

Collaborative Target Localization and Inspection Using a Heterogeneous Team of Autonomous Vehicles

David Burns Van Covern

**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**

Dr. Charles F. Reinholtz, Chairman
Alumni Distinguished Professor of Mechanical
Engineering and Engineering Education

Dr. Alfred L. Wicks, Co-Chairman
Associate Professor of Mechanical Engineering

Dr. Craig A. Woolsey
Associate Professor of Aerospace and Ocean Engineering Engineering

December 4, 2007
Blacksburg, Virginia

Keywords: Autonomous Vehicle, Air-Ground Team, Unmanned Teaming

Collaborative Target Localization and Inspection Using a Heterogeneous Team of Autonomous Vehicles

David Burns Van Covern

ABSTRACT

Autonomous vehicle development is a rapidly growing field that has vast possibilities for both military and commercial applications. Removing people from dangerous tasks will save lives. Continued research is necessary in order to build these new technologies and mature those already established. One area of potential in the unmanned vehicle community is that of fully autonomous cooperation. This area of research will allow multiple unmanned platforms to perform new functions on a larger scale by combining their capabilities in a coordinated manner.

This thesis addresses the emerging need of research related to fully autonomous cooperation between a heterogeneous team of vehicles, by taking a system level approach and integrating the necessary technologies. Software was developed and then tested that combines an unmanned ground vehicle and an unmanned aerial vehicle in order to perform a task that utilizes the strengths of each platform. The ground vehicle is programmed with a route for which it sends look-ahead waypoints to the aircraft. As it traverses the route, the aircraft searches for possible targets. If a target is detected, the approximate coordinates are sent over the network and the ground vehicle then further localizes and inspects the target. Once the inspection is completed, the ground vehicle continues on its previous route. This thesis demonstrates that pairing ground and aerial vehicles in a fully autonomous target localization problem can indeed provide a team functioning more efficiently than either alone.

Acknowledgments

I owe thanks to many people for their contributions to my graduate education and the work presented in this thesis. First off, I would like to thank my wife, Amy Van Covern, for her steadfast love and encouragement despite the long lab hours and the multiple lengthy trips away from home. Her support and understanding was greatly appreciated throughout my graduate studies. I also would like to thank my parents David and Wynne Ellen Van Covern. Their love and support throughout my life has provided me with the foundation necessary for my successes.

My advisors Dr. Reinholtz and Dr. Wicks have also contributed greatly to my graduate education. They have provided the leadership that has enabled the many research opportunities in intelligent robotics at Virginia Tech. The programs Dr. Reinholtz advised, and the friendly atmosphere he promoted, are what caught my eye as an undergraduate and convinced me to pursue a graduate degree. The outstanding unmanned vehicle program in the Mechanical Engineering department at Virginia Tech would not exist without his work and ability to motivate both undergraduate and graduate students.

I would also like to thank the great team of colleagues that I have had the opportunity to work with throughout my years in unmanned systems at Virginia Tech. The other members of the 'ONR Team', Cheryl Bauman, Ben Dingus, and Tom Alberi, were all helpful in laying down the groundwork for this thesis. Working closely with Cheryl during her thesis research also let me know part of what I had to look forward to one year later. Mike Avitabile and Ben Hastings are also to thank for sparking my interest in electronics development, which broadened my experience as a graduate student. Finally, the Urban Challenge team has been a major part of my graduate experience. Everyone involved with the project was a pleasure to work with. Working along side David Anderson, Patrick Currier, and Jesse Farmer was a great experience, as we converted the Ford Escape Hybrid to be a clean and reliable base vehicle for autonomous use. I owe the entire team of programmers and vehicle developers much thanks for all their hard work, allowing me to be a part of one of the top teams in the 2007 DARPA Urban Challenge.

Contents

Chapter 1: Introduction	1
1.1 Thesis Overview	1
1.2 Motivation	2
1.3 Example Mission Scenario	4
Chapter 2: Test Platforms	5
2.1 Unmanned Aerial Vehicle	5
2.1.1 Base Platform	6
2.1.2 Piccolo Autopilot	8
2.1.3 TASE Gimbal	9
2.1.4 PC-104 Computer	9
2.2 Unmanned Ground Vehicle	9
2.2.1 Base Vehicle	10
2.2.2 Drive-by-Wire Conversion	10
2.2.3 Sensor Array	11
2.2.4 Vehicle Interface Software	13
Chapter 3: Inter-Vehicle Communication	14
3.1 ITT Mesh Hardware	14
3.2 JAUS Messaging Standard	16
Chapter 4: Related Experimentation	18
4.1 Camp Roberts TNT Events	18
4.1.1 Surrogate UGV Demonstration	19
4.1.2 Autonomous Queuing	20
4.2 AUV Fest	23
Chapter 5: Software Algorithms	26

5.1	GPS Waypoint Navigation	26
5.1.1	Desired Speed Calculation	28
5.1.2	Desired Steering Angle Calculation	30
5.2	LIDAR Target Localization	31
5.3	J AUS Message Handling	35
5.4	Target Localization and Inspection	36
Chapter 6: Experimental Results		38
6.1	Individual Component Tests	38
6.1.1	Waypoint Navigation	39
6.1.2	LIDAR Target Detection	43
6.2	Fully Autonomous UGV Mission	45
Chapter 7: Conclusions		51
7.1	Future Work	52
References		54
Appendix A: Acronyms		56
Appendix B: Initial UAV Control Message		58
Appendix C: Custom JAUS Message		59
Appendix D: Fully Autonomous Target Localization Plots		61

List of Figures

2.1	UGV and UAV	6
2.2	VT Rascal Components	7
2.3	Cliff and Rocky	10
2.4	NovAtel SPAN System	12
2.5	SICK LIDAR Sensor	13
3.1	Mesh Network Example	16
4.1	Light Reconnaissance Vehicle	20
4.2	Surrogate UGV Results	21
4.3	Communication Flow Diagram	22
4.4	November TNT Results	23
4.5	AUV Fest Mission Diagram	25
5.1	Software Flow Chart	27
5.2	Rollover Free Body Diagram	28
5.3	LIDAR Polar Plot	33
5.4	Look-Ahead Example	35
5.5	Inspection Diagram	37
6.1	RDDF Waypoints	40
6.2	Waypoint Navigation Speed Tests	40
6.3	Waypoint Navigation LBO Tests	41
6.4	Waypoint Navigation LBO Closeup	42
6.5	Target Merge Failure	44
6.6	Example Target Setup	46
6.7	Circumnavigation Problem	47
6.8	Final Course 1 Example	48
6.9	Final Course 2 Example	49

6.10 Incorrect Target	50
D.1 Course 1 Plots	62
D.2 Course 2 Plots	63

List of Tables

- 3.1 JAUS Message Subset 17
- 5.1 LIDAR Correlation Coefficients 32
- 6.1 Average Lap Times 41
- 6.2 Scan Displacement 44
- C.1 Custom JAUS Message 60

Chapter 1

Introduction

The collaborative target localization and inspection process explained in this thesis was developed to demonstrate coordination among a heterogeneous team of autonomous vehicles. Building upon autonomous research platforms created at Virginia Tech, focus was placed on the design of software. The software would allow fully autonomous coordination between air and ground assets, taking advantage of the strengths of each. This chapter will give an overview of this thesis, followed by a discussion of the motivation behind the research. The example mission scenario that was chosen to demonstrate coordination is also included.

1.1 Thesis Overview

Unmanned vehicle development is an important area of research for industrial and military use. Chapter one discusses the motivation behind the research and includes a brief mission

scenario for which this thesis was based. To aid algorithm development and testing, a robust set of robotic vehicles is necessary. An overview of the Unmanned Aerial Vehicle (UAV) and Unmanned Ground Vehicle (UGV) test platforms used in this research are described in Chapter 2.

Chapter 3 focuses on the inter-vehicle communication, which is central to the elegant coordination between vehicles. In all coordinated use of autonomous vehicles, it is necessary to incorporate some form of data transfer between vehicles. This communication is what separates coordinated multiple vehicle autonomy from swarming robot applications. Chapter 5 explains the algorithms developed for the ground vehicle in order to navigate Global Positioning System (GPS) coordinates, detect, and localize targets. After developing these algorithms, it is necessary to prove their viability through testing. The software written for this thesis was tested in the field, and Chapter 6 provides the experimental results demonstrating the use of these algorithms to perform a cooperative reconnaissance mission. Conclusions and recommendations for future work are included in Chapter 7.

1.2 Motivation

The use of unmanned vehicles for military operations is an area of growing interest for the government. Removing the human from the vehicle in dangerous military situations will save lives. The significant defense funding for unmanned air and ground vehicles is evidence that these technologies are important to the military. The 2006 defense budget directly addresses this topic:

Unmanned vehicles are well suited to the dangerous missions that U.S. troops often undertake at great risk. The Department continues to make major investments in the development and procurement of unmanned ground vehi-

cles (UGVs), unmanned underwater vehicles (UUVs), unmanned aerial vehicles (UAVs), and unmanned combat aerial vehicles. DOD has only begun to exploit the potential of these vehicles. [1]

The budgeted \$1.9 billion for the development and procurement of UAVs and UGVs for fiscal year 2007 directly reflects the government's interest in this technology [2].

While the military currently makes use of some autonomous vehicles in their unmanned fleet, continued developments in autonomous behavior are necessary before autonomy will become prevalent. Cooperation between a heterogeneous team of autonomous unmanned vehicles is an even further step. Autonomous teaming has the potential to provide superior capabilities compared to existing technology by taking advantage of the strengths of each asset. For example, an aerial vehicle can be teamed with a ground vehicle to improve the overall sensing capabilities of the team. Small fixed wing aircraft can travel quickly over large distances, providing wide coverage. Challenges arise, however, in obtaining detailed information about a point of interest detected by the aircraft. Due to the limitation in minimum safe flight altitude, objects on the ground may only be detected as a few pixels in the plane's imagery. Also, uncertainties in an aircraft's attitude and position often translate to large uncertainties when localizing ground features. In contrast, ground vehicles have a greatly increased payload capacity, enabling more accurate sensing capabilities. However, ground vehicles are limited by the terrain they can traverse and their view can be more easily obstructed [3]. Noting these characteristics, the advantages of an air-ground collaboration become apparent.

This research, funded by the Office of Naval Research (ONR), is part of a larger grant with the goal of developing the theory and technology to improve the control and sensing of heterogeneous teams of autonomous robots [4]. The collection of Autonomous Aerial Vehicles (AAVs), Autonomous Ground Vehicles (AGVs), Autonomous Surface Vehicles (ASVs), and Autonomous Underwater Vehicles (AUVs) at Virginia Tech pro-

vides an opportunity for research in collaborative autonomy that will enhance the use of these technologies in military operations. The goal of this thesis is to demonstrate the potential advantages of using a heterogeneous team of autonomous vehicles to identify, localize, and inspect targets on the ground.

1.3 Example Mission Scenario

A military convoy plans to traverse a potentially dangerous section of roadway. To improve situational awareness, the route is loaded onto an UGV, which is sent out to autonomously traverse the route before the convoy. A small UAV is also deployed to aid with the reconnaissance. The UGV sends look-ahead GPS coordinates to the Unmanned Aerial Vehicle. The UAV circles these waypoints, looking for targets in the path of the ground vehicle. Once a target is detected and localized by the aircraft, the UAV broadcasts the target coordinates to the other assets on the network. Upon the receipt of a target message, the UGV re-plans its route toward the target and uses Light Detection and Ranging (LIDAR) and the UGV's GPS location to determine a more accurate location of the object of interest. Once the UGV localizes the target that was sent by the aircraft, it circumnavigates the object, obtaining surveillance video with a side-mounted camera. After the inspection is completed, the UGV continues on its previous route, repeating the process as necessary.

Chapter 2

Test Platforms

The vehicles used for the development and testing of the algorithms discussed in this thesis were both research platforms built at Virginia Tech. Both vehicles, shown in Figure 2.1, began as commercial base vehicles which were made autonomous by students at the university. The UAV was developed from the SIG Rascal platform and the UGV was developed from the Club Car XRT-1500. This chapter will first discuss the autonomous the aerial vehicle, followed by a discussion of the ground vehicle.

2.1 Unmanned Aerial Vehicle

The Rascal UAV was created by a team of researchers in the Aerospace Engineering department, lead by Dr. Craig Woolsey. The development of this vehicle was based closely off of a design implemented by researchers at the Naval Postgraduate School (NPS). In part, this is because in the early stages of the ONR funded research, the UGV team demon-



Figure 2.1: Virginia Tech's Rocky UGV (left) and Rascal UAV (right)

strated the ability to provide coordinated control of the NPS owned aircraft at Camp Roberts. An existing relationship with NPS, and their use of a Rascal as a reliable UAV provided the model for Dr. Woolsey's team to follow. This cooperation allowed the team to quickly produce a reliable autonomous aerial platform. The major components of this aircraft will be discussed in the following sub sections.

2.1.1 Base Platform

The UAV used by both NPS and Virginia Tech was based off of the SIG Rascal Remote Control (RC) model airplane kit. This airframe, with a wingspan of 110 inches, is one of the largest fixed wing hobby models on the market. The wing area of 1522 in² with a flat bottom design provides the lift and stability desirable for a UAV [5]. The large fuselage, allows adequate space for the hardware necessary for autonomous flight and research. A 1.6 cubic inch nitro fuel engine was incorporated into the design to provide the necessary thrust. This engine consumes an average of 1.3 oz/ min of fuel, so two 32 oz fuel tanks were included to provide approximately 50 minutes of flight time [6]. While a gas engine would provide the capability for longer flights, the team was afraid that the electrical noise and extra complexity of the spark system might increase the risk of failure. Figure 2.2 shows the VT Rascal and the major components added by Dr. Woolsey's research team.



