Nittalion Battalion

Team Members: Adam Dean, James D'Iorio, Gary Geiger, Lee Gorny, Michael Petersheim, Alexia Ruiz, Pramod Vemulapalli

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

1.0 Introduction

As part of the graduate-level Mechanical Engineering Department Advanced Mechatronics course at Penn State, several graduate and undergraduate students designed the Nittalion Battalion tank robot to compete in the IGVC competition. This report provides a summary of this effort including the team organization and design process, innovation in the design, electronics design, software strategies, systems integration, efficiency of power and materials, and details on safety, reliability, and durability.

2.0 Team Organization and Design Process

To prepare for the competition and the robot build, the team competencies were identified by a skills survey, the members were partitioned into teams, and each team prepared a project dependency chart that detailed how competition requirements mapped to hardware and software design requirements. Additionally, the project dependency benefited all the team members because each could understand the various steps involved in making a robot with the competition objectives in mind. It also led to a careful scrutiny and understanding of the various rules, regulations and the goals of the contest and their consequences in terms of design constraints.

Fig 1. Nittalion Battalion out for testing

Table 1. Team members and responsibilities.

Member Name	Dept	Class	Responsibility	Hours
Sean Brennan	ME	Asst Professor	Faculty Advisor	50
Adam Dean	ME	2nd year PhD	Team Leader	200
James D'Iorio	ME	1st year MS	Power Layer	150
Gary Geiger	CS	1st year MS	Power Layer	50
Lee Gorny	ME	1st year MS	Power Layer	150
Michael Petersheim	ME	1st year MS	Intelligent Layer	200
Alexia Ruiz	ME	1st year MS	Sensing Layer	200
Pramod Vemulapalli	ME	1st year MS	Sensing Layer	200

The purpose of partitioning the class members into two teams early on was to foster the creation of novel ideas by competing one half of the team against the other. Not only did this lead to significant design improvements to the robot throughout the semester,

but also it served to grade and monitor each team and individual's performance. The two teams were partitioned by skill set such that each team was as equally distributed as possible in areas of programming, hardware, and sensor experience. Once the semester ended, the two design teams were combined from the class leaving only students interested in the IGVC team. Their responsibilities were again partitioned as shown in Table 1.

Analysis of the project dependency chart revealed many design constraints. First, it was highly desirable to field a robot with a capability to turn in place to escape blind alleys that are blocked. Second, the robot needed the capability to climb relatively large grades; specifically to climb a 10 degree grade under wet conditions with little possibility of wheel slip. Third, the robot needed to carry large loads, 150 lbs or greater, to support essentially unlimited battery power and maintain the mandated payload. The robot should also perform equivalently well under different soil types to minimize performance changes on concrete, sand, grass, gravel, etc.

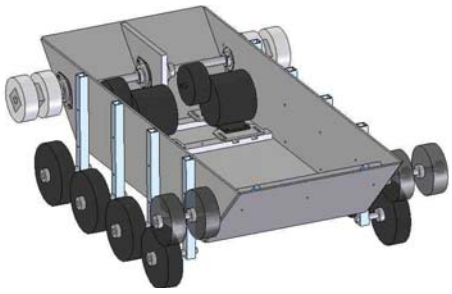

Fig. 2. Early CAD drawing of the tank chassis

These requirements motivated a tank-like chassis design. The chassis was fabricated from 1/8" aluminum sheets using a water-jet cutter, with the precision cuts tig-welded together at the seams. Aluminum was chosen for the frame material as the balance of rigidity and weight such that the frame provided support for all of the internals of the tank. A preliminary CAD drawing of the tank (minus the treads) is shown in Fig. 2. Power is delivered to the lower chassis through a RoboteQ AX2850 motor controller supported by two 12V 18Ah batteries connected in series and uses one rotary shaft encoder on each driveshaft for PID control. Actuation occurs through two 350 Watt DC motors with a 10:1 gear reduction. Each motor drives a separate hard acetyl track for skid-steering type propulsion. Utilizing this type of steering system maximizes maneuverability allowing the robot to turn with zero radius and eliminates the complexity of steering wheels. The track is allowed to deflect through a rear tensioning system, thus reducing the chances of dislodging the track from the road wheels. Finally, for an additional level of redundancy, three nearly-equivalent chasses systems were constructed from the same design to allow two robots for head-to-head competition while keeping one in reserve in case the other two were damaged.

Early in the design process, priority was given to a design approach that allowed as much flexibility in design changes as possible while at the same time allowing all team members to work in parallel. For this reason, the robot design was then partitioned into three layers: power, sensing, and intelligent control. The power layer includes the drivetrain and power supply and is located within the bottom-most level of the robot chassis. The sensing layer consists of a removable platform that can be removed quickly and which holds all of the sensing equipment, thus enabling the two design teams to quickly switch the sensing and intelligence layers between the two robot chasses. The intelligence layer consists of several removable embedded computer systems that are mounted inside a waterproof container on top of the sensing layer.

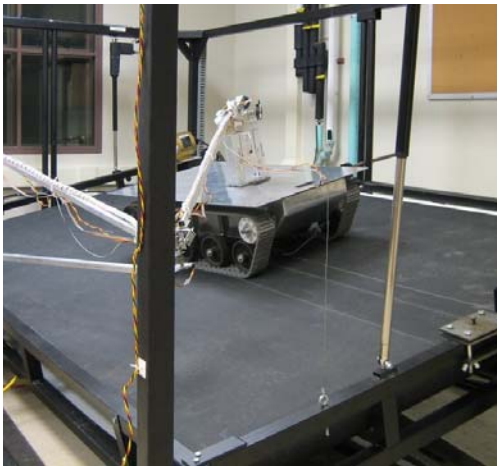

Fig. 3.: Testing the robot on a rolling roadway simulator.

