Design of a Helicopter Deployable Ground Robotic System for Hazardous Environments

Michael S. Rose

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
Corina Sandu
Robert L. West, Jr.

January 25, 2010
Blacksburg, Virginia

Keywords: Robotics, Tracked Vehicle, Motion Planning, Teleoperation, Situational Awareness

Design of a Helicopter Deployable Ground Robotic System for Hazardous Environments

Michael S. Rose

ABSTRACT

The use of robotics in hazardous environments is becoming more common, where autonomy can handle the dull, dirty and dangerous jobs that humans have previously supported. This thesis focuses on the design of a helicopter deployable unmanned ground vehicle for use in hazardous environments, and presents the benefits of heterogeneous unmanned vehicle teams for operation in beyond line-of-site hazardous environments.

The design of a ground robot that is capable of being flown on the undercarriage of a Yamaha RMAX unmanned air vehicle is presented. The robot is size, weight, and power limited and must be capable of traversing rough, unstructured terrain. The results of testing show that the design criteria for size, weight, and mobility are met. A path planning algorithm is developed using the A* search algorithm for the planning of optimal paths through rough terrain. The algorithm makes use of a vehicle/terrain interaction model to compute the cost of path traversal. In the CONOPS, the terrain model is generated real-time during a mission through the use of a stereovision system carried on the helicopter, which station-keeps above the ground robot. The algorithm simulates the robot on the terrain and presents the best feasible path to the operator to aid in teleoperated robot navigation. Simulations of the planning algorithm provided feasible paths over a rough terrain environment.

A user study was conducted that evaluates the abilities of both mono- and stereo-vision systems in providing the teleoperator with adequate situational awareness with the intent of proving that stereovision data is more effective at aiding the user in making timely navigation decisions. The results of the study showed that the helicopter-mounted stereovision system was more efficient than the monovision system with respect to navigation time, the number of invalid moves, and total moves required for navigation of a simulated rough terrain environment.

To my loving wife Diane

Acknowledgments

I would like to acknowledge and thank Dr. Kevin Kochersberger, my advisor, for his willingness to bring me into the Unmanned Systems Lab and allow me to work on this project. Without him, this work would not have been possible. Many thanks to him for all of the helpful ideas and insight that led to the completion of this work. I would also like to thank the rest of my committee for their support. A special thanks goes to David Conner for his helpful insights into path planning.

I would also like to show my gratitude to those that helped me accomplish my research in various ways: Mohamed Saleh for his help in acquiring stereovision images, Amber Franklin for her help in recruiting volunteers for the user study, and all of the students that helped to dig and construct the debris field used during this work. I also owe a great deal of thanks to Mike Bromley and Johnson Miles for the numerous hours of manufacturing work put in to fabricating the ground robot. I'd also like to express my appreciation for my colleagues and friends at the Unmanned Systems Lab. I could always count on them for help, advice, and of course, the occasional distraction to help me keep my sanity.

I owe my deepest gratitude to my parents for their endless love and support. They have been an integral part in my education through their encouragement and example. They have taught me the importance of education and the principle of hard work.

Finally, I would be remiss in not thanking Diane, my loving and ever patient wife. I know that she endured through just as much, if not more than I did throughout my research and the writing of this thesis. I will be forever grateful to her for her continuing encouragement and support of my education.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

This thesis presents a portion of the work conducted by the Unmanned Systems Lab at Virginia Tech to develop a helicopter deployable ground robotic system for use in hazardous environments. For the purpose of this research a hazardous environment has been defined as one that presents a serious risk of injury, detriment to health, or the potential for loss of life to a human operator. Hazards may be present in many forms; for example, the wreckage caused by a natural disaster or explosive device, exposure to radiation or toxins, or in the arena of war or other hostile environment. Robots are seen completing many tasks in place of humans in research facilities, manufacturing facilities and assembly lines, and even in our day to day life. However, the operation of unmanned robotic systems in hazardous environments poses some additional difficulty, whether it be due to beyond-line-of-sight operation, the potential latency or loss of information cause by wireless links, or perhaps due to the hazards presented themselves that cause difficulty in acquiring sensory information or that hinder mobility and navigation. The development of an unmanned ground vehicle (UGV) for use in hazardous environments provides for keeping humans safe and out of harms way. While other UGVs have been previously developed, the addition of a UGV that is deployed by an aerial platform will be a great asset to the systems that have already been developed for use in hazardous environments.

1.1 Motivation

The primary goal of this research is to develop a system that can be deployed in order to collect information, be it sensory or physical, about a disaster site resulting from the detonation of an explosive device in an urban setting. In addition to the ground robotic system being developed, aerial support for the mission is being developed that can be deployed separately, or in conjunction with the ground robot. One key aspect of the aerial support is a terrain mapping mission that will provide some of the navigational information and situational awareness for the robot mission[1]. Situational awareness is defined as "*perception* of the elements in the environment within a volume of time and space, the *comprehension* of their meaning, and the *projection* of their status in the near future"[2].

The primary motivation of developing this system is to prevent injury or the loss of human life. As such, a robotic system is developed that can be deployed in place of a human or group of humans for dull, dirty or dangerous tasks. In order to successfully implement the use of a robotic system in hazardous environments, the system must be capable of operating in the urban, post-blast, disaster site for which it is intended. In addition, the method in which the system operates must have an intuitive, easy to use interface, and the operator must be provided with enough information in order to complete the designated task.

A ground robot operating in the designated scenario will be required to navigate over and around a number of obstacles. For this reason, mobility is a primary concern. In addition, mission personnel will remotely operate the vehicle in a beyond line-of-sight environment. The operator must be presented with enough information to gain a good enough perception of the robot's surroundings in order to operate the vehicle successfully. The information may be presented in the form of images, video, and other sensory information of the surrounding environment. Generally, this information comes from sensors or cameras on-board the ground vehicle. Because this ground robot will be deployed from an unmanned aerial vehicle (UAV), the UAV will be able to continuously provide additional information during robot operation that will aid the operator of the ground vehicle. The additional information provided by

the UAV is one of the primary motivations in developing a joint aerial and ground robotic platform.

The following sections present a mission overview, a brief explanation of the complete system architecture, and a presentation of the specific work conducted for this thesis.

1.2 Mission Objectives

The primary objective of the mission is to collect information about the explosive device in order to identify the origin. In order to do this, the mission has been broken down into three subtasks: aerial data collection, 3-D terrain mapping, and ground data collection. In order to fulfill these tasks, a modular system architecture has been developed, presented in Figure 1.1. The interchangeable systems allow for a single UAV to be adapted to the requirements of the specific mission scenario, or for multiple UAVs to be deployed, each with a different system, if required by the mission scenario. In addition, the modular system architecture allows the individual systems to be platform independent.

Each of the interchangeable systems collects information that can be used in support of ground robot operations. The aerial data collection system collects data from various sensors that aids in localizing areas of interest for robot deployment. The 3-D terrain mapping system collects high-resolution aerial imagery of the areas of interest and provides a means for building a 3-D terrain model of the areas of interest. In addition to the interchangeable systems, the permanent systems on-board the UAV are required for successful UAV operation. The E-box, a custom designed electronics package provides a way to communicate with the payloads on-board the UAV and acts as a communications relay between the ground robot and the ground control station. The flight controller allows for autonomous and operator-assisted flight of the UAV, and the TASE camera is a gimbaled video camera that provides an additional eye-in-the-sky view for surveillance and robot operations monitoring.

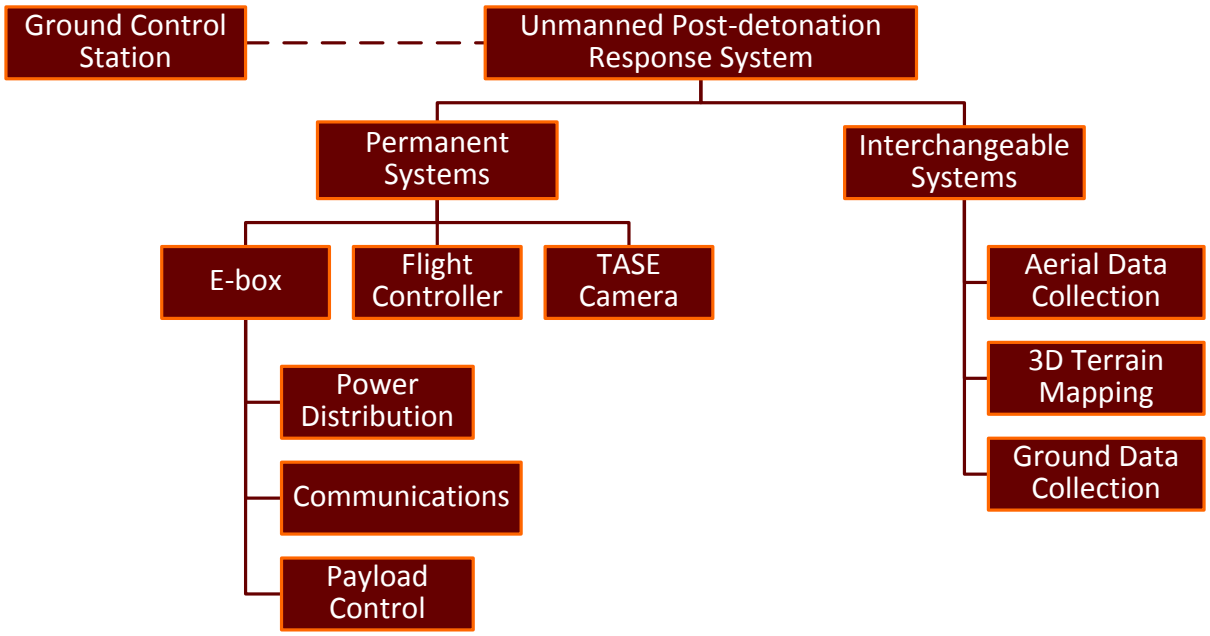


Figure 1.1: System architecture

1.3 Concept of Operations

A potential concept of operations (CONOPS) has been outlined in Figure 1.2, with the ultimate goal being information acquisition. Steps 1 and 2 would be conducted in support of the robot deployment mission in order to provide the mission commander with sufficient information to choose an optimal robot deployment site for data collection. During steps 1 and 2 the terrain mapping system would be used to generate a terrain map that could be used for path planning and robot operations as discussed in chapters 4 and 5. The aerial data collection system is used to collect data which could be overlaid on a map of the post-detonation site in order to locate a region of interest where a high probability exists of gaining more information about the detonation event as discussed in [3]. Once a deployment site for the ground robot is determined the UAV will be equipped with the ground data collection system and will fly autonomously to the deployment site. Once at the deployment site, the ground robot will be deployed via a lightweight monofilament or braided synthetic tether, such as a high test fishing line. Mission personnel will then teleoperate the robot

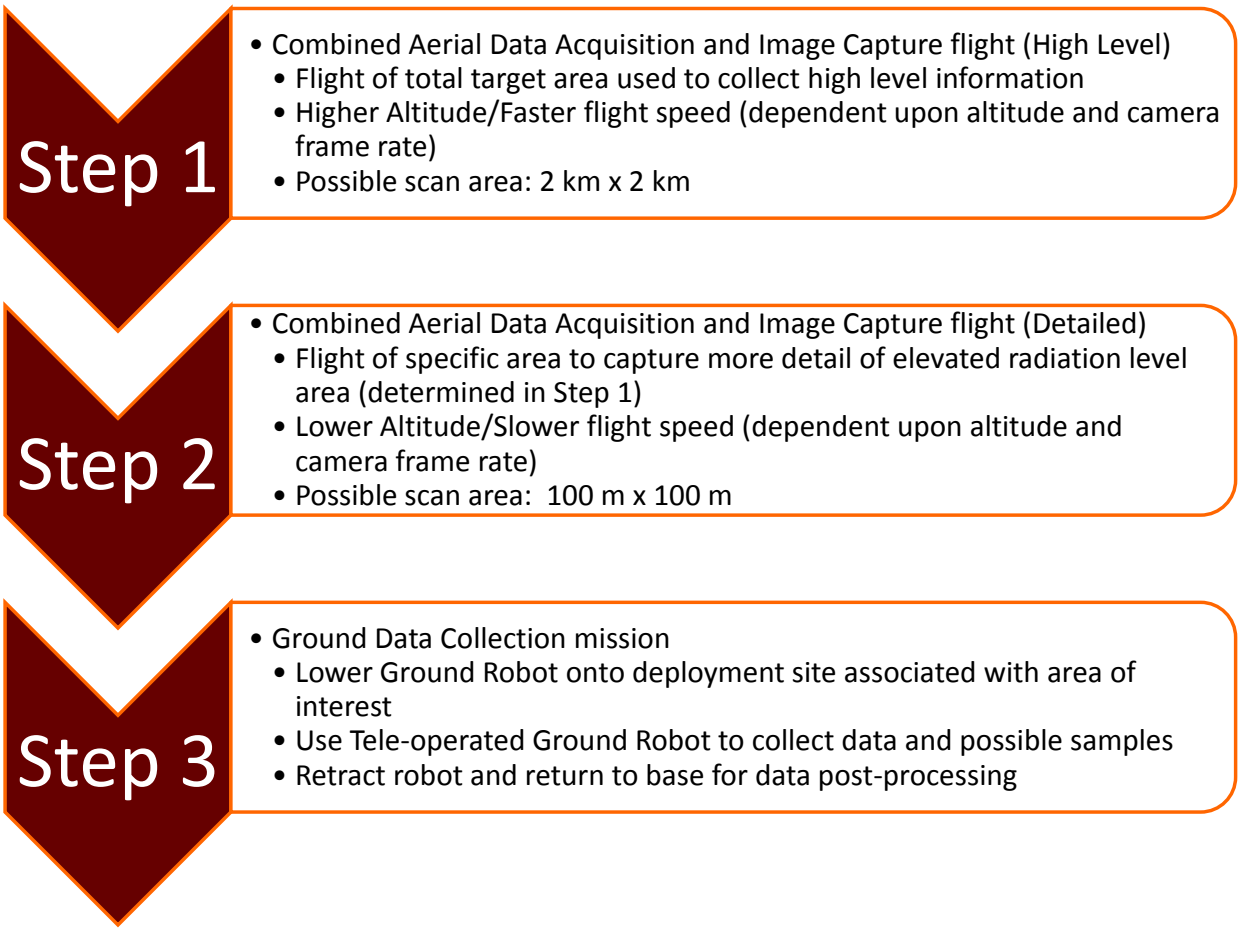


Figure 1.2: Concept of operations

in order to collect additional data at the site. During robot operation, the UAV will hover directly above the robot and autonomously follow the robot as it moves. Upon acquisition of a sufficient amount of data, the ground sampler is retracted to the helicopter and the helicopter returns to base.

1.4 Description of Specific Work

The research discussed in this thesis will address three key aspects of the ground robotic system:

1. The design and fabrication of a suitable mobility platform (chapter 3)
2. A study of path planning and the benefits it may play in increasing the situational awareness of the robot operator and decreasing mission time (chapter 4)
3. Improvement of the robot/operator interface by examining methods of increasing situational awareness through vision system studies and user interface development (chapter 5)

As mentioned previously, hazardous environments can make it difficult for unmanned systems to operate. A suitable mobility platform is paramount to ensure the success of the mission. Although various aspects of the ground robot mission have been presented, only the mobility platform design and performance characteristics will be addressed by this thesis. Elements of the ground robot mission such as sampling and sensor capabilities, integration of the robot into the aerial platform, and control of the the tethering mechanism will not be addressed in depth by this work, but will remain as future areas of research.

The ability of the operator to use the robotic platform is independent of the quality of the mobility platform. If the operator is not provided with good means to gain situational awareness, then the mission may fail. In addition, hazardous environments can cause navigational

difficulties due to obstacles and other attributes of a possibly unstructured environment. By implementing a path planning algorithm that can provide a suitable path to the operator from one point to the next, or perhaps even aid the operator in navigating the robot, then the operator can gain further situational awareness, focus less on navigation, and focus more on other mission objectives. This research will address specifically the development of a rough terrain path planning algorithm. The algorithm makes use of the information gained during the 3-D terrain mapping portion of the mission to determine the best path through the rough terrain. A robot/terrain interaction model has been developed to be used by the path planning algorithm. Various methods for presenting the planned paths to the robot operator are suggested, but an evaluation of the effectiveness of the presentation methods is saved for future research.

Finally, a number of vision systems are evaluated for their ability to provide adequate situational data to the robot operator. A simulated supervisory control study is completed in order to evaluate the effectiveness of each of the vision systems. The primary goal of the study is to evaluate the use of 3-D terrain data in beyond line-of-sight robot operations as a method of providing the robot operator with increased situational awareness. The study presented in this thesis lays the foundation for future real-time studies of the effectiveness of using 3-D terrain data in remote robot operations. The following chapters will discuss the development of the mobility platform and some of the research done to improve the situational awareness of the operator of the robotic platform.

Chapter 2

Literature Review

There are many aspects involved in successfully designing and implementing an unmanned vehicle for use in a hazardous environment. The task is further complicated by involving a tethered payload in the system. This chapter contains an overview of some of the work previously done in the area of unmanned vehicle deployment in hazardous environments along with some of the requirements for successful operation.

2.1 Robotics for Hazardous Environments

As mentioned in the introduction, the use of a robotic system in an environment that may cause harm or death to human personnel is ideal. However, often times such environments add certain complications for the use of a robotic platform. These complications can be in the form of mobility constraints, detriment to sensors or electronics, or perhaps degraded wireless transmissions. Mobility concerns are further compounded when using small robotic systems. In addition, there are often trade offs between the precision, dexterity, mobility and strength of a robotic system and its endurance and reliability[4].

The research of robotics in hazardous environments began as early as the 1940s when robots

were used in the context of nuclear applications. A newer use of robotic technology is in search and rescue (SAR) applications. The first use of robotic technology in an Urban Search and Rescue (USAR) application in the United States was during the aftermath of the World Trade Center disaster[5],[6].

The authors of [6] offer a discussion of the capabilities of various robotic systems based on size and modality. The size of a robotic systems is classified as man-packable, man-portable or maxi, and the modalities of robotic systems are ground, aerial, underwater, and water surface. With respect to USAR operations it is suggested that ground robots are better suited to search missions, while aerial vehicles are better suited to reconnaissance and mapping missions. Because ground vehicles must often traverse rough, complex terrain such as rubble, mobility is of great concern. While much research has been dedicated to the development of UGVs for use on rough terrain[7],[8],[9], little work has been done to characterize rubble in order to facilitate better designs.

The benefit of having a modular payload capability is discussed in [10], which makes use of a rugged, man-packable, all-terrain UGV that is capable of supporting a wide range of payloads. The use of both chemical sensing and radiological sensing payloads are discussed by the authors.

In [11], the design and testing of a man-packable UGV for use in tunnel and sewer reconnaissance is outlined. The UGV is equipped with all of the necessary electronics and sensors for teleoperation and payload control. In addition, the robot was designed to be water-tight due to the possibility of encountering water in the tunnel system. In addition to the discussion about the design of the robot, the article includes a discussion of mechanical testing and possible improvements that can be made to the robot. Both durability and consideration of environmental impacts on the robot design are emphasized as important factors for successful operation of robotics in hazardous environments.

2.2 Cooperative Unmanned Air and Ground Systems

A helicopter deployed robotic system combines the advantages of a hovering/flying platform with those of a ground platform. For the hazardous environments for which this system is intended to operate, it is important that the rotor down-wash of the helicopter not disturb the disaster site due to the potential widespread dispersion of contaminates. By hovering out of ground effect and deploying a tethered robot from the helicopter, contaminant dispersion can be avoided, while still maintaining the capability of inspecting and exploring the site at close range. In addition, by carrying a ground robot to the disaster site via an aerial platform, the time to arrive at the site is reduced and the ground robot has the capability of accessing areas that may otherwise be inaccessible.

The advantages of a hovering robot are discussed in [12] where it is suggested that they are highly maneuverable, faster than small UGVs, and can gain access to areas that may be inaccessible to a UGV. In the context of USAR, a hovering robot can be dispatched prior to emergency crews arriving on site in order to gain situational awareness about the hazardous environment which may aid in mission planning.

The authors of [13] propose the use of a UAV carried tethered payload for use in search and rescue operations. The payload would be carried to the site of interest by the UAV and lowered from the UAV platform. The tethered payload would then proceed to deploy multiple smaller hovering robotic agents in order to search the target location and gain additional information about the target site. Tethered payload stability is discussed and the use of fans as a propulsion method to stabilize the payload is proposed. The work from [13] is continued in [14] with the modeling and development of a physical prototype of the suggested UAV/tethered payload system.

The dynamics of slung load flight in conjunction with a VTOL aircraft is a well studied problem[15]. The author provides a detailed model and analysis of the dynamics of helicopter slung load flight, focusing primarily on slung load transport flight and slung load flight of a

land mine detection sensor.

The author of [1] addresses some of the difficulties associated with the deployment and navigation of a ground robot while remaining tethered to a UAV. Some of these difficulties include utility cable detection and avoidance, the selection of a suitable deployment site, and robot tracking and navigation. The author proposes a method of using 3-D stereoscopic imagery to detect the location of utility cables so they may be avoided. In addition, the use of stereoscopic imagery is discussed as a means of determining a suitable deployment site for the tethered robot in beyond line-of-sight (BLOS) operations. Finally, a method of tracking the location of the robot in real-time using a monocular camera and a free computer vision tool called ARToolkit is discussed and implemented. The feedback from the tracking system is displayed to the operator on a 3-D terrain map to assist in the teleoperation of the robot and is also used by the helicopter in order to follow the robot as it navigates the deployment site.

Project Stork, a test of UAV-UGV collaboration conducted by the military is discussed in [16]. Project Stork successfully achieved the following objectives: delivery of a UGV by a UAV into a desired area and an extended range BLOS communication system. Project Stork deployed a UGV via parachute from a fixed wing UAV and then was able to extend BLOS communications from the then current capability of 1-2 km to 26 km utilizing the UAV as a communications repeater similar to the method proposed by this thesis. In addition, Project Stork demonstrated the benefits of UAV-UGV collaboration for providing persistent surveillance information. While [16] does not mention the use of surveillance on-board the UAV, The helicopter deployed robot system suggested in this thesis enhances the benefits of having surveillance on the ground provided by the robot with an eye-in-the-sky surveillance vantage point provided by the UAV. As will be discussed in chapter 5, the vantage point of the UAV can be used to improve mission operations and ground robot control.

2.3 Beyond Line-of-Sight Robot Operation

Effective beyond line-of-sight robot operation is an important aspect of the proposed mission scenario as the robot will be operating in environments that are potentially harmful to humans, likely in an urban setting, where the operator will not be able to directly observe the robot. Some factors that affect effective BLOS operation are robot teleoperation, communications, and situational awareness.

The Joint Collaborative Technology Experiment[17] successfully demonstrated the use of a Yamaha RMAX vertical take-off and landing (VTOL) UAV to extend BLOS communications with multiple UGVs. In addition, a Mongoose UAV, an additional, but smaller VTOL platform was used to maintain persistent intelligence, surveillance, and reconnaissance (ISR) of the target area. The system implemented successfully demonstrates the use of VTOL UAVs to improve BLOS operations of a UGV by improving communications and situational awareness.

The system described by [17] differs from the implementation proposed in this thesis by using a separate UAV for communications and ISR. Multiple UAVs were required due to the mission objective and the use of multiple UGVs. The RMAX was tasked with maintaining an optimal control link between the unmanned systems and the ground control station (GCS). This required the RMAX to periodically adjust its location in order to maintain the optimal link. The Mongoose was used solely to provide persistent ISR for the mission commanders. The implementation proposed here-in of BLOS operation of the ground robot requires the use of a single UAV. The single UAV is tasked with both functions: to relay communications between a single UGV and the GCS, and to provide video surveillance only of the area in the immediate vicinity of the UGV, thus providing the UGV operator with additional situational awareness.

2.4 Supervisory Control for Effective Robot Operation

Supervisory control is defined to mean a human-robot interaction where the human sends commands to the robot and receives information from the robot, and the robot itself closes an autonomous control loop to carry out the commands[18]. Supervisory control is considered to be a level of autonomy somewhere between full operator control and full automation. Figure 2.1 provides an illustration of how supervisory control fits into the automation spectrum. While not always the case, supervisory control is often employed to reduce operator workload for one or more tasks and improve the efficiency of robot operations. As Figure 2.1

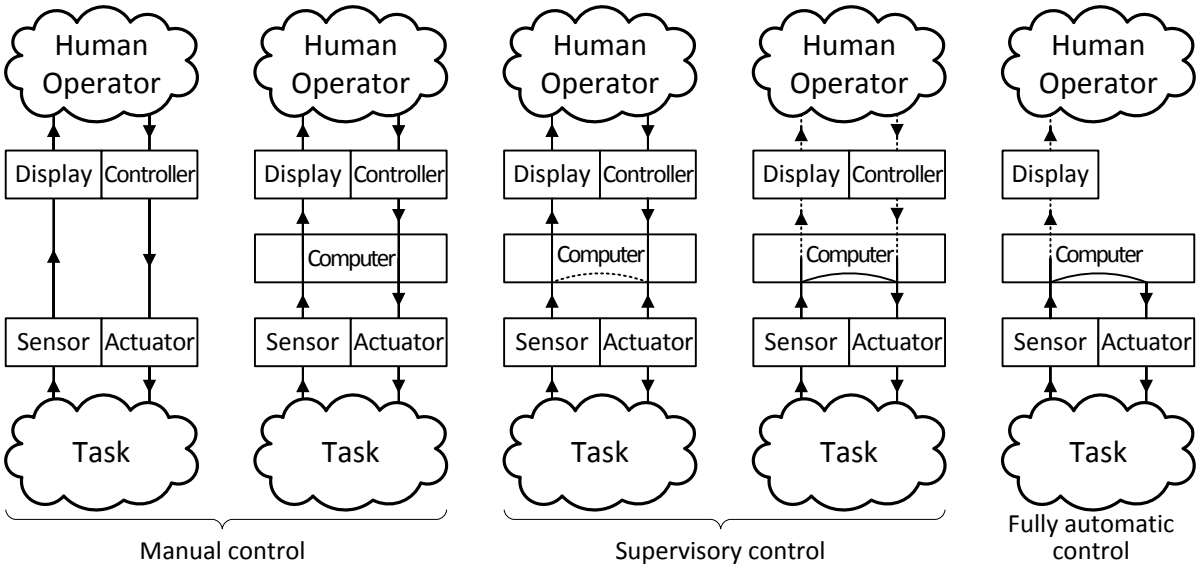


Figure 2.1: Graphical representation of varying levels of autonomy from full operator control to full automation[18] Used with permission

demonstrates, varying levels of supervisory control exist. At the lowest level, supervisory control could involve an operator issuing directional commands to drive a robot and a computer interpreting the commands and sending the appropriate control signals to the motors. This low-level supervisory control is most often referred to as *teleoperation*, where control decisions depend on the human operator, and if human control stops, the robot stops. At a higher level, supervisory control involves operator issued commands such as tasks, goals, or objectives, that are interpreted by the robot and carried out.

The authors of [19] discuss the benefits of supervisory control for improved robot operation. The strengths and weaknesses of both human and computer control are acknowledged, and a comparison of teleoperation with supervisory control is made, concluding with a discussion of the added benefits of supervisory control. Intelligent autonomy is proposed as a method of improving operator awareness of the environment and increasing efficiency by autonomously mapping the region traveled by the robot and autonomously recording sensor readings and alerting the operator if a pre-defined threshold is reached.

In addition, [19] proposes that an increase in situational awareness can be achieved by making use of a bird's eye perspective, achieved through the autonomous mapping. The limited field-of-view provided by the on-board navigation cameras of the robot limit the telepresence of the operator. While these on-board navigation cameras are important to providing certain details about the robot's environment, the addition of a bird's-eye-view helps to increase the situational awareness of the operator by providing the operator with a broader perspective of the robot's surroundings.

In addition to the enhancements listed above, the authors implement a point-and-click method of navigating the robot, where a target is indicated on the bird's-eye-view map and the robot autonomously navigates to the selected target. In addition to the point-and-click navigation ability, the operators are given the ability to take full control and teleoperate the robot. A user study conducted to assess the benefits of these enhancements shows that 90% of the users believed the point-and-click method of driving the robot was beneficial to robot operations. In addition, the robot operators suggested the bird's-eye-view was beneficial in providing an understanding of exactly where the robot was in relation to it's surroundings. While the testing conducted in the paper was done indoors, these same benefits can be realized by using a UAV to provide aerial surveillance of a robot, providing the UGV operator an increased sense of situation awareness in a hazardous environment.

Chapter 3

Ground Robot Design

This chapter addresses the design of the robot to be used for sample collection. The requirements and criteria dictating the design of the robot will first be discussed. The key design features that meet the design requirements will be presented and then an analysis of how the design helps to meet the overall mission goals will be given. The chapter concludes with a summary of the robot design and a list of suggested future improvements to the robot and a road map of the additional work that will be required for successful robot deployment.

3.1 Design Requirements and Criteria

The design of the ground sampling robot was dictated by two factors: the parameters of the flight platform and the intended operating scenario. In section 1.3, the CONOPS suggests that the robot will be delivered to the deployment site via UAV. The UAV chosen for this mission is a Yamaha RMAX helicopter (see Figure 3.1) equipped with a WePilot flight control system. Table 3.1 lists some of the physical specifications of the RMAX helicopter, and Figure 3.2 shows an inset of the RMAX landing gear explaining the dimensions in Table 3.1. The physical characteristics of the RMAX helicopter create a design envelope for

the size and weight of the ground robot. In addition to the total weight of the robot, the payload capacity of the RMAX defines the weight budget for other helicopter systems such as the robot pod, the collection of any physical samples by the robot, and the other permanent systems on board the helicopter. Upon examining the capabilities and performance data of

Figure 3.1: Virginia Tech's RMAX helicopter

Table 3.1: Physical specifications of the RMAX helicopter[20]. See Figure 3.2

RMAX Specifications	
Max Payload	28 kg
Endurance	60 min
Landing Gear Dimensions	
Length	540 mm
Width	510 mm
Height	260 mm

the RMAX platform, a set of design specifications for the size and weight of the robot were established and can be found in Table 3.2.

Next, the intended operating scenario was examined in order to define criteria for mobility

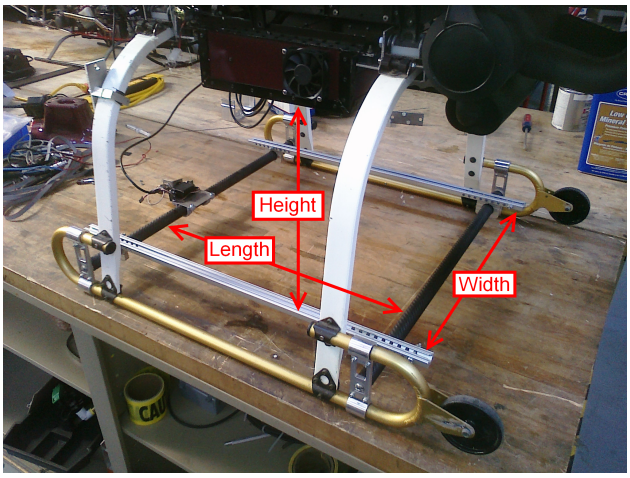


Figure 3.2: RMAX landing gear dimensions

performance and robot operations. Although the exact operating scenario for the robot is still unclear, it was suggested that various types of terrain may be encountered at the deployment sight, and that the robot should be fully capable in the worst case scenario. Because the robot is being designed for use in hazardous environments such as post-detonation scenarios or urban disaster search and rescue missions, there exists the potential for large variability in terrain types at the deployment site. For this reason, it was important to assess the potential performance of various mobility systems over a wide variety of terrain. Although many mobility platforms exist, including legged platforms, modified wheel platforms (such as the Boston Dynamics RHex), wheeled platforms, and tracked platforms, the two platform types considered for this application were wheeled and tracked platforms. This choice was made in order to promote design simplicity and eliminate complex mechanical systems for the sake of platform reliability. A good review of the trade-offs between wheeled and tracked platforms is provided in [21]. Specifically, [21] suggests that, when compared to wheeled vehicles, tracked vehicles have improved mobility, better traction on sloped terrain, increased maneuverability, and greater route flexibility. The use of a tracked vehicle does, however, impose additional difficulties such as a single point failure in the event of track breakage or derailment and added complexity from the standpoint of semi- or fully-autonomous navigation due to the uncertainties associated with skid-steer locomotion. It was determined that the ability to

traverse a variety of terrain and navigate obstacles is of higher importance than the speed of the platform. This design decision impacted the selection of the type of mobility platform and the drive train component selection. Based on these decisions, the design criteria for mobility were established and are presented in Table 3.2. The initial goal was not to design

Table 3.2: Desired robot design specifications

Desired Design Specifications	
Max Length	540 mm
Max Width	510 mm
Max Height	200 mm
Max Weight	8 kg
Traversable Slope (longitudinal)	45°
Traversable Slope (lateral)	35°
Ground Speed	0.25 $^{m}/_{s}$

a completely new mobility platform for the robot deployment mission, but to acquire a COTS platform and adapt it to the mission at hand. However, upon review of the physical specifications of the helicopter and of the various robotic platforms currently available from various manufacturers, none fit within the design requirements as specified by the mission. While many platforms exist that meet the mobility criteria established, many of the platforms exceeded both the size and weight limits imposed by the physical parameters of the RMAX helicopter. Upon further research, it was found that significantly smaller and lighter robots with similar capabilities as those required by our system are an area of focus of the Unmanned Systems Integrated Roadmap of the Department of Defense[22]. Upon completion of this assessment, it was determined that a completely new mobility platform would need to be designed in order to meet the current robot design specifications.

3.2 Key Design Features

The design of mobile robots is not new, nor is the design of tracked mobile robots. As such, this thesis does not contain a detailed discussion of the entire robot design. However, there exist key aspects of this design that will be discussed in this thesis due to their relation to fulfilling the design requirements established in Table 3.2. The discussion of the ground robot design has been broken up into three sections: drive train, chassis, and electronics & communications. A compilation of information regarding the robot design, including engineering drawings can be found in Appendix A. In addition, Figure 3.3 provides a brief overview of some of the key features that will be discussed in the following sections.

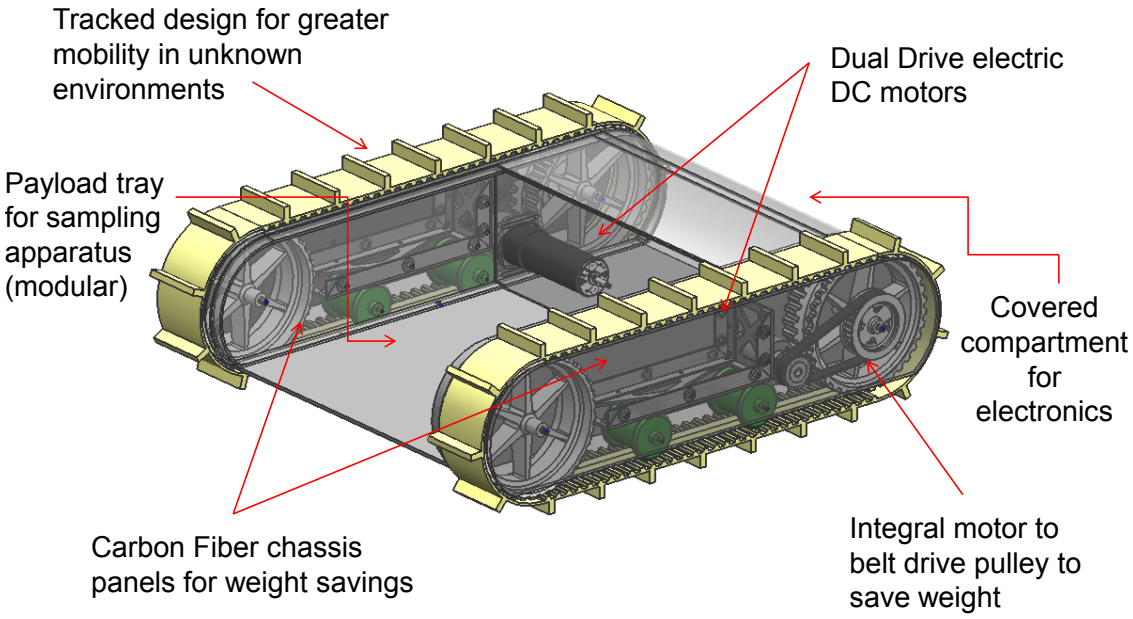


Figure 3.3: An overview of the key design features

3.2.1 Drive Train

This section discusses the key design features of the drive train. The drive train includes the electric motors, the track system including the drive and idler pulleys, and the belt and pulley linking the motor to the track drive pulley. Upon settling on the decision to design a tracked

robot the acquisition of a set of tracks to drive the robot became of utmost importance. Tracks for a small robot such as this one are not widely available. Most companies such as iRobot and Qinetiq that manufacture small tracked robots also manufacture a custom designed track for these robots. Some tracks are available for purchase, however none appear to be robust enough for use in the conditions for which this robot was designed. Appendix B contains a list of the various tracks that were investigated for use on the ground robot, most of which are designed for small, indoor, hobby robots. Instead of selecting a track designed specifically for use on a robot of this type the products of various timing belt manufacturers were investigated.

It was discovered that BRECOflex offers custom polyurethane timing belts. These belts are available in various widths, lengths and can be equipped with various weld-on profiles. All information regarding the BRECOflex product line is available at the BRECOflex website[1]. These belts are designed for use in power transmission systems or indexing systems in assembly lines, but have been used previously by others as prototype tracks for tracked vehicle designs. Although the tracks were not designed specifically for this purpose, they functioned reasonably well during vehicle testing. The belts utilize a standard metric tooth design with 10 mm pitch. In addition, a profile was added to the surface of the belt in order to give the robot added traction. An engineering drawing for the belt showing the tooth profile and the belt surface profile can be found in Appendix A.

In order to minimize the weight of the ground robot, a custom track drive pulley was designed. Although pulleys that interface with the track are available for purchase from the belt manufacturer, the pulleys are solid aluminum. While the manufacturer does offer the manufacturing of lightening holes for a fee, the amount of material that would need to be removed is cost prohibitive.

Figure 3.4 shows a CAD model of the final pulley design used on the robot. A spoked design was used as it provides for a light-weight design while maintaining rigidity and strength.