Figure 2.2: The VT Rascal with major components highlighted

2.1.2 Piccolo Autopilot

Low level guidance and waypoint navigation for the Rascal UAV was provided by the Piccolo avionics system, manufactured by Cloud Cap Technologies. This complete avionics package includes the autopilot, flight sensors, navigation, wireless communication, and payload interface. The autopilot system collects data from three gyros, three accelerometers, both dynamic and barometric pressure sensors, an air temperature sensor, and a GPS receiver. Inertial sensors allow the built-in flight control algorithms to adapt well to a wide range of small aircraft, including the Rascal. Wind speed estimation is performed by comparing the actual air speed calculated by the pressure sensors to the ground speed calculated from the GPS data. This wind speed data is used to more accurately control the aircraft's flight. Ten servo outputs are provided, which send a Pulse Width Modulation (PWM) signal to the RC aircraft motors. Waypoint navigation on board the Piccolo uses the GPS position and speed to generate turn rates to follow the waypoints. If a single waypoint is given, or the last waypoint is reached, the autopilot commands the aircraft to circle this point with a user defined radius [7].

Communication with the autopilot is through the Cloud Cap ground station. This ground control station provides a command and control interface link to the avionics during flight. A 900 MHz frequency-hopping spread-spectrum link is used to provide a robust long-range method of communication. The ground station also has the capability of sending differential GPS corrections to the airplane to improve positional accuracy. A GPS receiver identical to the one in the aircraft is included with the base station to provide this functionality. In addition to the ground station, the Piccolo includes a serial port that can be used to send commands to the autopilot. This can be used either in conjunction with the wireless link or by itself [8].

2.1.3 TASE Gimbal

The Rascal UAV payload includes the TASE Gimbal camera system, also manufactured by Cloud Cap Technology. This small, inertially stabilized platform, reduces the effect of aircraft motion on the video image. A Sensor Point Of Interest (SPOI) mode provides the ability to train the camera on a fixed point while the aircraft is in motion. When using this mode, a GPS point is passed to the Gimbal which then calculates the pan and tilt to achieve the desired field of view. Measuring only 4.4 inches in diameter and having a weight of less than two pounds, the TASE turret can be integrated well into small UAVs [9]. An analog video link provides the images for processing on-board the Rascal's PC-104 computer.

2.1.4 PC-104 Computer

The computer aboard the Rascal uses PC-104 technology. This computer handles all of the advanced navigation, image processing, and communication. The PC-104 form factor was chosen due to its compact size and ability to integrate easily with standard computing hardware such as the PC card. A 1.4 GHz processor with 1 Gb of RAM was chosen to provide the necessary processing power [6]. The machine runs Windows XP for its Operating System (OS) in order to be compatible with the communication drivers that are used. A 8 GB compact flash solid state storage drive is used in place of a hard disk because of its improved vibration resistance. This flash drive contains the OS, software, and data logs.

2.2 Unmanned Ground Vehicle

Virginia Tech's off road UGV, Rocky, was one of two similar platforms originally developed for the Defense Advanced Research Projects Agency (DARPA) Grand Challenge.

It was designed to traverse up to 150 miles through a desert environment, avoiding k-barriers, tank traps, and natural obstacles along the way. Rocky, and his predecessor, Cliff are shown in Figure 2.3. This section provides a description of the vehicle and its conversion to autonomy.



Figure 2.3: The Virginia Tech DARPA Grand Challenge entries Cliff (left) and Rocky (right).

2.2.1 Base Vehicle

Rocky was created from a diesel Club Car XRT-1500. This base platform was chosen due to its small turning radius and off-road capabilities. The short wheel base and tight turning ability makes this vehicle well suited for obstacle avoidance and reactive control. IntelliTrack automatic, full-time, all-wheel-drive on the XRT platform enhanced the off road performance and eliminated the need for automated gear shifting. The large payload capacity of approximately 800 lbs and spacious bed area allowed for easy addition of sensors and computers [10].

2.2.2 Drive-by-Wire Conversion

The XRT 1500 platform was converted to drive-by-wire by adding actuators for throttle, steering, and brake control. To actuate the throttle, a motor and controller from

QuickSilver Controls is used. This motor, coupled through a 25 to 1 gear reduction, drives a pulley which pulls the throttle cable. Steering control is achieved by coupling a larger QuickSilver motor with a 5 to 1 gear head to the rack and pinion input using a roller chain coupling. The QuickSilver products combine high torque microstep motors with servo style controllers resulting in a relatively inexpensive solution for the high torque output and precise control capability [11]. A RS-485 serial connection to the motor controllers offers a simple absolute position command interface. A Hydraforce proportional valve controller combined with a Carlisle Hydrostar hydraulic brake actuator provides the interface needed to actuate the four wheel disk brake system. The proportional valve was controlled by supplying an analog voltage of 0 to 10 VDC via National Instruments Data Acquisition (DAQ) hardware. For emergency situations, a redundant method of applying brakes was also incorporated into the vehicle. This emergency system was designed to be a zero power state, fail safe solution. A pneumatic cylinder was arranged so that when the unit was filled with air, a spring was compressed. When the air was released, the spring and piston would pull on a cable, manually applying the rear brakes. Control of this emergency brake was arranged so that when power to a pneumatic solenoid was cut off, the air would be released and brakes would be applied. A remote relay system was also included in the emergency stop system to allow the vehicle to be disabled wirelessly.

2.2.3 Sensor Array

In contrast to the UAV, the payload capacity of the UGV did not limit the computing and sensor selection. Rocky is equipped with sensors for localization and obstacle detection. For this particular target detection and localization research, a digital video camera was also attached to the UGV.

A NovAtel SPAN Inertial Navigation System (INS) was used for localization. The components of this system are pictured in Figure 2.4. The SPAN system utilizes a NovAtel

ProPak-LBplus GPS receiver and a precision Honeywell HG1700 Inertial Measurement Unit (IMU). The ProPak accepts differential corrections from several different sources to increase absolute positional accuracy. The service used for this vehicle is Omnistar HP, which under optimal conditions will provide accuracy to 0.1 m Circular Error Probable (CEP). The addition of the IMU to the GPS system provides the user with a high-accuracy position estimate even at times when GPS satellites may become obstructed. The SPAN system uses a Kalman filter to model system error and provide the user with the most accurate position, velocity, and attitude possible. This information is sent to the control computer over a RS-232 serial connection [12].



Figure 2.4: The NovAtel SPAN inertial navigation system.

For obstacle detection, a SICK LMS-220 LIDAR unit is used (Figure 2.5). This sensor provides distances to objects in a fan-shaped scan plane. The technology is based on time-of-flight measurements of light. Extremely short pulses of light from an infrared laser are reflected off of a rotating mirror inside the unit. The light is then reflected off of objects in its path and back into the scanner where it is detected and time-of-flight is calculated. Distance to the object is directly proportional to the time between the transmit and received pulses. The LMS-220 has a 180 degree field of view, with an angular resolution of 0.5 degrees. Distances can be reported at a frequency of 75 Hz, with 10 mm resolution [13].

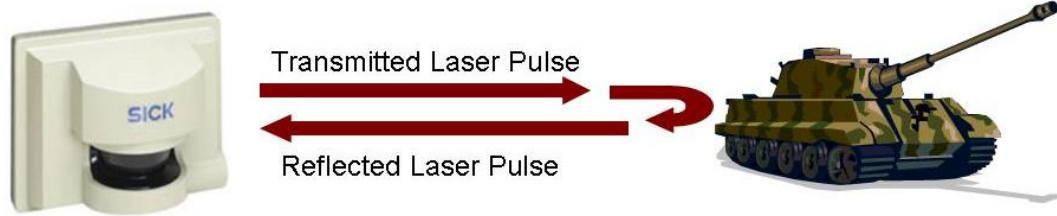


Figure 2.5: The SICK LIDAR sensor uses a time of flight calculation on infrared laser pulses to determine distance to objects.

2.2.4 Vehicle Interface Software

To translate the desired speed and steering angle to actuator commands, a vehicle interface program was written to run on Rocky's on-board computer. Speed is controlled by separate throttle and brake controllers. Precautions were taken to avoid commanding both throttle and brake simultaneously. Throttle is controlled using a closed loop Proportional Integral Derivative (PID) controller. The speed feedback is input to the controller from the INS, eliminating the need for the addition of an extra feedback device such as wheel encoders. The output of the PID controller is converted to a serial position command which is sent to the QuickSilver throttle controller. The brakes are controlled using a piecewise function, applying brakes at different rates in an open loop manner, depending on the desired deceleration. Steering control is achieved by directly translating the desired steering angle command to a motor encoder count value and sending it to the steering motor controller. Rate control was not included into the design because it was not necessary in the reactive algorithms that were used for the Grand Challenge event. Before commanding throttle or steering, safety checks are performed. A simple rigid body rollover prevention calculation was used to avoid dangerous operation. This check was given the authority to override steering and or throttle commands.

Chapter 3

Inter-Vehicle Communication

An area of interest in any autonomous cooperation discussion is the means for inter-vehicle communication. The wireless hardware and communication protocol chosen has a direct impact on the capabilities of the vehicle network. For the research presented in this thesis, the ITT Mesh Networking hardware was selected for the physical layer and the Joint Architecture for Unmanned Systems (JAUS) was used as the specification for the data format and methods of communication. This chapter will describe both technologies, including a discussion of why they were selected.

3.1 ITT Mesh Hardware

Mesh networking is a communications method in which multiple messaging paths can exist between any two nodes. It is designed to be capable of dynamically adjusting to broken paths and changing routes. The ITT mesh network was originally developed to

satisfy a DARPA proposal to create an ad hoc wireless network that supported broadband data rates, end-to-end IP support, voice and video capability, built-in position localization (without GPS), and support for vehicular mobility up to 250 MPH [14]. ITT Industries was the only contractor that developed hardware that was able to meet all the requirements. This technology was later made commercially available by Motorola.

The mesh technology allows high data rate communication over long ranges and communication in non line of sight environments by the ability for communication to 'hop' through other nodes in the network. An example ad hock mesh network is shown in Figure 3.1. Like the 802.11 wireless standard, the mesh hardware uses the license-free 2.4 GHz ISM band. An important difference is the ITT hardware uses the Quadrature Division Multiple Access (QDMA) radio. The QDMA radio differs from 802.11 radio platforms in that it has much higher immunity to interference [15]. The 802.11 radio protocol was originally developed to be a low cost, high bandwidth, replacement for a Local Area Network (LAN) cable in indoor networks. The 802.11 technology does not, however, scale well to wide area mobile applications.

The ITT Mesh hardware selected for use in the air-ground cooperation was the WMC 6300 Wireless Modem PC card. This was originally chosen because it was already being used by the Naval Postgraduate School. Rocky needed to use this technology to communicate with the NPS vehicles. However, once the Virginia Tech team adopted the technology, many of its qualities were appreciated. This technology also allows future expansion to larger teaming operations without changing the communication hardware. To integrate the PC card into the Rascal UAV, a PC card adapter module was included into the PC-104. On Rocky, a laptop with a PC card slot was used.

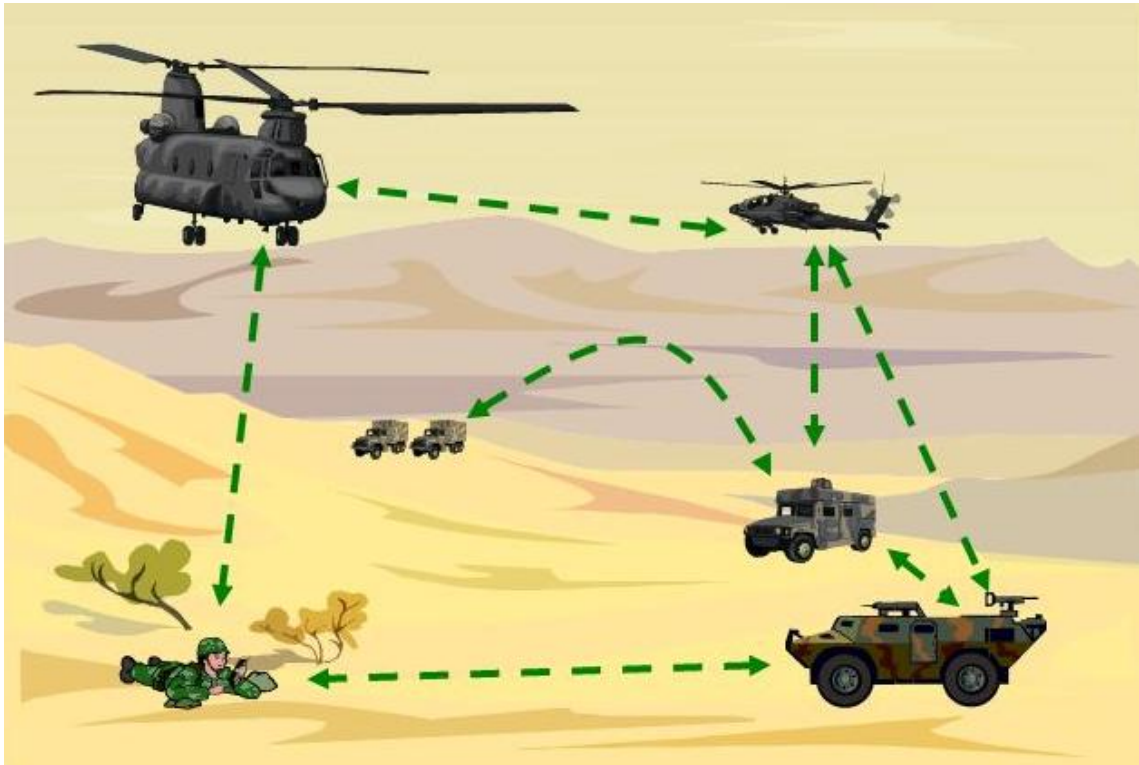


Figure 3.1: An example Mesh network. Dotted lines show possible communications paths.

3.2 JAUS Messaging Standard

Joint Architecture for Unmanned Systems (JAUS) is an architecture for the interfaces in the field of unmanned vehicles. Messages are defined that are independent of hardware or vehicle platform to promote usability in all unmanned systems applications. Having a domain-level messaging protocol, such as JAUS, allows simple interoperability between unmanned systems developed separately. This encourages the re-use of components which leads to faster, more cost effective development.