3.0 Effective Innovation in the Design

There are two innovative design processes for which the team is particularly proud. The first is the use of a Rolling Roadway Simulator to test the robot dynamics. The second is the development of a 3D fully customizable MATLAB-based visualization GUI to optimize sensor placement. Each are described in more detail below.

A rolling roadway simulator consists of a large treadmill that is used to simulate vehicle motion on a controlled surface by moving the surface underneath the vehicle, quite similar to how a wind-tunnel moves air around a stationary aircraft. The rolling roadway utilized in this test, the Penn State University Rolling Roadway Simulator (PURRS), is a 5’ x 10’ treadmill that can roll and pitch and drive at speeds up to 30 mph. This system included an instrumented arm that attached to the top of the robot for relative position and orientation feedback. This testing capability enabled the team to model the dynamics of the robot within a Simulink representation. While details of this topic are beyond the scope of this report, the accuracy of the dynamic fit of the model, even under very aggressive maneuvering, is shown in Fig. 4. Such testing enabled the team to very accurately predict climbing ability, the friction curve, and stability of any feedback control law.

The resulting analysis showed limitations in the control effort where the robot track friction can saturate and begin to slip, thereby creating instabilities in tracking performance under very

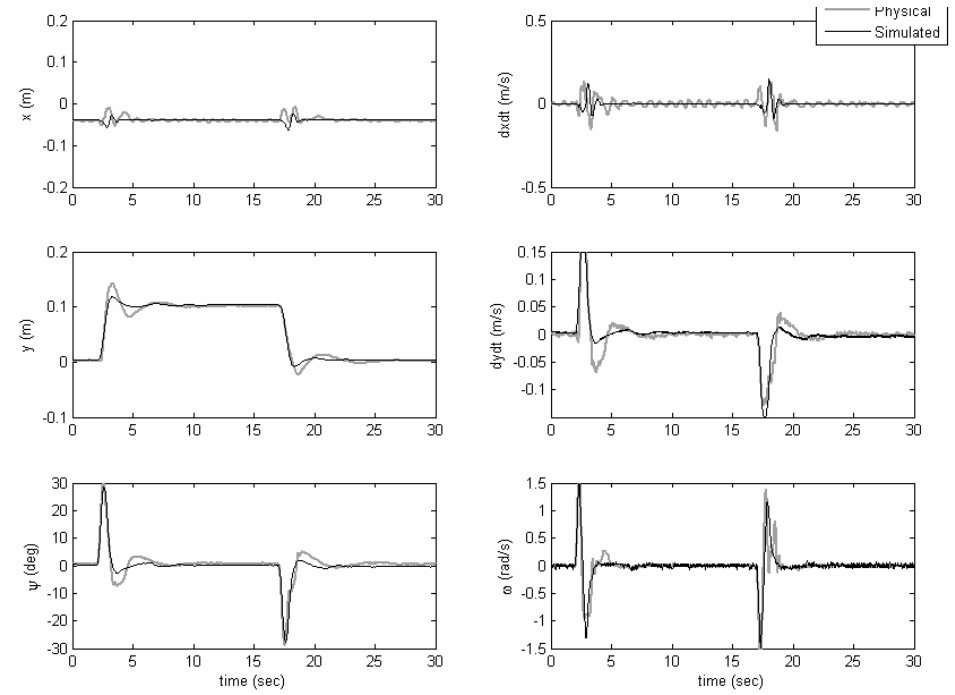

Fig. 4. Experimental versus simulated response of the robot to a square-wave tracking command showing x (top), y (middle), and yaw (bottom) states in the left columns, and their respective state derivatives (right columns).

high gain feedback control. However, high-gain feedback control is necessary to achieve low tracking error in point-to-point navigation. Thus, there is a tradeoff in controller gains where tracking performance requires gains to be high whereas stability requires gains to be low. Because the team accurately determined the robot's dynamic model, there was much greater understanding on how to design the chassis controller to best balance these two competing requirements. This work is currently in review for a journal submission.

The second innovative design approach was the use of a MATLAB-based simulation GUI to simulate the results of various sensor placements and localization strategies. The GUI provided an intuitive, interactive map-like interface that displayed a history of simulated sensor readings taken from the robot as it traversed through a simulated course. This serves as a virtual robot/environment/plant for testing and debugging robot locomotion, sensing functionality, and sensor placement. The entire visualization program was broken down into different sub functions which were allotted to individual members of the team, and finally integrated to form the complete Visualization GUI. The code for the visualization GUI was the basis for the simulation GUI that was subsequently written to simulate robot sensor output for various obstacle courses.

