



Design and Realization of an Intelligent Ground Vehicle

A Major Qualifying Project Report Submitted to the Faculty of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for

the Degree of Bachelor of Science

in Robotics Engineering, Electrical and Computer Engineering,

Mechanical Engineering, Computer Science

by Team Prometheus:

Justin Barrett

Rob Fitzpatrick

Chris Gamache

Ricardo Madera

Adam Panzica

Benjamin Roy

Daniel Sacco

Viktoras Truchanovicus

Bohua (Jack) Wang

Keywords:

1. Ground Vehicle
2. Robotics
3. Stereo Vision
4. GPU
5. CUDA
6. IGVC

Primary Advisor

Taskin Padir

Co-Advisors

Michael Ciaraldi

William Michalson

Stephen Nestinger

Kenneth Stafford

April 29, 2010

Project Number: TP1 MQP IGV1

Table of Contents

Table of Figures	6
Table of Tables	10
Abstract	11
Acknowledgments.....	12
Authorship Page.....	13
1: Introduction	14
1.1 Project Need	14
1.2 The Intelligent Ground Vehicle Competition	15
1.3 Problem Statement.....	16
1.3.1 Autonomous Challenge.....	17
1.3.2 Design Competition.....	17
1.3.3 Navigation Challenge	18
1.3.4 Joint Architecture for Unmanned Systems (JAUS) Challenge	18
1.4 Team Organization.....	19
1.5 Report Overview	20
2: Prometheus Design Overview.....	21
2.1 Mechanical System Features	21
2.2 Sensor System Features	22
2.3 Control System Features.....	22
2.4 Power System Features.....	22
3: Mechanical System Design	23
3.1 Research.....	23
3.1.1 Vehicle Overview	23
3.1.2 Chassis Frame.....	23
3.1.3 Cover	26
3.1.4 Wheels	27

3.1.5 Front Wheel	28
3.1.6 Motors.....	30
3.1.7 Motor Controllers	33
3.2. Development.....	35
3.2.2 Chassis Frame.....	35
3.2.3 Cover	36
3.2.4 Front Wheel	37
3.2.5 Rear Wheels and Power Train.....	37
3.2.6 Sprocket Reduction.....	39
3.2.6 Motor Controller.....	39
3.3 Design Analysis.....	43
3.3.1 Center of Gravity.....	43
3.3.2 Coefficient of Friction & Climb Angle.....	45
3.3.3 Front Wheel turning angle based on rear wheel speed differential.....	46
3.3.4 Turning Dynamics.....	48
3.3.5 Acceleration Performance	49
3.4 Results.....	51
3.4.1 Chassis Frame.....	52
3.4.2 Front Wheel	52
3.4.3 Rear Wheels and Power train	52
3.4.4 Motor Controller.....	52
3.5 Discussion.....	53
3.5.1 Chassis Frame.....	53
3.5.2 Front Wheel	54
3.5.3 Rear Wheels and Power train	54
3.5.4 Motor Controller.....	55
3.5.4 Future Work	56
4: Sensor System Design	57
4.1 Research.....	57
4.1.1 LIDAR.....	57

4.1.2 Stereo Vision	59
4.1.3 DGPS.....	62
4.1.4 Encoders.....	63
4.1.5 Digital Compass.....	63
4.1.6 IMU.....	64
4.2 Development.....	65
4.2.1 LIDAR	65
4.2.2 Stereo Vision	66
4.2.3 DGPS.....	85
4.3 Results.....	88
4.3.1 LIDAR	88
4.3.2 Stereo Vision	89
4.3.3 DGPS.....	91
4.3.4 IMU.....	93
4.4 Discussion.....	94
4.4.1 LIDAR.....	94
4.4.2 Stereo Vision	95
4.4.3 DGPS.....	95
4.4.4 IMU.....	96
5: Controls System Design	97
5.1 Research.....	97
5.1.1 Hardware	97
5.1.2 Software.....	100
5.2 Development.....	102
5.2.1 NI-cRIO	104
5.2.2 Main Board Computer Software Architecture.....	113
5.3.2 Internal System Network	114
5.3.3 Navigation	116
5.3.4 Control Center: The Graphical User Interface	122

5.4 Results.....	123
5.4.1 cRIO.....	123
5.4.2 Networking.....	124
5.4.3 Software Architecture.....	124
5.4.4 Mapping.....	125
5.4.5 Path-finding.....	125
5.5 Discussion.....	126
5.5.1 cRIO Discussion.....	126
5.5.2 Networking.....	126
5.5.3 Software Architecture.....	126
5.5.4 Mapping.....	127
5.5.5 Path-finding.....	127
5.5.7 Control Center.....	127
6: Power System Design.....	128
6.1 Research.....	128
6.1.1 Power Estimates.....	128
6.1.2 Battery Options.....	129
6.1.3 Emergency Stops.....	130
6.2 Power System Development.....	130
6.3 Results.....	133
6.4 Discussion.....	134
7: Results.....	135
8: Budget Estimate.....	137
9: Sponsors.....	138
6.1 WPI.....	138
6.2 The NVIDIA Corporation.....	138
6.3 The National Instruments.....	139
10: Bibliography.....	140
Appendix A: On-board computer specifications.....	143
Appendix B: NI cRIO Specifications.....	145

Appendix C: Hardware Control System Research	146
Mini ITX	146
Mini ITX System.....	146
Micro ATX.....	147
ATX	148
Laptop	149
PC104 Single Board Computer	150
National Instruments Compact RIO	151
Appendix D: Motor Performance Data	153
Appendix E: Battery Type Research	156
Appendix F: The Software Architecture.....	158
Appendix G: Competition Rules	163

Table of Figures

Figure 1: Stanford’s 2007 DARPA entrant “Junior”	14
Figure 2: ASIMO walking hand in hand	15
Figure 3: Bluefield State College - Anassa IV in action.....	16
Figure 4: Example Map for the Navigation Course	18
Figure 5: Sub-teams and their primary members (white font) and secondary members (black font)	20
Figure 6: Prometheus and its major components	21
Figure 7: Wheelchair Chassis of Y-clops.....	24
Figure 8: ATV Chassis of IONO	24
Figure 9: Welded-Aluminum Chassis Base.....	25
Figure 10: Finite Element Analysis of the Chassis Base	26
Figure 11 Chassis Cover	27
Figure 12: Initial Design of the Front Steering Mechanism	29
Figure 13: Front wheel steering mechanism	30
Figure 14: A scaled down view of the real size drawing of the chassis frame.....	35
Figure 15: Chassis frame building in progress	36
Figure 16: Exploded view of the wheel-sprocket assembly.....	37
Figure 17: Cutting sprocket tooth contour	38
Figure 18: Cutting chamfers using manual lath	38
Figure 19: Bridge Circuit Schematic	40
Figure 20: Motor Controller Logic.....	41
Figure 21: Motor Controller on Breadboard.....	Error! Bookmark not defined.
Figure 22: Final Motor Controller Board.....	42
Figure 23: Vehicle Center of Gravity, Coordinate System, and the Angled Incline	43
Figure 24: Polygon of Contact.....	44
Figure 25: Actual Center of Gravity of Vehicle.....	45
Figure 26: A schematic showing the vehicle making a right turn along with a table defining the variables.	47
Figure 27: Minimum turning radius diagram	48
Figure 28: Interpolation of T_{motor} as a function of Velocity based on the motor datasheet	49

Figure 29: Vehicle acceleration performance as tested on leveled grass.	51
Figure 30: Completed Vehicle.....	51
Figure 31: State Transition Logic.....	53
Figure 32: Front wheel steering system completed	54
Figure 33: Sprocket and chain completed	55
Figure 34: The Hokuyo URG-04LX (left) and SICK LMS291-S05 (right) LIDAR systems	59
Figure 35: Pentax C418DX.....	61
Figure 36: The Sokkia Axis 3 DGPS Receiver	62
Figure 37: The US Digital E8P Optical Encoders.....	63
Figure 38: The PNI V2Xe 2-Axis Compass Module	64
Figure 39: The InterSense NavChip IMU	65
Figure 40: LIDAR and cRIO	66
Figure 41: Stereo System Configuration	67
Figure 42: Image Rectification Process	68
Figure 43: Examples of calibration pictures.....	70
Figure 44: Rectified Images.....	72
Figure 45: Color distributions in RGB and modified	73
Figure 46: RGB image and its gray scale conversion achieved through SOM color reduction	75
Figure 47: Example of SOM map used for image segmentation.	78
Figure 48: Original color image and resulting segmentation without clustering	79
Figure 49: Segmentation with (right) and without (left) filtering	80
Figure 50: Pixel Block Matching	81
Figure 51: Original rectified left image and resulting non-filtered disparity map	82
Figure 52: Disparity map without filtering (left) and disparity map after filtering (right)	84
Figure 53: Line Detection	85
Figure 54: The UTM grid overlaid on the surface of the Earth	86
Figure 55: The geometry behind converting latitude and longitude to UTM.....	87
Figure 56: LIDAR Data test	89
Figure 57: Rectification	90
Figure 58: Color segmentation on outdoor environments	90
Figure 59: Original rectified image and resultant disparity after averaging.....	91
Figure 60: Line detection on football field.....	91

Figure 61: The dynamic DGPS data taken from walking around the perimeter of the WPI football field .	92
Figure 62: The stationary DGPS data taken from the center of the WPI football field	92
Figure 63: The 50% CEP (red), RMS (blue), and 95% CEP (green) errors calculated from the stationary DGPS data	93
Figure 64: Velocity information gathered from the IMU during a data set where the IMU was being pushed across the surface of a desk	94
Figure 65: Heading information gathered from the IMU during a data set where the IMU was being pushed on a cart	94
Figure 66: GPGPU vs. CPU speed	99
Figure 67: GPGPU vs. CPU image processing times	99
Figure 68: Vehicle Control System Diagram.....	103
Figure 69: Logical separation of the control system.....	104
Figure 70: cRIO Unit.....	105
Figure 71: cRIO CPU Program.....	106
Figure 72: cRIO FPGA Program.....	107
Figure 73: cRIO System Program Flow Diagram	108
Figure 74: Software package diagram	113
Figure 75: Libraries and their dependencies.....	114
Figure 76: Network packet diagram.....	115
Figure 77 Network Thread Diagram.....	116
Figure 79: Tentacles of speed sets 1, 3 and 5	119
Figure 78: Example of three tentacles and their radii. Support area is in gray.	119
Figure 80: Control Center.....	123
Figure 81: Optima Yellow Top Battery.....	131
Figure 82: Power Distribution Diagram	132
Figure 83: Temporary Mounting Board	133
Figure 84: Final Power Board Design	134
Figure 85: An autonomous intelligent ground vehicle Prometheus.....	136
Figure 86: Tesla C1060 GPGPU	138
Figure 87: Mini ITX Motherboard	146
Figure 88: Mini ITX System.....	147
Figure 89: Micro ATX Motherboard.....	148

Figure 90: ATX Motherboard	149
Figure 91: Laptop	150
Figure 92: PC104 Single Board Computer	151
Figure 93: cRIO System	152
Figure 94: Lib2dmaps diagram.....	158
Figure 95: libcomponents diagram	159
Figure 96: libdispatcher diagram	160
Figure 97: libtentacles diagram.....	161
Figure 98: libutilities diagram	162
Figure 99: 2010 Practice Map for the Navigation Course.....	174

Table of Tables

Table 1: Team Prometheus	19
Table 2: Pugh Analysis of Possible Wheel Configurations	28
Table 3: The Pugh chart comparing the wheelchair motors to custom motors.....	31
Table 4: Wheelchair motor specifications (RPM and Torque measurements are at output shaft from gearbox)	31
Table 5: Detailed analysis of the NPC T-64's specifications	32
Table 6: NPC T-64 performance	33
Table 7: Motor Controller Data.....	34
Table 8: Motor Controller Truth Table.....	41
Table 9: A comparison of technical specifications for the Hokuyo URG-04LX and SICK LMS291-S05 LIDAR systems	59
Table 10: Grey Point camera setups, ordered by preference.....	61
Table 11: Lens Specifications	61
Table 12: Calculated values of the 50% CEP, RMS, and 95% CEP errors from the stationary DGPS data collected at WPI	93
Table 13: Control System Matrix	98
Table 14: Example of a Probability map after an object has been detected.....	117
Table 15: FPGA Utilization.....	124
Table 16: Estimated power usage.....	129
Table 17: List of project requirements and their status	135
Table 18: Vehicle design evaluation	135

Abstract

This project focuses on designing and building an intelligent unmanned ground vehicle (UGV) which will become WPI's first entry to the Intelligent Ground Vehicle Competition (IGVC) in June 2010. IGVC challenges students to build and program a fully autonomous UGV that can locate and avoid obstacles, stay within the boundaries of a lane, navigate to GPS waypoints and implement a communications system using the Joint Architecture for Unmanned Systems (JAUS) protocol. WPI's intelligent UGV, Prometheus has a welded-aluminum chassis with two rear differential drive wheels and a steered front wheel. The vehicle power train consists of sealed lead acid batteries, bi-directional DC motor and chain sprocket sets. The software approach employs a stereo image processing implemented on NVIDIA's GPU. The image processing algorithms include segmentation using neural networks, rectification and pixel disparity calculation for line and obstacle distance detection. The sensor fusion technique processed on a National Instruments cRIO controller incorporates data from a LIDAR, differential GPS receiver, wheel encoders, and a compass. Navigation planning is accomplished by creating an obstacle probability map using the fused data and the path planning is implemented based on *Driving with Tentacles* approach. A GUI developed by using Java SWING provides controls and the means to monitor the vehicle state from a remote user station.

Acknowledgments

Team Prometheus would like to thank everyone who helped us out throughout the project especially our advisors Taskin Padir, Michael Ciaraldi, Stephen Nestinger, Kenneth Stafford and William Michalson, for their support, direction and patience. The team would also like to thank Brad Miller, Joe St. Germain, Neil Whitehouse and Mike Fagan for their valuable contributions to the project. For his efforts to help the team during the design phase, and for helping to obtain team sponsorship the team would like to thank Wayne Anderson. Furthermore, for their monetary and hardware contributions to this project the team would like to thank the NVIDIA Corporation, National Instruments, Optima Batteries, Philip Blackman, Genesis Strategies and Worcester Polytechnic Institute.

Authorship Page

This Major Qualifying Project has been completed by nine undergraduate students who contributed to all aspects of the project. The team has been organized into subgroups based on the vehicle design as a complete system. The contributions of team members to the writing of this report are summarized below.

Mechanical System

Rob Fitzpatrick

Adam Panzica

Benjamin Roy

Bohua (Jack) Wang

Daniel Sacco

Sensor System

Justin Barrett

Daniel Sacco

Chris Gamache

Controls System

Ricardo Madera

Viktoras Truchanovicus

Daniel Sacco

Power System

Daniel Sacco

Benjamin Roy

1: Introduction

1.1 Project Need

Robotics and intelligent systems are relatively new fields of technology and have been gaining the interest of various industries, organizations and the Department of Defense (DoD). U.S. military forces plan to spend \$5.4 billion in 2010 on unmanned vehicle technology. It is an increase of 18.4% over the 2009 spending.¹ The technology is mainly being used in military applications. Soldiers put themselves in danger when transporting materials, performing reconnaissance missions and performing search missions. In these instances, an autonomous intelligent ground vehicle would provide the ability to replace people with robots and ultimately save lives by keeping soldiers out of dangerous situations.



Figure 1: Stanford's 2007 DARPA entrant "Junior"

Advancement of the technologies involved in intelligent ground vehicles provides great benefits to society too. Currently, motor vehicles are dangerous and prone to accidents due to human error. In 2008 alone, there were 34,000 fatal car accidents². Navigation, mapping and object detection technologies have the potential to increase vehicle safety and prevent such accidents. Many automotive companies are incorporating intelligent systems into their newest models for a safer and more comfortable driving experience. Driver assistance technology which continuously evaluates the surroundings of the vehicle and provides information to the driver can also take control of the vehicle if needed.³ Smart cruise control, collision warning and airbag systems can potentially save thousands of lives daily and increase driving pleasure.

¹ (Keller 2009)

² (FARS 2008)

³ (Gavrila 2010)



Figure 2: ASIMO walking hand in hand

Furthermore, intelligent robots can be used to assist elderly or handicapped people. ASIMO humanoid robot developed by HONDA is capable of carrying objects while using tools, handling carts, walking hand-in-hand, recognize postures, gestures and greet people.⁴

1.2 The Intelligent Ground Vehicle Competition

The Intelligent Ground Vehicle Competition (IGVC) is an annual competition sponsored by the Association for Unmanned Vehicle Systems International (AUVSI), the US Department of Defense, Oakland University, TARDEC, QuinetiQ, NDIA of Michigan, Theta Tau, PNI Sensor Corporation, Robotic Systems and SAIC. The IGVC was started in 1994 by AUVSI to offer real-world experience to undergraduate and graduate students in the fields of vehicle mobility, control systems, sensors, and coding. It was also created to offer experience in intelligent vehicle design and unmanned military transport. This year, 2010, will mark the 18th year of the competition.⁵ This project is motivated by a university-level design competition and is focused on the design of an intelligent unmanned ground vehicle (UGV) that can navigate in an outdoor environment by following GPS (Global Positioning System) waypoints, staying within a lane and avoiding obstacles. An example UGV, the robot Anassa IV (a 2008 entry to IGVC from Bluefield State College) can be seen in Figure 3. The UGV must be able to carry a payload of 20 lbs. There are four major challenges that a vehicle can participate in:

- Autonomous challenge

⁴ (ASIMO 2010)

⁵ (IGVC 2010)

- Design Competition
- Navigation challenge
- JAUS challenge

Each of the challenges is discussed in detail later in this section. Since this is a competition, monetary awards are awarded for top 6 teams in each challenge. Detailed information on awards can be found in Appendix G: Competition Rules.



Figure 3: Bluefield State College - Anassa IV in action

1.3 Problem Statement

The goal of the project is to design and build an autonomous ground vehicle that meets the qualification requirements in the 2010 IGVC. Each vehicle participating in the IGVC must satisfy the following eight criteria (Competition rules are provided in Appendix G: Competition Rules):

- **Length:** The vehicle will be measured to ensure that it is over the minimum of three feet long and under the maximum of seven feet long.
- **Width:** The vehicle will be measured to ensure that it is over the minimum of two feet wide and under the maximum of five feet wide.
- **Height:** The vehicle will be measured to ensure that it does not exceed six feet in height; this excludes emergency stop antennas.

- **Mechanical E-stop:** The mechanical E-stop will be checked for location to ensure it is located on the center rear of vehicle a minimum of two feet high and a maximum of four feet high and for functionality.
- **Wireless E-Stop:** The wireless E-Stop will be checked to ensure that it is effective for a minimum of 50 feet. During the performance events the wireless E-stop will be held by the Judges.
- **Max Speed:** The vehicle will have to drive at full speed over a prescribed distance where its speed will be determined. The vehicle must not exceed the maximum speed of five miles per hour. No change to maximum speed control hardware is allowed after qualification. If the vehicle completes a performance event at a speed faster than the one it passed Qualification at, that run will not be counted.
- **Lane Following:** The vehicle must demonstrate that it can detect and follow lanes.
- **Obstacle Avoidance:** The vehicle must demonstrate that it can detect and avoid obstacles.
- **Waypoint Navigation:** Vehicle must prove it can find a path to a single 2 meter navigation waypoint.

In addition to the qualification requirement, the team set a goal to compete and place in all 4 challenges at the IGVC. Next, each of the four challenges the vehicle will participate in IGVC are described in detail. A complete set of IGVC rules can be found in Appendix G: Competition Rules.

1.3.1 Autonomous Challenge

For the autonomous challenge the vehicle must navigate a course of approximately 800 ft. long marked by continuous or dashed lines. The vehicle must be able to drive on grass, sand, pavement, simulated pavement or any combination of these. The course will include obstacles like construction drums, cones, light posts, street signs, natural obstacles including trees and shrubs etc. that must be avoided. The autonomous challenge will take place in any weather conditions and the vehicle must be able to withstand that. The vehicle will have a maximum of 5 minutes to complete the course under a maximum speed of 5 mph.

1.3.2 Design Competition

For the design competition, the team is judged based on a written report, oral presentation, and a judge's inspection of the vehicle. The requirements for the design competition are based on the judging criteria which include efficient use of space, serviceability, ruggedness, safety, original content, style, safety, reliability, durability, efficient use of power, and efficient use of materials and innovations. The team has made it a vehicle requirement to achieve all of these judging criteria.

1.3.3 Navigation Challenge

The navigation challenge requires the vehicle to navigate to 6 GPS waypoints. Besides all other requirements above this challenge requires the ability to reach a given destination in latitude and longitude with an accuracy of 1 meter or less. The vehicle must also be able to detect and avoid fences and drive through a 2 meter wide opening in a fence. For this challenge the vehicle must reach all of the waypoints in less than 6 minutes. Example of the navigation challenge course is shown in Figure 4.

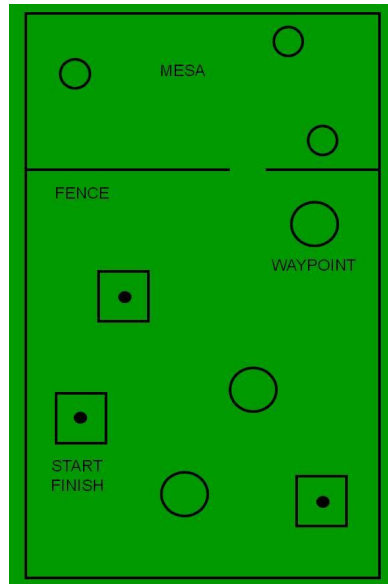


Figure 4: Example Map for the Navigation Course

1.3.4 Joint Architecture for Unmanned Systems (JAUS) Challenge

The Joint Architecture for Unmanned Systems originally developed by the department of defense is an open architecture for unmanned systems. It is a services based framework maintained by AS-4 Unmanned Systems Steering Committee which creates and revises all the standards for the framework. The team was provided with the standards needed to implement JAUS compliant interface.

Each vehicle will be required to interface with the Judge's COP (Common Operating Picture) providing information as specified below. The general approach to the JAUS interface will be to respond to a periodic status and position requests from the COP. This requires the support of the JAUS Transport Specification and the JAUS Core Service Set. The JAUS Transport Specification supports several communication protocols, the competition will use only the Ethernet based JUDP. The Core services required for the competition include the discovery, access control, and management services. The JAUS Mobility Service Set or JSS-Mobility defines the messaging to be used for position communications and waypoint based navigation. More detailed description about the JAUS protocol can be found in Appendix G: Competition Rules.

1.4 Team Organization

Team Prometheus consists of nine undergraduate senior engineering and computer science students. The team members have a multidisciplinary set of skills which is essential for successfully completing the project. Table 1 below shows all the team members, their majors/minors and the year of graduation. More information about each team member can also be found on the team's website www.igvc-wpi.org.

Justin Barrett	Undergraduate Robotics Engineering 2010
Rob Fitzpatrick	Undergraduate Mechanical Engineering 2010
Chris Gamache	Undergraduate Robotics/Computer Science 2010
Ricardo Madera	Undergraduate Computer Science 2010
Adam Panzica	Undergraduate Robotics/Mechanical Engineering 2010
Benjamin Roy	Undergraduate Electrical and Computer Engineering 2010
Daniel Sacco	Undergraduate Electrical and Computer Engineering 2010
Viktoras Truchanovicius	Undergraduate Computer Science 2010
Bohua "Jack" Wang	Undergraduate Mechanical Engineering 2010

Table 1: Team Prometheus

In order to manage the team efficiently and coordinate the use of the skills each member possesses, the team is organized in 4 sub-teams. Figure 5 depicts the organization of Team Prometheus and the primary and secondary members in each sub-team.

