Nittalion Battalion

Team Members: Richard Mattes, Anthony Cascone, Zack Correll, Jeremy Bridon, Wan Khalil, Matt Poremba, Adam Dean, Pramod Vemulapalli, Kshitij Jerath, Corey Adams, Gangyuan Jing

Table of Contents

Faculty Advisor Statement

I, Sean N. Brennan, certify that the design and development of Nittalion Battalion has been significant and that each student performing this work is a registered student. This work as part of a graduate class project and as an extracurricular project and represents a participation level equivalent to what would be awarded credit as a senior design project.

Sean N. Brennan, Department of Mechanical Engineering, Pennsylvania State University

Introduction

The Nittalion Battalion was Pennsylvania State University's entry to the Intelligent Ground Vehicle Competition (IGVC) last year, but it did not actively compete in the contest. This is the first time that the Nittalion Battalion will participate in any competition, IGVC or otherwise. The robot was designed as part of a graduate level Mechatronics course offered by the Department of Mechanical and Nuclear Engineering. The primary design objectives for the project were to develop a robust and capable platform that could compete in the IGVC competition and serve as a valuable research platform for vehicle dynamics and general robotics.

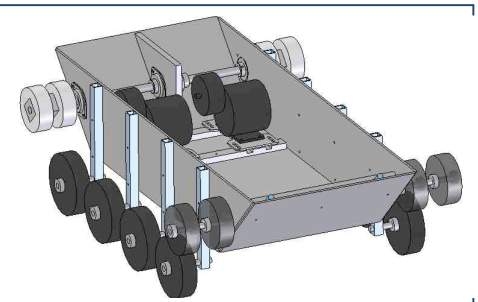


Figure 1: CAD drawing of the tank chassis

The past year saw many innovative additions and modifications to the robot. The upper deck was completely overhauled and rethought to improve modularity and accessibility of all components. Responsibility for development of the software was turned over to the Penn State Robotics Club. In the process, the software platform was moved from a MATLAB/XPC-based controller to a Player/Stage environment run on a laptop and Ampro PC/104 board. As a result, all software, controls, and algorithms were significantly altered. This report outlines the capabilities of the platform, the design process, and the software algorithms used.

Design Process

The original Nittalion Battalion design was based on the design requirements of the previous competition. The robot was designed with the ability to carry large payloads with ease. The robot's drive system was designed to handle 150 lbs of total weight, including the weight of all components, batteries, and the payload. Maneuverability requirements necessitated the decision to create a tank type robot (Figure 1) with differential drive. This type of construction enabled a zero turning radius that was advantageous in escaping from tight spaces with ease. Further, the construction eliminated complexity in path planning. Terrain traversal requirements were also considered and the robot was designed for harsh terrain, with specific focus on being able to climb a 10° incline on grass in wet conditions without slipping. Changes in terrain type were also accounted for, allowing the robot to operate on grass, concrete, sand, gravel etc. After the design process was complete and all design specifications met, the fabrication procedure was initiated. The base components were cut out of 1/8" aluminum using a water jet, and TIG-welded to create strong and weatherproof seams.

This year's team is organized into three major sections, viz. hardware abstraction and testing, control software and algorithm development, and platform development. The team consists primarily of members of the Penn State Robotics Club, and graduate students working in robotics-related research. The team architecture summary is provided in Table 1.

Table 1: Team Architecture Summary

Member Name	Department	Class	Responsibility
Dr. Sean Brennan	MNE	Assistant Professor	Faculty Advisor
Richard Mattes	ESM	Junior	Team Leader
Pramod Vemulapalli	MNE	Graduate	Lead Software Development
Anthony Cascone	MNE	Graduate	Hardware abstraction
Kshitij Jerath	MNE	Graduate	Algorithm Development
Zack Correll	EE	Junior	Vision Processing
Jeremy Bridon	CSE	Sophomore	Control Software
Matt Poremba	CSE	Sophomore	Control Software
Wan Khalil	MNE	Freshman	Control Software
Adam Dean	MNE	PhD candidate	Lead Platform Development
Corey Adams	MNE	Sophomore	Platform Development
Gangyuan Jing	DUS	Freshman	Platform Development

Despite the fact that the robot did not compete last year, the group was able to identify certain areas of improvement, such as accessibility issues and excessive weight. These issues led to a redesign of the upper half of the robot from scratch by the Platform Development team. The new upper deck was designed in SolidWorks with all of the design goals in mind. It was then constructed from aluminum, and now features a three-section hinged design comprising the forward deck, the rear capsule, and the rear deck (Figure 3). The forward and rear decks are constructed of 1/8" aluminum, whereas the rear capsule is constructed of 1/16" aluminium to minimize excess weight. The hinged design of the forward deck now provides accessibility to the motors and drive system. The rear capsule hinges back to reveal the rear deck, which serves as a mounting area for components such as the GPS receiver, computers,