Instead of mounting the various sensors several times and testing them in several different terrain and obstacle situations, the simulation GUI allowed our team to quickly test a myriad

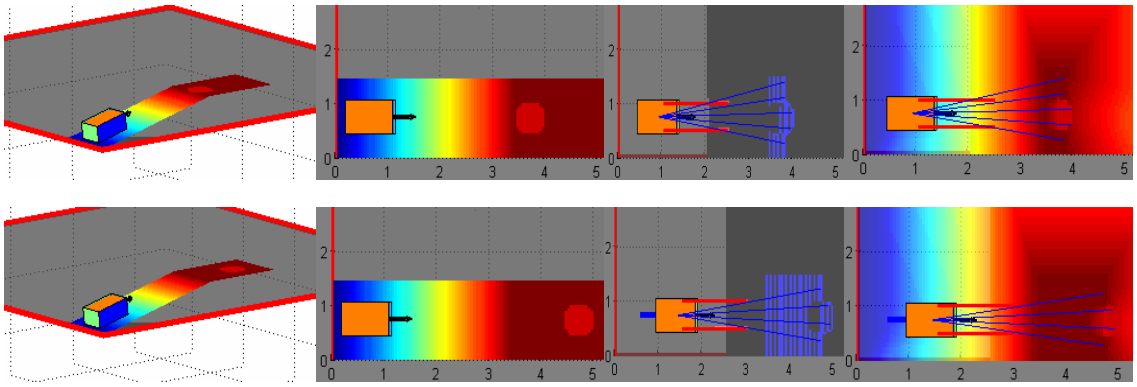

Fig. 5. GUI interface demonstrating the results of pothole detection tests after an upgrade. The columns show the 3D view at far left, 2D view, "occupancy grid" view, and vertical height maps at far right.

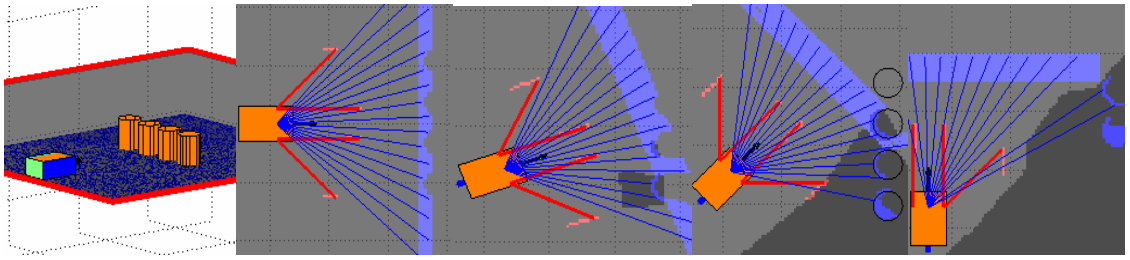

Fig. 6. GUI interface demonstrating the results of obstacle detection tests. At left is the 3D view. Subsequent columns show the 2D view and sensor readings for a heading of 0, 20, 45, and 90 degrees respectively to a row of barrels.

number of scenarios all virtually. The GUI, combined with the dynamic model obtained earlier, allowed the team to quickly test and tune different lane following, obstacle detection/avoidance and waypoint navigation algorithms. While experiments were still used for final verification, the benefit of being able to decrease the number of experiments is profound.

As an example of the GUI capability, a simulation of the effect of one particular sensor placement is given where the goal is to confirm whether the particular mounting of a given laser scanner allows detection of a 2" deep square pothole after an 10-degree upgrade and downgrade (Figs 5 shows upgrade). The goal was to maximize the forward distance viewable by the robot, minimize the laser mounting height, yet confirm that the potholes remain detected within the uncertainty of the sensor reading and robot orientation. Using the GUI, the optimal laser scanner placement was found to be approximately 36" from the ground and angled downward at an angle of 22.5 degrees.

Once the laser scanner was placed for optimal pothole detection, the sensor placement was tested with infrared sensors for obstacle avoidance. With the laser pointed 22.5 degrees down from horizontal, the team was concerned whether the robot would detect obstacles while

traversing a tight turn or non-normal geometry. Fig. 6 demonstrates sufficient sensor readings at different obstacle spacing intervals and heading angles. Fig. 7 shows the results of testing the robot through a switchback of obstacles with several pothole placements, demonstrating the ability to detect the two inch potholes around corners.

4.0 Electronic Design

The electronic design consisted of connecting the three design layers (power, sensing, and intelligent control) with minimum EMI noise passing between the systems. The high-current drive motors are isolated within the metal robot chassis, and all power leads up to the sensing and intelligence layers are twisted-pair, shielded wires safeguarded with fuses. The sensing and intelligence layers are also physically separated as a means to reduce the noise to the various components from the motors (Fig. 9). The wireless and push-button emergency stops are both