[1] http://www.brecoflex.com/

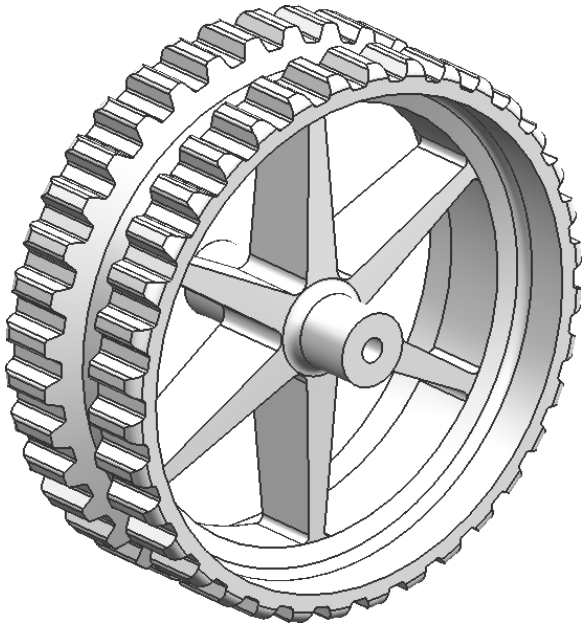


Figure 3.4: CAD model of the GSR track pulley

Due to the complexity of the pulley design, a rapid prototyping process was used to manufacture the pulleys. The Stratasys Titan Fused Deposition Modeling system was used to rapid prototype the pulleys from the Stratasys polycarbonate material. The polycarbonate material provides a part capable of withstanding functional testing. Stratasys provides a table of mechanical properties for this material. Some variation in the mechanical properties occurs due to the anisotropic properties that result from the method of building the rapid prototyped part. Figure 3.4 also shows that a self-tracking groove has been used at the belt/pulley interface in order to prevent track derailment.

Once the pulley design was generated, a method for interfacing the drive pulley with the drive motor was developed. The tracks are driven by the rear pulley in order to maintain tension along the terrain contact surface of the tracks. Both a gear drive and belt drive were considered for the motor/pulley interface, but a belt drive was chosen due to increased efficiency over a gear drive. In addition, the belt drive system provides for added weight savings over a gear drive system by eliminating heavy gears from the system and minimizing the motor size. The motor size is minimized by using a belt drive because the motor does

not need to be oversized in order to compensate for the transmission losses caused by the inefficiency of the gears. The smaller motor size leads to a lighter design, thus contributing less to the overall weight budget of the system.

The belt drive system was implemented by designing an integrated pulley into the track drive pulley as shown in Figure 3.5. The design shown in Figure 3.5 allows for additional

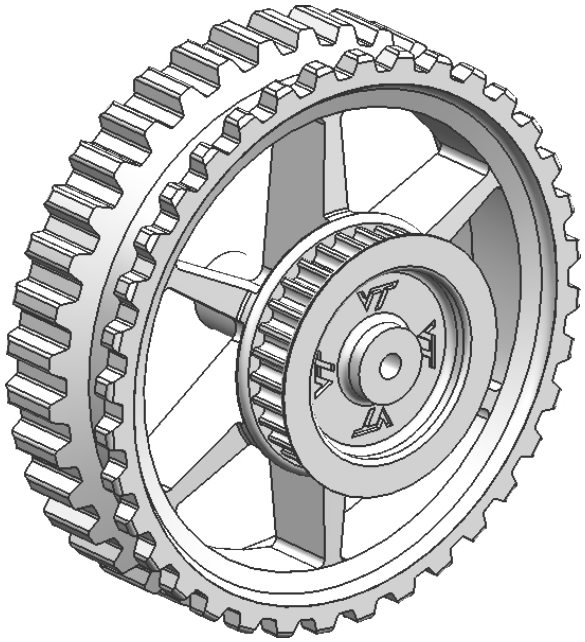


Figure 3.5: CAD model of the rear drive pulley

weight savings. Once again, this pulley design was manufactured using the same method as the pulley shown in Figure 3.4 which is used as an idler pulley at the front of the robot. The belt drive utilizes a 5 mm pitch HTD tooth profile and provides a 2:1 speed reduction from the motor to the drive pulley. The smaller pulley for the belt drive is situated on the outward side of the robot as shown in Figure 3.6. In order to ensure proper spoke sizing at the location where the belt drive pulley interfaces with the track drive pulley, analysis of the stresses was performed and is located in Appendix C.

Figure 3.7 provides a top view of the belt drive assembly. From this view we see that the belt drive system is contained entirely between the belt cover at the top of the image and the chassis of the robot at the bottom of the image. The belt cover and chassis will be discussed

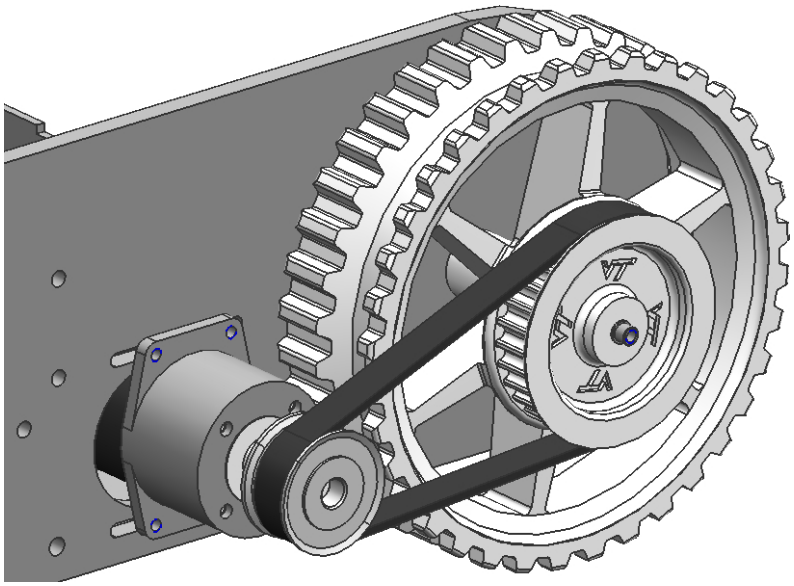


Figure 3.6: CAD model of the belt drive assembly. A plate covering the belt and pulleys has been removed for the purpose of this image.

further in section 3.2.2. While the belt pulley that is integrated into the track drive pulley is made of polycarbonate, the belt pulley that interfaces with the motor drive shaft is a conventional aluminum pulley, attached to the motor drive shaft via a set screw.

As mentioned above, by properly sizing the drive motors and avoiding oversized motors, the weight of the robot can be minimized. The design specifications from Table 3.2 were used to properly size the motors. The sizing of the motor is largely a function of vehicle weight, maximum traversable slope, wheel diameter and velocity. These parameters are used to calculate the torque required to drive the robot and the angular velocity at which the motor should operate. The motors were sized to be able to propel the robot up a minimum slope of 45° at a minimum velocity of 0.25 $^{m}/_{s}$. A detailed explanation of the motor sizing calculations is presented in Appendix D. From these calculations, it was determined that the drive train would need to supply a minimum of 4.6 Nm in order to meet the design criteria. While this value represents the total required torque, this value was used to size each motor in order to provide a safety factor of two. The task of finding a motor and gearhead combination that was suited for this application involved researching the products offered by various motor

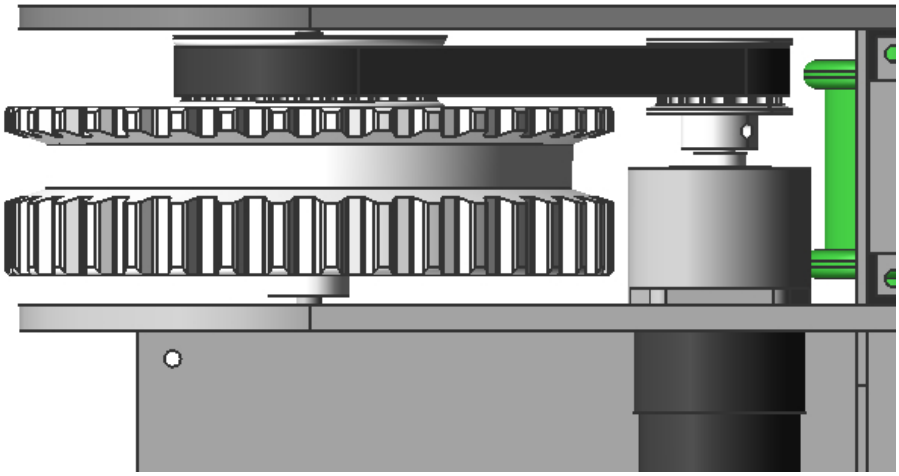


Figure 3.7: Top view of the belt drive assembly

manufacturers. Various motor suppliers were considered for use on this robot; however, Maxon Motor was chosen as the supplier for the motors. Maxon Motor provides a very good motor selection program that simplifies the motor selection process. The user is required to input the desired design criteria and the program returns a list of possible motor and gearhead combinations that are suited to the application. A more detailed report of the motor selection is also provided in Appendix D, including calculations verifying that the selected drive train components meet the design criteria.

3.2.2 Chassis

A light-weight rigid chassis was designed by using composite materials and aluminum. Figure 3.8 is an exploded view representation of the sandwich composite material used to construct the belt covers and the chassis side plates. The sandwich material consists of three layers of carbon fiber, a 1/8" balsa wood core, and three more layers of carbon fiber. Aluminum inserts were embedded in the balsa wood core at the axle mounting points and the motor mounting point in order to add extra stiffness and to prevent damage to the composite materials from fasteners. The sandwich composite material provides for a lightweight, rigid structure. The belt covers and chassis side plates will see mostly bending moments and

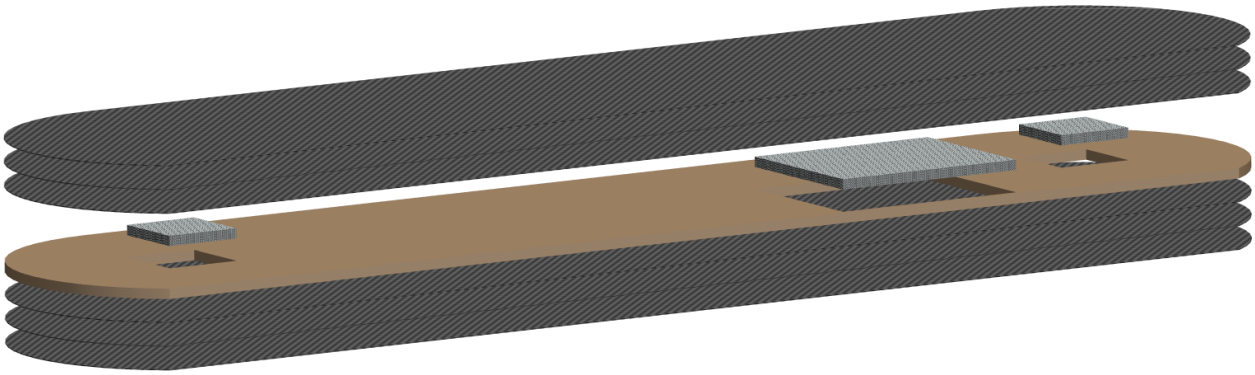


Figure 3.8: Exploded view of composite chassis plates

torsion forces. For this reason the sandwich material is an ideal design as little to no stress will be seen along the neutral axis of the material. The balsa wood core provides a light weight material to separate the carbon fiber and move it away from the neutral axis where the stresses will be higher.

The chassis side plate and belt cover are connected (as shown in Figure 3.9 by the truss-like sheet metal structure shown in Figure 3.10, which serves as a tray to house the batteries that power the robot. This structure increases the rigidity of the robot structure by preventing the chassis side plate and the belt cover from moving longitudinally and vertically in relation to one another. The battery tray also provides for proper spacing between the belt cover

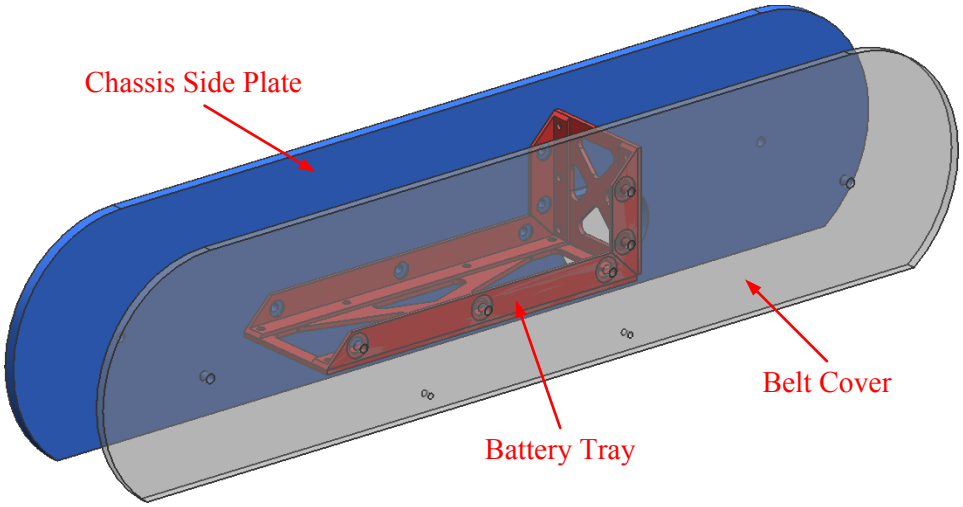


Figure 3.9: A representation of the battery tray's location in the robot chassis.

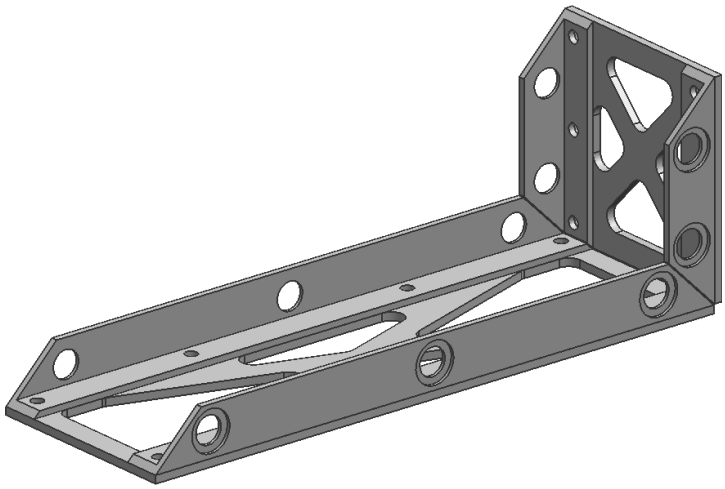


Figure 3.10: A battery tray that also gives added rigidity to the robot.

and chassis side plate in order to prevent any unnecessary friction that would be caused if one or both of the plates were to flex inward toward the tracks. In addition, the added stiffness keeps the complete track drive assembly in alignment which is important for good robot control and mobility. The mobility of the robot is also improved by housing the batteries in each of the battery trays, which increases the moment of inertia of the robot. This makes the robot better balanced and more stable which helps reduce the risk of rollover and allows the robot to traverse increasingly rough terrain. Furthermore, this design feature makes use of the otherwise unoccupied area inside the tracks and frees up space in the main bay of the robot for other robot components such as the data collection system.

3.2.3 Electronics and Controls

This section presents the electronics and controls architecture for the ground robot in their current state and presents a list of future improvements to be added to the design. At this point, the electronics and controls for the ground robot are fairly simple. The electronics consist of four basic components: an XBee radio, an Arduino Mega microcontroller board, a Sabertooth 2x25 motor controller and a 206W axis camera. This combination of electronics provides a light, compact package with the capability of fulfilling all of the processing and

control requirements of the robot. While the ultimate goal is to relay the robot communications from the ground control station (GCS) through the helicopter and down to the robot, the communications are currently implemented directly from a laptop computer to the ground robot via two XBee radios. Eliminating the helicopter from the communication architecture at this point in the design process allows for greater ease in debugging of the controls and testing of the robot's capabilities. Figure 3.11 provides a schematic of the electronics and communications system.

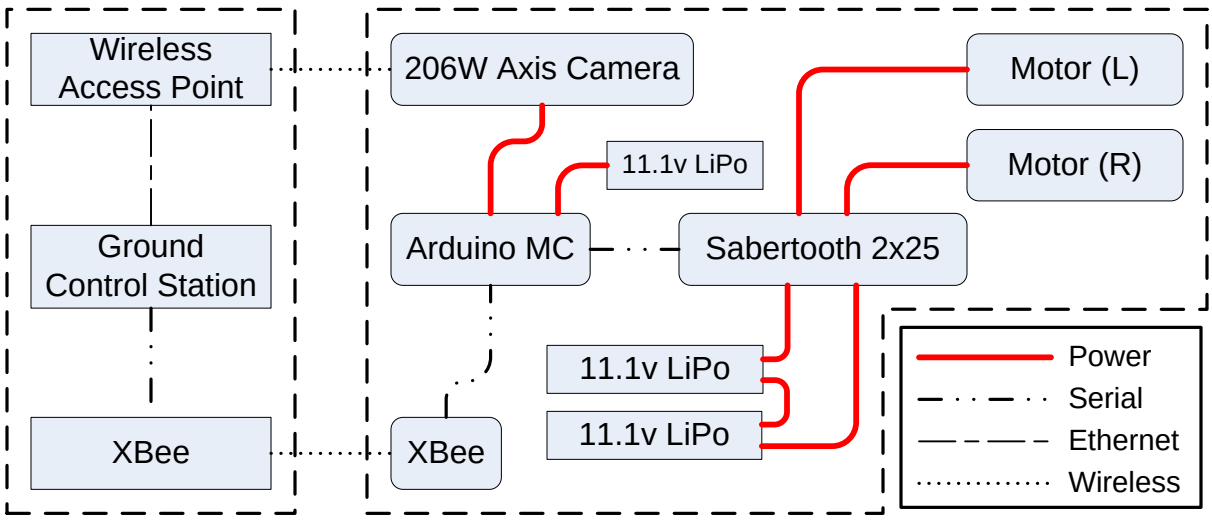


Figure 3.11: Electronics and communications schematic

Arduino is an open-source prototyping platform based on the Atmel ATMega series of microcontrollers. The Arduino platforms exist in many variations, each with different features and peripherals. All of the pins of the ATMega1280 are pinned out to headers on the board which makes for a very versatile development platform. The Arduino Mega was chosen for its use of the ATMega1280 microcontroller, which provides 54 digital input/output pins, 14 of which can be used to output a pulse-width modulated (PWM) signal. In addition, the Arduino Mega provides four hardware serial ports and 15 analog inputs. The Arduino Mega is used to parse the incoming commands from the GCS and send the appropriate command to the other robot electronics. In addition, the Arduino Mega is populated with a DC to DC voltage regulator which provides both a 5v and 3.3v output. The 5v output from the

regulator is used to power the Axis camera on-board the robot.

The XBee radio utilizes the ZigBee protocol for communication. The wireless link is a 2.4 GHz link. Simple serial commands are used to communicate with the device. As shown by Figure 3.11, the XBee radio interfaces directly with one of the Arduino serial ports. For robot mobility control, the Arduino receives the serial commands from the XBee, parses the commands, and sends the appropriate command to the Sabertooth 2x25 motor controller.

The Sabertooth 2x25 motor controller is a product of Dimension Engineering. The motor controller operates between 6-30v and is capable of outputting 25A continuous and 50A peak. The Sabertooth motor controller has the following input modes: analog (0-5v), R/C, simple serial, and packetized serial. A detailed explanation of how to use each of the input modes can be found in the Sabertooth 2x25 literature on the Dimension Engineering website[2]. Each of the four input modes is toggled by adjusting the settings on a dipswitch block located on the Sabertooth board.

The two input modes used to control the robot are the R/C mode and the packetized serial mode. Two input modes were used during different test phases due to their distinct benefits. The R/C signal mode is easy to use and has the advantage of working with a handheld transmitter with no additional electronics or computers required. This allows for quick and easy testing of the robot's capabilities. The disadvantage of R/C mode is that there is no error checking. If the Arduino sends the wrong signal to the Sabertooth, the command will be interpreted and used, possibly resulting in undesirable robot operation. The packetized serial mode resolves the error checking issue.

Controlling the motor controller with the R/C signal can be done by connecting a standard R/C receiver to the appropriate input pins on the Sabertooth and sending commands from an R/C transmitter, or by simulating an R/C signal using the PWM output pins of the Arduino board. Both transmitter and simulated signal modes are interchangeable and allow for easy control of the robot. The R/C signal setting uses two channels and offers the option

[2]http://www.dimensionengineering.com/

of an independent mode or mixed mode. Independent mode allows each channel to control a single motor. Mixed mode allows for easy control of the two motors in the differential drive configuration used by the robot by assigning one channel to forward and reverse commands and the other channel to left and right turning commands.

Packetized serial mode uses a simple protocol to send commands to the Sabertooth motor controller. This allows for error checking and for multiple Sabertooth motor controllers to be controlled by the same microcontroller board if the need arises. Just as with R/C mode, packetized serial mode allows for motor operation in an independent or mixed mode. However, where independent or mixed mode is a dipswitch setting when using the R/C signal, independent or mixed mode can be used interchangeably with the packetized serial protocol without changing dipswitch settings, because each mode uses a different set of command bytes to transmit a packet. Additionally, because the packetized serial setting only uses one of the signal inputs on the Sabertooth for communication, the second signal input pin provides for the implementation of an emergency stop if the feature is needed. This is a desirable feature in the event that the robot ground operations begin to adversely affect the stability of the helicopter.

Although the electronics of the robot are quite simple during this phase, the use of the Arduino Mega, the Sabertooth Motor Controller, and the XBee radio provide a compact electronics package, but still allow for significant expansion. The next phase of the robot will be equipped with a sampling and data collection system that will be controlled by the Arduino Mega. In addition, the robot will be equipped with various sensors, including attitude sensors, environmental sensors, and health monitoring sensors, all of which will be parsed by the Arduino and relayed back to the operator at the ground control station via the XBee radio. The sensor information will aid the operator of the robot by providing additional situational awareness that will help to complete the assigned task in a quick and efficient manner.

3.3 Robot Testing Results

This section discusses an evaluation of the robot design and presents the results of robot testing. Upon completion of the robot design and construction, several aspects of the robot's performance were evaluated to determine if the desired design specifications were met and for use in the planning algorithm as discussed in chapter 4.

3.3.1 Qualitative Mobility Assessment

The first set of evaluations were completed to make a qualitative assessment of the robots performance on various types of terrain. Because good mobility on a variety of terrain is an important design condition, the robot was tested on various terrain types. The testing included both flat and sloped terrain, and the robot was operated in both straight-line motion and a variety of turning motions. The turning motions included zero-degree radius turns in order to evaluate the effect of skid-steering on robot performance.

Table 3.3 provides a summary of the mobility testing. This mobility testing was completed in the 4.45 kg robot configuration in order to verify the proper function of the drive train components, assess the need of any significant design changes, and determine any possible failure points in the design.

Table 3.3: Qualitative results of mobility testing over varied terrain and surface types

Surface Type	**Flat**		**Slope**		
	straight	*turn*	*climb*	*traverse*	*turn*
Dirt	5	5	5	5	5
Gravel	4	3	3	4	1
Grass	5	5	4	4	4
Pine needles	3	3	3	3	3
Concrete/Asphalt	5	4	5	5	4
Debris Pile	5	3	5	4	3

The performance of the robot was rated on a scale from 1-5, 1 being a poor performance and

5 being a successful performance. The robot appeared to perform the best on dirt. From Table 3.3 it can be seen that the robot's turning performance was quite poor on the gravel surface. This is largely due to pieces of gravel getting caught between the track and pulleys, in some cases impeding the motion of the robot entirely. A solution for this will be discussed in section 3.4. The performance of the robot on grass was quite good, but the performance degraded slightly on sloped grass due to an apparent loss of traction as the slope increased. The pine needle laden terrain appeared to cause some trouble for the robot due to a loss of traction as well, on both flat terrain and sloped terrain. The loss of traction was largely dependent on the thickness of the pine needle layer. Once the robot was able to contact the underlying terrain, performance increased.

The robot performed well on concrete and asphalt surfaces with a slight drop in performance when turning. The corners of the robot track grousers appeared to catch on the asphalt when turning due to the nature of the skid-steer turn. This is an artifact of the track design which will also be discussed in section 3.4. The robot performed quite well on the debris pile, with the exception of turning. The large variations of the debris pile allowed the treads to gain traction and navigate straight line motion well. Although a less than satisfactory rating was given to the robot's ability to turn on the debris pile, the robot was able to turn successfully. The poor rating was given because the robot encountered difficulty while turning on the debris pile due to the unstructured nature of the surface features. Unlike the other surfaces where the operator was able to issue a command to the robot to turn and the turn occurred, the debris pile required the operator to navigate forward or backward slightly while turning in order to prevent the robot from getting stuck on the varied terrain. Although the robot encountered some difficulty while navigating the debris pile, the overall performance of the robot was quite good considering that the debris pile presented the most difficult surface type for perhaps a wheeled robot to navigate of all the surfaces tested.

3.3.2 Performance Testing

After completing the qualitative mobility testing, a study was completed in order to determine whether or not the robot design complies with the mobility performance criteria established in Table 3.2. The key goals of the study were to determine the maximum climbable slope, velocity at the maximum climbable slope, maximum traversable slope, and the maximum speed of the robot on flat terrain.

Experimental Setup

Figure 3.12 labels the heading of the robot that defines both the climbable slope and the traversable slope. The slope of the platform is adjustable, with the following possible angles: 0.0°, 7.0°, 12.0°, 16.5°, 21.0°, 25.0°, 29.0°, 33.0°, 37.5°, 41.5°, 45.0°, 50.0°, 54.5°, 59.0°, and 62.5°.

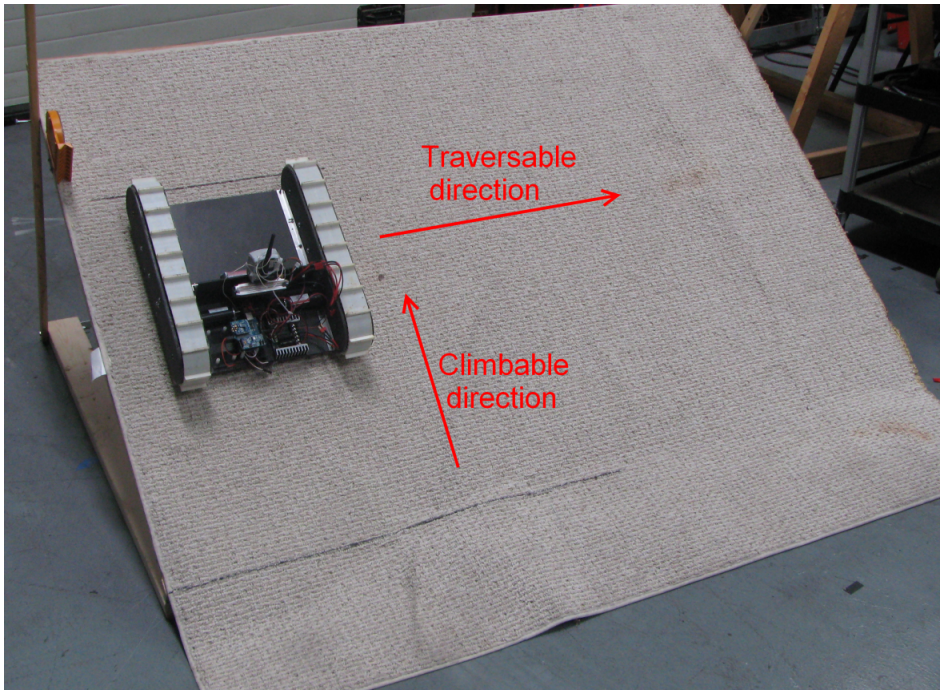


Figure 3.12: Test platform used to determine performance characteristics of final robot design

The goal of the in-lab testing was to assess the capabilities of the motors and other drive train

components under ideal conditions, whereas the qualitative mobility test assessed the robot's capability to navigate different terrain types. The test platform was covered in carpet in order to provide a high-friction surface. As mentioned previously in section 3.2.1, the tracks used for this prototype of the robot are not ideal, and the high-friction surface provided by the carpet aids in reducing the effects of the track design on the study by reducing the possibility of track slippage. In addition to the setup of the platform, some additional weight was added to the robot in order to simulate the addition of a sampling system and other sensors. The goal was to determine the performance parameters of the complete system, and as Table 3.2 suggests, the robot should have a maximum weight of 8 kg. The current weight of the robot is 4.47 kg. Enough extra weight was added to the payload bay of the robot to increase the overall weight to 7.2 kg, just shy of the 8 kg weight limit.

Experimental Procedure

In order to collect the required performance data the robot was driven up the slope in the climbable direction and across the slope in the traversable direction as indicated in Figure 3.12. When moving in the climbable direction, the time required to travel the length of the platform was recorded using a stopwatch. For each trial, the back of the robot was aligned with the black line at the bottom of the platform, and the robot was started from rest. The stopwatch was started simultaneously with the robot motion and was stopped when the front of the robot reached the top of the platform. In addition to recording the time, notes were recorded for each trial.

A trial in the climbable direction was marked as successful if the robot was able to reach the top of the platform without slipping backward more than 20 cm during motion or while the robot should be at rest. Five trials were completed at each platform angle. Although the lithium-polymer batteries used to power the robot maintain a relatively constant voltage during a discharge cycle, the order in which each trial occurred was randomly selected, and the batteries were recharged after each 30 minutes of operation in an effort to minimize the

effects of any variation in battery voltage levels on the results.

A similar procedure was used to test the robot's performance in the traversable direction. This study was a test of the robot's stability and ability to remain on the slope while in motion. The time required to travel the distance of the platform was not recorded for trials in the traversable direction. A successful trial was recorded as one in which the robot did not drift more than 20 cm down the slope from start to finish, independent of whether the drift occurred during forward motion or while the robot was at rest.

Experimental Results

A summary of the results collected during the study is presented in Table 3.4. From this table we see that the maximum traversable slope in the longitudinal direction (climbable direction) is 50° and the maximum traversable slope in the lateral direction (traversable direction) is 37.5°. These results show that the robot design meets the design specifications from Table 3.2.

In order to determine the maximum velocity at the maximum climbable slope and the maximum velocity on flat ground, the length of the platform traversed was divided by the times recorded for each of the trials during the robot performance study. The velocity for each trial was then used to determine an average velocity at each slope. A plot of the recorded and average velocities is presented in Figure 3.13. From the velocity data, we see that the maximum velocity on flat terrain is $0.35\,^{\mathrm{m}}/_{\mathrm{s}}$ and the maximum velocity while climbing a slope of 50° is $0.25\,^{\mathrm{m}}/_{\mathrm{s}}$. These values are valid for the carpeted test surface and will vary for other surfaces; however, the testing provides an estimate of the performance that can be expected from the robot. In addition to the performance data collected during this study, some additional information has been gained regarding the strengths and weaknesses of the robot.

During testing, the track design proved to be one of the main limitations of the robot. The

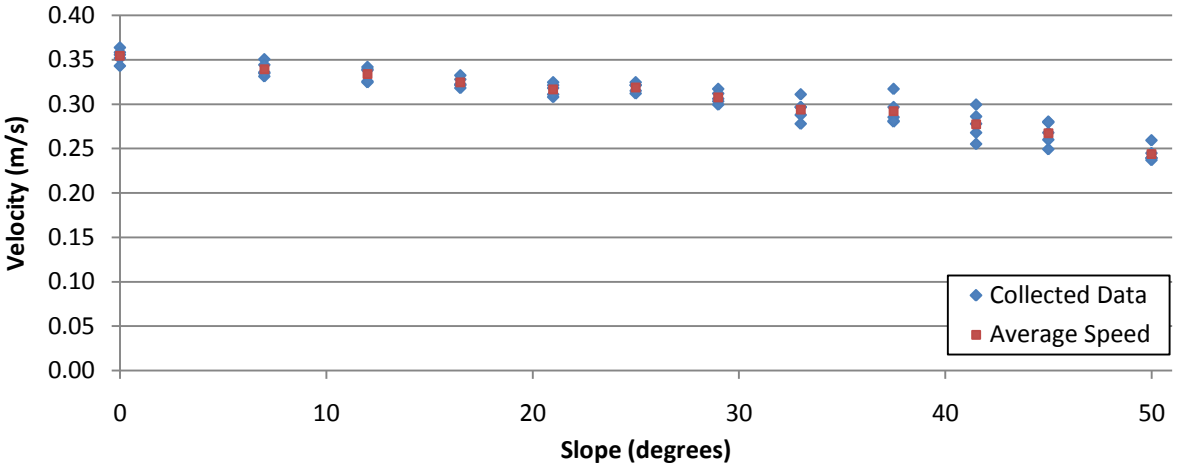


Figure 3.13: Velocity of robot on various slopes

tracks lost traction before the motor limits were reached, preventing the full torque of the motor being transmitted to the ground through the tracks. The loss of traction occurred when testing the robot on slopes greater than 50°. However, in some trials the robot was capable of climbing slopes of up to 62.5°. Data for these trials is not shown due to a violation of the drift criteria established in the Experimental Procedure section.

3.4 Final Design

Upon completion of the robot testing, it has been determined that the robot design is capable of meeting the objectives for which it has been designed from a mobility standpoint. Figure 3.14 presents a CAD model of the complete robot design. A photograph of the completed robot is presented in Figure 3.15. During the course of the robot design, evaluation, and testing, it has been determined that some additional improvements can be made to the robot. Some of these have been noted above, and others have not been mentioned.

As mentioned in section 3.3, the robot periodically collects debris in the track well when driving on loose terrain such as gravel. This is a cause for concern as it occasionally impedes the motion of the robot entirely. The proposed solution to this problem is the addition

Table 3.4: Robot specifications & performance data

Robot Design Data	
Physical Specifications	
Length	455 mm
Width	395 mm
Height	145 mm
Weight	4.45 kg
Performance Data	
Max. Traversable Slope (longitudinal)	50°
Max. Velocity @ 50° slope	0.25 $^{m}/_{s}$
Max. Velocity @ 0° slope	0.35 $^{m}/_{s}$
Max. Traversable Slope (lateral)	37.5°

of a debris skirt at the bottom of the belt cover and chassis side plate. The debris skirt would consist of a few rows of bristles that would hinder debris from entering the track well. In addition, another set of bristles could be located inside the track well near each pulley, in a configuration similar to a snow plow, in order to push any debris off of the inside of the track before reaching the pulley. Another solution, perhaps more robust, would be to redesign the track and pulley system. As mentioned previously, the tracks being used are a prototype solution only, and redesigning the track system would be beneficial from a traction standpoint as well.

Another issue that was occasionally observed is the slipping of the belt that connects the motor to the rear drive pulley. The belt slippage occurred due to occasional failure in the design of the tensioning mechanism. The failure is perceived as both beneficial and detrimental to the design of the robot. The slippage benefits the robot in the sense that it acts as a clutch when excessive amounts of torque are being transfered. The improvised clutch can prevent damage to the motors or the on-board circuitry due to excessive current draw. By improving the motor/rear drive pulley/belt interface the robot would be capable of carrying additional weight, up possibly steeper terrain, removing the detrimental effects

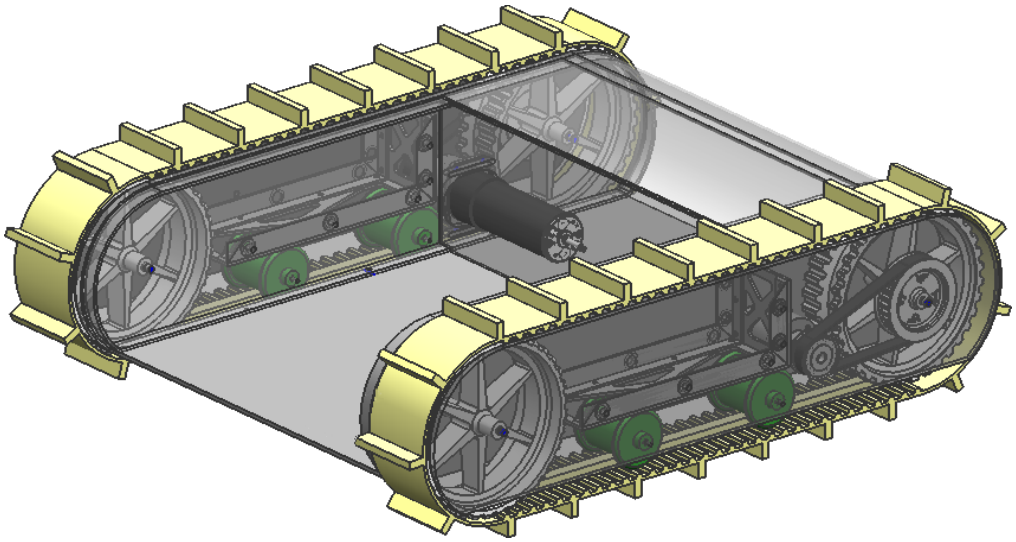


Figure 3.14: CAD rendering of the final robot design

of the slipping belt, therefore improving the robot design.

The improvement to this belt system could come in two ways: (1) an improved tensioning system, or (2) utilization of a belt and set of pulleys with a larger pitch. The second option may be more desirable as less tension would be required to keep the belts in place and prevent slipping. An increase in tension would increase the friction force in the system, requiring the motors to output more torque and draw more power[3]. When the belt did slip, the slip only occurred on inclines greater than 45° and was observed for the first time after roughly 10 hours of operation. Because 10 hours of operation well exceeds the endurance of the RMAX, the belt slip may not be of concern, yet still remains as a possible point of failure.

In addition to the improvements listed above, there exists some continuing work that will need to be completed in order to make the robot field ready. The sampling system and sensor suite for the robot has not been selected or designed. In addition, the integration of the robot onto the UAV platform must be completed. These items are outside of the scope of this thesis; however, they will be addressed in subsequent research. Furthermore, proper

[3]In an effort to reduce the friction of the robot system a set of low-cost, light-weight, plastic bearings have been used at the pulley/axle interface. The bearings used are manufactured by Igus, Inc. and have a dynamic coefficient of friction against steel of $\mu = 0.15$.