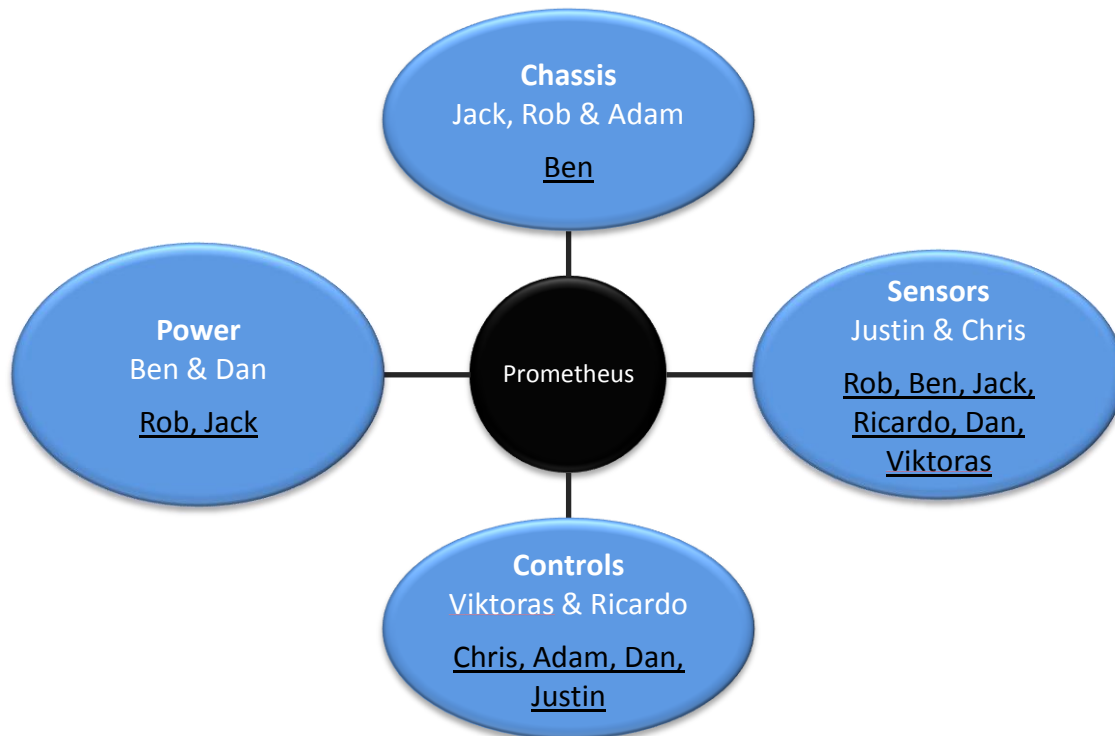


Figure 5: Sub-teams and their primary members (white font) and secondary members (black font)

1.5 Report Overview

This report introduces and explains the design and development process of the intelligent ground vehicle Prometheus. The rest of the report is organized as follows:

2: Prometheus Design – Provides a general overview of the vehicle and its subsystems.

Chapter 3 – Explains the design and development of the mechanical system.

Chapter 4 – Provides an overview of each sensor.

Chapter 5 – Explains the design and development of mapping, path planning and control of the vehicle.

Chapter 6 – Describes power system components and their integration.

Chapter 7 – Provides an overview of the results and accomplishments.

2: Prometheus Design Overview

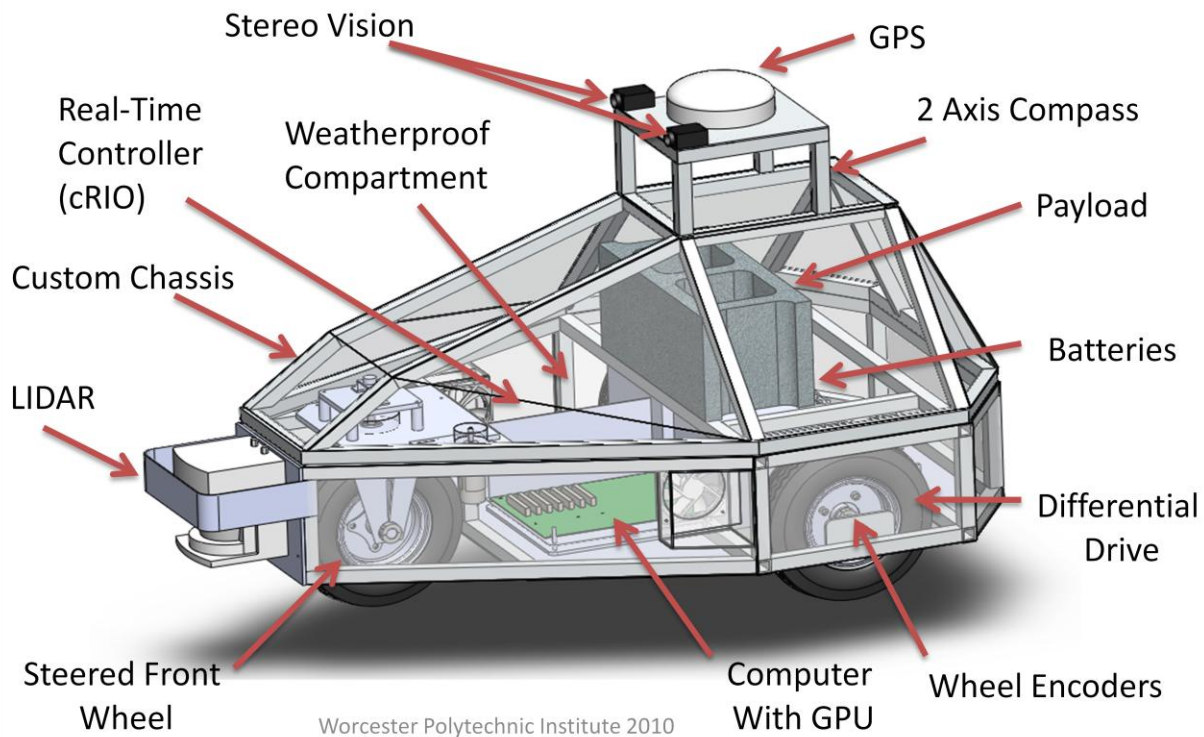


Figure 6: Prometheus and its major components

Before discussing the subsystems that make up Prometheus in detail, it is necessary to provide an overview of the vehicle. The components and features of the vehicle are summarized in four categories below.

2.1 Mechanical System Features

Feature	Components
Differential Drive	12" Pneumatic wheels Two 24V DC Brushed Motors Feedback Controlled Motor Drivers
Steered Front Wheel	Custom-made Steering Fork 24V DC Motor Actuated Encoder position feedback
Custom Built Frame	Welded Aluminum Square Tube Weather Proof Shell Top Opens for Maintenance and Payload Storage

2.2 Sensor System Features

Feature	Components
Obstacle Detection	SICK LMS291-S05 LIDAR
Lane Following	Stereo Vision system using 2xFL2G-13S2M/C Cameras
Waypoint Navigation	Sokkia Axis 3 DGPS Receiver
Localization	3xQuadrature Encoders

2.3 Control System Features

Feature	Component
Sensor Fusion	NI cRIO-9074
Mapping and path planning with	Main-boards computer ATX, 6GB DDR3, 4x2.66GHz
Image Processing	C1060 NVIDIA GPU using CUDA
Visualization of LIDAR data	Control Center - The graphical user interface written using JAVA SWING
Manual Control of the Vehicle	
Ease of Testing	

2.4 Power System Features

Feature	Component
On-board power source	2 Sealed lead acid batteries
Circuit Protection	Resettable fuses Main Circuit Breaker
Power Conditioning	UPS DC-DC Power Converter for Computer
Circuit control relays	

3: Mechanical System Design

The challenge for mechanical system is to design and build the chassis, power train, steering mechanism and the cover of the vehicle within the restrains set by the IGVC rules while achieving optimal performance in maneuverability, durability and power efficiency.

3.1 Research

In developing the mechanical system of the vehicle, the team reviewed the mechanical design of previous IGVC vehicles⁶ that had performed well to analyze the strengths and weaknesses of different design features. Then, the team also performed Pugh chart analyses of motors and wheels to choose the best option for each.

3.1.1 Vehicle Overview

In the following sections, the vehicle's mechanical system is defined in more detail. But to start, the following overall dimensions were measured from the CAD model created for the vehicle:

- Vehicle dimensions: 47 in x 35 in x 28 in
- Vehicle weight: 212.8 lb
- Wheel diameter: 12.5 in
- Center of gravity: 11.01 in from ground, 7.89 in forward from back wheels, and centered at 13.74 in
- Ground Clearance: 4.05 in

With the overview information of the vehicle, the following sections delve deeper into the research and design of the vehicle.

3.1.2 Chassis Frame

A common feature of vehicles that compete at IGVC is that they use commercially-available chasses, such as, electrical wheelchairs chasses (Figure 7) and All-Terrain-Vehicle chasses (Figure 8.)

⁶ (Akroush, et al. 2008), (Barry, et al. 2008), (Cardwell, et al. 2009), (Garcia and Zabala 2006), (Givental, et al. 2008), (Y-clops Robot 2005)



Figure 7: Wheelchair Chassis of Y-clops⁷



Figure 8: ATV Chassis of IONO⁸

A commercially-available chassis would greatly simplify the mechanical design of the vehicle because it would:

- Allow for more time to be used elsewhere on the vehicle
- Decrease the difficulty of building a custom chassis
- Allow for rapid development

⁷ (Y-clops Robot 2005)

⁸ (Garcia and Zabala 2006)

On the other hand, a commercially-available chassis would be:

- Expensive
- Not modular
- Not unique
- Inflexible

WPI's intelligent UGV Prometheus features a compact, durable, custom-design chassis manufactured by the team members. The vehicle is made up of welded 1 in x 1 in square 6061-T6 aluminum alloy tube with a thickness of 1/8 in. The chassis consists of two parts, the base and the cover. The base contains the motors, batteries, electronics, and LIDAR while the cover contains the stereovision, GPS antenna, and payload. The base, shown in Figure 9, is separated into compartments, the front compartment holding the front wheel, the second compartment holding the electronics, the third compartment holding the batteries and the motors, and the fourth compartment holding the power distribution components. The base was designed such that space would be efficiently used while the base could still accommodate all of the vehicle's parts and payload. Careful attention was paid to the location of each beam to evenly distribute the stresses throughout the structure. All unnecessary beams were eliminated or combined with other beams such that each beam serves for at least one structural purpose.

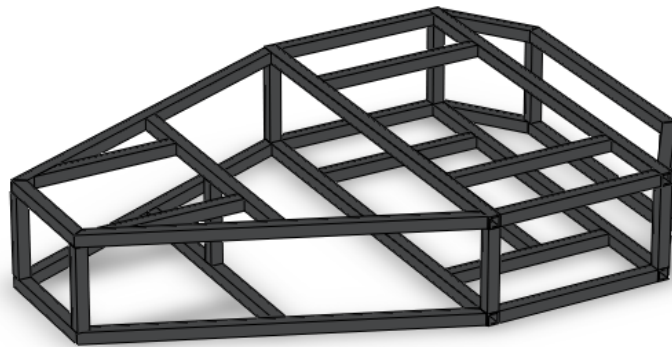


Figure 9: Welded-Aluminum Chassis Base

A Finite Element Analysis was performed to validate the design using Solidworks 2009. The "beam" mesh type was used which means the load can only be taken at the joint or uniformly distributed across a beam. The worst case scenario was simulated for 3 times the normal weight of the vehicle, which could occur while the vehicle is driving over a bump. As shown in Figure 10, the minimum

safety factor of the base is 19. A large safety factor is necessary for a prototype vehicle like this such that it can endure unexpected situations such as bumpy terrain, collisions with obstacles, and malfunctioning code or electronics.

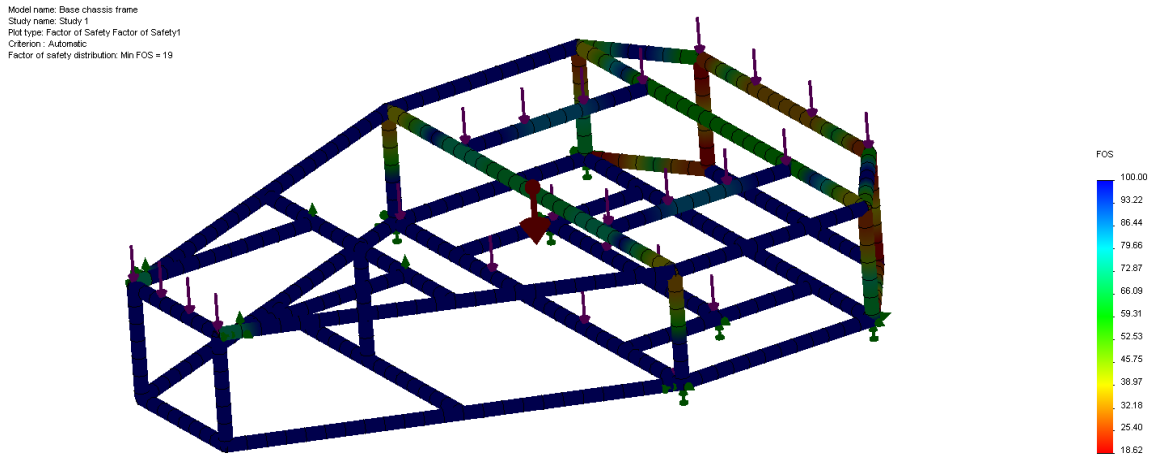


Figure 10: Finite Element Analysis of the Chassis Base

3.1.3 Cover

The cover of the chassis was designed in a similar manner as the chassis base, ensuring that each beam had a structural purpose. While creating the CAD model, it was important to remember that the two main purposes of the cover was to elevate the vision system to the necessary height of 2 ft 11 in and to create a housing in which the payload could be securely housed within the vehicle.

All angles for the cover bars were restricted such that they were only angled compared to either the x- or the y-axis. This is depicted in Figure 7 below.

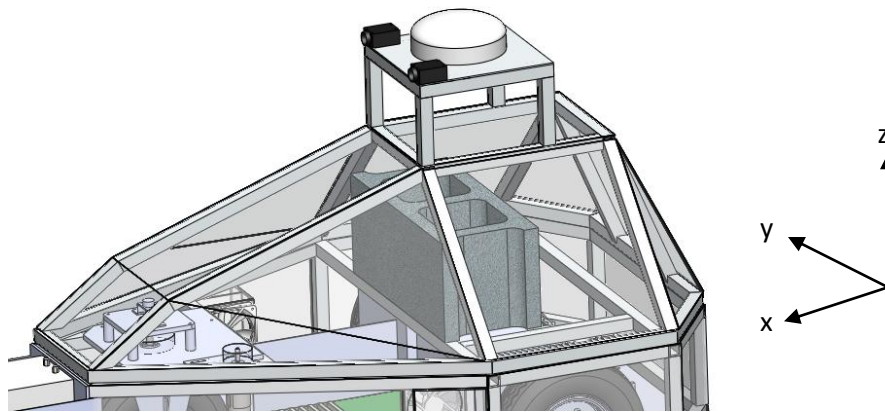


Figure 11 Chassis Cover

Once the base was constructed, it was necessary to redesign the cover such that there was sufficient space for the sensor connected to the front wheel, this change angles the two front bars on the left-hand side of the figure.

3.1.4 Wheels

The IGVC rules state that the vehicle must be driven using direct mechanical contact to the ground such as wheels, tracks, pods, or hovercraft. It was a team specific requirement to design a wheeled unmanned vehicle due to the robustness, maneuverability and simplicity associated with wheels. In the wheel selection process, casters, omni wheels, mecanum wheels, wheelchair wheels and treaded tires have been considered.

A Pugh analysis was completed for the four feasible wheel combination—two driven tires (where “tire” refers to either a wheelchair wheel or a treaded tire) and one caster, or four driven tires—and is presented in Table 1. Mecanum and omni wheels were not considered because these wheel types are not designed to drive on grass.

Criteria (10 best, 0 worst)	Importance	2 Driven Tires 1 Caster	2 Driven Tires 2 Casters	4 Driven Tires
Weight	10	9	8	5
Cost	30	9	8	7
Programmability	5	6	6	8
Turning Radius	20	7	7	9
Power Efficiency	30	9	8	3
Simplicity	15	9	8	6
Innovation	5	3	4	5
Aesthetics	5	4	4	7
Durability	10	6	6	10
	Total	1030	950	820

Table 2: Pugh Analysis of Possible Wheel Configurations

From the diagram, two driven tires and one caster was the most desirable choice. But this still presents the issue of the caster binding. To eliminate this issue, a steerable tire was designed to replace the front caster using a DC motor; for more information on the motor, refer to section 3.1.6 Motors.

The wheels decided to be used are handcart tires with a diameter of 12 inches. The treaded tires and aluminum hub allow for sufficient traction and durability. The pneumatic tires contribute slight suspension to the vehicle.

3.1.5 Front Wheel

Because traditional casters are designed to be used on flat ground they are not the right choice for an off road vehicle. The idea of a steered castor solves the problems that could happen, such as binding and insufficient strength. The idea of a steered caster will work if front wheel angle can be set to match the vehicle turning angle as determined by the speed differential of the two rear drive wheels.

Steering the front wheel eliminates that need to skid steer, which would not have functioned properly. If the vehicle were to turn full speed to the left, the front wheel would not be able to rotate, and would need to slide on the grass for the vehicle to turn. In addition, setting the angle ensures of the front wheel ensures that the position of the wheel is optimized such that the wheel is only rotating and not skidding.

The first version of the front steering mechanism involved the wheel being suspended by shocks so that the robot could absorb the vibrations of driving on uneven ground which would make the camera less effective. Figure 12 shows the design of the original steering wheel mechanism.

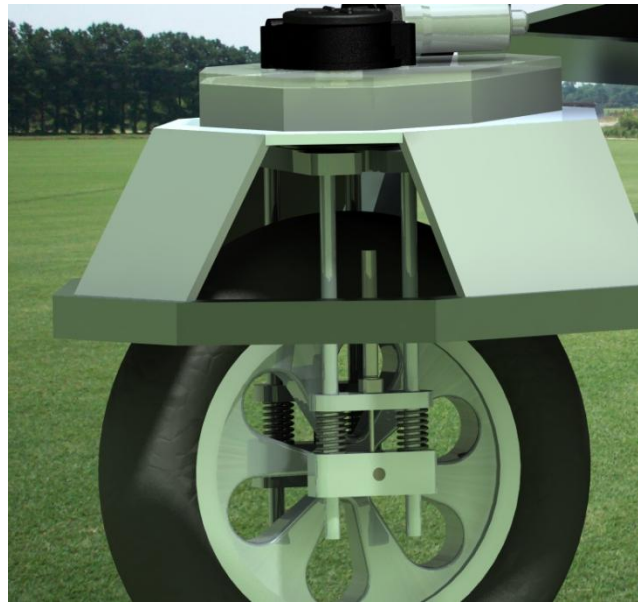


Figure 12: Initial Design of the Front Steering Mechanism

There are two major problems with the original design. The first is that because the front wheel will be suspended by springs and damped by a hydraulic piston, the wheel would be very complex and expensive to build. The shock absorbing was decided to be unnecessary.

The final design for the front wheel mechanism, shown in Figure 13, involves a single wheel pivoting around a single axis and is controlled by a DC motor. The design considerations include the availability of materials, ease of manufacturing, and the ability for the wheel to rotate continuously about its center axis. The support structure is a 0.375" aluminum plate and the steering fork is made out of 1/8" steel plate. The steering fork is a solid welded assembly that supports the wheel as well as controls the angle. There is a continuous potentiometer on the top to measure the angle of the wheel. The reason a continuous potentiometer was chosen is because it allows the wheel simply travel the shortest distance to match the driving angle. A DC geared motor connected to the steering shaft via an H series timing belt and pulley system controls the angle of the steered front wheel.

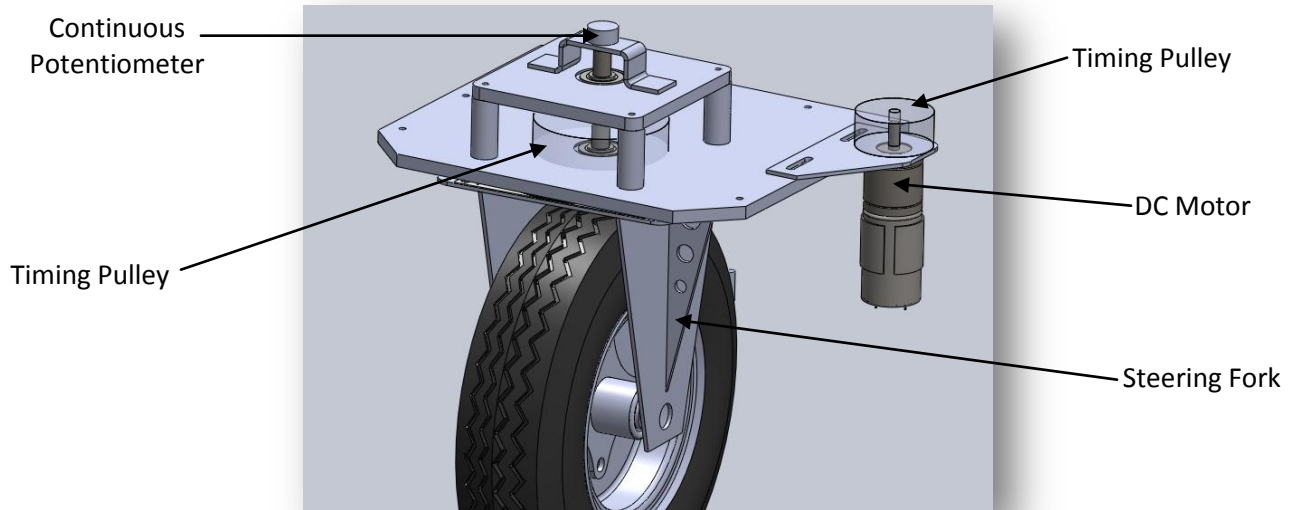


Figure 13: Front wheel steering mechanism

3.1.6 Motors

Motor selection is an important step in the design of unmanned vehicles. A motor needs to be powerful enough to meet the performance requirements of the robot (acceleration, top speed, etc.), but other concerns, such as weight and, most importantly in a battery powered vehicle, efficiency, must also be taken into account. Initial background research was conducted to determine the most suitable type and size of motor to use. This preliminary research focused on the motor systems used by robots in previous IGVCs that performed well in both the autonomous challenge and the overall ranking. Based on the results of this research, it was determined that all of the competitive teams used brushed DC electric motors, due to their good power to weight ratio and simplicity of integration into a design. There were two primary electric motor configuration used: a one-piece motor + gearbox package referred to simply as the ‘wheelchair motor’, and a custom-selected DC motor and separate gearbox.

The wheelchair motor was used by the most teams, including the winner of the autonomous challenge in 2008, Anassa IV⁹, and the overall runner-up in 2008, Wolf¹⁰. Custom electric motor and gearbox combinations were less frequent, although they were used by many competitive teams including Viper¹¹, the runner-up in the 2008 autonomous challenge, and μ CERATOPS¹², the overall winner in 2008.

⁹ (Cardwell, et al. 2009)

¹⁰ (Akroush, et al. 2008)

¹¹ (Givental, et al. 2008)

¹² (Barry, et al. 2008)

The Pugh chart comparing the wheelchair motors to custom motors is presented in Table 3. A gasoline engine is also included in the analysis for completeness.

Preliminary Motor Choice							
Motor Type		Wheelchair Motor		Custom DC Motor		Gas Engine	
Feature	Weight	Raw	Weighted	Raw	Weighted	Raw	Weighted
Power	5	5	25	8	40	10	50
Cost	10	10	100	7	70	2	20
Flexibility	7	5	35	10	70	5	35
Ease of Use	5	10	50	8	40	2	10
Size	5	8	40	8	40	2	10
Ease of Integration	6	10	60	7	42	2	12
Total		48	310	48	302	23	137

Table 3: The Pugh chart comparing the wheelchair motors to custom motors.

Based on the results of this analysis, further research was conducted on the wheelchair motor, as well as options for a custom DC motor solution. The wheelchair motors typically cost approximately \$600 each new, including wheels. Alternatively, one custom DC motor and gearbox combination that also appeared to be viable was the NPC-T64/74 series, sold by the company NPC Robotics. The NPC-T74 was used by the Viper robot¹³, and costs \$324. The NPC-T64 is a slightly less expensive version of the same motor that operates at a lower RPM and is less powerful.

Unfortunately, detailed data on the wheelchair motor proved difficult to find. Dynographs were impossible to obtain, even from vendors selling the motor for robotic applications. Based on information gathered from various IGVC reports¹⁴ and verified by NPC robotics, the general specifications of the wheelchair motor can be seen in table 3.

