given in the equation below.

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\sin(\psi) & -\sin(\theta)\cos(\psi) \\ \cos(\theta)\sin(\psi) & \cos(\psi) & -\sin(\theta)\sin(\psi) \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \\ z_t \end{bmatrix}$$

Here, θ is the pitch angle of the rotation, ψ is the yaw angle of the rotation, $\begin{bmatrix} x_0 & y_0 & z_0 \end{bmatrix}^T$ is a point on one of the sampled cross sections of the ellipsoid, $\begin{bmatrix} x_t & y_t & z_t \end{bmatrix}^T$ is the translation applied to the ellipsoid, which is the set of coordinates for the ground station in one of the links and the set for the helicopter in the other, and $\begin{bmatrix} x_1 & y_1 & z_1 \end{bmatrix}^T$ is the resulting point after the proper rotations and translations.

Then, the resulting rotated and translated Fresnel ellipsoid cross section points are coerced to their nearest locations in the grid defined by the state space and these coerced points are inspected for intersections with obstacles in the terrain map. The amount of protrusion of the obstacles into the first Fresnel ellipsoid is summed and this is used to define the Fresnel Ellipsoid Cost. The equations for this are process are given below.

$$F_{Diff} = \textsc{MapHeight}(x_1, y_1) - z_1$$

$$F_{Cost} = \begin{cases} F_{Diff} \ , \ if \ F_{Diff} > 0 \\ 0 \qquad\ , \ if \ F_{Diff} \leq 0 \end{cases}$$

$$F_{Sum} = \sum F_{Cost}$$

$$g_{FEC} = 100 \sum_{i=1}^{n} \left(F_{Sum,GroundStation-to-Heli} + F_{Sum,Heli-to-GroundVehicle} \right)$$

In the cost function term g_{FEC}, the sum consists of both of the Fresnel ellipsoid intersection sums for the two radio links and is performed over the states along the search path. Here, it should be noted that the factor of 100 was included as a gain to get the appropriate behavior for the path planner and was simply determined in an ad hoc manner. It was determined that this value makes the Fresnel Ellipsoid Cost comparable to the other radio signal cost function terms and therefore allows this term to have some significance in the overall path planning.

Next, while the Radio Link Budget Cost and the Fresnel Ellipsoid Cost serve to provide desired performance in the path planning with respect to the radio repeating objectives, some simulations have shown that one radio link may tend to get prioritized over the other. When this occurs, the resulting paths tend to have one very strong link and one very weak link. This becomes problematic because in a radio communication setting, both links need to be moderately strong and a severe degradation in one link could compromise the whole operation. This is an artifact of the path loss models having a logarithmic nature. Therefore, an additional term was included as a Radio Link Difference Cost to abate this issue. This term in the cost function penalizes the differences between the radio links and attempts to keep them relatively even throughout the operation. The cost for this term is given below.

$$g_{RLDC} = \sum_{i=1}^{n} |(P_t - P_r)_{GroundStation-to-Heli} - (P_t - P_r)_{Heli-to-GroundVehicle}|$$

The terms in this sum are the same terms used in the Radio Link Budget Cost above and as with the other terms, the sum is over the states along the search path.

The penultimate term in the g cost of the radio repeating cost function is a cost that penalizes the search if it goes out of bounds of the state space. The last term is a cost that penalizes intersections of the vehicles with obstacles in the terrain map. Both of these terms manifest themselves as binary costs of either zero if no violation is made or infinity if a violation is made. For the purposes of documentation, infinity will be used for the cost, but when implemented numerically, these costs were just set to very high cost values instead of strictly infinity. The costs for these terms are given below.

$$g_{OoBC} = \begin{cases} 0 \ , \ if \ \textsc{OutOfBounds}(\mathbf{x}_n) = \textsc{false} \\ \infty \ , \ if \ \textsc{OutOfBounds}(\mathbf{x}_n) = \textsc{true} \end{cases}$$

$$g_{WIC} = \begin{cases} 0 \ , \ if \ \textsc{WallIntersect}(\mathbf{x}_n) = \textsc{false} \\ \infty \ , \ if \ \textsc{WallIntersect}(\mathbf{x}_n) = \textsc{true} \end{cases}$$

Combining these costs together gives the overall g cost for the radio repeating path planning algorithm which defines the behavior for the search process. This cost is given below and as can be seen by inspecting the various terms, this cost seeks to cater the search to help minimize the distance traveled by the ground vehicle while also working to help maximize the health of the radio repeating radio links. By inspecting the relative magnitudes of the terms in the g cost, the search will tend to prioritize performance that maintains line-of-sight behavior and maximizes the received signal strengths of the radio links.

$$g = g_{GVDC} + g_{RLBC} + g_{FEC} + g_{RLDC} + g_{OoBC} + g_{WIC}$$

Lastly, the cost function f also includes a heuristic term h. For the radio repeating scenario, the heuristic is a distance-based measure that guides the ground vehicle toward its goal. Since the state space is four-dimensional and because the guiding heuristic is only a two-dimensional heuristic that guides the ground vehicle to its goal location, it is pivotal that the heuristic effectively guides the search. Therefore, both the inflation factor ϵ is applied to the heuristic, making the search a WA* variant, and a heuristic look-up table is used as well. As discussed in section 3.6, this heuristic look-up table more accurately guides the search which minimizes the amount of states that require inspecting in order to determine a path to the goal. It should also be noted that heuristic terms were not included for the other cost function components. This is due to the fact that a strict goal location for the helicopter is typically unknown before the search is executed and it would therefore be difficult to generate an informative heuristic for the remainder of the cost terms. Therefore, the final cost function for this path planning algorithm is given as follows.

$$f = g_{GVDC} + g_{RLBC} + g_{FEC} + g_{RLDC} + g_{OoBC} + g_{WIC} + \epsilon h_{HLUT}$$

4.2.3 Implementation and Testing

With the state space and cost function for the radio repeating path planning algorithm clearly developed, the path planning algorithm was simulated using the Matlab coding language. In order to test the performance of the radio repeating path planner, a number of randomized simulations were performed with the intent of determining an objective and unbiased assessment of the planner's performance. In each set of simulations, 10 random start and goal locations were chosen for the ground vehicle where the start positions were confined to one side of the environment being tested and the goal locations were confined to the opposite side of the map. This restriction was chosen to maintain consistency between the simulations and so that the ground vehicle would always have to traverse through the bulk of the map in its path planning. Additionally, also to maintain consistency between the simulations, the aerial vehicle's start position was always placed immediately adjacent to the ground vehicle and the ground station was at a fixed point towards one corner of the environment. An example start configuration for the system is shown in Figure 4.13.

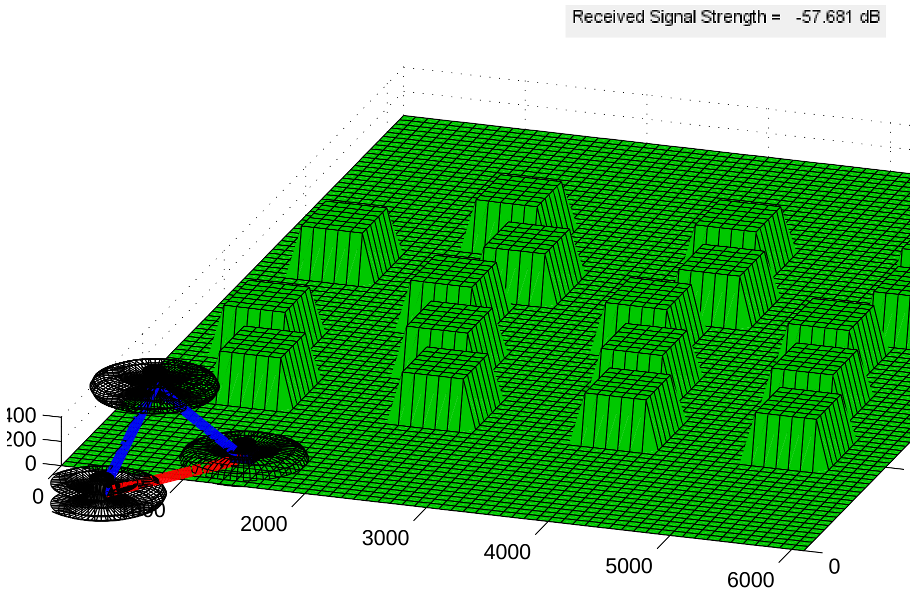


Figure 4.13: Example of starting configuration for radio repeating simulations

By inspecting the figure, it may be intuitive that different arrangements of obstacles may result in different performances from the algorithm. As one might expect, the more clutter there is in the environment, the less likely the planner may be able to maintain line-of-sight conditions and the planner's performance may suffer accordingly. Therefore, in addition to simply testing different start and goal locations for the same environment, a variety of environments were tested. More specifically, fifteen environments were tested in total with the environments segregated into three distinct categories. These classes of environments all featured different building geometries and have their features tabulated in Table 4.2. For the sake of narration, they were arbitrarily assigned the designations of Type I, Type II, and Type III environments.

As noted in the table, Type I maps are meant to model an open or farm type of environment where buildings are sparse and unordered. Type II maps are intended to model a campus type of environment where buildings are still relatively sparsely packed but more orderly in nature. Lastly, Type III maps are indicative of a suburban environment where buildings

Table 4.2: Radio repeating simulation map type features

Map Type	Type I	Type II	Type III
Features			
Sparsely/Densely Populated?	S	S	D
Buildings Ordered/Sporadic?	S	O	O
Environment Analog	Farm/Open	Campus	Suburb

are ordered and densely packed. Examples of these map types are shown in Figure 4.14. Coupling the fact that five environments were tested of each type and the fact that 10 simulations were run on each environment, 150 simulations were performed in total to assess the performance of the radio repeating path planner.

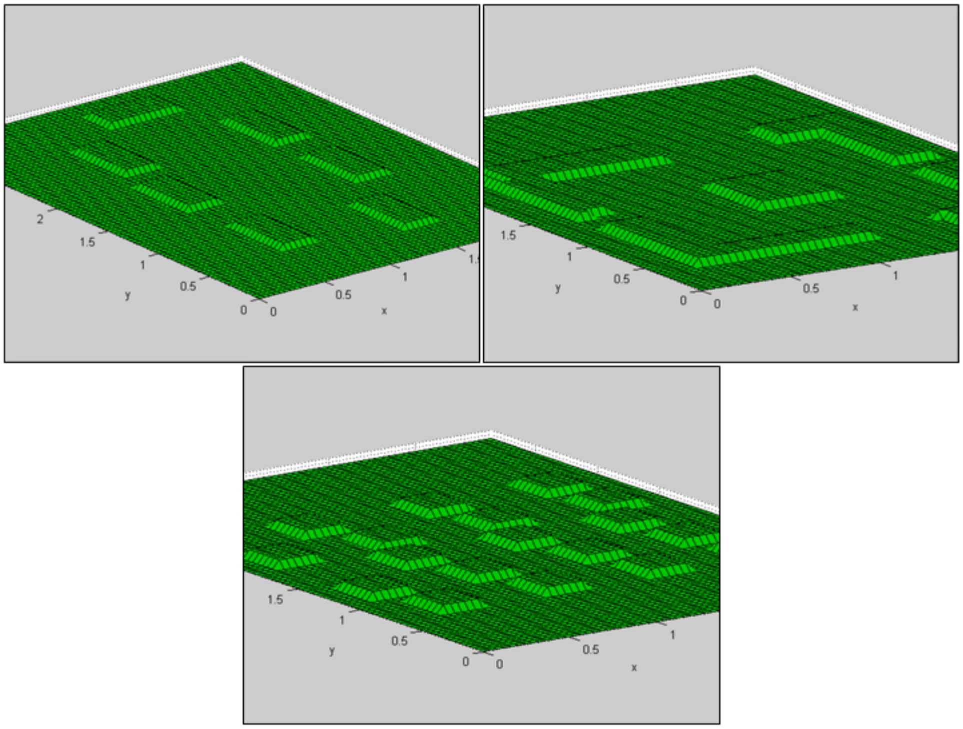


Figure 4.14: Example maps for radio repeating simulation environment types: (Top Left) Type I, (Top Right) Type II, (Bottom Center) Type III

More so, since it would not be too enlightening to test the radio repeating path planner without a baseline for comparison, four other naïve radio repeating variants were also formulated and tested on the same batch of environments as the radio repeating path planner. Each of these variants, which were generated by the author, plan a path for the ground vehicle but take a more simplified approach to navigating the helicopter. These serve as a useful group to compare the radio repeating path planner's performance with since they represent fairly intuitive but naïve ways to perform the radio repeating operation. These variants are described below.

Naïve 1 This radio repeating variant positions the helicopter at a single location at the center of the environment in question. One might believe that by positioning the helicopter in the center of the environment radio signal strengths will be reasonable over most points on the map. However, due to the non-line-of-sight conditions imposed by obstacles, this tends to not be the case in obstacle-rich environments.

Naïve 2 This variant strictly positions the helicopter above the goal position for the ground vehicle. Along the same lines as the first naïve variant, line-of-sight conditions could easily be obstructed for a large duration of the ground vehicle's travel.

Naïve 3 This variant maintains the helicopter over the ground vehicle as it traverses from its start to goal position. While the mobility of the helicopter may provide a false sense of confidence that this may enhance the communication with the ground station, since the ground vehicle will likely travel in non-line-of-sight scenarios with the ground station, so may the helicopter as it strictly travels above it. More so, with the helicopter directly above the ground vehicle, each of the vehicles are in a null zone of the other's antenna resulting in poor signal strengths.

Naïve 4 This last variant positions the helicopter at the center of the ground vehicle's path and maintains it there. While this could be the most informed naïve variant, the lack of mobility of the helicopter may cause non-line-of-sight conditions to occur.

The settings used in the various simulations are provided in Table 4.3 and the averaged

Table 4.3: Radio repeating simulation settings

Grid Size	Resolution (in)	Epsilon
5270	100	10000
Radio Frequency (GHz)	Radio Transmit Power (dBm)	Helicopter Height (m)
2.4	30	20

results from testing the radio repeating path planner and the naïve radio repeating variants are tabulated in Table 4.4. Although only one height was considered for the helicopter, simulations at other reasonable heights show that the simulation results presented are indicative of the performance of the algorithm since the same qualitative performance is seen at other heights as well. As can be seen by inspecting the Table 4.4, generally the radio repeating path planner outperforms the naïve variants in both the average received signal strengths for each of the radio repeating links and in terms of how much line-of-sight behavior was maintained throughout the simulation.

While the table presents a large amount of data to compare, a number of trends seem to emerge from inspecting the data. First, the radio repeating path planner seems to produce relatively equal signal strengths for each of the links. The planner seems to produce averages for both of the radio links between -53 and -58 dBm and the lower signal strength values only result when fewer line-of-sight scenarios could be maintained between the vehicles. This performance intuitively makes sense since the planner attempts to not only maintain line-of-sight but has cost function terms that attempt to preserve the received signal strengths of both radio repeating links. Therefore, the helicopter tends to stay relatively in the middle between the ground station and the ground vehicle thereby maintaining reasonably strong signal strengths for each link. This is in stark contrast to some of the naïve variants which tend to bias one link over the other. For example, Naïve II positions the helicopter strictly over the goal location for the ground vehicle, which tends to sacrifice the GS-to-Heli link while the Heli-to-GV link tends to improve as the ground vehicle approaches the goal.

Table 4.4: Average radio repeating path planner simulation results as compared to naïve radio repeating variants

Simulation	Plan	Naïve I	Naïve II	Naïve III	Naïve IV
Start/Goal Sets per Map	10	10	10	10	10
Number of Maps per Type	5	5	5	5	5
Overall Average Simulation Results					
Closed States	80	80	80	80	80
Open States	3847	202	195	194	195
Computation Time	6.59	0.87	0.87	0.80	0.87
Map Type I Average Results					
Avg RSS GS-to-Heli	-53.7	-68.8	-74.7	-62.1	-64.7
% NLOS GS-to-Heli	5	60	58	28	42
Avg RSS Heli-to-GV	-53.6	-50.2	-58.0	-66.3	-51.1
% NLOS Heli-to-GV	12	11	28	0	16
Map Type II Average Results					
Avg RSS GS-to-Heli	-53.2	-64.3	-79.3	-64.0	-63.9
% NLOS GS-to-Heli	5	40	76	36	38
Avg RSS Heli-to-GV	-57.5	-57.9	-65.2	-66.3	-56.6
% NLOS Heli-to-GV	34	60	57	0	30
Map Type III Average Results					
Avg RSS GS-to-Heli	-53.9	-68.8	-83.4	-64.5	-66.0
% NLOS GS-to-Heli	9	60	92	39	48
Avg RSS Heli-to-GV	-53.3	-55.1	-62.1	-66.3	-53.9
% NLOS Heli-to-GV	17	40	42	0	29

Secondly, the radio repeating path planner seems to provide pretty consistent performance despite the differences in the environment types. This tends to occur because regardless of the clutter in each of the environments, the fact that the helicopter is placed at an altitude above the buildings and that the environments do have a few uncluttered regions, the helicopter is able to maintain a large percentage of line-of-sight conditions throughout the simulations. This tends not to be the case for some of the naïve variants where the amount of building clutter seems to affect the amount of line-of-sight conditions that the variants can produce. For example, Naïve II, which positions the helicopter strictly over the goal location does not have the freedom to move the helicopter around and therefore is less likely to maintain line-of-sight conditions in more cluttered environments.

Thirdly, by actively considering the helicopter's navigation in the path planner, it seems to abate some of the deficiencies in the naïve variants. For example, Naïve I positions the helicopter at the center of the map which imposes two problems. The first is that since both the ground station and helicopter positions are fixed throughout the search process, the GS-to-Heli link is fixed as well and if this link is in a non-line-of-sight scenario, it will cause degraded signal strengths for that link. Also, due to the fixed position of the helicopter, the line-of-sight scenarios may be intermittent throughout the ground vehicle's navigation. Next, for Naïve II, the helicopter is situated strictly above the goal which can cause similar issues as Naïve I for the GS-to-Heli link but also can cause degraded signal strengths in the Heli-to-GV link when the ground vehicle is close to the start position of the search. For Naïve III, similar non-line-of-sight issues are found for the GS-to-Heli link and while it may deceptively seem that the Heli-to-GV link may be healthy, the antenna pattern null regions tended to cause degraded strengths in the Heli-to-GV link. Lastly, for Naïve IV, similar intermittent non-line-of-sight patterns can be observed. The path planner tends to alleviate many of these deficiencies by always searching for a better position for the helicopter in terms of not only line-of-sight, but also in terms of the antenna patterns, path losses, and ground vehicle location. By inspecting the data and noting that path planner outperforms the naïve variants, these claims have some validity.

Table 4.5: Best case radio repeating path planner simulation results as compared to naïve radio repeating variants

Simulation	Plan	Naïve I	Naïve II	Naïve III	Naïve IV
Map Type I Best Case Results					
Avg RSS GS-to-Heli	-48.9	-77.7	-59.4	-53.6	-54.1
% NLOS GS-to-Heli	0	100	0	0	0
Avg RSS Heli-to-GV	-51.2	-62.7	-67.8	-66.3	-50.1
% NLOS Heli-to-GV	1	74	71	0	10
Map Type II Best Case Results					
Avg RSS GS-to-Heli	-53.9	-55.4	-85.3	-59.5	-55.0
% NLOS GS-to-Heli	0	0	100	19	0
Avg RSS Heli-to-GV	-52.1	-58.8	-67.6	-66.3	-60.3
% NLOS Heli-to-GV	1	68	69	0	74
Map Type III Best Case Results					
Avg RSS GS-to-Heli	-51.8	-77.7	-85.3	-57.7	-54.9
% NLOS GS-to-Heli	0	100	100	11	0
Avg RSS Heli-to-GV	-50.7	-48.0	-58.2	-66.3	-53.4
% NLOS Heli-to-GV	1	0	23	0	26

Lastly, despite the fact that on average the path planner outperforms the naïve variants, it is shown in Table 4.5 that under the appropriate set of conditions for the start and goal points, the path planner can produce much better results than the naïve counterparts for the same set of conditions. Looking at Table 4.5, for some conditions, the planner can almost maintain complete line-of-sight throughout the operation and can produce signal strengths that are indicative of a very healthy radio link. This is in contrast to some of the naïve variants that still produce poor results for the chosen set of start and goal conditions.

Despite this seemingly spectacular performance, the radio repeating path planner does exhibit some weaknesses worth discussing in order to give an honest assessment of its abilities. First, due to the coupled nature of the state space and its four-dimensional nature, the path planner requires consideration of many more open states than the two-dimensional naïve variants in its planning process. Coupling this with the fact that the Fresnel ellipsoid cost function term is moderately computationally intensive to evaluate, the radio repeating path planner has a much higher computation time than the naïve variants even with its binary heap and hash table implementation. This would be even more exacerbated if the state space grew larger in size. Regardless, from the perspective of the radio repeating operation, it is not completely time critical that the paths be calculated in strict real-time. Typically, pseudo-real-time performance is acceptable and this average 6.59 second delay for planning will not compromise the overall operation. Additionally, although the comparative computational speeds between the path planner and the naïve variants are not likely to change, the path planner was rewritten in C++ for implementation with a real vehicle system which, from experience, tends to exhibit faster computation speeds than Matlab-based code. Therefore, planning would in actuality require less than this 6.59 second average.