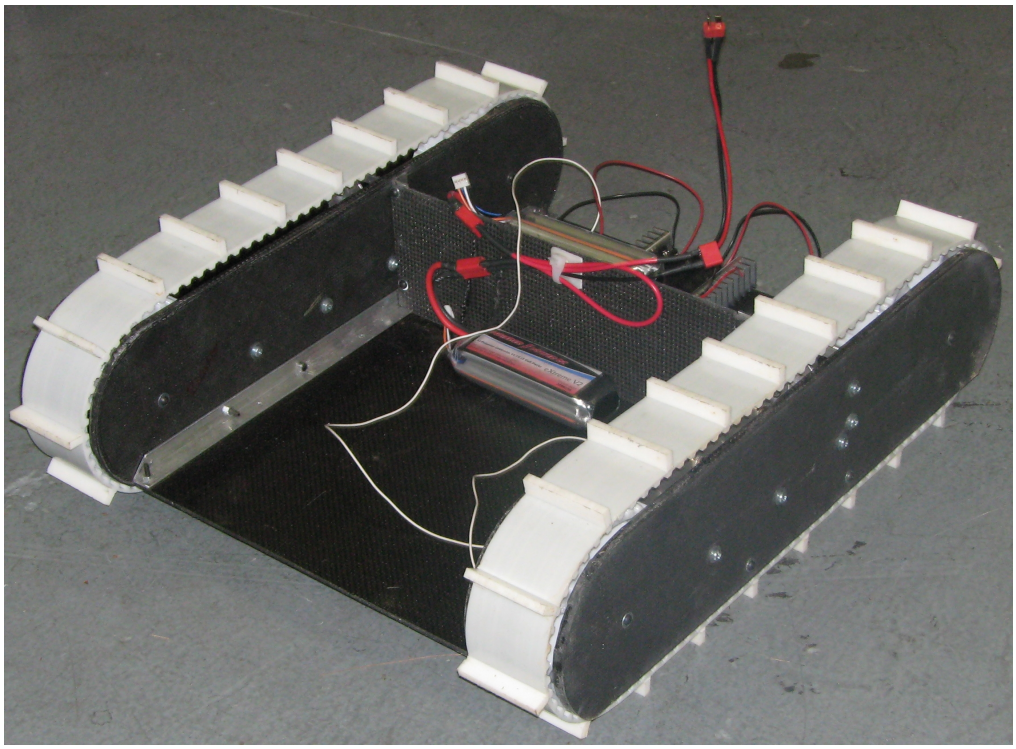


Figure 3.15: The completed robot

camera placement on the robot is of critical importance for successful teleoperation. The use of a wireless Axis camera was mentioned in section 3.2.3. Chapter 5 will discuss the work that has been done in determining proper camera placement for successful teleoperation as well as discuss additional ways to improve the overall situational awareness of the operator of a teleoperated vehicle.

Chapter 4

Path Planning for Ground Robot Navigation

This chapter focuses on the work completed toward developing a path planning algorithm that can be used to plan and present a path to the operator of the ground robot. The planned path can be provided to the operator as a navigational aid or could be used for autonomous waypoint navigation. The goal is to allow the operator to pick an area of interest and let the algorithm plan a path to the area of interest based on the robot starting position. By planning a path and presenting that path to the operator, the amount of time required to navigate the debris field will hopefully be reduced, thus reducing the overall time required for the sampling mission.

4.1 Path Planning Requirements

Due to the size and power limitations of the ground sampling robot, a path planning algorithm has been developed to assist the robot operator in navigating the debris field. The two main limitations of the data collection mission are time, due to the need for quick data

acquisition, and power, due to limited endurance of the UAV and the ground sampling robot. Locating and acquiring appropriate data in a timely manner is of utmost importance. Therefore, it is important that the operator be able to navigate the area of interest in a minimal amount of time. By planning a path for the operator prior to deployment, time will not be needlessly wasted searching for a path, because the algorithm will already have done this.

An additional concern is the power requirement of the robot, and the amount of the weight budget taken up by the power source. As was discussed earlier in section 3.2.3, the robot's electronics operate on battery power. By planning a path for the robot prior to deployment and presenting that plan to the operator, the power consumption of the robot can be reduced by eliminating the need for a wandering search for the target. The power consumption of the robot was not taken into account directly by the path planning algorithm as will later be discussed; however, by reducing run time, power consumption can be indirectly reduced.

4.2 Path Planning Algorithm Development

The algorithm that has been implemented for path planning is based on a basic forward searching algorithm[23] in the discrete domain. The discrete domain was chosen for the path planning algorithm due to the readily available 3-D stereoscopic terrain data acquired by the helicopter during the 3-D terrain mapping portion of the mission as outlined in section 1.2. While stereoscopic imagery is not a major focus of this thesis, it will be touched on briefly in section 4.2.1 in order to detail some of the required preparation of the stereovision data for use in the planning algorithm. The discrete domain planning problem requires a set of states called the *state space* and a set of actions called the *action space*. The nomenclature for the discrete domain planning problem is defined below:

1. N is a nonempty *state space*
2. For each *state* $n \in N$, a finite *action space* $U(n)$ exists
3. A *state transition function* f produces a state $f(n, u) \in N$ for every $n \in N$ and

$u \in U(n)$. The *state transition equation* is derived from f as $n' = f(n, u)$ where n' is referred to as a *successor* to n.

4. The *initial state* is defined as $n_I \in N$
5. A *goal set* is defined as $N_G \subset N$

In the above formulation, *states* can also be referred to as *nodes*. This formulation leads to the discrete domain forward search algorithm presented in Algorithm 4.1.

Algorithm 4.1 Forward Search [23]

```
1: Q.Insert(n_I) and mark n_I as visited
2: while Q not empty do
3:    n ← Q.PickNode()
4:    if n ∈ N_G then
5:       return SUCCESS
6:    end if
7:    for all u ∈ U(n) do
8:       n' ← f(n, u)
9:       if n' not visited then
10:         Mark n' as visited
11:         Q.Insert(n')
12:      else
13:         Resolve duplicate n'
14:      end if
15:   end for
16: end while
17: return FAILURE
```

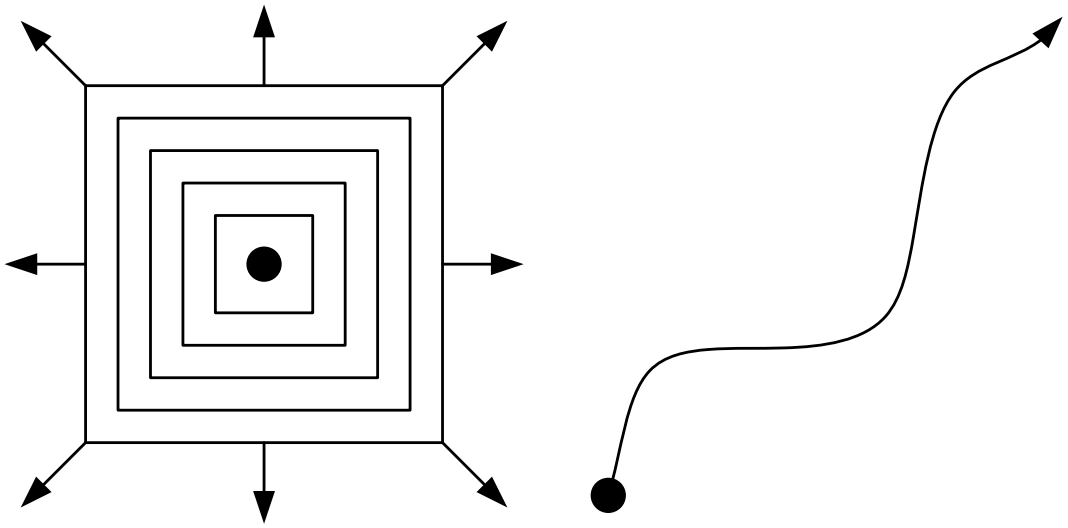
In Algorithm 4.1, there are three possible states or nodes: unvisited, dead, and alive nodes. An unvisited node is a node that the algorithm has not discovered yet. A dead node is a node that the algorithm has visited and has also visited all possible successors to that node. The alive nodes are the discovered or visited nodes for which the algorithm has not expanded all of the successor nodes. As defined above, a node n' is the *successor* to the current node n if there exists an *action* u from n to n'. In this algorithm a *priority queue* Q is kept of all alive nodes. As new nodes are discovered they are inserted into Q. To begin the algorithm, the only alive node is the start node (n_I), and as such, is inserted into Q. The algorithm

then picks a node from Q (various methods exist for picking the best node from the queue as will be discussed later). All of the successors to the new node are discovered by applying the state transition function $f(n, u)$ that provides a path from n to n'. Each new successor is then inserted into the queue if it has not previously been discovered. The algorithm then returns to Q to determine which node from Q should be the next node to be expanded. This process repeats until the queue is empty or until an element of the *goal set* is discovered, where N_G is the set of all goal nodes.

Various forward search methods exist, some of which include: breadth-first search, depth-first search, Dijkstra's algorithm, and A-star (A*). The difference between each of the search methods is the way in which a particular method expands through the graph by selecting elements from the priority queue as mentioned in the explanation of Algorithm 4.1. The breadth-first search and depth-first search methods use a simple method of selecting elements from the priority queue based on when the element entered the priority queue. The breadth-first search method uses a first-in first-out (FIFO) approach for the $Q.PickNode()$ function of Algorithm 4.1, and thus expands nodes in a wavefront pattern. The depth-first search method on the other hand uses a last-in first-out (LIFO) approach for the $Q.PickNode()$ function of Algorithm 4.1, which as the name suggests, causes the algorithm to search each successor node deeper, before expanding other nodes. Figure 4.1 represents the affect that each of these two methods of picking elements from the priority queue has on the outcome of the forward searching algorithm.

While breadth-first search and depth-first search[1] methods use a systematic approach to providing a solution to the path planning problem if a solution exists, they are expensive in both computation time and memory due to the brute force method in which nodes are expanded. In addition, these two methods merely provide a solution and do not guarantee that it is the best solution.

[1]Depth-first search is only systematic if the data set is finite. If the data set is infinite, depth-first search will explore infinitely deeper, while ignoring the rest of the graph.



(a) Breadth first search (b) Depth first search

Figure 4.1: A pictorial representation of node expansion using two different Forward Search Methods[23] Used with Permission

Alternatively, Dijkstra[24] presented a best path solution to the path planning problem known as Dijkstra's algorithm. Dijkstra's algorithm does this by calculating and minimizing the cost of progressing from the start node n_I to a node n. As Algorithm 4.1 suggests, Dijkstra's algorithm starts with an initial node n_I. The successors of n_I are expanded and the costs of arriving at the successor nodes is recorded. This cost is said to be

$$g(n') = g(n) + g(n, n')$$

where $g(n)$ is the cost from n_I to n, and $g(n, n')$ is the cost of progressing from n to n'.

At the start of the algorithm, $g(n) = 0$ as progressing from n_I to n where $n = n_I$ costs nothing. Although it is not mentioned in Algorithm 4.1, the initial node n_I is marked as dead and will not be revisited. The next step is the function $Q.PickNode()$, which for Dijkstra's algorithm picks the node contained in the priority queue with the minimum cost. The successors for this new node n are found and the costs $g(n')$ are calculated. If n' has never been visited, the calculated cost is recorded and n' is inserted into the priority queue.

However, if n' has previously been visited, the algorithm must determined whether the new path to n' or the old path to n' has a minimum cost as suggested in line 13 of the algorithm. The algorithm then returns to the $Q.PickNode()$ function, picking and expanding the node from Q with the least total path cost and continues until all nodes of the graph have been searched. By always picking the least cost path to a particular node in the graph, it is guaranteed that Dijkstra's algorithm will find the optimal path to any node in the graph. A detailed proof of the optimality of Dijkstra's algorithm can be found in [25]. While expansion of all graph nodes provides an optimal path to every node in the graph, the algorithm can be terminated once any or all nodes contained in the goal set N_G have been found.

An extension of Dijkstra's algorithm is the A* algorithm. In an attempt to force A* to expand far fewer nodes than Dijkstra's algorithm, the A* algorithm is directed by a cost + heuristic function. Again, the cost is denoted $g(n)$ and the new heuristic is denoted $h(n)$. As the nodes of the graph are expanded, the node with the minimum $f(n)$ value is picked by the $Q.PickNode()$ function as the next node to be expanded, where:

$$f(n) = g(n) + h(n) \tag{4.1}$$

The calculation of the cost function for A* is the same as for Dijkstra's, while the heuristic function is an estimate of the minimum cost from n to the goal node n_G. The particular method of calculating the cost function and the heuristic function for our A* algorithm will be further explained in section 4.2.2 and 4.2.3. If $h(n) = 0$ the A* algorithm will behave exactly like Dijkstra's algorithm. If $h(n)$ can estimate the true remaining minimum cost from n to n_G, then A* will expand only the optimal path and no other nodes[26]. In other words, the closer the heuristic estimate is to the remaining cost, the fewer the nodes the A* algorithm will expand.

In order for A* to find the optimal path, the heuristic employed by the algorithm must be admissible[26],[27],[28]. An admissible heuristic is one that never overestimates the minimum remaining cost to the goal node. If the heuristic overestimates the minimum remaining cost

to the goal node then it is said to be inadmissible. The use of an inadmissible heuristic will still yield a path; however, the path will not be optimal. It is the addition of the heuristic to A* that helps guide the A* search algorithm in finding the best path solution quickly by attempting to predict which nodes will provide a least-cost path.

Both Dijkstra's algorithm and the A* algorithm have benefits. If the heuristic used for A* is admissible, A* can expand far fewer nodes than Dijkstra's algorithm and in the worst-case, will expand no more nodes that Dijkstra's algorithm. However, A* search can only be used for point-to-point optimal path planning, while Dijkstra's algorithm can be used for optimal path planning to all nodes of the map. In addition to the algorithms discussed here, other forward searching algorithms exist such as Iterative-Deepening A* (IDA*), alpha-beta pruning, and Beam Search, among others. Many of the algorithms make improvements in computation time on the A* algorithm, particularly the IDA* algorithm[29]. In addition to forward searching algorithms, backward searching and bidirectional searching algorithms also exist. An overview of many of these algorithms is provided in [23].

Returning to the discussion of the A* algorithm, Algorithm 4.2 conveys a possible implementation of the algorithm. For the sake of demonstration and simplicity, Algorithm 4.2 shows the use of euclidean distance for both the cost $g(n)$ and the heuristic $h(n)$. The calculation of the cost and heuristic functions implemented in the final algorithm differ slightly, and are explained in section 4.2.2 and 4.2.3 respectively.

We see from Algorithm 4.2 that some book keeping must occur in order for the A* algorithm to be systematic and functional. The $OPEN$ set is the set of all alive nodes, while the $CLOSED$ set is the set of all dead nodes. The $OPEN$ set can be used to record not only the alive nodes, but the nodes' corresponding $g(n)$, $h(n)$, and $f(n)$ values along with the parent of each node. Without recording the parent of each node, the algorithm will be able to determine that an optimal path from n_I to n_G exists, but the path itself cannot be reconstructed. By recording the parent of each node, an algorithm need only start with n_G and work backward to n_I to determine the optimal path. Because Algorithm 4.2 checks to

Algorithm 4.2 A* Search

Require: n_I, n_G

```
1: CLOSED ← [ ]
2: OPEN ← n_I
3: g(n_I) ← 0
4: h(n_I) ← distance(n_I, n_G)
5: f(n_I) ← 0 + h(n_I)
6: while OPEN ≠ [ ] do
7:     n ← the node in OPEN having min(f(n))
8:     if n = n_G then
9:         Reconstruct path
10:        return SUCCESS
11:    end if
12:    remove n from OPEN
13:    add n to CLOSED
14:    for all n' do
15:        if n' ∈ CLOSED then
16:            continue
17:        else
18:            if n' ∉ OPEN then
19:                add n' to OPEN
20:                g(n') ← g(n) + distance(n, n')
21:                h(n') ← distance(n', n_G)
22:                f(n') ← g(n') + h(n')
23:                n'_parent ← n
24:            else if n' ∈ OPEN then
25:                g_tentative(n') ← g(n) + distance(n, n')
26:                if g_tentative(n') < g(n') then
27:                    g(n') ← g_tentative(n')
28:                    f(n') ← g(n') + f(n')
29:                    n'_parent ← n {record new parent of n'}
30:                else
31:                    DO nothing
32:                end if
33:            end if
34:        end if
35:    end for
36: end while
37: return FAILURE
```

see if n' already exists in the $OPEN$ set and checks to see which parent of n' provides the least cost path, a node will never be duplicated in the open set. This makes reconstructing the optimal path trivial. Algorithm 4.3 provides a sample algorithm for reconstructing the optimal path.

Algorithm 4.3 Reconstruct Path

Require: n_G

$i = 1$
$OptimalPath(i) \leftarrow n_G$
$n \leftarrow n_G$
while $n \neq n_I$ **do**
    i++
    Locate n in $OPEN$
    Get parent of n
    $OptimalPath(i) \leftarrow$ parent of n
    $n \leftarrow$ parent of n
end while
return $OptimalPath$

Although the elements of the optimal path will be located in the $CLOSED$ list instead of the $OPEN$ list, the $OPEN$ list is where it was suggested to initially store the cost and parent information for each node. For this reason Algorithm 4.3 references the $OPEN$ list to retrieve the parent information for each node. Regardless of where the parent information for each node is stored, the method of reconstructing the path remains the same.

4.2.1 Graph Construction

As A* is a graph search algorithm, a graph, or map, must be constructed that contains all of the state information required for the planning algorithm. For our algorithm, the graph is developed implicitly during algorithm run-time by defining a set of valid actions for the robot as defined in Figure 4.2, and by using the information provided by the stereovision imagery collected by the UAV. Each node is defined by both position and attitude state information, where the position is a 3×1 vector defined as $[x\ y\ z]$, and the attitude is a

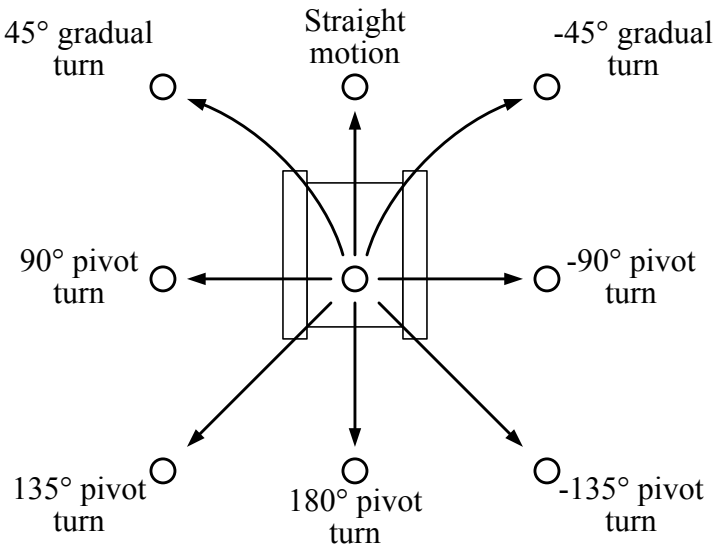


Figure 4.2: The action space of the robot

3×1 vector defined as $[\phi\ \theta\ \psi]$ representing the roll, pitch, and yaw of the robot. The action space depicted in Figure 4.2 defines the set of actions $U(n)$ that are valid at each node n. This finite set of actions define a possible change in position and yaw, where the position changes occur only in the orthogonal and $\pm 45°$ diagonal direction from the current node. Applying one of the actions through the state transition function $f(n, u)$ results in a new node in the graph, represented by n'. Each action has an associated cost as a result of the change in state, which will be explained in detail in section 4.2.2. Because the change in state information is different depending on the direction of travel between nodes, the result is a directed graph as depicted in Figure 4.3.

As mentioned above, the stereovision data aids in defining the graph by providing the position state information to the algorithm. The stereovision data is provided in the form of three $m \times n$ arrays, each of which contain the x, y, or z state information. When the algorithm initiates, the row and column index of the initial position state are determined and passed into the algorithm. As the algorithm progresses, the row and column indices of each future node are recorded and passed into three arrays in order to determine the corresponding robot position state. The robot attitude state information is calculated during algorithm execution

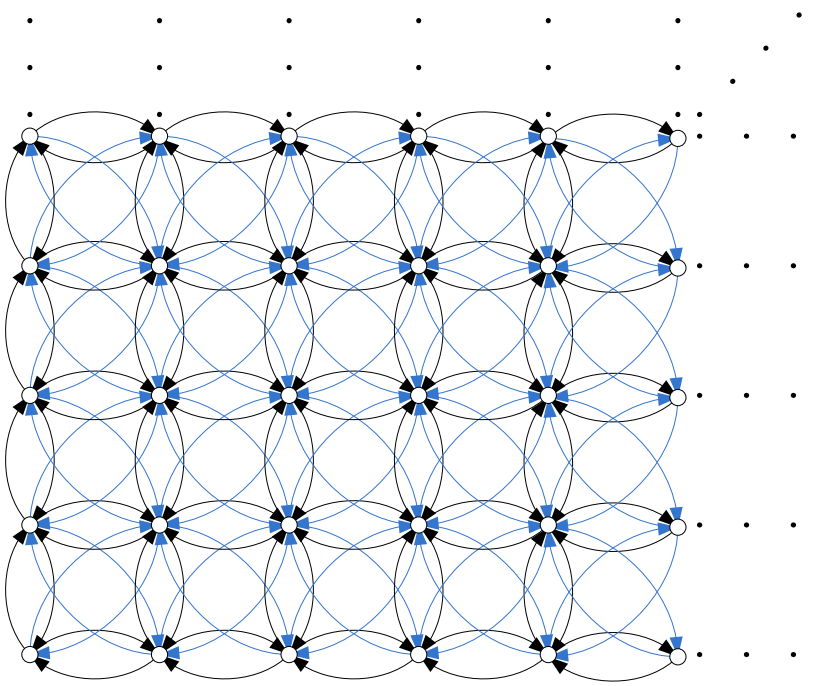


Figure 4.3: A portion of the graph used in the planning algorithm, indicating edges between nodes

by utilizing the x, y, and z terrain information provided by the stereovision data as discussed in the Robot-surface interaction model section of this chapter located on page 58.

Stereovision imagery provides a means of obtaining 3-D terrain topology information by correlating pixels between two images in order to ascertain depth information from the images. The correlation of two pixels in a stereo-pair provides a disparity. From this disparity the x-y-z coordinates of the pixels can be determined. While not all pixels in a stereo-pair will be correlated, the pixels that are correlated provide a point cloud, a set of points in 3-D space. The point cloud represents the 3-D geometry of the objects in the stereo images. Stereovision has been used to create a 3-D representation of a sample terrain field shown in Figure 4.4, which has been used for the development of the path planning algorithm. The point cloud generated by a stereoscopic system, shown in Figure 4.5, is non-uniform in the x, y, and z directions. Because of this characteristic it is difficult to represent the data in the form of a 2-D array. For this reason the 3-D stereo data has been restructured to fit a regular

Figure 4.4: Photo of terrain used for path planning simulations

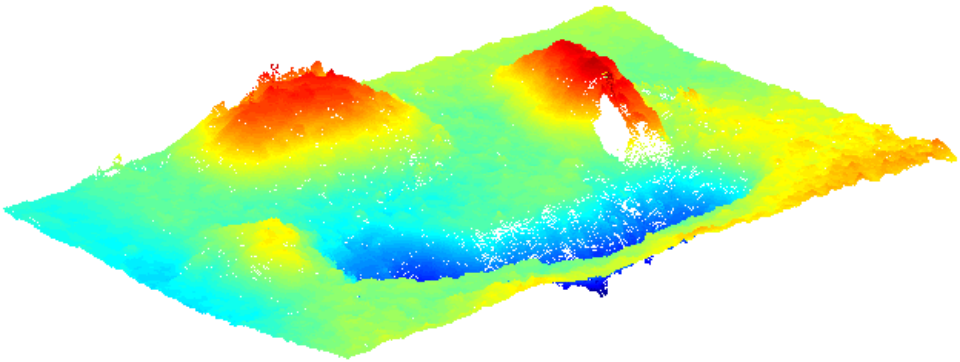


Figure 4.5: Point cloud model of terrain used for path planning simulations

grid in the x and y dimensions. Representing the 3-D data in this form is known as a digital elevation map or DEM. The procedures outlined in [30] and [31] were followed to produce the DEM of the terrain map used in the planning algorithm, which is shown by Figure 4.6. With the 3-D data represented as a DEM, the x, y, and z coordinates of each point can be stored in an array, rather than a vector. Storing the data in an array lends to a more simplified approach when implementing the graph search concepts of many discrete-domain path planning algorithms.

The cell size of the DEM directly affects the planning algorithm, as the cell size is inversely proportional to algorithm execution time. When converting from the stereo data to the DEM, a course cell size leads to less nodes in the map, which allows the algorithm to execute faster. However, the DEM cell size acts as a filter; as the cell size increases, some terrain information is lost. Alternatively, a smaller cell size allows for more of the terrain information provided by the stereo data to be transfered to the algorithm, but as the cell size decreases,

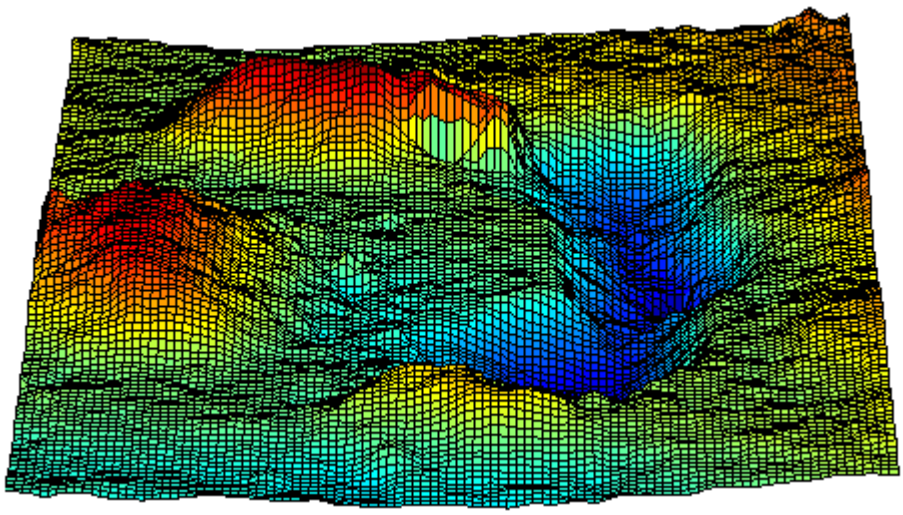


Figure 4.6: Surface model of terrain used for path planning simulations

the algorithm execution time increases. Some of the filtering effect caused by the stereo data to DEM conversion is desirable as the vision sensors used to obtain the stereo data inject some noise into the data. The DEM used for the path planning work in this thesis uses a 40 cm × 40 cm cell size, which is approximately 1/10 of the robot size. While various cell sizes were evaluated, the 40 cm × 40 cm cell size provided the best balance between algorithm efficiency and terrain information.

4.2.2 Cost Function

The cost function is a mathematical expression that is used to calculate the effort required to move along an edge from one node to its successor node(s). At the most basic level, the cost function for the A* algorithm is a measure of distance as suggested in Algorithm 4.2. As the algorithm progresses, the cost $g(n)$ to arrive at node n from the start node n_I is recorded by summing the cost of moving along each edge of the path:

$$g(n) = g(n_I, 2) + g(2, 3) + \cdots + g(n-1, n) \tag{4.2}$$

In order to determine the cost at a successor node n' we must simply calculate the cost along the edge from n to n' rather than recalculate the cost for the entire path. Therefore, the cost of arriving at the successor node is simply: $g(n') = g(n) + g(n, n')$. The method of calculating the cost is entirely dependent upon the parameters of the path that need to be minimized. With this in mind, the overall cost from one node to the next is a summation of the cost of each path parameter as seen in Equation 4.3.

$$g(n) = \alpha g_1(n) + \beta g_2(n) + \cdots + \zeta g_m(n) \tag{4.3}$$

This allows for the cost of each path parameter $g_k(n)$ to be weighted differently where α, β, and ζ are the weighting coefficients. The weights are assigned based on the importance of each path parameter to the overall cost of the path. A more heavily weighted parameter indicates higher importance and therefore more effort is made to minimize the effect of that parameter on the path.

In section 4.1 a set of requirements were outlined for the path planning process. The minimum time requirement has been incorporated into the cost function. By looking at the robot performance data from section 3.3, it can be seen that the time required to traverse a slope increases as the slope increases. It has also been found that when turning, the velocity of the robot is a function of the radius of curvature of the turn[32]. The performance data from section 3.3 also shows that the robot is incapable of traversing certain terrain. The cost function must account for all of these factors. In an effort to keep units consistent, the cost function is expressed in units of millimeter (mm) and all parameters of the cost function have been converted to units of distance. As is shown in Equation 4.3, the cost function is made up of three "distance" measurements: euclidean distance, slope equivalent distance, and turning equivalent distance.

Euclidean distance

The euclidean distance between a node and its successor node is calculated by taking the norm of the difference between the successor node (n') coordinates and the current node (n) coordinates:

$$dist = \|n' - n\| \tag{4.4}$$

Where $\|\cdot\|$ denotes the 2-norm and both n and n' are 3-element vectors of the form $[x\ y\ z]$, representing the coordinates of the node in 3-D space. The euclidean distance is the actual distance that the robot will travel between node n and n'.

Slope equivalent distance

As mentioned above, we know from the robot performance data in section 3.3 that the time required to traverse positively sloped terrain increases as the slope increases. In addition, the time required to traverse negatively sloped terrain is approximately equal to the time required to traverse flat terrain. From this information, it can be determined that the cost of traversing positively sloped terrain is greater than the cost of traversing flat or negatively sloped terrain. In order to incorporate this increased cost into the cost function, a new distance has been defined as the "slope equivalent distance". In order to calculate the slope equivalent distance, we have used the velocity data presented in section 3.3. Although the robot velocity on positively sloped terrain is a function of $\sin(\theta)$ as discussed in Appendix D, the portion of the curve over which the robot operates is fairly linear and can be closely approximated by a linear function. The velocity of the robot as a function of positively sloped terrain is defined as:

$$v_r = \begin{cases} -2.01\theta + 358.97 & \forall\ \theta > 0 \\ 358.97 & \forall\ \theta \leq 0 \end{cases} \tag{4.5}$$

$$\begin{aligned} \text{where: } v_r &= \text{velocity of the robot } (^{mm}/_{s}) \\ \theta &= \text{robot pitch angle (degrees)} \end{aligned}$$

From the velocity of the robot the time required to traverse from n to n' can be calculated:

$$t_s = \frac{dist}{v_r} \tag{4.6}$$

By using the time t_s and assuming the terrain is flat, we use the flat terrain velocity of $v_{flat} = 358.97\,^{\text{mm}}/_{\text{s}}$ from Equation 4.5 where $\theta = 0$ to calculate an equivalent distance:

$$dist_{eq} = t_s v_{flat} \tag{4.7}$$

This equivalent distance includes the cost contributed by the euclidean distance as calculated above and the cost of the slope. Because $dist$ and $dist_{eq}$ are both measures of distance, we can simply subtract $dist$ from $dist_{eq}$ to see the affect of the slope on the cost function:

$$dist_{slope} = dist_{eq} - dist \tag{4.8}$$

Turning equivalent distance

While straight-line motion for a tracked mobile robot is fairly simple, turning introduces nonholonomic constraints into the system[32]. These nonholonomic constraints prevent the robot from traveling at maximum velocity while turning. The velocity of the robot is a function of the radius of curvature of the turn. In addition to the nonholonomic dynamic constraints, because the robot is differential drive, the robot has the ability to pivot about a point known as the instantaneous center of rotation (ICR)[33],[34]. Although the ICR is not guaranteed to be coincident with the center of gravity (CG) of the vehicle, it is assumed for the path planning algorithm that a pure pivot rotation results in an ICR that is coincident with the CG of the robot. Because the robot must slow down to turn, or stop completely in the event of a pivot turn, we will take the additional time required to do so into account when

calculating the cost to move from n to n'. Because calculating the velocities and accelerations due to turning from the equations of motion for the robot would significantly complicate and reduce the efficiency of the algorithm, a simplified approach is taken for calculating turning costs. Once again, from the robot performance testing in section 3.3 the average angular velocity of the robot during a pivot turn is calculated to be $\dot{\psi} = 66.7\ deg/s$ where $\dot{\psi}$ is the angular velocity about the yaw-axis. From the angular velocity, the time required for the robot to change directions can be calculated as $t_t = (1/\dot{\psi})\Delta\psi$ where $\Delta\psi = |\psi(n') - \psi(n)|$. Using t_t, an equivalent turning distance can be calculated as:

$$dist_{turn} = t_t v_{flat} \tag{4.9}$$

where $dist_{turn}$ is the flat terrain distance that could be traveled in the amount of time required to turn. The calculation of the turning cost in the above manner makes the following assumptions:

a) all turns made by the robot during the course of the path are pivot turns
b) the robot can start and stop instantaneously (acceleration and deceleration not considered)
c) the robot is statically and dynamically stable during all turns

This method of calculating the cost of a turn appeared to be a fairly good estimate at first. However, a comparison of the paths produced by the planning algorithm using this method with actual robot performance on the debris field showed that a distinction should be made between 45° turns, where the robot could remain in motion, and turns greater than 45°, where the robot would be required to stop before turning. The distinction was made by implementing a weighting factor as shown in Equation 4.3. By applying a separate weighting factor to 45° turns from all other turns the following turning equivalent distance is obtained:

$$dist_{turn,actual} = \begin{cases} 0.01 dist_{turn} & \forall\ \Delta\psi \leq 45° \\ dist_{turn} & \forall\ \Delta\psi > 45° \end{cases} \tag{4.10}$$

Robot-Surface interaction model

In order to evaluate the cost of moving from node n to n' using the above criteria, the algorithm must gather some additional data about the robot orientation on the terrain as mentioned in section 4.2.1. By determining the orientation of the robot on the terrain we can ascertain the pitch and roll of the robot. The pitch and roll of the robot at a particular location on the terrain is a function of the x and y coordinates of the CG and the yaw angle of the robot. The orientation of the robot on the terrain is calculated by determining a statically stable orientation of the robot on the terrain. Static stability is defined as a robot orientation such that the projection of the robot center of gravity onto the x-y plane is contained within the convex hull of the projections of all the points supporting the robot onto the x-y plane. In order to determine this orientation, a subset of the terrain data is collected based on the location of the robot center of gravity on the map and the yaw angle of the robot. The subset of interest is determined by projecting the tracks of the robot onto the x-y plane of the map and collecting the points enclosed by the bounding box formed by the robot tracks as shown in Figure 4.7. The resulting subset contains 20-26 terrain data points per track depending on the robot orientation.

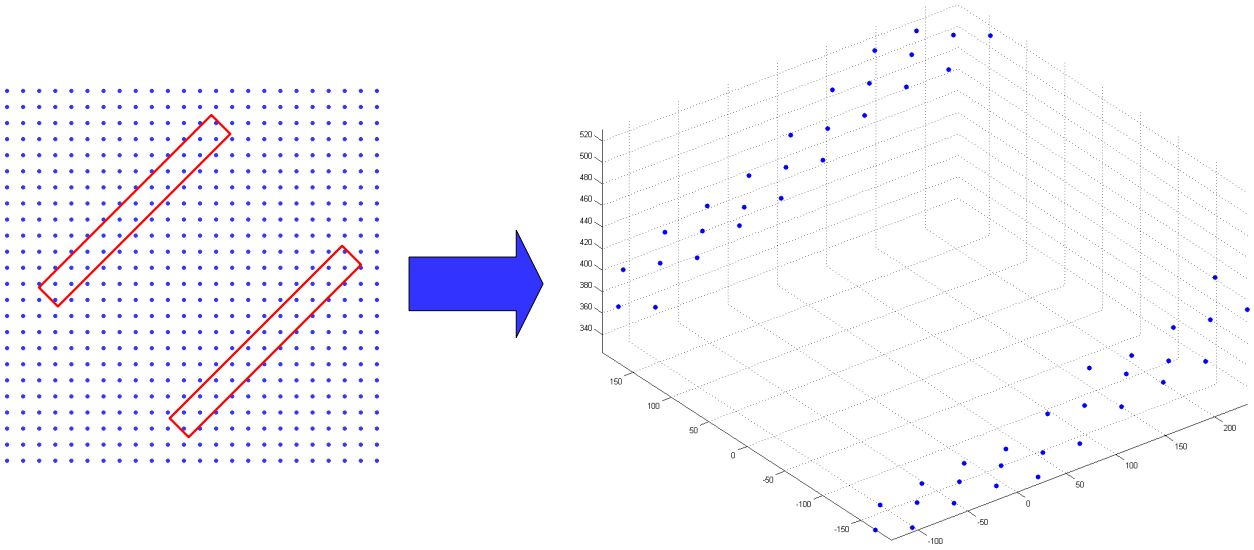


Figure 4.7: The subset of terrain data contained within the projection of the robot tracks

In order to place the robot on the terrain, we assume that the robot is a rigid body with the lower bound of the rigid body defined by a planar surface. The planar surface that best fits the subset of data enclosed by the boundary of the robot projection is determined by fitting a constrained least-squares plane to the subset of data. This is done by implementing the `fmincon` function from the Matlab® Optimization Toolbox. The cost function $f(d, a, b)$ to be optimized is:

$$f\left(d, a, b\right) = \sum_{i=1}^{n} \alpha_i \left(z_i - d + a x_i + b y_i\right)^2 \tag{4.11}$$

where the parameters a, b, and d of the cost function are the coefficients of the standard equation for a plane: $ax + by + cz + d = 0$ where it is assumed that the value $c = 1$. The cost function represented in Equation 4.11 provides us with a weighted least-squares plane that best fits the data. The α_i term allows for data points to be weighted differently depending on how well the plane should fit those data points. Because we want the plane to be fit to points such that the convex hull of the projection of those points contains the projection of the CG of the robot, we weight the points closer to the CG of the robot more heavily than the points further away from the CG by using the relationship shown in Equation 4.12.

$$\alpha_i = 1 - \left| \frac{x_i - X_{CG}}{L_{robot}} \right| \tag{4.12}$$

In addition to the above weighting condition we desire a plane that "rests" on top of the data. This is done by imposing a set of inequality constraints on the cost function such that the z value at a particular point on the plane must be greater-than or equal to the z value of its corresponding data point. This forces the optimization to find a plane that fits tightly to the top of the data, without any terrain data points being located above the plane. Figure 4.8 demonstrates this concept for a small dataset. The large green points are the three points used to define the plane (also contained in the subset of interest, denoted Z_{data}). The points denoted Z_{plane} are the projections of the Z_{data} points along the z-axis onto the plane. From the graph we see that the distance from Z_{data} to Z_{plane} is always greater than or equal to zero, or in other terms $Z_{plane} - Z_{data} \geq 0$. This inequality constraint, in

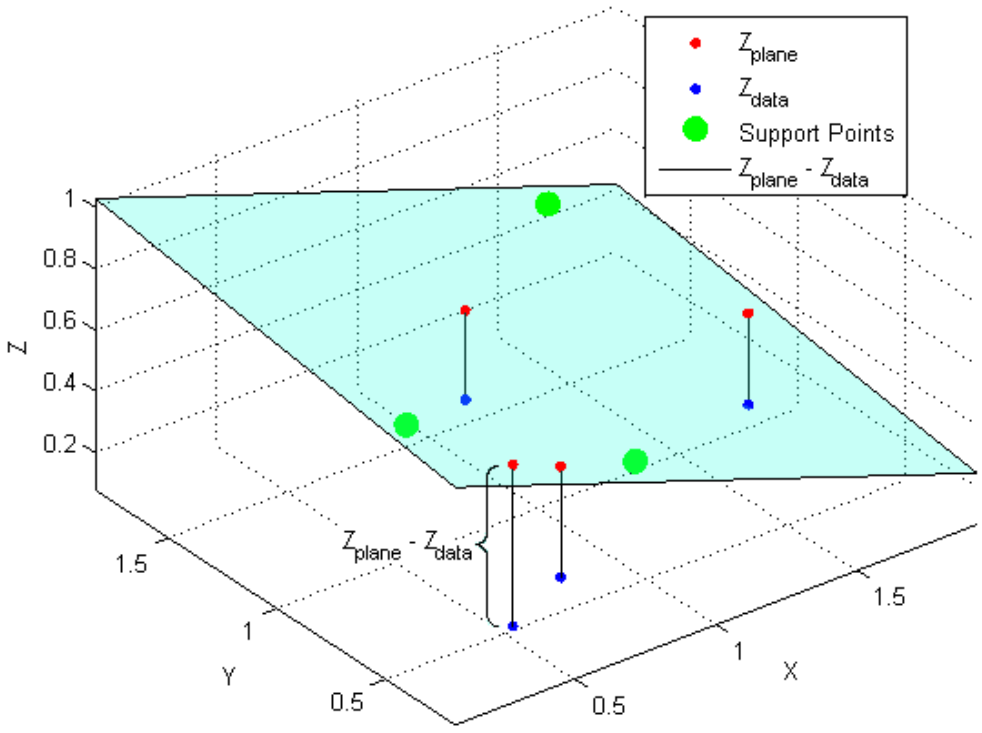


Figure 4.8: Best-fit plane example

more explicit terms is imposed on the cost function in Equation 4.11 in order to ensure the behavior seen in Figure 4.8:

$$d - ax_i - by_i \geq z_i \quad \forall\ i = 1, \cdots, n \tag{4.13}$$

Represented in matrix form, we see that n constraint equations are required to properly define the plane:

$$\begin{bmatrix} d - ax_1 - by_1 \\ d - ax_2 - by_2 \\ \vdots \\ d - ax_n - by_n \end{bmatrix} \geq \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \tag{4.14}$$

From the plane defined by a, b, and d three unit vectors can be defined that coincide with

the body-fixed reference frame of the robot ($\mathcal{F}_{Robot}$). Using the unit vectors from $\mathcal{F}_{Robot}$, the rotation matrix can be calculated that transforms $\mathcal{F}_{Robot}$ into the global frame ($\mathcal{F}_{global}$). In actuality, this rotation matrix is defined to be $[i_{Robot}\ j_{Robot}\ k_{Robot}]$ where i_{Robot}, j_{Robot}, and k_{Robot} are the unit vectors coinciding with $\mathcal{F}_{Robot}$. The pitch angle (θ) of the robot is then defined to be the angle between i_{Robot} and the x-y plane in the global frame, and the roll angle (ϕ) of the robot is defined to be the angle between j_{Robot} and the x-y plane in the global frame. With a method of calculating the pitch and roll angles of the robot along with the turn angle of the robot, the cost of moving from n to n' can be calculated.