Wheelchair Motor					
Torque (in-lb)	Amps	RPM	Power-Out (W)	Power In (W)	Speed - 12.5" wheel
120	16	94	133.4620678	192	3.50
60	25	120	85.18855392	300	4.46
0	0	140	0	0	5.21

Table 4: Wheelchair motor specifications (RPM and Torque measurements are at output shaft from gearbox)

¹³ (Givental, et al. 2008)

¹⁴ (Akroush, et al. 2008)

On the other hand, the NPC T-series motors had dnyographs of their nominal operating zones readily available¹⁵. A detailed analysis of the NPC T-64’s specifications can be seen in Table 5. Nominal operating voltage is 24V.

NPC T-64							
Torque (in-lb)	Amps	RPM	HP	Power-Out (W)	Power-In (W)	Speed - 12.5" wheel (mph)	Speed - 8" wheel (mph)
30	8.6	238	0.11	82.03	206.4	8.85	5.66
60	12.5	230	0.22	164.05	300	8.55	5.47
90	16.2	225	0.32	238.62	388.8	8.37	5.35
120	20	218	0.41	305.74	480	8.11	5.19
150	23.5	211	0.52	387.76	564	7.85	5.02
180	27.5	206	0.62	462.33	660	7.66	4.90
210	31.6	200	0.71	529.45	758.4	7.44	4.76
240	35.1	194	0.81	604.02	842.4	7.21	4.62
270	39.2	187	0.89	663.67	940.8	6.95	4.45
300	43.1	181	0.95	708.41	1034.4	6.73	4.31
825	110	0	0	0	0	0	0.00

Table 5: Detailed analysis of the NPC T-64's specifications

The NPC T-64 series DC motors have been selected for the final design based on the information available on its performance, its lower cost, and due to the fact that the wheelchair motor, at its maximum output speed and with a 12” wheel, would only be able to operate at ~4 MPH. The T-64 also is more powerful than the wheelchair motor, which allows for more flexibility in the design of the robot as weight would be less of a concern.

After analysis of the performance data for the T-64, an initial design required an 8” wheel to be directly mounted to the output shaft from the motor. This ensured that the motor would be operating in its peak efficiency range while maintaining a wheel speed of 3-5MPH, the typical speeds that the robot was expected to travel at. However, after a preliminary design review, the decision was made to utilize a simple chain drive with a 30 tooth and 48 tooth sprocket to create a 1.6:1 reduction to run 12” wheels and achieve better ground clearance. At the same time this change improved performance and power efficiency in the desired speed ranges. Key theoretical performance data for the robot utilizing this drive train can be seen in Table 6: NPC T-64 performance. All calculations assumed a robot weight of 400lbs, which is approximately twice the expected weight of the robot, to represent worst case performance. Additional performance information can be found in Appendix D: Motor Performance Data.

¹⁵ (NPC Robotics 2009)

Specification	Value
Worst Case Continuous Power Draw	640W*2
Nominal Power Draw	300W*2
Max Wheel Speed	5.36MPH
Average Efficiency @ Nominal Speeds	63.9
Maximum Incline @ 1MPH	25*
Speed at 15% gradient	4.21MPH
0-5MPH time	1.38s

Table 6: NPC T-64 performance

3.1.7 Motor Controllers

In order to drive the high power DC motors utilized by the robot as well as effectively control their speed, specialized motor controllers are needed. Motor controllers convert the low-power signals produced by computers/microcontrollers into high-power signals (typically Pulse Width Modulation) that can power the motors. Motor controllers vary in complexity and cost. Lower cost (per power level) controllers are simple H-Bridge plus MOSFET controllers that take a low-power PWM signal and raw power as inputs and convert it to a high-power PWM output. More expensive controllers have more advanced features, such as the ability to be controlled by serial commands, over-voltage and current-limiting protection, and coast vs. brake control when throttle is removed {{9 Anonymous 2009}}. More advanced controllers can implement closed loop speed control and include features such as regenerative braking¹⁶.

In the initial design the motor controller was going to be an all-inclusive unit. The input would be speed in MPH for each motor, and the motor controller set and maintained this speed using encoder feedback and programmed constants for the wheel parameters such as wheel diameter and encoder cycles per revolution. This ability is desirable because it reduces the processing load on the main computer and would ensure reliable speed control. In addition, the motor controller output needs to be 24V, and capable of supporting 30-60A continuous load, as these are the expected power requirements of the motors driving the vehicle. Based on the previous IGVC robots, the Roboteq AX series of motor controllers was identified as a potential candidate. Viper and Wolf both used AX series controllers¹⁷, as

¹⁶ (Roboteq INC 2009)

¹⁷ (Givental, et al. 2008)

well as several other IGVC robots. These controllers range between \$300-600 depending on features, making them reasonably affordable given the team’s budget constraints and their advanced features¹⁸.

However, as described in Section 2.3, the final design of Prometheus uses a National Instruments cRIO to perform low level tasks. This makes it possible to implement the desired features such as closed loop speed control on the cRIO without increasing computational load on the main computer. With this in mind, a simpler motor controller, which has only basic features such as current-limiting and voltage protection became a viable solution. The team explored two options: the Roboteq AX1500, a stripped down AX series controller with a minimal feature set {{12 Roboteq INC 2009}}, and the Victor 883, a basic controller commonly used in robotic applications {{13 IFI-Robotics INC 2009}}. A comparison of their features can be seen in Table 7.

Motor Controllers		
Controller	AX1500SC	Victor 883
Channels	1	1
Voltage	24V	24V
Continous Current	40A	60A
Surge Current	250A	200A
Input	Serial+PWM	PWM
Regenerative Breaking	Yes	No
Current Limiting	Yes	No
Price	\$275	\$150

Table 7: Motor Controller Data

There is a third option that was not explored at the time; the Texas Instruments (TI) Jaguar, which is used in combination with the cRIO. The original version of the Jaguar was only available in a 12V version, making it incompatible with Prometheus’ 24V motors. During the later stages of the design process, TI released a 24V version of the controller. The Jaguar motor controller offers many of the features of expensive motor controllers such as the Roboteq AX 2xxx series controllers, but at a significantly reduced cost of around \$80. They offer closed loop speed control, software and hardware controlled brake/cost operation, and direct voltage/pwm control. This makes the Jaguar a more attractive option than either the AX1500 or the Victor 883, and the final design of Prometheus employs Jaguars to drive the T-64 motors.

¹⁸ (Roboteq INC 2009)

3.2. Development

Based on the design decision discussed earlier in the report, CAD models of the final design have been constructed. The mathematical analysis of the vehicle confirms that it meets all of the required design specifications. To realize the design, the TIG welded chassis and frame initialize the construction of the chassis. CNC machined precision parts, which were constructed using generic machine shop equipment and power tools, are used on the chassis, precision parts such as the sprockets.

3.2.2 Chassis Frame

The chassis frame was built in three steps: cut aluminum tubes to the correct length and angles on both ends, setup the welding bench and weld sub-sections, and weld the complete chassis frame. Dimensional correctness is checked between each of these steps to ensure its accuracy.

To ensure the dimensional correctness of the welded chassis, a 1:1 scale drawing of the chassis frame is printed using the plotter in the Design Studio. Cut aluminum tubes were laid on the drawing to check its length and angles on two ends. After sections being welded, they were also laid on the drawing to check its accuracy.

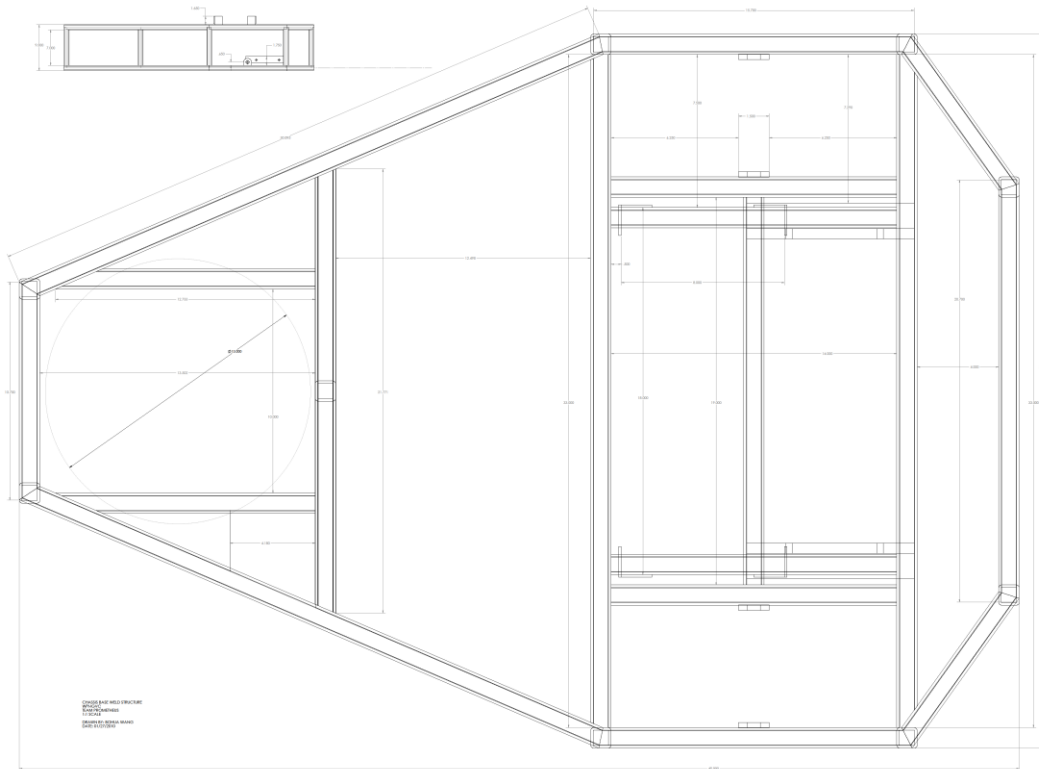


Figure 14: A scaled down view of the real size drawing of the chassis frame

The chassis frame is welded with a metal inert gas welder (MIG) welder. A welding setup is shown below in which C-clamps are used to fix the aluminum tubes. The chassis frame was welded in sub-sections first which are the rectangular section around the rear wheels, the trapezoidal section at the front and the smaller trapezoidal section at the back. Machining squares were used when welding the 90° angles, joints of other angles were compared directly to the real size drawing during the welding setup. Other than aluminum tubes, wheel tabs and motor brackets were also welded onto the chassis frame.



Figure 15: Chassis frame building in progress

3.2.3 Cover

The cover was constructed in the same manner as the base, requiring a more complicated jig system when being welded due to the non-perpendicular and non-planar welds that needed to be made. Using spare metal stock and clamps, sufficient welding jigs were shaped such that the bars would retain their shape relative to each other while they were welded together.

Three temporary legs were tacked to the top rectangle of the cover to ensure that the rectangle remained parallel to the ground while the vehicle is driving on a flat surface. The rectangle and its legs were centered in the lower frame of the cover and the supports were placed in their individual locations. Each joint was tacked and before they were then welded completely. Once the cover was finished being welded, the temporary legs were detached from the top rectangle.

Using hinges and air shocks, the cover was attached to the base of the chassis. Weather stripping was attached to the underside of the cover to ensure that it was weatherproof, and latches were attached to the base and cover such that the cover could be firmly closed during the competition.

3.2.4 Front Wheel

After deciding on the best wheel size, the CAD model was updated and the parts were adapted to make them fit to the new wheel. The material we would need to make the entire design is several lengths of steel shaft along with an aluminum plate and a steel sheet. The aluminum top pieces, along with the motor bracket and steering fork piece, were CNC machined. The steel standoffs shown were made on the lathe. One error that presented itself was that the hole for the wheel axis was misaligned by 0.75", but because a thick plate was used, the plate was relocated slightly to accommodate for this. This was done because each bolt can hold up to 3klbs a piece before breaking and thus six bolts will be more than able to mount the front wheel mechanism to the frame.

3.2.5 Rear Wheels and Power Train

To complete the power train, the 48-tooth sprockets need to be attached concentrically to the rear wheels, and the 30-tooth sprockets need to be mounted concentrically to the motor shaft. The sprockets were made in house to match the mounting hole patterns on the motor shaft and the wheel. For the 48-tooth sprocket, the concentricity issue was solved by mounting the sprocket directly on the wheel bearing assembly with a tight fit center hole. For the 30-tooth sprocket, a dowel pin going through the center hole of the sprocket and the motor shaft was used to align the two parts.

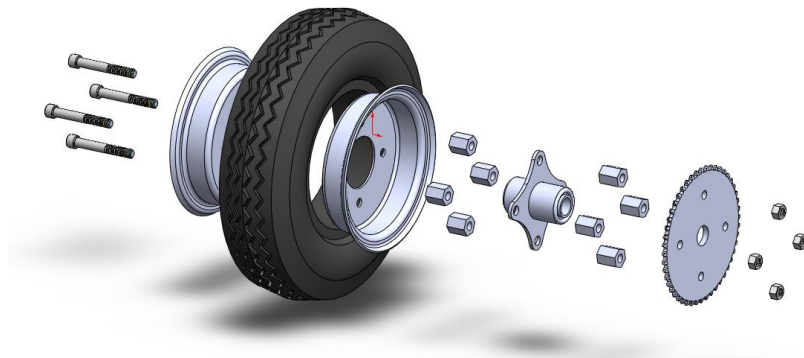


Figure 16: Exploded view of the wheel-sprocket assembly

As shown in Figure 16, the rear wheel-sprocket assembly consists of the original wheel assembly which includes the tire, wheel hub and the bearing mount, and the add-ons including the hex spacers,

sprocket, bolts and nuts. Two sets of four precision-machined hex spacer are mounted according to the original hole patterns of the wheel hub. The length of the spacers needs to be precisely the same in order for the sprocket side surface to be parallel with the wheel. Therefore, they are milled down with an end mill at the same time.



Figure 17: Cutting sprocket tooth contour

The sprockets are made out of T6061 aluminum plate with a thickness of 1/4 inch. After the tooth contour was cut, chamfers at the tip of the teeth were added with manual lath.

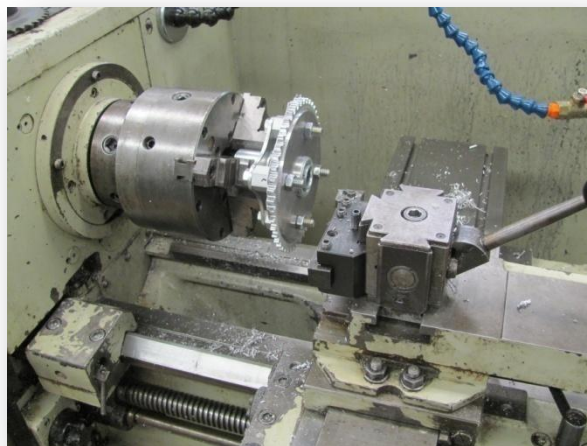


Figure 18: Cutting chamfers using manual lath

3.2.6 Sprocket Reduction

With the motors spinning at their maximum possible angular velocity, 238 rpm, the linear velocity of the back wheels would be 8.85 miles per hour, which is greater than the design requirements 5 miles per hour (140 rpm for 12" wheels). To reduce the speed, a chain and two sprockets were used to transmit the torque from the motors to the back wheels. Also, the angular velocity of the motor was reduced such that the motor operates within its efficiency zone (225 rpm). To calculate the necessary reduction, the following equation was used:

$$e = \frac{N_{Driver}}{N_{Driven}} = \frac{n_{out}}{n_{in}}$$

Where N_{Driver} is the number of teeth on the sprocket attached to the motor, N_{Driven} is the number of teeth on the sprocket attached to the wheel axle, n_{out} is the angular velocity of the wheel axle, and n_{in} is the angular velocity of the motor.

$$e = \frac{225 \text{ rpm}}{140 \text{ rpm}}$$

$$e = 1.61$$

The sprockets need a relation of 0.62 to one, and if the driving sprocket has 30 teeth, the driven sprocket can be calculated.

$$e = \frac{N_{Driver}}{N_{Driven}}$$
$$0.62 = \frac{48}{N_{Driven}}$$
$$N_{Driven} = 29.8$$

Since the vehicle needs to drive as close to 5 miles per hour as possible, the number of teeth was rounded to the nearest integer value of teeth, which is 30 teeth. This is such that the vehicle has the potential to operate at top speed. The vehicle can travel at maximum of 5.07 miles per hour.

3.2.6 Motor Controller

Since the vehicle's front steered wheel does not require a high current motor controller (maximum current for front steered wheel is 13 amps), a simple H-bridge motor controller using MOSFETs (Metal-Oxide-Semiconductor Field-Effect-Transistors) can be designed for controlling the front wheel angle. The motor controller takes in two digital signals, a direction bit and a PWM signal. It then uses digital logic to convert these signals into the signals required to drive a bridge of MOSFETs and provide power to the steering motor.

3.2.6.1 Bridge Design

The design for this circuit can be seen below in Figure 19. The motor controller's power bridge uses six MOSFETS. The main bridge is composed of four MOSFETS. Two P channel MOSFETs for current sourcing and two N channel MOSFETs for current sinking. The two P channel MOSFETs are then driven by two N channels. The P channels chosen are the IRF9540s which can support up to 100V and 19 amps. The N channels are the IRFZ44s which can support 60V and 50 amps. The P channel FETs require switching input voltages between 0V and -12V. In order to achieve this, a voltage divider composed of two 1K resistors is used with the center of the divider attached to the gate of the P channel MOSFETs and the connection to ground controlled by N channel MOSFETs. To protect the MOSFETs from dangerous voltages that could be generated by back EMF from the motor, zener diodes are placed across the drain and source pins. These diodes are included in the MOSFET packaging. Zener diodes are used because the reverse voltage will break down once it gets above the diode voltage threshold allowing the energy to be conducted to ground without damaging the MOSFETs.

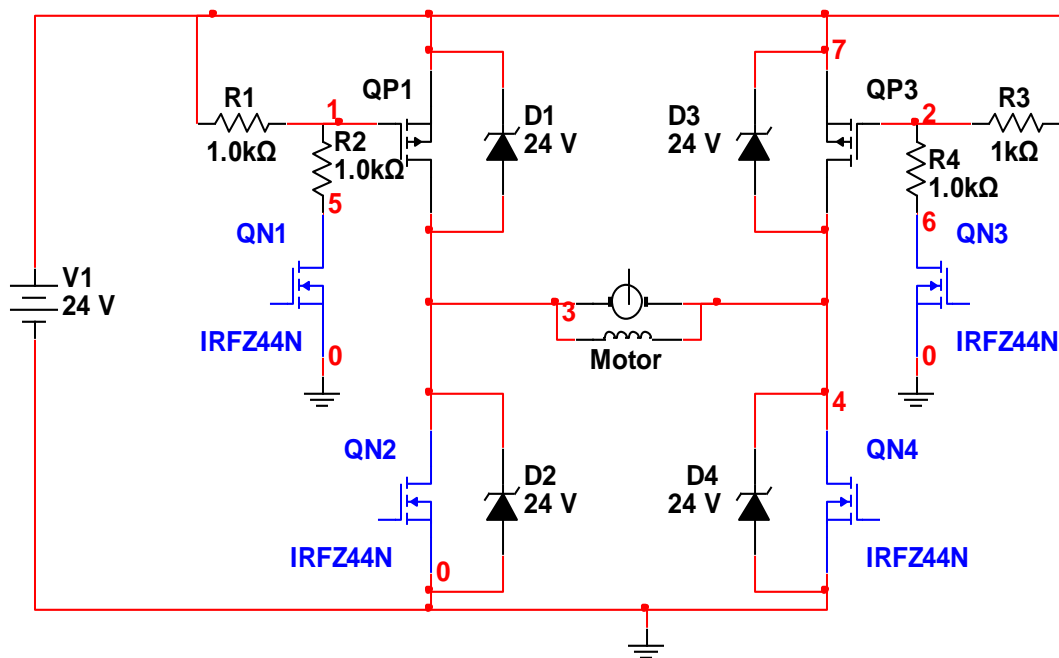


Figure 19: Bridge Circuit Schematic

3.2.6.2 Logic Design

The control logic for the motor controller is responsible for driving the MOSFETs. It takes in two 5V digital inputs for PWM and direction and converts it into four digital signals that drive the MOSFET bridge. The truth table for the four different bridge states can be seen in Table 8 where Q1-Q4

represent the MOSFET state. On = 1 and off = 0. The MOSFET numbers are respective to QN1-QN4 in Figure 19.

Direction	PWM	Q1	Q2	Q3	Q4
0	0	1	0	0	0
0	1	1	0	0	1
1	0	0	0	1	0
1	1	0	1	1	0

Table 8: Motor Controller Truth Table

While the bridge is switching, the current changes directions through the motor. Because of this, if the switching is not timed appropriately, there can be a small duration of time during the state transition where both of the MOSFETs on one side of the bridge are turned on at the same time. This causes a temporary short circuit in the system that over time will damage the MOSFETs. This phenomenon is called shoot-through current(Maxim Integrated Products n.d.). In order to prevent this from happening in the motor controller, the logic gates are used in such a way as to take advantage of their propagation delays. The motor controller logic diagram can be seen in Figure 20.

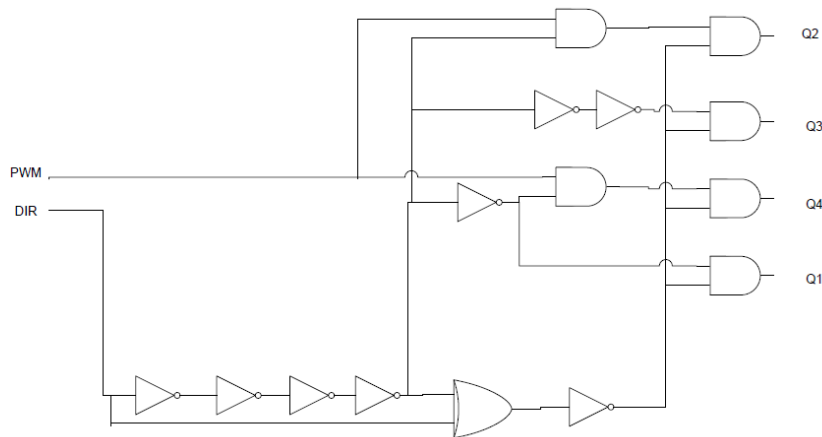


Figure 20: Motor Controller Logic

There is potential for shoot-through current whenever the motor changes direction. For this reason, whenever the direction bit changes, all of the output logic is turned off. The outputs then have time to completely change state before they are turned back on again simultaneously.

3.2.6.3 Board Design

The motor controller circuit was first built on a breadboard in order to test its operation. The bridge was built and tested first and then the driving logic was built and tested separately. A 5V voltage regulator was also used to provide power to the logic ICs. After being built and tested separately these three parts to the system were connected and tested together.

In order to be used on the vehicle, the final controller needed to be soldered onto a board. In order to make this board robust and easy to use, specific features were utilized. Sockets were used for all of the logic IC's so that they could be easily swapped if needed. Screw terminals were used for all connections going into and out of the board. The screw terminals make it difficult for the wires to come loose during vehicle operation. Heat sinks were used on the four main bridge MOSFETs as well as on the 5V voltage regulator to allow for heat dissipation. In addition to the heat sinks, a fan was placed above the MOSFET bridge to aid in cooling. The final board can be seen in Figure 21.