The decision to use JAUS as the messaging protocol for the work described in this thesis was partially due to the experience of others at Virginia Tech. Tools created by these researchers, such as the JAUS Toolkit for LabVIEW, also made integration of this standard much easier [16]. The movement toward the adoption of JAUS by various military

organizations was also motivation to use the standard. Use of the architecture is mandated for all programs in the Joint Robotics Ground Enterprise (JGRE) [17]. The collaboration with the Naval Post Graduate School also lead to the toward a common use of JAUS.

For the cooperative teaming exercise, the researchers at NPS and Virginia Tech agreed on a subset of the JAUS messages that would allow for all necessary communication between subsystems. While not fully JAUS interoperable, this was seen as a stepping point toward the full integration of JAUS in future collaboration between the two schools. The JAUS messages that were used are included in Table 3.1. This table also includes which nodes were receiving and sending each message. An experimental message was also defined to meet the need for reporting a target. This message used the same structure as the other JAUS messages; however, the message body was customized. The definition of this custom message is included in Appendix C.

Command Code	Description	UAV	UGV
0x040C	Set Global Waypoint	RX	TX
0x4402	Report Global Pose	TX	TX
0xD802 *	Report Target Location	TX	RX
* The report target message was an experimental message not defined by the JAUS standard			

Table 3.1: The JAUS messages used and the subsystems sending or receiving the message.

Chapter 4

Related Experimentation

Through the course of the ONR funded research, several experiments were performed that lead to the development of the algorithms presented in this thesis. Two experiments with the Naval Postgraduate School at Camp Roberts in California and one demonstration at AUV Fest in Panama City, Florida provided the opportunity to explore the topic of heterogeneous teaming of autonomous vehicles. This chapter will discuss these three experiments.

4.1 Camp Roberts TNT Events

The collaboration with the researchers at the Naval Postgraduate School resulted in two joint experiments at Camp Roberts as part of the Tactical Network Topology (TNT) quarterly exercises. The first experiment occurred in August of 2006. This experiment tested the mobile control of the NPS Rascal UAV using a surrogate UGV. The second experi-

ment at Camp Roberts in November 2006, built upon the first. This experiment added the Virginia Tech UGV, Rocky, to provide a fully autonomous ground vehicle. These joint TNT experiments with the Naval Postgraduate School will be discussed further in the following sections.

4.1.1 Surrogate UGV Demonstration

As an effort to begin a collaborative relationship with the Naval Postgraduate School, a team of researchers from Virginia Tech traveled to Monterey, California to participate in the August 2006 TNT experiments. For this initial exercise, it was decided not to ship the UGV across the country but to bring only the necessary equipment and software from the UGV to outfit a NPS owned Light Reconnaissance Vehicle (LRV) for the mission. The equipment that was added to the LRV included the NovAtel SPAN system, a notebook computer, and the ITT Mesh communication hardware. Figure 4.1 shows the team instrumenting the LRV for the experiment. The LRV would serve as a surrogate UGV, commanding the aircraft as the UGV would, however, not operating autonomously.

The communications protocol used was a User Datagram Protocol (UDP) based messaging scheme developed by NPS. This scheme was designed specifically for the control of the Rascal UAV. Appendix B includes a table describing the message format. This messaging scheme was used because it had previously been implemented by NPS and they had not yet migrated to JAUS. Using this protocol and the NovAtel INS, the team sent the current location of the LRV to the UAV as its target point. In order to fluently control the aircraft, this waypoint was updated once per second over the ITT Mesh network. Telemetry data was also sent via the aircraft to the other mesh nodes in the network. This data was read, parsed, and logged aboard the LRV for post experiment analysis.

The software running on the notebook computer inside the LRV interfaced with the INS and sent and received all messages over the Mesh network. Using this software in



Figure 4.1: The NPS owned Light Reconnaissance Vehicle being outfitted with NovAtel INS.

conjunction with the software onboard the Rascal, the UAV successfully circled the LRV and maintained the vehicle in its camera frame at ground speeds of up to 25 MPH through the desert. A graphical representation of the experiment and its results is included as Figure 4.2. This initial experimentation with mobile real-time control of the UAV proved to be successful and demonstrated the feasibility of future autonomous coordination between the two research teams. After this experiment, immediate discussion began about how it could be expanded for the November TNT experiment.

4.1.2 Autonomous Queuing

For the November 2006 TNT experiment, it was decided to incorporate the Virginia Tech UGV, Rocky, into the exercise. For this mission, the UGV would send look-ahead coordinates to the UAV. The UAV would then survey the path ahead of the ground vehicle. Using

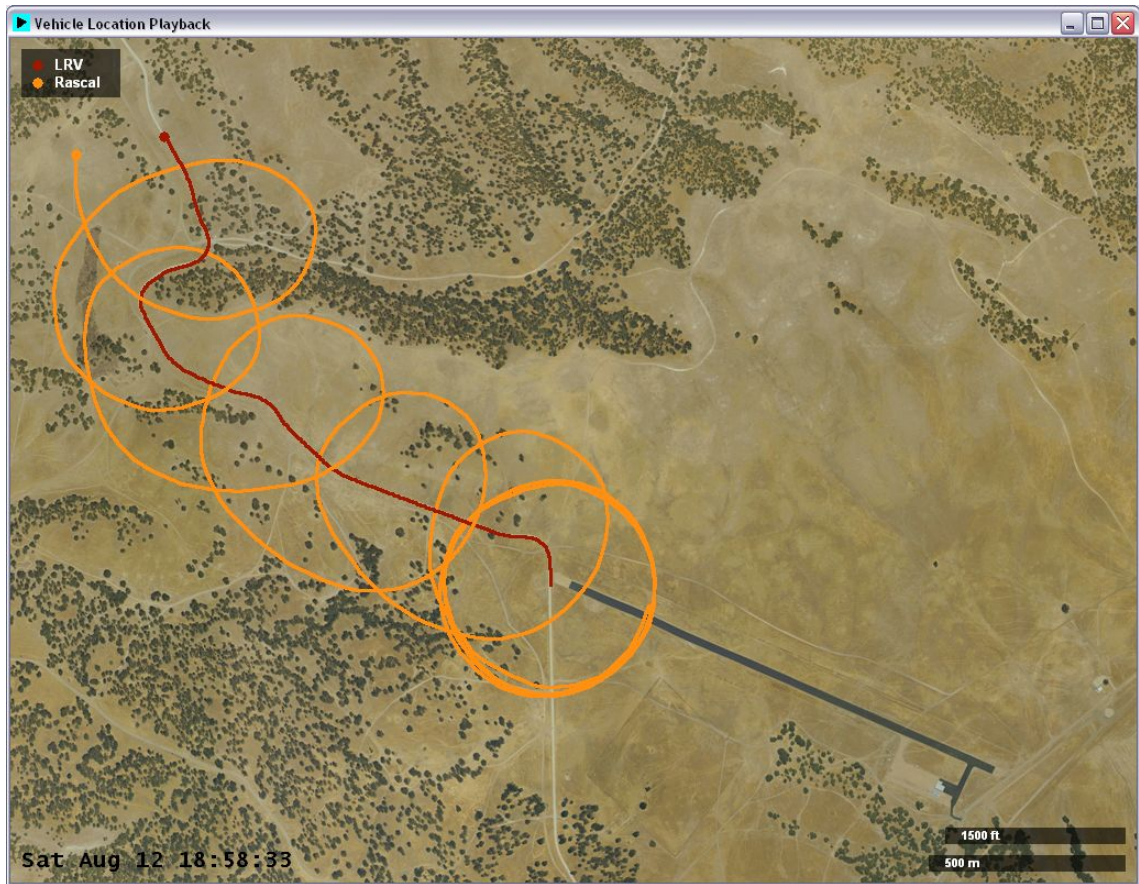


Figure 4.2: An aerial overlay of the surrogate UGV experiment results [USGS].

the Perspective View Nascent Technology (PVNT) software, developed by Wolfgang Baer of NPS, an operator at the ground station could identify the location of a target detected in the aircraft's imagery with an accuracy of approximately one meter [18]. Once detected, the target coordinates would be sent over the mesh network and received by the ground vehicle. The UGV would then drive toward the target and traverse a set of waypoints, circumnavigating the target. A side-mounted camera recorded video of the target for later analysis. A diagram showing the communication flow for this exercise is included as Figure 4.3.

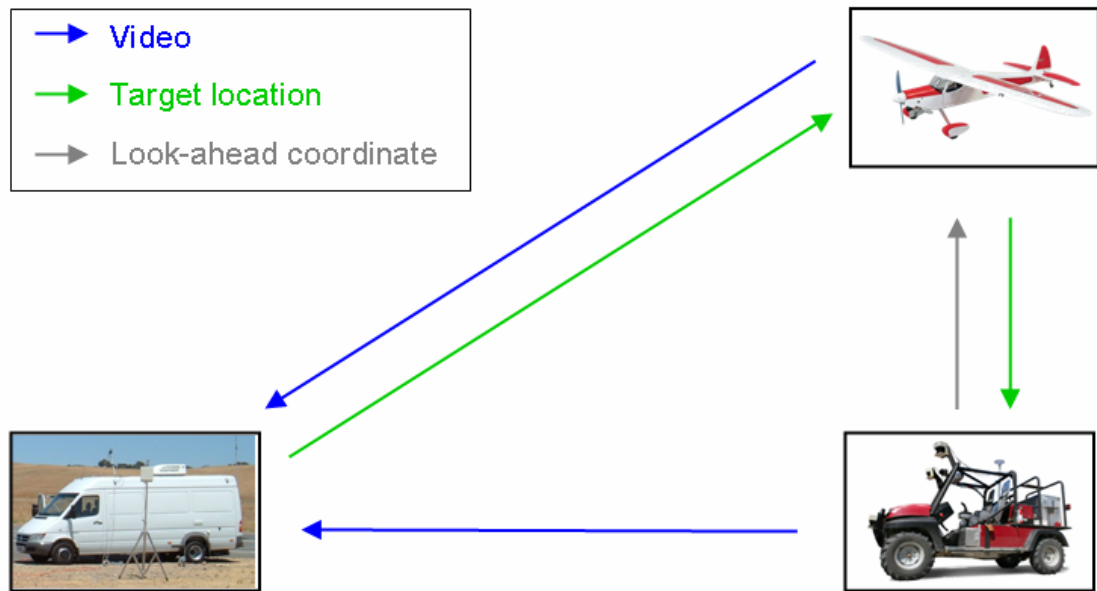


Figure 4.3: The flow of communications for the November TNT exercise.

At the day of the test, it was noticed that the communications between vehicles was not working properly. However, after modifying the software, the test was a success. The PVNT operator was able to identify and localize a white vehicle parked slightly off of the ground vehicle's path using the UAV's imagery. The UGV then approached and circled the target and returned to its previous path. A plot of the vehicle paths for this test is

included as Figure 4.4. Unfortunately, due to time constraints at the airfield, only one test was performed. Even though only a short experiment took place, it was believed that the experiment could have been scaled to a much larger course such as the route followed in the August experiment. The lost time due to messaging format confusion was one of the reasons why both NPS and Virginia Tech decided to move to the JAUS messaging standard for further work.

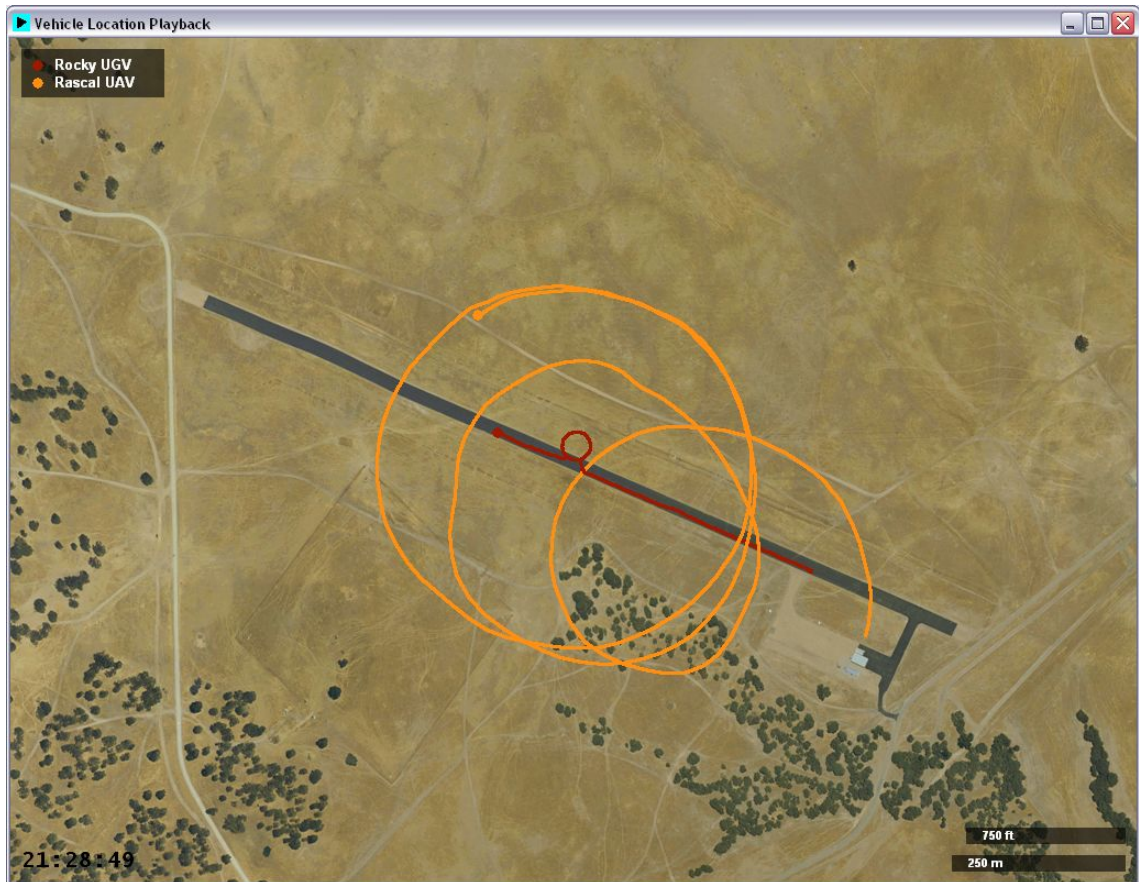


Figure 4.4: An aerial overlay of the results from the November TNT exercise [USGS].