Cost function algorithm

As mentioned above, the cost function is made up of three components. We also know that some terrain is not traversable by the robot and as such must be taken into account by the algorithm. The algorithm also accounts for the possibility of different slopes being encountered at node n and n' by calculating the slope at each node and using the greater of the two slopes at n and n' to calculate the cost. The implementation of the cost calculation is presented in Algorithm 4.4.

4.2.3 Heuristic Function

As discussed in the explanation of the A* algorithm on page 46, the heuristic used in the A* algorithm is an estimate of the minimum remaining cost from n to n_G. Also discussed on page 46, the A* algorithm performs better as the heuristic $h(n)$ approaches the true remaining cost to the goal node[26]. As such, we want to match the heuristic as close to Equation 4.3 as possible, such that:

$$h(n) = \alpha h_1(n) + \beta h_2(n) + \cdots + \zeta h_m(n) \tag{4.15}$$

Algorithm 4.4 Cost Function

Require: n, n'

$pitch \leftarrow max(\theta(n), \theta(n'))$
$roll \leftarrow max(\phi(n), \phi(n'))$
if $roll >$ roll threshold **then**
    $g(n') \leftarrow \infty$ {Robot not stable. Terrain not traversable}
    return $g(n')$
else if $pitch >$ pitch threshold **then**
    $g(n') \leftarrow \infty$ {Robot not stable. Terrain not traversable}
    return $g(n')$
end if
$dist \leftarrow \|n' - n\|$ {see eqn. 4.4}
if $pitch > 0$ **then**
    $v_r \leftarrow f(pitch)$ {see eqn. 4.5}
else if $pitch \leq 0$ **then**
    $v_r \leftarrow v_{flat}$
end if
$t_s = dist/v_r$
$dist_{eq} = t_s * v_{flat}$ {see eqn. 4.7}
$dist_{slope} \leftarrow dist_{eq} - dist$ {see eqn. 4.8}
if $\Delta\psi \leq 45°$ **then**
    $dist_{turn} \leftarrow 0.01\frac{1}{\dot{\psi}} |\psi(n') - \psi(n)| v_{flat}$ {see eqn. 4.10}
else if $\Delta\psi > 45°$ **then**
    $dist_{turn} \leftarrow \frac{1}{\dot{\psi}} |\psi(n') - \psi(n)| v_{flat}$ {see eqn. 4.10}
end if
$g(n, n') \leftarrow dist + dist_{slope} + dist_{turn}$
$g(n') \leftarrow g(n) + g(n, n')$
return $g(n')$

While matching the heuristic calculation to the cost calculation as closely as possible is ideal, the inclusion of a heuristic in the A* algorithm is intended to speed up the algorithm by expanding fewer nodes. Care must be taken to ensure that the heuristic is efficient to calculate, otherwise the benefits of including the heuristic in the algorithm are lost.

While the cost $g(n)$ is dependent upon the succession of nodes used to arrive at n, the heuristic $h(n)$ is path independent. This is beneficial as the heuristic for a node already contained in the $OPEN$ list need not be recalculated if a better path to that node is found.

An attempt to match the heuristic function to the cost function as closely as possible is made by calculating an element of the heuristic function for each of the same three path parameters as calculated by the cost function: euclidean distance, slope equivalent distance, and turning equivalent distance.

Euclidean distance

The euclidean distance term used for the heuristic is not a true euclidean distance from n to n_G, but a modified version of this. By reexamining the graph from section 4.2.1 we see that the robot can only move in the orthogonal and 45° diagonal directions. Because of this, the robot will not always be able to travel from n to n_G in a straight line. In order to calculate the minimum distance that can be traveled to reach the goal, we must look at the number of elements between n and n_G in both of the orthogonal directions. The remaining distance between n and n_G in the orthogonal directions will be used to define a maximum number of diagonal moves that can be traveled to arrive at the goal node. By revisiting the concept of organizing the nodes as a matrix array, we measure the number of rows from n to n_G as y and the number of columns from n to n_G as x. By doing so, the maximum number of diagonal steps the robot will be able to make will be:

$$steps_{diag} = min(x, y) \tag{4.16}$$

The remaining steps to the goal node will be movements along an orthogonal direction and will be:

$$steps_{orth} = max(x, y) - steps_{diag} \tag{4.17}$$

Figure 4.9 depicts a minimum-distance path from n to n_G, however, it does not depict the only minimum distance path because the order of the diagonal and orthogonal steps is commutative. However, a large number of changes in direction by the robot will be reflected in the calculated cost of traversing the path. To calculate the minimum distance from n to

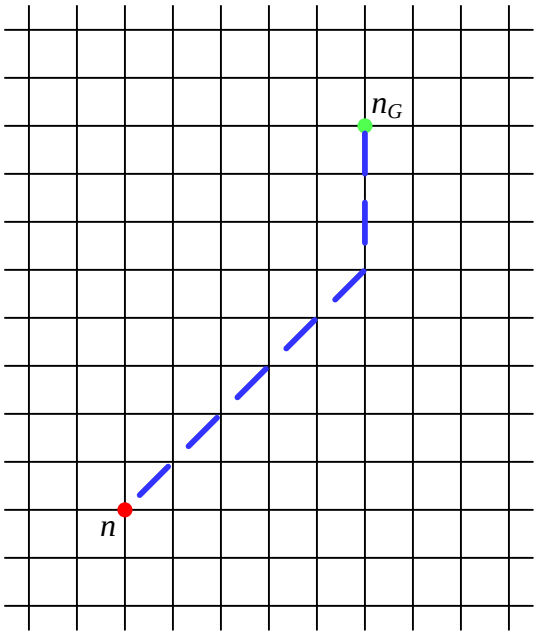


Figure 4.9: Minimum distance from n to n_G

n_G we take the coordinates of n to be $[n_x\ n_y\ n_z]$ and the coordinates of n_G to be $[n_{G,x}\ n_{G,y}\ n_{G,z}]$.

$$\begin{aligned} dist_{flat} &= \text{distance along}(steps_{diag}) + \text{distance along}(steps_{orth}) \\ &= \|[n_x\ \ n_y] + [sgn(x)min(|x|,|y|)\ \ sgn(y)min(|x|,|y|)]\| \\ &\quad + |max(|x|,|y|) - min(|x|,|y|)| \end{aligned} \tag{4.18}$$

Equation 4.18 provides the distance along the x-y plane. The heuristic should also account for the change in z distance as well. This is done in Equation 4.19, where it is assumed that the remaining terrain from n to n_G has a constant slope.

$$dist_h = \sqrt{dist_{flat}^2 + (n_{G,z} - n_z)^2} \tag{4.19}$$

Slope equivalent distance

The changes in slope along the path from n to n_G are unknown, however, the trending slope from n to n_G can be taken into account. This is done by calculating an "average" slope between n and n_G. The slope used for the heuristic calculation is:

$$\theta_h = tan^{-1}\left(\frac{n_{G,z} - n_z}{dist_{flat}}\right) \tag{4.20}$$

The heuristic slope is then substituted into Equation 4.5 to determine the velocity of the robot needed for the heuristic calculation. The calculated velocity is then substituted into Equation 4.6, and equations 4.7 and 4.8 are used to calculate the slope equivalent distance for the heuristic function.

Turning equivalent distance

The turning equivalent distance portion of the heuristic function checks to see if the straight line distance between n and n_G is the same as $dist_{flat}$.

$$dist_{turn,h} = \begin{cases} 0 & \text{if } dist_{flat} = \|n_G - n\| \\ 0.0674 v_{flat} & \text{if } dist_{flat} > \|n_G - n\| \end{cases} \tag{4.21}$$

The second condition of Equation 4.21 multiplies the flat ground velocity by 0.0675 because this is the time required for the robot to turn 45°. Only a single 45° turn is accounted for by the turning equivalent distance portion of the heuristic function when the second condition of Equation 4.21 because this is the minimum number of turns required to arrive at the goal node.

The heuristic value is calculated by summing the distances calculated for each of the above three parameters:

$$h(n) = dist_h + dist_{slope,h} + dist_{turn,h} \tag{4.22}$$

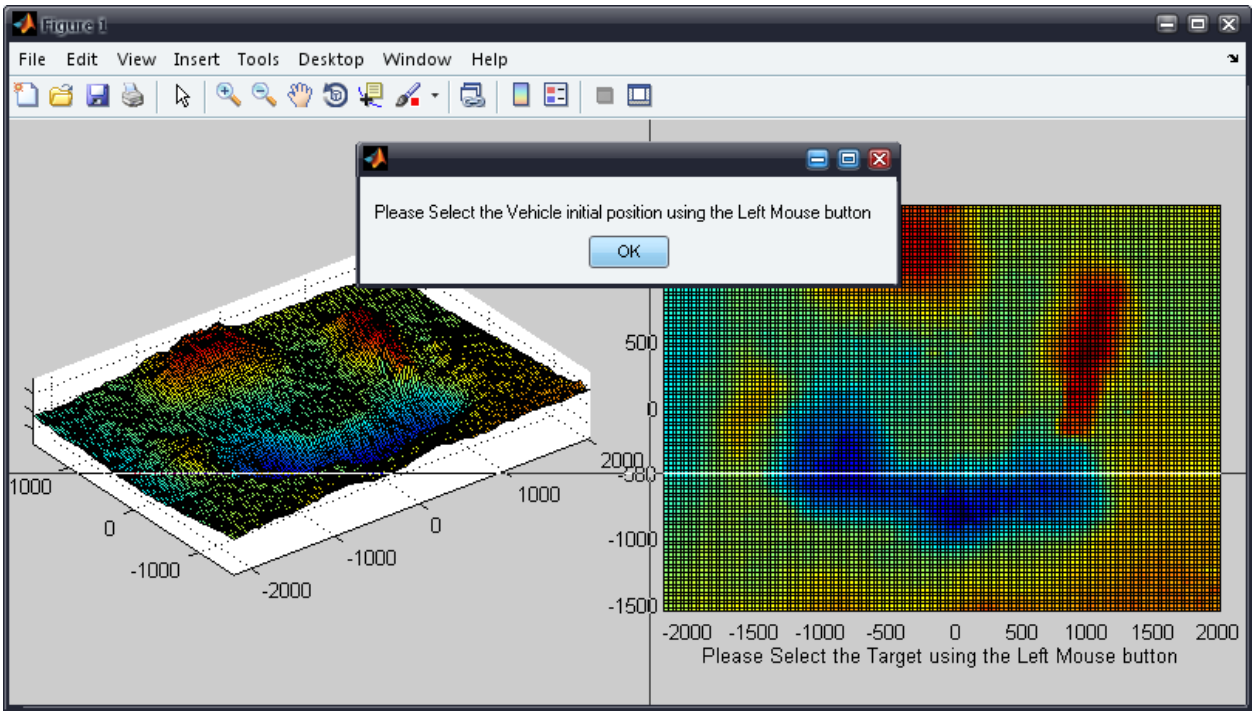


Figure 4.10: Path planning user interface

4.3 Path Planning Simulations

The path planning algorithm was developed and written in Matlab. The algorithm has been broken up into various functions in order to facilitate development and make the algorithm more modular. A list of all of the functions called by the A* algorithm and their descriptions is available in Appendix E.

The algorithm and path planning model make the following assumptions:

- No slip between track and terrain
- Terrain model is completely rigid (terrain will not move as robot drives over it)
- Terrain model is an exact replica of actual terrain (no imaging sensor uncertainties)
- Robot tracks are rigid bodies
- Dynamics of robot are neglected (only static stability is considered)

When the path planning algorithm is run, the user is presented with a view of the terrain

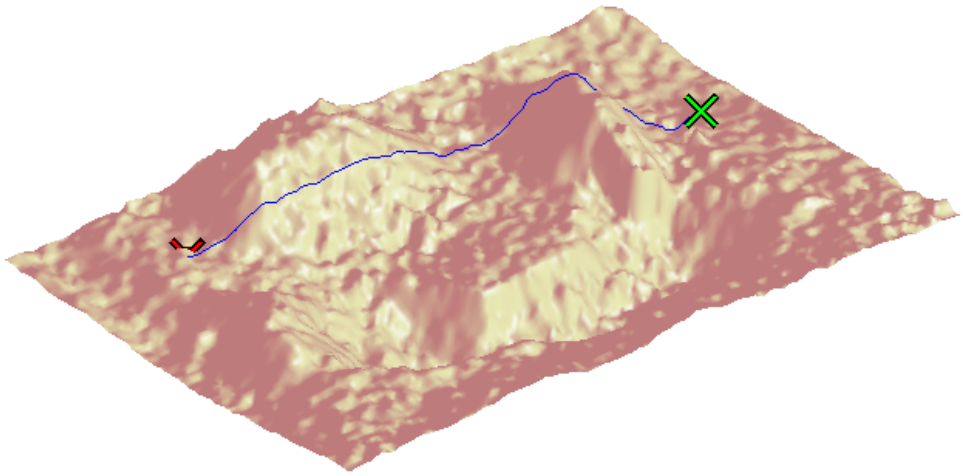


Figure 4.11: Path planned by algorithm shown on terrain model

field and is prompted to select the goal node and the initial node (Figure 4.10). The user does so by clicking on the terrain model as directed to select each point. The algorithm then runs in an attempt to plan a path from n_I to n_G. If a path exists, the path will be displayed to the user, overlaid on the terrain model (Figure 4.11) and an animation of the robot traversing the terrain model is displayed back to the user. Some frames of the animation that the user sees are depicted in Figure 4.12. In addition to the animated path, a sequence depicting the robot at various points along the path at the same time is shown to the user. A sample of this sequence is shown in Figure 4.13. If no path exists, the user is notified that the algorithm has failed.

The planned path can be used by the operator as a navigation aid. Using the robot location tracking software discussed in [1] in combination with other sensors on-board the robot, the updated robot position could be shown on the 3-D terrain map. The operator could use this virtual terrain model along with visual cues from the vision systems on board the helicopter and ground robot for successful navigation and teleoperation of the ground robot. The benefits of the 3-D data provided by stereovision for robot teleoperation are discussed further in chapter 5.

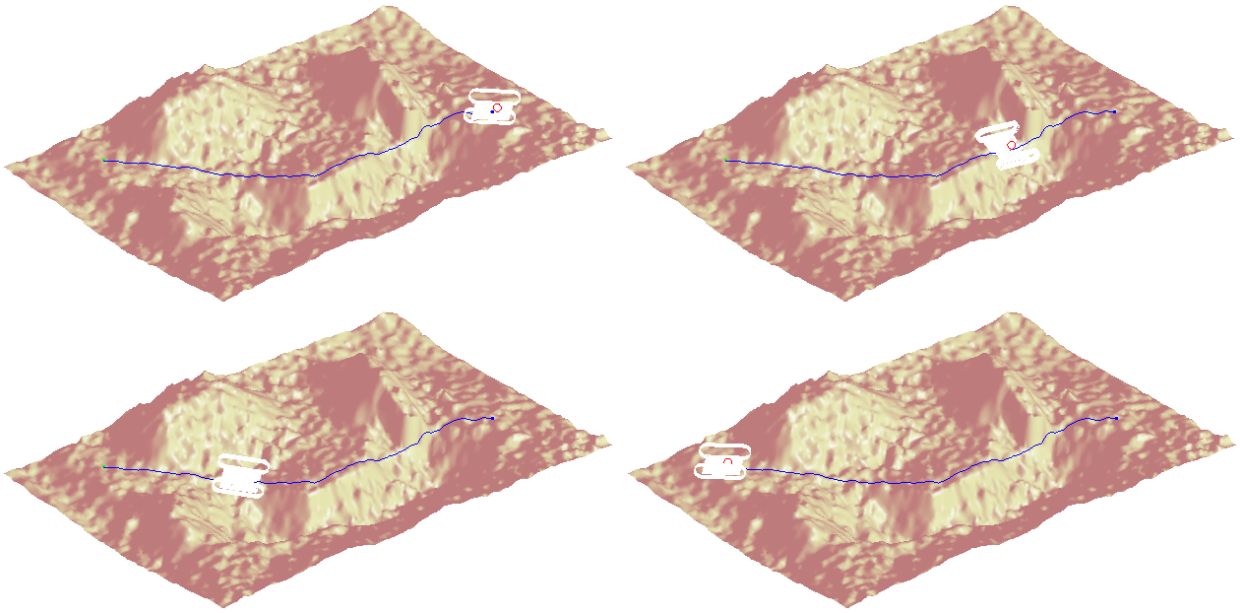


Figure 4.12: Robot at various steps along a planned path

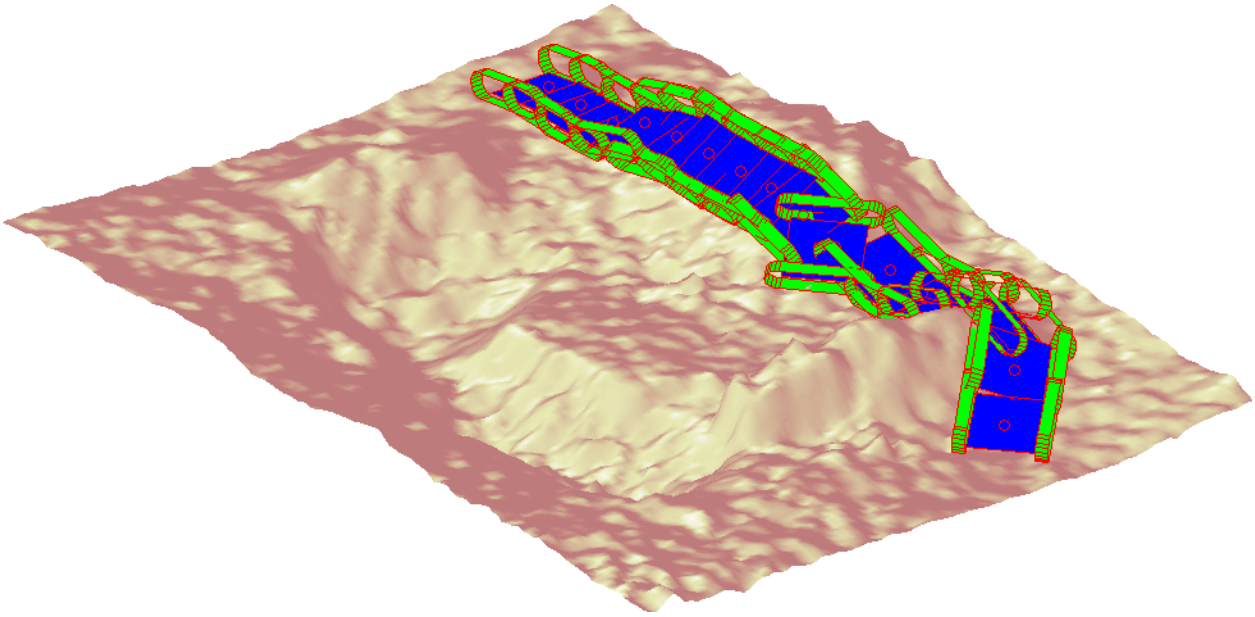


Figure 4.13: A sequence showing the robot's progression along the path

4.4 Simulation Analysis

In addition to the assumptions listed in section 4.3, the following items should be considered as future improvements to the path planning algorithm, specifically the first item in the list:

- Collision model to detect terrain collisions with undercarriage
- Track slippage model (dependent upon terrain properties)
- Incorporate terrain classes that aid in modeling track/terrain interactions (slip, friction, etc) based on the type of terrain being traversed
- Vehicle dynamics model
- Account for terrain model uncertainties

In addition, some improvements can be made to the heuristic value in order to more closely model the remaining cost. Figure 4.14 displays all nodes added to the $OPEN$ list during a path planning run overlaid on the terrain model. As can be seen, the algorithm expands a large number of nodes near the start node and tapers off as the goal node is approached. Upon examination of Figure 4.14, it appears the algorithm attempts to find a path around the red region near the start node where steeper slopes exist. The algorithm eventually makes the decision to traverse the large elevated region instead of going around it. This is likely due to the cost imposed due to turning. The heuristic does not account for multiple turns from n to n_G and therefore perceives the relatively flat terrain as a good alternative to the highly sloped terrain. Only after the algorithm has expanded a significant number of nodes does the algorithm "realize" the increasing number of turns is more costly than the steep slope. Although the algorithm functions exactly as it was programmed to function, it does not take into account the possibility of track slippage while traversing slopes and varied terrain. In attempt to validate the path displayed in Figure 4.14, the robot was taken to the debris field and an attempt was made to navigate the path. The robot was successful in navigating the path, but not without difficulty due to track slippage. A path around the red region proved to be a more reliable alternative. Again, accounting for terrain properties

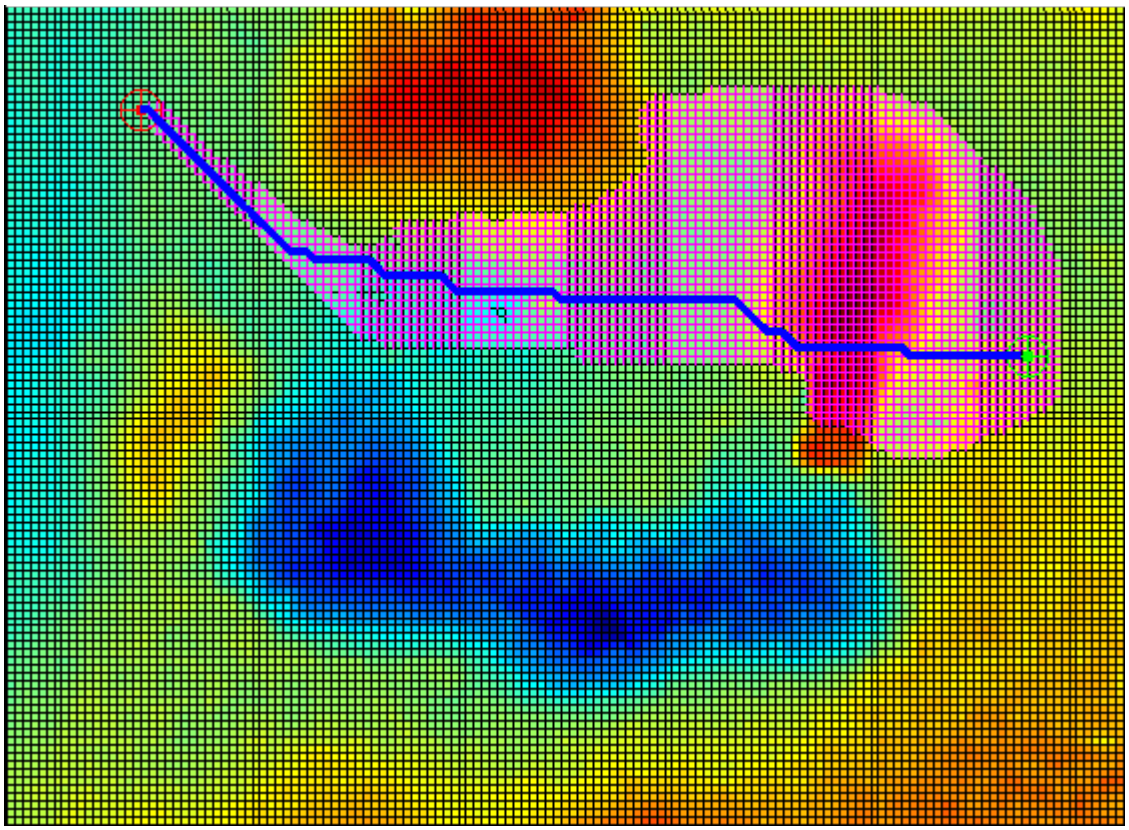


Figure 4.14: Sample node expansion during planning algorithm

and the possibility of track slippage would have improved the algorithm's ability to make this decision.

Chapter 5

Ground Robot Control through a Remote User Interface

Teleoperated robots are frequently provided operational commands based on video feedback from the robot, giving the operator updated positional information that will be used in the next set of mission instructions. Unlike fully autonomous systems, teleoperated robots require the operator to assess the robot's status before commanding the next move; these move commands may or may not be aided by machine intelligence. As mentioned in section 3.2.3, other sensory data may be provided to the operator to aid in defining the next move, such as environmental data or robot status. When a wireless link is used to communicate to the robot, significant delays can occur in the video signal and image resolution can be compromised. These factors combine to give the operator a relatively poor "feel" of the status of the vehicle, resulting in small, or even erroneous move commands, which can lead to a slow mission.

A study of operator performance using different visual feedback modes was conducted by McLean[35]. In these studies, users were asked to manipulate a robot arm to place a peg inside of a tube that was rotating on a turntable at a constant RPM. Results of this study showed that time to completion of the task were 48% greater for a 2-D monovision feedback

system vs. a 3-D vision system. The failure rate when using a 2-D view was almost twice the failure rate of the 3-D case. These experiments validated the importance of having additional spatial information available when performing a remote robotic operation. Kofman[36] has developed a non-contacting interface capable of measuring an operator's haptic motion which is translated into robot motion. The operator relies on multiple camera views of the robot to determine spatial position. Results of testing showed that processing delays affected the operator's ability to manipulate the arm, and that a 3-D presentation of the robot arm would aid the operator in performing complex tasks.

Culhane[37] has demonstrated an operator interface for ground robot navigation using LIDAR and/or a monovision camera. The results showed that operators are more comfortable navigating with raw LIDAR data than a monovision image. The raw LIDAR data provides depth information but no color information, which aids an operator in identifying objects. Alternatively, monovision visual data contains no depth information but provides the operator with a 2-D image of the actual surroundings. One significant disadvantage of LIDAR when compared to visual data is the expense to acquire the data in both computational time and cost if the desire is to geo-rectify the data. Stereopsis uses visual images to create three-dimensional maps, acquiring all spatial data simultaneously and thus allowing an easy geometrical correlation that a scanning LIDAR system does not afford.

The main challenge in designing a teleoperated robot system is to provide adequate telepresence to the operator so that the mission can be completed successfully. Some of the central issues are latency due to computational overhead, and the presentation of the robot's environment and surroundings (situational awareness) for effective decision making. This chapter examines various visual systems and displays that can be used to control the ground robot, and addresses the practical aspects of incorporating these systems onto the robot. A monovision system located on the robot such as the Axis camera, a nadir monovision camera above the robot on board the helicopter, and a stereo view of the area also acquired from the helicopter are evaluated using time and success/failure as metrics for performance.

5.1 Vision systems for robot situational awareness

The pros and cons of stereovision vs. monovision were examined via a user study conducted by Klomparens[38] to answer questions about human perception of terrain features. Figures 5.1(a)-(f) show images of a pallet oriented at several different angles. Subjects were shown the images and asked to estimate the pallet orientation angle. Figure 5.2 shows the results of the test, indicating that with monovision alone it is very difficult to resolve depth information. While this may be an intuitive result, the magnitude of the user errors was dramatic and reinforces the concept that monovision images are weak, at best, as a navigational tool.

(a) Wood pallet tilted at 00.00 degrees. Denoted as image B throughout the experiment.

(b) Wood pallet tilted at 04.10 degrees. Denoted as image E throughout the experiment.

(c) Wood pallet tilted at 13.07 degrees. Denoted as image F throughout the experiment.

(d) Wood pallet tilted at 15.89 degrees. Denoted as image A throughout the experiment.

(e) Wood pallet tilted at 21.29 degrees. Denoted as image D throughout the experiment.

(f) Wood pallet tilted at 26.52 degrees. Denoted as image C throughout the experiment.

Figure 5.1: A set of images used to measure an individual's perception of depth from 2-D data[38]

Culhane's[37] study of users navigating an obstacle course based on monovision images and

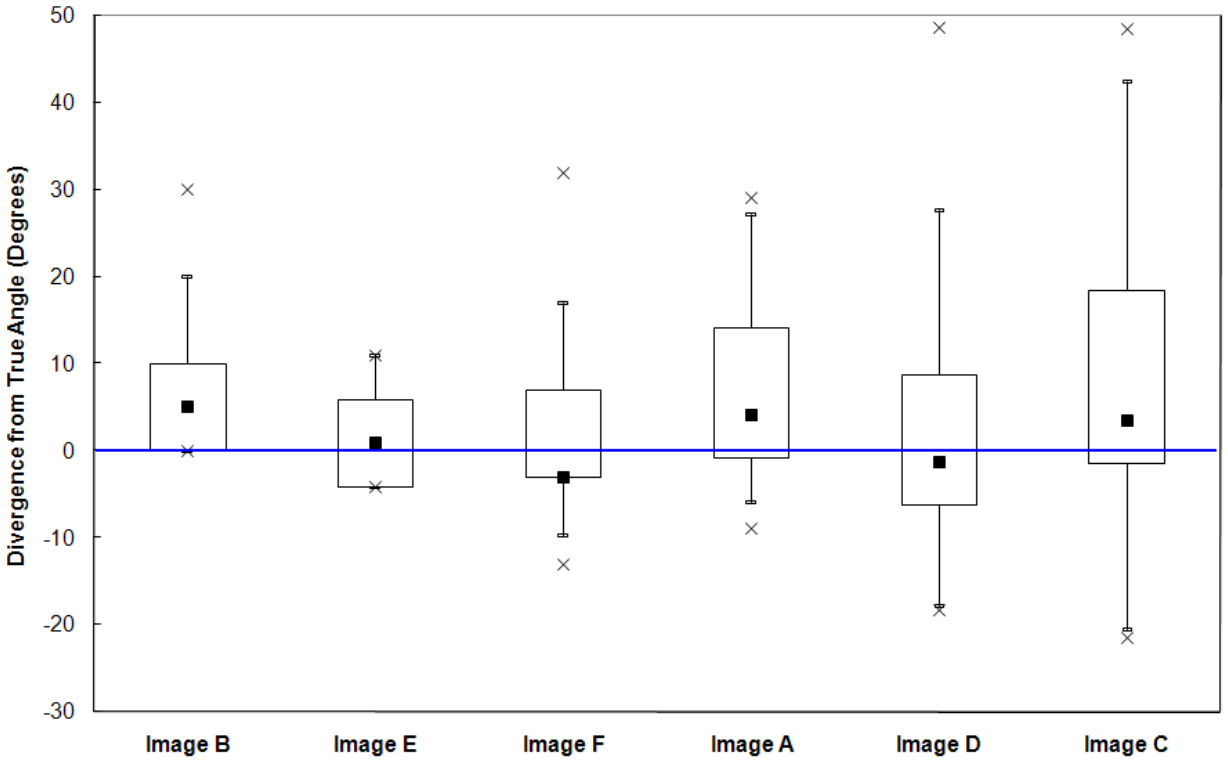


Figure 5.2: A boxplot which shows the divergence of angle estimations from the true angle. The box encloses the 25th to 75th percentile and is known as the inner quartile range. The whiskers enclose the 5th to 95th percentile. The "x" marks represent the minimum and maximum deviations, and the dot inside the inner quartile range represents the median.[38]

Figure 5.3: Laboratory obstacle course

LIDAR scans also showed these deficiencies. In this user study, a course was constructed, shown in Figure 5.3, to simulate the navigation of a vehicle through an obstacle rich environment. A cart containing the camera and a scanning LIDAR system were used for operator feedback in defining a successful path from beginning to end. This study measured total time, time per movement and the number of commanded moves to navigate the course. The results of this test showed that only a 20% success rate was obtained when monovision camera images were used as the sole source of operator feedback. Although the results show far more success in navigating the course when using the depth perception that LIDAR provides, the method of presenting the LIDAR data was inhibitive. Although LIDAR can accurately present distance data, the color information that is relied on by humans for object recognition was lost.

5.2 Evaluation of vision systems

A user study was designed to assess the capabilities of various vision systems in providing an operator with sufficient situational awareness to operate a vehicle beyond line-of-sight. The execution and results of the user study are presented below.

5.2.1 Concept of operations for the Ground Robot

Similar to the proposed mission scenario presented in section 1.3, in the proposed mission scenario for the user study, the ground robot operator is tasked with the objective of locating and identifying a particular object with the intent of relaying some information about the object back to the ground control station. The ability of the robot operator to locate and identify the particular object efficiently is largely dependent on the amount of environmental data received and properly interpreted by the operator. The operator must navigate the robot to the target site in the shortest amount of time without damage to the vehicle. The operator must make all command decisions based on the knowledge of operation of the robot and the information presented to the operator via the operator display interface.

5.2.2 Theory of stereopsis and rectification for display

Because stereovision is one of the three vision systems used for the user study, a brief discussion of stereopsis will be presented here. Stereopsis in humans is the visual process leading to perception of depth. The sense of depth emerges from the fusion of two slightly different projections onto each retina. In a similar configuration to human eyes, computer stereovision is achieved by using two cameras, separated by a distance called the baseline, to take two slightly different images of a single scene. A computer then compares the two images by making relative shifts, effectively placing one image on top of the other by translating them to find the parts that match. The shifted amounts are called the disparity values.

Using the camera placement geometry and the disparity values a depth map of the scene can be calculated.

As mentioned above, stereovision systems are usually made up of two cameras placed side by side, capturing images with slightly different perspectives of the same scene due to the distance between them. Stereo imaging has been widely used and proven effective in distance estimation through extensive research in the field of computer vision[39],[40]. Such systems apply stereovision to acquire a non-uniform grid (or cloud) of depth measurements for a pre-defined area. Figure 5.4 illustrates the methodology used to determine the distance (z) of a specific point in the image from the camera using triangulation. The distance between

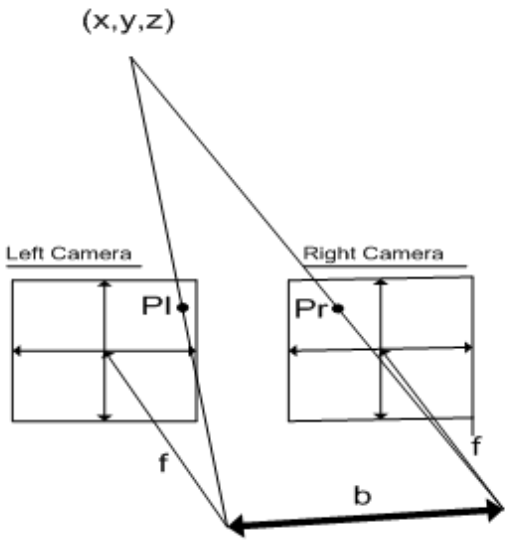


Figure 5.4: Principle of stereo vision

the centers of the two image planes is the baseline (b). Because the baseline is a known distance, the baseline can be used along with geometric relationships and triangulation to solve for z using Equation 5.1.

$$z = \frac{b \cdot f}{P_l - P_r} \tag{5.1}$$

Referring to Equation 5.1, f is the focal length and P_l and P_r are the location of the point in question from the 3-D scene in the left and right images respectively. The distance between

corresponding left and right pixels is called the disparity value, specifically for Figure 5.4, the horizontal disparity. The results of the calculation to obtain the z coordinate enable one to determine the x, y, and z coordinates and use the known xyz coordinate values to form a 3-D representation of the scene. Finding the corresponding pixel matches from the left and right images is the main challenge and the accuracy of the match will affect the result obtained. This is called the correspondence problem. The matching can be approached using edge-based or area-based algorithms[41].

The process begins by taking a pair of stereo images of the scene. The captured images are then processed to remove lens distortions and rectified to align the epipolar lines. Then edge detection is performed on the images, using pre-selected filter masks and coefficients. The images are then ready for pixel matching, in which candidate correspondences between the two images are identified. Additional constraints for texture validation, surface size validation, and disparity range reduce false matches and improve the accuracy of the resulting disparity image. The disparity values can be converted to x, y, and z coordinates, which represent a three-dimensional map of the scene in question.

5.2.3 Design of system and laboratory test configuration

It is hypothesized that the 3-D stereovision data acquired by the helicopter during the 3-D terrain mapping portion of the mission will be the most advantageous vision system for teleoperation of the ground robot. Because the robot is size, weight and bandwidth limited, it will not be able to carry a complex vision system that requires a large amount of vision processing. A study has been designed that will investigate the use of each of the vision system configurations and combinations mentioned previously, in an attempt to validate the hypothesis. Volunteers will be asked to navigate the debris field via simulated teleoperation of a ground vehicle in search of an object. The same debris field as used for path planning was used to simulate teleoperation in a real-world environment (Figure 5.5). The debris field consisted of several piles of earth, ranging in height from 8”-16” and with slopes of 20-50°,

Figure 5.5: Simulated debris field used for assessment of vision systems

and contained half of a metal barrel and a cinder block. The debris field was designed for simulated navigation of a small ground robot measuring 15" in length, 12" in width and 6" in height. The debris field was marked off with a 12x12 grid. In order to eliminate latency issues, all imagery required for the simulation was captured prior to the user study. The images for the in-situ monovision system were captured at each grid location at a height of 6 inches above ground level. The in-situ camera height is limited by the height of the robot. The aerial monovision image and the stereovision image were both captured at an approximate height of 15 feet, with a nadir view, simulating a vantage point from the helicopter. The grid pattern was used as a reference for each move location and allows for a discrete measure of move size and number of moves for all vision systems studied. The captured images were compiled into an image database for display to the user.