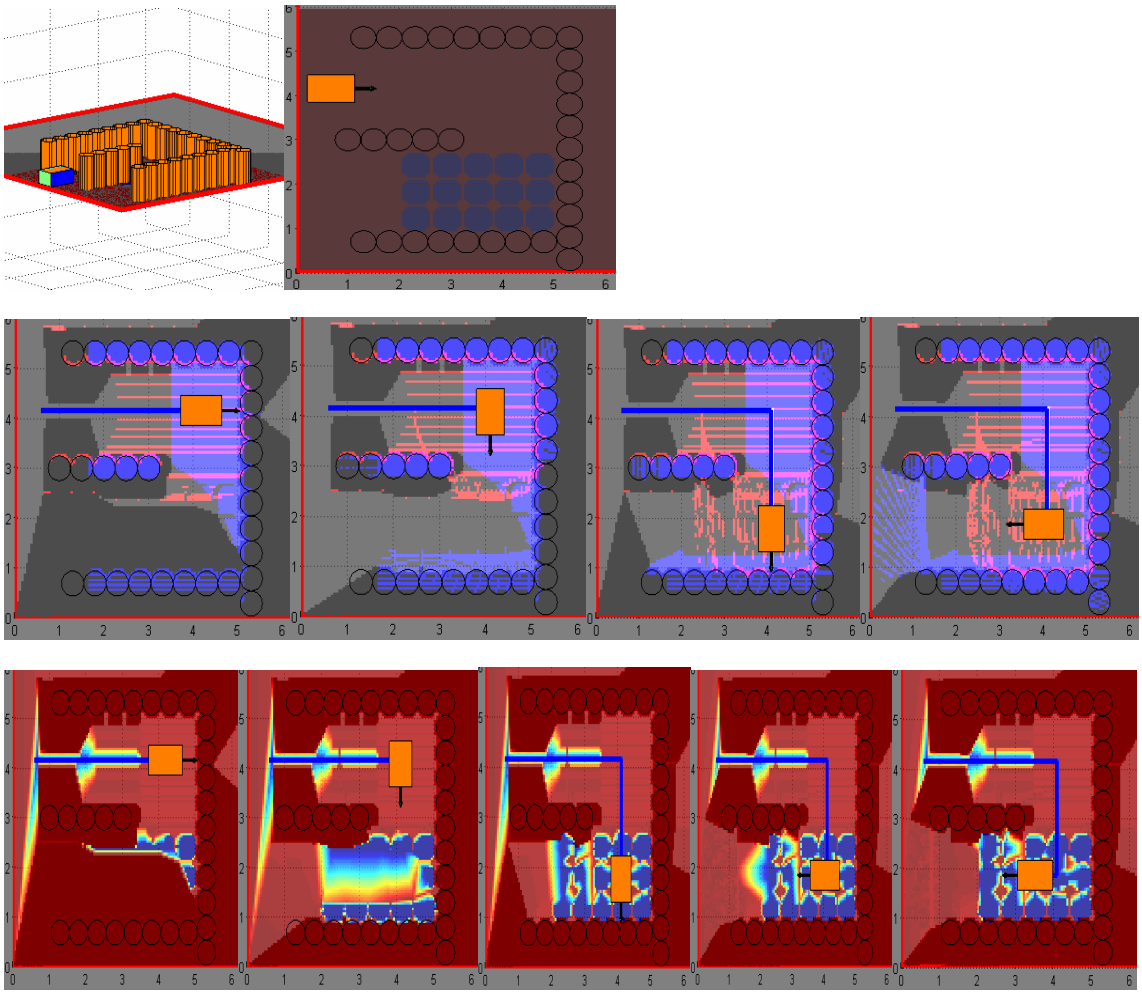

Fig. 7. GUI interface showing the results of pothole detection experiment through a switchback. The first row shows the 3D and 2D views, second row shows the occupancy grid from laser scanner and IR proximity sensing, and the third row shows the height map.

hardware based and sets the motor controller to emergency stop mode which grounds the motor terminals for an instant stop. The robot wiring harness connects the remote and on-board emergency stop switches to the motor controller and the intelligence system. In addition, a reset switch lets the robot motor controller alternate between radio control and RS-232 control. Fig. 8 shows the wiring diagram of both emergency stops. The wiring has been designed such that, by the flip of a switch and one serial command through the RS-232 port, the motor controller can be commanded to receive torque commands from either the intelligent layer or an RC radio.

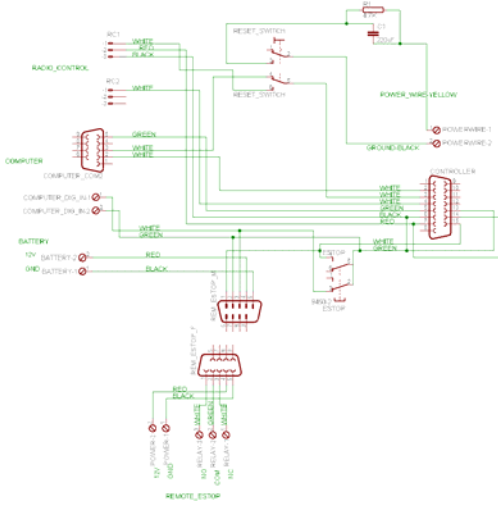

Fig. 8. Wireless and push-button E-stop wiring.

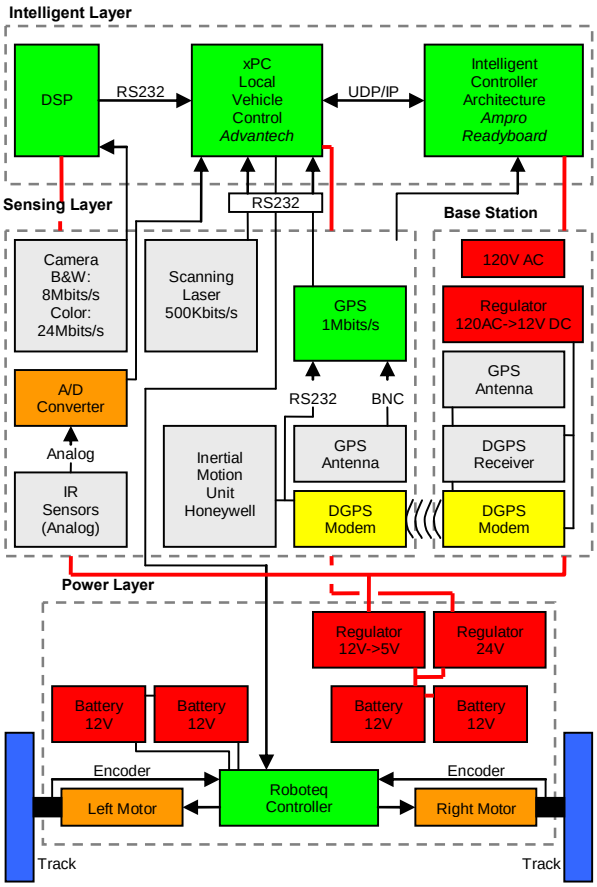

Fig. 9. Connectivity diagram showing all major robot components and bandwidth limitations