4.2 AUV Fest

To display a subset of the results from the ONR funded research at Virginia Tech to others in the Unmanned Systems community, a demonstration was planned for the 2007 AUV

Fest at Naval Surface Warfare Center (NSWC) in Panama City, Florida. AUV Fest is a demonstration of Autonomous Underwater Vehicles as well as AAVs, AGVs and ASVs and is sponsored by ONR approximately every 18 months [19]. The heterogeneous teaming exercise with NPS was used as a model for the Virginia Tech demonstration in this event. For AUV Fest, the mission demonstrated in the November 2006 TNT event was migrated to a riverine scenario. For this activity, an ASV built in Dr. Stilwell's Autonomous Systems and Controls Laboratory at Virginia Tech replaced the roll of the UGV in the previous experiment that took place at Camp Roberts. Also, the Virginia Tech Rascal participated instead of the NPS aircraft. The UGV was incorporated as a shoreline asset for this demonstration. During the experiment, it loitered on the shore until the target location was reported by the aircraft. At that time, the UGV oriented itself and its camera toward the target. During the entire operation, a NPS owned Scan Eagle UAV circled above, recording video and serving as an additional repeater in the communication network. A diagram of the operation is included as Figure 4.5. In place of the PVNT software used to identify the target in the November experiment, The UAV used a hue based vision algorithm to detect the target. The position and attitude of the aircraft as well as the camera orientation were used to calculate the target's position with an accuracy around 20 m. This software allowed the experiment to be fully autonomous, removing the human-in-the-loop scenario from the previous experiments.

The work presented in this thesis builds upon the AUV Fest demonstration, however, the UGV replaces the USV for the over-land testing in Blacksburg. The ability to improve the UAV's localization estimate using ground vehicle based LIDAR is also an addition to this work, but the same could easily be migrated back to a riverine environment if necessary. The remainder of this thesis will explain the algorithms that were created and the experimental results from testing the software. Also, conclusions and suggestions for future work will be included.



Figure 4.5: Aerial imagery of the operation area with vehicle mission diagrams overlaid [USGS].

Chapter 5

Software Algorithms

The software that was created for the UGV to perform the test scenario described in Section 1.3 was developed using National Instruments LabVIEW. This graphical code environment allows quick software development and simple interfacing with measurement and control hardware. Rocky's software is comprised of GPS waypoint navigation, LIDAR target detection, JAUS message handling, and target localization and inspection components. An upper level flow chart describing the overall software structure is included as Figure 5.1. The software is arranged using a state machine architecture. This chapter will describe the algorithms behind this software.

5.1 GPS Waypoint Navigation

In it's most basic form, GPS waypoint navigation simply changes the vehicle heading so that it is oriented toward the target waypoint while throttle is applied to achieve a prede-

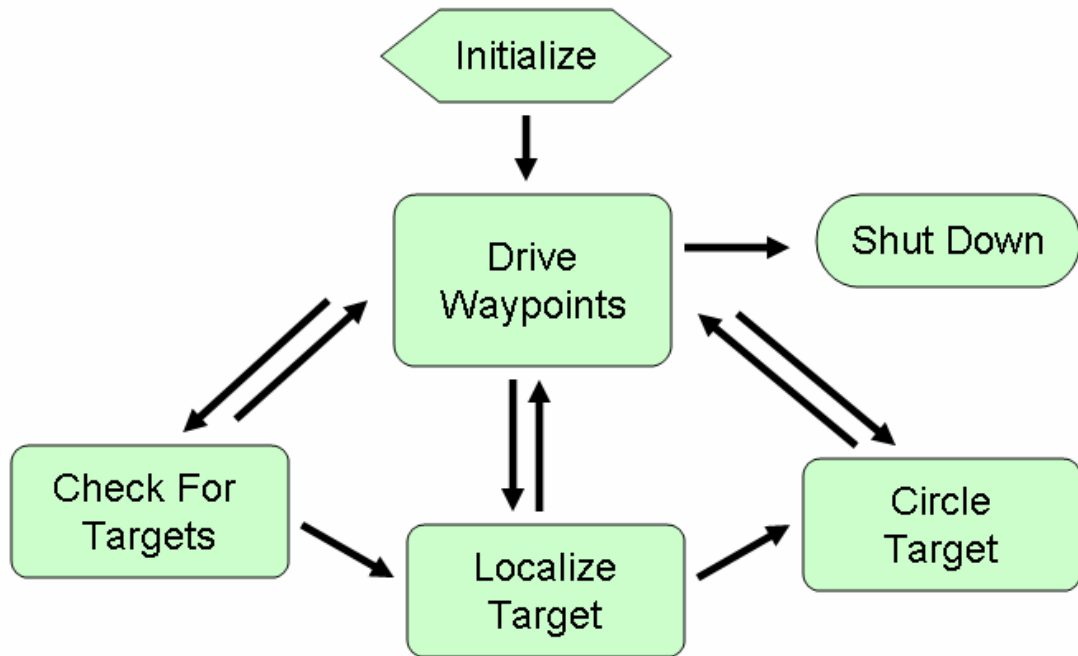


Figure 5.1: The upper level software flow for this thesis.

terminated speed. The navigation algorithms that are presented in this thesis not only include this basic requirement but also address acceleration control and rollover protection. These additions enable the vehicle to operate more smoothly and safely. The software builds upon previous work and includes some of the algorithms originally developed for the DARPA Grand Challenge. When creating a path for the vehicle to follow, a series of GPS waypoints are used to define points that the vehicle should pass through. The series of waypoints are loaded into the software using the DARPA Route Data Definition File (RDDF) format. This format, includes a comma delimited list of waypoint Latitude and Longitude, followed by the Lateral Boundary Offset (LBO) and speed limit [20]. Once parsed from the input file, these waypoints are translated to Northing and Easting coordinates for the particular UTM zone. This conversion is performed because distance calculations are simpler in a rectangular coordinate frame than the World Geodetic System (WGS) coordinate frame.

5.1.1 Desired Speed Calculation

When determining the speed to command, four speed limiting factors are compared and the most conservative is chosen. The first of these factors is a limit based on the current steering angle to prevent rollover. This rollover prevention was developed for the Grand Challenge by Joe Putney and Cheryl Bauman. This factor limits the maximum safe speed to drive the particular turning radius with the current vehicle roll. A rigid body model, shown in Figure 5.2 was used for this calculation. The torques about the potential roll edge were summed and then the equation was solved for the velocity from the centripetal force component:

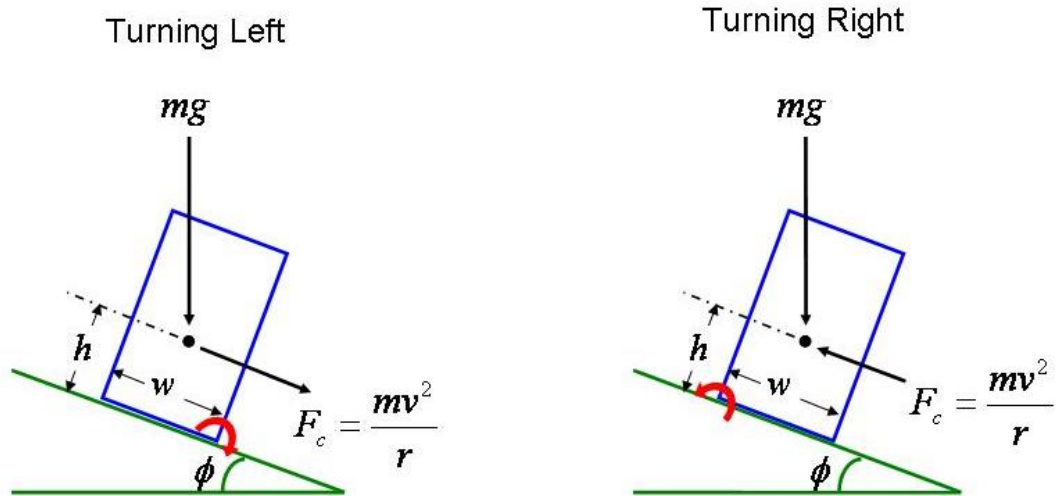


Figure 5.2: The free body diagram used to calculate the rollover speed.

<p style="text-align: center;">Turning Left</p> $\Sigma T = F_c h + mg \sin(\phi) h - mg \cos(\phi) w / 2$ $v^2 = rg \left(\frac{w/2}{h} * \cos(\phi) - \sin(\phi) \right)$	<p style="text-align: center;">Turning Right</p> $\Sigma T = F_c h + mg \sin(\phi) h + mg \cos(\phi) w / 2$ $v^2 = rg \left(\frac{w/2}{h} * \cos(\phi) + \sin(\phi) \right)$
(5.1)	

This speed was then reduced by a factor of safety of three to account for the rigid body assumption and to ensure that the UGV did not ever come close to rolling. The second speed limiter was the speed limit from the RDDF. This speed limit is set for each segment of the course and is chosen as the maximum speed the vehicle should travel for a particular section of the RDDF. Third, was a speed restriction based on the distance to the next waypoint and the turning angle that will be required once that waypoint is achieved. This ensures that the vehicle will decelerate to the safe turning speed before reaching a sharp turn. Because the reactive navigation approach is constantly re-calculating these parameters, a non-time-based equation relating the speed to the next waypoint distance was necessary. This equation was derived from the equation of motion:

$$v_f = v + a\Delta t, \quad (5.2)$$

and the definition of average velocity:

$$\bar{v} = \frac{d}{\Delta t}. \quad (5.3)$$

expanding the average velocity equation to include v_f and v yields,

$$\frac{1}{2}(v_f + v) = \frac{d}{\Delta t}. \quad (5.4)$$

Solving for Δt in equations 5.2 and 5.4, and combining these equations, a position based velocity solution can be obtained:

$$v_f^2 - v^2 = 2ad. \quad (5.5)$$

This equation can then be solved for the velocity limit that should be applied to achieve the desired deceleration for any position ahead of a turn. Finally, the last method that is used as a limit for the commanded speed is a manual speed setting on the user interface. This speed is often useful when the user wants to limit speed for debugging purposes.

Once a speed limit is determined it is passed through an acceleration limiter before it is output to the vehicle interface driver. This algorithm compares the desired speed to the current speed and determines whether the change in speed between the two, divided by the time between commanded speed iterations, would exceed the acceleration limit. If this was the case, the commanded speed in m/s, v_c , would be set using the equation:

$$v_c = a_{max} * \Delta t + v \quad (5.6)$$

where a_{max} is the acceleration limit, Δt is the cycle time, and v is the current vehicle speed. This final limit on acceleration, allows the UGV to perform more like a human driver, preventing unexpected lurching.

5.1.2 Desired Steering Angle Calculation

Along with the speed, the steering angle must also be sent to the vehicle interface driver. The steering angle is calculated by determining the angle between the current vehicle heading and the heading of the target waypoint:

$$\theta = atan2\left(\frac{N_v - N_w}{E_v - E_w}\right) - h_v \quad (5.7)$$

where $atan2$ is the signed arctangent function, subscript v denotes the vehicle coordinates, and subscript w denotes the waypoint. This angle is then limited by the available steering angles for the particular vehicle to create the ideal steering angle. At this point, rollover is once again considered and the steering angle may be limited based on the current speed

if necessary to prevent the vehicle from rolling. Referring back to Figure 5.2 and Equation 5.1, the turn radius that would cause roll can be calculated:

$$\left| \begin{array}{c} \text{Turning Left} \\ r_{left} = \frac{v^2}{g(w/2/h*\cos(\phi)-\sin(\phi))} \end{array} \right| \left| \begin{array}{c} \text{Turning Right} \\ r_{right} = \frac{v^2}{g(w/2/h*\cos(\phi)+\sin(\phi))} \end{array} \right| \quad (5.8)$$

from this turning radius, the steering angle can be calculated:

$$\left| \begin{array}{c} \text{Turning Left} \\ \theta_{left} = \text{atan}\left(\frac{w}{r_{left}}\right) \end{array} \right| \left| \begin{array}{c} \text{Turning Right} \\ \theta_{right} = -\text{atan}\left(\frac{w}{r_{right}}\right) \end{array} \right| \quad (5.9)$$

A factor of safety of two is then applied to this angle. If the ideal steering angle is less than this value then the ideal value is sent to the vehicle interface driver, otherwise the safe value is used.

5.2 LIDAR Target Localization

The LIDAR target detection software component uses one SICK LMS-221 to localize potential targets. This component also calculates a confidence indicating how well the set of points corresponds to a target. Scan data is continually received via a RS-422 connection to the sensor. This data is temporarily stored as an array of distance measurements with the corresponding measurement angles. Because small objects in the operating environment such as blades of grass or insects often result in close single or double returns in the scan data, these points are first filtered out of the array.

The target localization algorithms discussed in this thesis is not to be mistaken for an obstacle avoidance component. This algorithm differs by looking for only certain classifications of objects. When choosing the method for detecting individual targets from

LIDAR data, some assumptions about the target needed to be made. It was assumed that the targets had four flat sides. This assumption, applies well to many common objects of interest such as vehicles or small buildings. To determine what returns correspond to targets, the scan data was first segmented into discrete objects by looking for large gradients between sequential points. An abrupt change in distance measurements signifies a background-object transition. Once the scan data was divided, the segments needed to be summarized for further analysis. For each segment, a set of three defining contour points was created. These contour points consisted of the leftmost, closest, and rightmost scan points of the segment. According to the previous assumption, lines connecting these points would correspond to the two visible sides of the object. Figure 5.3 shows a sample LIDAR scan with lines connecting contour points for each object. The correlation between the actual scan data and the linear point-to-point estimation was calculated as the correlation coefficients, ρ , for both the left and right side of the object. Finally, the number of scan points on each side of an object was used as a weighting to combine the square of both side's correlation coefficients into a number representing how well the particular object fit the target shape assumption. The correlation coefficients for Figure 5.3 are included as Table 5.1. To remove the problems caused by attempting to correlate only one or two points to a line, objects in which only one side had more than two returns used the squared correlation coefficient for that side only.