A set of simple user interfaces (Figures 5.6-5.8) was designed using National Instruments' LabVIEW. These interfaces give the user instructions about the mission objective, explain the use of the display interface, and display the vision data to the operator. The controls, which consist of the option to move forward, turn right, or turn left and the option to abort the mission at any time, are explained to the user. Additionally, for the stereo vision display, the user is given the option to rotate the 3-D geometry about the vertical axis. The user is instructed that the objective of the mission is to locate and navigate to a particular object in the debris field. For simulation purposes, the following instructions are given regarding the operating limitations of the ground robot:

Figure 5.6: Operator Display Interface presented to user for assessment of the in-situ monovision and the aerial monovision (display similar for both systems)

> The robot is only capable of navigating relatively flat ground. All other parts of the terrain are impassable by the robot. You will be notified of these areas if you attempt to navigate them. Additionally, the robot is not waterproof. Attempting to navigate through water will cause the robot to be destroyed and result in a failed attempt!

The user is also given instructions on how to start the simulation. Once the user begins the simulated teleoperation exercise the user interface displays the appropriate image(s) to the user. For the simulations involving the aerial monovision, an indicator is displayed on the image giving the user ground robot position and orientation information. For the stereovision simulation, a 3-D model of the robot is displayed on the stereo representation of the terrain to provide the user with ground robot position and orientation information. During the course of navigating the debris field, the user may encounter several obstacles that cannot be navigated by the ground robot due to the limitations listed above. If the user attempts to navigate into an obstacle a message appears indicating that the obstacle is impassable. Also, as indicated above, the debris field contains a water hazard that will cause a failure in the mission if an attempt to navigate through the water is made. The simulation ends when

one of the following three conditions are met:

a) the user successfully navigates to the given objective and completes the mission

b) the user navigates into the water hazard and fails the mission

c) the user makes the decision to abort the mission.

The following metrics are used to gauge the performance of each simulated navigation attempt: total simulation time, time per movement, number of invalid moves, and total number of commanded moves during course navigation, along with the outcome of the trial (pass or fail). An invalid move is counted for every attempt to navigate into an obstacle. A commanded move is counted for every movement (forward, left turn, or right turn). During the simulation the user interface automatically keeps track of the metrics. Once the simulation ends this data is written to a log file and automatically saved. In addition to the quantitative metrics some qualitative observations were made and recorded as well.

Figure 5.7: Operator Display Interface presented to user for assessment of the combined in-situ and aerial monovision systems

Each of the three vision systems were tested individually using the operator display interfaces shown in Figure 5.6 and Figure 5.8 and a combination of in-situ and aerial monovision was

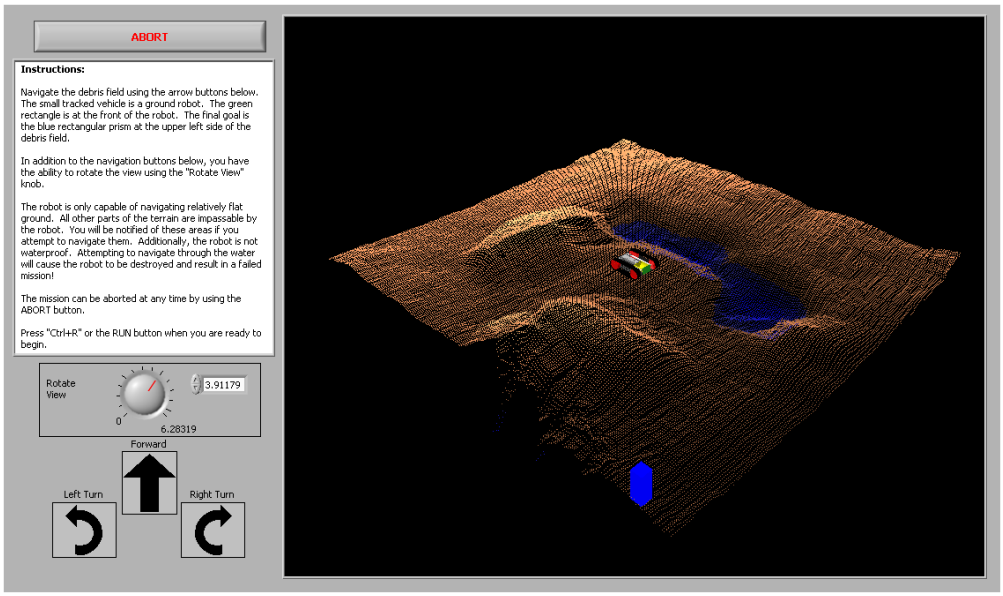


Figure 5.8: Operator Display Interface presented to user for assessment of the stereovision systems

tested using the operator display interface shown in Figure 5.7. Figures 5.9a-5.9c are sample images of the course from the three vision system configurations. The images shown and other similar images for the in-situ monovision system were used to give the operator feedback and awareness of their surroundings while operating the ground robot.

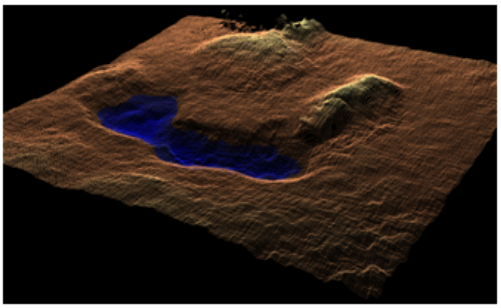
Each image type has strengths and weaknesses. While the in-situ monovision images provide terrain and obstacle height information to the user, this type of image does not give the user an overall awareness of their surroundings. The field of view is limited, and for this particular application the camera cannot be articulated. Teleoperators often refer to this as the soda straw, or keyhole affect. The operator must turn the robot left or right to gain a different perspective of the surroundings.

The external monovision image provides the user with adequate awareness of the target area, but lacks detail of the surroundings and provides no elevation data. While some elevation data can be obtained from shadows, the provided information is inadequate at best as discussed in [38]. The external stereovision image combines awareness of the target area with the ability to give elevation data. An added benefit of the stereovision image is the

ability to represent the data in a 3-D format. While this provides the operator with adequate detail and heightened situational awareness, the processing time for the image is much slower than that of a traditional monovision image which hinders real-time image update rate.

(a) In-situ monovision image of debris field

(b) Aerial monovision image of debris field



(c) Stereovision image of debris field captured from an aerial nadir view

Figure 5.9: Images used to conduct the vision system user study

5.2.4 Results of User Study

Following the description of the user study outlined in the previous section, subjects volunteered to participate in the experiment to evaluate the effectiveness of each of the vision systems in providing adequate data to the user for successful debris field navigation. The

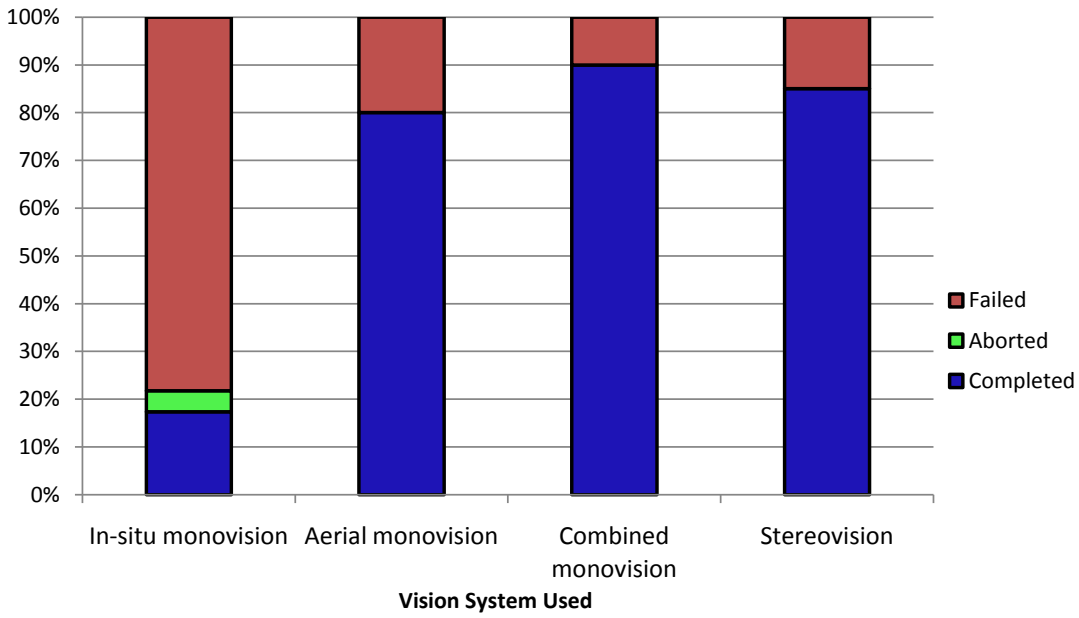


Figure 5.10: User study results grouped by vision system type and displayed as trial outcome percentages

test subjects had a varying range of familiarity with ground robot operations. Each user was randomly assigned one of the four vision systems: in-situ monovision, aerial monovision, combined in-situ and aerial monovision, and stereovision. The corresponding operator display interface was used by the user for debris field navigation. Each user was allowed one attempt to navigate the debris field with one of the vision systems. None of the users had prior knowledge of the debris field layout. Each saw the debris field for the first time during their navigation attempt. Each of the vision systems underwent 20 trials. Upon completion of the user study all results were compiled and analyzed. Results were grouped first by vision system type, then by mission outcome (pass/fail/abort). Figure 5.10 displays these results as the percentage of total trials for each of the four vision systems. It is easy to see that the in-situ monovision system resulted in a completion percentage below 20%, while the other three systems resulted in completion percentages at or above 80%. A possible cause for the low completion percentage of the in-situ monovision system was the camera field of view. Observations showed that it was very difficult for the operator to see the water

in the debris field due to it being at a lower elevation than the ground plane. The water was more recognizable when using the other systems as they offered a greater perspective of the entire debris field. Although the overall success of a system gives some indication of the effectiveness of that system, it is not an absolute metric. By analyzing the other recorded metrics one can determine which factors might make one system more successful than another. An analysis of each of the metrics was completed by first performing an F-test to determine equal or unequal variance between data sets. A one-tailed unpaired t-Test with a 95% confidence interval ($\alpha = 0.05$) was then performed to determine the statistical significance of any improvements[1] in the resulting metrics between each of the four vision systems. A presentation and analysis of the results follows.

One way to evaluate a system's effectiveness in providing adequate navigation and situational awareness data to the user is to examine the total number of moves required to navigate the course. During navigation the user attempts to make navigational decisions that will bring the user closer to the target objective. As a system provides the user with more situational awareness, the user will be enabled to make more decisions in advance or plan a navigational route through the debris field prior to executing that route. Seen below in Figure 5.11, the in-situ monovision system required nearly three times the required total moves to successfully navigate the course than the other three systems. For the in-situ monovision system, the user is required to move around the debris field to locate the target objective. This scanning method of locating the object resulted in a mean average of 168.5 total moves required to successfully navigate the course and locate the target. The other systems however give a greater perspective of the debris field and allow the user to locate the target objective prior to making any command decisions. The user then needs only to plan the appropriate navigation path to the target. These three systems (aerial monovision, combined monovision, and stereovision) required significantly less total moves (˜60) to successfully navigate the course and locate the target compared to in-situ monovision. Although there was some variation in the average number of total moves required to navigate the debris field among these three

[1] An improvement is considered to be any decrease in metric from one system to another

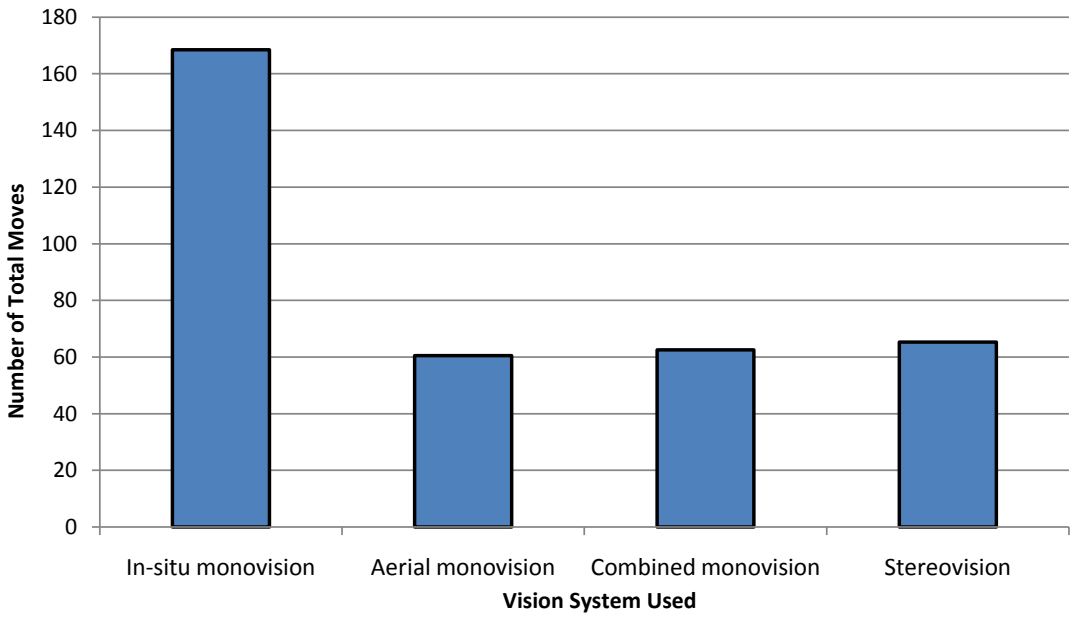


Figure 5.11: Average total moves required to successfully navigate the debris field

vision systems (aerial monovision[2], combined monovision[3], and stereovision[4]) no statistical significance was found in the variation of these results. Another metric that can be used to evaluate a system's ability to provide adequate navigation and situational awareness data is the total time required to navigate the debris field. This time was measured from when the user started the simulation and the vision system data was displayed on the interface to the end of the mission (completion or failure). The average total time required to successfully navigate the debris field for each of the four vision systems is presented in Figure 5.12. It can be seen that the in-situ monovision system performed poorly under this metric as well. It took the user an average of 226.9 seconds to navigate the debris field, 134.5 seconds longer than the next highest average time belonging to the combined monovision system. Aerial monovision[5], combined monovision[6], and stereovision[7] all performed significantly bet-

[2]Unpaired t-Test assuming non-equal variance, $p = 0.012$
[3]Unpaired t-Test assuming non-equal variance, $p = 0.013$
[4]Unpaired t-Test assuming non-equal variance, $p = 0.014$
[5]Unpaired t-Test assuming non-equal variance, $p = 0.005$
[6]Unpaired t-Test assuming equal variance, $p = 0.001$
[7]Unpaired t-Test assuming non-equal variance, $p = 0.003$

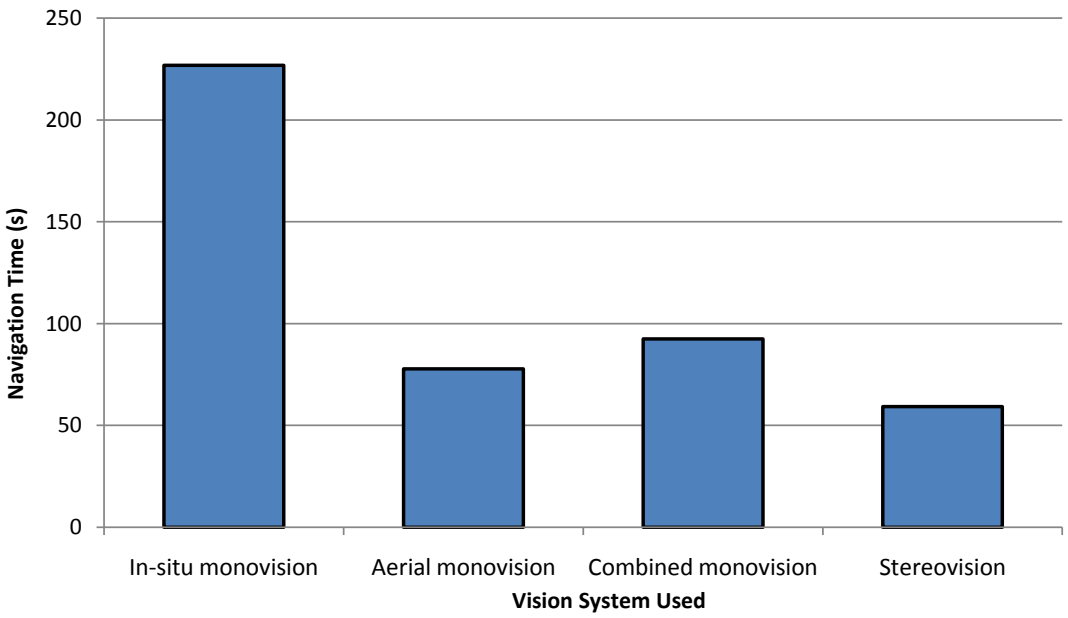


Figure 5.12: Average total time required to successfully navigate the debris field

ter than the in-situ monovision system for this metric. Furthermore, the stereovision system performed significantly better than the aerial[8] and combined monovision[9] systems for this metric. It took an average of 59.2 seconds for the users of the stereovision system to successfully navigate the debris field. For time-sensitive navigational missions stereovision is the best vision system candidate.

Related to the total time required to successfully navigate the course is the time required to make each move decision. The time required to make each move decision was recorded during each simulation, then the average was taken for each vision system (Figure 5.13). Just as with total time, the time was measured from when the user started the simulation and the vision system data was displayed on the interface to the end of the mission (completion or failure), the timer being reset after each move command. The time required to make a navigational decision was analyzed regardless of mission outcome, as it provides an indication of each vision system's ability to present data in a way that is quickly understood by the

[8]Unpaired t-Test assuming equal variance, $p = 0.013$
[9]Unpaired t-Test assuming equal variance, $p = 0.001$

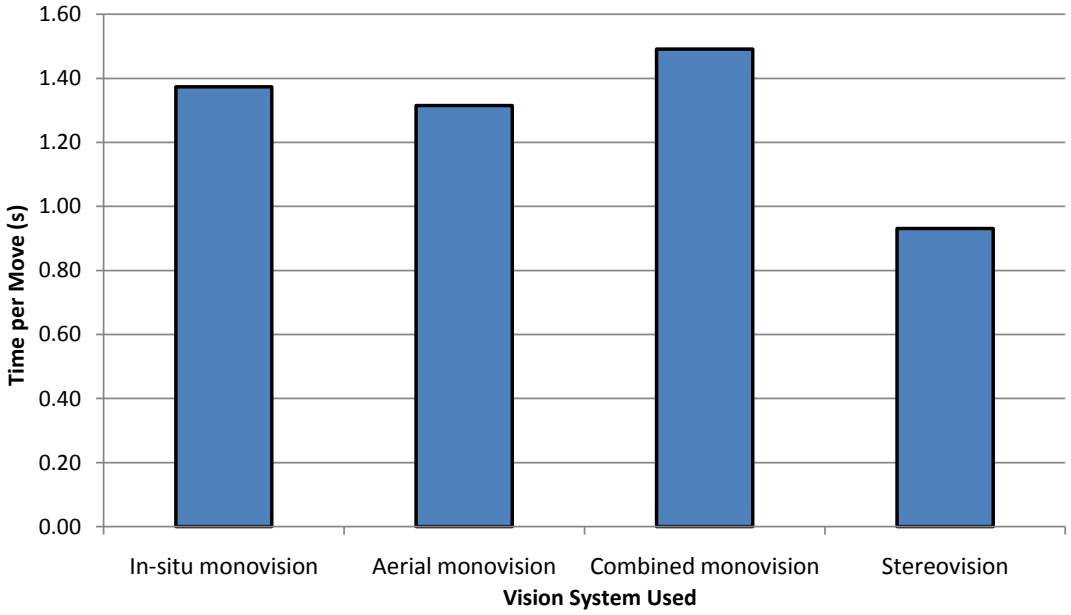


Figure 5.13: Average time required to make a navigational decision

user and allows them to make a navigational decision. The stereovision system required less time to make a navigational decision than the other three vision systems (in-situ monovision[10],aerial monovision[11], and combined monovision[12]). Statistical significance was found in this relationship, while no significance was found between these three systems. While it took the users an average of .99 seconds to make a navigational decision when using stereovision, it took nearly 50% longer for the other three systems.

The last metric used to analyze the effectiveness of each of the vision systems is the number of invalid moves. The number of invalid moves was recorded as mentioned previously for each user trial. The average number of invalid moves taken during successful attempts for each of the vision systems was analyzed and is presented in Figure 5.14. There was a statistically significant decrease in invalid moves made while using stereovision[13] and combined monovision[14] than when using in-situ monovision. While there was some variation in the

[10]Unpaired t-Test assuming non-equal variance, $p = 0.001$
[11]Unpaired t-Test assuming non-equal variance, $p = 0.001$
[12]Unpaired t-Test assuming equal variance, $p = 0.001$
[13]Unpaired t-Test assuming equal variance, $p = 0.01$
[14]Unpaired t-Test assuming equal variance, $p = 0.007$

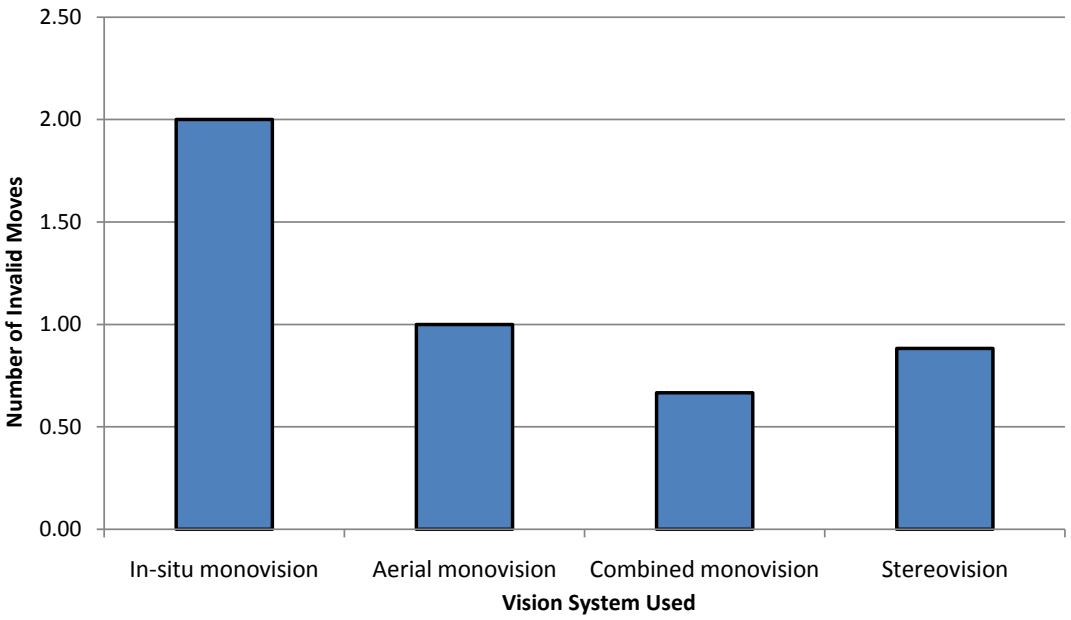


Figure 5.14: Average number of invalid moves made during successful debris field navigation

average number of invalid moves made when using stereovision and combined monovision, there was no statistical significance found for this variation.

5.2.5 Conclusions Drawn from User Study

We infer from the results of this study that an in-situ monovision system lacks the desired characteristics of a vision system for teleoperated navigation of a ground robot. While in-situ monovision provides the user with some detailed information about the robot's surroundings, it does not provide sufficient situational awareness to the user. The ability of in-situ monovision to provide situational awareness to the robot operator is largely dependent upon camera position, mounting height, mounting angle and field of view. According to the results of the study, the in-situ monovision system resulted in longer mission times, greater error in making navigational decision, and more importantly, a higher failure rate than the other three systems. For time-sensitive missions, these characteristics are undesirable. In contrast to the in-situ monovision system, the aerial monovision system provides little detailed informa-

tion about the robot's surroundings, but provides greater situational awareness. The aerial monovision system showed a decrease in total commanded moves and total navigation time from the in-situ monovision results. This is likely due to the greater situational awareness provided by the aerial monovision system, which allows the user to plan a navigational path prior to execution. There was no significant improvement in the metrics of time per move or the number of invalid moves from the in-situ monovision system to the aerial monovision system. One may conclude from this that the combined monovision system provides the user with both detailed information of the robot's surroundings and improved situational awareness, potentially leading to an even greater improvement in performance. However, a significant improvement was not manifested in the results. In fact, total navigation time and time per navigation system both increased slightly. This may have been due to the need for the user to assess the information provided by both vision systems and process the information into meaningful data. The fourth vision system, while never showing a decrease in performance from any of the other three vision systems, did show significant improvements in performance in each of the metrics. This may be attributed to the stereovision system's ability to provide the user with all of the required navigational data in one display.

It is proposed that the stereovision and the combined monovision systems are the best candidates for teleoperated navigation of a ground robot. The elevation at which the aerial monovision imagery and the stereovision imagery are taken will largely affect the vision system's ability to provide adequate data to the user. The elevations chosen for this study were scaled to match a specific scenario. However, a future study may investigate the relationships between robot size, obstacle size and the elevation at which imagery is obtained. An optimization of this relationship may further increase the vision systems' ability to provide adequate data to the user for robot teleoperation.

Additionally, in order to provide a more accurate assessment of the discussed vision systems, some real-time teleoperation scenarios should be investigated. Such an investigation will factor in any latency caused by the communications system used and the processing required to display the data to the user. In addition, this real-time simulation should take into account

the reaction time afforded to human operators when the ground robot nears an obstacle. By taking these real-world properties into account, the most efficient approach to visual situational awareness can be more accurately selected.

Stereovision imagery provides many benefits in addition to the ones found by this user study. Because stereovision data is three-dimensional, the user can be provided with the ability to rotate the scene in order to gain a different perspective than the one offered by the in-situ monovision camera. Furthermore, the stereovision data can be used for the path planning algorithm as discussed in chapter 4. In addition to the analysis done for the path planning algorithm, additional analysis can be done on the stereo data and presented to the user. This could include coloration based on elevation, slope angle or terrain roughness. Future research could provide more information of how to best display this information to the user.

Chapter 6

Conclusion and Recommendations

This thesis presents the research and work done to design a helicopter deployable ground robot for use in hazardous environments. A summary of the work conducted towards this research has been collected here along with some suggestions for further research that may improve the overall system and add to work in this field. While each chapter of this thesis addresses a distinct area of work, all contribute to the overall design and successful implementation of a helicopter deployable ground robot for use in a hazardous environment.

6.1 Summary of Work

A set of design specifications were established for the design of the ground robot based on the mission objectives and the environmental conditions of the intended hazardous, rough-terrain deployment site. These design specifications were used to design and build a small,

helicopter deployable ground robot. Analysis of the design has been presented and the final design underwent testing in order to ensure that the design met the previously defined design specifications. Testing and analysis showed that the design of the robot meets the design specifications and is suitable for the intended application.

In addition, a path planning algorithm was developed to aid robot navigation. The A* search algorithm was employed and a cost function and heuristic function were developed that accurately model the optimization criteria. A terrain interaction model was developed and is used by the planning algorithm to determine path costs. Successful simulations of the path planning algorithm have been presented.

Finally, various vision systems were examined in order to improve teleoperation of the robot. A study was conducted that examined the use of stereovision images acquired from the helicopter for use in robot operations. In addition, the use of insitu monovision cameras and aerial monovision imagery were studied and compared with helicopter-provided stereovision imagery for remote robot operation. It was concluded that there are various benefits provided from the use of stereovision imagery, including increased telepresence and situational awareness for the robot operator. Stereovision provides the ability to analyze the terrain and provide the results of this analysis to the operator as an additional navigation aid to the path planning algorithm.

6.2 Suggestions for Future Work

While the research presented here provides solid groundwork for the eventual field deployment of this robotic system, the system is not complete and leaves room for future work to be done in this area of research. With regard to the mobility platform, the test results showed some deficiencies in the design. The first deficiency, as mentioned in chapter 3, is with the track system. The tracks used on the robot for this work were not intended for this particular application. A redesigned track, with consideration of the track-terrain interactions will greatly improve the mobility characteristics of the robot. Another design deficiency is the ground clearance of the robot. Although the robot is size limited based on the dimensions of the aerial platform being used for deployment, an increased amount of ground clearance would improve the robot mobility. This could be done by simply increasing the distance from the chassis bottom plate to the ground, or alternative mobility platforms could be investigated, such as the use of an articulated track design.

Although efforts were made to minimize the weight of the ground robot by making use of composite materials and the use of additive manufacturing, future work could include an in-depth stress analysis of the robot components using finite element analysis techniques. This stress analysis could lead to an optimization of the robot components in order to minimize the weight of the robotic system. In order to properly conduct the weight optimization, a better understanding of the vehicle loads would be required. These loads could be obtained by employing computational tools that simulate the ground robot's motion over various terrain such as those discussed in [42]. This simulated mobility testing would be well complimented

by performing mobility tests on real terrain. This real-terrain mobility testing could be characterized using similar methods as those discussed in [43], and the test data could be used to improve the mobility characteristics of the ground robot and be used to improve the path planning simulations discussed in chapter 4.

In addition to items listed above for improving the mobility platform, there still exist many technologies that need to be developed prior to deploying the ground robotic system in a real-world scenario. Some of these technologies include research into methods of deploying and retracting the tethered robot from the helicopter, integration of the robot onto the aerial platform, and further development of the robotic subsystems, including the sensors on-board the robot, the sampling capabilities of the robot, and improvements to the communications and control of the robot. Much of this work is already in progress at the Unmanned Systems Lab, and will be integrated into the system as the research is completed.

With regard to the work done towards developing a path planning algorithm for rough terrain environments, there exist several areas that could be improved. The first area of improvement is the algorithm efficiency. The goal of this work was to develop and implement an algorithm in order to prove its effectiveness. The algorithm speed and efficiency were not a primary concern; however, in order to implement this algorithm for use in an actual mission scenario, some effort should be made to improve the algorithm efficiency. This could by done by converting the planning algorithm from Matlab to C++. Another limitation of the algorithm is the high computational expense of the cost function, particularly the computation of the robot-surface interaction model. Although the cost function and the robot-surface interaction model accomplish the goals of developing a rough terrain path

planning algorithm for time optimal paths, research can be done to find ways of improving the algorithm and increasing the efficiencies of these computations.

During the planning simulations it was recognized that the lack of a collision detection model in the algorithm may be problematic. During several instances it was observed that the terrain surface penetrated the bottom chassis of the robot. Once again, the inclusion of this collision detection model would decrease the speed of the algorithm; however, it would be beneficial for obstacle and collision avoidance.

Another method of improving the algorithm would be to include terrain types and track slippage in the planning algorithm. As discussed in chapter 4, the stereo imagery only provides terrain topology information to the algorithm. From the mobility testing that was conducted and presented in chapter 3, it was observed that the ground robot performs differently on different terrain types. A method of including the results of the mobility testing, or other terrain classification data in the algorithm may improve the correlation of the paths planned by the algorithm to the feasible paths over the actual terrain. In addition to the terrain data, a probabilistic approach could be taken to determining track slippage which may improve the robustness of the algorithm and the feasibility of the planned paths.

In addition to the algorithm improvements listed above, it was suggested in chapter 4 that the planned paths could be presented to the user in order to aid in beyond line-of-sight navigation of the robot. Research must still be conducted into this area in order to determine how to present the path to the user, how to present the robot's progress along the path to the user, and methods of determining whether presenting the path to the user aids or hinders the user's ability to operate the robot successfully. These trials may be conducted by providing

the user with a 3-D terrain model and an overlay of the path on the terrain. Both GPS and computer vision methods are proposed as possible solutions for presenting the robot's progress along the path, although the uncertainty of GPS measurements may prove to be problematic for tracking the robot's location. Future research into methods of tracking the robot's location may provide a solution to the problem of using 3-D terrain data combined with the planned path for robot navigation.

With respect to vision systems to aid in robot navigation, the simulated supervisory control study of various vision systems presented in chapter 5 laid a foundation for future research into the benefit that various vision systems can provide to the user of a teleoperated robot. In order to improve this work, a real-time vision system study needs to be conducted. A real-time vision system study will introduce some additional complexity into the problem, including the latency caused by the communications system used for the study and the processing required to display the data to the user. In addition, the reaction time afforded to human operators when the ground robot nears an obstacle will be introduced into the simulation. A real-time simulations also presents some additional challenges. The study conducted for this research found stereovision to be an effective method of providing situational awareness to the operator during robot navigation. As mentioned previously, effective methods of tracking the robot's location and presenting that location to the operator are still being investigated. These methods must provide accurate positional information to the user, such that the user's ability to operate the robot is not hindered by the use of 3-D terrain data. By conducting a real-world teleoperation study, the most efficient approach to visual situational awareness can be more accurately selected.

In addition to the benefits described in this body of work, stereovision provides the benefit of viewing the terrain from additional angles and the benefit of processing the data to obtain additional information about the environment. When using stereovision data for navigation, the scene can be rotated because it is three-dimensional, in order to gain a different perspective than the one a monovision camera on-board the robot may provide. Analysis can also be performed on the 3-D terrain data such as the analysis performed for the path planning algorithm as presented in chapter 4. Additional analysis can be done on the stereo data and presented to the user, such as coloration based on elevation, slope angle or terrain roughness. In addition to terrain analysis, sensor feedback could be displayed to the user on the 3-D data as a color map in order to aid in robot operations. Future research could provide more information about the effectiveness and benefits of presenting this additional information to the user during robot operations.

By implementing some of these suggestions for future work, a more robust helicopter deployable ground robot for use in hazardous environments can be developed. However, each of the three design aspects (mobility, navigation, and operator interface) contribute to the future deployment of the remotely operated robotic system in hazardous environments. In conclusion, the research presented has laid the groundwork for interesting additions to this area of study in the future, some of which are already being researched by other students in the Unmanned Systems Laboratory.

Bibliography

[1] Hager, D., *Situational Awareness of a Ground Robot from an Unmanned Aerial Vehicle*, Master's thesis, Virginia Polytechnic Institute and State University, 2009.

[2] Endsley, M. R., "Toward a theory of situation awareness in dynamic systems," *Human Factors*, Vol. 37, No. 1, 1995, pp. 32–64.

[3] Brewer, E. T., *Autonomous Localization of* $1/R^2$ *Sources Using an Aerial Platform*, Master's thesis, Virginia Polytechnic Institute and State University, 2009.

[4] Trevelyan, J. P., Kang, S.-C., and Hamel, W. R., *Springer Handbook of Robotics*, chap. 48, Springer, 2008, pp. 1101–1126.

[5] Casper, J. and Murphy, R. R., "Human-Robot Interactions During the Robot-Assisted Urban Search and Rescue Response at the World Trade Center," *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, Vol. 33, No. 3, June 2003, pp. 367–385.

[6] Murphy, R. R., Tadokoro, S., Nardi, D., Jacoff, A., Fiorini, P., Choset, H., and Erkmen, A. M., *Springer Handbook of Robotics*, chap. 50, Springer, 2008, pp. 1151–1173.

[7] Duan, X., Huang, Q., Rahman, N., Li, J., and Li, J., "MOBIT, A Small Wheel–Track–Leg Mobile Robot," *Proceedings of the 6th World Congress on Intelligent Control and Automation*, Dalian, China, June 2006, pp. 9159–9163.

[8] Matthies, L., Xiong, Y., Hogg, R., Zhu, D., Rankin, A., Kennedy, B., Hebert, M., Maclachlan, R., Won, C., Frost, T., Sukhatme, G., McHenry, M., and Goldberg, S., "A portable, autonomous, urban reconnaissance robot," *Robotics and Autonomous Systems*, Vol. 40, 2002, pp. 163–172.

[9] Park, H.-W., Kim, S.-H., Park, N.-C., Yang, H.-S., Park, Y.-P., Kim, S.-H., Park, Y.-H., and Kang, Y.-H., "Design of Tracked Vehicle with Passive Mechanism for Uneven Terrain," *SICE-ICASE International Joint Conference*, 2006, pp. 3132–3136.

[10] Yamauchi, B., "PackBot: A versatile platform for military robotics," *Proceedings of SPIE*, Vol. 5422, 2004, pp. 228–237.

[11] Bruch, M., Laird, R., and Everett, H., "Challenges for deploying man-portable robots into hostile environments," *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. 4195, 2001, pp. 29–40.

[12] Greer, D., McKerrow, P., and Abrantes, J., "Robots in Urban Search and Rescue Operations," *Proceedings of the 2002 Australasian Conference on Robotics and Automation*, November 2002, pp. 25–30.

[13] Ratner, D. and McKerrow, P., "Aerial tethered robotic system with hovering-hopping agents for security and rescue operations," *AUVSI Unmanned Systems North America Conference*, Vol. 24, No. 5, Nov/Dec 2006, p. 25.

[14] McKerrow, P. and Ratner, D., "The design of a tethered aerial robot," *IEEE International Conference on Robotics and Automation*, 2007, pp. 335–360.

[15] Bisgaard, M., *Modeling, Estimation, and Control of Helicopter Slung Load System*, Ph.D. thesis, Aalborg University, 2007, Department of Electronic Systems, Section for Automation and Control.

[16] Schulteis, T. M. and Price, J. G., "Project stork UAV/UGV collaborative initiative," *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. 5422, 2004, pp. 414–425.

[17] Wills, M., Ciccimaroa, D., Yee, S., Denewiler, T., Stroumtsos, N., Messamored, J., Brownd, R., Skibba, B., Clapp, D., Wit, J., Shirts, R. J., Dion, G. N., and Ansea, G. S., "Joint collaborative technology experiment," *Proceedings of SPIE - The International Society for Optical Engineering*, Vol. 7332, 2009, pp. 14.

[18] Sheridan, T. B., *Telerobotics, automation, and human supervisory control*, MIT Press, Cambridge, MA, USA, 1992.

[19] Nielsen, C. W., Gertman, D. I., Bruemmer, D. J., Hartley, R. S., and Walton, M. C., "Evaluating Robot Technologies as Tools to Explore Radiological and Other Hazardous Environments," *12th Topical Meeting on Robotics and Remote Systems for Hazardous Environments*, March 2008, pp. 9, INL/CON-07-13568, available at: [`http://www.osti.gov/bridge/product.biblio.jsp?osti_id=927622`].