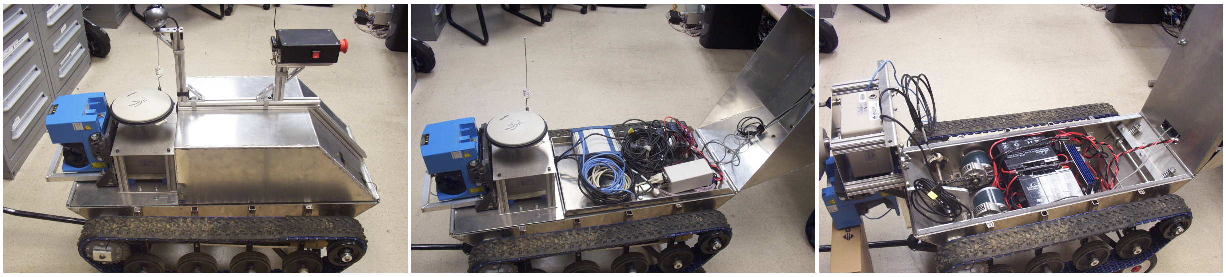


Figure 2: Depiction of new capsule arrangement showing all compartments closed, top capsule open, and all decks open

and power distribution box. The rear deck also hinges back, providing complete access to the inside of the tank tub where the batteries, motor controller, and R/C receiver are located. The modularity of the new design alleviates accessibility problems which were encountered last year due to the presence of Plexiglass panels that had to be unscrewed to access components. This new ease of accessibility allows swift addition and removal of various components, as well as quick troubleshooting in case of communication and power issues. The rear capsule is TIG-welded to provide a secure and watertight seal for all of the components housed within.

The drive system consists of two 350 W motors with a 10:1 gear reduction. Each motor individually powers the right or left tread of the tank. The gear reduction allows the power to be utilized at an acceptable speed. Shock absorbers are mounted on the rear wheel of each tread. The shock absorbers allow for play in the treads which may otherwise dislodge the tread from its wheels. Each motor shaft is equipped with a rotation encoder to provide accurate odometry information. Both, the

Figure 3: Depiction of new capsule arrangement showing all compartments closed, top capsule open, and all decks open

motors and the encoders, are controlled by the *RoboteQ AX2850* motor controller. The RoboteQ controller provides several different interfaces for communication and is highly configurable with the manufacturer provided software. All systems are powered by two 12 volt 18Ah lead-acid batteries connected in series, providing a total of 24 volts regulated by a 24v-24v regulator.

The design of the treads, which are the final link in transferring power to the ground, is worthy of separate mention. The treads were fabricated from acetyl, and designed in such a way that disassembly and reassembly on the fly can take place with minimal effort. The choice of plastic instead of metal allows for the use of the robot in many different environments without risk of long term damage of the driving surface or the treads. New from last year, however, is the addition of rubber grip on the treads. The rubber greatly increases the traction of the treads outside and inside, minimizing slip that is characteristic of differential drive tank-type robots. The rubber also allows us to test the robot in various indoor environments with almost no risk of damaging expensive floors.

The team spent a considerable amount of time and effort on this project. Over the duration of building the robot, an estimated combined time of 600 man-hours have been logged.

Innovations

Many interesting and novel technologies and approaches have been implemented on the Nittalion battalion for the IGVC. As previously described, the completely modular structure of the robot improves accessibility and results in considerable weight savings. The software environment chosen is also worthy of mention, as it facilitates collaboration, modularity, and distribution of computing resources. The Player environment provides interesting capabilities in the data fusion arena. The robot's laser obstacle detection strategy is an excellent example. Obstacles registered by the vision subsystem are superimposed onto a laser scan, creating virtual laser obstacles from lane edges and outlines. To the rest of the system, however, the combined data appears as though it were a laser. This strategy allows modifications of data without affecting other algorithms. Another innovative approach is the use of a rolling roadway simulator for testing vehicle dynamics and ramp climbing ability. The test data is then compared to modeled and calculated data from software simulation.

Power and Electronics

The power system is essential for the proper functioning of the robot. It is necessary that required power is provided to all components without interfering with component communication lines. Several measures have been taken to ensure efficient power distribution while minimizing interference. The batteries are contained in the tank tub bottom and are physically separated from the electronics that are located on the rear deck. Two 12V 18Ah lead-acid batteries in series are provided for the electronics, and two more are present for the motors. Appropriate fuses have been provided for the batteries. A twisted pair 24V supply line connects the two batteries dedicated to the electronics to our custom-made power distribution box, located on the rear deck. This box provides regulation to 24, 12 and 5 volts for various components on the robot via pairs of modular banana plug connections. Each component has a power cable terminating in a pair of banana plugs that fit into the color coded supply sockets. Thus, all signaling and communication processes, with the exception of signals to the motor controller, take place above the rear deck, isolated from the main power supply infrastructure. Signals to the motor controller travel through a shielded serial cable.