Right Side		Left Side		Entire Object	
ρ	# points	ρ	# points	weighted ρ^2	# points
0.68807	15	---	0	0.47344	15
0.99998	59	-0.9995	16	0.99974	75
0.45583	17	-0.46494	19	0.21221	36
0.72004	4	-0.5006	14	0.31013	18

Table 5.1: Correlation coefficients, ρ , for the objects in the example LIDAR plot (objects sorted left to right).

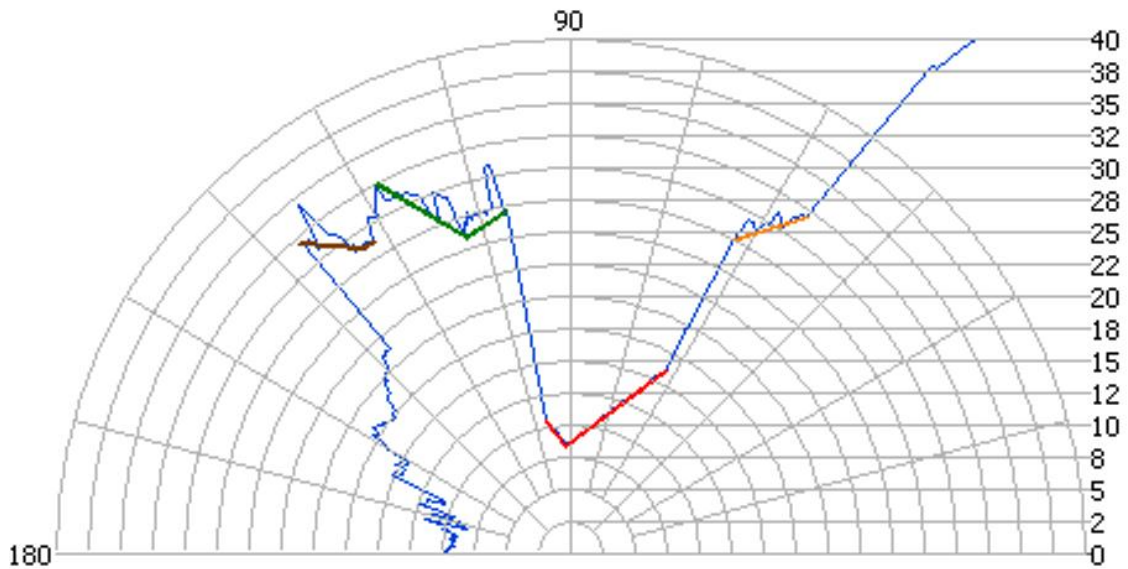


Figure 5.3: An example LIDAR polar plot showing contour points connected.

Because the targets that are to be localized are large, the object array is filtered by removing objects that have less than 8 points. This corresponds to objects smaller than 1.25 m at a distance of 20 m. Filtering the data in this manner further strengthens the assumption that the segments of scan data with the highest correlation coefficients are possible targets. For the remaining scan segments, the object center is calculated as a representation of the location of the target. This center is calculated by determining the midpoint of the line connecting the leftmost and rightmost points of the segment. This method works well for finding the center of parallelograms if two sides are detected. However, if only one side of the target is seen, the center will be the midpoint of the side detected, because no depth information is available. Detecting the edge of the object will most often not be a problem, as the target will still be inspected, just not properly localized.

After individual targets are defined, the GPS position of each possible target is determined. To calculate the target Northing, N_t , and Easting, E_t , the equations

$$N_t = N_v + N_{vl} + N_{lt} \quad (5.10)$$

$$E_t = E_v + E_{vl} + E_{lt} \quad (5.11)$$

were used where subscript v , denotes vehicle location, vl , denotes the vehicle to LIDAR scanner transform, and lt denotes the LIDAR scan output transform. The vehicle coordinate frame to LIDAR scanner transformation is necessary due to the offset between the GPS antenna and the scanner. Using basic trigonometry, N_{vl} and E_{vl} are calculated,

$$N_{vl} = x * \cos(\phi) + y * \sin(-\phi) \quad (5.12)$$

$$E_{vl} = x * \sin(\phi) + y * \cos(-\phi) \quad (5.13)$$

where x and y are the offsets in the vehicle coordinate frame and ϕ is the vehicle heading. Transforming the location of the target based on the sensor data can be accomplished in a similar manner. The target to LIDAR sensor transforms, N_{lt} and E_{lt} are calculated,

$$N_{lt} = d * \sin(\theta - \phi - \gamma) \quad (5.14)$$

$$E_{lt} = d * \cos(\theta - \phi - \gamma) \quad (5.15)$$

where d is the reported distance to target, θ is the corresponding scan angle, and γ is the angular offset of the scanner relative to the vehicle frame. Using this series of equations, the GPS location of the target is calculated to obtain a more accurate location than the aircraft alone can provide.

5.3 JAUS Message Handling

A separate thread running on the Navigation computer manages all JAUS communication. This module constantly reads the UDP buffer for JAUS messages then parses and stores the data for use by the other components. The custom Report Target Location message discussed in Section 3.2 is parsed in this module, storing all unique sets of target coordinates in an array for use by the target localization and inspection routine. Upon the receipt of each target, the target ID is compared to preexisting targets to avoid duplication. The JAUS message handler also is used to periodically send the UGV's position with the Report Global Pose Message, and send the aircraft a look-ahead position via the Set Global Waypoint message. The look-ahead waypoint is calculated by performing a linear interpolation between points on the RDDF to determine a position 200 m ahead in the path of the ground vehicle. Figure 5.4 shows how the UGV would calculate the look ahead waypoint for an example set of waypoints.

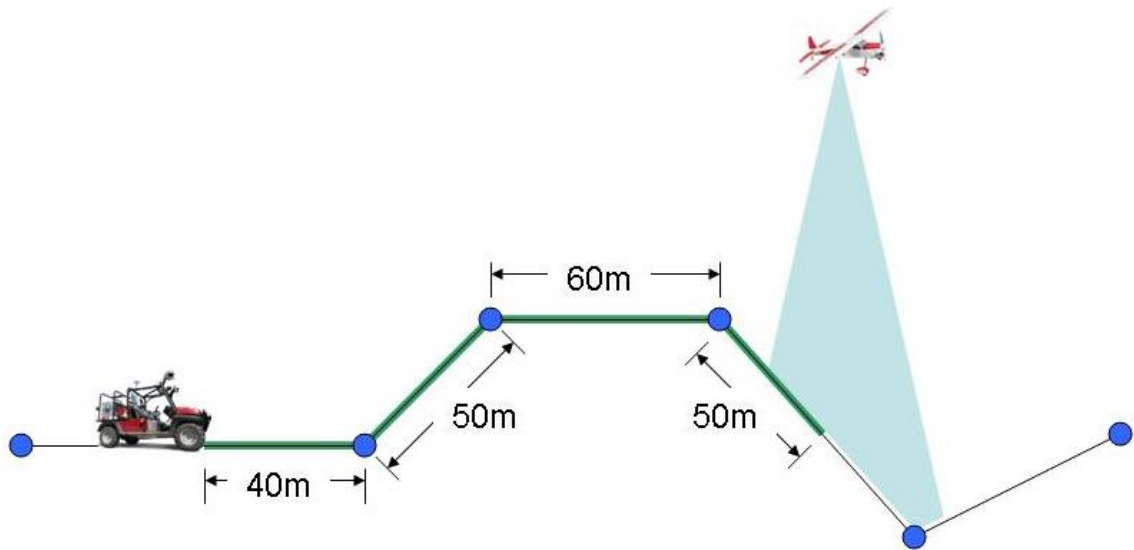


Figure 5.4: An example of how the look ahead waypoint is calculated.

5.4 Target Localization and Inspection

The target localization and inspection component of the software is the primary structure that enables the coordinated mission described in Section 1.3. While the vehicle is driving a list of waypoints, it is also comparing its location to a list of targets that have been sent over the network via a JAUS message. Figure 5.5 shows the planned localization and inspection behavior, highlighting the inspection parameters. Once the vehicle comes within the user defined approach radius of a target, a temporary path will be generated to guide the vehicle toward the target. At this point, the localize target routine is activated. This module reads the output from the LIDAR target detection component and filters the possible targets based on the error radius of the UAV's target estimate. The remaining objects are then sorted by the correlation coefficient and the vehicle is directed toward the location of the best candidate target. To reduce the effect of false estimates, a rolling average was used. This approach uses previous estimates of the target location to limit the rate of change of the vehicle's path. The localization process is repeated, further improving the estimate of the target's exact location. Once the UGV comes within the inspection radius of the target, a new set of temporary waypoints are generated to guide the ground vehicle in a circle around the target. The UGV will drive at least one complete circle and then exit this circumnavigation procedure, returning to the original path defined in the RDDF. The target localization and inspection procedure is then repeated until the UGV reaches the final waypoint in its course.

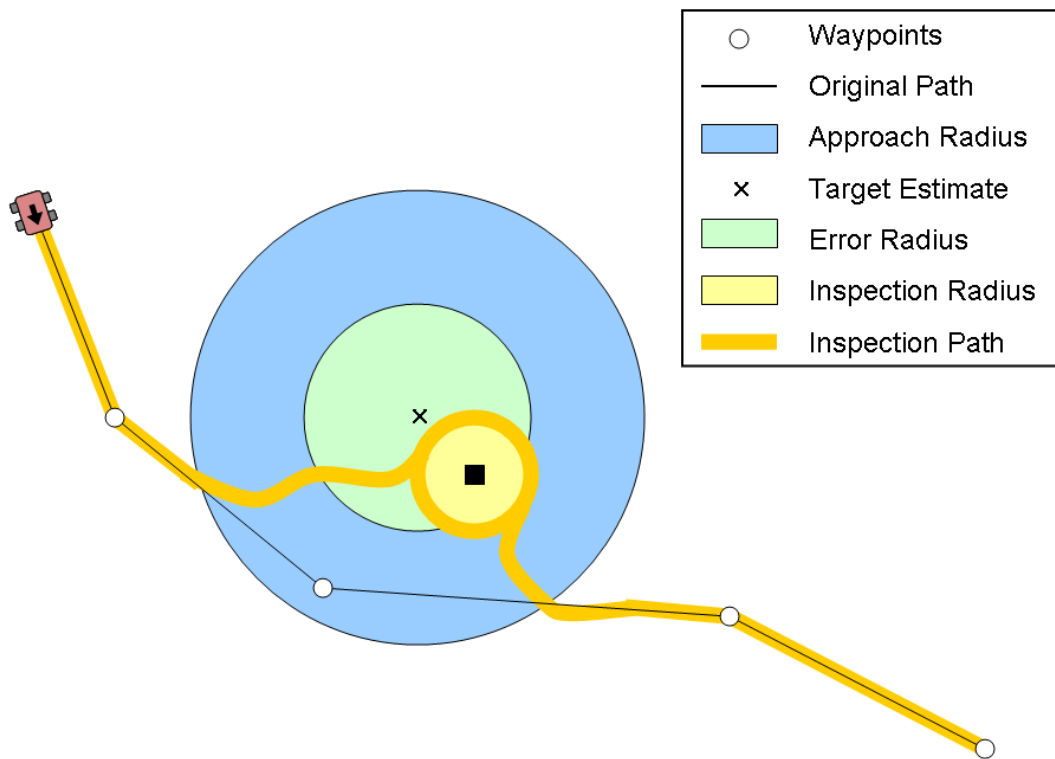


Figure 5.5: A diagram identifying the various inspection parameters.

Chapter 6

Experimental Results

To demonstrate the algorithms discussed in the previous chapter, field tests were performed. These tests included individual component tests that demonstrated all behaviors required of the ground vehicle in the mission scenario described in Chapter 1, Section 1.3. This chapter presents the results of the experiments that were performed.

6.1 Individual Component Tests

As the software was developed, each component was first tested separately to verify its functionality. This experimentation also allowed for tuning of the algorithm parameters to provide the best results. A basic structure for the experiment was the ability to traverse a course defined by a series of waypoints using the DARPA RDDF format, therefore, this was tested alone first. Distinguishing targets using the SICK LIDAR sensor was also independently tested.

6.1.1 Waypoint Navigation

To test the waypoint following software, a RDDF was created that would provide a variety of types of navigation paths, including sharp turns, a slalom, straightaways, and other turns of varying angles. The waypoints used for this test are plotted as Figure 6.1. For this experimentation, the RDDF led the vehicle over the course in a counterclockwise manner. The course was traversed at varying speeds and lateral boundary offsets in order to examine the effect of modifying these parameters on navigation. To attempt to remove the uncertainty due to changes in GPS accuracy, all waypoint navigation tests were performed around the same time of day, and the localization Circular Error Probable (CEP) was recorded at the start of each navigation trial. The localization CEP for the tests had an average of 9 cm. A deceleration of 1 m/s^2 was used for the turn ahead portion of the speed control and a max acceleration of 2 m/s^2 was applied for the overall limit. Figure 6.2 shows the plot of a 20 mph trial, with the vehicle speed shown in the third dimension as a color scale. Trials were performed with maximum speeds of 5, 10, 15, and 20 mph. There were not any noticeable variations in vehicle path due to modifying the maximum speed. This was expected as the vehicle control decelerates before turning. The plot of the 20 mph run shows the vehicle speed as an effect of acceleration and deceleration through different segments of the course. It can be seen that for this RDDF, the vehicle speed remains between 4 and 12 mph for the majority of the course, only hitting the 20 mph limit on the two long, straight sections. At one point, the speed actually reaches slightly above the limit. This occurs on a steep downhill section of the course, for which the braking is loosely tuned. Additionally, there is no 0 mph segment shown, as this data was collected on one representative lap of several, and did not include the start or stop points.