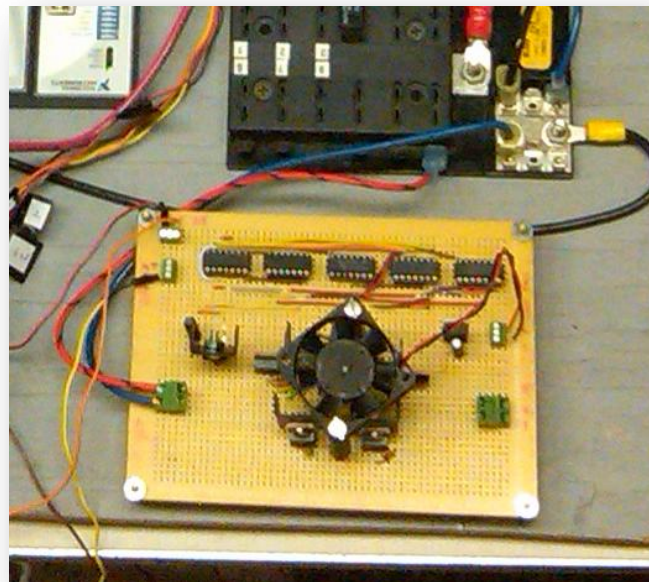


Figure 21: Final Motor Controller Board

3.3 Design Analysis

With the design of the vehicle complete, analysis is now necessary to ensure that the vehicle operates as intended. In Figure 22, the top view of the base compartment is shown. In relation to this view, the vehicle's center of gravity is labeled and the vehicle's coordinate system is also shown, where x is the distance from the back of the vehicle to the front, y is the distance from the right of the vehicle to the left, and z is the distance from the bottom of the vehicle to the top.

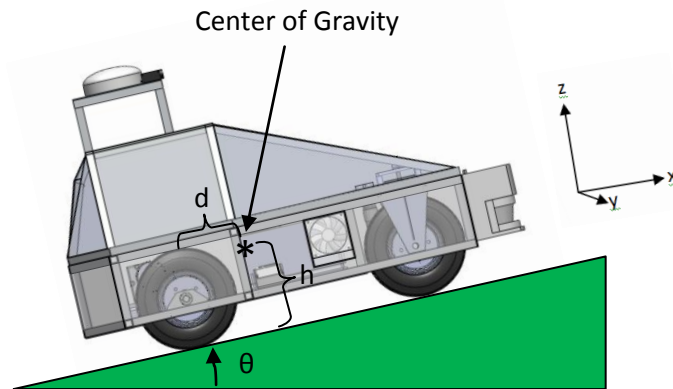


Figure 22: Vehicle Center of Gravity, Coordinate System, and the Angled Incline

The height from the ground to the center of gravity is h , the distance from the rear axle to the center of gravity is d , and the angle of the incline is θ .

3.3.1 Center of Gravity

The maximum incline angle of 15%, or 8.53 degrees, presents two design challenges. First, the center of gravity must remain within the polygon of contact, shown in Figure 23, so that the vehicle will not tip over as it climbs the incline. And the second challenge is ensuring that the coefficient of friction on the back wheels is sufficient such that the vehicle can back up the incline.

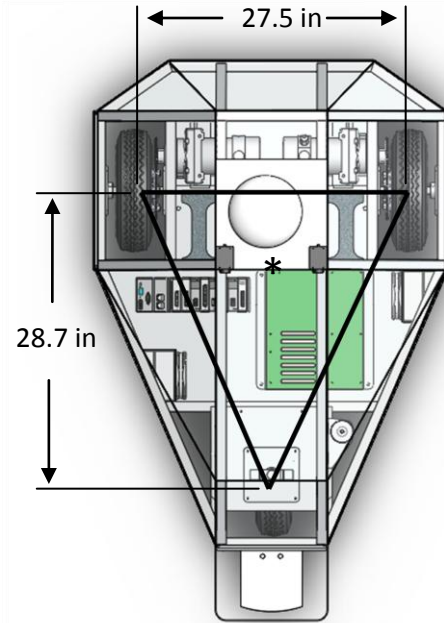


Figure 23: Polygon of Contact

Based off of the CAD model, the CG is 11.01 in from the ground, the distance of the center of gravity from the back of the robot was calculated using the following equation with the variables set in Figure 22:

$$d = h * \tan(\theta)$$

$$d = 11.01in * \tan(8.63^\circ)$$

$$d = 1.68 \text{ inches}$$

Based on this calculation it is noted that the CG should be no less than 1.68 inches towards the front wheel from the points of contact on the back wheels. Assuming that CG is 11.01" above the ground, 1.68 inches forward from the back wheels, and centered on the width of the vehicle such that the vehicle will be equally weighted on the left and right sides, the vehicle will be able to climb a 15% incline.

After designing and constructing the vehicle, the center of gravity was calculated to be 11.01 inches off of the ground, 7.89 inches forward from the back wheels, and 13.74 inches from the side, as shown in Figure 24.

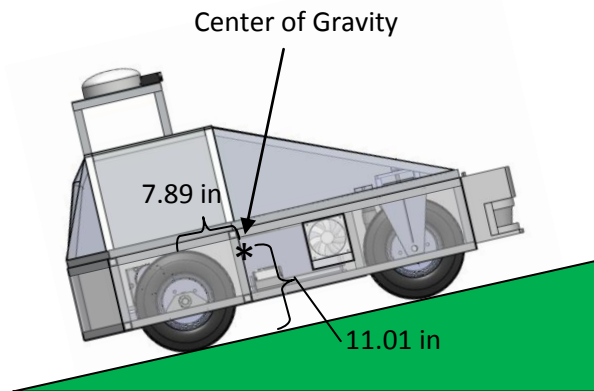


Figure 24: Actual Center of Gravity of Vehicle

With this center of gravity, the previous equation can be used once more to calculate the actual angle that the vehicle could climb without tipping over.

$$d = h * \tan \theta$$

$$\tan \theta = \frac{7.89in}{11.01in}$$

$$\theta = 35.6^\circ$$

The vehicle could climb a 35.6 degree (or 58%) incline, which is less than the 56.8 (or 84%) incline it could drive up with the wheel's coefficient of friction.

3.3.2 Coefficient of Friction & Climb Angle

It is also important to consider the 15%, or 8.53 degree, incline while choosing the wheels. The coefficient of friction for the back wheels required by the vehicle to reverse up the 15% incline is determined directly from the angle of the incline, as shown in the equation below, also using the variables defined in Figure 22:

$$\mu = \tan \theta$$

$$\mu = \tan (8.63^\circ)$$

$$\mu = 0.145$$

At a minimum, the rear wheels need to have a coefficient of friction of 0.066. This is more than sufficient since the vehicle's wheels will be driving on grass where the average coefficient of friction

between grass and rubber is 0.35. The ideal coefficient of friction between wet grass and rubber is 0.2, which is still greater than the minimum required coefficient of friction.¹⁹

The coefficient of friction for the wheels was measured using a force gage to measure the weight of the wheels and the force needed to cause wheel slip. The weight of each wheel was measured to be 6.75 pounds, and the wheel slip force (which is equal to the force of friction) was measured to be 13 pounds on dry grass for the combined weight of three wheels. Using the following equation, the coefficient of friction for the wheels on dry grass was calculated.

$$\mu = \frac{F_f}{N}$$

$$\mu = \frac{13}{6.75 * 3}$$

$$\mu = 0.64$$

This measured coefficient of friction is greater than the required 0.152 for the 15% incline, which allows the vehicle more than enough traction to reverse up the incline.

Theoretically, if the same equation used to calculate the minimum coefficient of friction is used, the maximum climb angle for the vehicle can be calculated.

$$\mu = \tan \theta$$

$$0.64 = \tan \theta$$

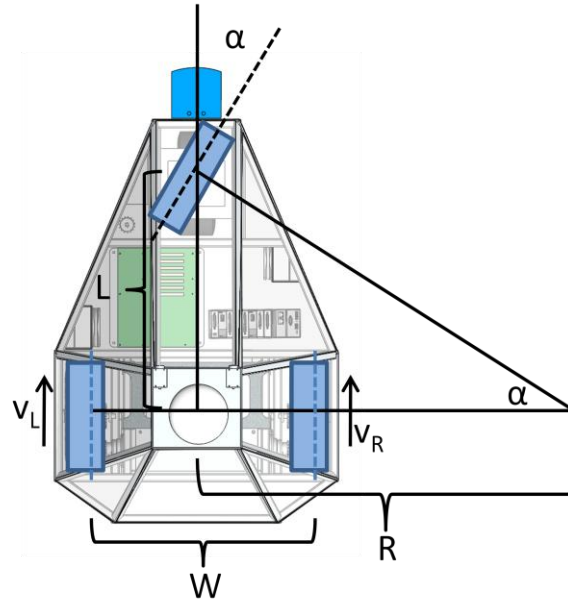
$$\theta = 56.8^\circ$$

The maximum reverse angle of climb is much greater than the required 15%, this angle would be the equivalent of a 65% incline.

3.3.3 Front Wheel turning angle based on rear wheel speed differential

The coupling of the front steered wheel with the rear differential drive can add to turning accuracy and avoid caster binding situation. However, this also created the redundancy in the turning angle as set by the front wheel and the rear wheel speed difference. Therefore, the turning angle of the front wheel must be aligned with the turning angle as set by the speed difference of the rear wheels. The following equations can be used to calculate the front wheel turning angle α as a function of the rear wheel speeds v_L and v_R .

¹⁹ (Cenek, Jamieson and McLarin n.d.)



Parameter	Variable	Value
Left Wheel Speed	v_L	
Right Wheel Speed	v_R	
Front Wheel Turning Angle	α	
Wheel Base Width	W	27 in
Wheel Base Length	L	28.625 in
Turning Radius	R	

Figure 25: A schematic showing the vehicle making a right turn along with a table defining the variables.

Using the fact that the angular velocities of the two rear wheels about the turning center are the same, and by rearranging the equation to solve for R in term of the independent variables v_L and v_R , one can obtain.

$$\frac{v_L}{R + \frac{1}{2}W} = \frac{v_R}{R - \frac{1}{2}W} \Rightarrow R = \frac{W(v_L + v_R)}{2(v_L - v_R)}$$

By expressing α in terms of L and R , and then substitute R with the equation derived above, the front wheel turning angle α is now a function of the rear wheel speeds v_L and v_R .

$$\alpha = \tan^{-1}\left(\frac{L}{R}\right) = \tan^{-1}\left[\frac{2L(v_L - v_R)}{W(v_L + v_R)}\right]$$

Therefore, the previous equation can be used to calculate the angle of the front wheel knowing the rear wheel speeds. This way, a smooth driving of the robot can be achieved.

3.3.4 Turning Dynamics

Two factors limiting the turning capability of a vehicle are wheel slipping and tipping over. Under lateral acceleration, the wheel will start to slip sideways before the center of gravity can leave the polygon of contact. Since lateral acceleration reduces once the wheel starts to slip, the vehicle won't tip over on level ground due to lateral acceleration. Therefore, the turning radius of the vehicle is only limited by the friction coefficient of the wheels.

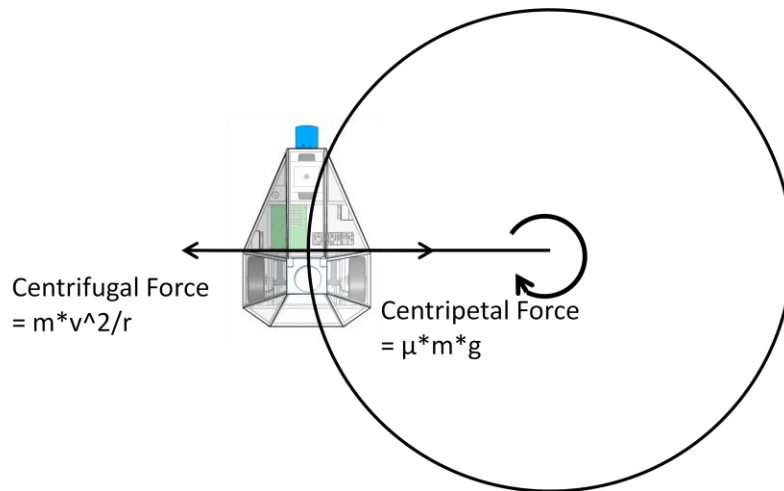


Figure 26: Minimum turning radius diagram

Figure 26 depicts the situation in which the vehicle is driving at its maximum speed ($v=5\text{mph}$) with a turning radius such that the tire starts to slip. The centripetal force is provided by the friction of the tire pulling vehicle toward the center of the turning circle and this force is limited by the friction coefficient of the tire (μ). The centrifugal force is caused by vehicle's tendency to maintain its momentum and is always equal and opposite to the centripetal force, its expression is derived from Newton's second law of motion $F=ma$. By equating the two forces:

$$\frac{m * v^2}{r} = \mu * m * g$$

$$r = \frac{v^2}{\mu * g} = \frac{(2.2 \frac{m}{s})^2}{0.64 * 9.81 m/s^2} = 0.77m$$

This turning radius is approximately equal to the half the width of the vehicle. A turning radius this small means that the vehicle should be able to do sharp turns under maximum speed in the competition course without concerning about tipping over or tire slipping.

3.3.5 Acceleration Performance

The vehicle acceleration curve can mathematically be simulated to verify that the vehicle can indeed reach the desired maximum speed of 2.2m/s. The simulation is based on Newton's second law of motion $F=ma$ where F equals to the pulling force of the motor minus the drag as the result of the rolling friction, m is the mass of the vehicle and a is the acceleration which is the derivative of the velocity. This equation is used to derive the vehicle speed v as a function of time with the initial condition of $v(0)=0$.

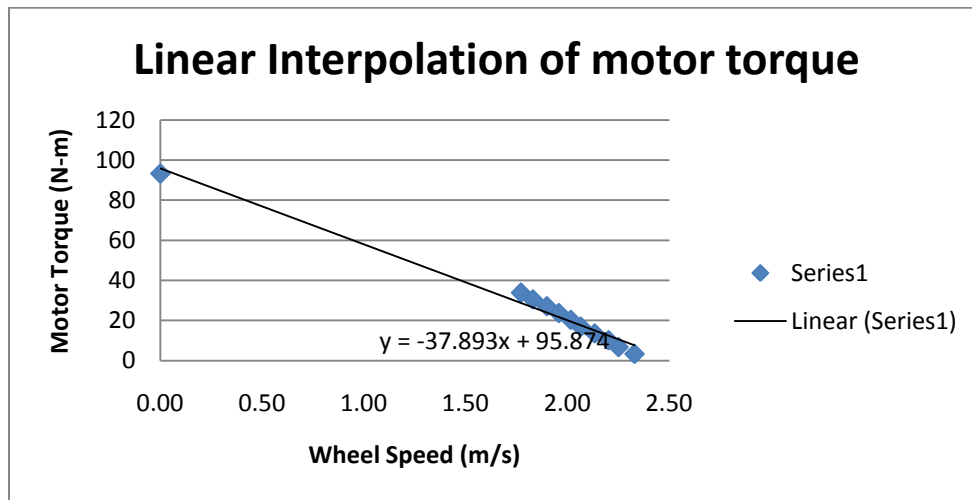


Figure 27: Interpolation of T_{motor} as a function of Velocity based on the motor datasheet

As shown by the data points in Figure 27, motor torque T_{motor} decreases as the wheel speed increases. A linear relationship was assumed so that T_{motor} can be expressed as a function of wheel speed v .

$$T_{motor} = -38 * v + 96$$

The driving force of the vehicle produced by the motor can now be calculated by using the sprocket transmission ratio, and assuming a power train efficiency of 90%.

$$F = \frac{T_{motor}}{R_{wheel} \left(\frac{30}{48}\right)} * \eta_{powertrain} = \frac{T_{motor}}{0.159m * \left(\frac{30}{48}\right)} * 0.9 = T_{motor} * 9.1[N]$$

Using Equations X and Y, one can obtain

$$F = -345.8 * v + 873.6$$

The drag is assumed to be solely resulted from the rolling friction of the vehicle with a friction coefficient $\mu = 0.06$ typical for vehicle tire on grass.

$$f = \mu mg = .06 * 68kg * \frac{9.81N}{kg} = 40N$$

Thus the left hand side in $F=ma$ becomes

$$F - f = -345.8 * v + 833.6$$

And the right hand side is,

$$m = 150lb = 68kg$$

$$a = \frac{dv}{dt}$$

Therefore the resulting equation:

$$-345.8 * v + 833.6 = 68 * \frac{dv}{dt}$$

This is in the form of a first order ordinary differential equation:

$$\frac{dv}{dt} = -5.09v + 12.25$$

With initial condition $v(t=0)=0$, the solution is:

$$v = -0.2 * e^{-5.09*t+2.49} + 2.41$$

The solution is plotted in Figure 28. The simulated maximum speed of the vehicle is 2.4m/s which is slightly higher than the max speed of 2.2m/s or 5 mph as set by the IGVC rules. The actual acceleration performance of the vehicle is measured on the completed vehicle with velocity feedback from the optical encoder attached to the drive motors.

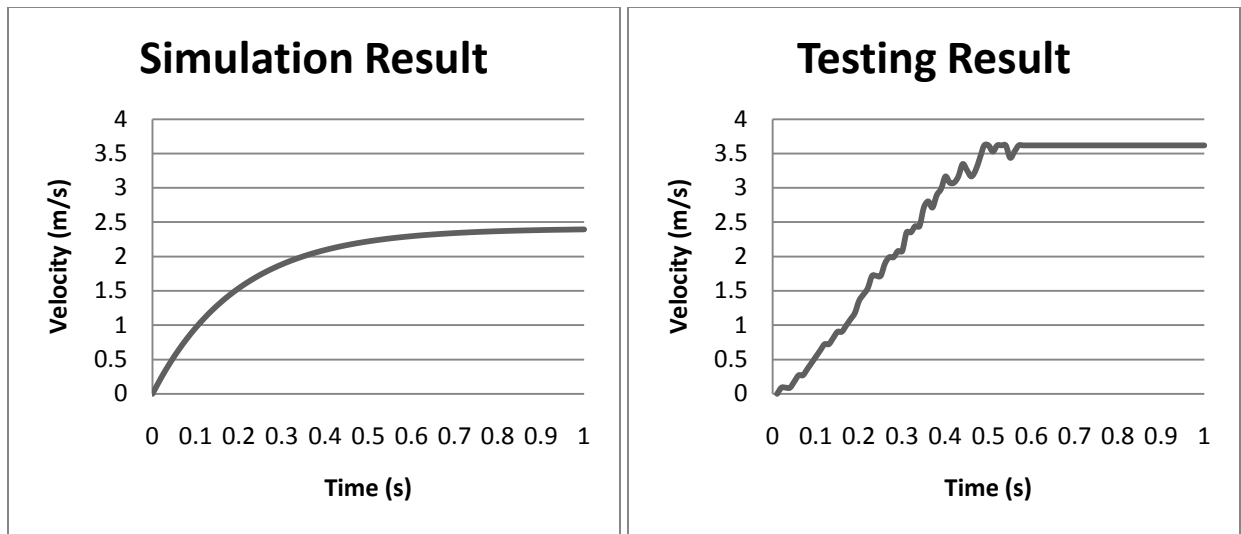


Figure 28: Vehicle acceleration performance as tested on leveled grass.

As shown, there is a difference between the simulation and testing result. This will be discussed further in the 3.5.3 Rear Wheels and Power train.

3.4 Results

The complete chassis, now fully constructed, is shown in Figure 29.



Figure 29: Completed Vehicle

3.4.1 Chassis Frame

With the completion of the chassis frame, the following values were recorded for the complete vehicle:

- Vehicle dimensions: 47 in x 35 in x 28 in
- Vehicle weight: 212.8 lb
- Wheel diameter: 12.5 in
- Center of gravity: 11.01 in from ground, 7.89 in forward from back wheels, and centered at 13.74 in
- Ground Clearance: 4.05 in

These resulting measurements meet the IGVC specifications along with the team specifications.

3.4.2 Front Wheel

The front steered wheel achieves continuous rotation, allowing for the wheel to spin in any desired direction. The algorithm depicted in Figure 25 correctly calculates the angle of the wheel such that it can be driven with minimal skidding.

3.4.3 Rear Wheels and Power train

The following values were recorded for the rear wheels:

- Maximum speed: 5 mph
- Maximum climb angle: 35.6 degrees
- Acceleration time from 0-5 mph: 0.6 seconds

The climb angle is overly sufficient, the maximum speed is met, and the vehicle accelerates quickly enough to maneuver easily through the IGVC obstacle course.

Also, the actual acceleration of the motors, shown in Figure 28, did not coincide with the theoretical acceleration. The top speed of the vehicle is actually 3.6m/s instead of 2.4m/s as simulated. The motors reach top speed faster than anticipated, due to possible calculation errors that are discussed in section 3.5.3.

3.4.4 Motor Controller

Using a bench top power supply with electrical current sensing, it was determined that the custom motor controller was able to successfully drive a 24V DC motor at currents below 2 amps. The logic implemented successfully converted 5V digital PWM and direction signals into signals to drive the H-Bridge properly. The logic also successfully prevented shoot-through current at state transitions by

using propagation delays to turn off all MOSFETs in the H-Bridge before turning them back on after the state transition occurred. This is shown in Figure 30.

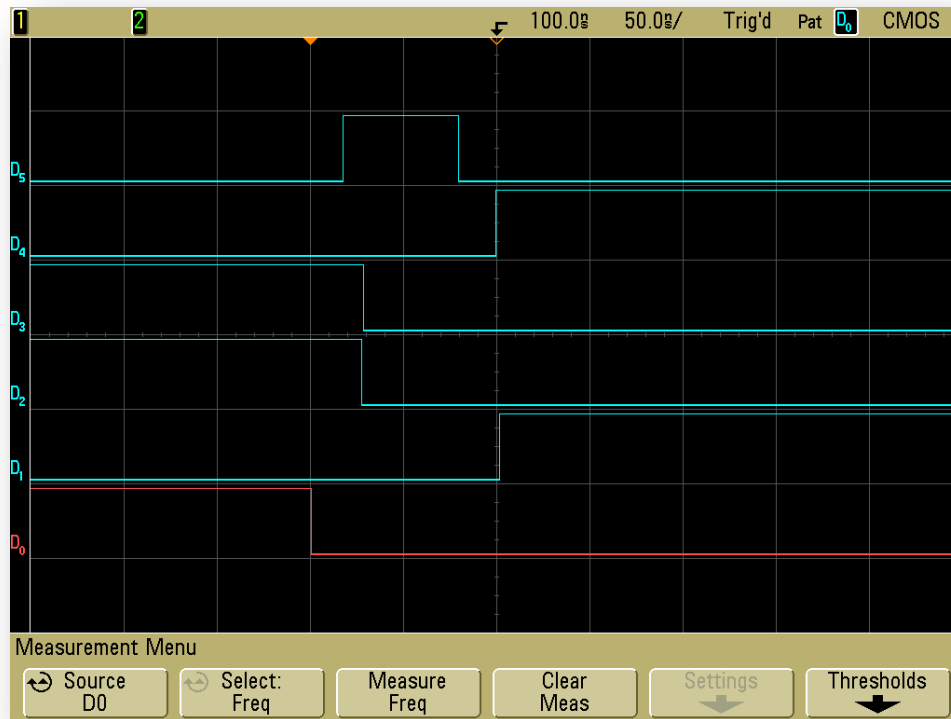


Figure 30: State Transition Logic

In this diagram D0 is the change in state. D1, D2, D3, and D4 are the four MOSFET switches Q1, Q2, Q3, and Q4. The MOSFETs corresponding to Q1-Q4 can be seen as QN1-QN4 in Figure 19. D5 is the signal that turns off the outputs. It can be seen that all four of the outputs transition to the low state when the state changes. They are then allowed approximately 50 ns to change state before all being turned back on simultaneously.

3.5 Discussion

3.5.1 Chassis Frame

The constructed chassis can support the weight of one average-sized person without observable deflection. During the welding process, a few bars developed small amount of bending near the welding point due to the high temperature needed to melt the joint, but these deflections are minor and do not affect the functionality of the chassis frame. And for a welded structure as large as this vehicle, it would be surprising not to have deflections.

3.5.2 Front Wheel

The aluminum support plates, after being cleaned, were attached to the chassis. The pieces of the front steering fork needed to be welded together. Because the steel was only a 1/8" all of the welds couldn't be done at once because it would cause excessive warping of the pieces. The welds had to be done in 1 inch sections. After all the welding was finished the steering fork was slightly warped but not enough to affect the function of the piece. Figure 31 depicts the steered front wheel.