Figure 4 is the connectivity diagram showing all major robot components. Links to the device hosts from the motor controller and GPS receiver are RS-232. The link to the device host from the SICK laser is RS-422. Links out of the device hosts to the wireless router are Ethernet. The webcam is connected to the Laptop directly via USB.

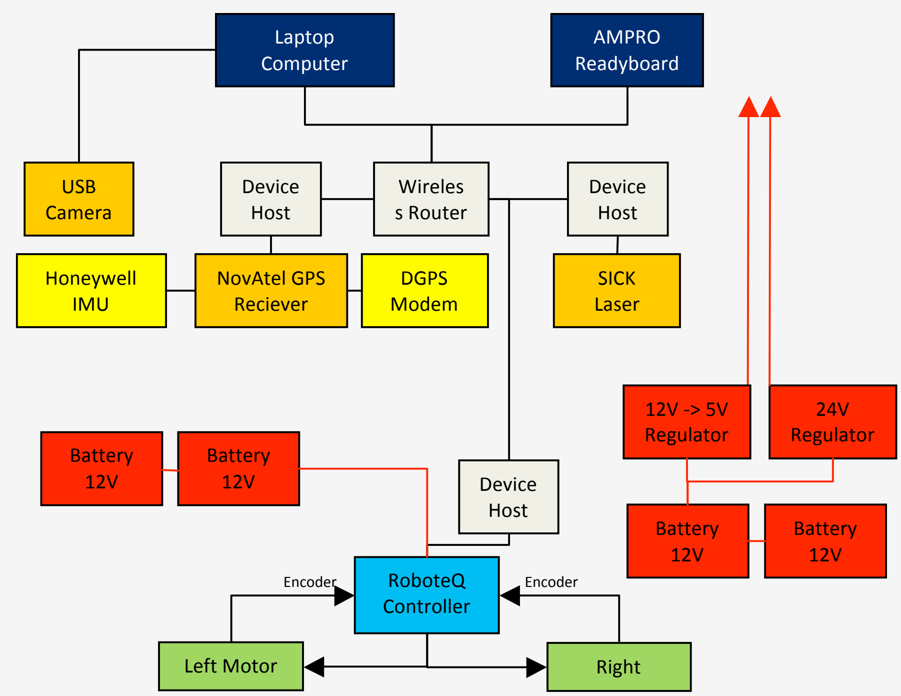


Figure 3: Connectivity diagram showing all major robot components

Sensors

The Nittalion Battalion uses a multitude of sensors to gather different types of data from the environment. For primary obstacle detection, a front-mounted SICK LMS-200 laser is used. The SICK laser has a range of up to 30 meters and can provide scans at 37.5Hz over its RS-422 connection. The serial connection is converted through a device master to 100Mbit Ethernet, where it is made available on the local network. Vision data is provided by a Logitech QuickCam Pro 5000. This commercially available USB webcam uses the UVC protocol to provide images up to 640x480 to the host computer.

Accurate position information is provided by the NovAtel SPAN system. The heart of this system is the NovAtel OEM4 dual frequency GPS receiver. Inertial corrections to the GPS data are provided by a Honeywell HG1700 IMU with ring-laser gyros. The SPAN system provides latitude, longitude, and bearing at 100Hz over a serial to Ethernet device master. With DGPS enabled, position data is accurate to 2 cm and bearing is accurate to 0.005 degrees. A brief summary of the sensors is provided in Table 2.

Table 2: Summarized Sensor Description

Sensor	Description	
Logitech QuickCam Pro 5000 **($ 75)**	▪ Commercially available USB webcam ▪ Uses UVC protocol ▪ Image resolution provided: 640 X 480 ▪ 30 frames per second	
SICK LMS – 200 **($ 4,400)**	▪ Range : 30 meters ▪ Scan rate : 37.5 Hz ▪ 0.5 degree resolution	
NovAtel DL4plus OEM4 dual frequency GPS receivers **($ 6,000 each)**	▪ Dual frequency (L1/L2) receiver ▪ Pseudo-range and carrier phase positioning capability ▪ Position accuracy of 2 centimeters (in differential carrier phase mode with tactical grade IMU)	
Honeywell HG1700 military tactical-grade IMU **($70,000)**	▪ Ring-laser gyro system with laser-calibrated MEMS accelerometers ▪ Drift bias: 10 deg/hr ▪ Acceleration bias: 3 milli-g ▪ Velocity and angle sampling rate: 600 Hz	

Software Overview

One of the major advancements for this robot is the change in the software environment. This year, the software is based on the Player/Stage environment running in Linux. Player provides a robust and configurable abstraction layer for robots, eliminating low-level manipulation of sensors from higher-level intelligence and algorithms. Stage builds on Player and provides a visual simulation environment that allows algorithm and behavior development to be independent of the physical robot. Such a setup allows swift and parallel development of different systems of the robot.