5.0 Software Strategies

5.1 Vision Algorithm

An onboard vision system is used to keep track of the lines that set the boundaries of the course and to avoid obstacles such as painted potholes on the grass. The algorithm used for the image processing first converts an RGB image sent by the camera into a gray

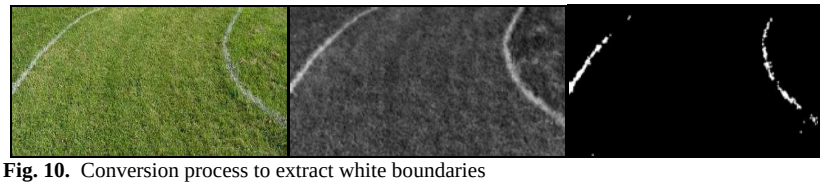

Fig. 10. Conversion process to extract white boundaries

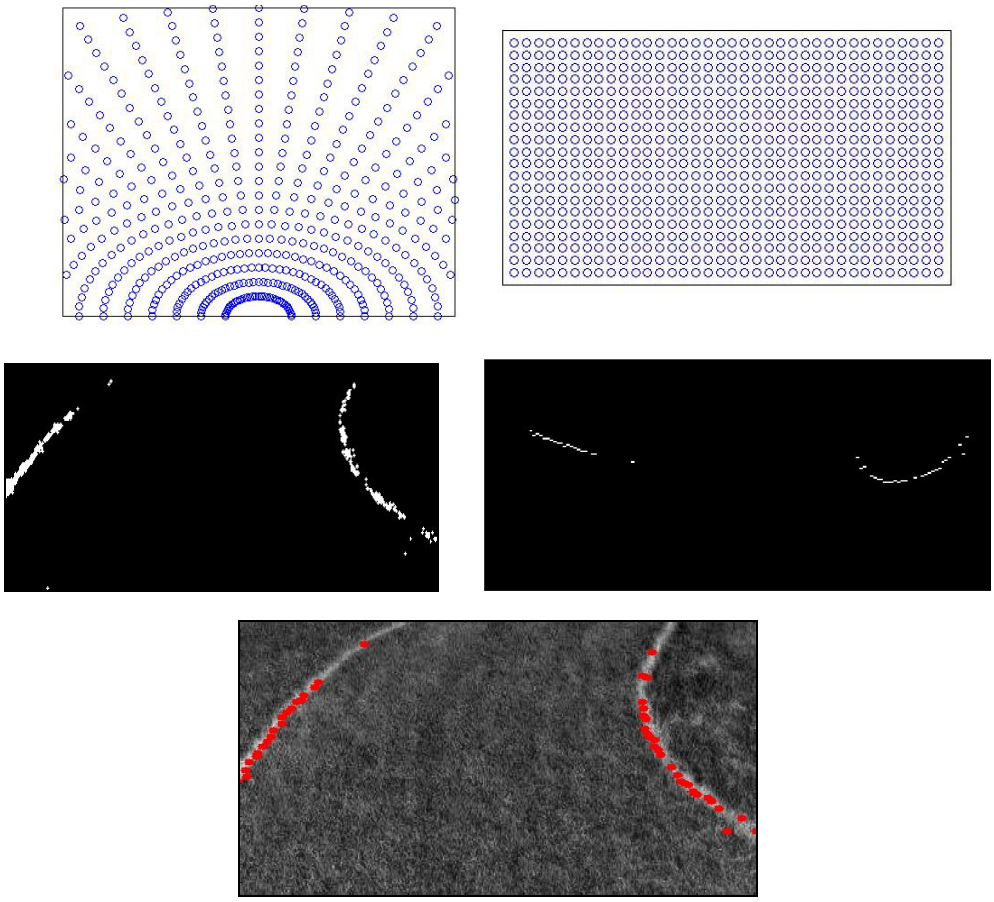

Fig. 11. Conversion from the binary image plane to the warped plane where impassible regions are detected. The regions are used to create an image scan similar to a laser scan where lines become virtual obstacles.

scale image using a color-weighted averaging method. After testing several methods, a suitable yet simple grayscale conversion method was found wherein we multiply the blue pixel values by two and subtract the green pixel values. The resulting image is then converted to binary using a specified brightness threshold. A comparison of an original, grayscale, and binary images are shown in Fig. 10. After the image is thresholded, it is warped using the model presented in Fig. 11 where the polar representation of the image is warped to a Cartesian representation. This warping allows the algorithm to quickly search along radial lines by simply searching down each column in Cartesian representation. This model looks for and stores the first non-zero binary pixel in each column, creating a distance estimate to the nearest painted line in a particular radial orientation. These distance measurements produce data in a format exactly compatible with laser scanner data, where the lane markers are represented as virtual obstacles and are merged with laser scans for obstacle avoidance and wall following. This process is especially beneficial as it intuitively fuses the vision data with other sensory data, thus reducing the complexity of specifying priority between vision data and other sensory data and allowing vision algorithms to

be debugged directly with algorithms previously validated for laser data, e.g. wall following and other types of obstacle avoidance.

5.2 Waypoint Algorithm

A waypoint controller was designed to regulate the speed and the steering of the robot based on the error between the robot's current GPS position and the desired GPS position. The gains of the controller were tuned through experimentation using a state-feedback control architecture prototyped on the rolling roadway simulator. The algorithm is set such that, once the error is reduced below a threshold of 25 centimeters, a new waypoint is commanded as the desired GPS position.