Additionally, the radio repeating algorithm will likely not be successful in environments where there is a very significant amount of building clutter and where the buildings are tall and potentially have heights at or above that of the helicopter. These environments may be characteristic of completely urban areas and as may be expected, line-of-sight is difficult to maintain in these environments. Therefore, it is anticipated that the radio repeating path planner will not provide much better performance than the naïve counterparts for heavily urban environments and thus, the path planner was not exhaustively tested in these environments. Instead, the next section provides a brief description of potentially extending this work to multiple aerial vehicle scenarios if heavily urban environments are encountered.

Overall though, simulation and testing has shown that the radio repeating algorithm does produce desirable signal strength results for a variety of different open, campus, and suburban simulated environments. More so, the planner tends to outperform the naïve counterparts both in terms of the best case results and on average. Therefore, in simulation, the path planner is considered to be successful at effectively accomplishing the air/ground vehicle radio repeating operation. The only apparent deficiency in the algorithm is that by using empirical models for path loss calculations, the simulated signal strengths are averaged results and may not capture all of the complexities of radio multipath for intricate non-line-of-sight scenarios. However, these models were required for computational efficiency and based on the paths generated by the path planner, the paths produce vehicle behaviors that experience has shown to be indicative of healthy radio links. In order to evaluate this path planner even more thoroughly, Chapter 5 includes testing on an actual vehicle system and comparisons are made to gain confidence that the simulated paths exhibit the desired behavior for the radio repeating operation.

4.2.4 Extension to Multiple Aerial Vehicles

One of the interesting features of the A* based approach to path planning is that it can be grid-based in nature and is extensible to as many dimensions as the search requires. This provides two distinct benefits to using this approach for the radio repeating algorithm. The first is that the approach is orderly and systematic and the search can be easily modified to a variety of scenarios. The second, and the primary focus of this section, is that the radio repeating algorithm can be easily extended to a higher-dimensional search where more than one aerial vehicle can be included.

The purpose of this extension is the fact that with simply one aerial vehicle as a radio repeater, line-of-sight or close to line-of-sight conditions cannot be maintained for all environments. If an urban environment is considered with a complex building structure, it may not be possible to maintain line-of-sight or even near line-of-sight behavior with just one aerial vehicle. Fortunately, due to the general nature of the A* process, more aerial vehicles can be added to the search process and more radio links can be considered. For example, screen shots of simulations are shown in Figure 4.15 for a case where two and three helicopters are considered in the search process.

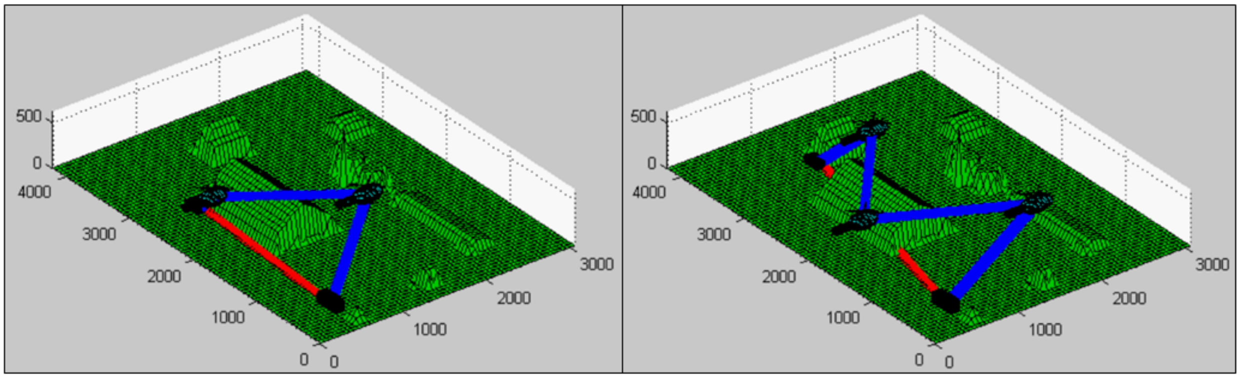


Figure 4.15: Multiple helicopter radio repeating example: (Left) Two helicopter radio repeating example, (Right) Three helicopter radio repeating example

Although this seems like an impressive benefit to the algorithm, it should be noted that these extensions are subject to the same set of weaknesses as the one aerial vehicle variant described above. Most notably, since the problem is coupled, the state space for the two helicopter case is six-dimensional and the state space for the three helicopter case is eight-dimensional. This greatly expands the number of states that need to be considered, which results in much more significant computation times, especially with a heuristic that can only guide the ground vehicle towards its goal location. Table 4.6 contains the sizes of the open and closed lists for a two helicopter and a three helicopter simulation. In these simulations, the start and goal states were virtually the same. The two helicopter simulation had the ground station at point (500,500), the ground vehicle start point at (200,2000), one of the helicopter's start point at (500,700), and the other helicopter's start point at (500,1000). The goal point for the ground vehicle was also set to (1500,4000). The three helicopter case was run with the same conditions as the two helicopter case except that the third helicopter was positioned at a starting point of (500,1200). It should be noted that since these variants were coded with the slower array based data structures, as opposed to the binary heap and hash table data structures, the computation times are not presented since they are not indicative of the actual performance of the algorithm. The sizes of the open and closed lists give a much more accurate indicator of the computational complexity of these algorithms though

Table 4.6: Multiple aerial vehicle radio repeating simulation data

Variant	Epsilon	Closed States	Open States	Optimal Path Size
Two Heli	10000	41	24913	41
Three Heli	10000	41	234881	41

and that is the reason that these are presented in Table 4.6.

This extension to multiple aerial vehicles provides the algorithm with more versatility and would allow line-of-sight conditions to be maintained in a wider variety of environments. While this extension seems applicable for the cases of two or three helicopters, if many helicopters are used or if a more swarm-based problem is attempting to be solved, the use of the techniques discussed in this thesis may be too computationally intensive for efficient implementation.

Chapter 5

Full System Testing

While the radio repeating path planner produces desirable results and simulates the radio repeating operation well, the true goal of this work was to progress another step and to actually implement this path planner on a real vehicular system. This non-trivial task of developing a fully functioning multiple vehicle system capable of performing the radio repeating operation presented a number of practical complexities that required a substantial amount of development time. For example, aside from setting up a network so that the vehicles could communicate with each other, message protocols had to be developed, the path planning code needed to be written in an executable format, conversions needed to be implemented to convert from the local path planning reference frame to a global one, a controller for the UAV needed to be developed, user interfaces had to be produced, and the architecture for the flight operation had to be formalized. While it is not the aim of this chapter to present all of the intricacies of the full system development, this chapter will provide an overview of the full radio repeating system and results from the field testing that was performed. Additional details describing the various software modules can be found in Appendix B. Lastly, it should be noted that while the discussion here focuses strictly on the radio repeating application, much of the software and infrastructure developed from this work can be reused for other path planning and control applications. It is the hope of the author that the system developed here will find use in applications in the Unmanned Systems Lab for years to come.

5.1 System Architecture

As noted in previous sections, the radio repeating mission consists of cooperatively planning paths for a ground vehicle and an unmanned helicopter so that the ground vehicle can reach some desired target location in an environment while the helicopter acts as an aerial radio repeating node that can reposition itself to improve the radio signal strength received by the ground vehicle. In theory, the helicopter component of the system is intended to be completely autonomous, while the ground vehicle can be generalized to a number of scenarios. For example, the ground vehicle could be an autonomous or tele-operated robot, a manned vehicle, or even a convoy of troops with radio equipment. With this idea in mind, the helicopter for the operation was chosen as a Yamaha RMAX unmanned helicopter while the ground vehicle was simply chosen as a truck that was driven by a member of the Unmanned Systems Lab. By allowing the ground vehicle to contain a driver, the requirements of the radio repeating system became to path plan for the helicopter and ground vehicle simultaneously, to present a recommended path for the driver of the ground vehicle to follow, and to make the helicopter autonomously move along its path as the ground vehicle progressed along its own path. This simplified the system in that autonomy only needed to be applied to one of the two vehicles, but added complexity in that since the driver of the ground vehicle has an element of choice, the path planning could no longer be done a priori and must be done during the flight operation. This way, if the driver decides to choose an alternative path during the mission, the path planning algorithm could re-plan paths for the helicopter and ground vehicle when that decision was made. With this general concept of operations, the radio repeating system could be developed.

Shown in Figure 5.1 is a general data flow diagram for the radio repeating system which describes the information that is required to be transferred between vehicles. As can be seen in the figure, it was decided to have one central computer in the ground station that handled much of the complicated processing and planning while the computers on-board each of the vehicles only had relatively simple tasks allocated to them. For example, the computer on the helicopter, a commercially available wePilot flight control system, simply received flight commands, flew based on those flight commands, and sent status information back to the ground station. Similarly, the ground vehicle was tasked primarily with receiving path information from the ground station, displaying it in a meaningful way to the ground vehicle operator, and sending status information back to the ground station. Therefore, as noted in the figure, the ground station computer serves as the central hub of data processing.

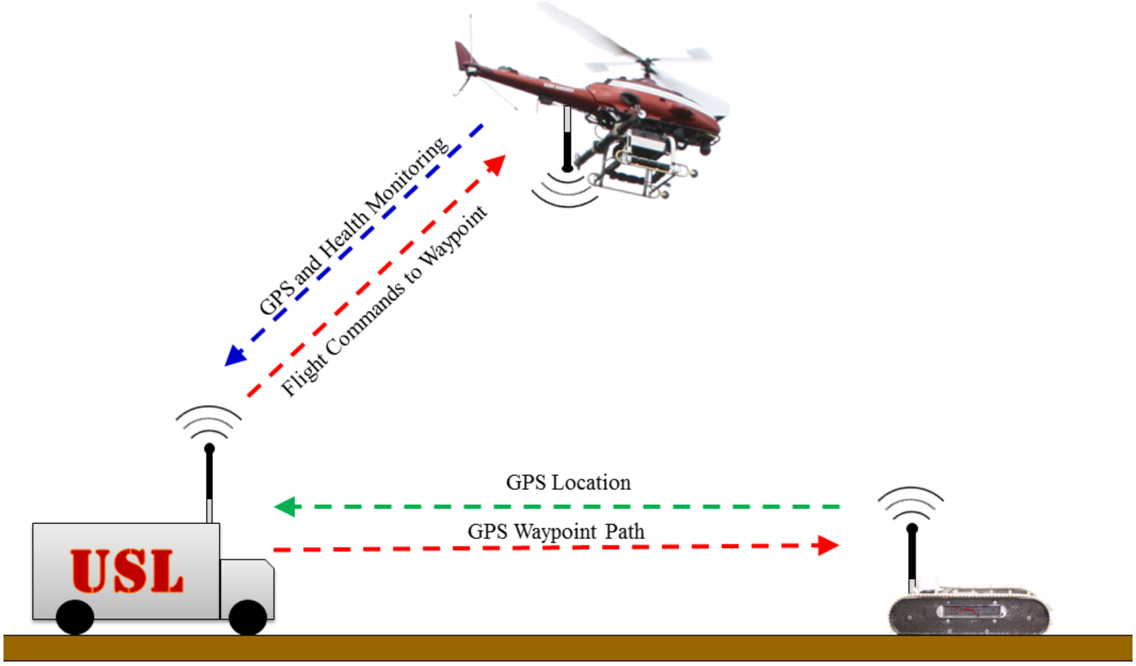


Figure 5.1: Data flow diagram for radio repeating system

Although this data flow diagram looks rather simple and straightforward, there is a good amount of information that is being passed around between the various components of the system. Figure 5.2 is a representation of the system architecture used for the radio repeating system. It clearly outlines all of the system components and the channels of data flow between them. The details of the various components of this system are given below as well. As can be noted by inspecting Figure 5.2, even though Figure 5.1 depicts data being transferred directly between the ground station and the ground vehicle, the true intention is for the data to flow through the radio repeating links between them.

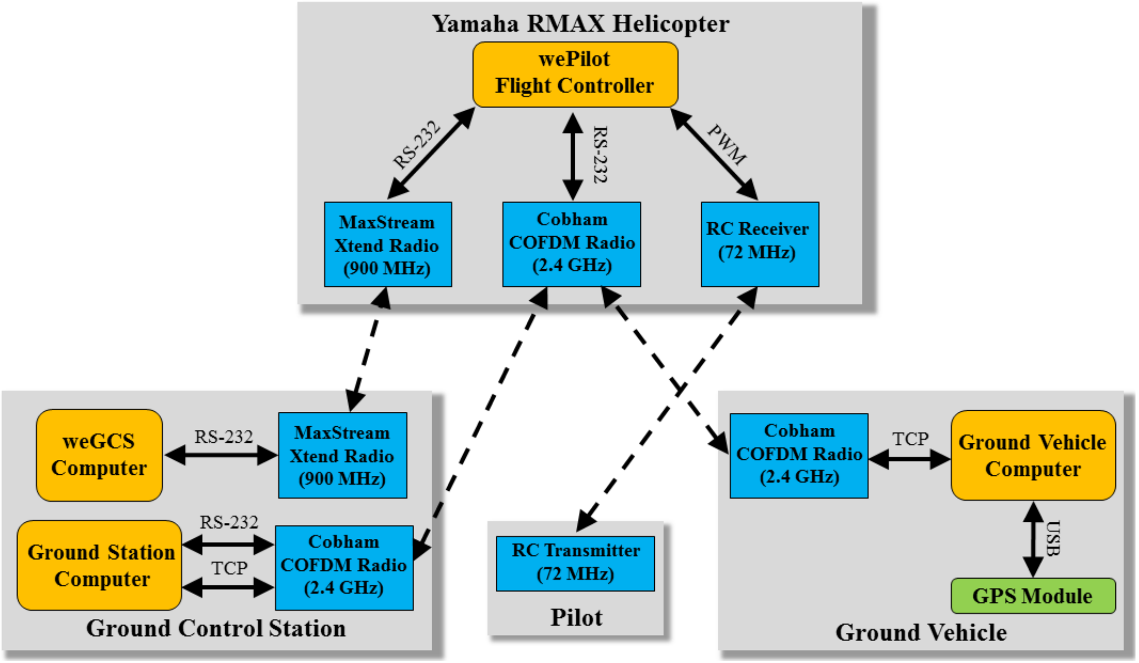


Figure 5.2: System architecture diagram for radio repeating system

Starting by inspecting the system as a whole, there are four primary components. There is the ground control station where much of the processing is performed and where the ground station operators reside. There is also the Yamaha RMAX helicopter which serves as the unmanned helicopter for the operation. This helicopter is equipped with a commercially available wePilot flight controller and a number of radios for communicating between the various system components. Next, there is the ground vehicle which is a manned vehicle that simply communicates to the ground control station and displays path information to the driver of the vehicle. Lastly, there is a pilot for the Yamaha RMAX helicopter. This pilot is responsible for starting the helicopter and raising it to a safe altitude before the wePilot flight controller is activated. The pilot serves as an essential component to any flight operation but is not critical to the radio repeating mission and will only briefly be discussed below. Starting with the ground control station, it possesses the following components:

1. **weGCS Computer-** This computer is the Ground Control Station (GCS) for the wePilot flight controller. It serves as the central means of mission planning, engaging the autopilot, and controlling the helicopter once the autopilot is activated. This is used extensively in many flight missions but for the radio repeating scenario it will only serve as a health monitoring station and a supplemental means of controlling the helicopter if any oddities occur during the flight operation. Instead of using the weGCS computer to control the helicopter, the Ground Station Computer will control the helicopter by means of an additional serial port installed on the wePilot Flight Controller that accepts flight commands at a rate of 50 Hz.

2. **MaxStream Xtend Radio-** This radio is used to allow the weGCS Computer to communicate with the wePilot Flight Controller onboard the Yamaha RMAX helicopter. It is set to operate at a frequency of 900 Megahertz (MHz) and provides an RS-232 serial interface for the communication between the computer and the flight controller. By using this interface, the weGCS can send flight commands and receive status updates from the helicopter throughout the flight. For the purposes of the radio repeating mission, there is no need to send flight commands via the weGCS but it serves as a powerful tool for helicopter safety monitoring. Also, it should be noted that the frequency was set to 900 Megahertz to be far from any of the other radios in the system which is intended to minimize the effects of interference and establish a reliable connection between the weGCS and the helicopter's flight controller.

3. **Ground Station Computer-** This computer is the primary source of processing and data transmitting and receiving for the radio repeating mission. It is a Dell Latitude D630 laptop computer and is set to run the National Instruments LabVIEW software. The LabVIEW code that the Ground Station Computer uses is discussed extensively

in Appendix B but as an overview, this computer performs all of the path planning, generates and sends flight commands to the helicopter, and sends recommended paths to the ground vehicle. In order to do this, it uses RS-232 serial communication to receive status updates and send flight commands to the 50 Hz serial port onboard the helicopter. Additionally, it uses a TCP (Transmission Control Protocol) connection to interact with the computer in the ground vehicle.

4. **Cobham COFDM Radio-** This radio uses a coded orthogonal frequency-division multiplexing scheme (COFDM) to mitigate the effects of interference in its radio transmission. It also features the capability to establish a mesh network of radios which is essential to the idea of radio repeating. The Cobham radios feature both serial and Ethernet connections which allow the transmission of multiple channels of data across the radio network. Thus, the Ground Station Computer is connected via RS-232 and Ethernet to one of the Cobham radios so that it can interact with the helicopter and the ground vehicle, respectively. Lastly, it should be noted that these radios have been set to 2.4 Gigahertz (GHz) for their frequency of transmission.

Next, looking at the Yamaha RMAX unmanned helicopter, it is equipped with the following components for the radio repeating mission:

1. **wePilot Flight Controller-** The primary component onboard the Yamaha RMAX helicopter is a wePilot Flight Controller. This is a commercially available autopilot system that stabilizes the attitude of the helicopter so that it remains in a hover state. Additionally, while it features a number of ways to interact with the helicopter, for the purposes of the radio repeating mission, the wePilot Flight Controller is used to attitude stabilize the helicopter and its 50 Hz serial port is used to specify flight commands to the helicopter. By taking advantage of this port, the radio repeating software on the Ground Station Computer can autonomously control the helicopter.

2. **MaxStream Xtend Radio-** This radio is the same as the MaxStream radio discussed

in the Ground Control Station section above except that it is the other radio in the pair. It completes the RS-232 communication link between the weGCS Computer and the wePilot Flight Controller. In a typical mission, this would be the primary means of interaction for controlling the helicopter but for the radio repeating mission, this line of communication is strictly used for health monitoring purposes.

3. **Cobham COFDM Radio**- This Cobham radio is the same as the the Cobham radio that was discussed in the Ground Control Station section. It is simply another radio in the mesh network of radios. This radio completes the RS-232 communication channel to the 50 Hz serial port on the wePilot Flight Controller and allows the Ground Station Computer to send flight commands and receive status information from the wePilot Flight Controller throughout the mission.

4. **RC Receiver**- This radio controlled (RC) receiver receives the pulse-width modulated signals from the pilot's transmitter and inputs them into the wePilot Flight Controller. This channel of communication is essential for the pilot to take off and land the helicopter but once the helicopter is engaged, this channel of communication is not used until the pilot is ready to land the helicopter. It should be noted that the frequency of this communication is 72 MHz.

Next, looking at the Ground Vehicle, it has a number of components that allow it to interact with the Ground Control Station. These components are as follows:

1. **Ground Vehicle Computer**- The primary means of processing, receiving, and sending data onboard the Ground Vehicle is by means of a Dell Latitude D630 laptop which serves as the Ground Vehicle Computer. Although the software running on this laptop is discussed extensively in Appendix B, a brief overview of its function is to pull data from a GPS module, send its current GPS location to the Ground Station Computer, receive path data from the Ground Station Computer, and display it to the operator. This computer uses a USB (Universal Serial Bus) GPS module to gather the Ground

Vehicle's position information and uses an Ethernet connection to send data to and receive data from the Ground Station Computer via TCP.

2. **Cobham COFDM Radio-** This radio is the last in the mesh network of Cobham radios and completes the TCP connection between the Ground Station Computer and the Ground Vehicle Computer.

3. **GPS Module-** This GPS module provides GPS information to the Ground Vehicle Computer through a USB connection. It was developed by another graduate student at the Unmanned Systems Lab, Ken Kroeger, and sends the following information to the Ground Vehicle Computer: Latitude, longitude, height above mean sea level, ground speed, course over ground, number of satellites used in calculating the GPS values, and whether there is a valid GPS lock or not. While these are all useful parameters for a general GPS module to output, only the latitude, longitude, and whether or not there is a GPS lock are relevant for the radio repeating mission.

By assembling all of these components together and including the pilot who is responsible for taking off and landing the helicopter with the 72 MHz transmitter, the radio repeating mission could be performed and tested. If a more in depth discussion of the Ground Control Station and Ground Vehicle software modules are desired, Appendix B provides an extensive discussion on how they operate.