The software development process this year is devised for maximum collaboration. Every member of the team involved in software development or testing uses the Fedora 8 operating system, either in a dual-boot situation or in a virtual machine. All of the software and resources generated by the team are stored online in a subversion repository, allowing each individual to download a complete working copy of all the team-generated software resources. When a member changes a source file, he may commit the changes to the subversion system. Subversion also acts as an archival tool, and maintains all revisions of files committed to the system, allowing the team to “roll back” to a previous version if something goes wrong. Using subversion also ensures that, in the event of a system failure, no software is lost.

Vision and Image Processing

Vision and image processing operations on the robot are performed using the OpenCV Toolbox. Image processing is the main component used for detecting painted edges of a path, as well as detecting potholes and other non-traversable areas. For performing edge detection on a provided path, the image is first converted into the HSV colorspace and split into its individual channels. Depending on the requirements of edge detection, the channels are set up to have different threshold values. For example, bright white lines have a higher value in the “S” (saturation) channel and the “V” (brightness) channel than grass or dirt. Further, a set of thresholds can be tailored for detecting various kinds of features, such as white lines, yellow lines, orange road cones, etc. Based on the initial processing performed from the channel data, the images are converted to binary images with the desired features represented in white (Figure 5).

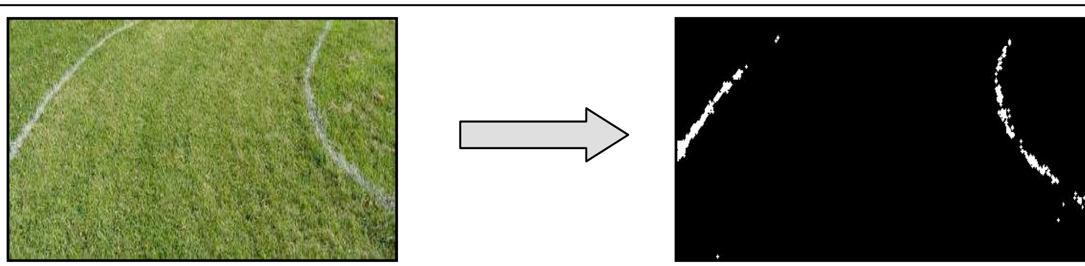


Figure 3: Conversion process to extract white boundaries

The binary images are then warped from a viewer perspective to a bird’s eye view perspective for compatibility with the mapping algorithm and the map generation process. This warping algorithm (Figure 6), starts with an image that has been captured looking ahead, by our top mounted webcam. As shown, the image perspective is transformed such that the image is represented in a top-down 2-D perspective, much like the data from a laser scan.

Mapping Algorithm

The mapping algorithm used for path planning purposes is based on a local area occupancy grid maintained within the robot. Creation of an occupancy grid requires several pieces of data to be written