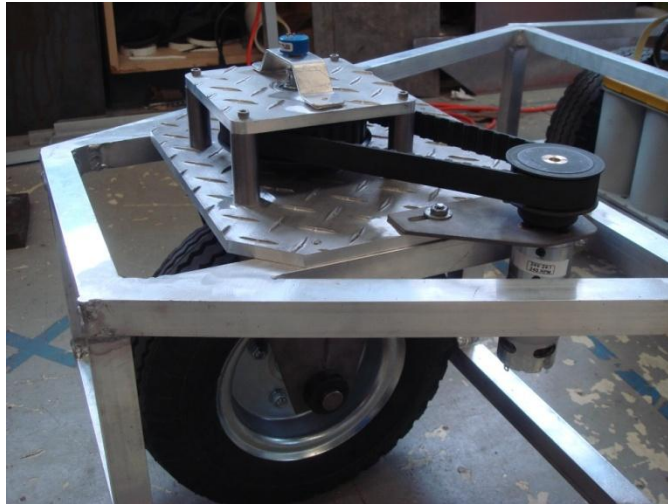


Figure 31: Front wheel steering system completed

3.5.3 Rear Wheels and Power train

As shown in Figure 32, all the parts were assembled successfully. Two sprockets were aligned on the same surface by using a custom made spacer at the inner end of the wheel axle to adjust the position of the wheel assembly. A portion of a #35 chain stock was separated by grinding off the tip of a linking pin and then pressing out the pin. The chain was wrapped on the sprockets and joined by inserting a connecting link. In the initial test drive, the power train runs smoothly.



Figure 32: Sprocket and chain completed

The calculated acceleration and actual acceleration of the motors, shown in Figure 28, displayed a discrepancy between the two values. This could be a result of the assumed linear relationship between motor torque and wheel velocity. It could be caused by the assumptions made for the dragging force in the simulation or by the motor controllers we used. However, both graphs show that the vehicle can reach its top speed in about 0.6 seconds. Future work could be done in further inspecting this discrepancy and constructing a more coherent mathematical model.

3.5.4 Motor Controller

During testing using the vehicle's steer wheel motor, heating issues were discovered in the N-Channel MOSFETs in the bottom two legs of the bridge. The heating issues are believed to be caused by the existing switching frequency being used and the increased electrical current. To combat the heating issues, heat sinks with thermal paste were attached to all of the MOSFETs and a fan was placed above them. However, the MOSFETs were still damaged due to high junction temperatures.

In the interest of time, the team decided to stop working on the custom motor controller and purchased another Jaguar motor controller to control the orientation of the front wheel. This decision was made in order to increase the rate of progress on the vehicle in order to allow more time for testing once the vehicle was completed. The MOSFETs were being driven at a switching frequency of 10 KHz and this is believed to be the reason for the overheating. The team believes that this switching frequency could be reduced down to as low as 100 Hz with negligible effect on the quality of the output.

The team also believes this significant decrease in switching frequency would eliminate the thermal problems in the N-Channel MOSFETs and allow the motor controller to work properly.

3.5.4 Future Work

The vehicle is much heavier than is desired. Some future work would be to find ways with which to remove weight, the simplest of which is to replace the current lead-acid batteries with a more compact and lightweight solution, such as nickel-metal hydride batteries or lithium-ion batteries.

More future work would be to make the vehicle even more compact than it already is. Currently, the space inside the cover is solely being used to house the payload. A study could be performed on better, more space efficient means to package all of the compartments within the current chassis, or one of a new design.

4: Sensor System Design

Applying power to the vehicle's motors will make it move, however, in order to effectively avoid obstacles, follow lanes and navigate, it will need to be able to sense its surroundings and make decisions based on the feedback it receives. The vehicle will require sensors that can provide feedback on the current speed, orientation, and geographic location of the vehicle, as well as the vehicles proximity to both intrusive and non-intrusive obstacles. Unmanned ground vehicles utilize a wide variety of different sensors to provide these types of feedback, and from the long list of possibilities several specific sensor systems have been identified that will allow the vehicle to navigate autonomously with both accuracy and speed. This chapter will explain, in detail, the sensor options that were researched and the specific systems that were chosen for the vehicle.

4.1 Research

Before specific sensor systems were chosen for the vehicle, research had to be conducted on the specifications and performance capabilities of a range of potential systems. The following sections outline the research that went into each sensor system on the vehicle.

4.1.1 LIDAR

Light Detection and Ranging (LIDAR) Systems are high-accuracy sensors that use reflected laser light to determine distances between the sensor and solid objects. A rotating mirror is used to sweep a laser beam through an arc in front of the sensor, and the amount of time it takes for the laser light to reflect back to the sensor is recorded at a set number of discrete measurement points. Multiplying these recorded times with the speed of light results in a series of distance measurements that can be used to make a two dimensional map of the area directly in front of the sensor. This capability is extremely useful in autonomous vehicles because accurate mapping is essential for path planning and obstacle avoidance algorithms that help the vehicle navigate through its environment²⁰.

As discussed in Section 2, the vehicle also incorporates a stereo vision camera system in addition to the LIDAR to gather additional information about the vehicle's environment. The combination of LIDAR and stereo vision systems on the vehicle is a very important part of the sensing strategy adopted in the design. If one sensor or the other were used purely by itself, a large amount of valuable data would go unutilized. For example, the LIDAR provides very accurate distance data (10mm accuracy over

²⁰ (Keith n.d.)

a 30m range)²¹, but can only see objects in its fixed line of view. The stereo vision system, on the other hand, has the ability to see obstacles in a wider range of locations. It also has the capability to sense some obstacles that the LIDAR cannot detect. Detection of non-intrusive obstacles such as boundary lines, potholes, and sand pits is handled solely by the stereo vision system. By combining the data returned from both the LIDAR and stereo vision systems, a more accurate representation of the environment directly in front of the vehicle is created. By using both sensors in tandem, the hope is to achieve the accuracy that is provided by the LIDAR and the sensing area that is provided by the stereo vision system.

LIDAR was not the only option considered for distance sensing. Infrared and ultrasonic range-finding sensors, which are inexpensive and accurate over 1-2 meters, were also considered. However, in order to match the accuracy and range of LIDAR, an array of multiple sensors needed to be implemented. Integrating and processing the data returned from a sensor array would have added additional complexity and extra processing to the process of determining distances, but the price of such a sensor array is considerably less than a fully-integrated LIDAR system. However, when the capabilities of each system were thoroughly compared, using LIDAR emerged as the more desirable, albeit more expensive option.

Vehicles that have participated in the IGVC in the past were also equipped with LIDAR as their primary navigation sensor. The most popular navigation method seen in previous IGVC competitions was to use the LIDAR for mapping and a low-tech camera for boundary line detection. The data from the two systems was integrated into a single cohesive map that the vehicle could use to navigate through the course.

Two potential LIDAR systems have been considered for use on the vehicle, a Hokuyo URG-04LX Laser Range Finder, and a SICK LMS291-S05 LIDAR system. Table 9 compiles and compares the technical specifications of both sensors.

²¹ (SICK n.d.)

Specification	Hokuyo URG-04LX	SICK LMS291-S05
Maximum Sensing Range	4 meters = 13.123 feet	80 meters = 262.47 feet
Maximum Scan Rate	10 Hz	75 Hz
Field of View	270°	Either 100° or 180°
Angular Resolution	0.36°	1°, 0.5°, or 0.25°
Power Consumption	2.5 watts at 5VDC	20 watts at 24VDC
Size	5.0 x 5.0 x 7.0 cm	18.5 x 15.6 x 21.0 cm
Approximate Price	\$2400	\$6000

Table 9: A comparison of technical specifications for the Hokuyo URG-04LX and SICK LMS291-S05 LIDAR systems²²



Figure 33: The Hokuyo URG-04LX (left) and SICK LMS291-S05 (right) LIDAR systems²³

The Hokuyo LIDAR has the benefit of being smaller, more power efficient, and less expensive, but it also has a smaller sensing range and slower scan rate than the SICK system. It was decided that the vehicle would benefit more from the increased sensing range and scan rate of the SICK system, despite the increases in size and power consumption compared to the Hokuyo system. The SICK LIDAR system being used on the vehicle was received as a donation from a WPI alumnus.

4.1.2 Stereo Vision

Stereo vision is a powerful sensor system that uses two cameras to determine 3-D information about the surrounding environment. Team Prometheus elected to use a stereo vision system to detect lines, potholes and to supplement the LIDAR for obstacle detection.

²² (Hokuyo Automatic Co. 2005)

²³ (Travis 2009)

Cameras are the most important design decision for the system and were chosen based on four metrics of performance; image resolution, sensor type, frames per second and cost. The choice of image resolution was a trade-off between distance calculation accuracy and image processing time. With a high image resolution there are more individual pixels in each picture making the analysis of that image more accurate. However image processing even in the GPU takes a lot of time, therefore it was determined that the image resolution could not exceed 2 Mega pixels.

Three sensor types were considered for the vision system, CMOS, CCD, and 3CCD. CMOS is the least expensive and also gives the lowest quality image. This low quality is a direct result of CMOS' Rolling Shutter capture method. Rolling Shutter exposes an image frame from top to bottom and does not capture the entire frame at once. This leads to blurring and under certain lighting conditions the appearance of flickering. The image noise generated by the CMOS capture method makes it an inadequate choice for the vision system.

CCD produces a higher quality image and is more commonly used in stereo vision applications. CCD uses the Global Shutter capture method, which allows the entire frame to be captured at the same time. 3CCD incorporates three CCD sensors one dedicated to red, blue, and green. This resolves the necessary estimation when using only one CCD sensor however 3CCD cameras are more expensive than CCD therefore, CCD was the image sensor of choice.

Frames per second (fps) are important because higher fps allows the vehicle to make better informed decisions while navigating. However too many fps at high resolutions could prove difficult to process. Therefore to avoid paying for a camera that could not be fully utilized the team determined, a frame rate of 7 to 30 fps was an acceptable choice.

SONY's handycams was the first set of cameras considered. Although these cameras are reasonably priced at \$200 each, they use CMOS sensors and do extra processing and image refinement for commercial viewing quality. Our application needs the raw image data and would have to undo some of the image refinements performed by the camera. Point Grey's cameras were also considered and are more desirable because they are developed specifically for robotic applications. Three Point Grey camera setups satisfy the requirements and are shown in Table 10.

Camera	Resolution	FPS	Image Sensor	#of Cameras	Price
Bumblebee [®] XB3	1288x964	16fps	1/3CCD	3	\$3,500.00
GRAS-14S3M/C	1384x1032	21fps	½ CCD	2	\$2,490.00
FL2G-13S2M/C	1024x768	15fps	1/3 CCD	2	\$1,598.00

Table 10: Grey Point camera setups, ordered by preference

The three different camera options are very similar in quality each having its own strong point. Since all options met the camera specifications discussed above, two FL2G-13S2M/C cameras from Point Grey have been selected for the stereo vision system due to their lower cost.

In order to choose a proper lens for the stereo vision system, a minimum distance for the field of view had to be determined. It was decided that at 3 meters in front of the camera, the entire width of the lane must be visible. With the distance chosen, the calculations for the focal length showed that any lens between 4-4.8mm fit the specifications. A manual iris and focus was also necessary since auto adjusting lenses would need to be re-calibrated each time the focus and iris change. The Flea2 cameras only offer a C-type lens mount which further reduced the lens options. The final choice was a Pentax C418DX lens, which fit all of the requirements.



Figure 34: Pentax C418DX

Pentax C418DX Lens	1/3" CCD Sensor
Horizontal Field of View	53.13
Vertical Field of View	41.11
Focal Length	4.8mm
Focus	.3 m->inf.
Manual Iris	0->1.8
Min distance to see full course width	3 m
Price	\$140

Table 11: Lens Specifications

With the cameras and lenses selected the only physical design choice left is the baseline distance between the two cameras. Distance calculations are more accurate with larger baselines,

however if the cameras are too far apart there will not be enough image overlap to successfully perform stereo vision processing. An acceptable baseline could not be determined through research and was chosen to be .18 meters after a considerable amount of testing.

4.1.3 DGPS

In order to compete effectively in the Navigation Challenge, the vehicle will have to have a way of knowing what its current geographic location is. In order to obtain this information, the vehicle will use a Differential Global Positioning System (DGPS) receiver capable of determining the vehicle's global latitudinal and longitudinal position. Using a DGPS receiver is highly recommended for the IGVC Navigation Challenge because it provides a direct measurement of the vehicle's global position. This position can then be compared to the positions of the Navigation Challenge waypoints in order to determine the vehicle's proximity to any given waypoint. DGPS is not the only method of determining the vehicle's location, but it is one of the most convenient because it provides continually updated, direct measurements of the vehicle's position in the same frame of reference as the course waypoints.

The DGPS receiver being used in the vehicle's navigation system is a Sokkia Axis 3 DGPS Receiver. The Axis 3 receiver can receive information from up to 12 GPS satellites at once, and communicates over a standard RS-232 serial connection using NMEA formatted data strings²⁴. Other DGPS receivers were considered in the research process, but all were rejected due to either their high cost or lack of accuracy. The Axis 3 receiver is being used on the vehicle because it can be easily integrated into the rest of the system structure, and because a receiver and antenna unit is available for donation.



Figure 35: The Sokkia Axis 3 DGPS Receiver²⁵

²⁴ (Ltd 1999)

²⁵ <http://www.afds.net/sokkiaaxis3pic.jpg>

4.1.4 Encoders

Another sensor that is being used in the vehicle's navigation system is an optical wheel encoder for each drive wheel. Optical encoders operate by attaching a disc with a series of slits cut into it onto a rotating axle. Inside the encoder housing there is a light source and a light receiver located on opposite sides of the disc. Each time the axle rotates a certain amount, the slit in the encoder disc allows the light from the source to hit the receiver. When this happens, the encoder sends a signal to its output indicating that the axle has moved a certain number of degrees. By including a large number of slits on a disc, a very accurate measurement of the rotational movement of the axle can be made²⁶.

The encoders being used on the vehicle are two US Digital E8P Optical Encoders. They have a range of 180 to 512 slits per encoder disc (referred to as counts per revolution), quadrature output (used to determine the direction of rotation of the axle), and can more than handle the amount of RPMs that the vehicle's drive motors spin at. The encoders are mounted inside the rear housing of the drive motors, connected directly to the motor drive shafts. This way, it is possible to get a more accurate measurement of the speed of the motors, and a higher resolution measurement of the rotation of the drive wheels²⁷.



Figure 36: The US Digital E8P Optical Encoders²⁸

4.1.5 Digital Compass

The final sensor being used in the vehicle's navigation system is a digital compass. Digital compasses use sensors called magnetometers to determine the strength of the magnetic fields that exist around the compass module. Specifically, they are able to determine the difference between the magnetic field of the Earth and other magnetic fields that are created from outside magnetic interference. This means that, even if a digital compass is located in an area that is subject to magnetic

²⁶ (Inc. 2009)

²⁷ (Digital n.d.)

²⁸ http://usdigital.com/assets/images/galleries/2009-04-30_004-Edit_1.jpg

interference, it will still be able to provide an accurate measurement of magnetic North. Before the compass can start returning accurate measurements, however, it must first undergo a calibration process in the environment that contains the magnetic interference. The ability of the compass to resist the effects of magnetic interference is especially desirable for the vehicle, because components such as the drive motors and power circuitry have the potential to emit magnetic interference. Once the compass is mounted inside the chassis, it will have to be calibrated to resist the internal magnetic interference of the vehicle in order to return accurate and usable heading information²⁹.

The digital compass being used in the vehicle's navigation system is a PNI V2Xe 2-Axis Compass Module. The V2Xe compass is capable of returning heading values with 2° worth of accuracy and 0.01° worth of resolution. Several other compass modules were considered for the vehicle, but were ultimately rejected due to high levels of heading measurement drift or complex communications protocols. The V2Xe compass is being used on the vehicle because it uses a simple RS-232 based communications protocol, and because it provides a relatively high degree of accuracy and resolution given its moderately low price compared to other compass modules³⁰.



Figure 37: The PNI V2Xe 2-Axis Compass Module³¹

4.1.6 IMU

Another sensor that was considered, but ultimately not used in the vehicle's navigation system was an inertial measurement unit (IMU). Most IMUs use a combination of three accelerometers and three gyroscopes to provide a measurement of acceleration in the X, Y, and Z axes, and a measurement of rotation about the X, Y, and Z axes (roll, pitch, and yaw). An IMU was available for a limited period of time, and during this time several data sets were taken for subsequent analysis.

The limited number of IMU experiments that were performed led to some useful individual data sets, and some combined IMU/DGPS data sets that were used for initial sensor fusion testing. There

²⁹ (ASD 2010)

³⁰ (Corporation 2005)

³¹ http://www.gorobot.org/files/imagecache/product_teaser/files/products/v2x3.gif

were some initial problems with interpreting the data and producing accurate representations of the velocity of the IMU, but the accuracy of the velocity measurements was improved as errors were systemically removed from the data analysis process. The most recently discovered error was a small amount of DC bias in the returned velocity information, which when integrated across all of the velocity samples created a noticeable error in velocity measurement.

The rotational information returned from the IMU, however, was very accurate and much more stable than the velocity information. Graphs of the IMU's heading over entire data sets showed very little error when compared with the actual heading of the IMU during that time.



Figure 38: The InterSense NavChip IMU³²

4.2 Development

The following sections explain in detail the development process that each sensor system went through before being integrated into the vehicle.

4.2.1 LIDAR

In order to detect obstacles the vehicle uses a SICK LMS-291 LIDAR range-finder. The unit is mounted to the front of the chassis and is used for detecting cones, barrels, poles, and other physical objects that are more than 7 inches off of the ground (the height at which the unit is mounted at). It is connected to the vehicle's cRIO via an RS-232 connection. Before the vehicle was built, the LIDAR and cRIO combination was tested on a lab bench in the configuration shown in Figure 39.

³² http://www.est-kl.com/uploads/pics/NavChip_Devel_web.jpg



Figure 39: LIDAR and cRIO

The LIDAR unit scans a 180 degree field of view and returns distance information in centimeters with an associated angle. Using an RS-422 connection at 500 Kbps the unit can return distance information at up to 75Hz. The cRIO does not have an RS-422 port so the connection used was RS-232 at 38.4 kbps. At this transfer rate, the LIDAR provides distance data at 4 Hz in increments of 0.5 degrees. The stream of data sent to the cRIO is 360 numbers separated by a space. The first number sent is the distance to an object at 0.5 degrees and the second number is the distance to an object at 1 degree. The rest of the numbers relate similarly up to 180 degrees. The cRIO receives the information using the SICK LMS driver provided by National Instruments. This driver establishes communication with the LIDAR unit, receives the data packets, and converts them from polar coordinates into Cartesian coordinates. This data is processed by the cRIO and transferred to the computer.

4.2.2 Stereo Vision

The basic configuration of the stereo vision system used for the vehicle is shown in Figure 40. The main objective of the system is to find distances to objects, lines and potholes. Distance is directly related to the inverse of pixel disparity and is shown in the equation below. Therefore the problem can be simplified into a search for pixel disparities between matching pixels in the left and right images.

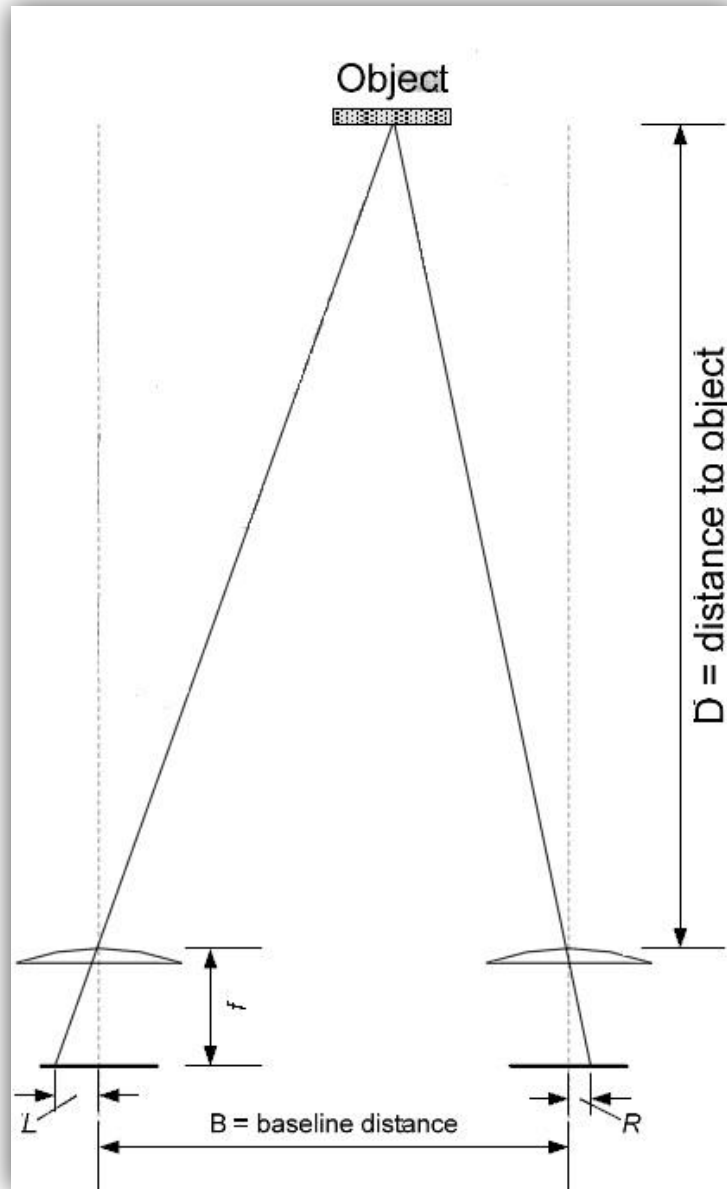


Figure 40: Stereo System Configuration³³

$$D = \frac{Bf}{d}$$

where f is the camera focal length, D is the distance to the object, B is the baseline distance between the two cameras and d is the disparity³⁴. Since B and f are known constants the only unknown variable that is needed to find distance is disparity.

³³ (Stam 2008)

To find the disparity Team Prometheus used a three step process consisting of; rectification, segmentation and pixel matching. The disparity map that results from pixel matching is then used to identify objects and mark their positions on the vehicles 2D probability map. To handle the immense amount of image processing required the Team uses NVIDIA's Tesla C1060 graphics processor. Since general purpose graphic processing is a relatively new concept most of the stereo vision functionality implemented on the GPU was designed by the team.

4.2.5.2 Image Rectification

Image rectification is the process of transforming multiple images onto a common image plane. In the case of the vehicle's stereo vision system this involves transforming both the right and left camera images. The rectification process uses epipolar geometry to transform the images such that the matching pixel to a point $p(x, y)$ in the left image is located at some point $p(x', y)$ in the right image. This greatly simplifies the pixel matching function only requiring a search along a single line rather than across the entire image.

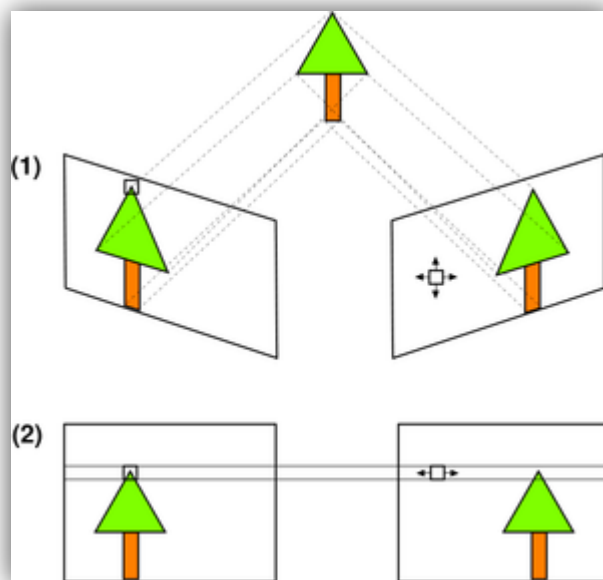


Figure 41: Image Rectification Process³⁵

However to perform image rectification the two cameras must first be calibrated to recover the intrinsic and extrinsic parameters of the stereo vision system. Intrinsic parameters define the focal length, image center in pixels and optionally the distortion parameters of the lenses, these parameters

³⁴ (Stam 2008)

³⁵ (Wikipedia 2010)

are unique to each camera. The camera intrinsic parameters are defined by the matrix shown in the equation below³⁶.

$$\text{Intrinsic parameters} = \begin{bmatrix} \alpha & -\alpha \cot(\theta) & u_0 \\ 0 & \beta / \sin(\theta) & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where u_0 is the image center in the x direction, v_0 is the image center in the y direction and α and β are the camera focal length in the x and y direction measured in pixels. θ is the angle between the lense and image sensor plane and in practice is extremely close if not equal to 90°.