5.2 Full System Test Results

With all of the system components and various software modules in place, the radio repeating mission was able to be tested on a vehicular system. This test provided two primary benefits for this work. The first was the validation that the full system was operating properly and that the radio repeating operation could be performed in a real life scenario. The second was that the data taken from this test could partially validate the simulated results produced by the path planner. Specifically, the radio signal strengths measured throughout the flight were compared with those predicted by the planner and also with those measured from a naïve radio repeating variant test. By doing this, not only could confidence be gained about the predictions produced by the planner but also that the radio repeating path planner produces better paths than a naïve counterpart from the standpoint of received signal strengths. As an additional note, it requires mentioning that the results discussed in this section provide confidence in the above claims but not true and absolute validation. This is due to the small diversity of data gathered from this test with respect to the facts that only one set of flight tests was performed and that only one environment was tested. Unfortunately, due to the large amount of time and money that is required for flight operations, coupled with the fact that regulations only allow the USL to fly unmanned aircraft at one test facility, further testing in different scenarios was prohibited. However, regardless of this fact, based on the data obtained from the flight test and the simulations performed in the previous chapter, the path planner for the radio repeating scenario produces favorable results.

Figures 5.3, 5.4, and 5.5 visually depict the setup used for the radio repeating test. Figure 5.3 displays the Yamaha RMAX unmanned helicopter which was used as the aerial component for the operation. As described in the system architecture above, this helicopter was equipped with its standard flight electronics, its wePilot autopilot system, and a Cobham COFDM radio. In addition to this, a boom was added to hold the antenna sufficiently far away from the main body of the helicopter so that reflections from the structure of the helicopter were minimized. Figure 5.4 displays the building set that the test was performed over. These buildings are located at Kentland Farms which is a Virginia Tech facility where the Unmanned Systems Lab is permitted to fly its unmanned aerial vehicles at. As can be noted from this figure, it is an open- or farm-based environment and is similar to the Type I environments described in Section 4.2.3. Lastly, Figure 5.5 displays an image taken from the test and displays the location of the fixed ground station, the helicopter in flight, and the mobile ground vehicle. For this test, the ground station (GS) is depicted as the white USL truck in the right hand side of the image and the mobile ground vehicle (GV) is depicted as the gray truck in the left portion of the image. Also, in this test, all of the vehicles were equipped with the appropriate equipment as described in section 5.1 and the computers onboard each of the vehicles were continually communicating and transferring the appropriate status information and control commands between them.

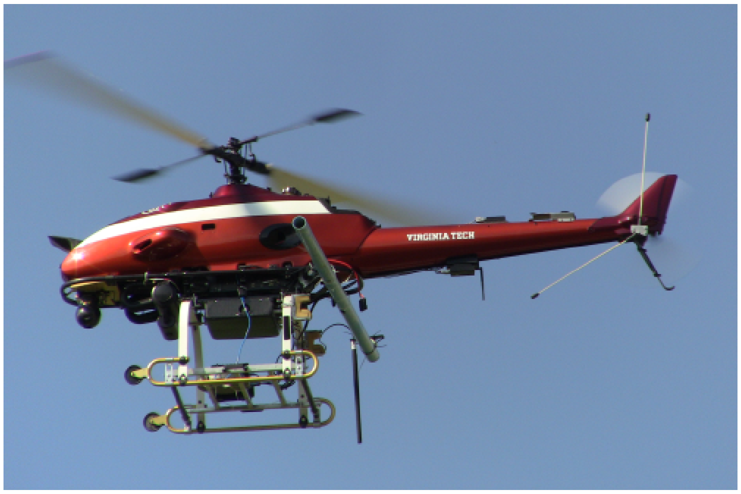


Figure 5.3: RMAX unmanned helicopter used in radio repeating test (*Image courtesy of Matt Torok, USL Graduate Student*)

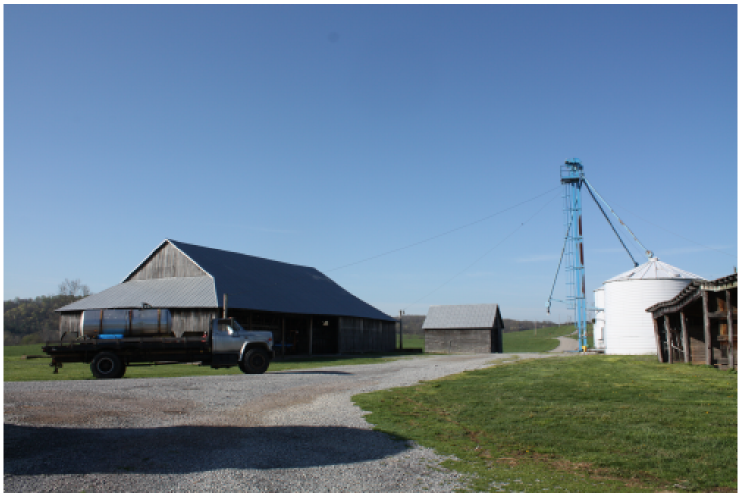


Figure 5.4: Kentland Farms building set used in radio repeating test (*Image courtesy of Matt Torok, USL Graduate Student*)

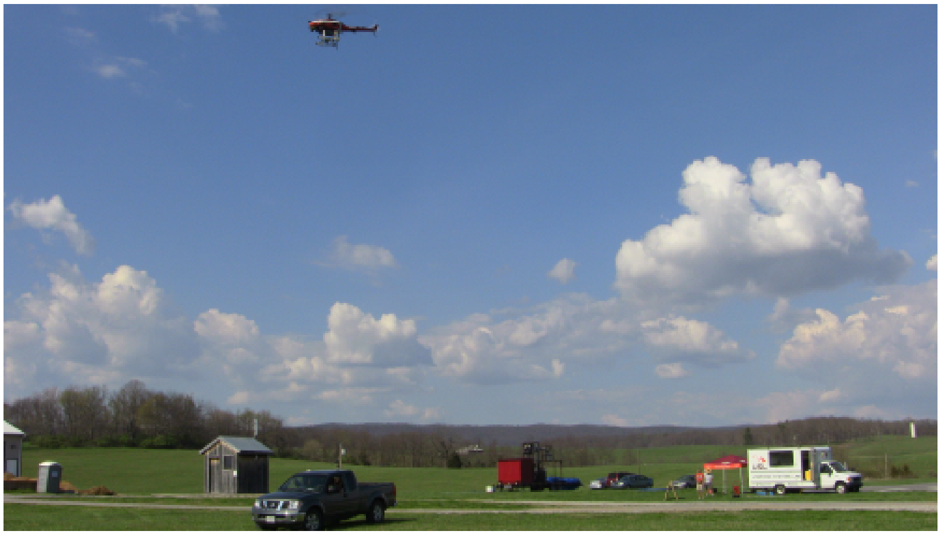


Figure 5.5: Image of radio repeating test in operation (*Image courtesy of Matt Torok, USL Graduate Student*)

With this setup, the radio repeating operation was tested in the region depicted in Figure 5.6. As can be noted by inspecting Figure 5.6, the test region was confined to be inside of the blue bounding box which was the region where a terrain map was available for the path planner. Additionally, a few sampled points from the paths generated for the ground vehicle and the helicopter are depicted in the left portion of the figure as the blue and green icons, respectively. By considering the 13 pairs of points, it can be noted that the operation was initialized with the GS, GV, and helicopter in a line-of-sight scenario and the operation then progressed to non-line-of-sight scenarios between the GS and GV towards the ends of the paths. Also, as was expected based on the simulations of section 4.2.3, the helicopter's path follows a trajectory that is indicative of maintaining healthy radio repeating links. The helicopter begins by extending itself outward in order to make the angles between the various antennas shallower so that better antenna gains could be received. Then, in a similar fashion, the helicopter continues to follow the ground vehicle attempting to similarly improve its position relative to the antenna pattern geometry. Finally, as the ground vehicle approaches non-line-of-sight behavior with the ground station, the helicopter swings away from the large central building in order to attain line-of-sight conditions throughout the remainder of the operation. This provided line-of-sight conditions for the radio repeating links throughout the operation. The right portion of 5.6 on the other hand depicts sampled points from a naïve radio repeating test where the helicopter was fixed at a single point in the test environment. By taking radio signal strength measurements at the vehicle configurations depicted in both portions of Figure 5.6, both the superiority of the planner with respect to the naïve variant and the accuracy of the predictions made by the path planner could be assessed. Lastly, it should be noted that for the sake of obtaining accurate radio signal strength measurements, for this portion of the flight operation, the vehicles were manually placed at their respective waypoints generated by the planning variants.

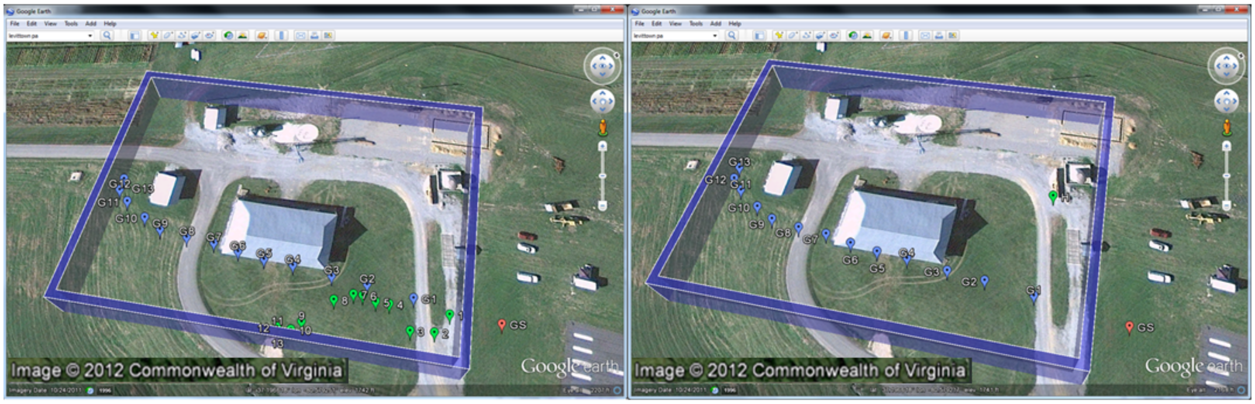


Figure 5.6: Radio repeating test sample points overlaid in Google Earth (© 2012 Google): (Left) Locations where radio signal strength measurements were taken when using the path planning variant, (Right) Locations where radio signal strength measurements were taken when using a naïve variant. The blue overlay points indicate positions of the ground vehicle, the green points indicate positions of the helicopter, the red point indicates the position of the ground station, and the blue perimeter indicates the boundary imposed on the path planner.

Figure 5.7 displays the received signal strength data captured from sampling the points from Figure 5.6 along the path planner's paths and the naïve variant's path. A number of interesting features can be noted from this plot. First, when considering the GS-to-Heli radio link, it is interesting to note that both variants produce similar results. The naïve variant obviously produced relatively constant signal strength values due to the fact that the helicopter was stationary as data was gathered and only had slight variations that are to be expected with radio data. The path planning variant started out at lower signal strength values at the beginning of the data set but saw increases in strength as the flight progressed. This is due to the fact that at the start, the helicopter was positioned relatively high above and close to the ground station which is a low gain configuration when considering the torus shape of the antenna radiation patterns. As the helicopter moved farther away from the ground station, it entered the main lobe of the antenna patterns where the gains are higher. Therefore, at this portion of the test, higher signal strengths were observed. Lastly, around data points 6 or 7, the path planning variant's helicopter location was a similar distance away from the ground station as the naïve variant's helicopter location was. Received signal strengths would be expected to be similar between the variants at these points and indeed this was the case.

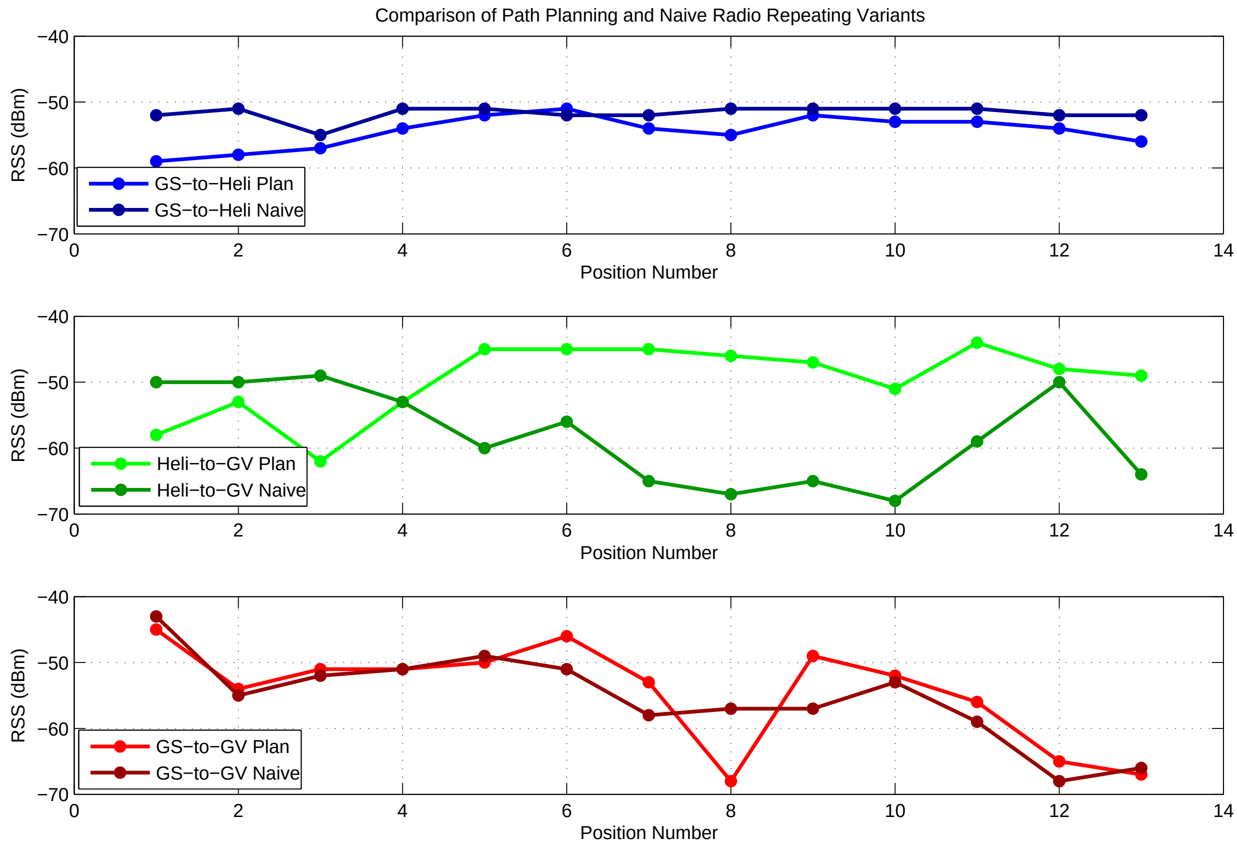


Figure 5.7: Comparison of received signal strengths for radio links in path planning and naïve radio repeating scenarios

The GS-to-Heli link does not provide much distinction between the performance of the variants. The true difference between their performances is shown in the Heli-to-GV link. As can be seen from inspecting the path planner's link first, the first few data points produced moderate signal strengths when the helicopter was almost directly over the vehicle and where the antenna gains were low. However, as the ground vehicle progressed along its path and extended farther away from the helicopter in data points 4 through 10, the signal strengths became quite high. This is due to the fact that the ground vehicle was moving to positions where the antenna patterns produced high gains and these gains outweighed the losses associated with the additional distances between the vehicles. Also, the signal strengths degraded slightly as the vehicle navigated behind the building around data points

12 and 13. However, due to the fact that a line-of-sight condition was still maintained, the signal strengths stayed relatively high. This is in stark contrast to the naïve link which started out with strong signal strengths at the first few points but had severe signal strength degradations as the ground vehicle lost line-of-sight behavior for a majority of the remaining points in the test. This was the same performance as was seen from some of the naïve radio repeating variants in the simulations and shows very well that the path planner has the capacity to produce better results than a naïve variant.

Lastly, because of the short distances and sparse obstacles considered in the test area, it is not expected that the ground link would be substantially worse than the repeating links. In fact, since the antenna patterns seemed to be one of the most influencing factors for the signal strengths in the relatively small test area, some points in the GS-to-GV data actually produced similar or better signal strengths than the repeating links. Another point to be made here is that aside from a few outliers like points 8 and 9, which could have resulted from irregularities in the terrain coupled with slight offsets in the measurement locations, the data from both the path planning and naïve variants are in agreement. This obviously should be the case since the path for the ground vehicle was the same in both instances.

Regardless of this, the fact remains that when considering the repeating links, the path planning variant produced signal strengths that were similar to or outperformed those produced by the naïve variant. Table 5.1 displays the test data in a numerical format and the difference between the Heli-to-GV links can be easily seen by inspecting the average received signal strength column. Additionally, since it was not convincing in this test that the repeating link was significantly better than the ground link due to the small environment and the sparsity of obstacles present, another test was performed where the benefit of the repeating link was easily observed. In this test, the ground station and helicopter were set at fixed locations and the ground vehicle was driven to a region where no line-of-sight existed between the ground vehicle and the ground station. Radio signal strengths for the repeating and ground links were gathered from this test and they show very convincing data that the repeating links outperform the ground link. The locations where readings were taken are shown in

Table 5.1: Received signal strength (RSS) data from radio repeating flight test

Radio Link	Average RSS (dBm)	Median RSS (dBm)	Minimum RSS (dBm)	Maximum RSS (dBm)
GS-to-Heli Plan	-54.5	-54	-59	-51
GS-to-Heli Naïve	-51.7	-51	-55	-51
Heli-to-GV Plan	-49.7	-48	-62	-44
Heli-to-GV Naïve	-58.2	-59	-68	-49
GS-to-GV Plan	-54.4	-52	-68	-45
GS-to-GV Naïve	-55.3	-55	-68	-43

Figure 5.8 and the corresponding signal strength measurements are shown in 5.9. As can be seen, the resulting signal strengths of the GS-to-GV link were dangerously low to the point where the radio link was almost lost while both of the repeating links, the GS-to-Heli and the Heli-to-GV, exhibited reasonable signal strengths. This attempts to show that the entire basic concept behind the radio repeating operation is well founded and that the addition of an aerial repeater node provides great benefit to beyond-line-of-sight ground vehicle operations.

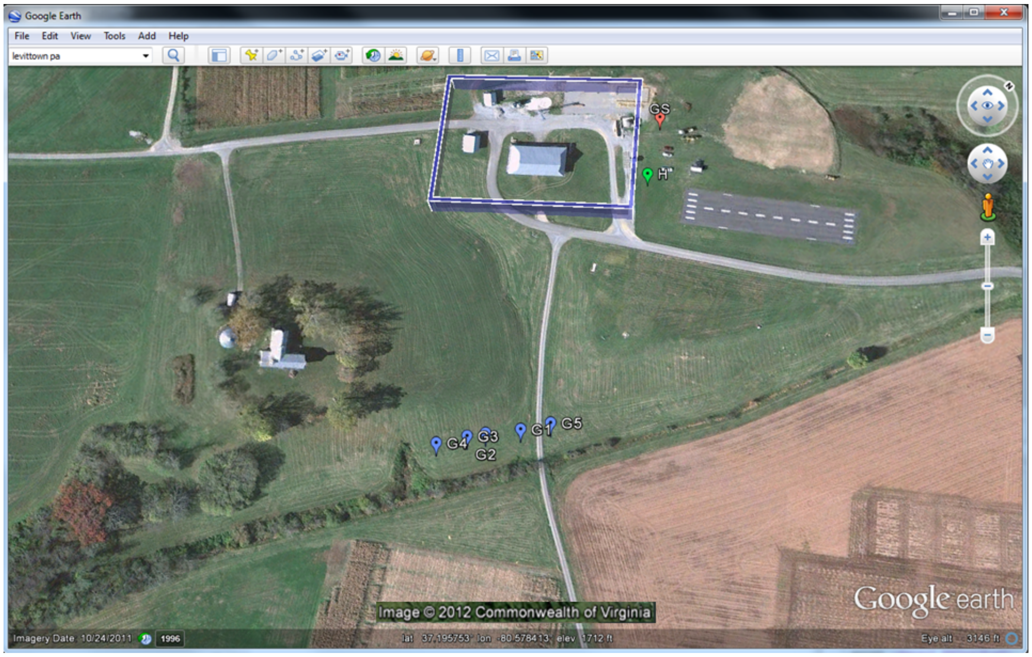


Figure 5.8: Beyond-line-of-sight radio repeating test sample points overlaid in Google Earth (© 2012 Google) - Blue points indicate ground vehicle positions, the green point indicates the helicopter position, the red point indicates the ground station location, and the blue perimeter indicates the boundary imposed on the path planner

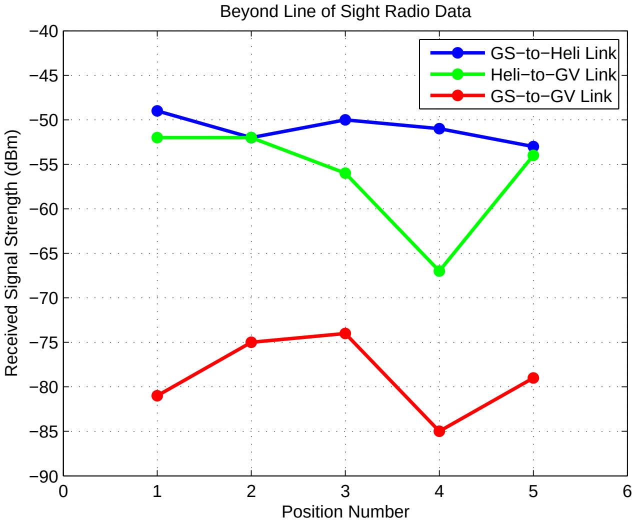


Figure 5.9: Beyond-line-of-sight radio signal strength data

While the data produced results that support the path planning variant as being superior to a naïve counterpart, the data was also inspected to determine how well the path planner predicted the actual radio performance. This comparison is shown in Figure 5.10 and a number of interesting features can be seen. First, in general, the predicted signal strengths seem to line up with the measured data moderately well. There are a few minor discrepancies but the path planner seems to predict the behavior of the radio link well on average. Considering the GS-to-Heli link first, the predictions for data points 1 to 3 are quite accurate. However, data points 4 through 7 are are generally predicted to be worse by the planner than were actually encountered. There could be many potential causes for this effect. For example, slight differences between the positions or angles of the antennas in the field test versus the simulation could have caused differences in the actual and predicted signal strengths. Also, since the helicopter did not move significantly between points 4 and 7, the simulation could have predicted that the vehicles were in low gain areas of the antennas which occur between the main and side lobes of the antenna radiation patterns. Instead, in the field test, the vehicles could have been offset enough that they were actually within the higher gain main lobes of the antenna patterns. This would obviously have resulted in larger gains and the improved signal strengths seen in the plot. For the remainder of the points though, the GS-to-Heli link was moderately well characterized by the path planner.