5.3 Obstacle Avoidance and Wall Following Algorithm

The wall following algorithm combines laser scanner, infrared, and vision data using an array of radial distances to obstacles at interval angles, centered at the front of the robot. Any detected objects, lines, potholes, etc., are all stored as equivalent obstacles. The fused sensor data is processed by the Intelligent Controller to perform the conceptual equivalent to either left- or right-wall following using a D-star algorithm. The algorithm then computes the closest filtered perpendicular distances from the robot in the longitudinal and lateral directions and subtracts them from a specified lateral traveling distance and an allowable longitudinal distance. A

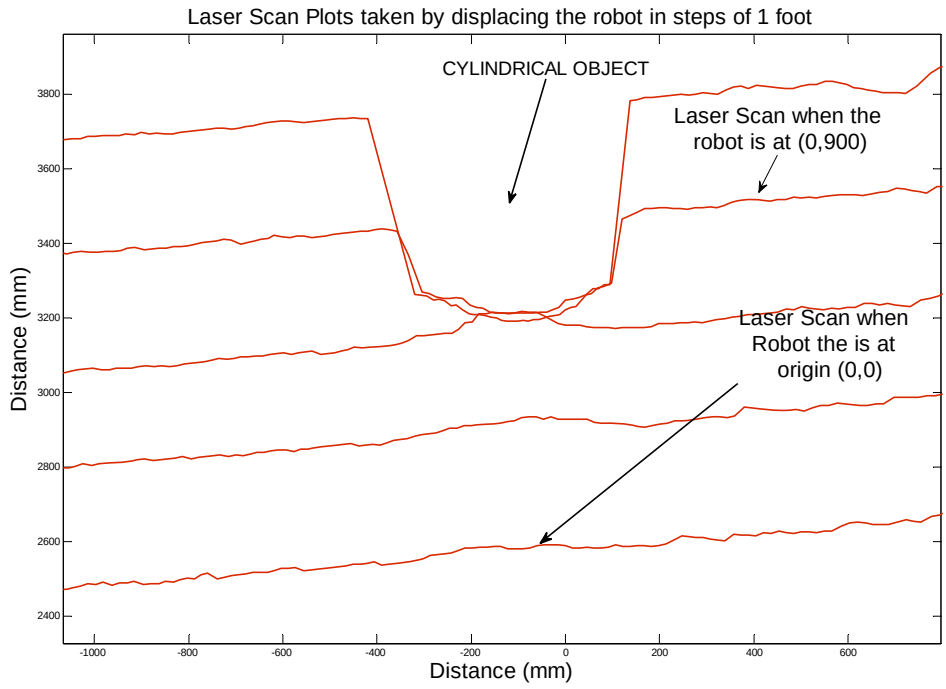

Fig. 12. GUI interface demonstrating the results of pothole detection tests after a downgrade. The columns show the 3D view, occupancy grid, and vertical height maps respectively.

proportional controller is then used to maintain the robot at a desired lateral distance from obstacles while turning the robot away from obstacles detected longitudinally.

5.4 Intelligent Controller

Early in the controller design process, it was found that simple waypoint following is insufficient for navigation through complex terrain due to the need to dynamically re-route the robot through "learned" obstacles. To achieve this higher level of path planning, an intelligent control (IC) architecture is used. The IC is programmed in an Ampro ReadyBoard 800 Single Board Computer (SBC) with Windows XP operating system. The IC code is a proprietary architecture developed by the Applied Research Laboratory (ARL) at Penn State for use in advanced weapons and robotics guidance. The IC uses an internal model of the world, populated in real-time with data from the xPC platform, to make decisions for the achievement of mission goals. The IC uses a Perception-Response strategy, where Perception performs data fusion and inferencing on obstacle sensor data received from the xPC platform. The Response portion of the code architecture involves activation of behaviors to achieve a desired outcome, a process shown in Fig. 13. For example, the IC directs the robot to a final destination using the D-star path-planning algorithm to route the robot around perceived obstacles. It returns transit commands to the xPC platform. In addition, data logging to a remote laptop is performed using an Engenius NL-2511CD PLUS EXT2 wireless card via an 802.11b ad-hoc network.