The next variable adjusted was the Lateral Boundary Offset. The LBO is used to tighten the path that the ground vehicle traverses through narrow sections of its mission, and to reduce the restrictions on areas where the exact path of the vehicle is not

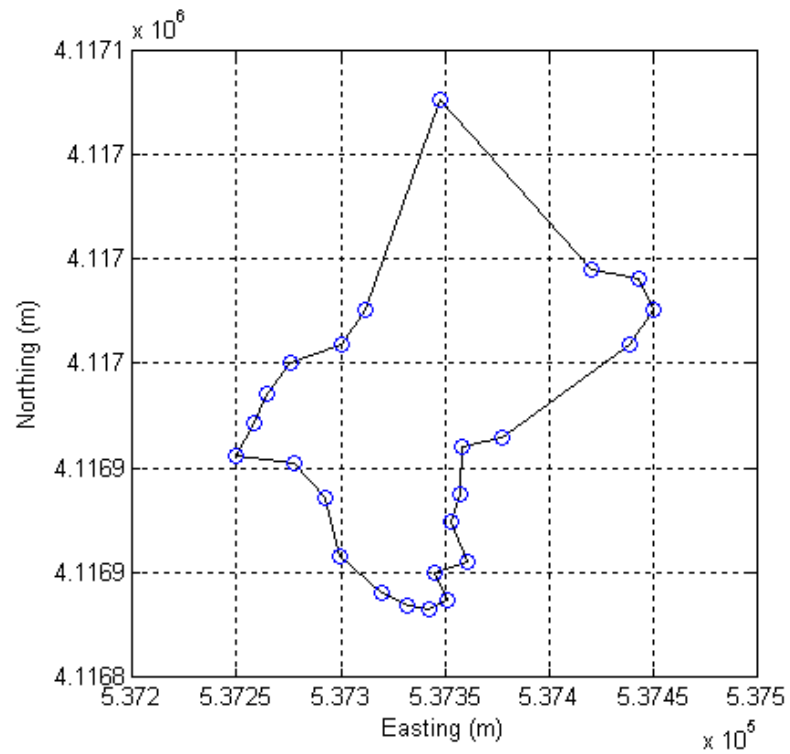


Figure 6.1: The waypoints used for the navigation tests.

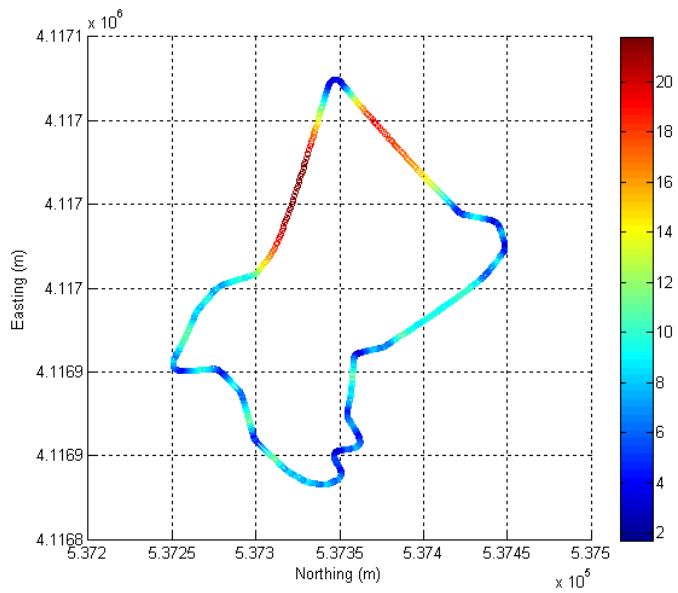


Figure 6.2: A colorbar plot of the vehicle location and speed for a maximum speed of 20 mph.

as important. Modifying this parameter relaxes the distance required to achieve a waypoint. The same test course as before was used to demonstrate the effect of changing the LBO while using a constant speed limitation of 10 mph. The results from this test were plotted and included as Figure 6.3. The effects of changing this parameter is much more noticeable than modifying the speed limit. Increasing the LBO also reduced the total time required to traverse the loop. This is due to the smoothing of the turns, allowing less deceleration and an overall shorter vehicle path. The average lap times for each LBO setting are shown in Table 6.1.

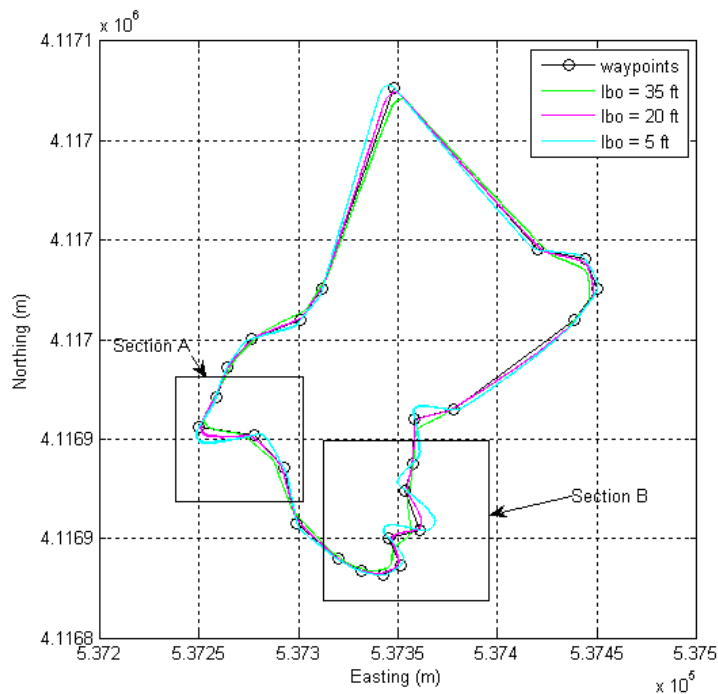


Figure 6.3: Overlay plots of the RDDF and vehicle location for varying Lateral Boundary Offset.

Table 6.1: The average lap times for varying LBO values.

LBO (ft)	Lap Time (s)
5	248
20	218
35	199

Changing the Lateral Boundary Offset has interesting effects on the overall vehicle path. Figure 6.4 shows zoomed-in views of sections A and B from Figure 6.3. In general, as the LBO increases, the turns are less sharp and the vehicle has to travel a shorter distance. It can be seen that the UGV approaches the point-to-point path the closest when the LBO is set to be 20 ft. This shows that when increasing the LBO in order to follow waypoints tightly, there is an effective limit before the results diminish. Because the test platform, Rocky, is an Ackermann steered vehicle, a finite turning radius exists. For this vehicle the turning radius is approximately 18 ft, explaining why out of the three LBO values chosen, 20 ft is the value that follows the point to point path the closest. A LBO of 5 ft accurately passes through all waypoints, however, if the angle between waypoints is greater than 40 degrees, the vehicle tracks off course.

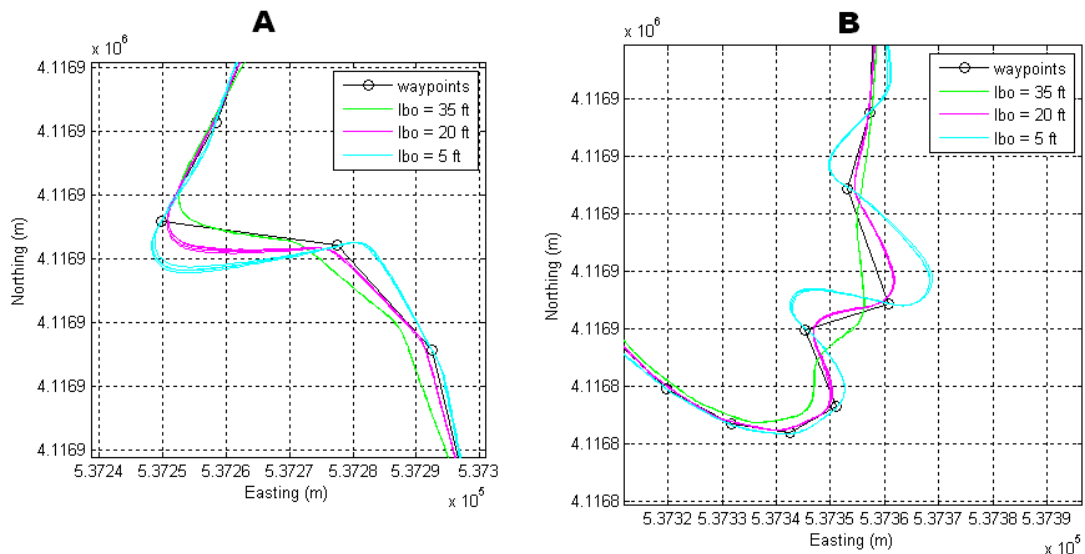


Figure 6.4: Sections A and B from Figure 6.3 shown on a 10 m grid.

The above tests of the basic waypoint navigator demonstrate the expected performance for different RDDF parameters. These tests expose the strengths and weaknesses of the simple waypoint navigation algorithm included in this thesis. The algorithm produces repeatable results for any RDDF and is useful for many autonomy applications where

the vehicle route is most important. However, this approach is limited in that it does not optimize the vehicle path due to speed. Instead of rounding corners to prevent rollover, this method simply slows to a safe speed before the turn. Because speed optimization is not necessary for the work presented in this thesis, this algorithm will be sufficient as it provides a stable framework for testing other components.

6.1.2 LIDAR Target Detection

The objective of the LIDAR target detection independent tests was to verify the functionality of the detection algorithm when viewing a target from varying distances and angles. Acquiring a practical estimate of how well this algorithm performed was necessary before attempting any autonomous trials. This procedure also allowed parameter values that worked well in practice to be selected.

While testing the ability of the software to properly identify targets, the limitations of using a single plane scanner became apparent. On uneven terrain, the vehicle pose directly effects target detection. It was expected that the applicable sensor range would be effected; however, the extent may have been underestimated. A small angle of the sensor relative to the ground often causes the sensor's field of view to include large portions of ground or sky, thus decreases the usable data. Using basic trigonometry, the effect of vehicle pitch on scan displacement is summarized in Table 6.2. To maximize the effectiveness of the SICK laser scanner, its mounting height was adjusted. The sensor was raised from its original height of 12 in to a height of approximately 40 in. This new height was chosen as it is on the order of half the height of the vehicle sized target assumption. Placing the sensor so that the scan plane would intersect the target at mid height on even terrain, maximizes the range in both negative and positive UGV pitch for which the target will still be detected. Raising the scanner also greatly reduces scan blockage due to overgrown grass in the operating area.

Table 6.2: Scan displacement effects due to sensor pitch.

UGV Pitch (deg)	Displacement at 10 m (m)	Displacement at 20 m (m)	Displacement at 30 m (m)
1	0.17	0.35	0.52
2	0.35	0.70	1.05
3	0.52	1.05	1.57
4	0.70	1.40	2.09
5	0.87	1.74	2.61

The effectiveness of the detection algorithm is also limited when the target is close to other obstacles. These obstacles may obstruct the view of the target or may be merged into the target if they are too close. Figure 6.5 shows an example, where if the object gradient threshold is too large, objects may be merged giving poor classification results. Alternately, if the threshold is too low, then targets may be broken up, creating smaller objects that are filtered due to size. This figure shows a scenario where a vehicle is placed near a pile of brush. In part A, the object gradient is adjusted too loosely and the objects are merged, providing a poor target correlation. Part B shows the same data with an appropriately adjusted object gradient threshold.

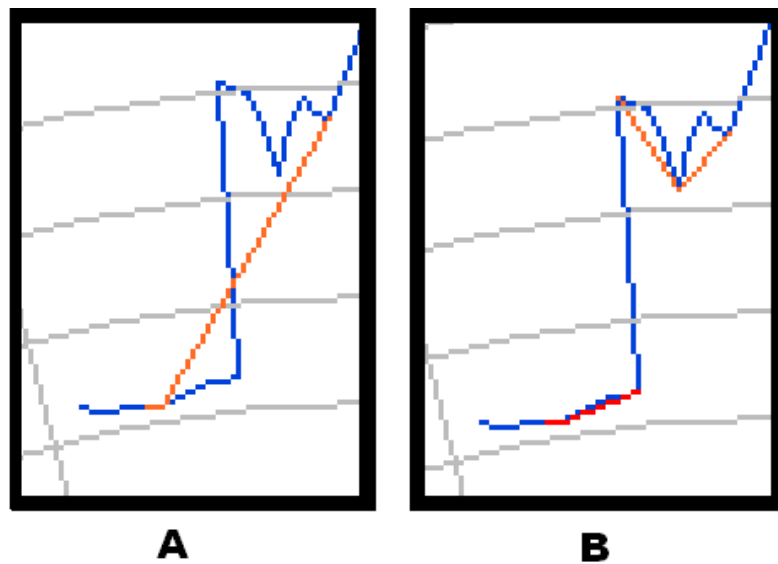


Figure 6.5: A display of failure due to a large object gradient threshold.

Testing the LIDAR target detection component separately allowed the parameters to be tuned to detect objects resembling the target assumption, while rejecting objects that did not. The methods for detecting and localizing targets worked well when the terrain was relatively smooth and the targets were not occluded. The target detection failures that occurred were most often due to limitations in the single plane LIDAR configuration and not the software algorithm. Using a multi-plane scanner would allow the same algorithms to achieve better results.

6.2 Fully Autonomous UGV Mission

Finally, the full scope of the ground vehicle's mission was tested using a notebook computer to simulate the aircraft's detection of a known target. This allowed multiple tests of the mission, without requiring the UAV, and its support team to be present. The software and the communication hardware on the ground vehicle were configured exactly as they would be if the aircraft was in the experiment. The use of the JAUS messaging standard allowed the notebook computer to appear as a seamless replacement UAV simulator to the ground vehicle.

A passenger vehicle was selected as the target for this particular experiment. A picture of an example test setup is included as Figure 6.6. The location of the target was recorded by marking its location and then using the UGV's localization system to determine the position. The autonomous testing allowed the LIDAR target detection component to be validated more thoroughly than the static testing alone. The dynamic nature of the autonomous testing prevents the occasional target occlusion from causing an entire localization failure. Because the target detection component runs four times per second, false objects or bad target correlations do not prevent a proper localization. Also, the sensor movement between iterations lessens the probability of never detecting the target.



Figure 6.6: A picture of an example setup used for LIDAR target detection.