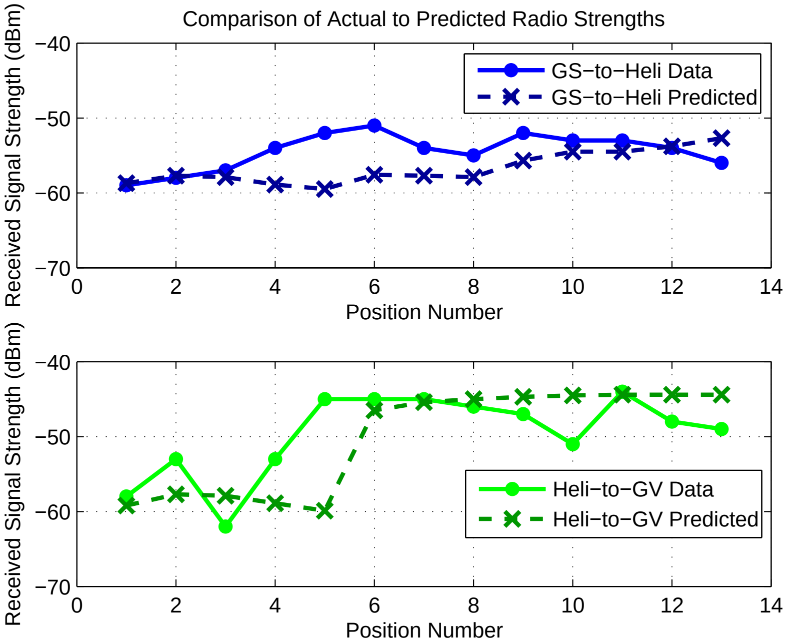


Figure 5.10: Comparison of radio repeating simulation and actual data

Considering the Heli-to-GV link, the same thematic performance is seen for both the planner and the test data. While the helicopter was positioned over the ground vehicle in the first few points (1-4), the vehicles were in low antenna gain locations which produced the initial low signal strengths. Both the simulated and actual data show this occurrence although the planner does not predict the extra oscillation in points 2 and 3 of the actual data. After this, the distance between the helicopter and the ground vehicle extended and the vehicles entered the main lobes of the antenna patterns. This caused a substantial gain increase which is also displayed by both the planner and the actual data. There is another slight oscillation in point 10 and once the ground vehicle reaches its final destination behind one of the buildings, the signal strength drops off slightly in the actual test data. This drop off is also not in the planner's predictions.

Although the planner produces signal strength estimates that follow the same trends as the data, the main differences between the planner and the test data are worth discussing and can be attributed to a few factors. The first is that it should be noted that the planner uses empirical path loss models and therefore is only able to predict radio signal strengths on average. In order to gain the computational efficiency required to perform a path planning operation, many of the intricate details of radio wave propagation were only implicitly taken into account. Therefore, some of the fluctuations which may have occurred from multipath propagation would not be explicitly seen in the simulation data. Also, the path planner tends to consider the ground vehicle to be on a perfectly flat surface. In actuality, there are some small bumps and hills in the terrain of the test area near points 8 through 10 which may have caused the vehicle and its antenna to rotate. This could specifically account for the differences in the data sets around points 9 and 10. Additionally, slight offsets in the antennas could cause the antenna null regions to manifest themselves at different locations than expected by the planner. This is likely the cause of the delay between the actual and predicted data curves around point 5. Lastly, due to the approximations made in the generation of the map used for the environment, some slight offsets between the simulated and actual environment could exist. This could cause the planner to believe there is line-of-sight at some locations when in reality there is not. This is likely the reason why the drop off at points 13 and 14 were not predicted. Regardless though, many of these factors are minor and could be abated by either slightly better modeling practices in the antenna locations or in the map of the environment. All in all, given its simplicity, the path planner predicted the performance of the radio links well in this test and this provides confidence that the radio repeating path planner will aid in planning for beyond-line-of-sight ground vehicle operations.

Finally, after the data points were taken, the operation was performed with the helicopter in a fully autonomous mode and with all of the software operating properly. Figure 5.11 displays the flight path taken by the helicopter throughout the operation with points from the corresponding planned paths for the ground vehicle and the helicopter overlaid as blue and green icons, respectively. These points were based on the initial conditions of the given radio repeating operation and exhibit the same thematic behavior as observed in many of the simulations and in the paths described above. By inspecting the segment of the helicopter's

GPS log within the path planner's boundary, which is represented by the green line, the helicopter does indeed autonomously follow along the path determined for it. There are a few overshoots and deviances seen in the flight path but this can easily be attributed to a number of practical factors. First, the GPS positions of the helicopter and ground vehicle have uncertainty associated with them. This is due to the fact that GPS data is usually only accurate to within a few meters which is especially true in the case of the inexpensive GPS module used for the ground vehicle. Secondly, due to this inaccuracy, a dead zone of a 3 meter radius was included in the controller for the helicopter to stabilize its trajectory and this is likely responsible for some of the undershoots in the helicopter's flight path especially around points 11 and 12. Regardless of these slight practical issues, the fact that the system performs as expected and navigates the helicopter along its path as the ground vehicle correspondingly progresses along its own path was an enormous success. Additionally, given the fact that the radio signal strength data is supportive of the results produced by the path planner and the fact that the path planner outperformed a naïve variant in field testing, confidence is gained that not only would the path planner produce favorable results but that the radio repeating operation would be beneficial to enhancing communication links in beyond-line-of-sight vehicle operations.

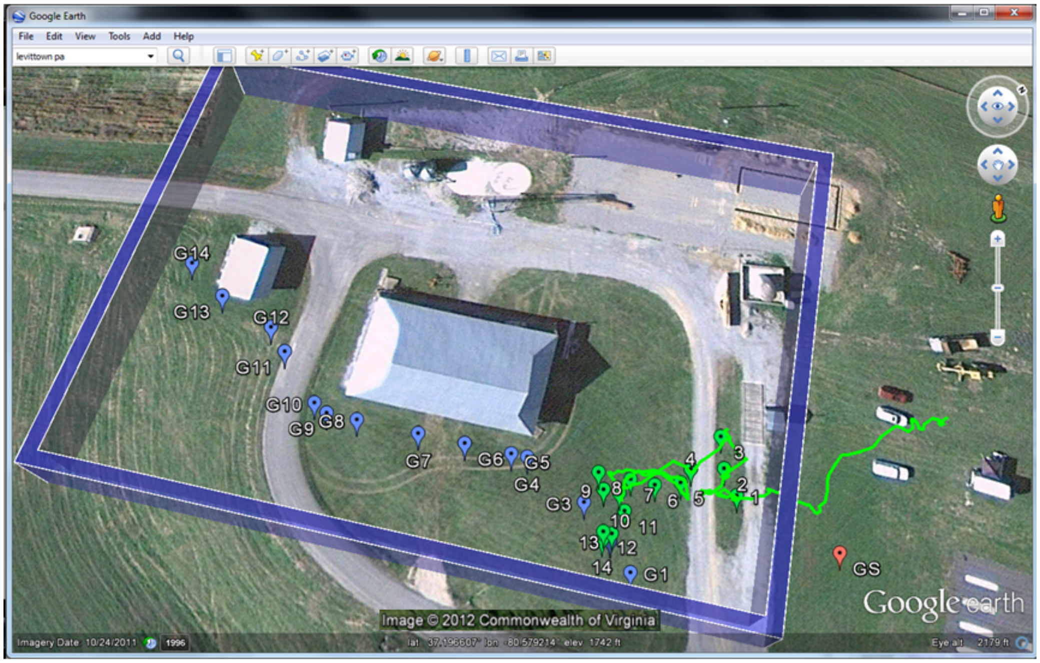


Figure 5.11: Autonomous helicopter flight path for radio repeating test overlaid in Google Earth (© 2012 Google) - Blue points indicate the planned ground vehicle positions, Green points indicate the planned helicopter positions, the red point indicates the ground station position, the blue perimeter indicates the boundary imposed on the path planner, and the green curve indicates the path taken by the helicopter

Chapter 6

Roadway Detection

While the radio repeating A* implementation described in the above chapters is a general algorithm that can be applied to a variety of environments, one limiting constraint persists: In order to effectively evaluate the terms in the cost function and tend the search to potential line-of-sight behavior, a well known map needs to exist of the environment where the search is being performed. While this may seem like a very strict constraint for some, the Unmanned Systems Lab performs research in the area of stereovision for terrain map reconstruction and recent work from other researchers at the lab show promising results [26]. Therefore, it would not be unreasonable to assume that a stereovision flight operation was performed to generate a terrain map before embarking on a radio repeating operation and because of this, this constraint theoretically does not limit the applicability of the radio repeating work above for the Unmanned Systems Lab.

However, in the case where a stereovision system like the one used by the Unmanned Systems Lab is unavailable, the question naturally arises of how to generate such a terrain map and if the A* techniques of this thesis would be able to facilitate such a terrain map generation. This question is indeed grandiose and could form a very significant area of research that is beyond the scope of this work. Regardless, this chapter highlights some initial work towards answering that question and hopes to set the stage for future developments in this area.

Essentially, without the use of stereovision to gather depth perception about a specific area of interest, it would be difficult to construct a three dimensional terrain model of an environment. However, since the A* technique is applicable to grid-based searches, it is very applicable to image processing where an image is essentially represented as a grid of pixels. By making use of an aerial vehicle with a single camera, which is not an unreasonable setup, the A* technique can be used to identify different features of the images taken by the aerial vehicle. By tagging these images with GPS locations, these features could be used to populate a two-dimensional map of the environment. The work of this chapter seeks to use this concept in the area of roadway detection. That is, the A* techniques, along with other image processing methods, are used to identify roadways in images which may be potential regions for a ground vehicle to traverse. This is obviously not as detailed as a full three-dimensional map, but is quickly executable and can provide rough information about the environment being traversed.

This chapter is split into two main sections. The first discusses the roadway detection algorithm that was developed in this work and how it relates to the A* techniques described above. An outline of the algorithm's process is given as well as some discussion on the various assumptions that the algorithm requires. The second section then evaluates this technique on a variety of aerial imagery and discusses its strengths and some potential areas for improvement.

6.1 Algorithm for Roadway Detection

As noted above, images can essentially be represented as a grid of pixels where each grid cell contains values representing the color content of a particular pixel. In a color image, there is typically a 0 to 255 value for each of the red, green, and blue content of a pixel, while for a black and white image there is only one value used to describe the white content of a pixel. Since the A* algorithm is a search algorithm that operates on a discrete grid environment, it is quite suitable for performing processing on images. More so, by referring to the A* Path Planning chapter above, by eliminating the heuristic, h, for the search, the A* technique results in a breadth first search from a start location to a goal location. In the case of roadway detection, this breadth first ability is quite useful since as long as the cost function is formulated appropriately, the search can act as a fill operation on a roadway segment in the image.

The only remaining factors in implementing this technique are to determine a suitable cost function, goal point, and starting point for the search. Beginning with the cost function, it is relatively straight forward for a filling operation. The only defining features of a filling operation are that it is radially expansive and only fills up to a specific boundary. The radially expansive feature is easily obtained by simply eliminating the heuristic from the cost function and evaluating states based on their distance from the starting point. Therefore, this filling implementation is no more complicated than the simple implementations in Chapter 3. However, in order to accomplish the second task of staying within a boundary, an additional cost term is included which penalizes the search for going outside of the roadway region. While there are potentially many ways to perform this task, this implementation uses color to define the roadway and penalizes pixels that are too dissimilar from the starting point, which is assumed to be within a roadway. This penalty is a binary one where zero penalty is assessed if the point in question is similar in color to the starting point and essentially infinite if the point in question is not similar in color to the starting point. An intuitive way to implement this concept is to consider a sphere of colors in a red-green-blue space and if the color of a point is outside of the sphere, a penalty is assessed. This concept is shown visually in Figure 6.1. The overarching assumption in this implementation is that a roadway is of a relatively uniform color and that it is surrounded by a background that has a distinctly different color. For many cases where roads are paved with asphalt and are bordered by curbs or grass, this is usually a valid assumption and will be adopted here. This does however introduce complications for images that do not fit this mold as will be discussed in the following section. Lastly, as a matter of practicality, another infinite penalty is assessed for the search traveling outside of the image. This is in a similar vein as in the radio repeating path planner. By considering these factors, the cost function for the fill operation can be written as follows.

$$f = g_{Dist} + g_{Color} + g_{OoBC}$$

$$g_{Dist} = \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{x}_{i-1}\|_2$$

$$g_{Color} = \begin{cases} 0 \ , \ if \ \|\mathbf{x}_{Color,n}\|_2 \leq \textsc{MaxColorNorm} \\ \infty \ , \ if \ \|\mathbf{x}_{Color,n}\|_2 > \textsc{MaxColorNorm} \end{cases}$$

$$g_{OoBC} = \begin{cases} 0 \ , \ if \ \textsc{OutOfBounds}(\mathbf{x}_n) = \textsc{false} \\ \infty \ , \ if \ \textsc{OutOfBounds}(\mathbf{x}_n) = \textsc{true} \end{cases}$$

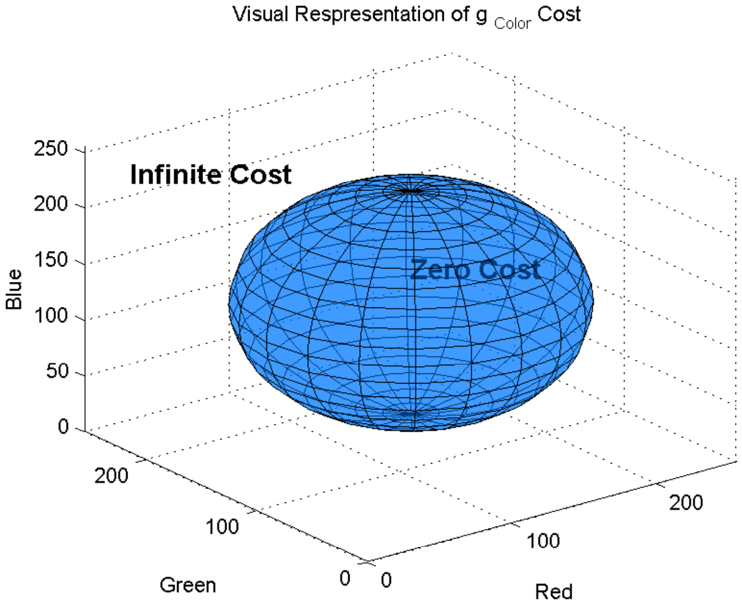


Figure 6.1: Visual respesentation of color norm cost term

With the cost function suitably defined for the desired performance of the path planner, a goal position now needs to be set. For a fill operation though, the concept of an end point is slightly ambiguous so a goal point cannot be defined in the typical manner. Instead of defining a specific goal point, the algorithm simply continues to evaluate pixels until all pixels of a non-infinite cost have been discovered. Due to the optimality properties of the A* algorithm and the way the cost function was defined, this will produce a $CLOSED$ list that contains all points in a local region that are of a similar color to the starting roadway point and are within the bounds of the image. For a roadway that is bordered well or has a contrasting enough background, this resulting $CLOSED$ list can either contain a subset of the points in the roadway or, if fortunate enough, can contain all pixel locations in the roadway. This resulting $CLOSED$ list is used to define the detected roadway points.

All of the above roadway detection performance is predicated on the assumption that the start point for the search is in fact within the roadway. This is so that the local region expands from within the roadway and evaluates points based on the roadway color. Unfortunately, this task of selecting a start point within the roadway region is the most complex task of the roadway detection algorithm and as might be expected due to the general complexity of images, a number of filtering operations must be performed on the image to accomplish this task. All of the operations for accomplishing this and for implementing the fill operation are encapsulated within the following two algorithms which will now be discussed. These are the edge detection algorithm (Algorithm 3) and the roadway fill algorithm (Algorithm 4).

Algorithm 3 Roadway Detection Algorithm - Edge Detection

```
$ImageColor \leftarrow$ Import Image
$ImageColor \leftarrow$ ScaleImage($ImageColor, ScaleFactor$)
$ImageBW \leftarrow$ ConvertToGrayscale($ImageColor$)
Contrast stretch $ImageBW$
Set $CannyLow$ and $CannyHigh$ thresholds
$EdgeImage \leftarrow$ CannyEdge($ImageBW, CannyLow, CannyHigh$)
while PercEdges($EdgeImage$) $> EdgeThreshold$ do
    Increment $CannyHigh$
    $EdgeImage \leftarrow$ CannyEdge($ImageBW, CannyLow, CannyHigh$)
end while
```

In order to properly place the start point for the A* fill process within a roadway, the most likely location of a roadway in an image must be determined. Also, in order for this location to be determined, an image containing a roadway must be filtered to remove various non-roadway features. For this feature removal process to be effective though, a roadway must be properly defined. Therefore, this algorithm uses the following defining characteristics for a roadway.

- A typical roadway has defined edges that produce strong intensity gradients along these edges.
- A typical roadway has long and continuous edges. More so, if the roadway is a dominant feature in the image, these edges will be among the longest edges in the image.
- The edges of a typical roadway will have gradients that are opposite in direction and these edges are spaced at a distance which can be treated as a known quantity [12].

Algorithms 3 and 4 operate by sequentially filtering the original image based on these defining characteristics of a roadway. Considering the first condition, when roadways are surrounded by curbs or a well-differentiated background, the roadway will have well-defined edges. This is a powerful feature to use for roadway detection since it can differentiate it from many of the other features in the image. In order to take advantage of this feature, after scaling an image to simplify the computational complexity of the required tasks, Algorithm 3 starts by converting the image to grayscale so that each pixel only contains a single 0 to 255 numerical value defining its intensity. The changes in intensity between pixels will be used to determine potential edges in an image. Before the edge detection is done formally though, the image is also contrast stretched to accentuate the intensity changes in the image.

After this, a typical Canny Edge Detector [9] is used to extract edges from the image. This detector works by blurring the image slightly to eliminate high frequency noise in the image, calculating the intensity gradients at every pixel, suppressing gradients that are not of the highest magnitude in their local regions, and applying a hysteresis thresholding technique. This technique requires a high threshold, denoted as $CannyHigh$ and a low threshold, denoted as $CannyLow$. Essentially, the technique traces along the edges with gradient values above *CannyHigh* and continues until no more gradients are found with gradient values above *CannyLow*. By using imposing this hysteresis, the algorithm connects edges that may have been a single edge but were not picked up by the simple pixel-wise edge detection. As may be expected, different images contain different features that produce different gradient values. Therefore, in order to adapt this Canny Edge Detector to a variety of images, Algorithm 3 implements a scheme where the edge detection is repeated with incremented *CannyHigh* values until only a certain percentage of the image is left as edge content. The inspiration for this is that if the roadway is a dominant feature in the image, it will produce some of the strongest gradients, whereas other edges of lower gradient values may simply be noise or less prevalent features in the image. An example of the output of Algorithm 3, or equivalently the Canny Edge Detector, is shown in Figure 6.2. To contrast this, an original color image is shown in Figure 6.3 which was taken by the Unmanned System Lab's stereovision system.