[20] Bekey, G., *Autonomous robots: from biological inspiration to implementation and control*, The MIT Press, 2005.

[21] Hornback, P., "The Wheel Versus Track Dilemma," *ARMOR Magazine*, March-April 1998, pp. 33–34.

[22] Defense, S. o., "Unmanned Systems Integrated Roadmap 2009-2034," Tech. rep., Department of Defense, Washington, DC, 2009.

[23] LaValle, S., *Planning algorithms*, Cambridge University Press, 2006.

[24] Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische Mathematik*, Vol. 1, 1959, pp. 269–271.

[25] Cormen, T. H. et al., editors, *Introduction to Algorithms*, MIT Press, 2nd ed., 2001.

[26] Barr, A. and Feigenbaum, E. A., editors, *The handbook of Artificial Intelligence*, Vol. I, HeurisTech Press, Stanford, California, 1981.

[27] Nilsson, N. J., *Principles of Artificial Intelligence*, Tioga Publishing Co., 1980.

[28] Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence*, Vol. 1, Wiley-Interscience publication, 2nd ed., 1992.

[29] Shapiro, S. C., editor, *Encyclopedia of Artificial Intelligence*, Vol. 2, Wiley-Interscience publication, 2nd ed., 1992.

[30] Arrowsmith, J. R., "Notes on Lidar interpolation," Department of Geological Sciences, Arizona State University.

[31] Detweiler, Z. and Ferris, J. B., "Uniform Grid Spacing for Three Dimensional High Fidelity Terrain Profiles," *Proceedings of the 16th International Conference of the International Society for Terrain-Vehicle Systems*, Paper ID Number 14, 2008, ISTVS, Turin, Italy.

[32] Shiller, Z. and Serate, W., "Trajectory planning of tracked vehicles," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 117, 1995, pp. 619.

[33] Morales, J., Martínez, J., Mandow, A., García-Cerezo, A., Gomez-Gabriel, J., and Pedraza, S., "Power Analysis for a Skid-Steered Tracked Mobile Robot," *2006 IEEE International Conference on Mechatronics*, July 2006, pp. 420–425.

[34] Martínez, J. L., Mandow, A., Morales, J., Pedraza, S., and García-Cerezo, A., "Approximating Kinematics for Tracked Mobile Robots," *International Journal of Robotics Research*, Vol. 24, No. 10, 2005, pp. 867–878.

[35] McLean, G., Prescott, B., and Podhorodeski, R., "Teleoperated system performance evaluation," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 24, No. 5, 1994, pp. 796–804.

[36] Kofman, J., Wu, X., Luu, T., and Verma, S., "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Transactions on Industrial Electronics*, Vol. 52, No. 5, October 2005, pp. 1206–1219.

[37] Culhane, A., *Development of an Obstacle Detection System for Human Supervisory Control of a UAV in Urban Environments*, Master's thesis, Virginia Polytechnic Institute and State University, 2008.

[38] Klomparens, D., *Automated Landing Site Evaluation for Semi-Autonomous Unmanned Aerial Vehicles*, Master's thesis, Virginia Polytechnic Institute and State University, 2008.

[39] Davies, E. R., *Machine Vision: Theory, Algorithms, Practicalities*, Morgan Kaufman, San Francisco, California, 3rd ed., 2005.

[40] McDonald, A., Crossley, S., Bennett, J., Cookmartin, G., Morrison, K., and Quegan, S., "Stereo vision measurements of soil surfaces and their utility in testing microwave backscattering models," *International Geoscience and Remote Sensing Symposium (IGARSS)*, Vol. 5, Hamburg, Ger, 1999, pp. 2634–2636.

[41] Marr, D. and Poggio, T., "A Computational Theory of Human Stereo Vision," *Proceedings of the Royal Society of London, Series B 204*, 1979, pp. 301–328.

[42] Lee, R. C., *Stochastic Terrain and Soil Modeling for Off-Road Mobility Studies*, Master's thesis, Virginia Polytechnic Institute and State University, 2009.

[43] Worley, M. E., *Experimental Study on the Mobility of Lightweight Vehicles on Sand*, Master's thesis, Virginia Polytechnic Institute and State University, 2007.

Appendices

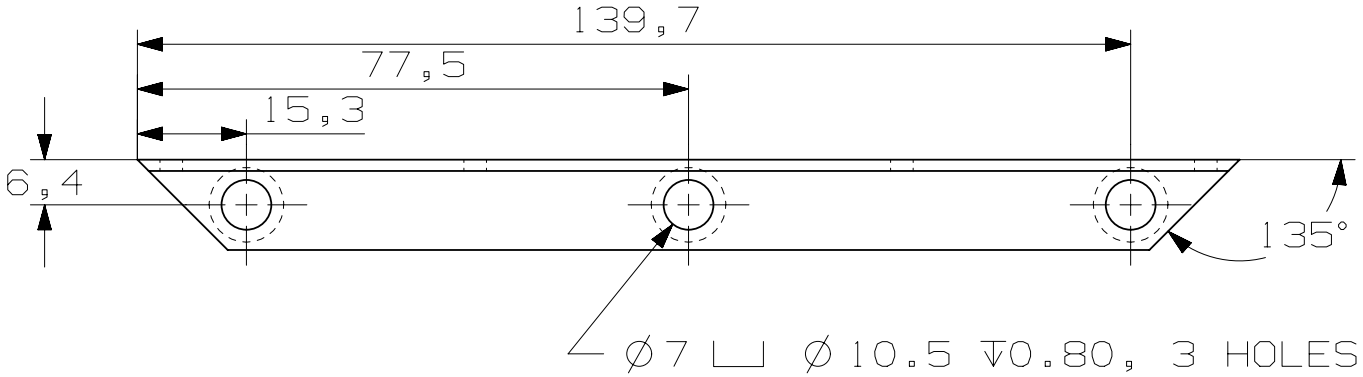
Appendix A

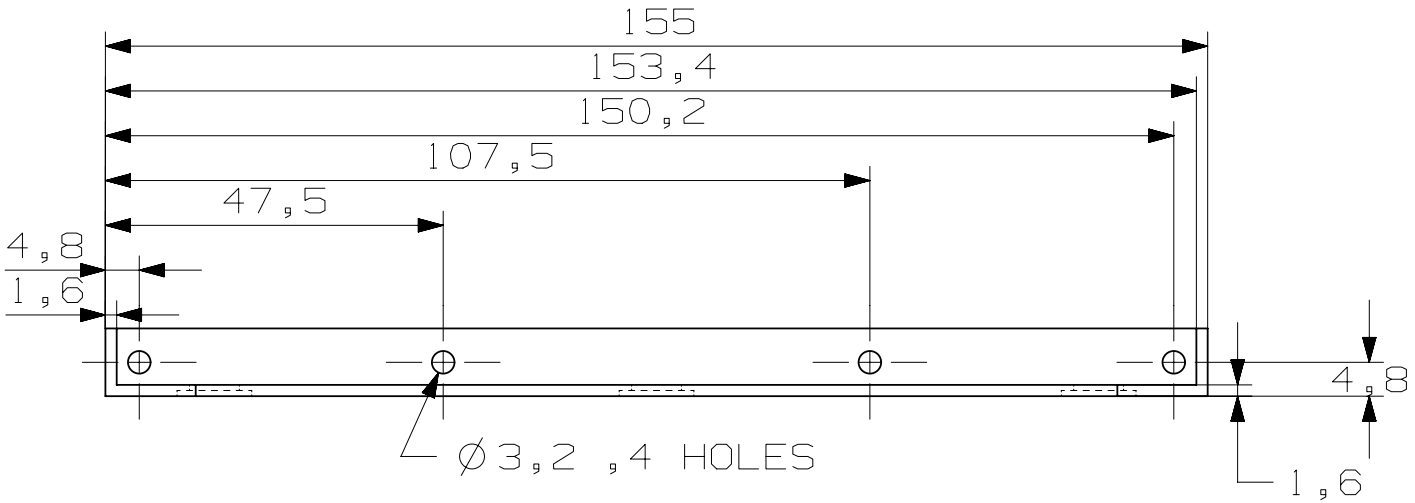
Robot Engineering Drawings

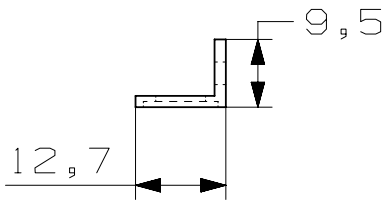
This appendix contains a set of engineering drawings for all custom designed parts used in the ground robot assembly. The table below presents some useful information regarding these parts.

ID #	Part Name	Material	Qty
RP01	Battery Tray - Bracket 1	Aluminum	4
RP02	Battery Tray - Bracket 2	Aluminum	4
RP03	Belt Cover	Composite[a]	2
RP04	Front Pulley	Polycarbonate[b]	2
RP05	Rear Drive Pulley	Polycarbonate[b]	2
RP06	Chassis Bottom Plate	Composite[c]	1
RP07	Chassis Side Plate	Composite[a]	2
RP08	Pulley Axle	Stainless Steel	8
RP09	Idler Pulley	Nylon	4
RP10	Motor Bulkhead	Composite[c]	1
RP11	Motor Mount Bracket	Aluminum	2
RP12	Mounting Bracket - RP06 to RP07 (Left)	Aluminum	1
RP13	Mounting Bracket - RP06 to RP07 (Right)	Aluminum	1
RP14	Robot Track	Polyeurethane	2

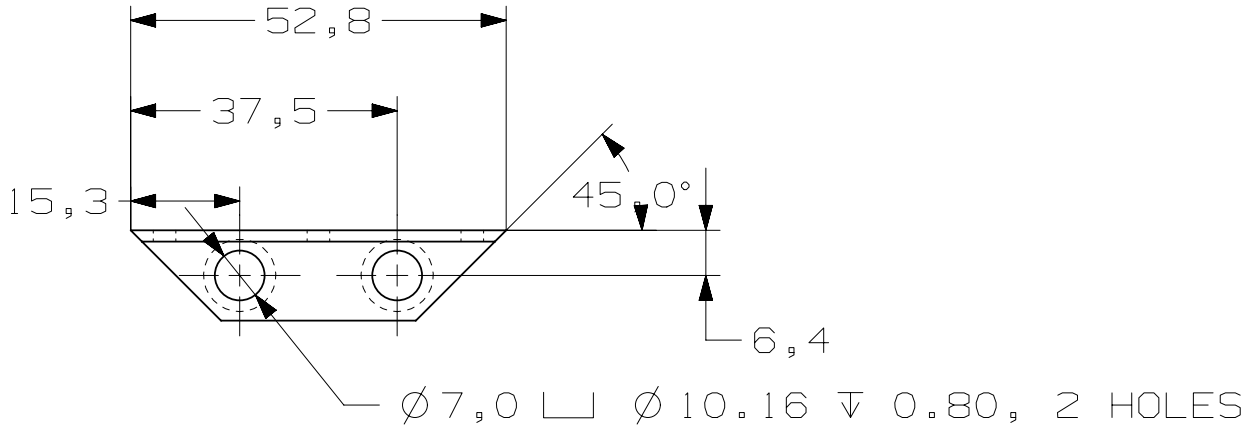
[a]Carbon Fiber w/ 0.125" balsa wood core and aluminum inserts at key interface points
[b]Rapid Prototyped using Stratasys Titan
[c]Carbon Fiber laminate - 15 ply

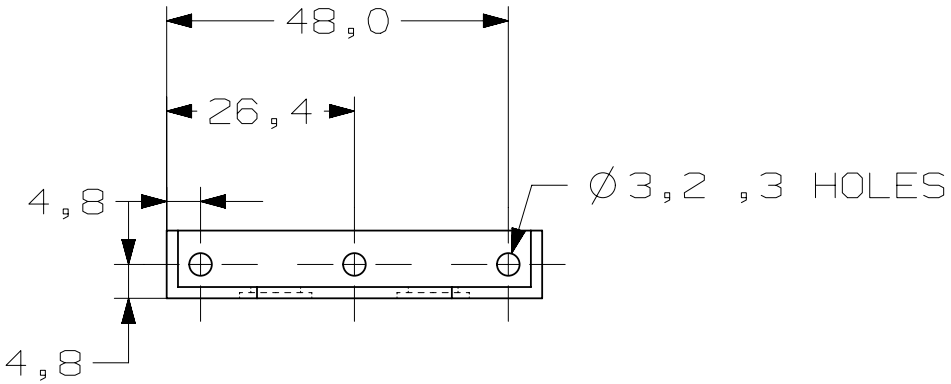


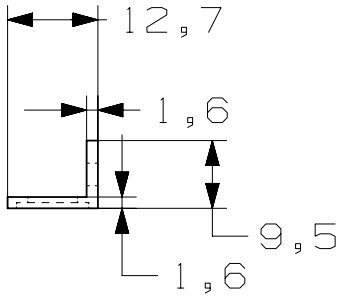




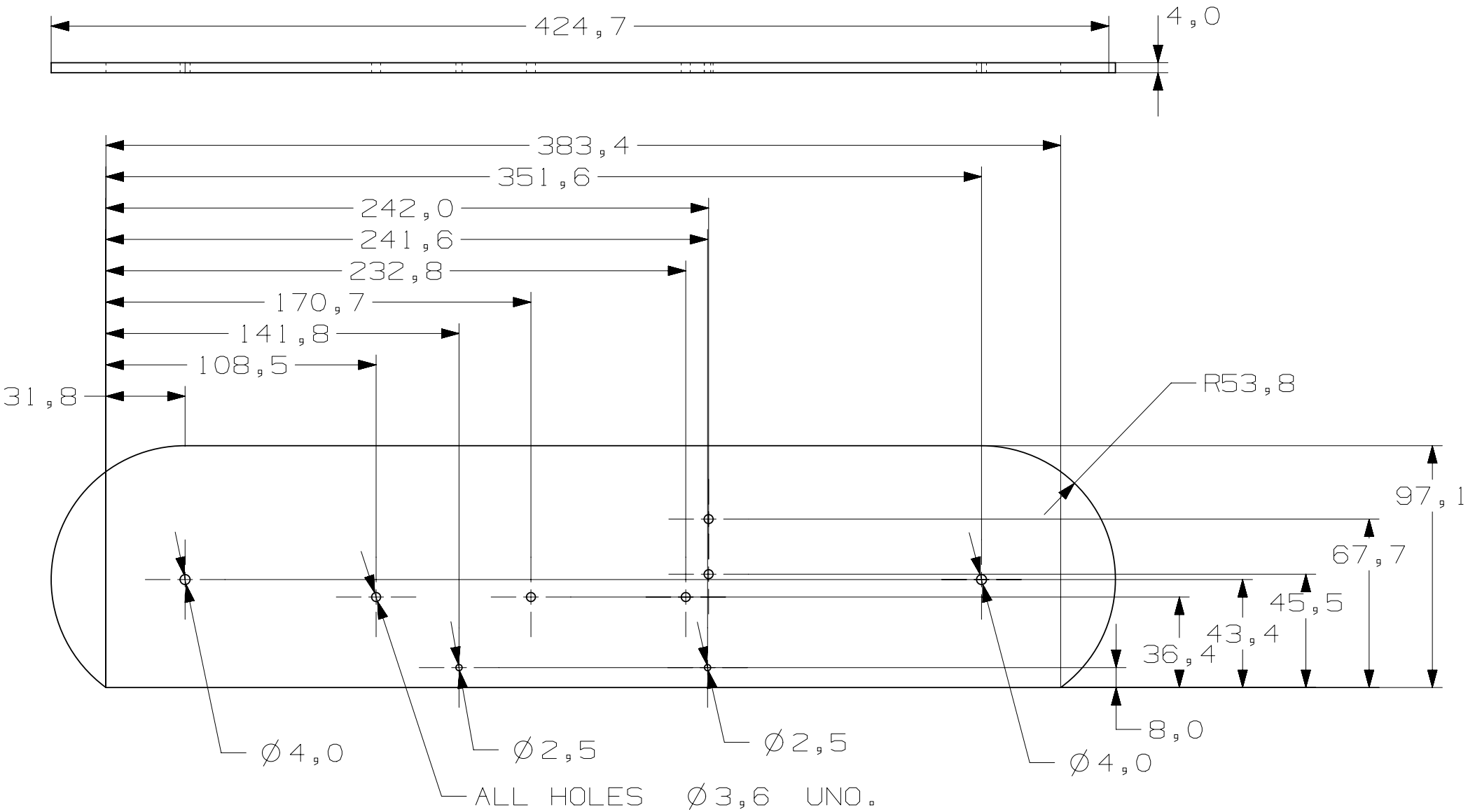
NOTE:
1. ALL DIMENSIONS IN MILLIMETERS
2. MATERIAL IS ALUMINUM





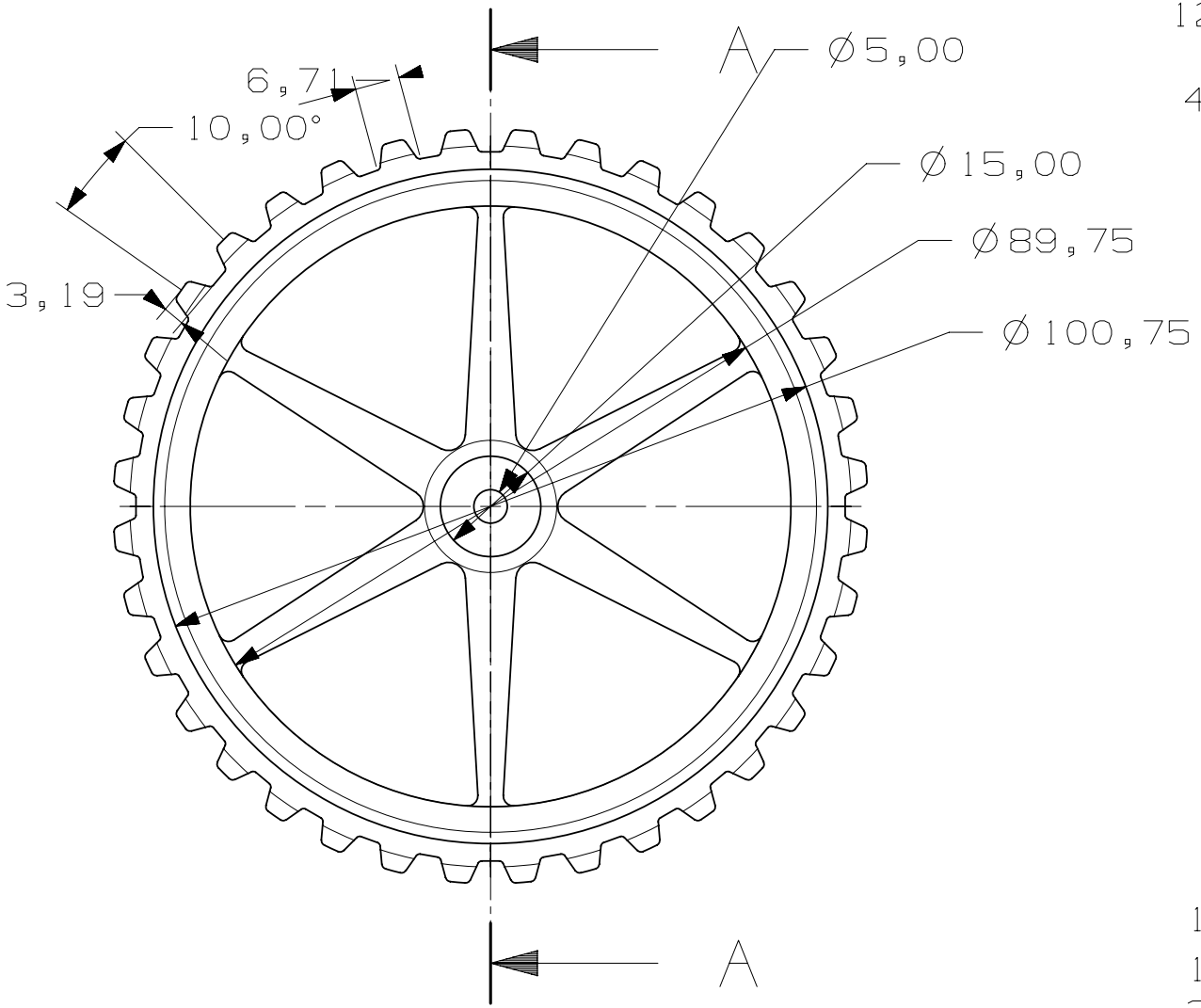


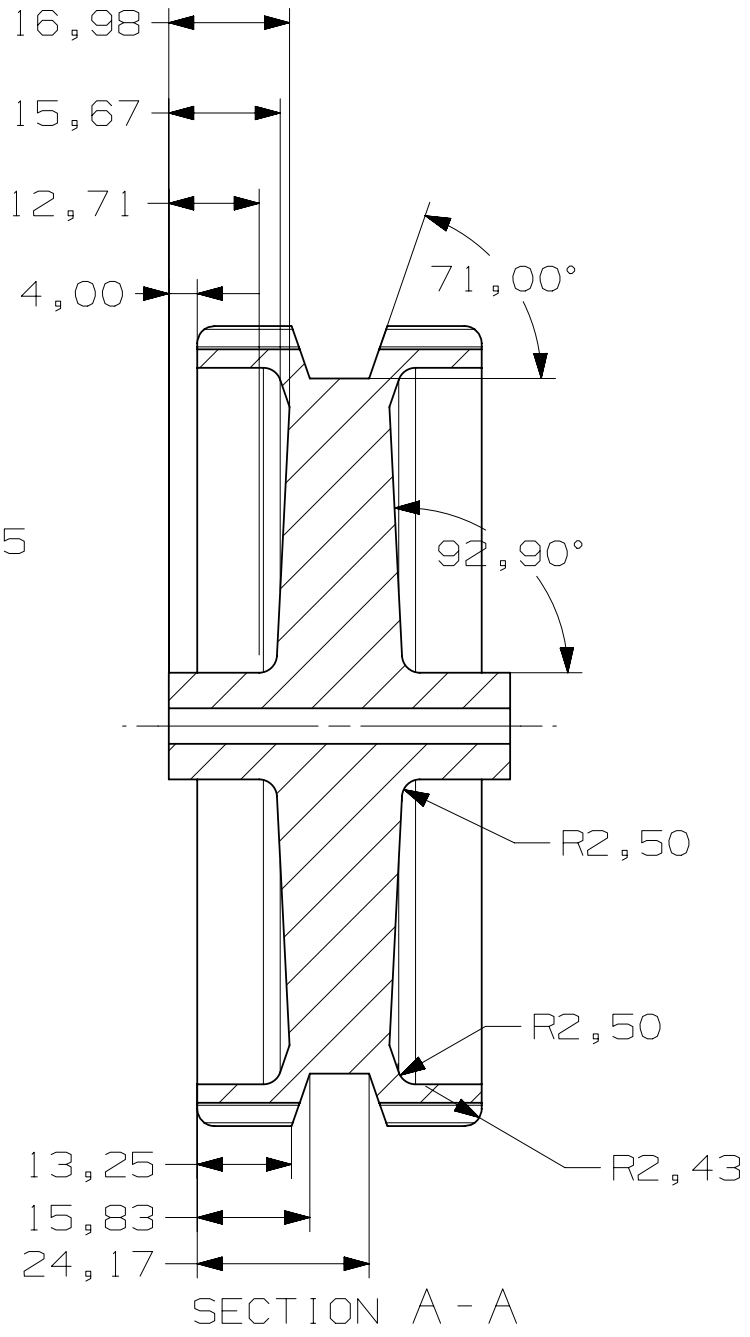
NOTE:
1. ALL DIMENSIONS IN MILLIMETERS
2. MATERIAL IS ALUMINUM



NOTE:
1. MATERIAL IS CARBON FIBER COMPOSITE WITH BALSA WOOD CORE
2. ALL DIMENSIONS IN MILLIMETERS

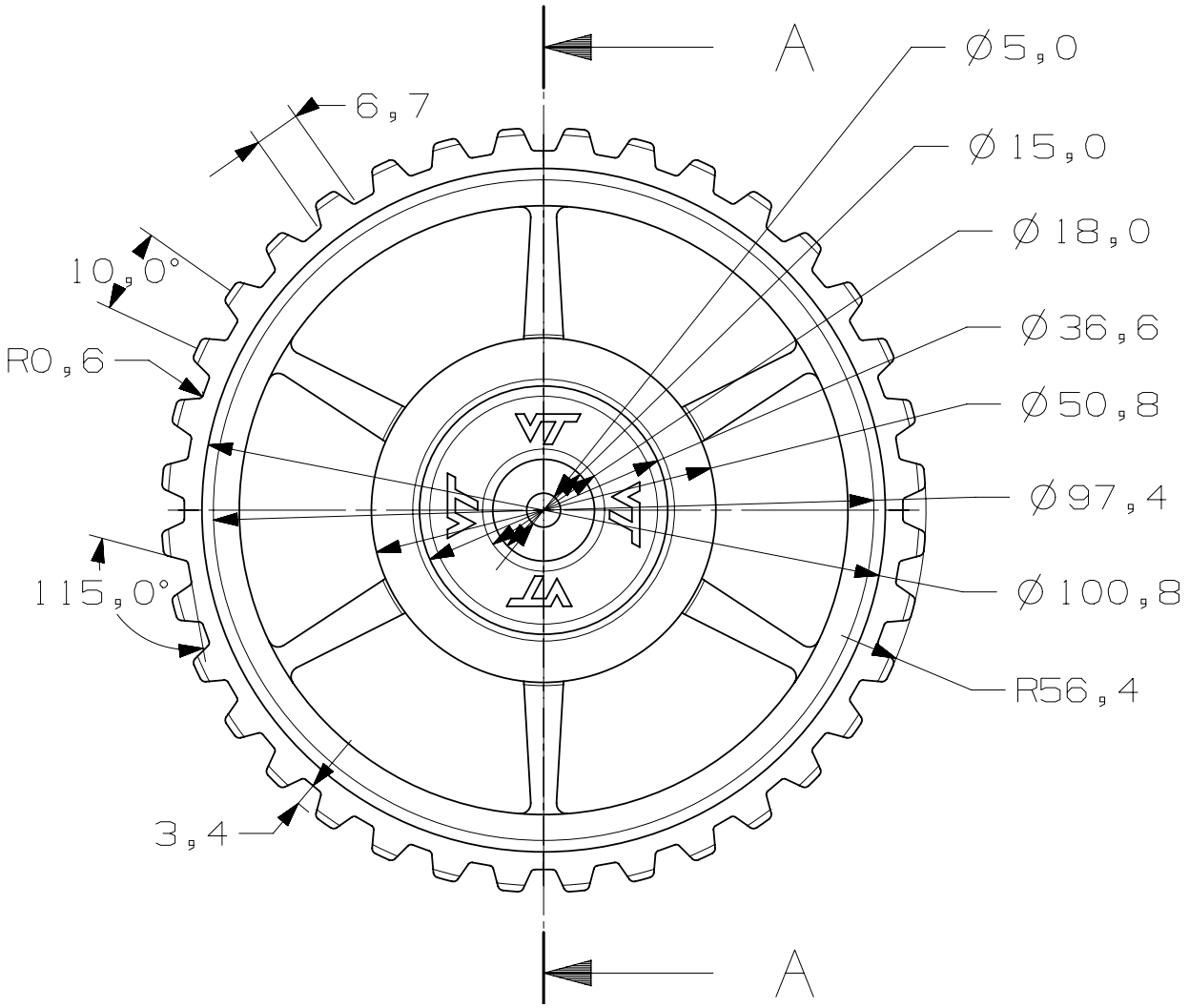
NOTE:
1. ALL DIMENSIONS ARE IN MILLIMETERS

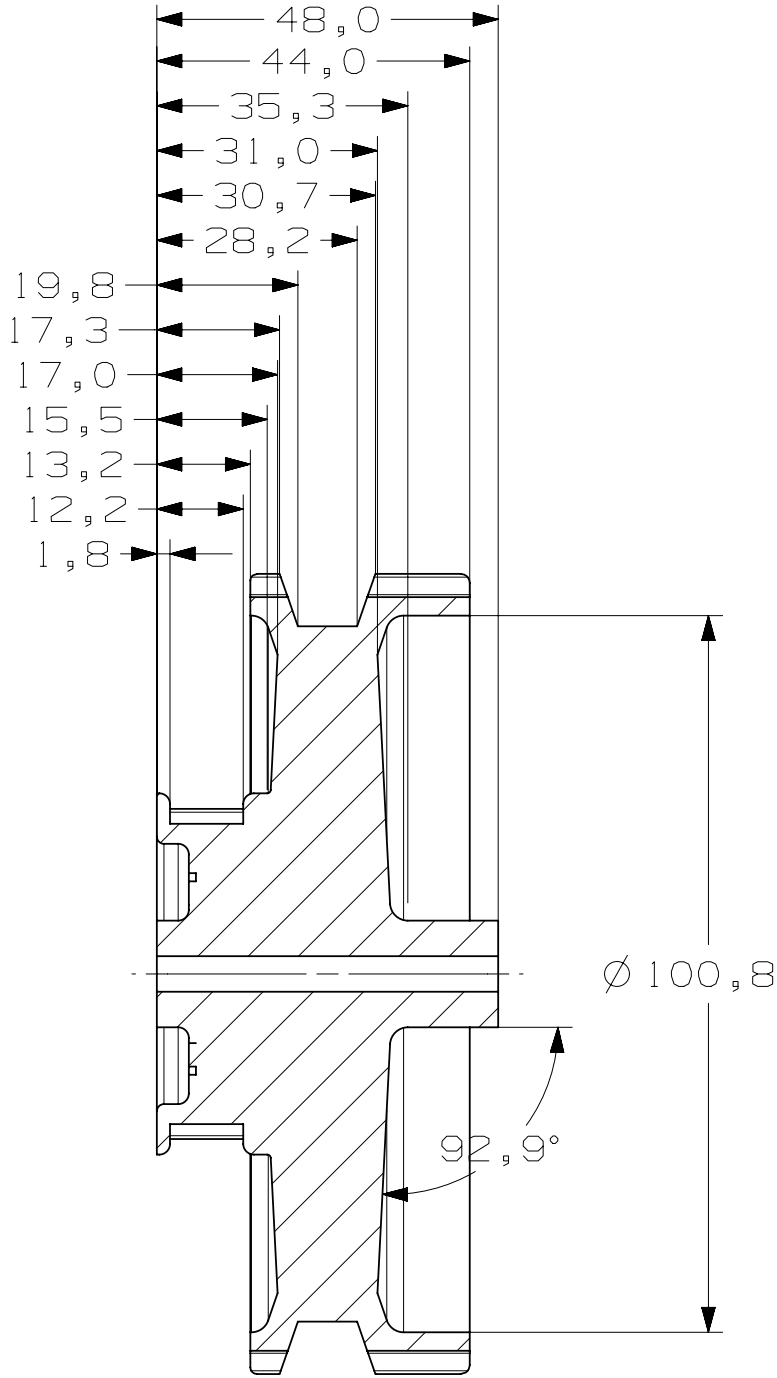




SECTION A - A

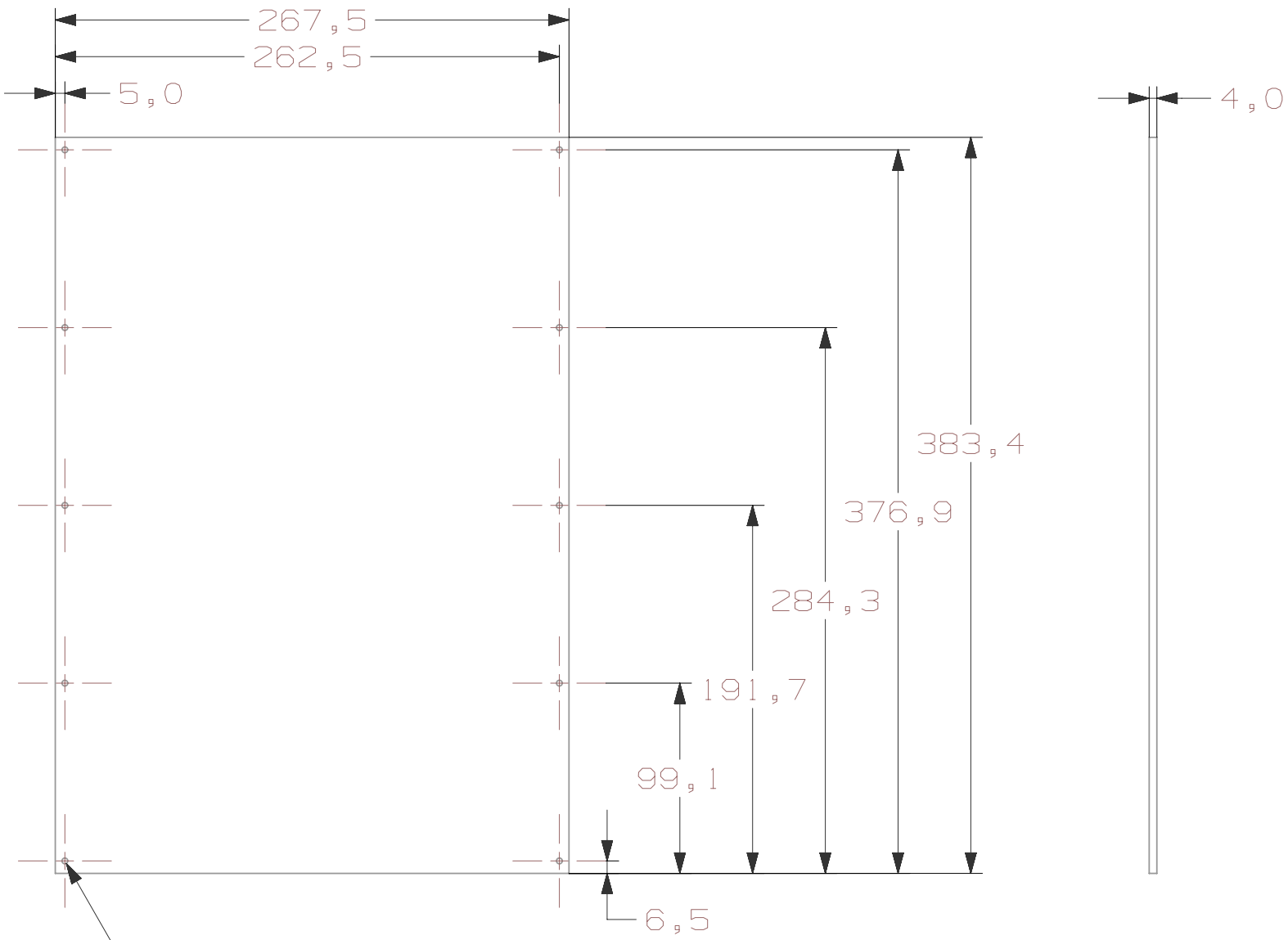
NOTE:
1. ALL DIMENSIONS ARE IN MILLIMETERS



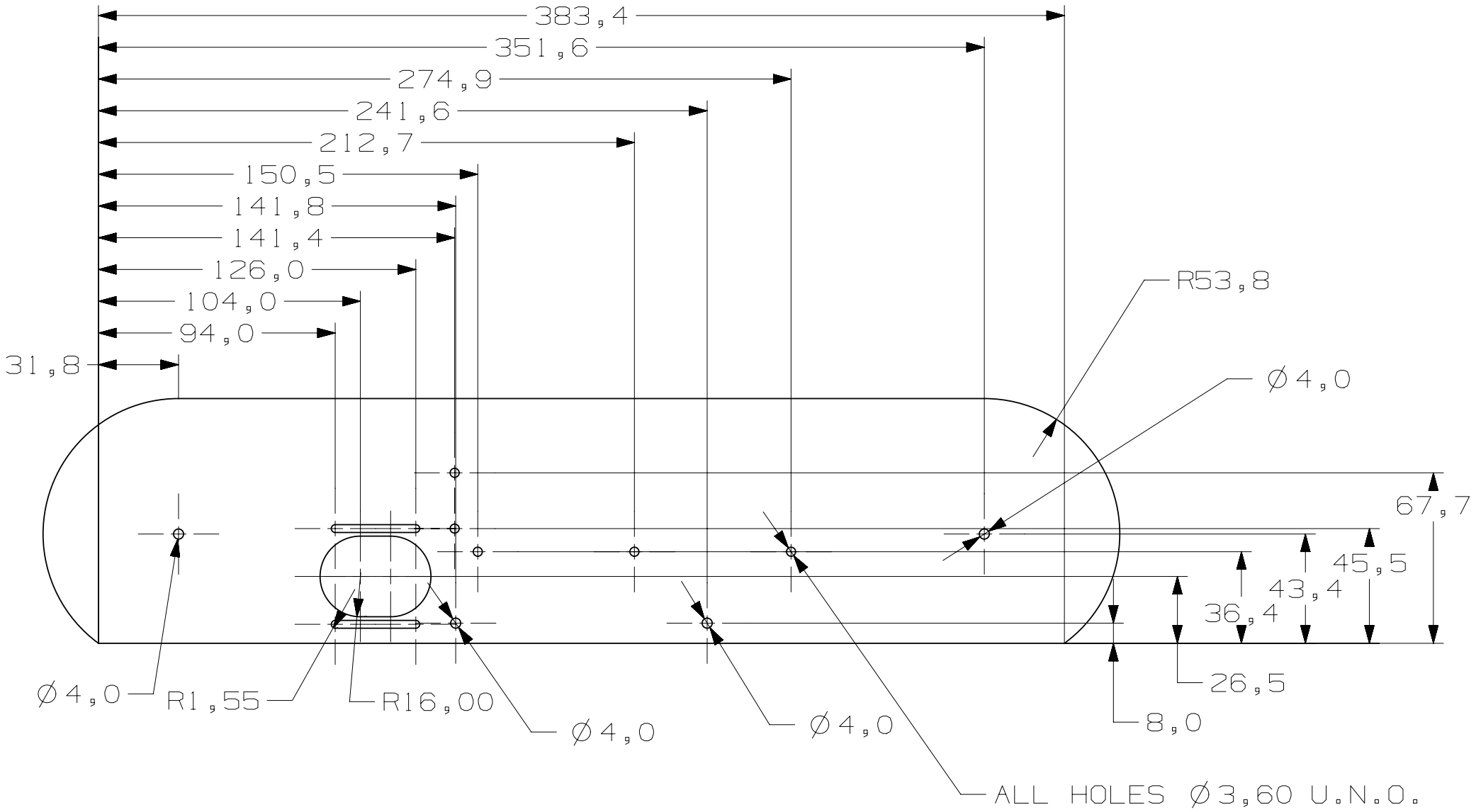


SECTION A-A

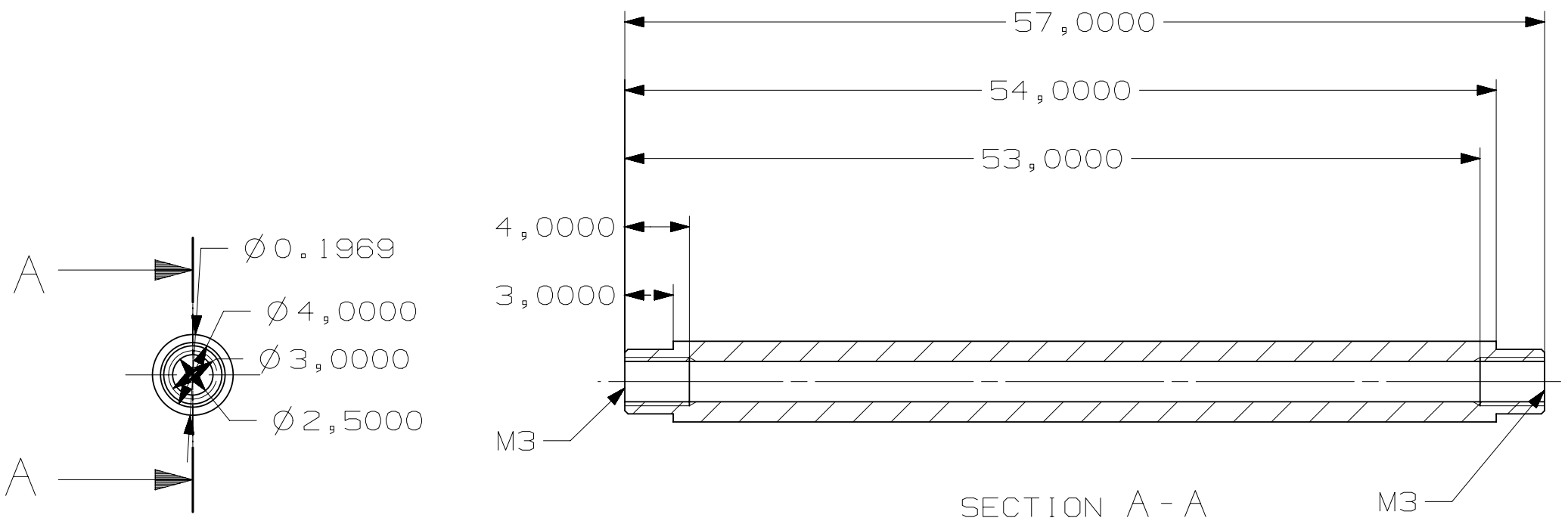
CHASSIS BOTTOM PLATE



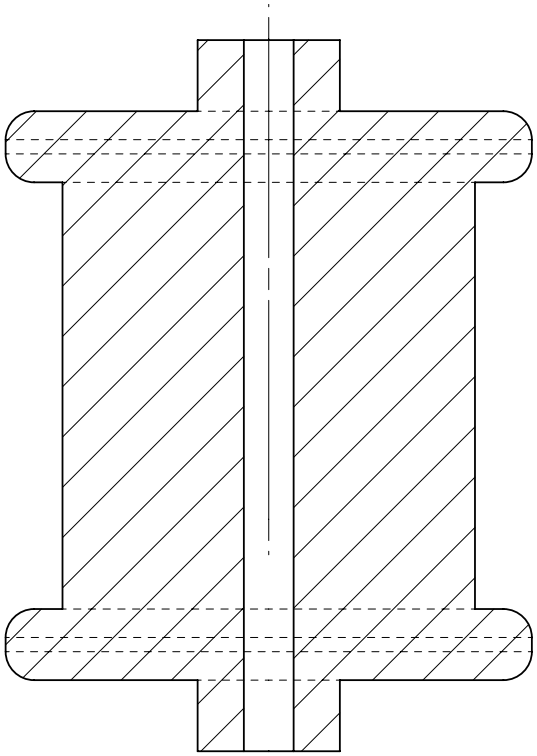
NOTE:
1. ALL DIMENSIONS ARE IN MILLIMETERS
2. MATERIAL IS CARBON FIBER COMPOSITE CONSISTING OF 15 LAYERS USING A $[0/90/\pm45]_s$ QUASI-ISOTROPIC LAYUP