Performing the autonomous UGV testing also allowed the circumnavigation code to be tested. The initial tests exposed a problem that had been overlooked when the code was developed. Upon beginning the circumnavigation, the UGV oscillated about the waypoint circle before properly navigating around the target. An example vehicle trajectory demonstrating this problem is included as Figure 6.7. After inspection, it was realized that the vehicle geometry was not considered when the circumnavigation procedure was created. Neglecting the turning radius of the vehicle caused this problem. The code was modified to allow the UGV to begin the circumnavigation procedure one turning radius before it reached the inspection circle. This modification allowed the vehicle to perform the inspection smoothly.

The fully autonomous UGV mission ran as planned. Two courses and three target estimations were used to validate the software. For each target estimation, a test was performed with inspection radii of 10 and 15 m. Out of these 12 scenarios, the UGV successfully circumnavigated the target in all but one. The following paragraphs describe this collection of tests.

One of the successful trials on the first course is shown as Figure 6.8. In this figure, the UGV travels from south to north, with the target estimate placed past the actual target.

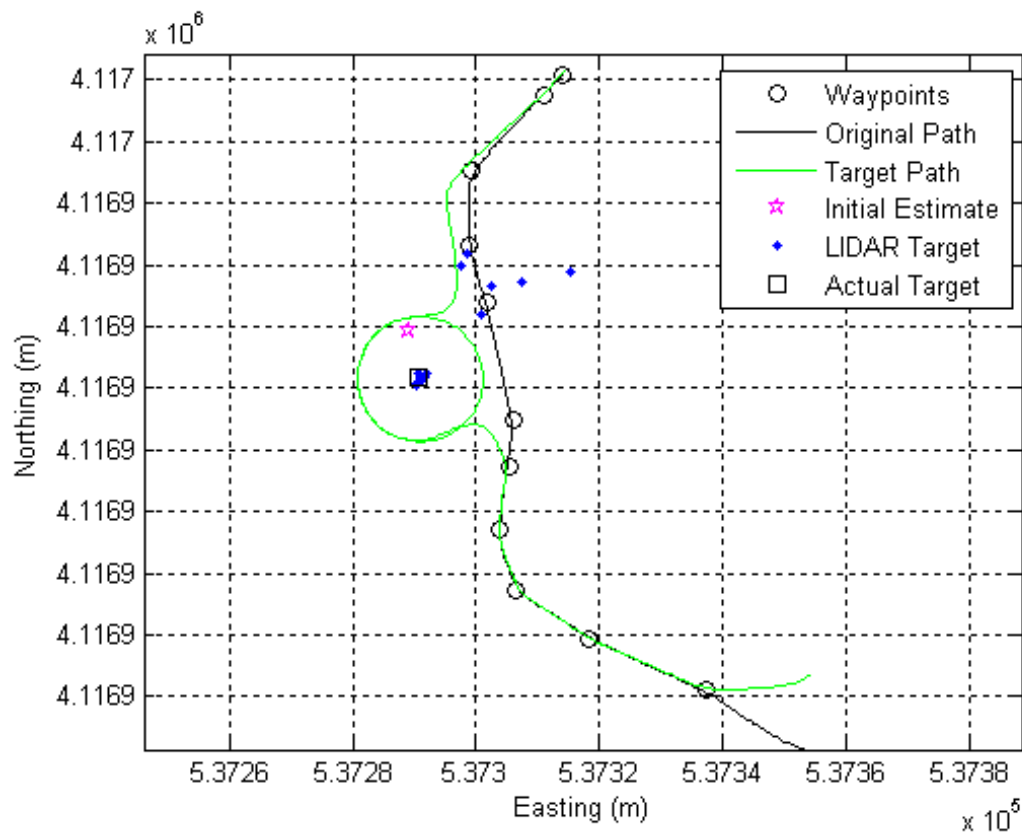


Figure 6.8: The vehicle data collected for an inspection with a 10 m radius.

was estimated to be north of its actual location. For one trial of the second set, the UGV chose a path that never detected the actual target. On this trial, the ground vehicle chose a telephone pole as the best target estimate and proceeded to circumnavigate the pole. The results of this test are included as Figure 6.10. It is suspected that the terrain the vehicle traversed did not allow the target to be seen before it committed to circumnavigating the telephone pole. This occurred on the 15 m run of the same target estimation that was successful in Figure 6.9. Because the estimate was on the near side of the actual target, the LIDAR target detection code would have needed to localize the target at a range of approximately 25 m. At this range, the only target that had been detected was the telephone pole, so the UGV circled a point somewhere between the pole and the target estimate. This circle did not include the actual target.

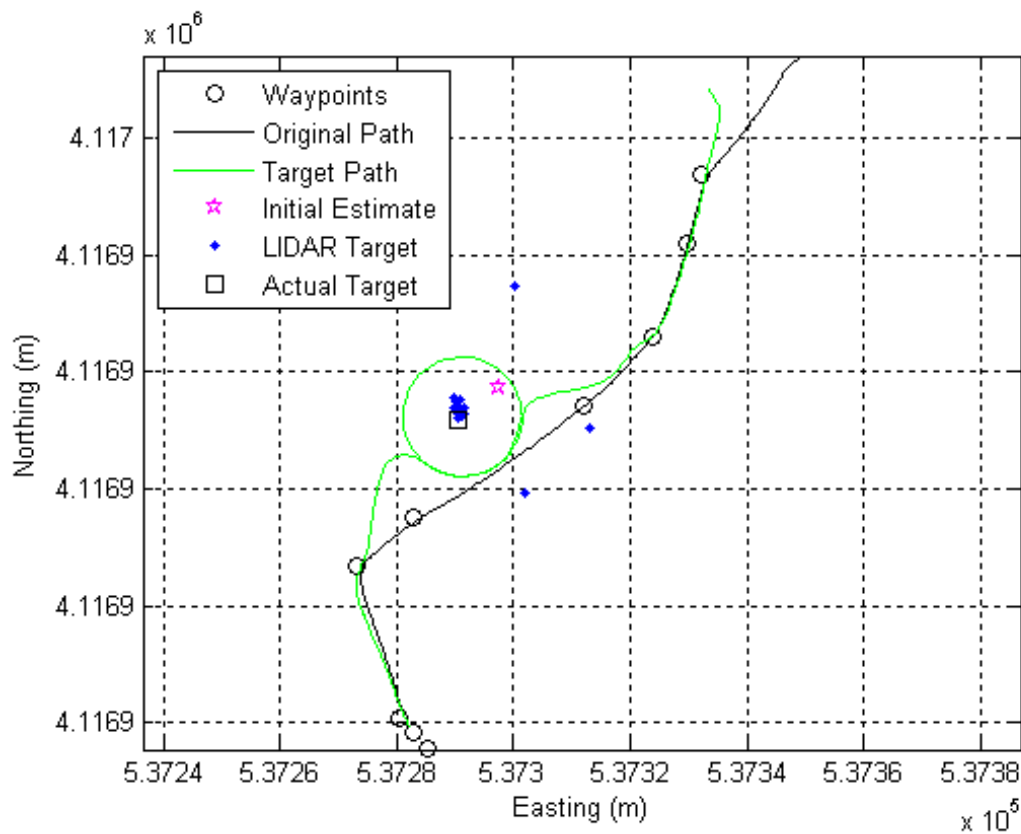


Figure 6.9: The UGV identified the target to be slightly north of its actual location.

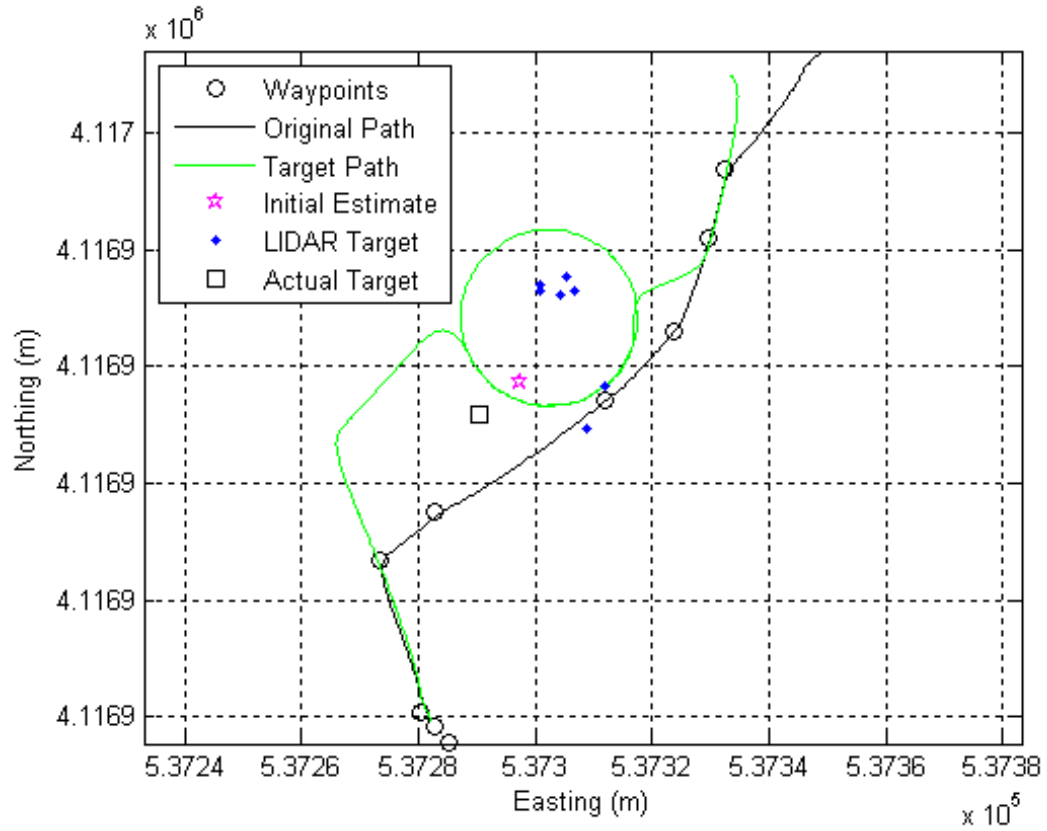


Figure 6.10: A plot of the vehicle mis-identifying the target.

Overall, the UGV was able to correctly identify and circumnavigate the target in 11 of 12 trials, using various inspection parameters and target estimates. Plots for all 12 trials are included in Appendix D. These tests validated the autonomous target detection component, demonstrating its success in localizing targets given an estimate of the actual location.

Chapter 7

Conclusions

Autonomous vehicle development is an area of increasing interest. This technology will remove humans from many dull, dirty, and dangerous tasks that they currently must encounter. Adding cooperative behaviors to this technology is an even further advancement that will enable many additional possibilities in the unmanned realm. This thesis investigated fully autonomous cooperation between a ground and an air vehicle with the goal of obtaining a team more capable than either individual asset.

Software was developed to incorporate a ground vehicle into a target search and inspection procedure with an aerial vehicle. This team used circular autonomy, as opposed to a master slave configuration to maximize the utility of the pair. The ground vehicle was programmed with a route to travel, for which it sent the UAV look-ahead coordinates to search for targets along the way. Once a target was detected by the UAV, the estimated GPS location was sent over the network and stored by the UGV. When the UGV neared the target location, it further localized the target using LIDAR and performed an

inspection by circumnavigating the object. After the inspection was completed, the entire process was repeated until the mission end.

The software algorithms and resulting experiments presented in this thesis demonstrate successful fully autonomous coordination between air and ground vehicles. Using the JAUS standard, this software was easily tested with different UAVs and other simulated subsystems. The collaborative target localization and inspection procedure allowed the team to achieve accurate position information and video surveillance of objects on the ground. This process took advantage of the wide coverage that the aircraft provided and coupled that with the detailed sensing of the ground vehicle, creating a result better than what either component could produce alone. The success of this experimentation shows just one of many advantages obtained when intelligent communication exists between autonomous vehicles. This use of multiple assets in an autonomous network will likely be prominent in the future of unmanned robotics.

7.1 Future Work

There are several ways the technology presented in this thesis could be expanded in the future. A few of these possibilities are, improving the target identification procedure, adding a multi-plane LIDAR system, and expanding to a many vehicle cooperative mission. Following paragraphs discuss this additional work that could be done to continue similar heterogeneous vehicle teaming research.

If similar target identification is required for future experimentation, some improvements could be made. To prevent the problem caused when only one side of the target is in view, a behavior could be added that moves the UGV to a position where two sides are visible before the circumnavigation procedure begins. This would allow depth information to be measured before calculating the target location. A second method of improving

the target localization procedure would be to add a side-mounted LIDAR sensor. The addition of a side mounted LIDAR sensor would also aid with the target inspection process, obtaining location information for all sides of the target, and allowing target size to be captured.

The use of a multi-plane LIDAR scanner would make several improvements to the overall robustness of the detection software. Using a scanner that provides multiple planes of data will improve the likelihood that a scan plane will intersect with the target, even over more uneven terrain. A multi-plane scanner also provides a better ability to filter out ground plane data, allowing for the addition of a more relaxed target shape assumption. The distance gradient between additional vertical planes can be used to determine what returns might be the ground so that it may be filtered, leaving only target data.

While the search and localization procedure presented in this thesis might be a good approach to finding targets near a road, other algorithms using similar technology could be developed for large area searches. A multi-aircraft, multi-ground vehicle approach could quickly and effectively search a large area. In this environment the aircraft would detect possible targets and subsequently dispatch a ground vehicle on a route toward the point of interest. This type of operation could easily be scaled to incorporate many autonomous vehicles operating together over a wide area.

Ultimately, much work needs to be done at the experimental level on all aspects of unmanned ground robotics before the technology becomes widespread. However, steps like the ones taken in this thesis research are the key to building a successful use of autonomous ground vehicles in the future.