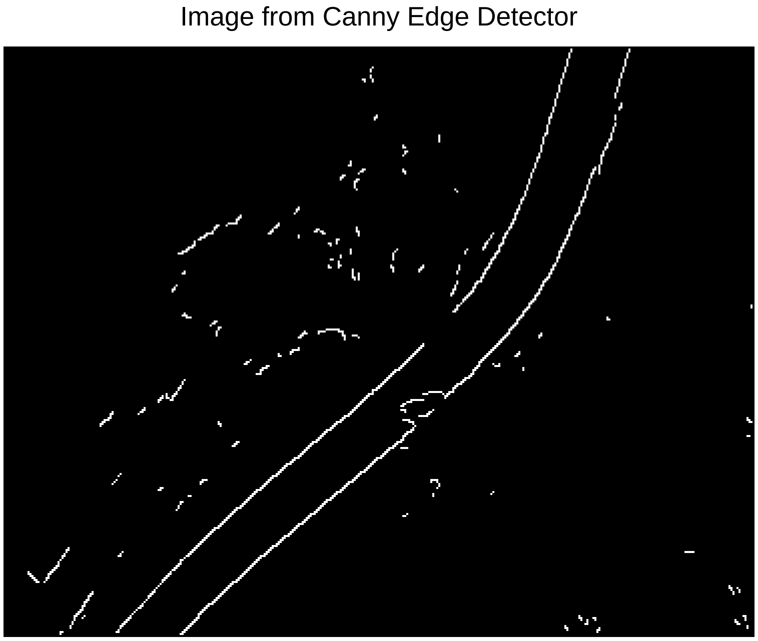


Figure 6.2: Example output from the Canny edge detector

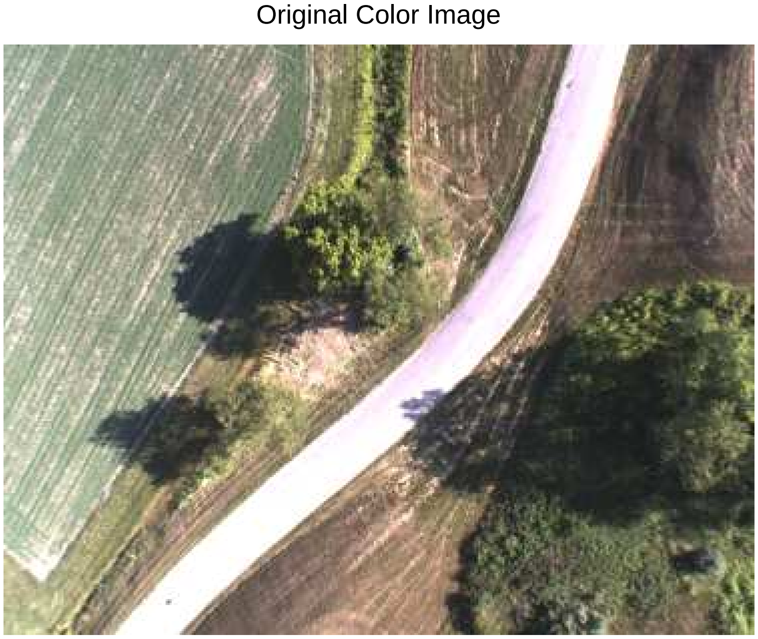


Figure 6.3: Example of original color image used for roadway detection

Algorithm 4 Roadway Detection Algorithm - Roadway Fill

```
$EdgeImage \leftarrow$ DisregardShortEdges($EdgeImage, LengthThreshold$)
$EdgeImage \leftarrow$ FillEdgeGaps($EdgeImage, FillThreshold$)
$EdgeImage \leftarrow$ EvaluateOppGrad($EdgeImage, MinRoadWidth, MaxRoadWidth$)
$SeedPoint \leftarrow$ ChooseSeedPoint($EdgeImage$)
$RoadwayImage \leftarrow$ AStarFillRoadway($ImageColor, SeedPoint, MaxColorNorm$)
```

Continuing with this edge detected image as an input to Algorithm 4, a number of other operations are performed to get rid of extraneous edges in the image and to extract roadway edges. Considering the second roadway feature, since roads are typically straight and continuous, they will be of the longest edges in the edge detected image. This is especially so if the roadway is a dominant feature in the image. Therefore, extraneous edges are filtered out by disregarding edges that are below a certain length threshold *LengthThreshold*.

After this operation is performed, the third defining feature of a roadway is employed. That is, edges in the image are tested to determine whether others exist in their vicinity with gradients in the opposite direction [12]. This is a powerful technique for disregarding extraneous edges that persist after considering the previous two characteristics of a roadway. Considering the example of a roadway that is black asphalt and has green grass on either side of it, when converted to black and white, one edge of the roadway will have a gradient that transfers from the intensity value of the grass to that of the road. Conversely, the other roadway edge will have a gradient that transfers from an intensity value of the road to the grass. This results in opposite gradients at the roadway edges. Coupling this with the assumption that the width of the roadway is known to a certain tolerance, which is not an unrealistic assumption if the altitude of the UAV taking the images is known, edges can be tested to determine how many points meet this opposite gradient criterion. Additionally, to maximize the amount of gradients found in a roadway edge, an edge filling operation is also performed to ensure that noise or small blockages of the roadway segments do not parse the roadway into too small of segments.

With these steps performed, the last remaining step prior to applying the filling path planner is to determine the most appropriate location for the start point for the planner to be located. Again, following the assumption that the roadway is a dominant feature in the image, the most likely candidates for roadway edges are those that rank as the two with the highest amount of successfully found opposite gradient points. Therefore, these two edges are chosen as the roadway edges and the start point is chosen as a random point between these segments. It should also be noted at this point that by choosing only the two edges with the highest amount of opposite gradient matches, there is the implicit assumption that this algorithm only works well when there is one primary roadway in the image. That is not to say that the roadway cannot curve around and such, but that the algorithm will not work well with two or more distinct roadways in the image. With the start point appropriately selected, the color of a local region around that point is treated as the roadway color, denoted $ImageColor$, and the A*-based fill operation is performed. To visually show the results of this process, Figure 6.4 shows the edge detected image after short edge segments are removed, Figure 6.5 shows the two edges that were selected as roadway edges after the opposite gradient criterion was applied, and Figure 6.6 shows the detected roadway region after the fill operation was applied.

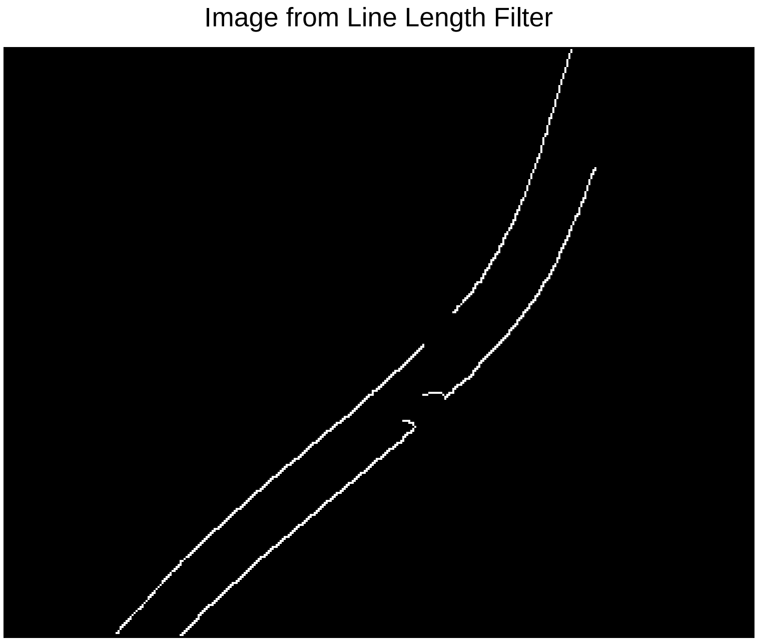


Figure 6.4: Example of output from DisregardShortEdges filter

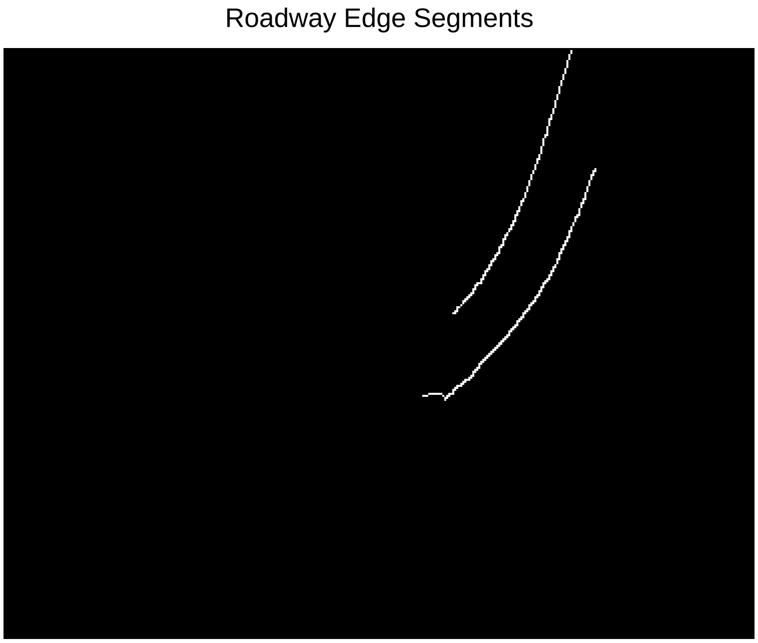
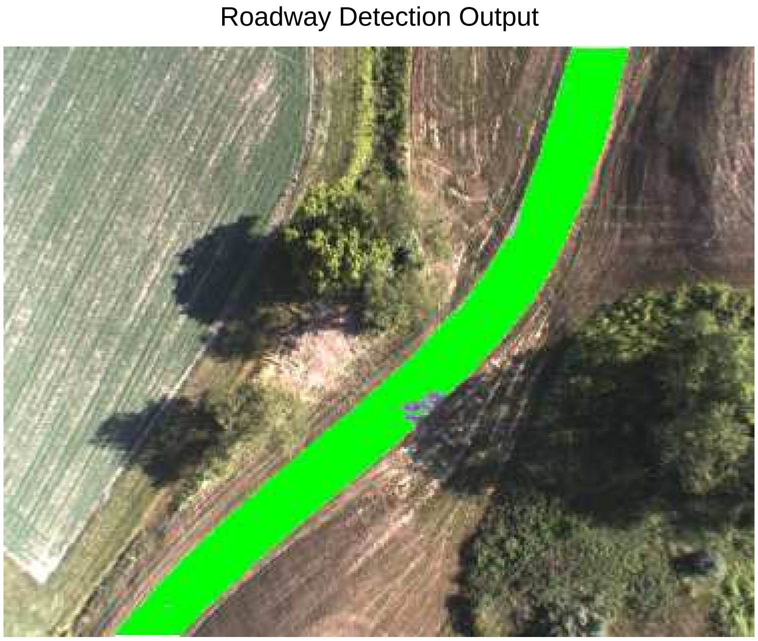


Figure 6.5: Example of determined roadway edge segments



Roadway Detection Output

Figure 6.6: Example of output from the roadway detection process

6.2 Roadway Detection Testing Results

In a similar fashion as was done with the radio repeating path planner, the roadway detection algorithm was tested on a variety of different images in order to assess its performance, strengths, and weaknesses. More specifically, the set of imagery used to test the algorithm was broken up into five different categories or types of images and numerous trials were run on images of each category. These categories were determined based on the features of the images and ranged from simple situations where the roadway was a definite dominant feature in the image to complex ones where the roadway was hardly distinguishable from its surroundings. These categories are arbitrarily numbered and are described in more detail immediately below. Also Figure 6.7 displays representative examples of images from each category and it should be noted that all images used in testing were obtained from the Unmanned Systems Lab's stereovision system. By testing the algorithm on a wide range of imagery types, the performance of the roadway detection algorithm could be accurately assessed in a variety of different scenarios.

Type I This type of image contains a roadway that is surrounded by a very well-contrasting grass background and may have a few tree shadows partially covering the roadway itself. This is representative of a category of images where the roadway may not have completely defined edges due to the tree shadows but is definitely distinct from its background. Due to the very distinct and feature lacking background, these images are of the simplest to detect roadways in.

Type II This type of image contains a roadway with very well defined edges but does not have a well-contrasting background. This is the case where a dirt or gravel road is surrounded by a dirt background that is not too distinct from the roadway itself. However, due to the presence of the well-defined edges, these images are also considered to be simple for the roadway detection algorithm.

Type III This type of image is similar to Type II where the background is of a similar color to the roadway. However, in this image category, the edges of the road are not completely clearly defined. Due to the fact that the fill operation does not have highly contrasting color boundaries to confine it, this type of image requires the $MaxColorNorm$ quantity to be set low so that the fill does not escape the roadway. This tends to cause complexities for the algorithm and only allows it to fill in the roadway partially.

Type IV This type of image also contains tree shadows that cover the entire roadway and also has edges that are not very well defined either. The former complexity inhibits the fill operation from expanding through the whole roadway and the latter causes the edge detector to potentially miss the roadway edge. These are among the most challenging images for the image detector where the roadway is a main feature in the image.

Type V This type of image represents the most challenging and complex of the five categories. This category contains images where not only is the roadway not well defined but it also contains a number of other features like buildings that they themselves might produce strong gradients in the image. Images of this type have been found to be beyond the capabilities of the roadway detection algorithm in its current state mainly due to the fact that the roadway is no longer a very dominant feature in the images and the characteristics of a roadway defined above are not sufficient to differentiate it from the other features of these images.

Table 6.1 provides results of testing the roadway detection algorithm on the five types of images described above. Here, at least five images of each type were gathered from images taken from the USL's stereovision system and each image was tested 10 times to determine the performance of the algorithm. This resulted in at least 50 trials of the algorithm for each image type and 120 trials for the Type I images where more aerial imagery was available. The average results of this testing are displayed in Table 6.1 along with the settings used for each image category. Additionally, Figure 6.8 shows representative results for the roadway detection algorithm on the various image types. As is to be expected from the descriptions above, the algorithm is largely successful at filling the roadways of the Type I and II images, partially successful in filling the roadways for the Type III and IV images, and unsuccessful at filling in the roadways of the Type V images.

By inspecting Table 6.1 more thoroughly, it can be seen that throughout the course of the testing, many of the parameters remained the same for each of the image sets. That is, many of the edge detection parameters stayed constant across many or all of the image types. The only values that required tuning for a specific image type were those that defined the expected width of the roadway and the one responsible for the amount of tolerable color deviation from the starting point in the fill operation. When considering the implications of these parameters, the fact that they require changing seems very reasonable in general. In addition, given more development time, a more sophisticated roadway detector may easily

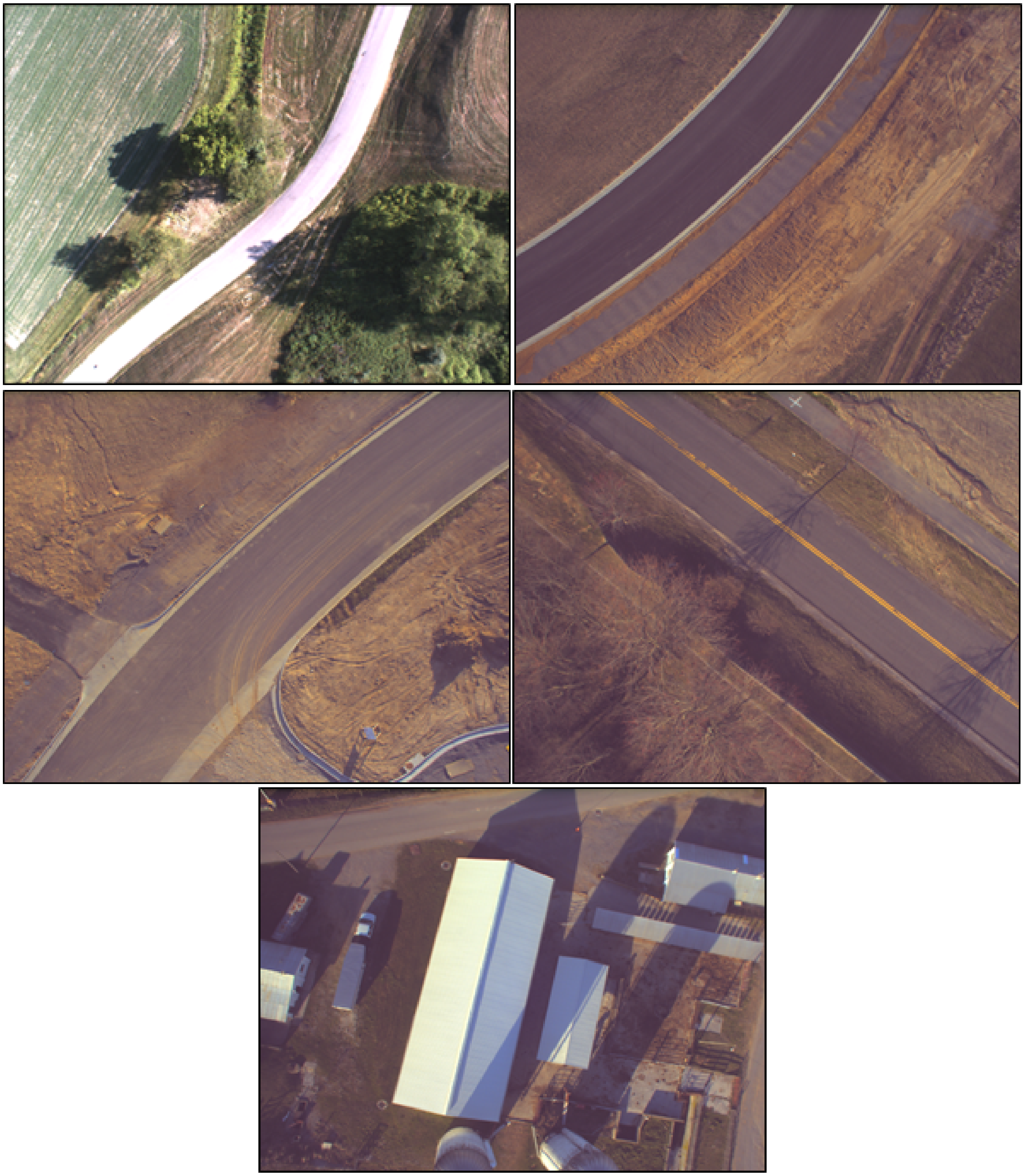


Figure 6.7: Roadway detection image types for testing: (Top Left) Type I, (Top Right) Type II, (Middle Left) Type III, (Middle Right) Type IV, (Bottom Middle) Type V

Table 6.1: Average roadway detection algorithm test results for various image types

Image Type	Type I	Type II	Type III	Type IV	Type V
Features					
Well-Defined Edges?	N	Y	N	N	N
Contrasted Background?	Y	N	N	N	N
Roadway Main Feature?	Y	Y	Y	Y	N
Roadway Not Blocked by Shadows?	Y	Y	Y	N	N
Image Size (Rows)	1200	2064	2064	2064	2064
Image Size (Columns)	1600	2532	2532	2532	2532
Settings					
Scale Factor	4	4	4	4	4
Canny Low Threshold	100	100	100	100	100
Edge Threshold	4	4	4	4	4
Length Threshold	100	100	100	100	100
Fill Threshold	10	10	10	10	1
Minimum Road Width	30	80	80	30	30
Maximum Road Width	70	150	150	110	110
Maximum Color Norm	100	50	15	15	15
Results					
Number of Images Tested	12	5	5	5	5
Trials per Image	10	10	10	10	10
Roadway Detected? (%)	100	100	100	60	0
Roadway Filled (Avg %)	97	100	64	17	N/A
Wrongly Filled Pixels (Avg %)	0	0	0	11	N/A
Average Computation Time	2.009	5.940	5.189	4.540	N/A

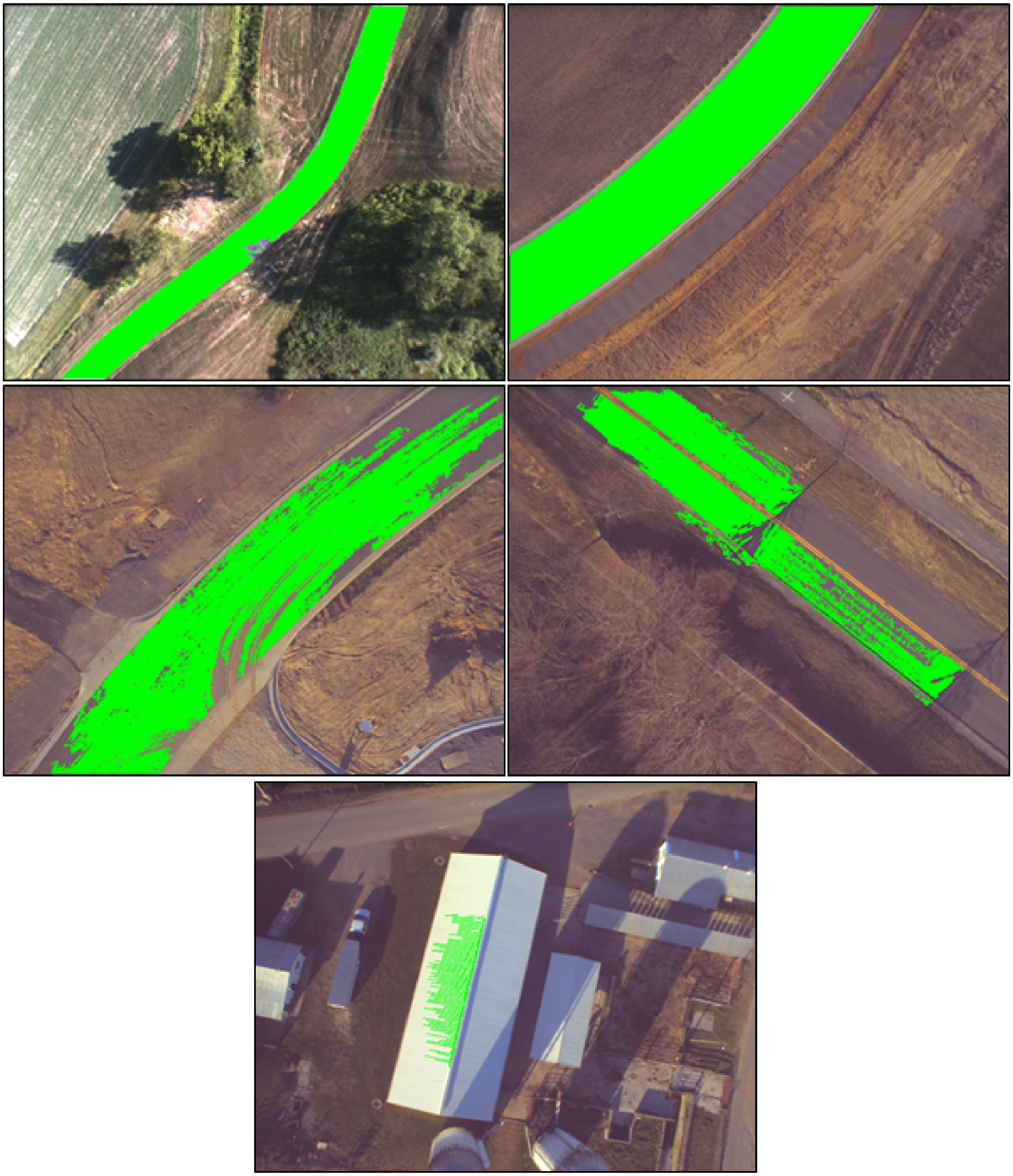


Figure 6.8: Examples of results from roadway detection algorithm on different types of images: (Top Left) Type I, (Top Right) Type II, (Middle Left) Type III, (Middle Right) Type IV, (Bottom Middle) Type V

be crafted where these parameter values are set automatically. For instance, since typical roadways are likely of pretty standard widths, the road widths may be able to be determined from the altitude of the aircraft and by comparing the altitude with a calibration curve relating road widths to pixels. Also, the $MaxColorNorm$ term could be automated by potentially sampling random points around the image and determining the variation of color between roadway points and background points.