The extrinsic parameters of the camera system define the rotation and translation of the cameras relative to each other. In practice one camera is used as a reference and the other camera is rotated and translated into the same plane as the reference camera. The calibration process used for the stereo vision system is described in the following section.

4.2.5.2.1 Calibration

For camera calibration Team Prometheus utilized OpenCV functions already developed for this purpose. The calibration process done in OpenCV involves first taking a series of pictures with a chessboard in different configurations, and then presenting these pictures to the calibration program which creates a mapping to convert the original left and right images into rectified form.

Successfully using the calibration program can be a challenge and there are some subtle but important factors involved in obtaining the correct results. The necessary steps are as follows:

1. The first step is to obtain pictures of a chessboard in different configurations. There is a separate program called calibration used for this purpose. A series of 15 pictures are taken and examples of acceptable chessboard pictures are shown below.

³⁶ (Forsyth and Ponce 2003)



Figure 42: Examples of calibration pictures

2. Next the pictures need to be listed in a text file called `stereo_calibration.txt` located in the config folder of stereo vision. It is important to note that the text files first line is the width-1 and height-1 of the chessboard in squares. If the actual width and height are used the program will not successfully find the squares in the chessboard.
3. After taking the pictures and setting up the text file the main calibration program is run. The main program should always be run in debug mode because the program can recognize the chessboard squares top to bottom, bottom to top or not at all. There is normally a mix of all three cases and any left right image pair that does not have the squares recognized top to bottom, needs to be removed from the file `stereo_calibration.txt`. Examples are shown in Figure 4.



Figure 32: Left image shows chessboard squares recognized in the correct order (top to bottom). The right image shows the incorrect order (bottom-top).

4. Once an acceptable set of images has been obtained the program will output four text files (mx_left.txt, my_left.txt, mx_right.txt, my_right.txt) and exit. The text files contain a x, y mapping to rectify each individual pixel.

Now when the stereo vision program is run the mappings are loaded into GPU memory and used to convert incoming images to rectified form using the equation:

$$\text{rectified image}[i][j] = \text{original image}[\text{mapping}_y[i][j]][\text{mapping}_x[i][j]] .$$

However, since the mappings are stored in floats and are very rarely perfect integers the following process is used to interpolate between the four overlapping pixels.

$$pos_x = \text{mapping}_x[i][j]$$

$$pos_y = \text{mapping}_y[i][j]$$

$$pixel_1 = \text{original image}[\lfloor pos_y \rfloor][\lfloor pos_x \rfloor]$$

$$pixel_2 = \text{original image}[\lfloor pos_y \rfloor][\lceil pos_x \rceil]$$

$$pixel_3 = \text{original image}[\lceil pos_y \rceil][\lfloor pos_x \rfloor]$$

$$pixel_4 = \text{original image}[\lceil pos_y \rceil][\lceil pos_x \rceil]$$

$$percent_1 = pos_x - \lfloor pos_x \rfloor$$

$$percent_2 = 1 - percent_1$$

$$percent_3 = pos_y - \lfloor pos_y \rfloor$$

$$percent_4 = 1 - percent_3$$

$$\text{rectified image}[i][j] = pixel_1 * (percent_1 * percent_2) + pixel_2 * (percent_1 * percent_4) + pixel_3 * (percent_2 * percent_3) + pixel_4 * (percent_3 * percent_4)$$

where i and j denote index positions within the image, $\lfloor \cdot \rfloor$ is the mathematical floor function and $\lceil \cdot \rceil$ is the mathematical ceiling function. The results from this entire process are shown in Figure 43. It can be noticed that the purple line passes through the same set of points in both images.



Figure 43: Rectified Images

4.2.5.3 Image Segmentation

Image segmentation is the next step in developing an accurate and reliable stereo vision program. It works to partition the image into meaningful regions of similar color and texture allowing later algorithms to identify and calculate distances to objects. There are many different methods of image segmentation to choose from including model based, histogram based, graph partitioning, clustering, edge detection and neural networks.

Our implementation uses a self organizing map with simulated annealing (SOM_SA)³⁷ and can achieve near optimal segmentation of color images with a significant reduction in computational cost. The SOM_SA algorithm is a two part approach to color image segmentation. First, a neural network variant called a self-organizing map (SOM) is trained to perform nonlinear color reduction from modified $L^*u^*v^*$ space to gray scale. The second part consists of a simulated annealing (SA) algorithm, which is used to cluster the self-organizing map into K clusters and produce a segmented image. Images come in from the cameras in RGB format and as a result must be converted to modified $L^*u^*v^*$ space.

RGB to Modified $L^*u^*v^*$

To successfully train a SOM to perform color reduction on RGB images we need a meaningful measure of distance between two pixels. This property cannot be provided by RGB, because the color space is non-uniform. Therefore we need to perform a non-linear conversion on the RGB image into a

uniform color space namely modified $L^*u^*v^*$. The conversion from RGB to modified $L^*u^*v^*$ is achieved using the equations listed below³⁸.

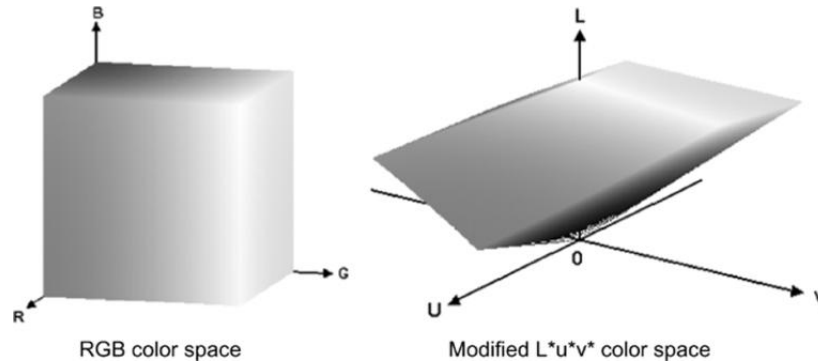


Figure 44: Color distributions in RGB and modified³⁸

The first step is converting the RGB values into CIE XYZ space using the 3x3 transformation matrix show below.

$$\begin{matrix} X \\ Y \\ Z \end{matrix} = \begin{bmatrix} .412453 & .35758 & .180423 \\ .212671 & .715160 & .072169 \\ .019334 & .119193 & .950227 \end{bmatrix} \begin{matrix} R \\ G \\ B \end{matrix}$$

Next we need to calculate $u' u'$ and $v' v'$.

$$u' = \frac{4X}{X + 15Y + 3Z} \quad v' = \frac{9Y}{X + 15Y + 3Z}$$

We will also need to use $(u'_n, v'_n) = (.1978, .4683)$ which are constants computed using the CIE XYZ points for white. The modified $L^*u^*v^*$ values are then obtained as follows:

$$L^* = 10\sqrt{Y} \quad u^* = 13L^*(u' - u'_n) \quad v^* = 13L^*(v' - v'_n)$$

SOM Color Reduction

Self Organizing Map (SOM) color reduction works to create a non-linear color reduction from modified $L^*u^*v^*$ color space to an alternative lower dimensional space. This conversion preserves the topology of the original color image and significantly reduces the time complexity of color clustering the

³⁸ (Dong and Xie 2005)

image. To perform the color reduction, the SOM map must first be trained. The training uses a neural network based approach, which maps a set of 3-D image points onto a two-dimensional array of nodes M , such that points neighboring in original 3-D image project to the nearby nodes in M . The SOM map size by default is 16×16 , with each node or neuron i containing a weight vector represented by $w_i = [w_{i1}, w_{i2}, w_{i3}]^T$. The weight values used correspond to L^* , u^* and v^* values from the initial image with each neuron representing a cluster center of image points. The default map size is recommended for testing because it contains exactly 256 nodes which can be used to perform a non-linear conversion from modified $L^*u^*v^*$ color space to gray scale. An example of an RGB image and its gray scale conversion achieved through SOM color reduction can be viewed in Figure 46 below. However to obtain this effect the SOM map must first be trained, the procedure first shown by Guo Dong and Ming Xie is outlined below³⁸.

1. Initialization: Start by randomly initializing the weight vectors of each neuron. In practice random numbers generated between -100 and 100 worked best, since many $L^*u^*v^*$ values fall into this range. By default training is performed in two phases, using the neighborhood radius $r = 16,5$ and learning rate $\alpha = .05, .02$. The first phase is used to order the SOM nodes, while the second phase refines their weight vector values. Neighbors are updated within the neighborhood radius using a Gaussian distribution function. In practice two training phases did not provide sufficient spacing between SOM nodes. Therefore Team Prometheus used three training phases, with a neighborhood radius $r = 16,5,1$ and learning rate $\alpha = .04, .015, .0075$ to train the SOM map. Here the first two phases generally serve the same purpose, while the third phase is used to ensure there is significant difference between SOM nodes.
2. Input: During each training phase image points are iteratively chosen and presented to the network. In each training phase all image points are used exactly once and presented to the entire network simultaneously.
3. Competitive Process: For each iterative time step t , an image point $x(t) = [l(t), u(t), v(t)]^T$ is presented to the network of nodes M . The winning node c is determined to be the node with the shortest distance between image point and node weight vector. This is shown by equation below.

$$\|x(t) - w_c(t)\| = \min_i \{\|x(t) - w_i(t)\|\}$$

4. Cooperative Process: The effect on the winning node c 's neighbors is determined by a Gaussian function centered at c with radius $R_c(t)$.
5. Adaptive Process: The weights of nodes with in the neighborhood radius $R_c(t)$ are updated using the following equation:

$$w_i(t + 1) = \begin{cases} w_i(t) + \alpha(t)h_{ci}(t)[x(t) - w_i(t)] & \text{if } i \in R_c(t) \\ w_i(t) & \text{else} \end{cases}$$

In this equation $\alpha(t)$ is the learning rate and $h_{ci}(t)$ is the Gaussian function centered at winning node c .

6. Iteration: The next image point is selected at time $t+1$. Both the learning rate α and neighborhood radius R are linearly reduced using equations below, where T is the total number of pixels in the image.

$$\alpha(t + 1) = \alpha(0)(1.0 - t/T)$$

$$R(t + 1) = R(0)(2.0 - t/T)$$

The procedure is repeated from step 3 until all image pixels have been presented to the network. After all iterations have been complete in a given training phase the following statements should be true ($t = T, \alpha = 0, R = 1$).

7. After training, color reduction is performed by determining the best matched SOM map node for every pixel in the input image. The new pixel value x is determined to be index of the best matching node and is given by

$$x = \underset{i}{\operatorname{index @ argmin}}\{\|x - w_i\|\}$$

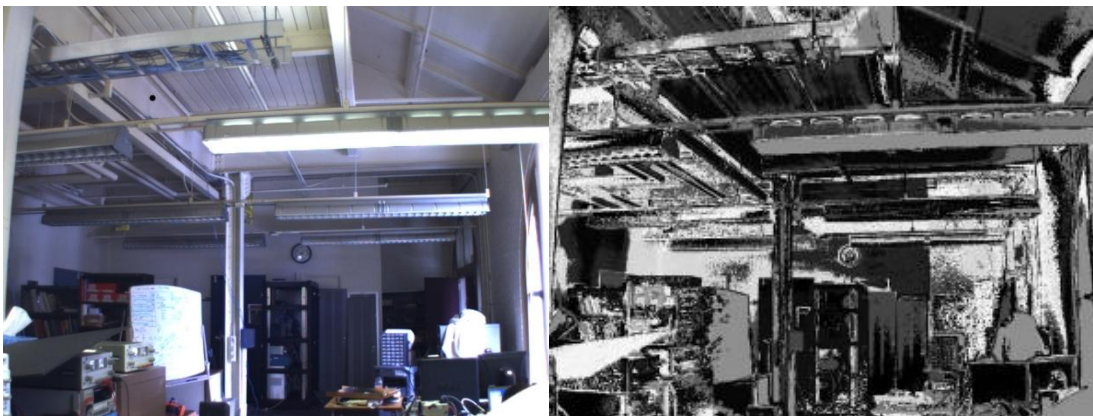


Figure 45: RGB image and its gray scale conversion achieved through SOM color reduction

SA Color Clustering

SOM color reduction maps a set of 3-D image points onto a 2-D array of nodes M . The objective of color clustering is to further organize M into K clusters, so that nodes associated with a cluster c_i are more similar than nodes associated with any other cluster. Simulated Annealing (SA), a type of hill climbing algorithm, attempts to find the optimal clustering solution by injecting randomness into the system. The principle idea is to initially inject a high amount of randomness, which is then gradually reduced to a low final level. This avoids the local optimum problem that plagues many hill climbing algorithms and offers a good solution to color clustering.

However to successfully perform the simulated annealing algorithm a method is needed to quantify the uniformity of color clusters. The energy function J serves this purpose and is defined by the equation listed below, where Z represents the set of cluster centers $\{z_1, z_2, \dots, z_K\}$ and C represents the set of clusters $\{c_1, c_2, \dots, c_K\}$.³⁸

$$J(c_1, c_2, \dots, c_K) = \sum_{i=1}^K \sum_{m_j \in c_i} \|m_j - z_i\|$$

The SA procedure initially starts at a high temperature of T_{max} and assigns each node in M to a random cluster in C . The cluster centers $\{z_1, z_2, \dots, z_K\}$ are calculated by equation listed below³⁸.

$$z_i = \frac{1}{n_i} \sum_{m_j \in c_i} m_j$$

Next, a point p from the data set M is randomly selected to be redistributed from its initial cluster c_i to a randomly selected cluster c_j , where $c_i \neq c_j$. The energy function is then recalculated from J to J' and if J' is less than J the redistribution of point p is accepted. Otherwise the redistribution of point p is only accepted if the probability of the function $e^{-(J' - J)/T}$ is higher than a randomly generated number between 0-1. For each value of the temperature T , a given number of points N_T are redistributed. After N_T points have been redistributed the temperature T is lowered by the function $T_t = \alpha T_{t-1}$, where α is the annealing factor, $\alpha = 0.8 - 0.99$. This process continues until $T \leq T_{min}$. The pseudo code for simulated annealing is shown below³⁸.

```

T = Tmax
randomly assign M to K clusters
while (T ≥ Tmin)
    for (i = 0 to NT, i = i + 1)
        J(c1, c2, ..., cK) = ∑i=1K ∑mj ∈ ci ||mj - zi||
        randomly redistribute point p from ci to cj
        J' = J + ||p - zi|| - ||p - zj||
        if J' < J then J = J'
        else if e-(J'-J)/T < rand(0 to 1) then J = J'
    end for
    T = αT
end while

```

GPU Implementation

The implementation of the SOM_SA algorithm for the vehicle's stereo vision system is a slight variation on the original algorithm and consists of a two phase process. The first phase consists of training and clustering a master copy of the SOM map and is largely preformed in the CPU. The second phase is image segmentation, which is preformed almost exclusively inside the GPU. It was necessary to separate the training and segmentation phases because training offers little opportunity for data parallelism and incurs a high computational cost.

The procedure for the first phase begins by creating and initially training a SOM map on an image loaded into GPU memory. The initial training is done using three training phases, with a neighborhood radius $r = 16,5,1$ and learning rate $\alpha = .04, .015, .0075$. After the initial training is complete the SOM map is saved to file, so it can be used again for subsequent training and segmentation. The next step is to fine-tune the initialized SOM map on a series of reinforcement images. Fine-tuning consists of running one training phase per image on the SOM map using a neighborhood radius $r = 1$ and learning rate $\alpha = .0005$. The fine-tuning ensures that the resulting SOM map is sufficiently distributed throughout the Modified L*u*v* color space. In practice reinforcement images should be taken in a variety of different settings to ensure a good distribution. In the system the master SOM map was created using four different sets of fifteen images.

Once the training has been fully completed the clustering begins. There are two important factors involved in SA clustering; the algorithmic runtime and the number of clusters K. The algorithmic runtime is dependent on the variables N_T , T_{max} , T_{min} , and α . These variables need to be configured so that the energy J is guaranteed to asymptote out at a low final level. In practice we used the following values:

$$\alpha = .99, \quad T_{max} = 3000, \quad T_{min} = 15, \quad N_t = 500$$

The second important factor K determines whether the final segmented image is over or under segmented. In practice we used 7 color clusters, which tends to slightly over segment the image but is necessary for our implementation because of an assumption made by later algorithms about the segmentation. In short the assumption is every segment boundary is not necessarily an object boundary, however every object boundary is a segment boundary. This assumption is not satisfied by under segmented images.

The SOM map resulting from phase one allows us to use one master map for all image segmentation. This greatly reduces the per image runtime in phase 2 from about 3 minutes in the original algorithm to 8ms in the modified version. An example of the master SOM map used for image segmentation is shown Figure 46.

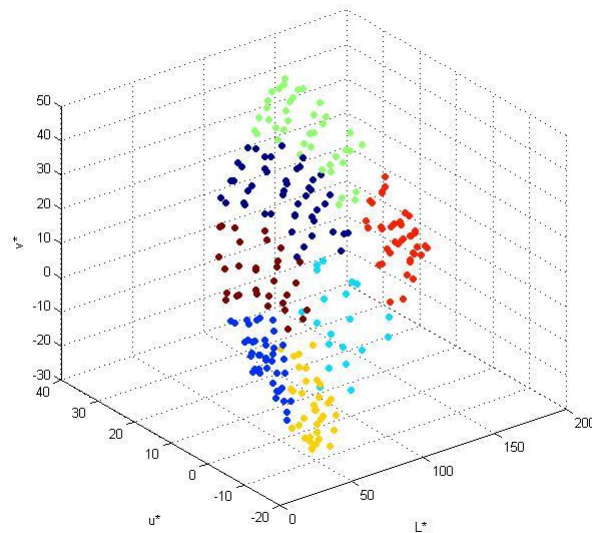


Figure 46: Example of SOM map used for image segmentation.

Phase two starts by loading the master copy of the SOM map into GPU memory. Next each incoming image is segmented using a two step process that first converts the RGB image into modified $L^*u^*v^*$ space, then segments the image based on the master copy of the SOM map. Both steps are implemented completely in GPU with a high amount of parallelism. The first step, image conversion spawns one thread of operation per image pixel and follows the mathematical process described in the section above. The second step also uses one thread of operation per pixel and first determines the matching SOM node for each pixel p using the equation $w_c = \operatorname{argmin}_i \{\|p - w_i\|\}$. After determining w_c we use the color cluster value associated with w_c as a pixel value for the outputted segmented

image. The result is an image segmented based upon cluster value, an example segmentation image along with pseudo code is shown below.

segment(Input: Luv image, Output: segmented image)

for \in pixel p in Input

$$w_c = \operatorname{argmin}_i \{\|p - w_i\|\}$$

Output @ index of p = cluster value of w_c

end for



Figure 47: Original color image and resulting segmentation without clustering

Figure 47 depicts the resulting segmentation images however they contain some noise. To deal with the noise a k-nearest neighbors filtering technique was applied on the resultant segmented image. The k nearest neighbors algorithm is also run in the GPU using one thread of operation per pixel and sets the value of the current pixel to the cluster value that appears the most times in its neighborhood radius r . In practice we used a radius of $r = 2$, which gave us a 5x5 kernel to perform the k nearest neighbor's algorithm over. A sample of segmentation before and after filtering is shown in Figure 48.

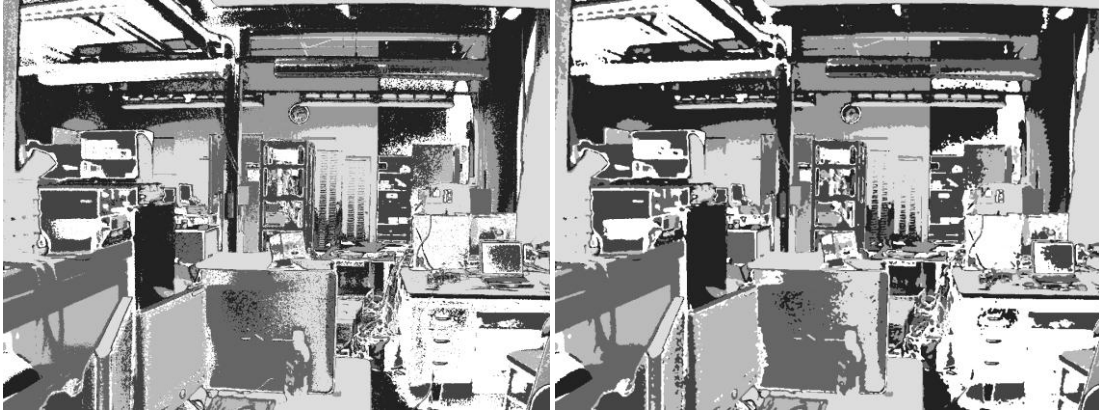


Figure 48: Segmentation with (right) and without (left) filtering

4.2.5.4 Stereo Matching

Stereo Matching is the process of matching pixels between camera images in a stereo vision system. There are many different methods of pixel matching that can be generally classified into two groups based on the sparse or dense disparity maps they produce. Algorithms that produce sparse maps generally use some form of scale invariant features. Scale invariant features can be matched with a high degree of accuracy leading to an accurate but sparse disparity map. However without extra image processing it is impossible to guarantee that the scale invariant features will be generated on objects of interest. Algorithms that produce dense disparity maps match blocks of pixels using a sum squared difference (SSD) or sum absolute difference (SAD) approach. These algorithms match a large portion of the pixels between both stereo images, however they also contain a high percentage of miss matched pixels. Additionally object detection and image segmentation can be used to supplement both methods.

Team Prometheus chose to use a sum squared difference approach to solve the pixel matching problem. The team chose this method for two reasons, a dense disparity map is more desirable and NVIDIA already published an open source version done in CUDA. NVIDIA's SSD approach is fully optimized for the GPU hardware and saved the team time in both implementation and image processing. Additionally to improve the accuracy of the resulting disparity map, the team also implemented a segmentation based averaging technique. The combination of both algorithms produces a dense disparity map that has acceptable accuracy.

SSD Based Pixel Matching

The SSD approach used by NVIDIA matches blocks of pixels between the left and right rectified images. Figure 49 shows the block matching approach using a 6x6 block size.