References

- [1] United States Office of Management and Budget. *Budget of the United States Government: Department of Defense, 2006*. Available Online: <http://www.whitehouse.gov/>
- [2] United States Office of Management and Budget. *Budget of the United States Government: Department of Defense, 2006*. Available Online: <http://www.whitehouse.gov/>
- [3] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas. "Cooperative Air and Ground Surveillance," IEEE Robotics and Automation Magazine, September 2006.
- [4] Virginia Center for Autonomous Systems. *Heterogeneous Teams of Autonomous Vehicles - Advanced Sensing and Control*. Proposal to the Office of Naval Research, Virginia Tech, 2005.
- [5] SIG Manufacturing Company, Inc. *SIG Rascal 110 ARF*. Available Online: <http://www.sigmfg.com/>
- [6] Monda, Mark. "Re: Questions about the Rascal UAV" Email to David Van Covern, November 20, 2007.
- [7] Cloud Cap Technology. *A Highly Integrated UAV Avionics System*. Hood River, OR, April 2003. Available Online: <http://www.cloudcaptech.com/>
- [8] Cloud Cap Technology. *Communications for the Piccolo Avionics*. Hood River, OR, September 2006. Available Online: <http://www.cloudcaptech.com/>
- [9] Cloud Cap Technology. *TASE Gimbal Integration Guide*. Hood River, OR, July 2007. Available Online: <http://www.cloudcaptech.com/>
- [10] Ingersoll-Rand. *Club Car Rough Terrain XRT 1550*. Available Online: <http://www.clubcar.com/>
- [11] Quicksilver Controls, Inc. San Dimas, CA. Available Online: <http://quicksilvercontrols.com/>
- [12] NovAtel, Inc. *OEM 4 Family User's Manual Volume 1*. December 2005. Available Online: <http://www.novatel.com/>

- [13] SICK AG. *LMS200/211/221/291 Laser Measurement Systems - Technical Description*. Germany 2006. Available Online: <http://www.sick.com/>
- [14] Motorola North America. *Motorola's Mesh Networking Technology History*. 2004. Available Online: <http://www.motorola.com/>
- [15] Motorola North America. *QDMA and the 802.11 Radio Protocol Compared*. 2004. Available Online: <http://www.motorola.com/>
- [16] Ruel Faruque. "A JAUS Toolkit for LabVIEW, and a Series of Implementation Case Studies with Recommendations to the SAE AS-4 Standards Committee," Master's thesis, Virginia Tech, 2006.
- [17] JAUS Working Group. Available Online: <http://www.jauswg.org/>
- [18] Wolfgang, Baer. "Generating one-meter terrain data for tactical simulations," Military Intelligence Professional Bulletin, Oct-Dec 2002.
- [19] Dan Broadstreet. "Warfare Center to Host Autonomous Unmanned Vehicle Fest 2007," Navy Newsstand, December 2006. Available Online: <http://www.news.navy.mil/>
- [20] Defense Advanced Research Projects Agency. *Route Data Definition File*. August 2005. Available Online: <http://www.darpa.mil/>

Appendix A

Acronyms

AAVs Autonomous Aerial Vehicles

AGVs Autonomous Ground Vehicles

ASVs Autonomous Surface Vehicles

AUVs Autonomous Underwater Vehicles

CEP Circular Error Probable

GPS Global Positioning System

LIDAR Light Detection and Ranging

UGV Unmanned Ground Vehicle

UAV Unmanned Aerial Vehicle

NPS Naval Postgraduate School

PWM Pulse Width Modulation

RC Remote Control

DAQ Data Acquisition

ONR Office of Naval Research

DARPA Defense Advanced Research Projects Agency

INS Inertial Navigation System
IMU Inertial Measurement Unit
PID Proportional Integral Derivative
JAUS Joint Architecture for Unmanned Systems
QDMA Quadrature Division Multiple Access
LAN Local Area Network
JGRE Joint Robotics Ground Enterprise
OS Operating System
SPOI Sensor Point Of Interest
RDDF Route Data Definition File
LBO Lateral Boundary Offset
UTM Universal Transverse Mercator
TNT Tactical Network Topology
LRV Light Reconnaissance Vehicle
UDP User Datagram Protocol
PVNT Perspective View Nascent Technology
NSWC Naval Surface Warfare Center
WGS World Geodetic System

Appendix B

Initial UAV Control Message

The binary UDP string used for the control of the NPS Rascal UAV.

Data	Type	Notes
Control	double	0 = Not Controlling, 1 = Take Control
Latitude (degrees)	double	
Longitude (degrees)	double	
Altitude (m)	double	Above mean sea level
Waypoint index	double	Piccolo waypoint number
Orbit Radius (m)	double	
Time on orbit (s)	double	
Turn direction	double	0 = clockwise, 1 = counter clockwise
Next Waypoint	double	waypoint to target when control is released
Camera Mode	double	0 = downward, 1 = target waypoint, 2 = manual
Pan (deg)	double	used when camera mode = 2
Tilt (deg)	double	used when camera mode = 2
Camera Zoom	double	zoom percentage of camera lens
Release	double	1 = release payload

Appendix C

Custom JAUS Message

Code D802h: Report Target Location The Report Target Location message provides the position of a target or point of interest. The position of the platform is given in latitude, longitude, and elevation, in accordance with the WGS 84 standard.

Field #	Name	Type	Units	Interpretation
1	Presence Vector	Unsigned Short	N/A	See mapping table that follows.
2	Time Stamp	Unsigned Integer		Bits 0-9: milliseconds, range 0...999 Bits 10-15: Seconds, range 0...59 Bits 16 – 21: Minutes, range 0...59 Bits 22-26: Hour, range 0..23 Bits 27-31: Day, range 1...31
3	Latitude (WGS 84)	Integer	Degrees	Scaled Integer Lower Limit = -90 Upper Limit = 90
4	Longitude (WGS 84)	Integer	Degrees	Scaled Integer Lower Limit = -180 Upper Limit = 180
5	Elevation	Integer	Meters	Scaled Integer Lower Limit = -10,000 Upper Limit = 35,000
6	Position RMS	Unsigned Integer	Meters	A RMS value indicating the validity of the position data. Scaled integer Lower Limit = 0 Upper Limit = 100
7	Target ID	Unsigned Short	Radians	An ID number assigned to the target by the identifying vehicle

Vector to Data Field Mapping for Above Command								
Vector Bit	15	14	13	12	11	10	9	8
Data Field	R	R	R	R	R	R	R	R
Vector Bit	7	6	5	4	3	2	1	0
Data Field	R	R	7	6	5	4	3	2

"R" indicates that the bit is reserved.

Table C.1: Custom JAUS formatted message D802h: Report Target Location.

Appendix D

Fully Autonomous Target Localization Plots

Included are plots from each of the 12 fully autonomous target localization runs that are described in Chapter 4. Plots A through F are for data collected on course one and plots G through L are for data collected on course two.

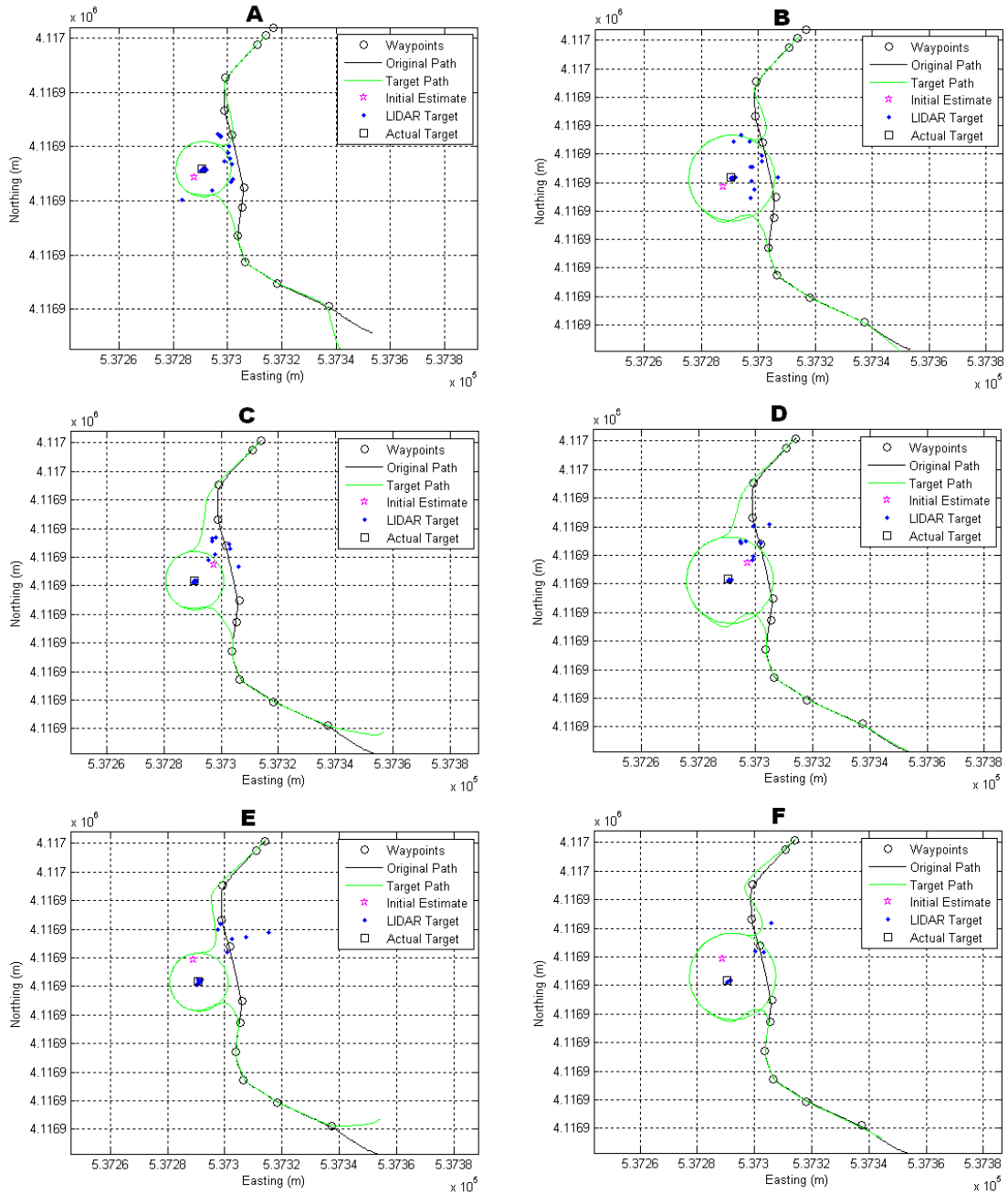


Figure D.1: Plots of data recorded from course 1 trials.

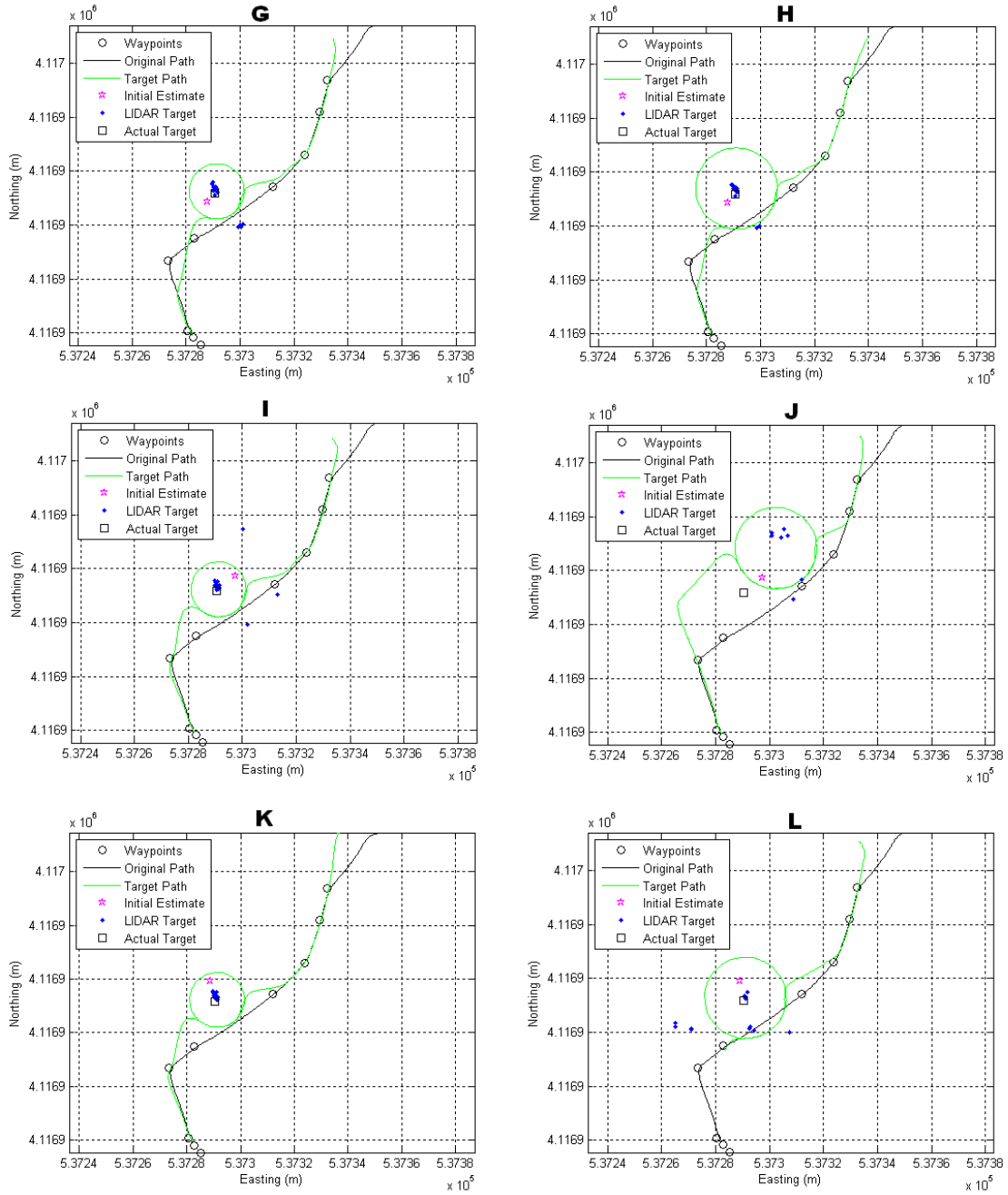


Figure D.2: Plots of data recorded from course 2 trials.