Finally, when considering the results, it can be seen that for simple imagery like the Type I and Type II categories, the roadways were detected in 100 percent of the sample images and they were filled almost completely. The only discrepancy in the Type I results was due to the presence of tree shadows that limited the fill operation. Also, in the Type III images where the background was similar in color to the roadway, the roadways were still found 100 percent of the time, but only 64 percent of the roadway area was filled on average. This is due to the fact that the $MaxColorNorm$ parameter needed to be brought down so the fill operation did not bleed outside of the roadway. Next, in the Type IV images, roadways no longer had clearly defined edges nor did they have very contrasting backgrounds, which coupled with the presence of distinct tree shadows, severely limited the performance of the roadway detector. In this case the roadways were only found a fraction of the time and even when they were, only a small fraction of the roadway was filled. Lastly, in the Type V images, in addition to the complexities in the Type IV images, the roadway was no longer the main feature in the images and the detector was unable to differentiate the roadway from the other features of the images. Regardless, for sample images that contained roadways that were well-defined, the image detection algorithm has produced successful results.

Chapter 7

Conclusion

This thesis presents the research, development, and implementation of a system capable of performing a radio repeating operation with the specific goal of providing enhanced communication links for unmanned vehicle operations. More specifically, this thesis presents work that adds an aerial component to beyond-line-of-sight unmanned ground vehicle operations so that radio links between the vehicles and a ground station can be made more reliable and so that the communication range of the UGV can be extended to farther and more complicated scenarios. In order to attain this goal, this thesis poses the two vehicle radio repeating navigation problem as a discrete path planning problem that, when solved, provides desirable paths for both the aerial and ground vehicles in the setup. More so, in addition to performing the various path planning, modeling, and simulation work associated with this development, this thesis also presents the system architecture and software development that culminated in allowing this radio repeating operation to be performed on an actual vehicular system. Lastly, as another application of the A* algorithm and to provide an initial development at abating the operation's requirement of a stereovision 3D terrain map, an algorithm for detecting roadways in images is also discussed. With these various components, this thesis contains a full set of developments necessary for the radio repeating operation. This chapter is intended to concisely summarize the contributions of this thesis and to offer recommendations for future work that can be performed to enhance this radio repeating operation further. Because of this, this chapter is split into two primary sections. The first discusses the major developments throughout this thesis and the second offers a discussion on suggestions for potential future work that can be performed.

7.1 Summary of Contributions

With the motivation of allowing first responders to investigate post-disaster environments in a safe and effective manner coupled with the growing interest in deploying unmanned vehicles in complex environments, the development of a communication-aware unmanned vehicular system is of paramount importance. This thesis presents the development of one such system which employs a UGV and an unmanned helicopter in a radio repeating operation. A variety of the most notable contributions to this development are discussed briefly here.

First, in order to effectively generate trajectories for the ground and aerial vehicles so that a ground vehicle operation can be performed while healthy communication links are maintained, this radio repeating operation was formulated as a path planning problem. More so, since the received signal strengths of the radio repeating links in the operation are very dependent upon both of the positions of the vehicles simultaneously, the problem required formulation with a coupled state space where the positions of both vehicles were considered. Due to this complexity, Chapter 3 presents an agglomeration of path planning developments that improve the computational efficiency of the A* algorithm so that the planning for this coupled planning problem could be performed in a rapid manner. Additionally, while the use of the WA* variant tends to sacrifice the strict optimality of the A* routine, the systematic nature of this heuristic-based search was still maintained which is of pivotal importance for a scenario like radio repeating where maintaining strong signal strengths throughout the search process is critical. While the developments presented in Chapter 3 are well-studied in the literature, the compilation of these elements for the particular application of radio repeating was non-trivial and served as a contribution of this work.

Next, a relatively significant contribution of this thesis was the quantification of critical terms in the radio link budget for the radio repeating system. Due to the fact that the path planning used in this thesis is an inherently predictive planning technique, the quantification of various gain and loss terms in the radio links were essential for effective planning. This thesis presents experiments that were performed which not only validated the omnidirectional Cobham antenna patterns used in the system but also validated path loss models for the particular environment of Kentland Farms. More specifically, the omnidirectional antenna patterns were experimentally validated to contain main and side lobes in the vertical plane while exhibiting symmetric performance in the horizontal plane. Also, based on a path loss experiment coupled with residual comparisons, student's t-tests, and correlation analyses, the Hata Open model was selected as being appropriate for line-of-sight conditions and the Hata Suburban model was selected as being appropriate for non-line-of-sight conditions at the Kentland Farms test site. Not only does this provide further validation of the empirical path loss models for the radio community, but this verified that simple empirical models could be used in a path planning setting to predict average expected path losses given a specific vehicle configuration.

Considering the radio repeating path planner itself, the major contribution of this thesis was the formulation and implementation of a novel path planner that attempted to predict and optimize radio link quality throughout an unmanned vehicle operation. The testing of this path planner in Chapter 4 demonstrated the applicability of using a UAV as an aerial radio repeater node for UGV operations. In this testing, based on the environments that were simulated, it was demonstrated that this radio repeating concept is applicable to environments that contain building geometry commonly associated with open, campus, or suburban areas. This claim is supported by the data that in the 150 tests performed, the radio repeating path planner produced respectably strong predicted signal strengths on average. For the open environments simulated and the settings used, the planner produced average signal strengths of -53.7 and -53.6 dBm for the GS-to-Heli and the Heli-to-GV links, respectively.

Similarly, for the campus environments simulated, the planner produced average predicted GS-to-Heli and Heli-to-GV signal strengths of -53.2 and -57.5 dBm, respectively, and for the suburban environments simulated, the planner produced average predicted strengths for these links of -53.9 and -53.3 dBm, respectively. Even more so, with the appropriately selected initial conditions and goal location, the path planner was able to produce average predicted signal strengths over the course of the generated paths as high as -48.9 and -51.2 dBm for the GS-to-Heli and Heli-to-GV links in a simulated open environment, -53.9 and -52.1 dBm for a simulated campus environment, and -51.8 and -50.7 dBm for a simulated suburban environment.

Additionally, when comparing the radio repeating path planner to a number of naïve radio repeating counterparts, the path planner consistently outperformed the naïve variants in terms of averaged signal strengths. The results in their full detail are displayed in Table 4.4 but in short, the radio repeating path planner produced average signal strengths where at least one of the radio repeating links was 8 dB or more higher than the average strengths of the links produced by the naïve variants. This was common across all simulations for all naïve variants and all map types. Additionally, in some cases, the radio repeating path planner produced average signal strengths that were as much as 26 dB higher than the ones produced by a naïve variant, which is a very significant result.

More so than just validating the path planner in simulations, this thesis also describes the development of the system architecture and software required to implement the radio repeating operation on a vehicular system. This proved as a very significant contribution of this thesis in that not only was the radio repeating operation able to be performed with a ground vehicle and unmanned autonomous helicopter, but the architecture and software developed could potentially find uses for future research initiatives by the Unmanned Systems Lab as well. Referring to the data displayed in section 5.2, it was determined that although some minor discrepancies existed between the simulated and collected radio signal strengths, the path planner produced the same thematic results as those observed in the field test. This, coupled with the observation that the performance of the path planner agrees with intuitive performance for maintaining healthy communication links, serves to provide confidence that the planner will produce favorable radio repeating results in other environments as well. Also, in addition to simply verifying that the radio modeling of the path planner was sufficient to predict received signal strengths on average, both a path planning radio repeating and a naïve radio repeating field test was performed and upon comparing the results, it was shown that the path planning variant outperformed the naïve variant. Quantitatively, while the GS-to-Heli links in the two variants produced comparable average signal strengths of -54.5 and -51.7 dBm for the path planning and naïve cases, respectively, the Heli-to-GV links exhibited strikingly different results. The path planner's Heli-to-GV link produced an average signal strength of -49.7 dBm throughout the test whereas the naïve variant's Heli-to-GV link produced an average signal strength of -58.2 dBm with measurements as low as -68 dBm at certain points during the flight test. This clearly shows the superiority of intelligently navigating the aerial vehicle and ground vehicle together rather than simply decoupling the problem and choosing a naïve placement for the aerial vehicle. Also, although only a single flight test was performed due to the temporal and monetary costs associated with flight operations, the data collected served as a powerful indicator of the performance of the radio repeating path planner and displayed promise that the radio repeating operation can be beneficial to beyond-line-of-sight unmanned vehicle applications.

Lastly, although the developments in roadway detection are not sophisticated enough to serve as an alternate for an accurate 3D terrain model in the radio repeating path planner, the roadway detection algorithm serves as a novel approach for detecting roadways in images. This roadway detector, while only discussed briefly, was an additional major contribution of this thesis that itself is worth merit. This detector defined a roadway as an image feature with well-defined, long, and continuous edges. More so, these edges produced opposite intensity gradients at either side of the roadway. By using this definition for a roadway, the image processing algorithm was able to combine techniques from edge detection and A* path planning to identify the roadways in aerial imagery and to fill them in once they were found. While the algorithm's success was limited to imagery where a single roadway was a dominant feature in the images, the algorithm produced very successful results for this class of imagery. For example, for the images with well-contrasting backgrounds and well-defined edges like the Type I and Type II images described in section 6.2, the algorithm exhibited a 100% success rate for detecting the proper roadway in 170 trials with an ability to fill 97% of the roadways in the Type I images and 100% of those in the Type II images. This performance only grew weaker when complications in the images presented themselves like poorly defined edges and tree shadows covering the roadways. For images with these complications like the Type III and Type IV images mentioned in section 6.2, the algorithm detected the correct roadway 100% of the time with an average fill percentage of 64% in the Type III images and 60% of the time with only an average percentage of 17% fill for the Type IV images. Regardless of this poor performance for the Type IV images, these images were among the most complex of the ones tested. For the image types where the roadway was well-defined though, the algorithm exhibited very successful results.

Overall, this thesis presents a number of significant developments and novel approaches for using A*-based techniques to perform a radio repeating operation with an unmanned vehicle team. The work presented here includes aspects of path planning, radio modeling, simulated testing, and full system implementation. Therefore, this work serves as a complete account of the various developments for the radio repeating operation. More so, initial developments of a roadway detection algorithm are also included to both serve as an additional application of the A*-based path planning methods and as an aid for additional unmanned vehicle operations.

7.2 Suggestions for Future Work

While the developments presented in this thesis represent a significant amount of progress towards the implementation of an unmanned system capable of a radio repeating operation, a number of improvements can be made to the various components of this thesis if this work is desired to be continued further. Therefore, this section presents a brief discussion of potential future work for some of the elements of this thesis.

First, when considering the radio repeating path planner, the A*-based path planning that was discussed has been shown to be very applicable for the case of one aerial vehicle and one ground vehicle. Additionally, a brief discussion was also presented in section 4.2.4 for an extension of this work to the cases of two and even three aerial vehicles. However, as can be noted, due to the computational complexity of searching through a coupled state space that contains states for the coordinates of multiple aerial vehicles, it is only suggested to use the approach of this thesis for two or potentially three aerial vehicles. If very complex environments are encountered where the use of a large number of aerial vehicles is desired, the path planning problem may need to be formulated by alternative means. For example, using swarm-based methodologies may be more advantageous for a large and more complicated system.

Next, in order to attain the computational efficiency required for the radio repeating path planner, empirical models were used for the quantification of the expected path losses in the radio link budget. While the data of section 5.2 has shown this to be a reasonable predictor of radio signal strengths for the environment tested, the use of these models does not capture all of the intricacies associated with radio propagation. At the same time, it would be impractical to expect that deterministic path loss quantification methods be investigated since as discussed in the literature search, these are very computationally intensive which make them infeasible for the purposes of path planning. Rather, instead of this, a more closed loop methodology could be employed to supplement the path planner with additional environment-specific information. As it stands, the path planner is entirely predictive in the sense that paths are planned a priori based on simulated and modeled results. However, taking inspiration from the radio mapping work presented in the literature search, corrective behavior could be added to the planner by either generating a connectivity map based on radio signal strength measurements or by updating the path loss models to reflect the measurements taken throughout an operation. For example, since the empirical path loss models are linear when plotted on a logarithmic scale for distance, linear least squares regression could be used to generate a potentially more accurate path loss model for a specific environment of interest. To this end, Appendix A contains a discussion of linear least squares regression and although an environment-specific path loss model may not be validated by extensive research, it may provide more accurate results for a specific operating environment.

In addition to this, if computational efficiency is a concern, it should be noted that the most expensive of the computations in the cost function for the radio repeating path planner is that which evaluates the Fresnel ellipsoids for the radio links. For the current version of the path planner, Fresnel ellipsoid cross sections are sampled at each state and intersections between the sampled points and obstacles are tallied. Since this involves the evaluation of numerous sample points at each state in the search process, this requires a large amount of computational effort. Therefore, if more computation speed is required, a new method of evaluating intersections with the Fresnel ellipsoids may need to be investigated.

Additionally, from a software standpoint, a more automated process needs to exist for incorporating terrain maps with the radio repeating path planner. For this work, due to the fact that the stereovision system at the Unmanned Systems Lab was in a continuous state of development during the time when the work of this thesis was done, the ability to generate 3D terrain maps was not available when field testing of the radio repeating system was performed. Because of this, the 3D terrain map used for the Kentland Farms test facility was generated by taking measurements from the area and crafting a finite element mesh over a CAD (Computer-Aided Design) model of that area. Therefore, the radio repeating software was written to accept that mesh and code may need to be written to make a modification so that point clouds from 3D terrain models can be accepted as well.

Next, also from a practical standpoint, some additional consideration can be given to effective ingress and egress routes for the vehicles in the radio repeating operation. Most of the work and simulations in this thesis assumed that the vehicles were already located in an environment of interest when the path planning was performed. However, in an actual disaster response operation, the environment of interest may be far from the location of the responders and because of this, it would be beneficial to incorporate an additional path planner to navigate the vehicles to the environment effectively. This planner may incorporate additional considerations like fuel burn and speed of travel and can be incorporated to ensure a more effective complete disaster response operation.

Finally, when considering the roadway detection algorithm, the algorithm was only written to detect a single roadway in an image. While this was not a limitation for the aerial imagery tested, a few modifications can be made to make the algorithm able to detect multiple roadways in images where networks of roadways may exist. Also, when considering the fact that the algorithm was not quite as successful at detecting roadways when their edges were not as clearly defined or when tree shadows were covering them, modifications may be required to make the algorithm more robust to these complexities.

Regardless of the numerous suggestions for future work presented here, this thesis contains many significant developments for an unmanned system capable of performing a radio repeating operation. However, by addressing some or all of the recommendations presented here, the work of this thesis can be made more robust to complexities, more computationally feasible, and more applicable to a wider variety of situations. Therefore, by combining the work of this thesis with potential future developments, a powerful and high impact system can be created that may provide substantial benefits to both first responders and the robotics community alike.

Bibliography

[1] V.S. Abhayawardhana et al. “Comparison of Empirical Propagation Path Loss Models for Fixed Wireless Access Systems”. In: *IEEE Vehicular Technology Conference.* Vol. 61. 1. 2005, pp. 73–77.

[2] Nadia S. Adawi et al. “Coverage Prediction for Mobile Radio Systems Operating in the 800-900 MHz Frequency Range”. In: *IEEE Transactions on Vehicular Technology* 37.1 (1988), pp. 3–72.

[3] Mustafa Ayad, Peter Nielson, and R.M. Voyles. “RF Mapping for Intelligent Repeater Placement and Signal Recovery”. In: *Emergency Management and Robotics for Hazardous Environments (EPRRSD).* 2011.

[4] Arun Ayyagari, Jeff P. Harrang, and Sankar Ray. “Airborne Information and Reconnaissance Network”. In: *IEEE Military Communications Conference, 1996 (MILCOM ’96).* 1996, pp. 230–234.

[5] Constantine A. Balanis. *Antenna Theory: Analysis and Design.* 3rd ed. Hoboken, NJ: John Wiley, 2005.

[6] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[7] Gary S. Brown. “Coherent Wave Propagation through a Sparse Concentration of Particles”. In: *Radio Science* 15.3 (1980), pp. 705–710.

[8] Gary S. Brown and William J. Curry. “A Theory and Model for Wave Propagation through Foliage”. In: *Radio Science* 17.5 (1982), pp. 1027–1036.

[9] John Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), pp. 679–698.

[10] L. Chaimowicz et al. “Deploying Air-Ground Multi-Robot Teams in Urban Environments”. In: *Multi-Robot Systems- From Swarms to Intelligent Automata Volume III.* Ed. by Lynne Parker, Frank Schneider, and Alan Schultz. Springer Netherlands, 2005, pp. 223–234. ISBN: 978-1-4020-3389-6.

[11] Chia Hsun Chiang et al. “A Comparative Study of Implementing Fast Marching Method and A* Search for Mobile Robot Path Planning in Grid Environment: Effect of Map Resolution”. In: *IEEE Workshop on Advanced Robotics and its Social Impacts.* 2007.

[12] Emmanuel Christophe and Jordi Inglada. “Robust Road Extraction for High Resolution Satellite Images”. In: *IEEE International Conference on Image Processing, 2007 (ICIP 2007).* Vol. 5. 2007, pp. 437–440.

[13] John M. Cimbala. *Correlation and Trends.* 2010. URL: `http://www.mne.psu.edu/me345/`.

[14] John M. Cimbala. *Two Samples Hypothesis Testing.* 2010. URL: `http://www.mne.psu.edu/me345/`.

[15] Cory Dixon and Eric Frew. “Electronic Leashing of an Unmanned Aircraft to a Radio Source”. In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference.* 2005.

[16] Cory Dixon, Eric Frew, and Brian Argrow. “Radio Leashing of an Unmanned Aircraft”. In: *Collection of Technical Papers - Infotech at Aerospace: Advancing Contemporary Aerospace Technologies and their Integration.* Vol. 2. 2005, pp. 1093–1102.

[17] Richard O. Duda and Peter E. Hart. “Use of the Hough Transform to Detect Lines and Curves in Pictures”. In: *Communications of the ACM* 15.1 (1972).

[18] Dave Ferguson, Thomas Howard, and Maxim Likhachev. "Motion Planning in Urban Environments: Part II". In: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2008.

[19] Dave Ferguson, Maxim Likhachev, and Anthony Stentz. "A Guide to Heuristic-based Path Planning". In: *Proceedings of ICAPS Workshop on Planning under Uncertainty for Autonomous Systems*. 2005.