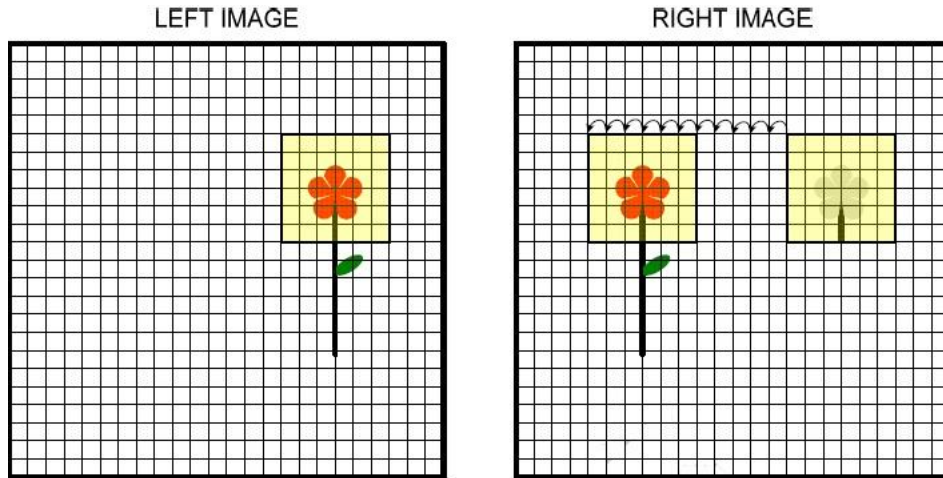


Figure 49: Pixel Block Matching³⁹

The SSD calculation is done using the following equation.³⁹

$$SSD_{x,y} = \sum_{i=x-R_H}^{x+R_H} \sum_{j=y-R_V}^{y+R_V} (Left_{i,j} - Right_{i-d,j})^2$$

where x and y are the pixel coordinates, R_H is the horizontal block radius, R_V is the vertical block radius and d is the current disparity being evaluated. The SSD is calculated for all blocks in the right image over a specified disparity range from d_{min} to d_{max} centered at the pixel position x, y in the left image. After all SSD values are calculated the pixel is determined to be matched if the lowest SSD value is less a minimally acceptable value Min_{SSD} . The disparity value d is then added to the disparity map at point x, y . Pixels that are not matched are assigned a value of $d_{min} - 1$ so they can be easily identified in later algorithms. The pseudo code for this algorithm is shown below and more information on the implementation as done by NVIDIA can be found here.³⁹

BlockMatching(Input: Left Image, Right Image, Output: disparity map)

for \in pixel point x, y in Left Image

$$SSD_{minval} = \sum_{i=x-R_H}^{x+R_H} \sum_{j=y-R_V}^{y+R_V} (Left_{i,j} - Right_{i,j})^2$$

$$d_{final} = d_{min} - 1$$

for ($d = d_{min}; d < d_{max}; d = d + 1$)

$$SSD_{x,y} = \sum_{i=x-R_H}^{x+R_H} \sum_{j=y-R_V}^{y+R_V} (Left_{i,j} - Right_{i-d,j})^2$$

if ($SSD_{x,y} < SSD_{minval}$ and $SSD_{x,y} < Min_{SSD}$)

$$SSD_{minval} = SSD_{x,y}$$

$$d_{final} = d$$

end if

end for

$$disparity\ map[y][x] = d_{final}$$

³⁹ (Stam 2008)

end for

Even though this algorithm was already written and optimized, there was still a considerable amount of tuning needed to get effective results. There are five user defined variables R_H , R_V , d_{min} , d_{max} and Min_{SSD} that all affect runtime and accuracy of the algorithm. R_H and R_V determine the block matching size and should be assigned the same value. Small block sizes are computationally fast but lead to many inaccurate matches. Large block sizes are computationally slow and if too large can lead to sparse and sometimes inaccurate matches. After considerable testing Team Prometheus chose to use a 5x5 block size or $R_H = 2$ and $R_V = 2$. d_{min} and d_{max} determine the maximum possible disparity value and should also be set to the same value. In practice d_{min} and d_{max} were both set to 150, this was able to detect most matches and could be completed in a reasonable time frame. Min_{SSD} determines the quality of each match and has to be adjusted each time the block radius changes. In practice a low value of 200 was used, which allows the algorithm to filter out many bad matches. Using these variable values the entire SSD algorithm runs in about 40ms. The resulting disparity map without segmentation based averaging is shown in Figure 50.

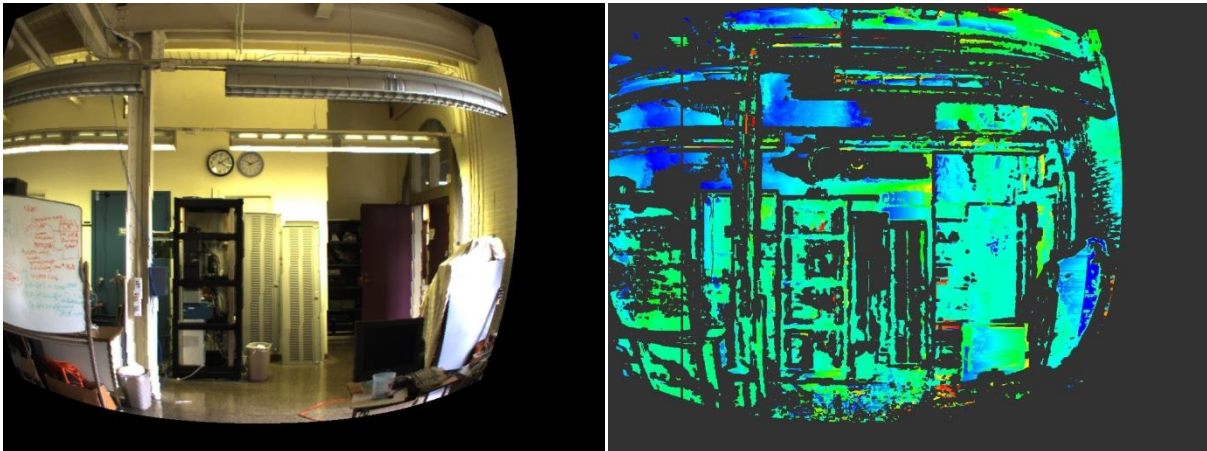


Figure 50: Original rectified left image and resulting non-filtered disparity map

Segmentation Based Averaging

Segmentation based averaging uses the color segmented image to refine the disparity map produced by NVIDIA's SSD program. For this method to be effective the following assumption is made about the segmented image. All object boundaries lie on segment boundaries, however not all segment boundaries lie on object boundaries. Using this assumption disparity values within objects can be averaged without being contaminated by unrelated disparities outside those objects. Figure 48 shows this assumption is indeed satisfied by the color segmentation method.

Ideally any disparity values that lie outside the current color segment should be considered outliers and entirely ignored. However to satisfy the assumption above the image had to be over segmented, therefore outliers do not necessarily belong to different objects⁴⁰. To solve this problem Team Prometheus used a variable $\alpha \leq .05$ to discount disparity values outside the current color segment. This allows small over segmented sections of the image to be effectively averaged, while still avoiding contamination from outlier pixels. The algorithm used for segmentation based averaging is shown in pseudo code below. Where w_r is the radius of the window size used for averaging, in practice $w_r = 10$ and $\alpha = .01$.

```

segmentBasedAveraging(Input: disparity map, segmented image, Output: disparity map)
  for  $\in x, y$  point in disparity map
    starty =  $y - w_r$ 
    endy =  $y + w_r$ 
    startx =  $x - w_r$ 
    endx =  $x + w_r$ 
    pointsegval = segmented image[y][x]
    sum = 0, count = 0
    for (  $i = start_y; i < end_y; i = i + 1$  )
      for(  $j = start_x; j < end_x; j = j + 1$  )
        dtemp = disparity map[i][j]
        if dtemp is a valid disparity
          currentsegval = segmented image[i][j]
          if( pointsegval == currentsegval )
            sum += dtemp
            count += 1
          else
            sum += dtemp *  $\alpha$ 
            count += 1 *  $\alpha$ 
          end if
        end if
      end for
    end for
    disparity map[y][x] = sum/count
  end for

```

The segmentation based averaging technique described above is effective in correcting the majority of disparity errors. However in practice the algorithm produced unacceptable results in areas of the image that consisted of a large number of invalid disparities. To solve this problem Team Prometheus does not calculate averages for disparity points that were initially invalid. This produces a sparser but also more accurate disparity map; the results of this approach are shown in Figure 51.

⁴⁰ (Gerrits and Bekaert 2006)

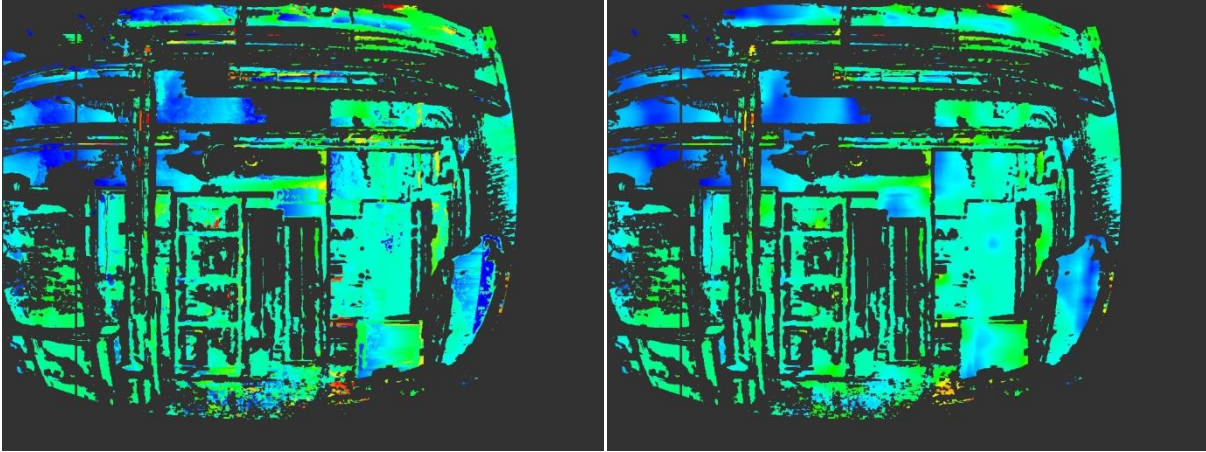


Figure 51: Disparity map without filtering (left) and disparity map after filtering (right)

4.2.5.5 Object Detection

To successfully perform object detection the ground plane must be removed from the disparity map generated in the previous section. Once the ground plane has been removed all the remaining points either belong to objects or background scenery. However since the background scenery falls outside the range of the vehicles local map it will not be recorded as an object. Team Prometheus does not currently have an implemented method to remove the ground plane and detect objects, nevertheless the team plans to implement the following process.

The process consists of two phases, initialization and object detection. In the initialization phase a reference disparity map d_{ref} is created using an open field devoid of any objects. In the object detection phase, blocks of pixels in the current disparity map are compared against d_{ref} . If the compared disparity values are significantly different the pixels are marked as an object and added to the vehicles local map. The entire process assumes that the vehicle is always navigating on a flat plane and is a conceptually easy solution to detecting objects on an open field and can be implemented in parallel.

4.2.5.6 Line and Pothole Detection

Team Prometheus does not currently have an implemented method to detect lines and potholes. The process the team plans to implement has been partially started and is outlined below.

1. The first step in the process is to segment the original RGB image based on white color. Since both the lines and potholes are white in the IGVC this is a reasonable operation. To perform the segmentation the image must first be converted into modified $L^*u^*v^*$ space,

where white color can be detected more reliably. In practice Team Prometheus determined white color to be any pixel with the following properties:

$$\text{a. } L^* > 150 \qquad -0.1 < u^* < 0.1 \qquad -0.1 < v^* < 0.1$$

Figure 52 shows the results of white color segmentation using these properties.

2. The white colored feature points detected in the first step are referenced to the disparity map and directly added to the vehicles local map.
3. Additionally dashed lines will be detected using a Hough Transform that is performed over all detected line points on the vehicles local map.

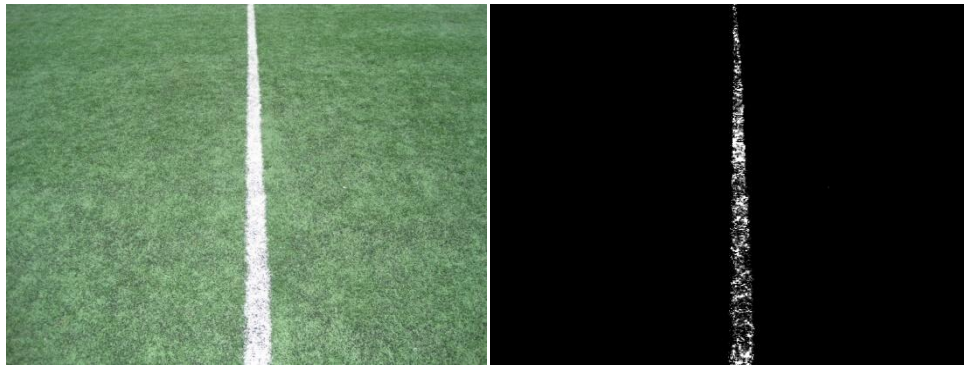


Figure 52: Line Detection

4.2.3 DGPS

The first step in developing the DGPS system was to formulate the equations that would take the raw latitude and longitude measurements returned by the DGPS and convert them into an (x,y) Cartesian point representing the same location on the Earth's surface (a Cartesian representation of the vehicle's position is necessary for mapping and path planning operations). One of the most accurate ways of doing this is through a series of equations that convert the latitude and longitude data into Universal Transverse Mercator (UTM) coordinates. The UTM coordinate system is a Cartesian coordinate system transposed onto the surface of the Earth, so it is capable of mapping any point on the Earth's surface in terms of its Northing (y) and Easting (x). The accuracy of the UTM conversion is so high because the equations take into account the fact that the Earth is not a perfect sphere. The equations treat the Earth as an ellipsoid, with major and minor axes that are defined by models of the Earth such as the WGS84 datum used by GPS. The WGS84 datum is used as the model of the Earth in the vehicle's navigation system. A map of the UTM grid overlaid on the surface of the Earth is shown in Figure 53:

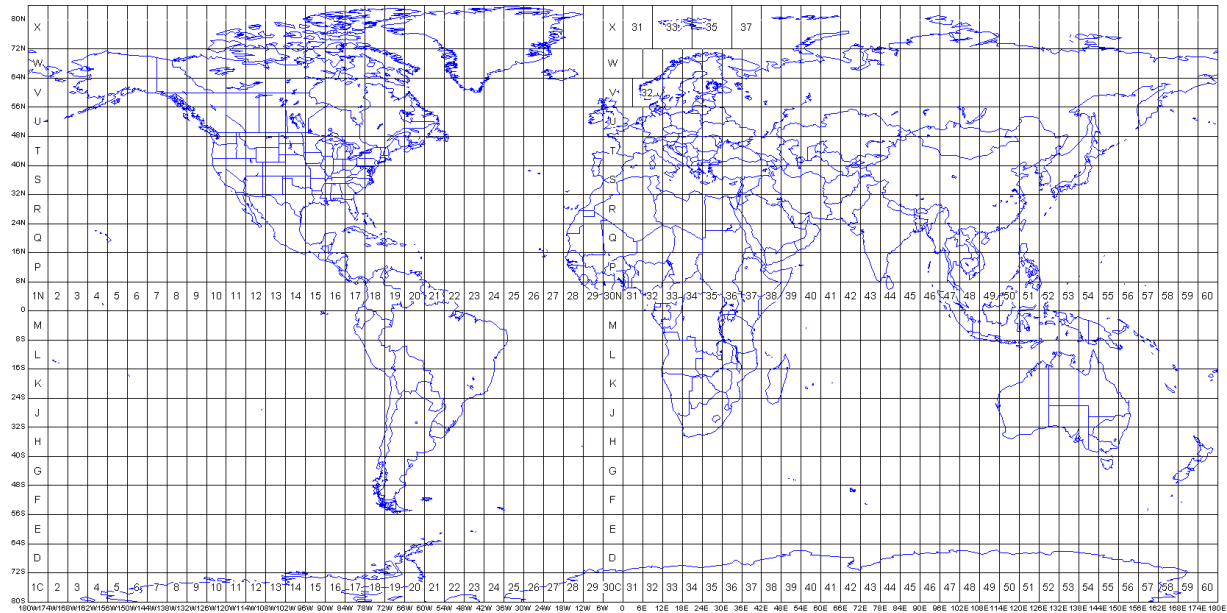


Figure 53: The UTM grid overlaid on the surface of the Earth⁴¹

The equations for the conversion from latitude and longitude to UTM Northing and Easting are illustrated below⁴²:

$$y = \text{Northing} = K1 + K2p^2 + K3p^4$$

$$\text{Where } K1 = Sk_0, \quad K2 = \frac{k_0 v \sin(2lat)}{4},$$

$$K3 = \left(\frac{k_0 v \sin(lat) \cos^3(lat)}{24} \right) (5 - \tan^2(lat) + 9e'^2 \cos^2(lat) + 4e'^4 \cos^4(lat))$$

$$x = \text{Easting} = K4p + K5p^3 + 500000$$

$$\text{Where } K4 = k_0 v \cos(lat), \quad K5 = \left(\frac{k_0 v \cos^3(lat)}{6} \right) (1 - \tan^2(lat) + e'^2 \cos^2(lat))$$

p = difference between the longitude of the point and the central meridian of the UTM zone

⁴¹ http://www.sdrmaps.com/html/New_Mexico_docs/Maps/UTM.gif

⁴² (Dutch 2009)

$S = \text{the meridional arc through the point}$

$$k_0 = 0.9996$$

$v = \text{the radius of curvature of the Earth perpendicular to the meridian plane}$

$$e' \approx 0.007$$

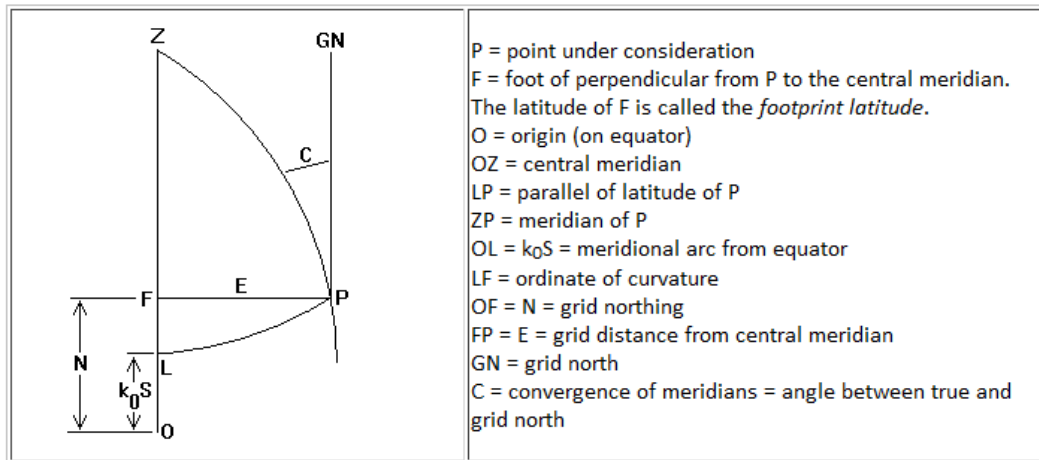


Figure 54: The geometry behind converting latitude and longitude to UTM⁴³

The UTM coordinate system is a global coordinate system, so its origin is based on the Equator. For the vehicle's navigation algorithms, the origin of the map is based on the starting position of the vehicle. To shift the origin, the first returned UTM point is treated as a location bias, and is subtracted from all subsequent points. By doing this, the first returned point becomes (0,0), and all subsequent points are referenced to that new origin.

The second step in developing the DGPS system was to collect real world data and analyze it to characterize the accuracy and relevant errors in the DGPS receiver. Two main data samples were taken from the athletic fields at WPI. The first sample was taken with the DGPS receiver stationary at the center of the football field, and the second sample was taken with the DGPS receiver moving around the perimeter of the football field. Both the stationary and dynamic data are important for characterizing the performance and accuracy of the DGPS receiver. The stationary data characterizes the drift and relative position accuracy of the receiver, while the dynamic data characterizes the ability of the receiver to maintain position accuracy while its position is constantly changing.

The final step in developing the DGPS system was to analyze the data taken from the WPI athletic fields and characterize the performance of the DGPS receiver. Three main errors can be calculated from the position data: the RMS position error, the 50% Circular Error Probable, and the 95%

⁴³ <http://www.uwgb.edu/dutchs/Graphics-Geol/FIELDMTH/UTMConvert.gif>

Circular Error Probable. All three of these quantities are measurements of how accurately the DGPS receiver can determine its current position while remaining stationary. To determine these errors, the error between each returned position and the true position must first be determined. Then, all of the returned positions with errors less than or equal to 50% of the maximum error are enclosed in a circle. The radius of this circle represents the 50% CEP error, and it means that any returned value from the DGPS receiver has a 50% probability of having the amount of accuracy that is defined by that circle. The RMS and 95% CEP errors are calculated in the same way, with the exception that the RMS error is calculated using 67% of the returned positions and the 95% CEP error is calculated using 95% of the returned positions. The 95% CEP error is often used as a worst case scenario for the accuracy of the DGSP receiver, while the RMS error is often used as an estimation of the average accuracy of the DGPS receiver⁴⁴.

The finalized DGPS system is capable of returning the current position of the vehicle with respect to its original starting point. It also has a known amount of position error that is accounted for in the overall Navigation System by introducing data from additional sensors.

4.3 Results

The following section contains all of the results that have been obtained from the sensor systems at the time this report was written.

4.3.1 LIDAR

Once the LIDAR unit is communicating with the cRIO, the data is converted from polar coordinates to Cartesian coordinates. The data is then able to be transferred to the PC. In order to test how well the LIDAR is working, it was set up in the hallway of the electrical engineering building and data was collected. This data was plotted and can be seen in Figure 55.

⁴⁴ (Mehaffey n.d.)

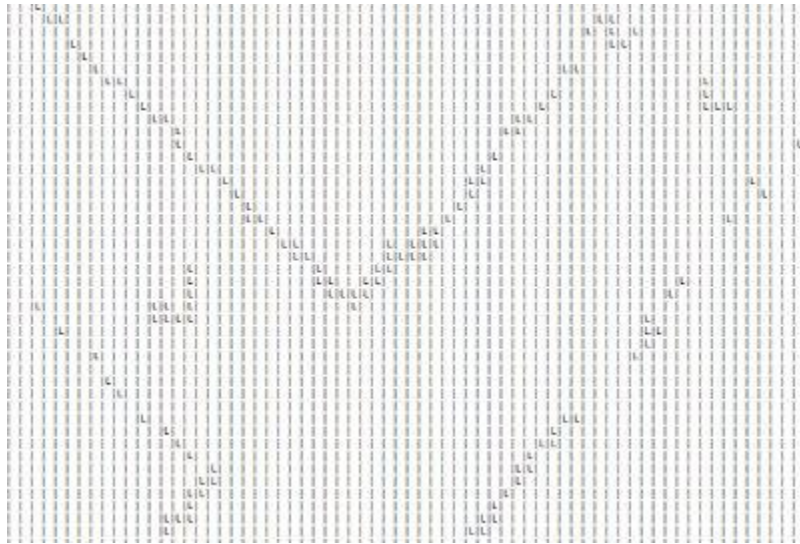


Figure 55: LIDAR Data test

In this figure, the hallways of the building can clearly be seen. In the center of the left hallway a trash can was placed and in the center of the right hallway a traffic cone was placed. Both of these objects can also be seen in the image created from the LIDAR data. The hallways in the building are not perfectly flat and bumps prevent the LIDAR from seeing the entire length of the wall. This phenomenon is shown in the image as “holes” in the walls. If left alone, the path finding algorithm might think these holes are possible paths. To prevent this from happening, the stored map assumes that areas behind obstacles, where the LIDAR can’t see, have a high probability of being un-drivable.

4.3.2 Stereo Vision

The software development for the stereo vision system is not yet complete; however the team has implemented the entire pixel matching process including segmentation and has begun line detection. Currently the entire process runs in 62ms or 16fps and consists of rectification, segmentation, line detection and distance calculation. The algorithms implemented so far show promising results and are summarized below.

1. Rectification: Successfully rectifies both stereo images in .2ms. The resulting rectification is accurate and reliable as shown in Figure 56



Figure 56: Rectification

2. Segmentation: Neural network based image segmentation runs in 8ms and performs well in a variety of environments. Figure 57 show the segmented images produced in some outdoor environments.



Figure 57: Color segmentation on outdoor environments

3. Stereo Matching: The pixel matching process with segmentation based averaging runs in about 51ms. The resulting disparity map has shown good results in an indoor environment as shown in Figure 58.

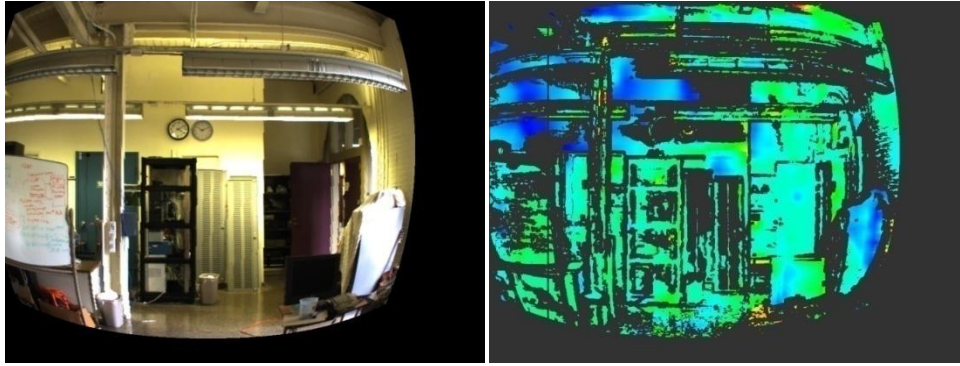


Figure 58: Original rectified image and resultant disparity after averaging

4. Line Detection: The current implementation of line detection runs in .2ms and can successfully detect white lines on a field as shown in Figure 59.