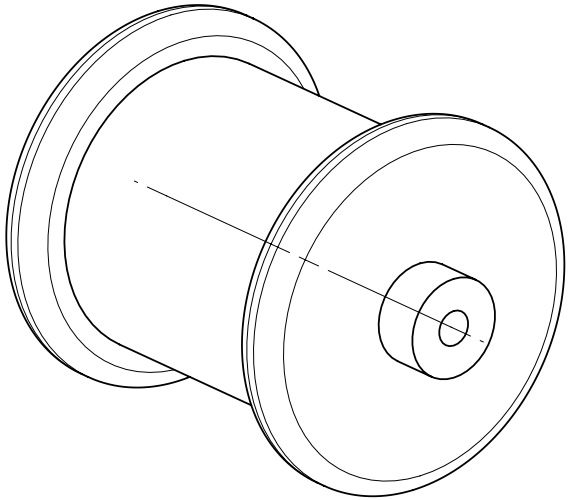


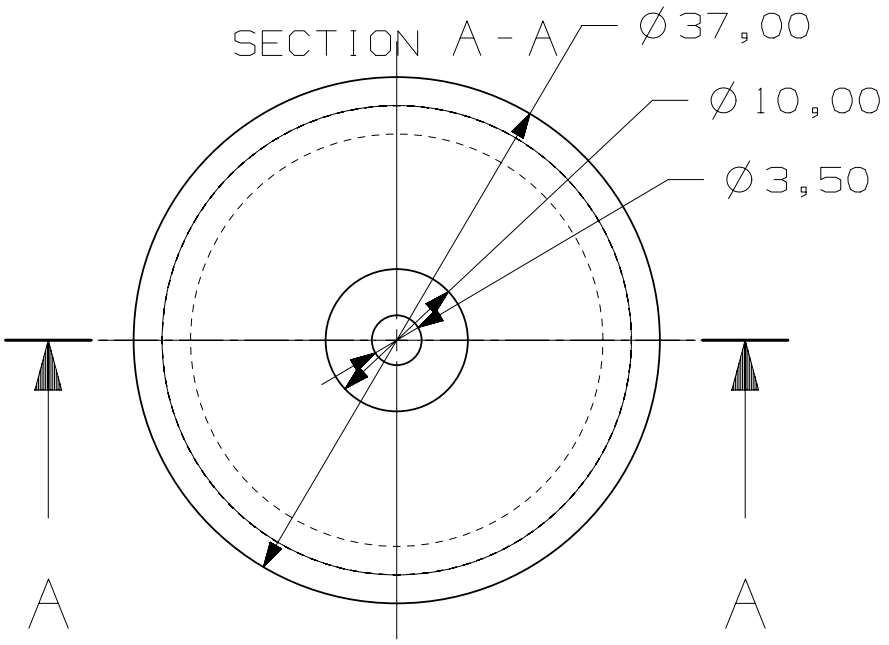
NOTE:
1. ALL DIMENSIONS ARE IN MILLIMETERS
2. MATERIAL IS CARBON FIBER COMPOSITE WITH BALSA WOOD CORE, THICKNESS EQUAL TO 5.00 MM

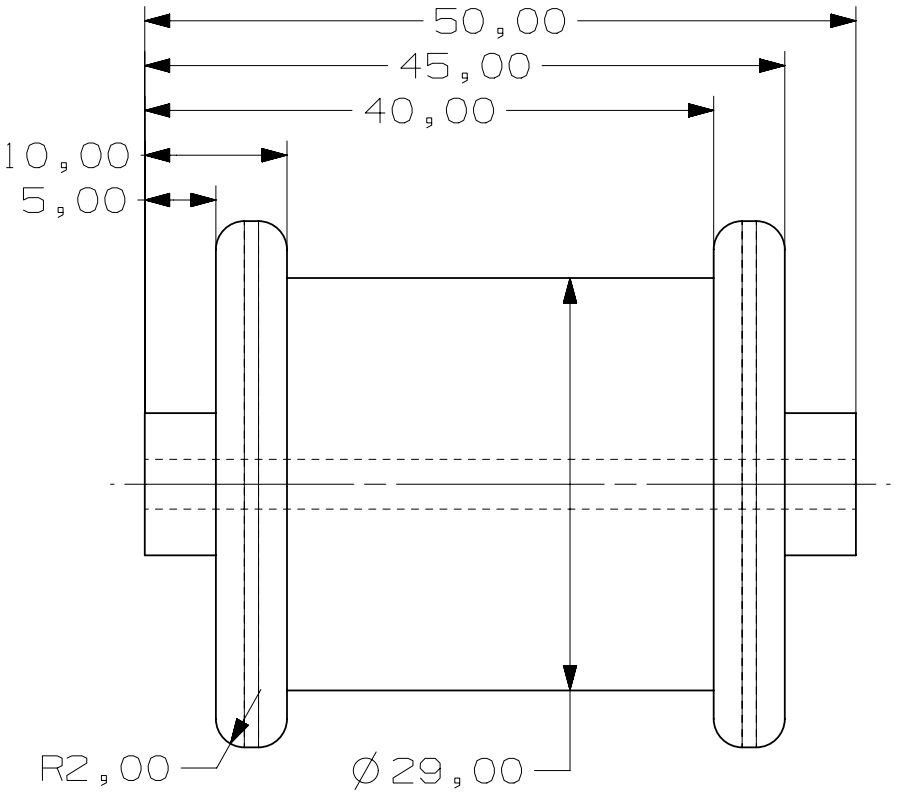


NOTE:
1. ALL DIMENSIONS IN MILLIMETERS
2. MATERIAL IS STEEL

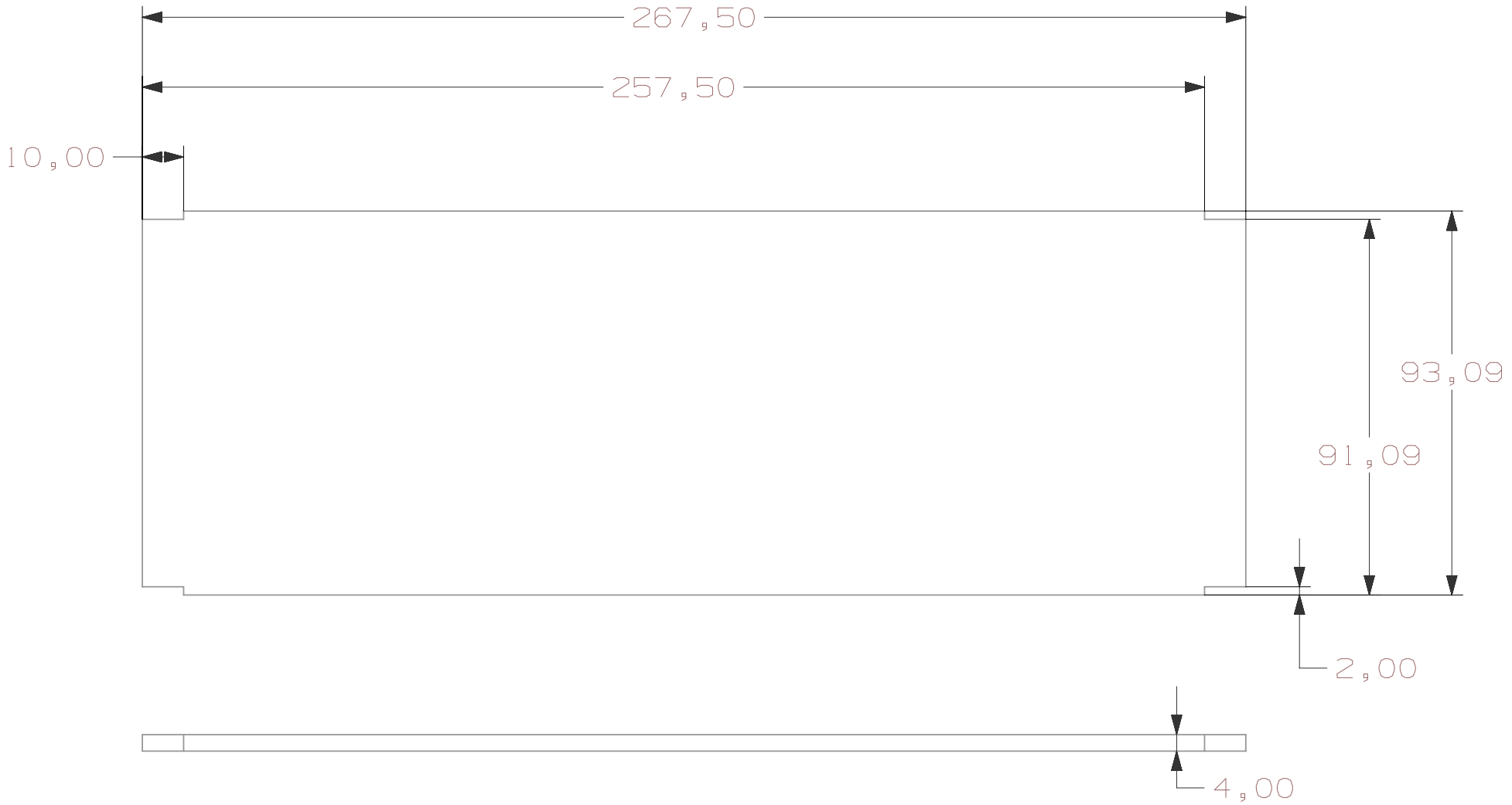




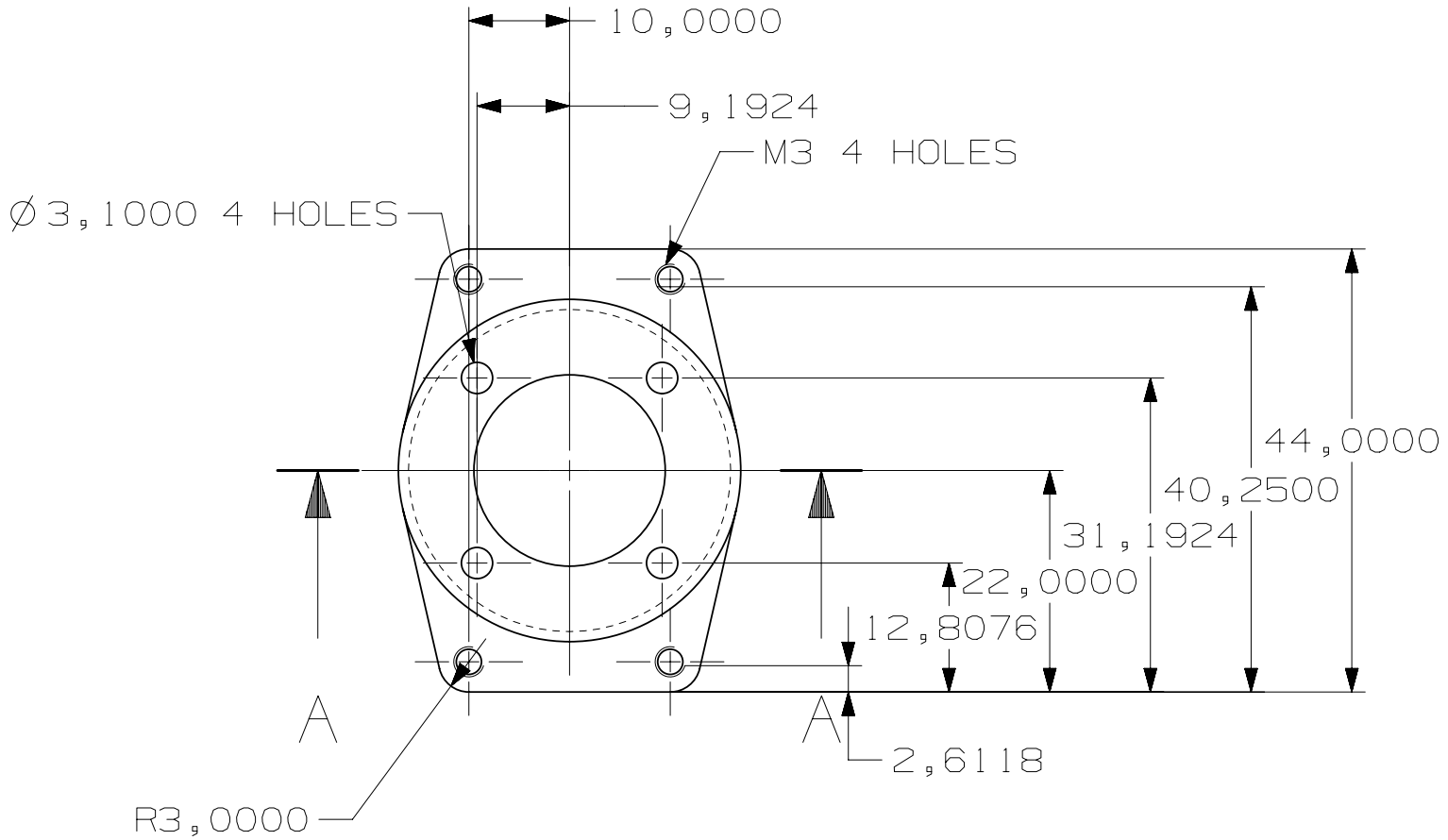

SECTION A - A
Ø37,00
Ø10,00
Ø3,50
A
A

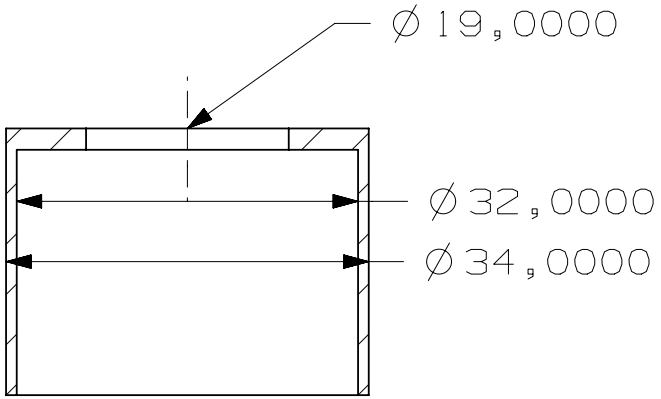

50,00
45,00
40,00
10,00
5,00
R2,00
Ø29,00

MOTOR BULKHEAD



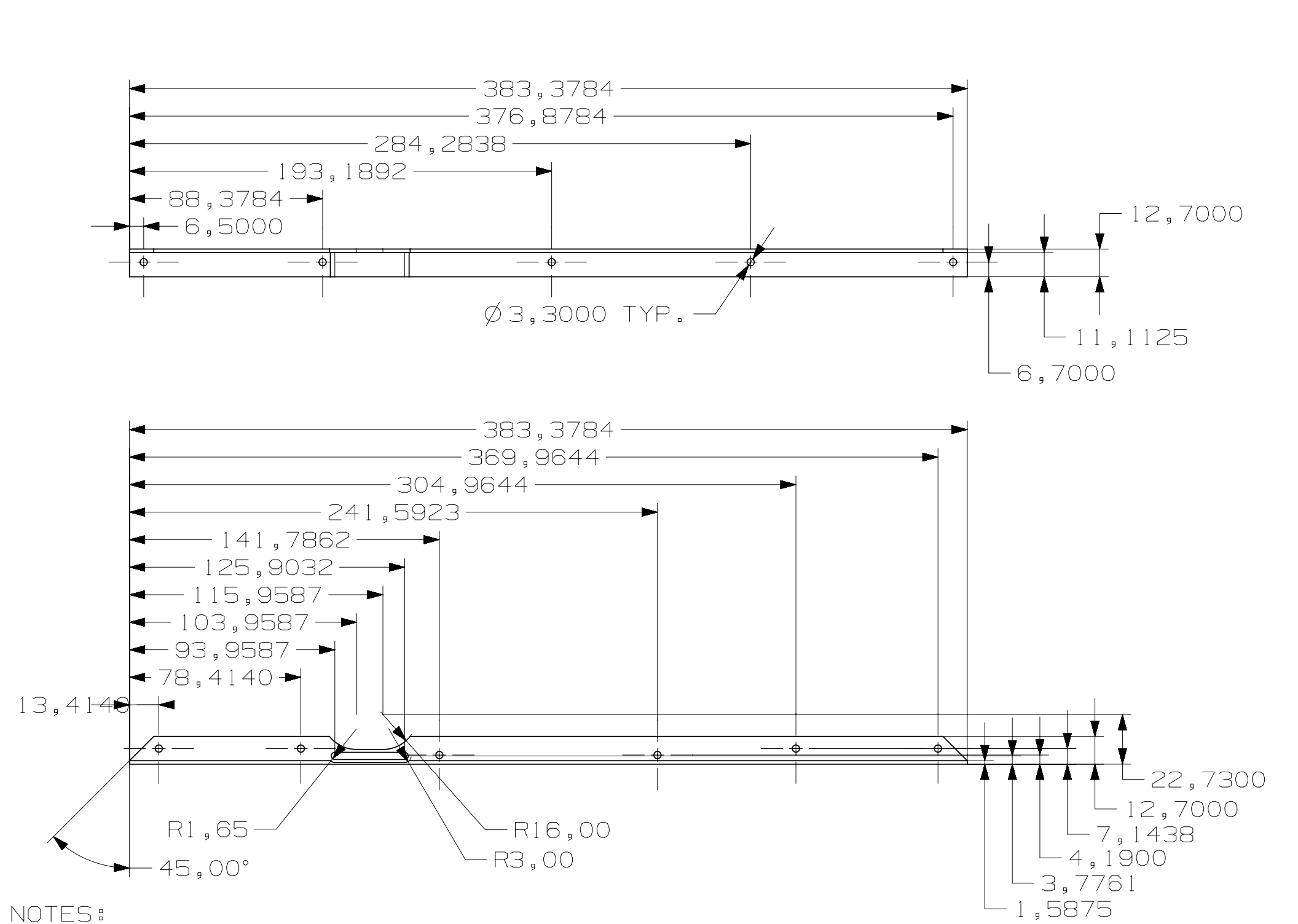
NOTE:
1. ALL DIMENSIONS IN MILLIMETERS
2. MATERIAL IS CARBON FIBER COMPOSITE CONSISTING OF 15 LAYERS USING A $[0/90/\pm45]_s$ QUASI-ISOTROPIC LAYUP



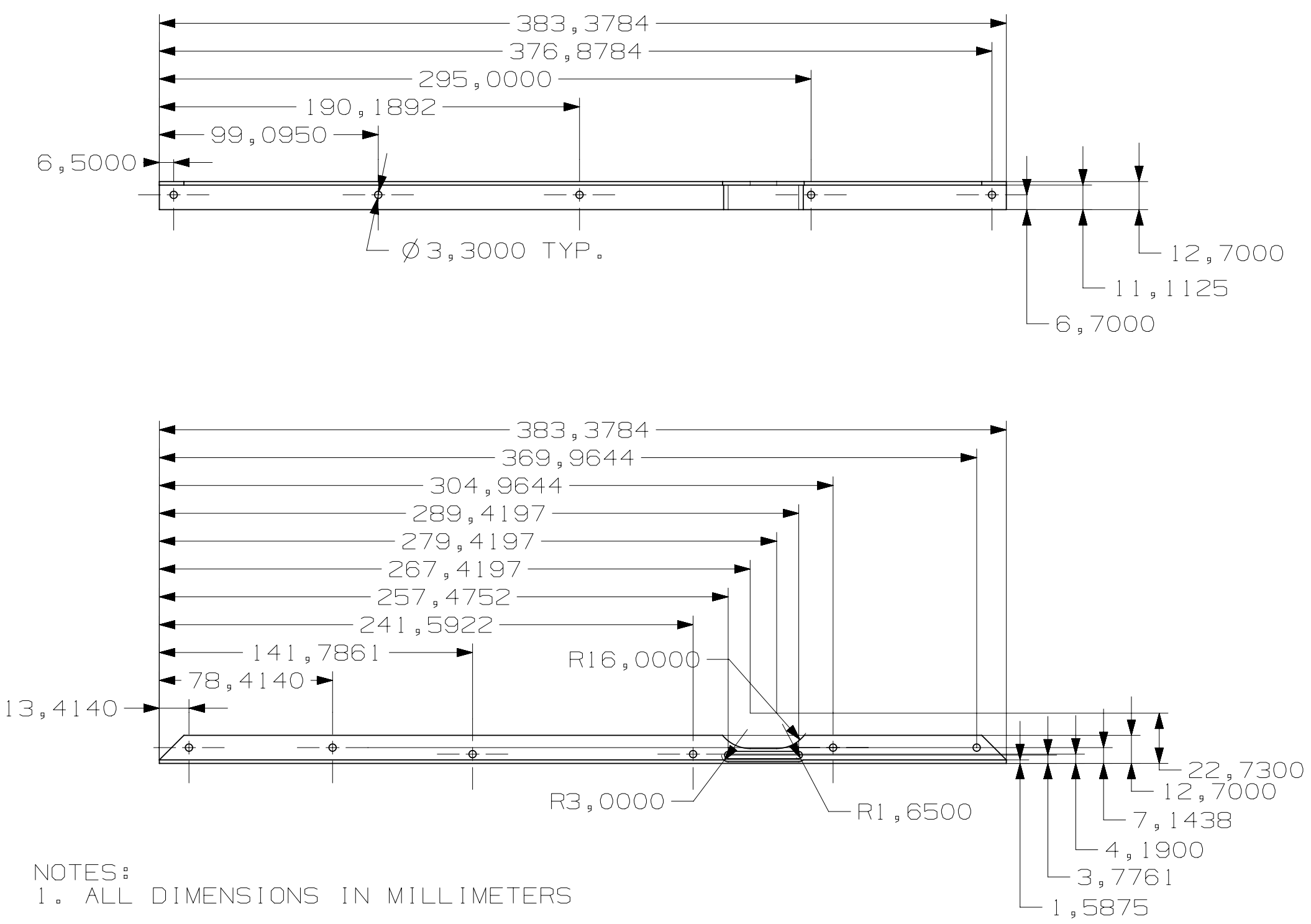


SECTION A-A

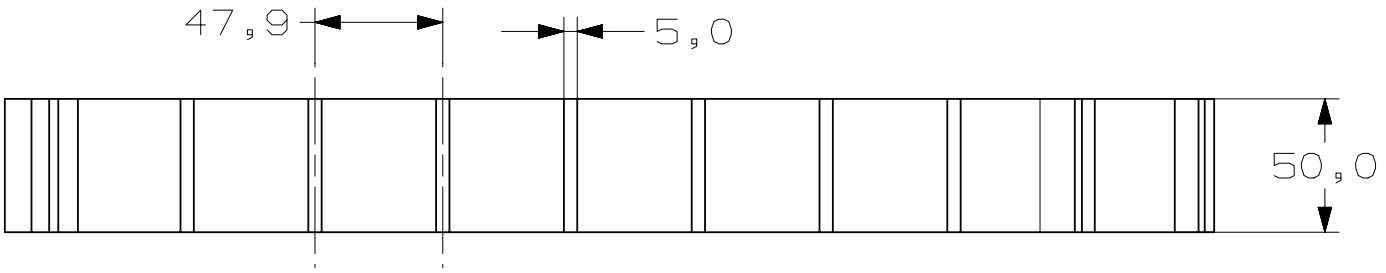
NOTES:
1. ALL DIMENSIONS IN MILLIMETERS
2. MATERIAL ALUMINUM

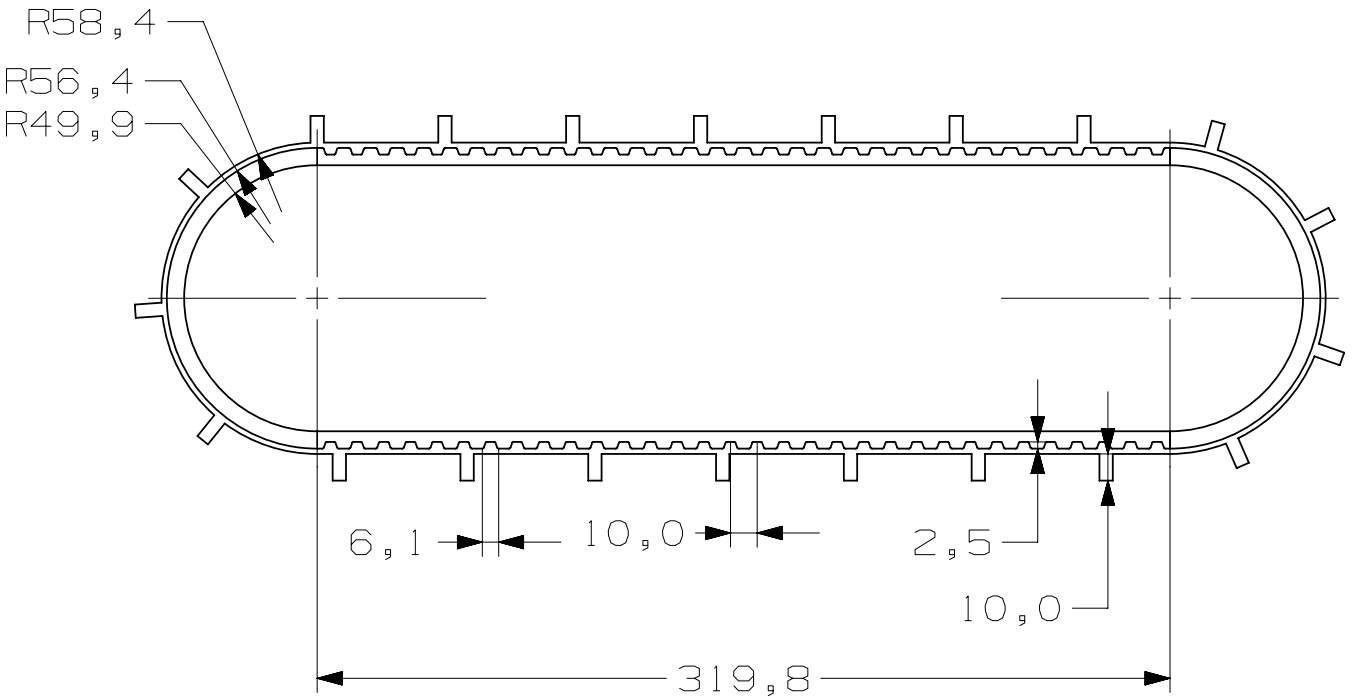


NOTES:
1. ALL DIMENSIONS IN MILLIMETERS



NOTES:
1. ALL DIMENSIONS IN MILLIMETERS





NOTE:
1. ALL DIMENSIONS IN MILLIMETERS

Appendix B

Summary of tracks for use on small tracked robots

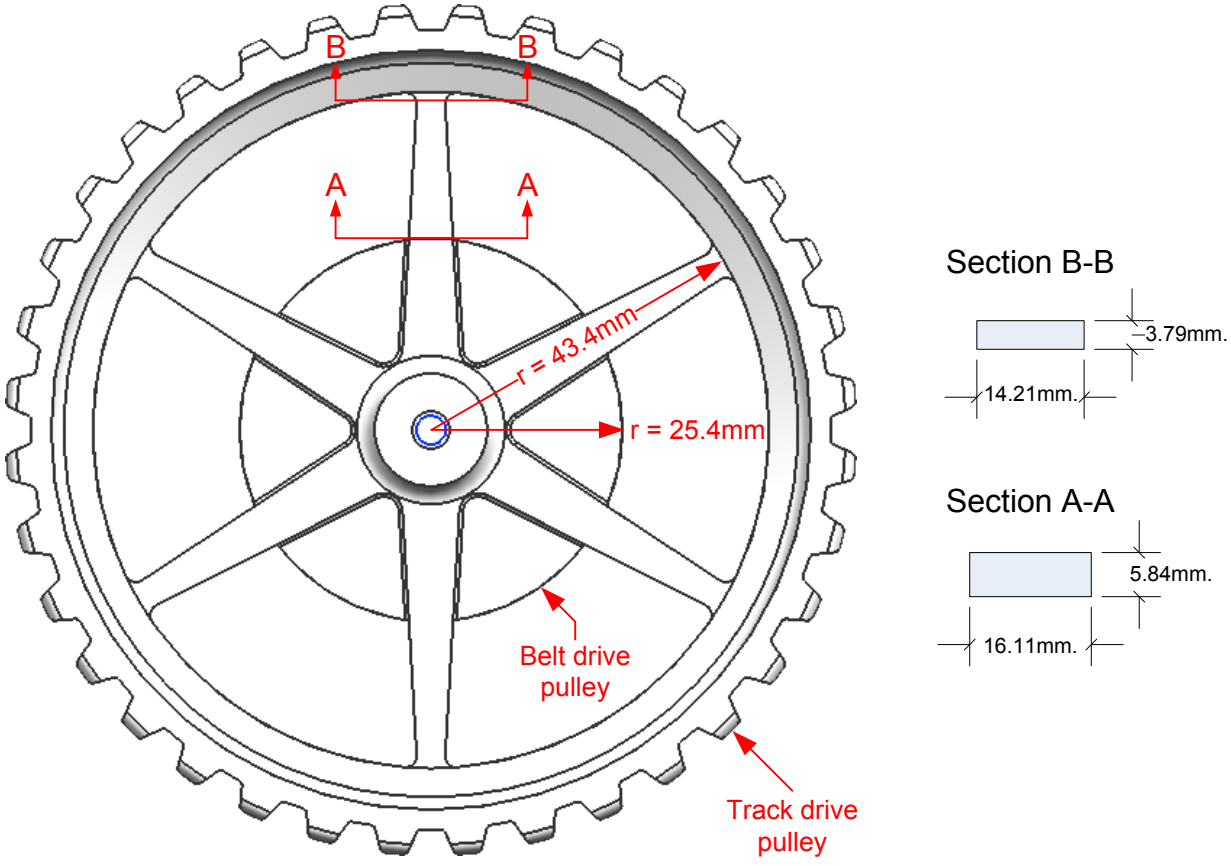
The following is a list of some of the treads available for use on a small tracked robotic vehicle. Many of these treads are better suited for hobbyist applications and don't appear to be able to stand up to a rugged, outdoor environment. The last tread on the list appears to be more suited for outdoor applications, however, the lack of internal cleats installed by the manufacturer is a drawback.

1. Lynxmotion treads (www.lynxmotion.com)
   (a) 2" wide tracks
   (b) Made of polypropylene links and rubber (shore A hardness = 45)
   (c) 21 links creates ca. 23" of track
   (d) Sprockets available in 2.75" and 3.75" diameter (metal sprockets)
2. Vex Robotic tank tread kit
   (a) 1.5" wide tracks
   (b) 170 links creates ca. 65.5" of track
   (c) Kit comes with drive sprockets and bogey wheels
   (d) Kit is all plastic
3. Tamiya (www.robotshop.ca)
   (a) Not much info available. Kit includes tracks, idler wheels, and drive wheels.
4. Boe-Bot tank treads (www.parallax.com)
   (a) Rubber treads
   (b) Dimensions not given
   (c) Drive sprockets and idler wheels are plastic

5. All terrain reinforced 2" wide tread (www.superdroidrobots.com)
   (a) 2" wide tracks
   (b) Rubber w/ nylon reinforcement
   (c) Available in roughly any length (24" minimum)
   (d) No internal cleats on treads. These must be added by consumer.

Appendix C

Pulley Stress Analysis



Stress analysis was done at the belt drive pulley/track drive pulley interface in order to ensure the strength of the track drive pulley spokes. The shear strength at two sections along the pulley was analyzed. The figure above provides a diagram of the pulley with the location of the two sections and the dimensions of each section.

Calculation of applied torque

The motors used on the robot have a stall torque of $T_{stall} = 1020\,\text{mNm}$. This torque will be used as the maximum possible torque at the motor. Using this value the torque at the drive pulley can be calculated:

Torque at the gearhead:
The torque at the gearhead output shaft can be calculated using the following equation:

$$T_{gh} = T_{stall} N_{gh} \eta_{gh}$$

Substituting in the gear ratio (N_{gh}) and the efficiency (η_{gh}) values obtained from the gearhead data sheet gives the following:

$$61.4\,\text{Nm} = (1.02\,\text{Nm})(86)(0.70)$$

However, the maximum permissible torque at the gearhead shaft according to the data sheet is $7.5\,Nm$, therefore, this will be the limiting torque, not the motor stall torque.

Torque at the drive pulley:
The torque at the drive pulley can be calculated using a similar equation to the gearhead torque equation:

$$T_p = T_{gh} N_b \eta_b$$

Substituting in the belt-drive ratio (N_b), the belt-drive efficiency (η_b), assumed to be 95%, and the limiting torque of the gearhead, we arrive at:

$$15\,\text{Nm} = (61.4\,\text{Nm})(2)(1.00)$$

The drive pulley will experience a maximum torque of $T_{max} = 15\,\text{Nm}$.

Shear Stress Analysis: Section A-A The maximum force experienced by all of the spokes due to the applied torque at section A-A (the interface between the belt drive pulley and the track drive pulley) can be calculated using the moment arm at section A-A:

$$F_{max} = T_{max}/r$$

Substituting in the values from the diagram and the torque as calculated above we arrive at $F_{max} = 590.6\,\text{N}$. This force will be distributed across the six spokes of the track drive pulley, resulting in an applied force at each spoke of $F_{spoke} = 98.4\,\text{N}$. The shear stress is determined by:

$$\tau = \frac{F}{A}$$

The resulting shear stress at each spoke is:

$$\begin{aligned} \tau &= \frac{F}{A} \\ &= \frac{98.4\,N}{0.01611m\;0.00584m} \\ &= 1.05 \times 10^6\,\text{Pa} \\ \tau &= 1.05\,\text{MPa} \end{aligned} \tag{C.1}$$

Using the shear strength of the rapid prototype polycarbonate material, we can determine the safety factor of the spoke at this section. Although the shear strength of the material is not given directly, it is estimated to be 1/2 the tensile strength of the material. Testing of the material can provide a more accurate value. The shear strength of the material is estimated to be $\tau_{PC} = 26\,\text{MPa}$. This provides a safety factor of $SF = 24.9$.

Although the calculations are not shown here, if the limiting torque of the gearhead were to be ignored, the shear stress in each spoke at section A-A would be $\tau = 8.56\,\text{MPa}$, providing a safety factor of $SF = 3.03$.

Shear Stress Analysis: Section B-B

The maximum force experienced by all of the spokes due to the applied torque at section A-A (the interface between the belt drive pulley and the track drive pulley) can be calculated using the moment arm at section A-A:

$$F_{max} = T_{max}/r$$

Substituting in the values from the diagram and the torque as calculated above we arrive at $F_{max} = 345.7\,\text{N}$. This force will be distributed across the six spokes of the track drive pulley, resulting in an applied force at each spoke of $F_{spoke} = 57.6\,\text{N}$. The shear stress is determined by:

$$\tau = \frac{F}{A}$$

The resulting shear stress at each spoke is:

$$\begin{aligned} \tau &= \frac{F}{A} \\ &= \frac{57.6\,N}{0.01421m\;0.00379m} \\ &= 1.06 \times 10^6\,\text{Pa} \\ \tau &= 1.06\,\text{MPa} \end{aligned} \tag{C.2}$$

Using the shear strength of the rapid prototype polycarbonate material as calculated above, we can determine the safety factor of the spoke at this section. The safety factor at this section is $SF = 24.5$.