[20] Jonathan Fink and Vijay Kumar. "Online Methods for Radio Signal Mapping with Mobile Robots". In: *2010 IEEE International Conference on Robotics and Automation*. 2010.

[21] Eric A. Hansen and Rong Zhou. "Anytime Heuristic Search". In: *Journal of Artificial Intelligence Research* 28 (2007), pp. 267–297.

[22] Masaharu Hata. "Empirical Formula for Propagation Loss in Land Mobile Radio Services". In: *IEEE Transactions on Vehicular Technology* VT-29.3 (1980), pp. 317–325.

[23] Mongying A. Hsieh, Vijay Kumar, and Camillo J. Taylor. "Constructing Radio Signal Strength Maps with Multiple Robots". In: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*. 2004.

[24] Yi Huang. *Antennas: From Theory to Practice*. Chichester, UK: John Wiley and Sons Ltd., 2008.

[25] Sven Koenig and Maxim Likhachev. "Fast Replanning for Navigation in Unknown Terrain". In: *IEEE Transactions on Robotics* 21.3 (2005).

[26] Bryan Krawiec et al. "Post Detonation Autonomous Robotic Response System". In: *Emergency Management and Robotics for Hazardous Environments (EPRRSD)*. 2011.

[27] *L-com CA-195R Data Sheet*. 2012. URL: http://www.l-com.com/item.aspx?id=22247.

[28] *L-com SMA Male Crimp, Right Angle Shrouded for RG58, 195-Series Cable*. 2012. URL: http://www.l-com.com/item.aspx?id=23963.

[29] *L-com TNC Female Crimp Bulkhead for 195-Series Cable*. 2012. URL: `http://www.l-com.com/item.aspx?id=23926`.

[30] Steven M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[31] Maxim Likhachev et al. "Anytime Search in Dynamic Graphs". In: *Artificial Intelligence Journal* 172.14 (2008), pp. 1613–1643.

[32] Aleksandar Neskovic, Natasa Neskovic, and George Paunovic. "Modern Approaches in Modeling of Mobile Radio Systems Propagation Environment". In: *IEEE Communications Surveys* 3.3 (2000), pp. 2–12.

[33] H.G. Nguyen et al. "Autonomous Communication Relays for Tactical Robots". In: *Proceedings of ICAR 2003 - The 11th International Conference on Advanced Robotics.* 2003.

[34] Hoa G. Nguyen, Nathan Farrington, and Narek Pezeshkian. "Maintaining Communication Link for Tactical Ground Robots". In: *Proceedings of AUVSI's Unmanned Systems North America 2004*. 2004, pp. 311–323.

[35] Hoa G. Nguyen et al. "Autonomous Mobile Communication Relays". In: *Proceedings of the SPIE - The International Society for Optical Engineering*. 2002.

[36] Hoa G. Nguyen et al. "Maintaining Communication Link for a Robot Operating in a Hazardous Environment". In: *Proceedings of the 10th International Conference on Robotics and Remote Systems for Hazardous Environments*. 2004, pp. 256–263.

[37] Charles W. Niessen. "Battlefield Connectivity via Airborne Communications Nodes". In: *Proceedings of the SPIE - The International Society for Optical Engineering.*

[38] Y. Okumura et al. "Field Strength and its Variability in VHF and UHF Land-Mobile Radio Service". In: *Review of the Electrical Communication Laboratory (Tokyo)* 16 (1968), pp. 825–873.

[39] Ramesh Chembil Palat, A. Annamalai, and Jeffrey H. Reed. “Cooperative Relaying for Ad-hoc Ground Networks using Swarm UAVs”. In: *IEEE Military Communications Conference, 2005 (MILCOM 2005).* Vol. 3. 2005, pp. 1588–1594.

[40] John David Parsons. *The Mobile Radio Propagation Channel.* Chichester: John Wiley, 2000.

[41] Michael S. Rose. “Design of a Helicopter Deployable Ground Robotic System for Hazardous Environments”. MS thesis. Virginia Polytechnic Institute and State University, 2010.

[42] *S Band 4 Section Collinear.* Available from Cobham plc. 2011.

[43] Beril Sirmacek and Cem Unsalan. “Road Detection from Remotely Sensed Images Using Color Features”. In: *Proceedings of 5th International Conference on Recent Advances in Space Technologies (RAST 2011).* 2011, pp. 112–115.

[44] Anthony Stentz. “Optimal and Efficient Path Planning for Partially-Known Environments”. In: *Proceedings - IEEE International Conference on Robotics and Automation.* 1994.

[45] Anthony Stentz. “The Focussed D* Algorithm for Real-Time Replanning”. In: *Proceedings of the International Joint Conference on Artificial Intelligence.* 1995.

[46] Jakov V. Toporkov, Roger T. Marchand, and Gary S. Brown. “On the Discretization of the Integral Equation Describing Scattering by Rough Conducting Surfaces”. In: *IEEE Transactions on Antennas and Propagation* 46.1 (1998), pp. 150–161.

[47] Joram Walfisch and Henry L. Bertoni. “A Theoretical Model of UHF Propagation in Urban Environments”. In: *IEEE Transactions on Antennas and Propagation* 36.12 (1988), pp. 1788–1796.

[48] Lin Zhao and Xili Wang. “Road Extraction in High Resolution Remote Sensing Images Based on Mathematic Morphology and Snake Model”. In: *2010 Proceedings of 3rd International Congress on Image and Signal Processing (CISP 2010).* 2010, pp. 1436–1440.

Appendix A

Derivations

This appendix includes derivations that are referenced in the primary text of the thesis. These are meant to supplement and clarify some of the mathematical details discussed throughout the thesis.

A.1 Fresnel Ellipsoid Derivation

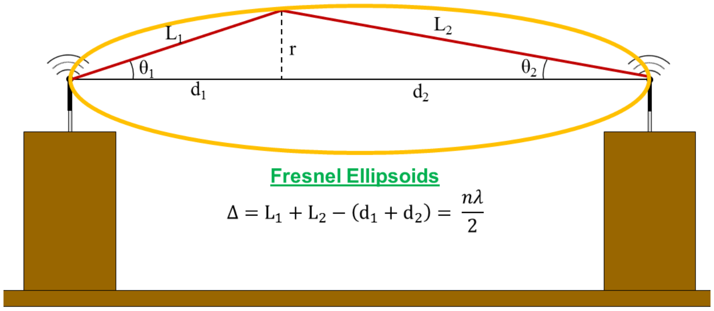


Figure A.1: Fresnel ellipse diagram

Shown in Figure A.1 is the cross section of a Fresnel ellipsoid for a radio transmitter/receiver pair. Fresnel ellipsoids, whose volumes define regions known as Fresnel zones, are important in radio network planning. In radio communication, assuming that a perfectly isotropic antenna is available, electromagnetic waves propagate equally in all directions from the transmitter. In a completely obstacle free environment, the line-of-sight or direct line wave will propagate to the receiver and all of the other waves will propagate off into the free space. However, in an obstacle-rich environment, some of these waves may reflect off of obstacles and may also reach the receiver. Since these waves will have taken a longer path than the direct line-of-sight path from the transmitter to the receiver, they will typically be out of phase with the direct line-of-sight signal and will likely destructively interfere with it causing a degradation in the received signal strength of the signal. This phenomenon is known as multipath and in order to minimize its negative effects, it is common practice to keep the volume near the transmitter and receiver clear of obstacles so that the strongest multipath signals do not propagate to the receiver. This concept is quantified by the Fresnel zones of a transmitter/receiver pair which are defined by the constraint that any multipath signal that reaches the receiver travels a distance of $\frac{n\lambda}{2}$ more than the direct line-of-sight signal. Here λ is the wavelength of the radio signal. These zones take the form of ellipsoids and the derivation of these Fresnel ellipsoids is given below. More specifically, a relation for calculating the maximum radius of the first Fresnel zone (n = 1) is derived since this zone will contain the strongest and most destructive multipath signals [40]. It should be noted that while [40] provides the equation for r_{max} below, no derivation is given. The derivation provided below was therefore produced by the author to provide further detail and insight into the concept of Fresnel ellipsoids.

As noted above, the the Fresnel zones are defined by the property that non-direct signals traveling from the transmitter to the receiver travel distances that are half wavelength increments longer than the direct line-of-sight signal. Therefore, referring to to Figure A.1, cross sections of the Fresnel zones are defined by the relation below.

$$\Delta = L_1 + L_2 - d_1 - d_2 = \frac{n\lambda}{2}$$

From the geometry in Figure A.1, we can write L_1 and L_2 as

$$L_1 = \frac{d_1}{\cos\theta_1} \quad , \quad L_2 = \frac{d_2}{\cos\theta_2}$$

Therefore, Δ can be written as

$$\Delta = \frac{d_1}{\cos\theta_1} + \frac{d_2}{\cos\theta_2} - d_1 - d_2$$

Assuming that small angle assumptions are valid, which occurs when $r \ll d_1$ and $r \ll d_2$, we can represent $\sin x$ as $\sin x \approx \tan x \approx x$, which gives

$$\theta_1 \approx \frac{r}{d_1} \quad , \quad \theta_2 \approx \frac{r}{d_2}$$

Typically, with this small angle assumption, we can also represent $\cos x$ as $\cos x \approx 1$. However, since this linear approximation would result in $\Delta = 0$, $\cos x$ will instead be approximated with the first two terms of its Taylor series expansion as opposed to just its first term. This decision also aligns well with the fact that we are interested in deriving the relation for an ellipse which is a quadratic curve, not a linear one. Thus, we have $\cos x \approx 1 - \frac{x^2}{2}$ and using the result for θ_1 and θ_2 above, we have

$$\cos\theta_1 \approx 1 - \frac{r^2}{2d_1^2} \quad , \quad \cos\theta_2 \approx 1 - \frac{r^2}{2d_2^2}$$

This allows us to write Δ as

$$\Delta = \frac{d_1}{1 - \frac{r^2}{2d_1^2}} + \frac{d_2}{1 - \frac{r^2}{2d_2^2}} - d_1 - d_2$$

Combining the fractions gives

$$\Delta = \frac{d_1 r^2}{2d_1^2 - r^2} + \frac{d_2 r^2}{2d_2^2 - r^2}$$

Since in our small angle approximations we assumed that $r \ll d_1$ and $r \ll d_2$, the r^2 term in the denominators of the two terms of Δ will be insignificant compared to $2d_1^2$ and $2d_2^2$, respectively, and may therefore be neglected. This gives

$$\Delta = \frac{r^2}{2d_1} + \frac{r^2}{2d_2}$$

$$\Delta = \frac{r^2}{2}\left(\frac{d_1 + d_2}{d_1 d_2}\right)$$

Applying the constraint that $\Delta = \frac{n\lambda}{2}$ gives

$$\frac{r^2}{2}\left(\frac{d_1 + d_2}{d_1 d_2}\right) = \frac{n\lambda}{2}$$

$$r = \sqrt{\frac{n\lambda d_1 d_2}{d_1 + d_2}}$$

Next, defining the total straight line distance from the transmitter to the receiver as $D \triangleq d_1 + d_2$, the equation for the radii of the Fresnel ellipses are

$$\boxed{r = \sqrt{\frac{n\lambda d_1 (D - d_1)}{D}} \quad , \quad n = 1, 2, 3, ... \quad , \quad valid\ for\ r \ll d_1, d_2}$$

The above equation provides a calculation for the radius of the n^{th} Fresnel zone ellipse at all points along its major axis. Note that when $d_1 = 0$ and when $d_1 = D$, $r = 0$. This implies that the transmitter and receiver locations correspond to the endpoints of the major axis of the ellipse. By using this, calculating the maximum minor axis radius r_{max} of the ellipse, and using symmetry, we can determine a parameterization for the 3-dimensional Fresnel zone ellipsoids. While this reasoning would allow us to find parameterizations of all n Fresnel ellipsoids, since we are primarily interested in the first Fresnel ellipsoid, n will be set as $n = 1$ for the remainder of the analysis. Making this assignment, r becomes

$$r = \sqrt{\frac{\lambda d_1 D - \lambda d_1^2}{D}}$$

Taking the derivative of r with respect to d_1 and setting it equal to zero will determine the d_1 value at which r is maximum. Taking the derivative, we obtain

$$\frac{dr}{dd_1} = \frac{1}{2}\sqrt{\frac{\lambda}{D}}\frac{D - 2d_1}{d_1(D - d_1)}$$

This derivative is undefined at $d_1 = 0$ and $d_1 = D$ which is understandable since an ellipse has a vertical slope at its major axis endpoints. Additionally, $\frac{dr}{dd_1} = 0$ at $d_1 = \frac{D}{2}$. Thus, the maximum first Fresnel zone radius is given by evaluating r at this point. This gives r_{max} as

$$r_{max} = \sqrt{\frac{\lambda D}{4}}$$

Next, since radio waves are typically defined in terms of frequencies instead of wavelengths, we can use the relation that $\lambda f = c$, where c is the speed of light and f is the frequency of the radio wave, to write

$$r_{max} = \sqrt{\frac{Dc}{4f}}$$

Inputting the speed of light and allowing f to be in units of Megahertz (MHz), the maximum radius of the first Fresnel zone is given below. This result agrees with that presented in [40].

$$\boxed{r_{max} = 8.66\sqrt{\frac{D}{f}} \ \ , \ \ where \ \ f \ [=] \ MHz \ , \ D \ [=] \ m}$$

With this, we can write a representation of the first Fresnel ellipsoid as shown below where r_{max} is given above and D is the straight line distance between the transmitting and receiving antennas.

$$\boxed{1^{st} \ Fresnel \ Ellipsoid \ \rightarrow \ \frac{x^2}{\left(\frac{D}{2}\right)^2} + \frac{y^2}{r_{max}^2} + \frac{z^2}{r_{max}^2} = 1}$$

As a final note, we would like to verify that our assumptions are valid. Taking f as a common radio frequency of 2400 MHz and the distance between the transmitter and receiver as $D = 100$ m, we have $r_{max} = 1.768$ m. This gives us that $\frac{r_{max}}{d_1} = \frac{r_{max}}{d_2} = 0.035$

and $\theta_1 = \theta_2 = \tan^{-1}\left(\frac{r}{d_1}\right) = 0.035$. This shows that $r \ll d_1, d_2$ and that our small angle assumptions are valid even at only 100 m of separation between transmitter and receiver.

A.2 Linear Least Squares Regression

This section derives a formula for finding the parameters of the best fit line through a series of points which applies to the future work recommendation of using least squares to generate a potentially more accurate path loss model for a specific environment. This derivation will formulate the problem as an unconstrained optimization problem and will use linear algebra techniques to solve the problem. It should be noted that while the least squares problem is common in engineering, this derivation was produced by the author without external reference. Regardless, since it is a commonly available result, [6] can serve as a reference for the derivation below.

The linear least squares problem is commonly written as: Find the parameters m^* and b^* such that the line $y = m^*x + b^*$ minimizes the sum of squares of the residuals $\hat{y}_i = y_i - \bar{y}_i$. Here y_i are the measured values, $\bar{y}_i$ are the estimated values produced by the evaluating the least squares regression line at the points x_i for $i = 1, 2, ..., n$, and n is the number of measurement pairs (x_i, y_i). This can be written mathematically as

$$minimize \quad \sum_{i=1}^{n} (y_i - \bar{y}_i)^2$$

Substituting in the equation for the estimates y_i, this optimization problem is equivalently written as

$$minimize \quad \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

Defining $a_i \triangleq \begin{bmatrix} x_i \\ 1 \end{bmatrix}$, $x \triangleq \begin{bmatrix} m \\ b \end{bmatrix}$, and $b_i \triangleq y_i$, the optimization problem can be written as

$$minimize \quad \sum_{i=1}^{n} (b_i - a_i^T x)^2$$

Using the standard 2-norm for vectors, this summation can be compactly written in a matrix form as

$$minimize \quad \| \, b - Ax \, \|_2^2$$

where

$$A = \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix} \in \mathbb{R}^{nx2} \quad , \quad b = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Performing some algebra, we have

$$minimize \quad \| \, b - Ax \, \|_2^2 \ \Leftrightarrow \ \ minimize \ \ (b - Ax)^T (b - Ax)$$

$$minimize \quad \| \, b - Ax \, \|_2^2 \ \Leftrightarrow \ \ minimize \ \ x^T A^T Ax - 2b^T Ax - b^T b$$

This is optimization problem is in a quadratic form and has a minimum if $A^T A > 0$. That is, $A^T A$ must be positive definite. Looking at $A^T A$, we have the following equivalences.

$$A^T A > 0 \quad \Leftrightarrow \quad z^T A^T Az > 0 \ \ \forall \ z \neq 0$$

$$A^T A > 0 \quad \Leftrightarrow \quad (Az)^T (Az) > 0 \ \ \forall \ z \neq 0$$

$$A^T A > 0 \quad \Leftrightarrow \ \| \, Az \, \|_2^2 > 0 \ \ \forall \ z \neq 0$$

Since $\| \cdot \|_2$ by definition only yields nonnegative results and only equals 0 when its agrument equals 0, it is qualitatively the same as $\| \cdot \|_2^2$ so we have

$$A^T A > 0 \quad \Leftrightarrow \quad \| Az \|_2 > 0 \ \ \forall \ z \neq 0$$

Since $\| Az \|_2 = 0$ only when $Az = 0$, we can define the null space of a as $\mathcal{N}(A)$ to get

$$A^T A > 0 \quad \Leftrightarrow \quad \mathcal{N}(A) = \{\mathbf{0}\}$$

$$A^T A > 0 \quad \Leftrightarrow \quad \dim(\mathcal{N}(A)) = 0$$

$$A^T A > 0 \quad \Leftrightarrow \quad \text{rank}(A) = 2$$

Here 2 is the number of columns in A. Also, based on the definition $a_i \triangleq \begin{bmatrix} x_i \\ 1 \end{bmatrix}$ which populates the A matrix, this rank condition is satisfied as long as we have measurements at two or more x_i values. This is typically satisfied, so we can make the claim that for a situation where two or more independent measurements are taken, we will have $A^T A > 0$ and therefore our optimization problem will have a well defined minimum. Returning to the problem, the minimum occurs by setting the gradient of the quadratic objective function to zero. This gives the optimal value of x, denoted as x^* as

$$2A^T A x^* - 2A^T b = 0$$

$$A^T A x^* = A^T b$$

Since $A^T A > 0$, $(A^T A)^{-1}$ exists and

$$\therefore \quad \boxed{x^* = (A^T A)^{-1} A^T b}$$

From x^* we can extract our optimal parameters m^* and b^* for our least squares regression line. Also, since $A \in \mathbb{R}^{nx2}$, we have $A^T A \in \mathbb{R}^{2x2}$, and due to the fact that 2 by 2 matrix inversions are easily calculated, this makes this regression calculation computationally inexpensive.

Appendix B

Radio Repeating Software

This appendix includes a detailed discussion of the software used in the radio repeating flight operation. It contains descriptions of the software modules used for both the Ground Control Station and the Ground Vehicle.

B.1 Ground Control Station Software

With the system architecture in place, the remaining complexity of creating a full system to perform the radio repeating mission was to generate the software for the Ground Station Computer and the Ground Vehicle Computer that handles all of the data transmission, interpretation, and path planning. This section discusses the primary module of software for the radio repeating mission which runs on the Ground Station Computer. This software was coded primarily in the National Instruments LabVIEW language but also made use of C++ dynamic-link library (.dll) files that handled the path planning operations.

First and foremost, in order to make a system that can perform all of the desired data communication described above while also doing path planning, the code for the path planning itself had to be written so that it could easily be called from a larger master program.

Therefore, even though Matlab provides a convenient environment for writing, debugging, and testing code, C++ is a better environment for generating code that is to be run in conjunction with other components of the system since code can be compiled as dynamic-link library (.dll) files. By doing this, not only can the code run faster since allocations can be performed during the compile step of the code generation but the code can be easily integrated as a module into a larger program. Therefore, all of the path planning code and a few other conversion files were rewritten in C++ and compiled as a set of .dll files. Since LabVIEW has a convenient interface for calling .dll files with a number of input arguments, .dll files were chosen to be used instead of executable files.

With the primary path planning code written as a set of .dll files, the remaining components of the Ground Station Computer software module could be written strictly in the LabVIEW environment. LabVIEW was chosen as the coding language for this system due to its efficiency, ease of performing communications with other devices, and ability to generate intuitive user interfaces. Additionally, by incorporating all of the code modules into one project file, LabVIEW provided a convenient method of code organization and management. Figure B.1 contains an architecture diagram for the Ground Station Computer software module and its various components are discussed below.