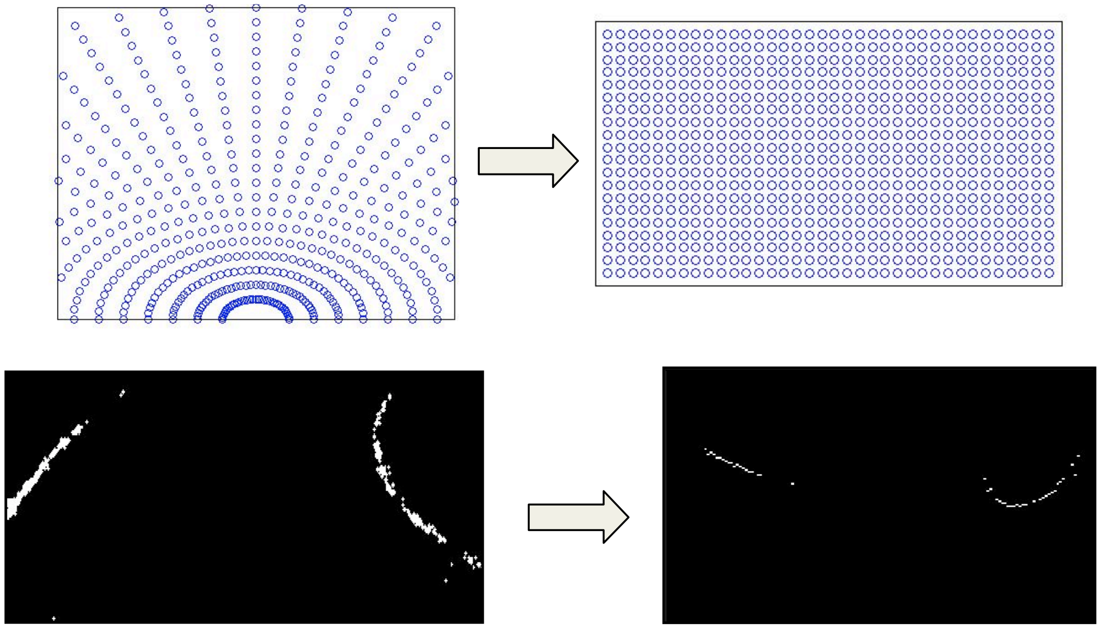


Figure 4. Conversion from the binary image plane to the warped plane where impassable regions are detected. Top shows onlooking perspective morphed into 2-D plane, bottom shows example transformation

into the map, including the robot's absolute position, its current heading, and coordinate data of obstacles. The data are provided reliably by the sensor array and written into a map file using Player. The map file is represented as a bitmap image, with each pixel representing a grid space of 1 meter by 1 meter.

Laser and vision data are fused together in a virtual laser device created in the Player environment. The laser data is collected, and then the vision data, now in a 2-D perspective, is superimposed on top of the laser scan. At any point that a laser scanline intersects a vision obstacle, the fusion algorithm creates a new laser point as though the laser had detected an actual obstacle where the vision algorithm "saw" the obstacle. The abstracted scan is then passed to the rest of the system as a normal laser scan would be passed.

Lane Following Algorithm

Data is taken from the map generated by the mapping algorithm and laser fusion. Lane edges, now defined as obstacles to either side, are judged impassible by the robot. The robot will attempt to stay in the center of the known good path. When other obstacles are present, they register on the occupancy grid the same way the lane boundaries do. The robot avoids the obstacles and lane edges, with no change in the underlying logic.

Waypoint Algorithm

A waypoint algorithm is used by the robot to determine its course to the next waypoint, from given waypoint data. Waypoint data are provided to the robot in the form of a text file listing the GPS coordinates of the goals. The robot runs a driver that reads these waypoints and the robot's current position. As the robot travels through the course, the current position of the robot is read by the driver, based on which the closest waypoint is determined and passed on to the global path planner. In certain situations, navigating around obstacles may bring the robot closer to a waypoint that is not the current target. In such situations this approach in especially advantageous, since it minimizes the driving time by switching to the closer waypoint. To prevent oscillation between waypoints, the driver keeps track of how many times the robot switches goals before reaching a target. After two switches, the current target is locked in regardless of distance, until the goal is reached. Current waypoint determination assists the path planning algorithm in determining the course of the robot to the goal.

Path Planning Algorithm

The robot employs two path-planning algorithms: a global path planner to navigate between waypoints, and a local path planner to ensure safe and collision free travel between waypoints. The **global path planner** is a modified version of Player's built in "wavefront" driver. The algorithm works using the robot's current position on the occupancy grid and determining adjacent unoccupied grids. The adjacent unoccupied grids are then assigned a cost of 1. Next, all unoccupied grids adjacent to the grids with cost 1 and away from the robot are assigned a cost of 2. The process is continued till the "wavefront" reaches the target waypoint. Once all unoccupied grids up to the target waypoint have been assigned a cost, the path planning algorithm calculates the shortest unobstructed path. This path is calculated by using greedily decreasing cost from target waypoint to current position. Any intermediate waypoints required to safely make it around obstacles are also calculated. The path planning algorithm is shown in simulation using the Stage robot simulator (**Error! Reference source not found.**). The left pane is the Stage environment with a virtual robot and virtual laser scanner to emulate our SICK unit. The right pane shows the mapping and wavefront algorithms in action, rendered in an OpenGL window. Red spots are the result of the mapping algorithm, which is similar to the actual map in the left pane. The large green dot is the goal waypoint for the robot, and the blue line is the shortest path to the goal. The blue dot indicates an intermediate waypoint generated by the global path planner, to avoid the red obstacles. The blue waypoint will be passed to the local path planner until it is reached, then the next waypoint will be passed.