Again, the calculations are not shown here, but with the limiting torque of the gearhead ignored, the shear stress in each spoke at section B-B would be $\tau = 8.84\,\text{MPa}$, providing a safety factor of $SF = 2.9$.

Appendix D

Motor Selection Calculations and Validation

This appendix outlines the motor selection calculations in detail and provides a set of calculations validating that the motor's performance meets or exceeds the requirements of the ground robot design specifications. The robot design criteria and robot parameters that were used to size the motors are presented in the table below.

Robot Design Specs & Parameters	
Robot Mass	8 kg
Track Diameter	136.75 mm
Speed	0.25 $^{m}/_{s}$
Traversable Slope	45°

The force (F) required to propel the robot up the desired slope is calculated by Equation D.1:

$$F = W(\sin\theta + \mu_R) \tag{D.1}$$

where θ is the slope of the terrain (45° in this case), W is the weight of the robot, and μ_R is the friction of the system. Equation D.1 makes the assumption that the robot is traveling on a rigid surface and that no track slippage will occur. The calculated force (F) and the desired velocity (v) are used to calculate the power (P) required to propel the robot up the slope at the desired velocity (Equation D.2).

$$P = Fv \tag{D.2}$$

The angular velocity (ω) at which the motor must operate in order to drive the robot at the desired velocity is a simple calculation provided by Equation D.3, which can be converted

to RPM for use in selecting a motor.

$$\omega = \frac{v}{r} \tag{D.3}$$

Finally, the required torque can be determined from Equation D.4.

$$T = \frac{P}{\omega} \tag{D.4}$$

The torque calculated by Equation D.4 is the total torque required to drive the robot.

The following calculations are provided to ensure proper motor sizing:

$$\begin{array}{l|rcl} F = mg(\sin\theta + \mu) & F & = & (8\,\text{kg})(9.81\,{}^{\text{m}}/_{\text{s}^2})(\sin(45^\circ) + 0.15) \\ & & = & 67.2\ \text{N} \\ P = Fv & P & = & (67.2\,\text{N})(0.25\,{}^{\text{m}}/_{\text{s}}) \\ & & = & 16.816\,\text{W} \\ \omega = v/r & \omega & = & (0.25\,{}^{\text{m}}/_{\text{s}})/(0.0684\,\text{m}) \\ & & = & 3.656\,{}^{\text{rad}}/_{\text{s}} \\ & & = & 34.9\,\text{rpm} \\ T = P/\omega & T & = & (16.816\,\text{W})/(3.656\,{}^{\text{rad}}/_{\text{s}}) \\ & & = & 4.598\,\text{Nm} \end{array}$$

After determining the required angular velocity and torque to satisfy the design parameters, it was decided that a belt drive with a 2:1 ratio would be used to connect the motor to the drive pulley. In addition, the calculated torque $T = 4.598\,\text{Nm}$ is the total torque required to meet the design specifications. Each drive motor could provide only half the required torque in order to meet the design requirements, but the total calculated torque was used to size a single motor in order to provide a safety factor of two. Various motors were examined and with the help of Maxon Motor's motor selection program a motor was selected.

The motor selected for the robot drive train is the:

Maxon RE30 24V brushed DC motor (part no. 310007)

This motor is paired with a planetary gearhead in order to provide the required speed reduction and the required torque to the drive pulley. The selected gearhead is the:

Maxon GP32C Planetary gearhead with an 86:1 ratio (part no. 166942)

The "C" in GP32C stands for ceramic. This motor is constructed using ceramic shafts for the planetary gears which allow for increased loading. The motor and gearhead data sheets are provided on pages 136 and 137 respectively.

The following calculations are to validate that the selected motors meet the design requirements:
The motors are designed to run at 24v. Two 3-cell lithium-polymer batteries connected in series were chosen as the motor power supply. These batteries provide 22.2v nominal with an actual operating range around 24v. A summary of the motor and gearhead parameters that will be required to validate the motor performance are provided in the table below:

Motor Parameters		**Gearhead Parameters**	
Nominal Voltage	24v	Ratio	86:1
Speed/Torque Constant	8.69 $^{rpm}/_{mNm}$	Efficiency	70%
No load speed	8810 rpm	**Belt drive Parameters**	
Torque constant	25.9 $^{mNm}/_{A}$	Ratio	2:1
Speed constant	369 $^{rpm}/_{V}$	Efficiency	90%

Torque required per drive pulley:	2.3 Nm
Torque required at gearhead:	2.3/(2*0.9) = 1.28 Nm
Torque required at motor:	1.28/(86*.07) = 0.0213 Nm

Using the torque required at the motor, the speed loss due to torque can be calculated.

Speed/Torque gradient:	8.69 $^{rpm}/_{mNm}$
Speed loss	= 8.69 $^{rpm}/_{mNm}$ * 21.33 mNm
	= 185.4 rpm
No load speed (@ 24v):	8810 rpm
Speed under load	= 8810 rpm - 185.4 rpm
	= 8624 rpm

Now we validate that the speed at the motor under load meets the design specification for robot speed.

Speed available at motor:	8624 rpm
Speed available at gearhead:	(8624 rpm)/(86) = 100.78 rpm
Speed available at drive pulley:	(100.78 rpm)/2 = 50.39 rpm

The available rotational speed available at the drive pulley is 50.39 rpm. This value well exceeds the 34.9 rpm rotational speed requirement as calculated earlier. This means that the robot can travel at higher speeds, or the motors can be supplied with less voltage in order to run at the 34.9 rpm rotational speeds established. The robot testing presented in section 3.3 further validates the proper selection of drive train components.

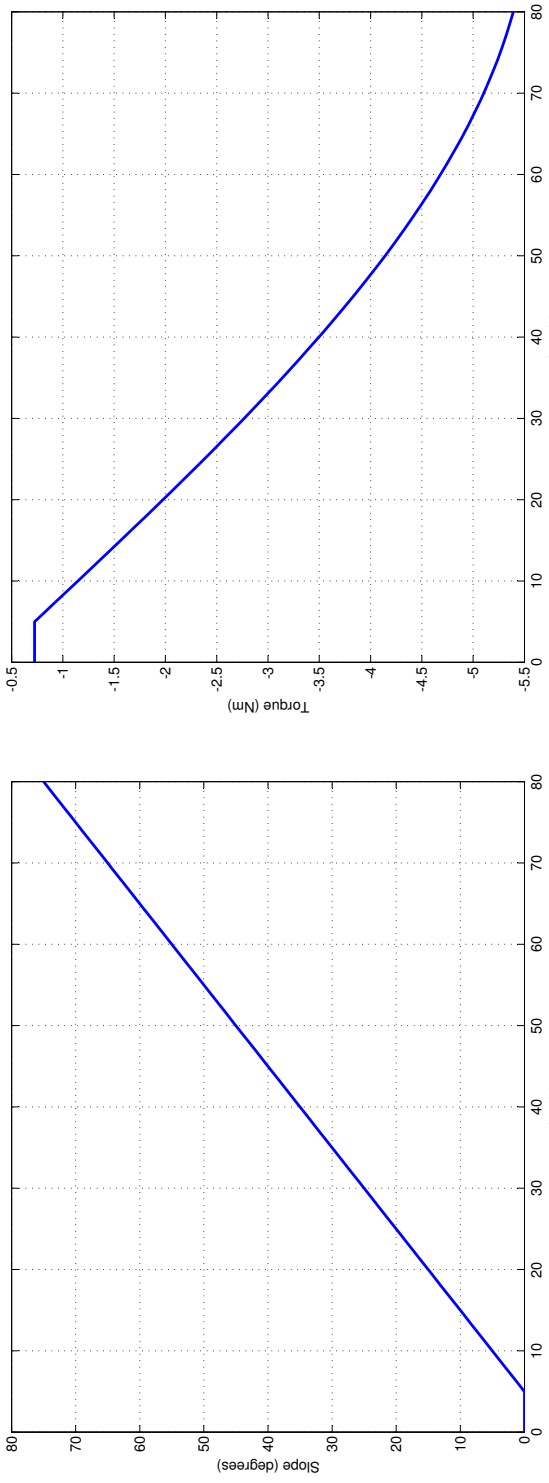
Although the rotational speed of the motors varies linearly as a function of torque, the above analysis can be used to develop a velocity curve for the motors as a function of slope. This curve is provided by:

$$v(\theta) = \frac{\pi r \left(\kappa - \frac{\omega_{nl} g m r (\sin\theta + \mu\cos\theta)}{N_b \eta_b N_{gh} \eta_{gh}}\right)}{30 N_b N_{gh}} \tag{D.5}$$

Nomenclature:

κ	=	Speed/Torque gradient
N_b	=	ratio of belt drive
η_b	=	efficiency of belt drive
N_{gh}	=	ratio of gearhead
η_{gh}	=	efficiency of gearhead
ω_{nl}	=	No load speed of motor

To further validate the motor selection, a dynamic model of the motors was built using the SimScape toolbox as part of Simulink®. This model incorporates both the motor dynamics and some of the robot dynamics, including pulley inertias and the ability to apply various loads to the pulleys to simulate slope traversal. The Simulink® motor model is presented on page 135. The following plots depict the outputs of the model due to a ramp input of the slope. The ramp input begins at $t = 5s$ with a value of 0° after which the slope angle changes at 1°/s. From Figure D.1 we can see that assuming no-slip conditions, the robot would be capable of traversing a slope of up to 75° with only a small loss in velocity.



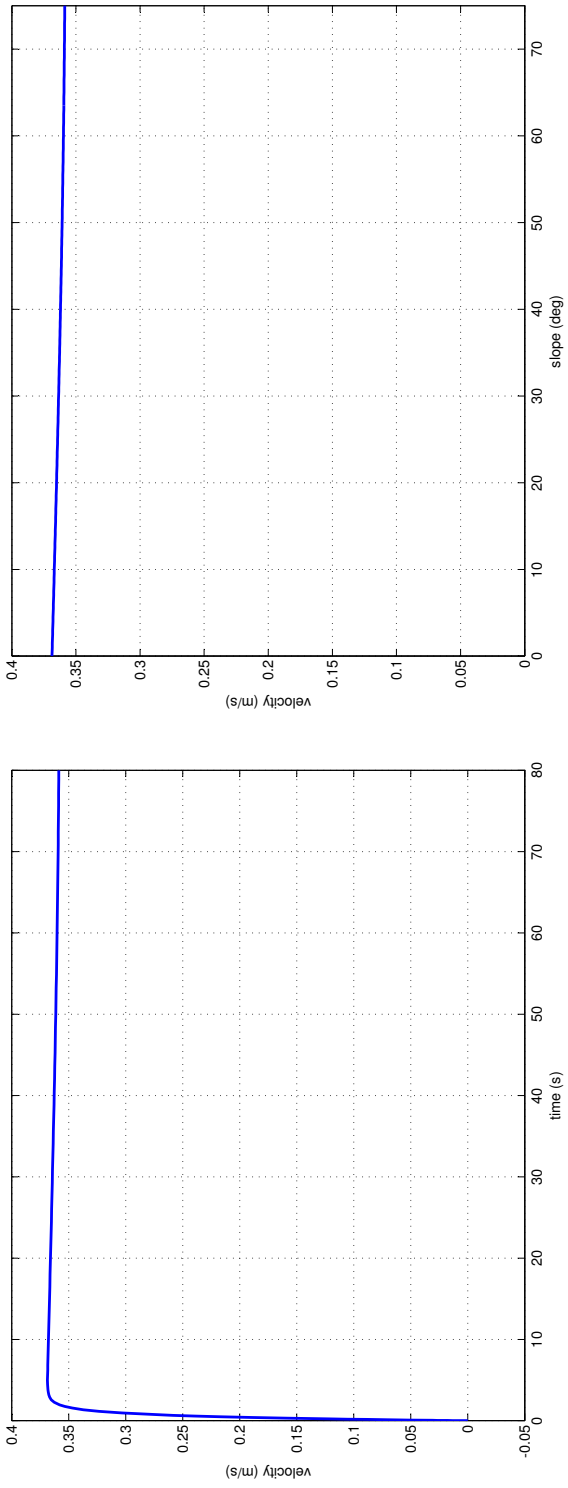
(a) The ramp input provided to the motor model

(b) The torque applied to the robot track as a result of the ramp input

(c) The angular speed of the motor as a result of the ramp input

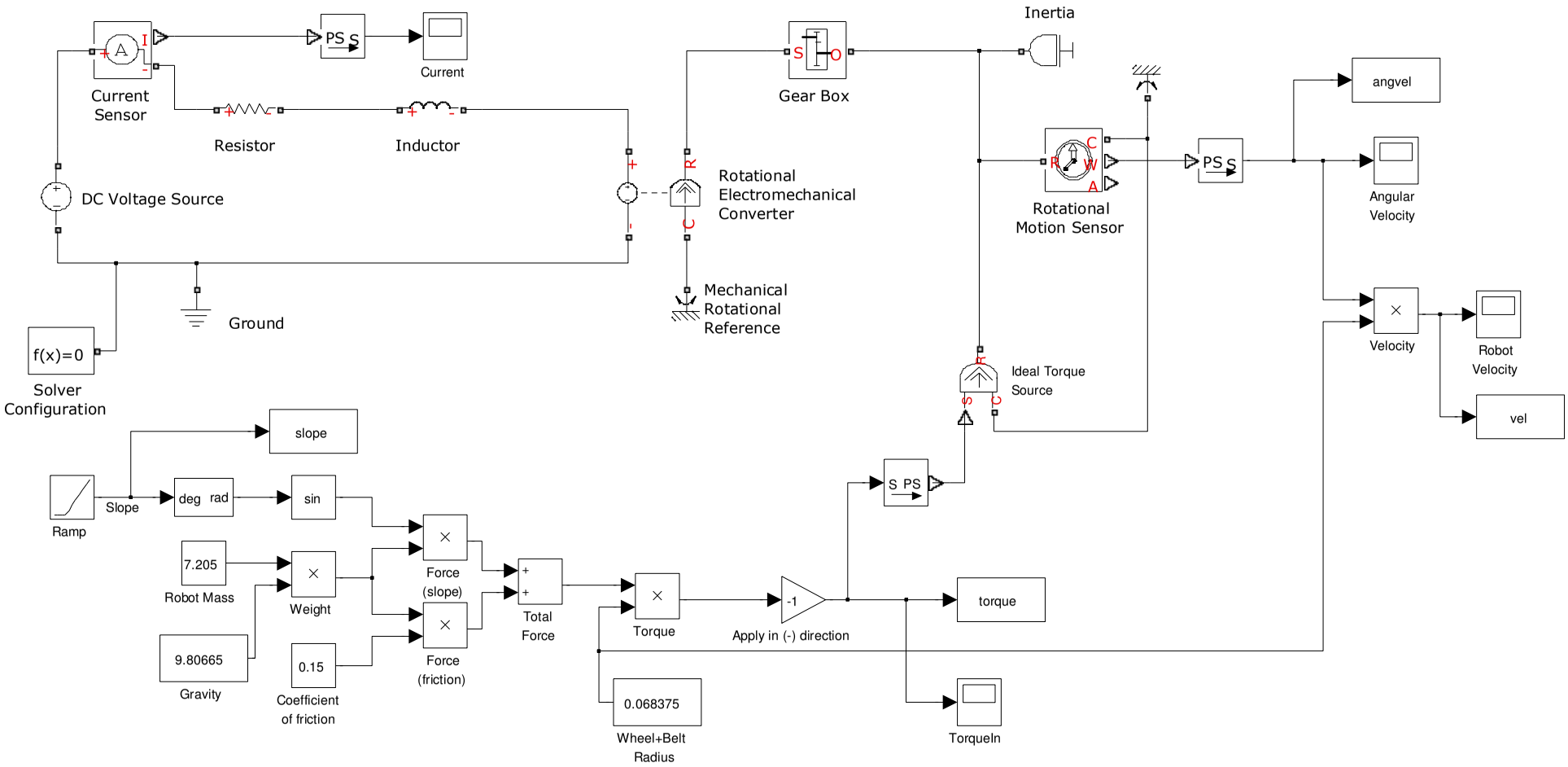
(d) The current drawn by the motor as a result of the ramp input

Figure D.1: A series of plots produced using the output of the Simulink motor model

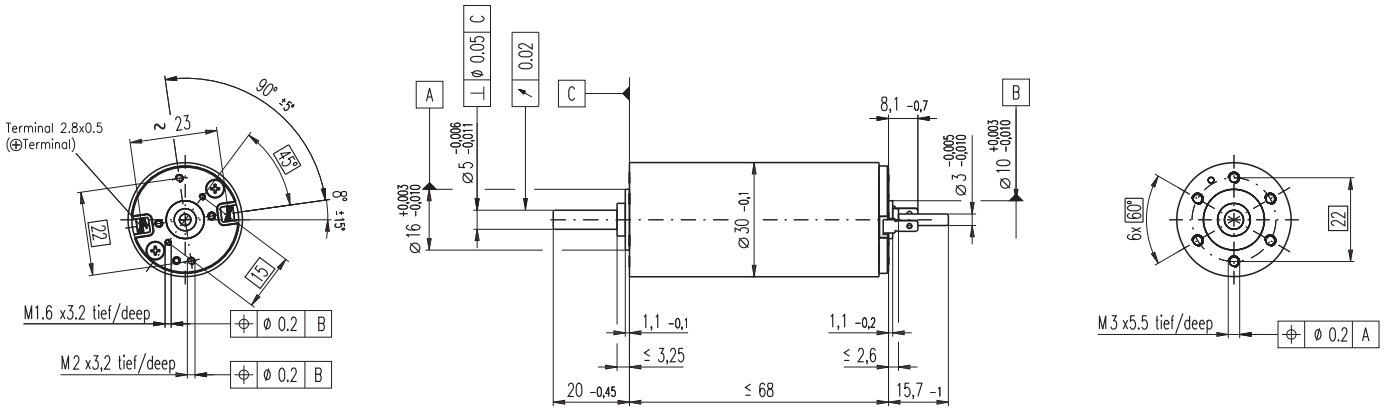


(e) The velocity of the robot as a result of the ramp input (f) The velocity of the robot as a function of the terrain slope

Figure D.1: A series of plots produced using the output of the Simulink motor model


Current Sensor
Current
Resistor
Inductor
DC Voltage Source
Ground
Rotational Electromechanical Converter
Mechanical Rotational Reference
Gear Box
Inertia
Rotational Motion Sensor
angvel
Angular Velocity
Velocity
Robot Velocity
vel
Ideal Torque Source
f(x)=0
Solver Configuration
slope
Ramp
Slope
deg rad
sin
7.205
Robot Mass
Weight
9.80665
Gravity
0.15
Coefficient of friction
Force (slope)
Force (friction)
Total Force
Torque
Apply in (-) direction
-1
torque
TorqueIn
0.068375
Wheel+Belt Radius

RE 30 ∅30 mm, Graphite Brushes, 60 Watt



M 1:2

Stock program
Standard program
Special program (on request)

			Order Number							
	according to dimensional drawing		310005	310006	310007	310008	310009			
	shaft length 15.7 shortened to 8.7 mm		268193	268213	~~268214~~	268215	268216			
Motor Data										
	Values at nominal voltage									
1	Nominal voltage	V	12.0	18.0	24.0	36.0	48.0			
2	No load speed	rpm	8170	8590	8810	8590	8490			
3	No load current	mA	300	212	164	106	78.5			
4	Nominal speed	rpm	7630	7900	8050	7810	7750			
5	Nominal torque (max. continuous torque)	mNm	51.7	75.5	85.0	83.4	88.2			
6	Nominal current (max. continuous current)	A	4.00	4.00	3.44	2.20	1.72			
7	Stall torque	mNm	844	991	1020	936	1020			
8	Starting current	A	60.5	49.8	39.3	23.5	19.0			
9	Max. efficiency	%	86	87	87	87	88			
	Characteristics									
10	Terminal resistance	Ω	0.198	0.362	0.611	1.53	2.52			
11	Terminal inductance	mH	0.0345	0.0703	0.119	0.281	0.513			
12	Torque constant	mNm / A	13.9	19.9	25.9	39.8	53.8			
13	Speed constant	rpm / V	685	479	369	240	178			
14	Speed / torque gradient	rpm / mNm	9.74	8.71	8.69	9.22	8.33			
15	Mechanical time constant	ms	3.42	3.25	3.03	3.17	3.01			
16	Rotor inertia	gcm²	33.5	35.7	33.3	32.9	34.5			

Specifications

Thermal data

17	Thermal resistance housing-ambient	6.0 K / W
18	Thermal resistance winding-housing	1.7 K / W
19	Thermal time constant winding	16.2 s
20	Thermal time constant motor	714 s
21	Ambient temperature	-30 ... +100°C
22	Max. permissible winding temperature	+125°C

Mechanical data (ball bearings)

23	Max. permissible speed	12000 rpm
24	Axial play	0.05 - 0.15 mm
25	Radial play	0.025 mm
26	Max. axial load (dynamic)	5.6 N
27	Max. force for press fits (static)	110 N
	(static, shaft supported)	1200 N
28	Max. radial loading, 5 mm from flange	28 N

Other specifications

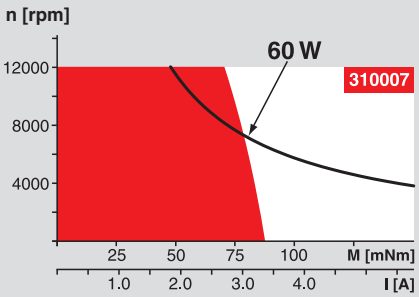
29	Number of pole pairs	1
30	Number of commutator segments	13
31	Weight of motor	238 g

Values listed in the table are nominal.
Explanation of the figures on page 49.

⚠ Tolerances may vary from the standard specification.

Option
Preloaded ball bearings

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

Assigned power rating

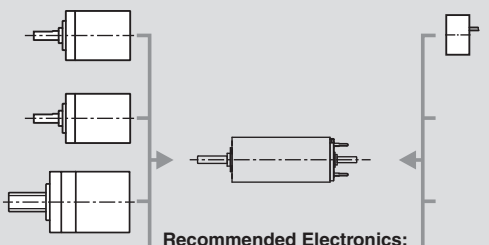
maxon Modular System

Overview on page 16 - 21

Planetary Gearhead
∅32 mm
0.75 - 6.0 Nm
Page 232 / 234 / 235

Koaxdrive
∅32 mm
1.0 - 4.5 Nm
Page 238

Spindle Drive
∅32 mm
Page 251 / 252 / 253



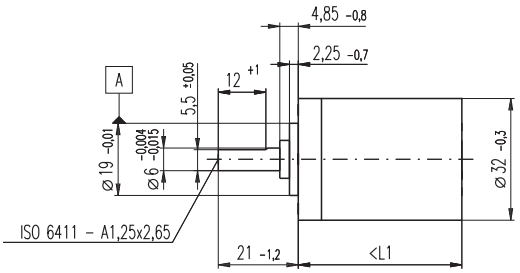
Encoder MR
256 - 1024 CPT
3 channels
Page 265

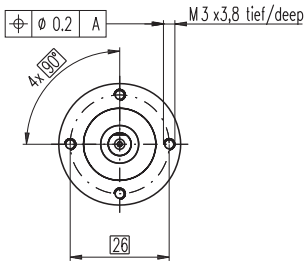
Recommended Electronics:

ADS 50/5	Page 282
ADS_E 50/5	283
EPOS2 Module 36/2	302
EPOS2 24/5	303
EPOS2 50/5	303
EPOS P 24/5	306
Notes	**18**

Planetary Gearhead GP 32 C ∅32 mm, 1.0 - 6.0 Nm

Ceramic Version





M 1:2

Technical Data

Planetary Gearhead	straight teeth
Output shaft	stainless steel
Shaft diameter as option	8 mm
Bearing at output	ball bearing
Radial play, 5 mm from flange	max. 0.14 mm
Axial play	max. 0.4 mm
Max. radial load, 10 mm from flange	140 N
Max. permissible axial load	120 N
Max. permissible force for press fits	120 N
Sense of rotation, drive to output	=
Recommended input speed	< 8000 rpm
Recommended temperature range	-20 ... +100°C
Extended area as option	-35 ... +100°C

Option: Low-noise version

Stock program
Standard program
Special program (on request)

Gearhead Data		Order Number											
		166930	166933	166938	166939	166944	166949	166954	166959	166962	166967	166972	166977
1 Reduction		3.7 : 1	14 : 1	33 : 1	51 : 1	111 : 1	246 : 1	492 : 1	762 : 1	1181 : 1	1972 : 1	2829 : 1	4380 : 1
2 Reduction absolute		$^{26}/_{7}$	$^{676}/_{49}$	$^{529}/_{16}$	$^{17576}/_{343}$	$^{13824}/_{125}$	$^{421824}/_{1715}$	$^{86112}/_{175}$	$^{19044}/_{25}$	$^{10123776}/_{8575}$	$^{8626176}/_{4375}$	$^{495144}/_{175}$	$^{109503}/_{25}$
3 Max. motor shaft diameter	mm	6	6	3	6	4	4	3	3	4	4	3	3
Order Number		**166931**	**166934**		**166940**	166945	166950	**166955**	166960	166963	166968	166973	166978
1 Reduction		4.8 : 1	18 : 1		66 : 1	123 : 1	295 : 1	531 : 1	913 : 1	1414 : 1	2189 : 1	3052 : 1	5247 : 1
2 Reduction absolute		$^{24}/_{5}$	$^{624}/_{35}$		$^{16224}/_{245}$	$^{6877}/_{56}$	$^{101062}/_{343}$	$^{331776}/_{625}$	$^{36501}/_{40}$	$^{2425488}/_{1715}$	$^{536406}/_{245}$	$^{1907712}/_{625}$	$^{839523}/_{160}$
3 Max. motor shaft diameter	mm	4	4		4	3	3	4	3	3	3	3	3
Order Number		166932	166935		166941	166946	**166951**	166956	166961	**166964**	166969	166974	166979
1 Reduction		5.8 : 1	21 : 1		79 : 1	132 : 1	318 : 1	589 : 1	1093 : 1	1526 : 1	2362 : 1	3389 : 1	6285 : 1
2 Reduction absolute		$^{23}/_{4}$	$^{299}/_{14}$		$^{3887}/_{49}$	$^{3312}/_{25}$	$^{389376}/_{1225}$	$^{20631}/_{35}$	$^{279841}/_{256}$	$^{9345024}/_{6125}$	$^{2066688}/_{875}$	$^{474513}/_{140}$	$^{6436343}/_{1024}$
3 Max. motor shaft diameter	mm	3	3		3	3	4	3	3	4	3	3	3
Order Number			**166936**		**166942**	166947	**166952**	166957		166965	**166970**	166975	
1 Reduction			23 : 1		86 : 1	159 : 1	411 : 1	636 : 1		1694 : 1	2548 : 1	3656 : 1	
2 Reduction absolute			$^{576}/_{25}$		$^{14976}/_{175}$	$^{5887}/_{10}$	$^{359424}/_{875}$	$^{79488}/_{125}$		$^{1162213}/_{686}$	$^{7962624}/_{3125}$	$^{457056}/_{125}$	
3 Max. motor shaft diameter	mm		4		4	3	4	3		3	4	3	
Order Number			166937		166943	166948	166953	166958		166966	166971	166976	
1 Reduction			28 : 1		103 : 1	190 : 1	456 : 1	706 : 1		1828 : 1	2623 : 1	4060 : 1	
2 Reduction absolute			$^{138}/_{5}$		$^{3588}/_{35}$	$^{12167}/_{64}$	$^{89401}/_{196}$	$^{158171}/_{224}$		$^{2238912}/_{1225}$	$^{2056223}/_{784}$	$^{3637933}/_{896}$	
3 Max. motor shaft diameter	mm		3		3	3	3	3		3	3	3	
4 Number of stages		1	2	2	3	3	4	4	4	5	5	5	5
5 Max. continuous torque	Nm	1	3	3	6	6	6	6	6	6	6	6	6
6 Intermittently permissible torque at gear output	Nm	1.25	3.75	3.75	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5	7.5
7 Max. efficiency	%	80	75	75	70	70	60	60	60	50	50	50	50
8 Weight	g	118	162	162	194	194	226	226	226	258	258	258	258
9 Average backlash no load	°	0.7	0.8	0.8	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
10 Mass inertia	gcm²	1.5	0.8	0.8	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7
11 Gearhead length L1	mm	26.4	36.3	36.3	43.0	43.0	49.7	49.7	49.7	56.4	56.4	56.4	56.4

maxon Modular System

+ Motor	Page	+ Sensor / Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor / brake) + assembly parts											
RE 25, 10 W	77/79			81.0	90.9	90.9	97.6	97.6	104.3	104.3	104.3	111.0	111.0	111.0	111.0
RE 25, 10 W	77/79	MR	264	92.0	101.9	101.9	108.6	108.6	115.3	115.3	115.3	122.0	122.0	122.0	122.0
RE 25, 10 W	77/79	Enc 22	266	95.1	105.0	105.0	111.7	111.7	118.4	118.4	118.4	125.1	125.1	125.1	125.1
RE 25, 10 W	77/79	HED_ 5540	268/270	101.8	111.7	111.7	118.4	118.4	125.1	125.1	125.1	131.8	131.8	131.8	131.8
RE 25, 10 W	77/79	DCT 22	277	103.3	113.2	113.2	119.9	119.9	126.6	126.6	126.6	133.3	133.3	133.3	133.3
RE 25, 20 W	78			69.5	79.4	79.4	86.1	86.1	92.8	92.8	92.8	99.5	99.5	99.5	99.5
RE 25, 20 W	78	MR	264	80.5	90.4	90.4	97.1	97.1	103.8	103.8	103.8	110.5	110.5	110.5	110.5
RE 25, 20 W	78	HED_ 5540	269/272	90.3	100.2	100.2	106.9	106.9	113.6	113.6	113.6	120.3	120.3	120.3	120.3
RE 25, 20 W	78	DCT22	277	91.8	101.7	101.7	108.4	108.4	115.1	115.1	115.1	121.8	121.8	121.8	121.8
RE 25, 20 W	78	AB 28	316	103.6	113.6	113.6	120.2	120.2	126.9	126.9	126.9	133.6	133.6	133.6	133.6
RE 25, 20 W	78	HED_ 5540 / AB 28	269/316	120.8	130.7	130.7	137.4	137.4	144.1	144.1	144.1	150.8	150.8	150.8	150.8
RE 25, 20 W	79	AB 28	316	115.1	125.0	125.0	131.7	131.7	138.4	138.4	138.4	145.1	145.1	145.1	145.1
RE 25, 20 W	79	HED_5540 / AB 28	316	132.2	142.1	142.1	148.8	148.8	155.5	155.5	155.5	162.2	162.2	162.2	162.2
RE 30, 60 W	80			94.5	104.4	104.4	111.1	111.1	117.8	117.8	117.8	124.5	124.5	124.5	124.5
RE 30, 60 W	80	MR	265	105.9	115.8	115.8	122.5	122.5	129.2	129.2	129.2	135.9	135.9	135.9	135.9
RE 35, 90 W	81			97.4	107.3	107.3	114.0	114.0	120.7	120.7	120.7	127.4	127.4	127.4	127.4
RE 35, 90 W	81	MR	265	108.8	118.7	118.7	125.4	125.4	132.1	132.1	132.1	138.8	138.8	138.8	138.8
RE 35, 90 W	81	HED_ 5540	268/270	118.4	128.3	128.3	135.0	135.0	141.7	141.7	141.7	148.4	148.4	148.4	148.4
RE 35, 90 W	81	DCT 22	277	115.5	125.4	125.4	132.1	132.1	138.8	138.8	138.8	145.5	145.5	145.5	145.5
RE 35, 90 W	81	AB 28	316	133.5	143.4	143.4	150.1	150.1	156.8	156.8	156.8	163.5	163.5	163.5	163.5
RE 35, 90 W	81	HEDS 5540 / AB 28	268/316	150.6	160.5	160.5	167.2	167.2	173.9	173.9	173.9	180.6	180.6	180.6	180.6
A-max 26	105-112			71.2	81.1	81.1	87.8	87.8	94.5	94.5	94.5	101.2	101.2	101.2	101.2
A-max 26	106-112	MEnc 13	275	78.3	88.2	88.2	94.9	94.9	101.6	101.6	101.6	108.3	108.3	108.3	108.3
A-max 26	106-112	MR	264	80.0	89.9	89.9	96.6	96.6	103.3	103.3	103.3	110.0	110.0	110.0	110.0
A-max 26	106-112	Enc 22	267	85.6	95.5	95.5	102.2	102.2	108.9	108.9	108.9	115.6	115.6	115.6	115.6
A-max 26	106-112	HED_ 5540	169/170	90.0	99.9	99.9	106.6	106.6	113.3	113.3	113.3	120.0	120.0	120.0	120.0
A-max 32	113/115			89.4	99.3	99.3	106.0	106.0	112.7	112.7	112.7	119.4	119.4	119.4	119.4
A-max 32	114/116			88.0	97.9	97.9	104.6	104.6	111.3	111.3	111.3	118.0	118.0	118.0	118.0
A-max 32	114/116	MR	265	99.2	109.1	109.1	115.8	115.8	122.5	122.5	122.5	129.2	129.2	129.2	129.2
A-max 32	114/116	HED_ 5540	268/270	108.8	118.7	118.7	125.4	125.4	132.1	132.1	132.1	138.8	138.8	138.8	138.8

Appendix E

Path Planning Function Descriptions

Matlab Function	Description
`Boundary.m`	Because the robot orientation relies on terrain data other than the information contained in the current node, a boundary is defined that prevents the robot from getting too close to the edge of the map.
`costfunc.m`	Calculates the cost from n to n' as defined in section 4.2.2
`expand_node.m`	Expands the current node by finding all possible successor nodes and records the $g(n)$, $h(n)$, and $f(n)$ values along with other state information
`heuristic.m`	Calculates the heuristic from n to n_G as defined in section 4.2.3
`insert_open.m`	Facilitates the insertion of the data collected by `expand_node.m` into the $OPEN$ list
`min_fn.m`	Finds the node in the $OPEN$ list with the minimum $f(n)$ value
`node_index.m`	Finds the matrix indices corresponding to the node passed into the function
`OrientRobot.m`	Determines the roll and pitch angles of the robot along with the Z_{CG} at a given node. Also calculates the 3×3 rotation matrix required to plot the robot on the terrain
`PlaneFit.m`	The plane equation passed into `fmincon` (see Equation 4.11)
`RobotCoverage.m`	Contains the robot dimensions and returns this information to `OrientRobot.m`
`RobotonTerrain.m`	Contains the 3-D robot model and applies the appropriate coordinate transformations to plot the robot on the terrain
`ValidMove.m`	Determines if a potential successor to n is indeed a successor based on the roll and pitch limits of the robot

Appendix F

Institutional Review Board Approval Forms

The following pages contain:

- A letter from the Institutional Review Board granting permission to conduct a human subjects study
- A copy of the consent form provided to the volunteers of the human subjects study

The study was conducted between March 17, 2009 and March 24, 2009. The primary researcher conducting the human subject study trials was Amber Franklin, acting at the time as an undergraduate research assistant in the Unmanned Systems Lab. These Institutional Review Board approval documents support the research presented in chapter 5.

Office of Research Compliance
Institutional Review Board
2000 Kraft Drive, Suite 2000 (0497)
Blacksburg, Virginia 24061
540/231-4991 Fax 540/231-0959
e-mail moored@vt.edu
www.irb.vt.edu

FWA00000572(expires 1/20/2010)
IRB # is IRB00000667

DATE: November 18, 2008

MEMORANDUM

TO: Kevin Kochersberger
Amber Franklin
Dylan Klomparens

FROM: David M. Moore

Approval date: 11/18/2008
Continuing Review Due Date:11/3/2009
Expiration Date: 11/17/2009

SUBJECT: **IRB Expedited Approval:** "Evaluation of Stereovision for Supervisory Control of an Autonomous Helicopter Descent" , IRB # 08-712

This memo is regarding the above-mentioned protocol. The proposed research is eligible for expedited review according to the specifications authorized by 45 CFR 46.110 and 21 CFR 56.110. As Chair of the Virginia Tech Institutional Review Board, I have granted approval to the study for a period of 12 months, effective November 18, 2008.

As an investigator of human subjects, your responsibilities include the following:

1. Report promptly proposed changes in previously approved human subject research activities to the IRB, including changes to your study forms, procedures and investigators, regardless of how minor. The proposed changes must not be initiated without IRB review and approval, except where necessary to eliminate apparent immediate hazards to the subjects.
2. Report promptly to the IRB any injuries or other unanticipated or adverse events involving risks or harms to human research subjects or others.
3. Report promptly to the IRB of the study's closing (i.e., data collecting and data analysis complete at Virginia Tech). If the study is to continue past the expiration date (listed above), investigators must submit a request for continuing review prior to the continuing review due date (listed above). It is the researcher's responsibility to obtain re-approval from the IRB before the study's expiration date.
4. If re-approval is not obtained (unless the study has been reported to the IRB as closed) prior to the expiration date, all activities involving human subjects and data analysis must cease immediately, except where necessary to eliminate apparent immediate hazards to the subjects.

Important:
If you are conducting **federally funded non-exempt research**, please send the applicable OSP/grant proposal to the IRB office, once available. OSP funds may not be released until the IRB has compared and found consistent the proposal and related IRB applicaton.

cc: File

VIRGINIA POLYTECHINC INSTITUTE AND STATE UNIVERSITY

Informed Consent for Participants
in Research Projects Involving Human Subjects

Evaluation of a Ground Control Station Interface for Autonomous Helicopter Navigation

Investigators: Dr. Kevin Kochersberger, Amber Franklin

I. Purpose of this Research

The goal of this research is to evaluate two different displays for their effectiveness in simulated navigation of a UAV. Subjects will have a basic knowledge of the uses and operations of UAVs.

II. Procedures

Subjects will be given a brief explanation of the mission scenario as well as how to operate the ground station interface. Subjects will then attempt to use the specified ground station interface to navigate a simulated helicopter with reference to either a monovision nadir view or a visual display with a terrain data overlay. The research will take place in the Unmanned Systems Lab at Plantation Road. The performance metrics that will be recorded are time per command given, commands given, and total time.

III. Risks

The risks involved in this research will be similar to those of sitting at a computer for no more than 30 minutes. All efforts will be made to ensure operator comfort during the test.

IV. Benefits

The benefit of this research is in evaluating the different display types for UAV navigation in order to increase the operational range of current UAV technology.

V. Extent of Anonymity

All subjects will be identified in the study using only a number. No personal data will be recorded. Photographs or video may be taken, and those will be stored on a secure computer at the Unmanned Systems Lab. If any of these images are used in publications or reports, the participant will be asked for consent and not identified in the document.

VI. Compensation

Participants will not be compensated for this research.