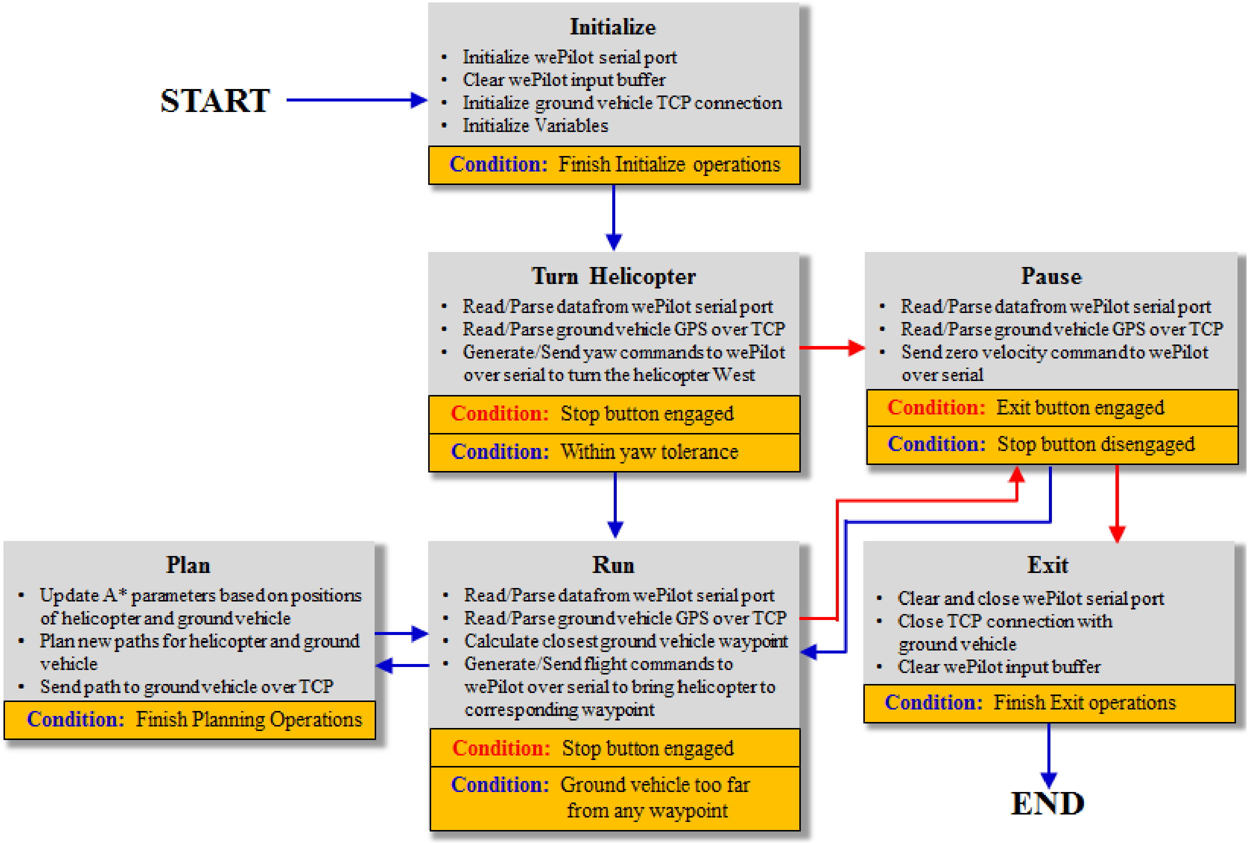

START
Initialize
Initialize wePilot serial port
Clear wePilot input buffer
Initialize ground vehicle TCP connection
Initialize Variables
Condition: Finish Initialize operations
Turn Helicopter
Read/Parse data from wePilot serial port
Read/Parse ground vehicle GPS over TCP
Generate/Send yaw commands to wePilot over serial to turn the helicopter West
Condition: Stop button engaged
Condition: Within yaw tolerance
Pause
Read/Parse data from wePilot serial port
Read/Parse ground vehicle GPS over TCP
Send zero velocity command to wePilot over serial
Condition: Exit button engaged
Condition: Stop button disengaged
Plan
Update A* parameters based on positions of helicopter and ground vehicle
Plan new paths for helicopter and ground vehicle
Send path to ground vehicle over TCP
Condition: Finish Planning Operations
Run
Read/Parse data from wePilot serial port
Read/Parse ground vehicle GPS over TCP
Calculate closest ground vehicle waypoint
Generate/Send flight commands to wePilot over serial to bring helicopter to corresponding waypoint
Condition: Stop button engaged
Condition: Ground vehicle too far from any waypoint
Exit
Clear and close wePilot serial port
Close TCP connection with ground vehicle
Clear wePilot input buffer
Condition: Finish Exit operations
END

Figure B.1: Software architecture for the Ground Station Computer

As can be seen in Figure B.1, the Ground Station Computer software was written as a finite state machine. That is, the software has a finite number of states that the execution can reside in and depending on which state the program is operating in at a particular moment in time, different tasks are executed. The purpose of writing the code in this fashion is two-fold. First off, by writing the code in a state machine format, processing is improved since not every task in the entire code needs to be executed at once. Secondly, thinking about the situation logically, the program might not want to execute the same set of tasks throughout the entire mission. For example, planning a path only needs to be performed at the beginning of the mission and if the Ground Vehicle ever deviates from its prescribed path too far. Thus, all planning operations can be encapsulated within one state which only needs to be executed when necessary. Additionally, obviously initializations and closing operations only need to be executed once throughout the entire execution so each of them can be encapsulated within their own states. The remainder of this section of text is devoted to describing the various states of the Ground Station Computer software and the transitions that occur between the states.

Starting with the Initialize state, this is the first state that is executed once the program is run and this state is responsible for performing all of the start-up procedures for the code. It initializes the communication channels between the Ground Control Station, the helicopter, and the Ground Vehicle. Since the communication between the Ground Control Station and the helicopter is done via RS-232, a serial port is initialized and its input and output buffers are cleared. More so, a separate string variable is initialized and cleared that will represent an input buffer for the serial communication. This input buffer is used to aid with serial data parsing as will be explained more thoroughly when the Run state is discussed. Additionally, this state also establishes a TCP connection with the Ground Vehicle where the Ground Station Computer acts as the server and the Ground Vehicle Computer acts as the client. Finally, all of the other variables pertinent to the execution of the code are initialized before the Initialize state is completed. After all of these operations have successfully been performed, the Initialize state finishes and the program shifts to the Turn Helicopter state.

The Turn Helicopter state is the next state in the sequence of the radio repeating code. This state is intended to be an initial controller for the helicopter and acts to turn the helicopter to face West. The purpose of this state is that with the helicopter facing in this direction, the northing and easting commands that will be used to control the helicopter in the Run state can be decoupled into lateral and longitudinal motions, respectively. The choice of having the helicopter face West was arbitrary but was chosen since the primary direction of travel for the helicopter in the testing area was westward. The yaw controller for this step of the code is a simple proportional controller with a k_p value os 0.5 and a saturation feature. The controller simply takes the current yaw angle of the helicopter, subtracts it from the desired yaw angle (-90 degrees) and sends a yaw rate command to the helicopter that is half of that difference if the absolute value of the difference between the desired yaw angle and the yaw angle is less than 10 degrees. If the difference is outside of this tolerance, the controller simply sends a 5 degrees per second in the appropriate direction. By using this control scheme, the helicopter rotates at a safe rate and slowly approaches its desired yaw value with little overshoot. It is implied by this discussion that the Turn Helicopter state also reads and parses serial data from the wePilot Flight Controller. Additionally, this state also reads and parses GPS data sent from the Ground Vehicle over TCP and displays this information to the operator.

All of these operations continue until either of two conditions are met. The first is that the helicopter reaches within 2 degrees of its desired yaw value of -90 degrees, which causes the Turn Helicopter state to exit and the Run state to begin. The second is if there is any oddity with the flight controller, the operator can engage the Stop button on the user interface and this will cause the helicopter to stop its motion. At this point it would be wise to exit the program, land the helicopter, and determine the source of the oddity.

Once the Turn Helicopter state completes itself, the program executes its Run state which is the primary state for the program to operate in. This state, as with the others, contains a set of functions that continue to operate until a state transition condition is met. In this Run state, the program essentially reads and parses status information from both the wePilot Flight Controller onboard the helicopter and from the Ground Vehicle Computer. It also generates flight commands that are sent to the helicopter to navigate it to an appropriate waypoint.

Expanding upon each of these tasks individually, the first task for the program is to read and parse data from the wePilot Flight Controller. As noted in the system architecture diagram above, this is done over serial communication. Therefore, in order to parse the data, the program starts by continuously reading the incoming bytes from the serial port and stores them into an input buffer string variable. At the same time, bytes are pulled sequentially off of the input buffer variable and they are inspected to see if they are the start byte for the message. Since both of these operations are being performed simultaneously, care has been taken to ensure that no race conditions occur. If the start byte for the message is found, the payload for the message is pulled out of the input buffer variable and the checksum is validated to give confidence that the message was received properly. If the checksum is correct, the parameters are scaled appropriately and the reading and parsing process continues. Due to LabVIEW's ability to specify loop timings and easily multithread segments of code, this reading and parsing process can be performed very quickly in excess of 50 Hz.

Next, in addition to reading and parsing the messages from the wePilot, the program also continuously parses the GPS position messages from the Ground Vehicle. As opposed to serial though, this is done over TCP and is a simpler process due to the level of abstraction that LabVIEW provides with TCP communication. Since there is more structure in the communication sequence with TCP than there is with serial, the messages from the Ground Vehicle are simply read and parsed with no need for sequentially picking through bytes and calculating checksums.

Thirdly, in addition to performing these communication tasks, the Run state also inspects the GPS locations received from the Ground Vehicle to determine which waypoint the helicopter should be guided to. Since the paths are planned for the helicopter and Ground Vehicle cooperatively, each waypoint for the Ground Vehicle has a corresponding waypoint for the helicopter. Therefore, a section of the Run state determines the waypoint that the Ground Vehicle is closest to and stores the desired waypoint for the helicopter as its corresponding one. For example, if the Ground Vehicle is closest to its 4^{th} waypoint, the 4^{th} waypoint for the helicopter is stored as its desired waypoint. In addition to this, this set of code also has a flag in it that if the Ground Vehicle is too far from any of its waypoints based on some tolerance set by the operator, then the code is sent into the Plan state where new paths are computed for the helicopter and Ground Vehicle based on their current positions. Although this might not be necessary with an autonomous Ground Vehicle, if a tele-operated or manned Ground Vehicle was used and if the operator decided to choose an alternative path to the goal, the path planner would be able to adapt to that desire and plan new paths for the helicopter and Ground Vehicle accordingly.

Lastly, once the desired waypoint for the helicopter is calculated, flight commands are generated and sent via the RS-232 serial port to the wePilot Flight Controller. As with the Turn Helicopter state, this flight command generator is a simple proportional controller with a saturation. In fact, since the helicopter was turned to face West during the Turn Helicopter state, two decoupled controllers are used to handle the northing and easting flight commands. These controllers work by using the GPS coordinates of the helicopter and the desired waypoint to calculate the error of the helicopter position in terms of northing and easting values. If either of the northing or easting locations are off by more than 3 meters, the helicopter is given a 1 meter per second velocity command in the appropriate direction. Otherwise, it is given a velocity command proportional to one tenth of the deviation ($k_p = 0.1$). This allows the helicopter to calmly approach the desired waypoints and has been tested to ensure safe operation of the helicopter. Once these commands are developed, a packet is structured and sent to the wePilot with the easting commands being longitudinal ones and the northing commands being lateral ones. It should also be noted that for the safety of the helicopter, it was decided to maintain it at a constant altitude. Aside from making the planning simpler by allowing it to remain in a 4-dimensional instead of a 5-dimensional state space, this ensures that the helicopter does not climb or descend in any unwanted fashion.

With all of these operations, the Run state performs all of the necessary communication tasks required throughout the main bulk of the mission. However, in the event of the user wanting to stop the helicopter or if a new path needs planned, the run state can be exited in two ways. The first is for the operator to click a Stop button in the user interface. This sends a single zero velocity command to the wePilot and switches the program to the Pause state. The second condition is that if the program is just initialized or if the Ground Vehicle's deviation from its path is beyond the tolerance specified by the operator, a zero velocity command is sent to the wePilot and the Run state will transition to the Plan state where new paths are generated. The Pause state will be discussed next followed by the Plan state.

In the Pause state, the program performs the same operations that are performed in the Run state except that instead of generating flight commands for the wePilot Flight Controller so that the helicopter is guided to a waypoint, it simply formats and sends zero velocity commands to the wePilot until the Pause state is exited by the operator. That is, this state reads and parses status data from the wePilot Flight Controller, reads and parses GPS data from the Ground Vehicle, and sends zero velocity commands to the helicopter. In doing this, the Pause state is operates exactly as is to be expected. It still processes incoming data but pauses the movement of the helicopter. This state is a safety mode and can be used at any point if the operator feels uncomfortable about the mission. Once the operator feels comfortable continuing the mission, the Stop button can be clicked again to disengage it and transition the program back to the Run state. Otherwise, if the operator would like to abort the mission, the Exit button can be clicked in which case the program will enter its Exit State.

The Plan state is the segment of the program that handles the path planning operations. This state can only be transitioned to from the Run state and since the state transition process involves sending a zero velocity command to the helicopter, it will stay stationary throughout the planning process. The reason for this is primarily safety but this also frees up processing power since without needing to send commands to the flight controller, much of the reading, parsing, and sending message functions do not need to be performed. In fact, since the path planning process is computationally intensive, it was decided to have it in its own state where no other external operations would be necessary. Therefore, this planning state simply consists of the following series of operations. First, the parameters for the A* path planning are updated based on the current positions of the helicopter and Ground Vehicle. Then, the path planning code for the radio repeating operation is executed and new paths are found. Finally, the Ground Vehicle's new path is sent to it via TCP communication. Expanding slightly on the path planning portion of this sequence of steps, the path planning code itself is first executed by calling a .dll file. Here, since there are a variety of inputs and outputs, many of them being arrays, pointers are used to pass the arguments back and forth between LabVIEW and the .dll. Also, since the search process may involve arrays of varying sizes as the search progresses, some memory is dynamically allocated so that no overflow conditions occur. Once the planning is complete, the paths are returned to LabVIEW and they are post-processed by executing another .dll that converts the paths from their local grid-based coordinates to GPS positions. By using compiled C++ code to perform these steps, the result is a Plan state that can often run in little time (often less than a second or two for the areas tested) and quickly return the program to its primary Run state. This state simply returns to the Run state after its planning operations are complete and does not wait for a user input to transition itself.

The final state in the Ground Station software is the Exit state. This state handles the closing and shut-down operations for the software and performs the following set of tasks. It clears the input and output buffers on the serial port that communicates with the wePilot Flight Controller, it clears the wePilot input buffer variable, and it closes the TCP connection with the Ground Vehicle. Once these operations have been performed, the program exits and the mission is deemed as being complete.

Although the software for the Ground Station may seem complicated initially, the finite state machine implementation serves to organize the code into useful segments that are only run when needed. This not only provides a more intuitive architecture for an operator but also minimizes the loading on the processor which allows for effective execution during a flight operation.

B.2 Ground Vehicle Software

As with the Ground Station, the Ground Vehicle contains a computer with a software module that allows it to read, parse, display, and send data to and from the Ground Station. This module was programmed in a similar manner as the Ground Station module with it being written as a finite state machine in the LabVIEW environment. Figure B.2 shows a diagram for the architecture of the Ground Vehicle software. It is simpler than the architecture for the Ground Station in that it only contains an Initialize, a Run, and an Exit state, but since the Ground Vehicle does not take an active role in sending commands to the helicopter or planning paths, these are all of the states that are necessary.

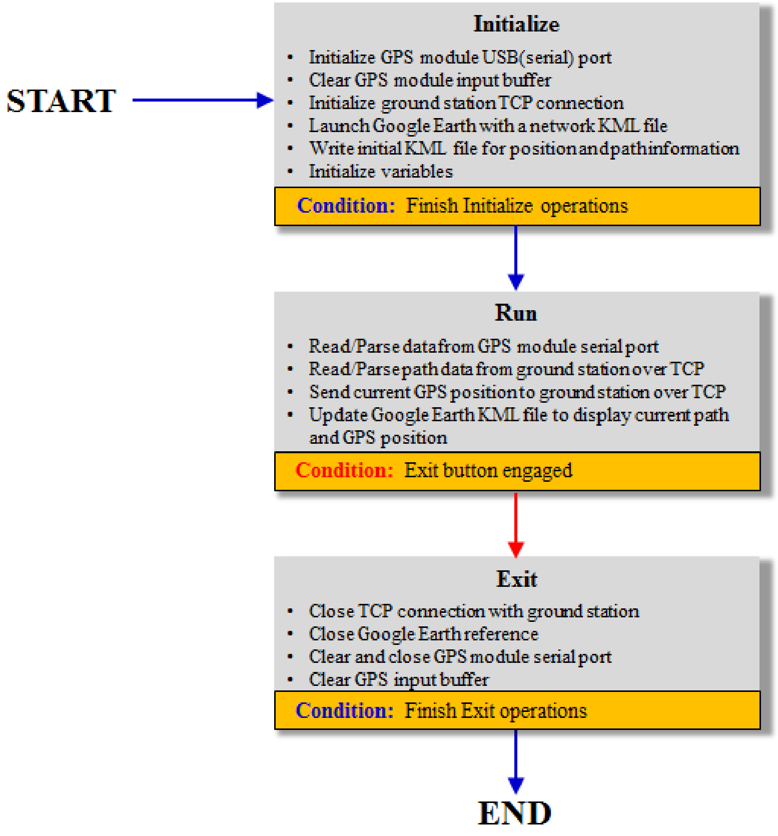

START
Initialize
• Initialize GPS module USB(serial) port
• Clear GPS module input buffer
• Initialize ground station TCP connection
• Launch Google Earth with a network KML file
• Write initial KML file for position and path information
• Initialize variables
Condition: Finish Initialize operations
Run
• Read/Parse data from GPS module serial port
• Read/Parse path data from ground station over TCP
• Send current GPS position to ground station over TCP
• Update Google Earth KML file to display current path and GPS position
Condition: Exit button engaged
Exit
• Close TCP connection with ground station
• Close Google Earth reference
• Clear and close GPS module serial port
• Clear GPS input buffer
Condition: Finish Exit operations
END

Figure B.2: Software architecture for the Ground Vehicle Computer

Starting with the Initialize state, this segment of code performs all of the start-up operations for the Ground Vehicle software. This software is responsible for gathering its GPS location, sending that location to the Ground Station Computer, receiving the path information from the Ground Station, and displaying that in a meaningful way to an operator. In order to provide this functionality, the Initialize state initializes all of the communication links and starts Google Earth (© 2012 Google) which is used to display the recommended path to the operator. First, the GPS module is connected via a USB port to the Ground Vehicle

Computer and an input buffer is initialized in a similar fashion as with the serial links in the Ground Station software. Next, a TCP connection is initialized between the Ground Vehicle and the Ground Station. This is also initialized in a similar manner as in the Ground Station software except for the fact that the Ground Station was initialized as a server and the Ground Vehicle as a client. After this, variables are initialized that are necessary throughout processing. Lastly, Google Earth is run and initialized. In Google Earth, details to display are saved in what are known as KML files. Path information can easily be stored in these files and they can be run to overlay the current location of the Ground Vehicle, the planned path, and the goal location to the user. Because of this, the Initialization state writes an initial KML file which is updated in the Run state as new information is obtained. Once all of these operations are finished, the Initialize state ends and shifts the program to the Run state.

The Run state for the Ground Vehicle is similar in nature to the Run state for the Ground Station in that it primarily deals with reading, parsing, and transmitting data. In this state, four main operations are performed. First, GPS data from the GPS module is read and parsed to obtain the current latitude and longitude of the Ground Vehicle. Secondly, as this data is obtained, the Run state sends this latitude and longitude data to the Ground Station software over TCP where it is used to determine where to move the helicopter to and if the Ground Vehicle has deviated from its path sufficiently to require re-planning of a path to the goal location. Thirdly, as path updates are sent from the Ground Station over TCP, the Run state reads and parses the path information for the operator to see. Finally, since looking at GPS coordinates in a numerical array does not provide a great deal of insight for a typical operator, this information is written to the KML file initialized in the Initialize state causing Google Earth to update its display with the most recent path, Ground Vehicle location, and goal location information.

With this path information, the operator may choose to drive along the recommended path towards the goal. If the Ground Vehicle operator instead decides to take an alternate path to the goal which is not the one recommended by the path planner, the path planner will generate new paths for the Ground Vehicle and helicopter once the Ground Vehicle has deviated sufficiently far from the recommended path. This tolerance is set by the operator in the Ground Station and this re-planning behavior is intended to provide flexibility to the Ground Vehicle operator in case obstacles are found or other information is obtained that prohibits the Ground Vehicle from traversing a recommended path. This Run state continues until the mission is complete in which case the operator can select an Exit button and the software will end the Run state and start its Exit state.

The Exit state is the final segment of the Ground Vehicle software module and performs all of the closing operations for the module. It ends the TCP connection with the Ground Station, closes the references that are used to run Google Earth, clears the input and output buffers for the GPS module serial port, closes the GPS module port, and clears the input buffer variable for the GPS module. Once all of these operations and formalities are finished, the software exits and the mission is considered to be complete.

As can be seen, this software provides a convenient way for the operator of the Ground Vehicle to obtain information about the desired trajectory to navigate on while also sending pertinent GPS information back to the Ground Station for helicopter control and path planning purposes. Finally, although this software was intended for a manned or tele-operated Ground Vehicle, only a few additions and modifications would need to be made to the Ground Vehicle software to allow it to deal with an autonomous Ground Vehicle as well.