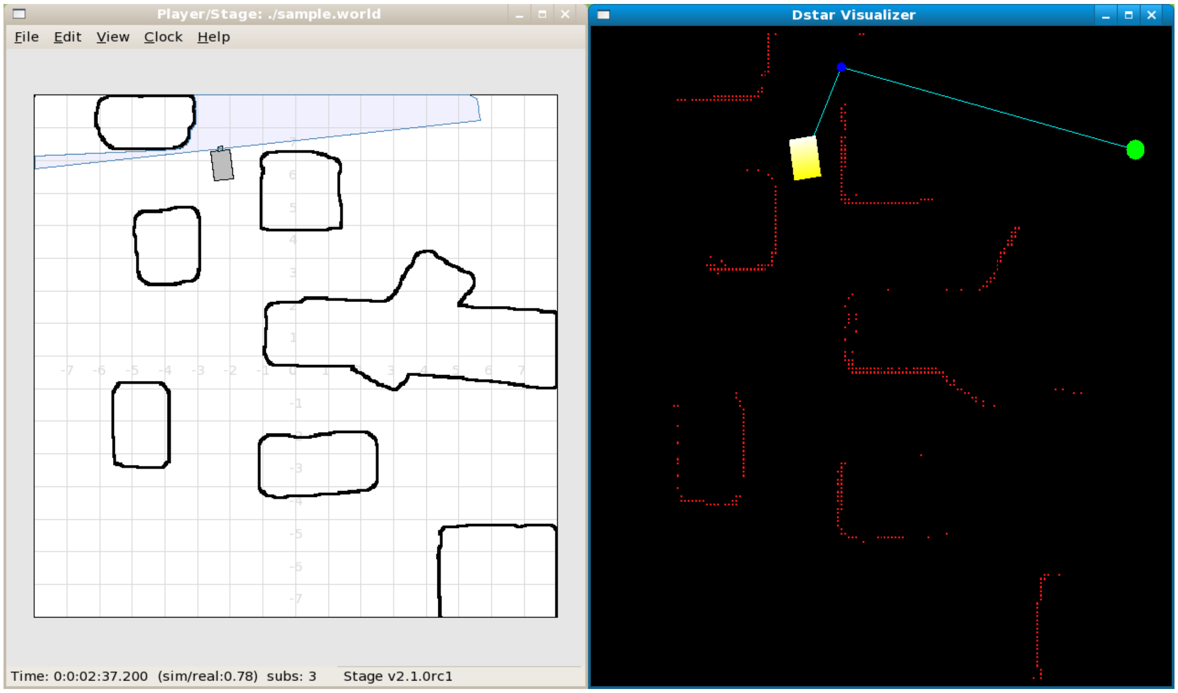


Figure 5: Simulation of robot in the Stage environment, showing map generation and global path planning

The **local path planner** is based on the Nearness Diagram (ND) Navigation algorithm[1]. The algorithm was designed to function in tight operating spaces, which is where a majority of other navigation algorithms have trouble functioning. The ND algorithm uses a laser to judge if there are any '*free walking areas*', or corridors between obstacles that are unoccupied. The absence of such areas enables the robot to know that it would be heading into a trap. Many other path planners fail in similar situations. Since a trap is not a *free walking area*, it is ruled out as a possible path to take. If the robot

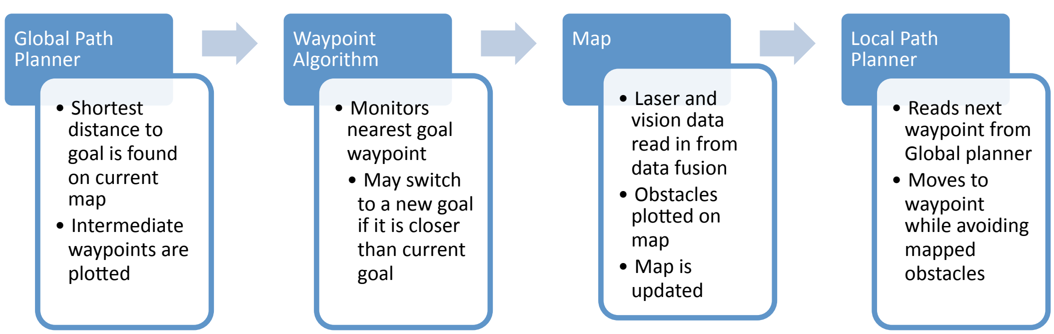


Figure 6: Flowchart of goal navigating process

[1] Nearness Diagram (ND) Navigation: Collision Avoidance in Troublesome Scenarios – Minguez, Javier; Montano, Luis. IEEE Transactions on Robotics And Automation, Vol. 20, No. 1, February 2004

ever becomes stuck in a trap, it can rotate in place until a suitable escape corridor is found. The ND algorithm uses waypoints generated by the global path planner to navigate the course. Once it reaches its target it acquires a new waypoint from the global planner and the process is repeated (Figure 6).