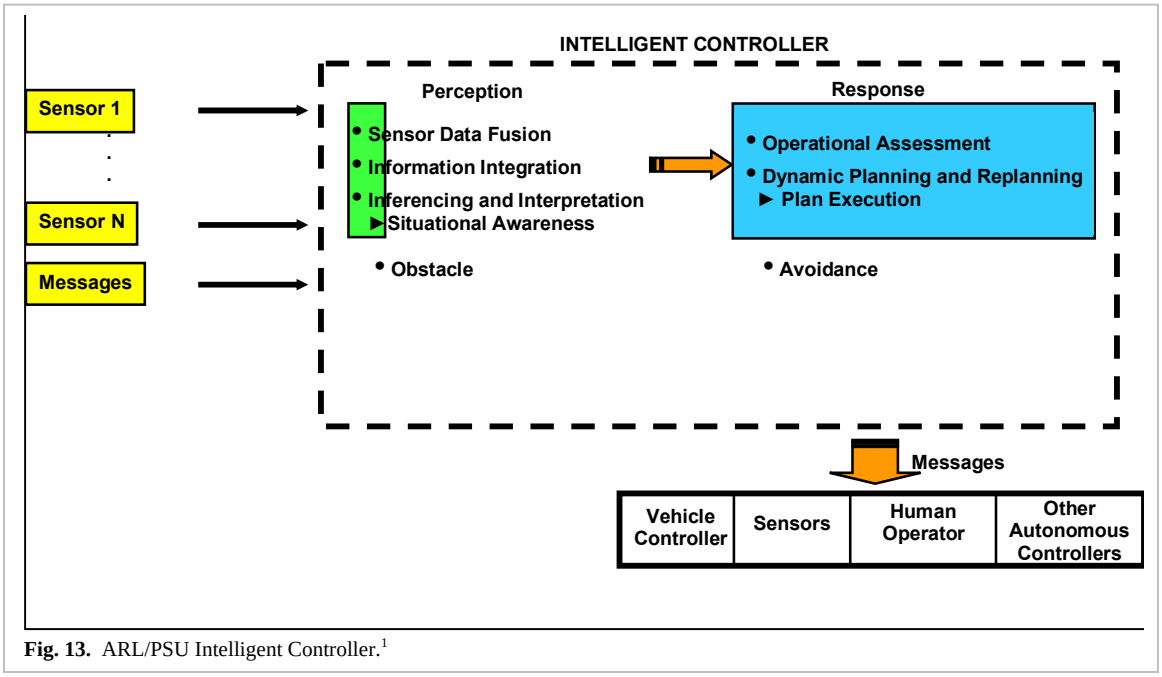

Fig. 13. ARL/PSU Intelligent Controller.[1]

6.0 Systems Integration

6.1 Sensors

The robot includes a wide range of sensor capabilities, each suited for different aspects of the competition. For the vision algorithm, images are captured using a COMedia C3088 CMOS camera, a low cost color camera module with digital output. The camera provides a 356x292 RGB image processed at a rate of 50Hz by an onboard TMS6713 DSK Digital Signal Processor. For obstacle detection, the robot uses a Hokuyo URG-04LX scanning laser and several Sharp GP2Y0A02YK infrared sensors. The vision, laser, and IR sensors are fused together by converting image data into distances to lines or potholes and combining the data with the infrared and laser scanner distance measurements. They are then used to create a temporary vector of angle and distance measurements to obstacles that is updated at every 0.01 seconds.

The GPS/INS sensing system used in the robot is the NovAtel SPAN system that uses OEM4 dual frequency GPS receivers and a Honeywell HG1700 military tactical-grade IMU with ring-laser gyros. This system outputs latitude, longitude and azimuth angles from North at 100 Hz rates, with DGPS accuracy of 2 cm and orientation accuracies near 0.005 degrees.

Table 2. Sensors

Sensor	Description	Picture
COMedia C3088 CMOS camera	A low cost color camera module with digital output that provides a 356x292 RGB image at a rate of 50Hz. ($100)	
Hokuyo URG-04LX scanning laser	Light weight (106 grams), 1 mm and 0.36 degree resolution, low power (2.5 Watts) camera with a field of view of 240 degrees. ($2700)	
Sharp GP2Y0A02YK Infrared sensor	Infrared sensor with a range of 10-80 centimeters with a low power consumption of 0.013 W. ($10)	
NovAtel DL4plus OEM4 dual frequency GPS receivers	The DL4plus is a dual frequency (L1/L2) receiver capable of both pseudo-range and carrier phase positioning and can transmit GPS measurements at a rate of 20 Hz. In differential carrier phase mode with the tactical-grade IMU, it has a position accuracy of 2 centimeters. ($6000 each)	
Honeywell HG1700 military tactical-grade IMU	This IMU uses a ring-laser gyro system with laser-calibrated MEMS accelerometers. Drift bias is only 10 deg/hr and an acceleration bias of 3 milli-g. Internally, velocity and angle changes are sampled at 600 Hz and made available to the NovAtel receiver at 100 Hz. ($70,000)	

Once the intelligent layer receives the vector of obstacles, desired waypoint and current GPS data, the D-star algorithm will either perform the waypoint navigation routine or wall following routine, depending on whether an obstacle lies in the path to the waypoint. When a desired waypoint is not specified and the robot is performing lane following, the D-star algorithm perform the left or right wall following and obstacle avoidance algorithms. These methods are especially advantageous because there is no need for an occupancy grid that requires a significant amount of memory storage. In cases of complex obstacles, the D-star algorithm performs the following: for fake potholes, the vision algorithm detects the circle as an obstacle and for real potholes the laser scanner or infrared sensors detects the change in height as an obstacle. While performing wall following, the algorithm follows the virtual wall as seen by the vision system or the obstacle wall as seen by the laser scanner or infrared sensor and continues to move through switchbacks and dead ends. Center Islands and barrel traps are detected by the laser scanner, and cause the robot to perform wall following while maneuvering around the obstacles. For dashed lines, the wall following algorithm switches sides whenever the guidance stripe has been lost.

7.0 Efficient Use of Power and Materials

Because the robot was initially designed for future use involving supporting heavy loads, the large amount of weight from the batteries in the competition-ready robot is quite negligible. The design called for a heavy load of batteries that could last up to several hours without recharging; although the competition is only several minutes per run, we did not see a need to re-wire lighter battery packs into the chassis only to save a small amount of space and incur a large detriment of power management problems. Our current approach should allow us to go through several hours of experimental runs and competition runs without having to recharge.

Our chassis was also designed to function in any weather. It has been tested in snow, rain and mud, and through such testing it was found that extra precautions in waterproofing the other layers of the robot were necessary.

The greatest efficiency in the robot design is the interchangeable parts. The entire top layer of the robot is designed to be easily removed and placed on another robot chassis in case one chassis has hardware failure. Also the batteries, motors, and motor controllers are all identical between the two robot chassis thus parts can be swapped out quickly and easily. The track is also quite efficient. It is easily maintained and assembled and the links can be quickly replaced using only a small screwdriver.