Efficiency

Several of the design decisions reflect the efficient use of materials and power by our robot. The choice of aluminum as the main building component provides strength for the platform, while minimizing weight. The use of thinner aluminum in non-structural parts of the robot: mainly the rear capsule, also aided weight reduction. Redesign of the upper deck and increased modularity made testing, maintenance, and configuration more efficient. Less time was spent disassembling and reassembling the robot to get at the components. Further, as much as 40% weight saving for the upper deck was realized with this transition. The motors were chosen to provide enough power to propel our 150 lb design weight across various terrains with ease, without being excessive. The use of rubber outer tires on the tank tread led to a substantial increase in traction, allowing the robot to tackle a wide variety of terrains.

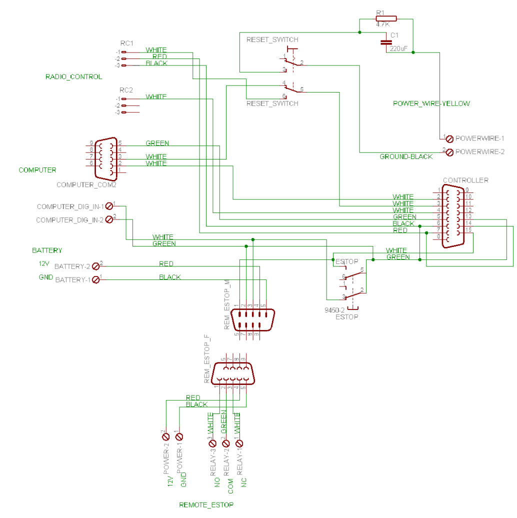


Figure 7: Wireless and Push Button E-stop wiring

Safety, Reliability, and Durability

Careful attention was paid during the design stage to make sure that all systems of the robot provided for safe and reliable functionality. One of the most important safety features is the ability to stop the robot immediately, either remotely or locally. The robot is equipped with a large emergency stop button placed prominently on the back. It is also equipped with a remote that has a line of sight range of 300 feet. These features provide enough flexibility to stop the robot in case of any exigency. The emergency stop works by toggling the emergency stop feature of the RoboteQ motor controller, which disables power to both motors. The E-stop push-button wiring is described in Figure 9.

The treads, which are a potentially dangerous area on the robot, were also designed with safety in mind. The tension in the treads has been kept low enough such that foreign objects may pass between the tread and its drive wheels with minimal damage. This is especially important when human hands are involved, as getting one's finger caught in the tank treads is a realistic probability.

Reliability in software has also received special attention. The Player platform allows for a modular architecture of sensors and nodes, with information passing between the nodes as TCP/IP traffic. This allows the use of an Ampro ReadyBoard and a laptop computer to perform different tasks in software, while communicating necessary information seamlessly between the two. This arrangement also allows for distribution of workload, so one computer is not responsible for all of the data collection, processing, and decision making functions. Further, nodes coming on and offline in case of an error will not take the entire system offline. This enables addition of extra nodes with minimal reconfiguration. In the event of a communication breakdown, the platform is programmed with behaviors that allow a graceful recovery and minimize danger to surrounding spectators and the robot.

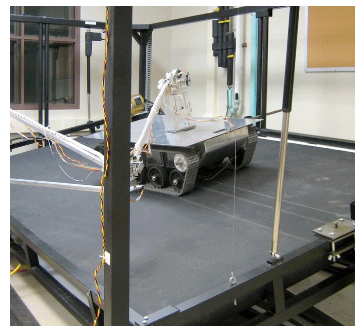


Figure 8: Ramp Climbing Ability on a roadway simulator

The Player platform also allows the simulation of much of our software in Stage without the use of a physical robot. This capability allows separate testing of the code. The code can be observed to be safe and behave properly, without risking damage to the robot or any of the team members. In addition, Stage also allows long term testing of software for reliability without the need for a physical robot to be present. This removes many issues, such as battery life and robot availability, from testing for problems like memory leaks, system crashing, or data errors.

Reliability was a key factor in choosing sensing equipment for the robot. Commercially available components were chosen to perform the required functions on the robot, where possible. These components have been used and abused by many other people, and have an established service record for reliability and durability. Employing such equipment provides recourse in the event of an equipment failure. Many of the components are under warranty against factory defects, and replacements for broken components can be easily obtained through retail channels.

Performance

The chassis uses two Unite 350 Watt DC motors with a built in 10:1 gear reduction. The motors are capable of operating at a no-load speed of 500 RPM with 24 volts DC and 3 amps. With a 5" drive sprocket, the maximum no load track speed is estimated to be 14.9 mph. In tests on grass the robot was measured to have a maximum speed of 4.5 mph.

The robot's runtime was tested in a constant driving condition, with no auxiliary electronics. The robot was equipped with 2 18Ah lead-acid batteries and driven non-stop. A total runtime of 1.5 hours was measured in 3 repeated tests. Taking into consideration the draw of auxiliary electronics with respect to the draw of the motors, the total drive time is estimated to be 1 hour. The total power requirement and a component-by-component break-up are provided in Table 3.

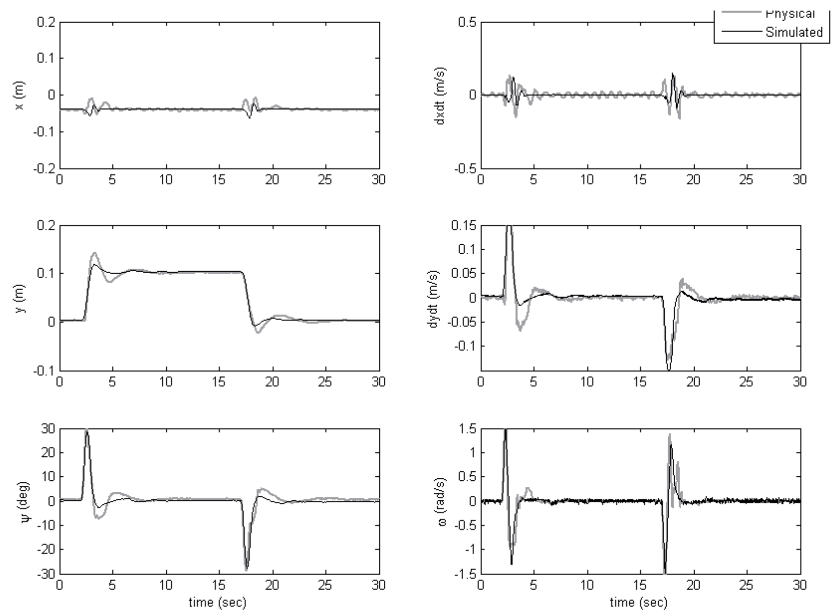
Table 3: Power Consumption Break-up

Component	Volts	Amps	Watts
Laptop	19	1.3	26
NovAtel GPS	12	0.29	3.5
Modem	12	0.1	1.2
Ampro IC Board	5	0.8	4
SICK LMS – 200	24	0.8	20
Motor 1	24	5	120
Motor 2	24	5	120
		TOTAL	294.7

There was some concern about the robot's sliding distance with the tracks locked during emergency stops at full speed and payload. Experiments were conducted that showed that the robot stopped within 1 foot on grass and within 3 feet on concrete due to the excessive slip. This agrees favorably with the friction curve analysis assuming a friction coefficient similar to a car tire on grass of 0.5. With an initial speed of 5 mph and applying basic kinematics it is estimated the robot would stop in 1.7 feet.

In hill testing, the robot was capable of climbing a 20 degree incline or 36% grade. The testing was done on a roadway simulator (Figure 10:). The simulator also allows modeling of chassis dynamics, allowing for accurate modeling and testing of vehicle response times (Figure 11). Using our roadway simulator, an accurate model of the chassis dynamics was created in SIMULINK. In order to have stability, low gains are required in motor controls, whereas accuracy requires higher gains. An accurate model of response times allows a tradeoff between in gains to be reached to provide optimal accuracy and stability. Testing with these gains allowed fine tuning of the path planning algorithms, as the behavior of the robot as it travels between the waypoints can be modeled and accounted for. This work was recently accepted into a special issue of the International Journal of Heavy Vehicle Dynamics, in a special issue on military tracked vehicles.

Figure 9: Simulated and experimental response times to square wave tracking commands, with corresponding state derivatives



With the optimal scenario of DGPS position accuracy of 2 cm, the accuracy of waypoint arrival is mostly dependent upon the path planning algorithms. Localizing to within 2 cm of the given waypoint is highly improbable, so the target waypoint radius is expanded. With the current control gains, the robot was able to arrive within about 0.5 meters of any waypoint with accuracy. The robot should be able to reduce this distance to 0.25 meters or less with optimization of the path following algorithm.

A summary of all performance criteria of the robot is provided in Table 4.

Table 4: Performance Tests

Test	Predicted	Measured
Max Speed	5 mph, predicted by power-curve of drive motor	4.5 mph, measured on grass
Battery Life	1 hour, predicted by amp-hour rating of system and runtime testing	1.5 hours, with no electronics
Ramp Climbing Test	15 degrees, predicted by friction curve	20 degrees, measured by ramp tests
Stop Distance	1.7 feet on grass, predicted by friction on grass	1 foot, measured on grass
Object detection distance	20 meters, predicted by range of laser scanner	20 meters
Accuracy of Waypoint	0.25 meters	0.5 meter

Arrival