8.0 Safety, Reliability and Durability

Nearly every aspect of the robot design was constructed with safety, reliability, and durability in mind. In regard to safety, the most important design feature is the ability to stop the robot immediately. The wireless emergency stop is hardware-based and has a line-of site range of 300 feet (far beyond the mandated 50 feet). It operates by grounding the terminals on the drive amplifiers such that the robot comes to an immediate stop. A push-button emergency stop is also hardware-based to also ground the motor terminals.

Mechanically, the tread design also has a factor of safety for finger-pinching such that the tensioning mechanism allows enough slack to pass a finger through the tread spool without damage. While unfortunate when it happened, this feature has proved handy on a few occasions where a poor placement of the hand led to a finger being wrapped through the drive sprocket. Although immensely painful, there is no permanent damage, skin breakage, or bruising.

The primary sensing and control software architecture, Simulink, also favors safety by allowing all engineers on the project, not just the core programmers, to be able to debug and understand the code quickly. This has enabled a significant degree of cross-testing capability. Further, all code segments are required to be modular with all modules first tested and software examined individually by group members NOT involved with the coding. Software must pass multiple levels of testing, specifically: every code segment must be tested in Simulink alone, and then tested on the robot alone. If it passes these tests, it is then integrated with the rest of the system and again tested in simulation, then again tested experimentally.

Even the sensors were designed for safety, where the visible region by the robot was required to be a minimum of 3 times the stopping distance of the robot at full speed and payload.

In regard to reliability, the tank style was chosen because the robot's ultimate objective is to be maneuverable over any type of terrain thereby providing predictable performance irregardless of the terrain environment. Knowing that tread-jump is a primary failure mode for tank drives, the tread was designed to be mountable in the field within a few minutes and without the need of tools. Whenever possible, the team used proven commercial off-the-shelf (COTS) components for the robot, especially in the drive train and power amplification circuits which, when hand-built, are prone to failure due to the levels of torque and power involved. The software also has several levels of built-in reliability, including the use of several parallel real-time operating systems which each can serve as a "watchdog" on the other. The intelligent control software is an especially proven architecture as it is currently deployed within guidance systems for

many torpedo systems in our arsenal, features mechanical shock protection up to 100 g, and such efficient thermal design that no cooling fans are needed.

In regard to durability, the size of the robot was chosen to be as small as possible with at least a factor of 3 margin between self-weight and payload capability (e.g. the tank can operate with a payload 3 times its weight). This has been tested with payloads in excess of 200 pounds. The drive motors, with some modifications, can travel to speeds of about 15 mph. Thus, the design of the robot is such that it will be capable of much more than the IGVC competition requires.

9.0 Performance

The chassis uses two Unite 300 Watt DC motors with a built in 10:1 reducer that can operate at a no-load speed of 500 rpm with 24 volts DC and 3 amps. With a 5 inch drive sprocket the maximum no load track speed is estimated to be 14.9 mph. In tests on grass the robot was measured at 4.5 mph.

With the total power requirements outlined below in Table 3, and using two 12 VDC, 20 Ah batteries the powertrain, and two more for the sensing and intelligent layers, the total run time is estimated to be, assuming a conservative 50% discount factor due to high current drain, 1 hour for the powertrain and 6 hours for the sensing and intelligent layers.

For the ramp climbing test the robot was experimentally tested to be capable of climbing a 20 degree incline or 36% grade.

During emergency stops at full speed and payload, there was some concern about the robot's sliding distance with the tracks locked. 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.

Table 3. Power consumption

Component	Volts	Amps	Watts
Advantech xPC	24	2	24
NovAtel GPS	12	0.29	3.5
Modem	12	0.1	1.2
Ampro IC Board	5	0.8	4
Hokuyo Laser Scanner	5	0.5	2.5
DSP	5	0.8	4
Motor 1	24	5	120
Motor 2	24	5	120
		Total:	279.2W

Using the Hokuyo laser scanner with a range of 13 feet at an angle of 22.5 degrees from horizontal, the maximum distance an object can be detected longitudinally is about 7.8 feet. In experimentation the laser scanner was able to detect an object on the ground at a longitudinal distance of 7.6 feet from the robot.

With the optimal scenario of DGPS position accuracy of 2 cm, the accuracy of waypoint arrival is mostly dependant upon path-tracking controller gains. Without incredibly high gains, the robot would not be able to reduce the position error to 2 cm because of the friction in the drivetrain. With the current control gains, the robot was able to arrive within about 0.5 meters of the waypoint. With optimization in the control effort, the robot should be able to reduce this distance to 0.25 meters or less as demonstrated by the repeatability of the measurements on the test course as shown in Fig. 14.

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	Not tested
Incline 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	7.8 feet, predicted by geometry of laser scanner placement	7.6 feet, measured to first sight of obstacle
Accuracy of Waypoint Arrival	0.25 meters	0.5 meter

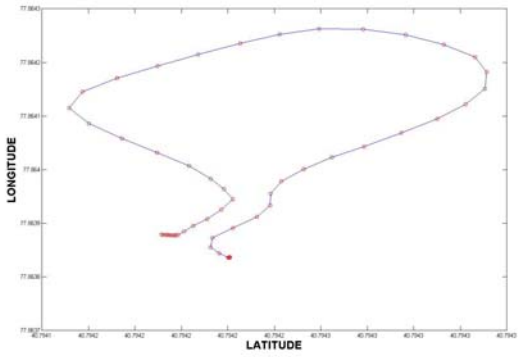

Fig. 14: Location of GPS path recorded and measured path.