Figure 59: Line detection on football field

4.3.3 DGPS

Figure 60 through Figure 62 show the results from the two DGPS data samples taken from the athletic fields at WPI, as well as the RMS and CEP errors calculated from the stationary data:

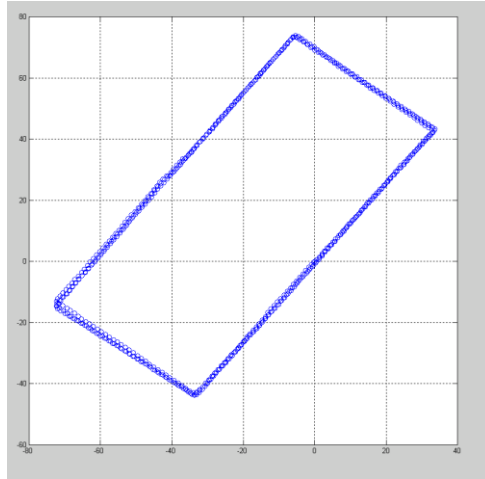


Figure 60: The dynamic DGPS data taken from walking around the perimeter of the WPI football field

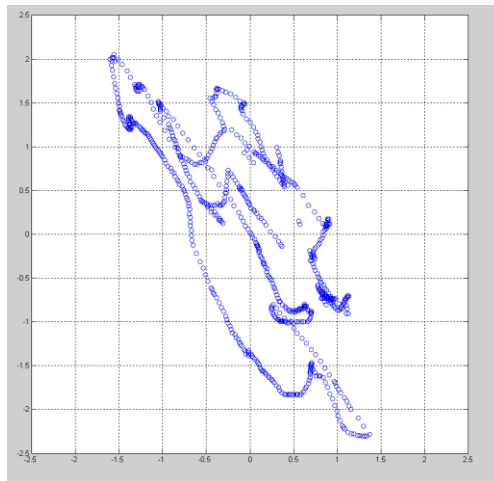


Figure 61: The stationary DGPS data taken from the center of the WPI football field

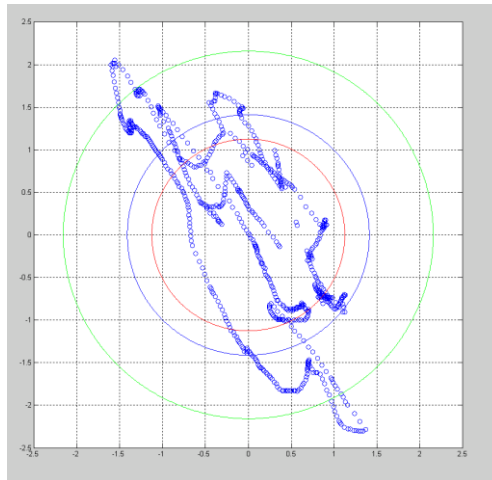


Figure 62: The 50% CEP (red), RMS (blue), and 95% CEP (green) errors calculated from the stationary DGPS data

As Figure 61 illustrates, there is a non-negligible amount of drift in the DGPS stationary data. To more specifically characterize this drift, the calculated values for the 50% CEP, RMS, and 95% CEP errors have been compiled and are shown in Table 12:

Type of Error	Value of Error
50% CEP	1.1257 meters
RMS	1.4119 meters
95% CEP	2.1583 meters

Table 12: Calculated values of the 50% CEP, RMS, and 95% CEP errors from the stationary DGPS data collected at WPI

4.3.4 IMU

Figure 63 and Figure 64 show a set of velocity and heading results taken from the IMU during two different data samples at WPI:

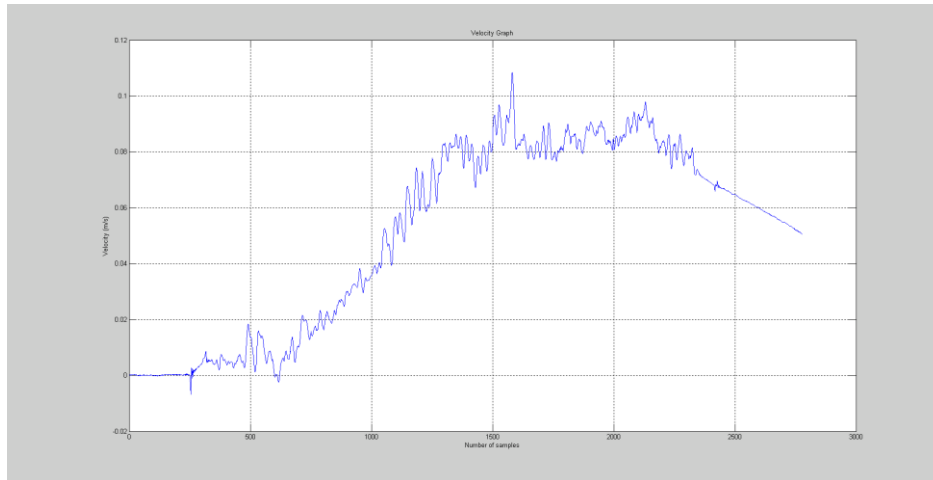


Figure 63: Velocity information gathered from the IMU during a data set where the IMU was being pushed across the surface of a desk

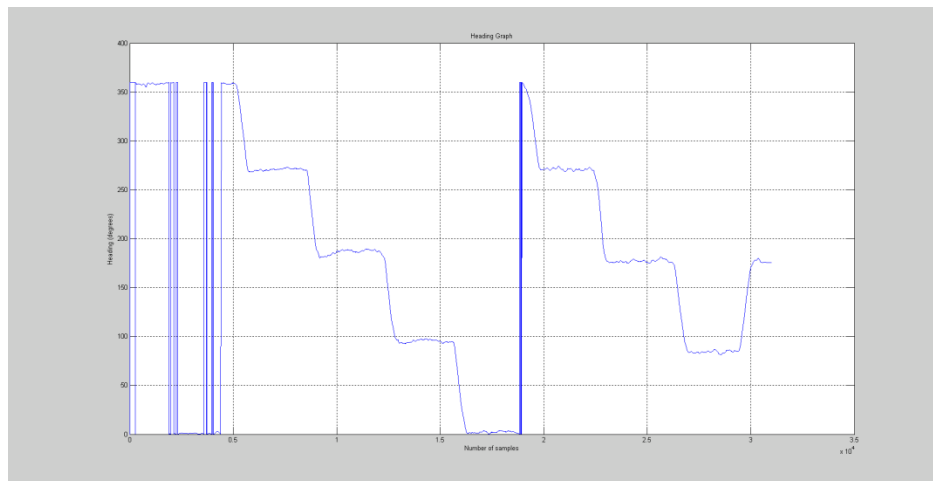


Figure 64: Heading information gathered from the IMU during a data set where the IMU was being pushed on a cart

As Figure 63 and Figure 64 illustrate, the velocity information is very noisy due to the sensitivity of the IMU accelerometers, while the heading information is much more consistent. Refining the velocity information is currently still a work in progress.

4.4 Discussion

4.4.1 LIDAR

The team was successfully able to receive data from the LIDAR at a rate of 4 Hz and at increments of 0.5 degrees. This information was then successfully used to update the probability map of the vehicle's surrounding terrain. Up to this point, no issues with the LIDAR or LIDAR data have been found during testing. The only work to be done for the LIDAR sensor is improving the communication

protocol so that information can be retrieved at a faster rate. The team acquired an RS232 serial module for the cRIO that supports data communication rates up to 500 kbps. This means that an RS232 to RS422 converter can be used to communicate with the LIDAR. This increased data rate will mean an increased LIDAR scan rate up to 30 Hz with an angular resolution of 0.5 degrees.

4.4.2 Stereo Vision

The team will finish the software implementation for stereo vision system before the Intelligent Ground Vehicle Competition on June 4th. Based on the results presented in this report, the system as designed will be able to detect obstacles, lines and potholes on a field with acceptable accuracy. Even though the vision system will satisfy the design requirements there is still room for future work.

1. First and most importantly none of the algorithms implemented are fully optimized for the GPU. Optimizing the current algorithms could significantly speed up the image processing. This would allow for more algorithms to be run or a higher fps camera to be used.
2. The segmentation paper³⁸ describes a supervised segmentation method to be used in tandem with the already developed unsupervised segmentation. The supervised segmentation adds detection of known obstacles to the segmentation process. This could be used to reduce the processing required for the stereo matching algorithm, which currently takes 51ms.
3. Additional algorithms can be used to further refine the disparity map. One such method is present here⁴⁵.

4.4.3 DGPS

Based on the results that have been obtained from the DGPS so far, some conclusions have been drawn about the performance of the system. The first conclusion is that due to the worst-case error of the DGPS and the position accuracy constraints of the competition, DGPS cannot be used as the sole source of position data for the vehicle. The data from the DGPS must be combined with data from other sensors in order to improve the accuracy of the vehicle's position measurements. The vehicle is currently outfitted with wheel encoders and a digital compass to provide the extra data that is needed to boost the accuracy of the vehicle's position measurements. However, at the time this report was written, the data fusion process is incomplete. Once it is completed, it will provide a much higher level of accuracy than any of the individual sensors could provide by itself.

Another conclusion that has been drawn from the DGPS data is that the environment of the DGPS receiver definitely plays a part in the accuracy of the measurements. The DGPS data taken from

⁴⁵ (Murray and Little 2000)

the relatively open WPI athletic fields has lower error values than the previous data sets taken in a more obstructive environment. Open areas allow the DGPS to make a larger number of strong connections with the GPS satellites, which in turn provides a higher level of accuracy in the returned data. The altitude of the DGPS receiver, as well as its proximity to the vehicle's wireless router needs to be researched further to determine if they will affect the accuracy of the position data.

4.4.4 IMU

Even though the vehicle will not be equipped with an IMU for the 2010 competition, future teams using the vehicle may wish to add one for future competitions. So analyzing the results obtained from the IMU is still an important part of the sensors research. During the limited amount of time that the IMU was available, several conclusions were drawn from the data it returned. The first data analysis was performed on the heading data returned from the IMU. Overall, this data has been reliable and consistent across several data samples, with minimal levels of noise. This makes the IMU valuable because it could potentially replace the compass currently installed on the vehicle, while also giving another set of data that could be used to indirectly determine the vehicle's location. This would provide another set of data to integrate with the DGPS and encoder data, and would increase the accuracy of the vehicle's position estimation.

The second data analysis was performed on the acceleration data returned from the IMU. Initial testing with this data has shown that it is very noisy, although this is due mainly in part to the sensitivity of the sensor. Also, there are some issues with DC biasing in the acceleration values that have led to inaccurate velocity measurements when the acceleration values are integrated. The DC bias in the signal adds up over the course of the integration and creates larger and larger errors as the data sample progresses. However, once the DC bias in the acceleration is subtracted, the acceleration and velocity data both start to look more representative of the physical movements of the IMU during the data sample. Another problem has arisen from this however, in that a method for effectively removing the DC bias while leaving the signal information unaltered is difficult to obtain. With more analysis and more time, it is definitely possible that the IMU could prove to be a valuable addition to the vehicle's sensor system. However, given the time constraints of the project and the low availability of the IMU, it appears that it won't be ready for practical use on the vehicle until next year's competition.

5: Controls System Design

The control system of any autonomous vehicle is one of the most important components as it acts as a brain and provides artificial intelligence. Prometheus is designed to perform multiple challenging tasks like object detection, path finding, and localization and mapping all at once in order to successfully qualify and compete in the IGVC. Furthermore, the team also decided to develop a graphical user interface (Control Center) to assist the development, testing and manual control of the platform.

5.1 Research

Since Prometheus is WPI's first intelligent ground vehicle competing at the IGVC, the team had to look into the reports of the past year's competitors. Most of the teams had a similar software technology choice while the hardware solutions varied amongst teams.

5.1.1 Hardware

In order to process sensor data and make decisions, the vehicle requires an onboard control system. The control system must be able to support all of the vehicle functionality outlined in the requirements. For embedded computing, there are many options for control system boards. For Prometheus the team researched various desktop computer control methods including Mini ITX motherboards, full Mini ITX systems, Micro ATX motherboards, and ATX motherboards. In addition the team researched laptops and embedded control systems such as the PC104 single board computer form factor and the National Instruments Compact RIO control system. Advantages and disadvantages of each researched system are outlined in Appendix C: Hardware Control System Research.

A decision-design matrix was used to evaluate the options. The criteria used for the matrix was power consumption, price, size, main processor speed, and graphics processing ability, modularity, and interfacing capabilities. These criteria have been developed based on initial understandings about the control systems and are presented in Table 13.

The motherboard chosen was the ATX EVGA E758 Intel X58. It is accompanied by an Intel Core i-7 920 2.66Ghz CPU, 6x2GB of G.SKILL DDR3 1600, an NVIDIA GeForce 210 GPU for onboard video, the NVIDIA Tesla C1060 for GPGPU processing, and a 320GB Western Digital HDD.

Criteria	W	Mini ITX	Micro ATX	ATX	SBC PC104	Laptop	NI cRIO	Mini ITX System
Power	4	6	4	4	7	8	4	6
Price	8	7	5	5	5	7	1	5
Dimensions	1	9	9	4	9	6	7	8
Processor speed	7	6	8	8	3	5	6	6
Graphics processing	9	1	8	8	3	8	3	1
Modularity	5	3	6	8	8	3	9	3
Interface	7	4	5	8	5	3	6	4
Total		183	258	284	200	237	187	166

Table 13: Control System Matrix

National Instruments' NI-cRIO controller has been used commonly in robotic applications to interface motors and motor controllers as well as some sensors. It can accompany the main board computer in path calculations, navigation tasks and image processing which is performed on the NVIDIA's Tesla 1060c GPGPU and only the obstacle distance and line information is sent to the computer for further processing.

The team decided to use a GPGPU for image processing as it is able to perform mathematical calculations much faster than a CPU. This is a design innovation in our approach. To show that GPGPU is an improvement the team ran speed and time image processing benchmarks using a regular desktop CPU⁴⁶ and the NVIDIA Tesla C1060 GPGPU. Results of the benchmarks are shown in Figure 65 and Figure 66.

⁴⁶ EVGA E760 Classified Motherboard, 3x1GB Corsair DominatorDDR3 , Intel Quad Core I7 920 2.66Ghz CPU, eVGA GTX280 GPU, NVidia Tesla C1060 GPGPU

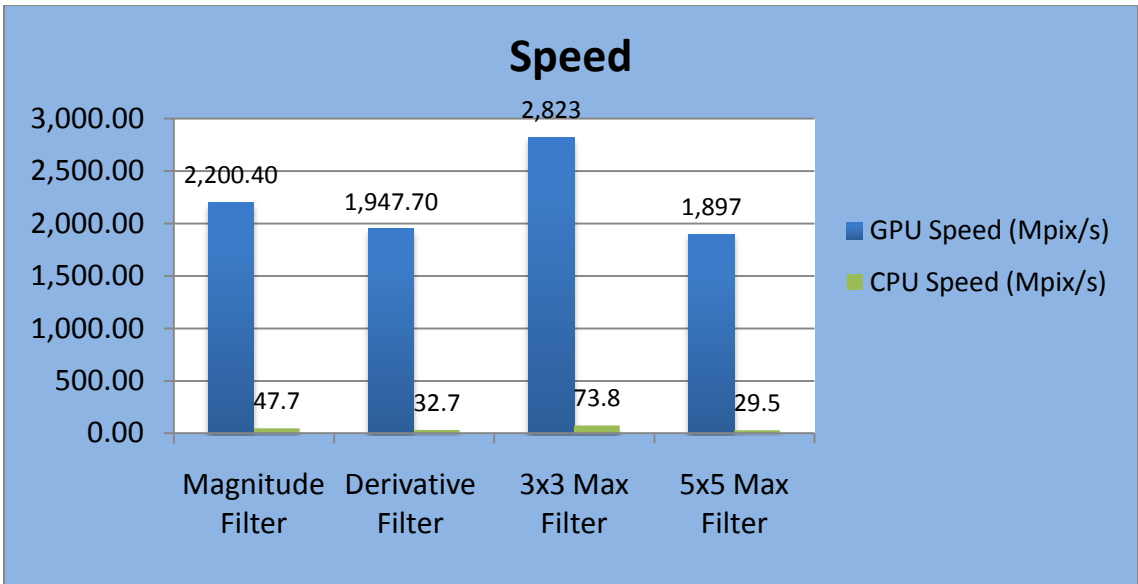


Figure 65: GPGPU vs. CPU speed

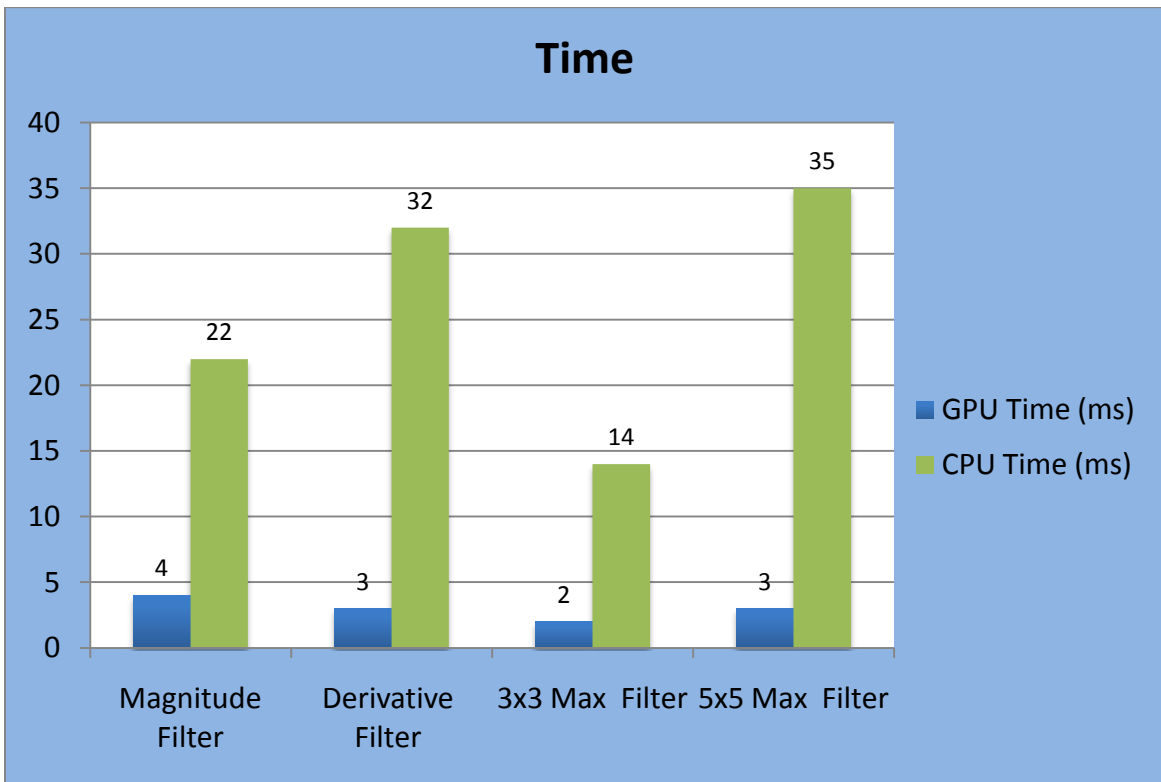


Figure 66: GPGPU vs. CPU image processing times

5.1.2 Software

As mentioned above, Prometheus has to perform multiple complicated tasks, calculations and interface with the cRIO and the Control Center which requires low level access to memory management and networking capabilities.

5.1.2.1 Programming Languages

The controls team's knowledge base included the following programming languages:

- C/C++
- Java
- Perl
- Objective-C

The team chose two languages, C/C++ for the development of the main board software and Java for the Control Center implementation. C/C++ is a very powerful, low level language used in most of the robotic applications⁴⁷ while Java is developer oriented providing garbage collection as well as memory management features.⁴⁸ Also, all JAUS protocol implementing libraries are in C/C++, more details about that can be found later in this section. Eclipse IDE⁴⁹ was chosen for the development environment of the main board software system because everyone on software engineering team was familiar with it and did not require any time to learn it.

There were a few options of development environments for the Control Center including Eclipse IDE and NetBeans IDE⁵⁰. The main difference between Eclipse and NetBeans is that NetBeans provides a graphical drag-and-drop development interface for desktop applications. Since the Control Center is not an essential part of the project the team did not want to invest more time than necessary and the NetBeans IDE was the best solution.

5.1.2.3 Path Planning

Throughout the project the team wanted to be as innovative as possible and it wasn't easy since the competition already existed for 18 years. However, path calculation is one field where innovation can be easily achieved. After some research the team came across this article that talked about path

⁴⁷ (cplusplus.com n.d.)

⁴⁸ (Oracle 2010)

⁴⁹ (Eclipse 2010)

⁵⁰ (NetBeans 2010)

calculation using tentacles⁵¹, which has never been used before in the IGVC. This method was invented by Felix von Hundelshausen, Michael Himmelsbach, Falk Hecker, Andre Mueller, and Hans-Joachim Wuensche and first used applied at both the C-Elrob (European Land Robot Trial) 2007 and the 2007 DARPA Urban Challenge. It uses a local ego-centered map of the environment and sets of tentacles to find the best possible path. The method is simple yet powerful and innovative and was chosen by the team as the main path planning algorithm. Other methods included similar ideas of path planning using a grid based map and finding best drivable grid cells but they were used in the competition before.

5.1.2.2 Network

In order to establish communication between the Main-Board computer and the cRIO, there were several options available. One of the communication methods the team looked at was making a direct connection between the two systems. The downside to this method was not having the ability to add more systems into the network. The team also considered using a wired Ethernet Hub. A Hub seemed like a viable option because it allows having more than two systems. The only issue with using a Hub is that every system would need to be wired. Thus, it was decided that the best method would be incorporating a wireless router onto the vehicle. A wireless router allows the vehicle to communicate with any system logged in to the network supporting both wired and wireless methods. The router also allows the team to incorporate an external GUI for controlling and debugging purposes. Once the decision was made, the team began looking into the requirements for completing the JAUS challenge in the IGVC; which is the only challenge that requires external communication with another system. The JAUS challenge requires the vehicle to connect to a Judge's station through the use of an external network. Since the vehicle already boasts its own network, the team had to figure out how to bridge the external network with the internal network. To solve this problem, a wireless adapter was chosen for the main-board computer. This adapter enables the vehicle to connect to two different networks simultaneously, one using the wireless adapter and the other through the integrated Ethernet jack on the Main-Board. Bridging two networks is possible through the editing of the Internet Protocol (IP) Address forwarding tables that are managed on every computer. Since the platform is Linux-based, the supplied IPTABLES program is capable of bridging the two networks.

⁵¹ (Felix von Hundelshausen 2008)

5.1.2.4 JAUS

There are several open source tools and libraries that provide JAUS Reference Architecture and Services based software solutions including JAUS Tool Set, OpenJaus, Junior Middleware and JAUS++.⁵² According to the competition rules only the Services based approach is supported and will be valid in the competition. Out of all options only the Junior Middleware and JAUS Tool Set are Services based open source implementations of the JAUS protocol.⁵² However, JAUS Tool Set is currently in a closed beta testing phase and will only be available to public the 20th of May. Since the competition is in June the latter option was eliminated. The only other option was to use the Junior Middleware library which is what the team did. Thorough the research the team also looked at the OpenJaus library which is a Reference Architecture based JAUS implementation. Although not supported by the competition it was still a useful implementation of parallel system architecture. The team decided to use it for the internal system architecture because of its ability to parallelize multiple system functionalities. It also came with an easy to understand example that could directly be incorporated in the main software architecture.⁵³

5.2 Development

For the vehicle's control system a distributed system approach is used. This system consists of the National Instruments cRIO and an ATX style main-board computer. Using this approach, the responsibilities of the systems are divided to allow for parallel development and parallel execution of tasks which increases development and system performance. A diagram describing the system can be seen in Figure 67. In the distributed system the cRIO is responsible for sensor interfacing and sending motor move commands to the motor controllers. Two-way communication occurs between the cRIO and the main computer. The computer is responsible for image processing, path calculations, and maintaining the vehicle's map.

⁵² (USARIC 2010)

⁵³ (kent 2010)

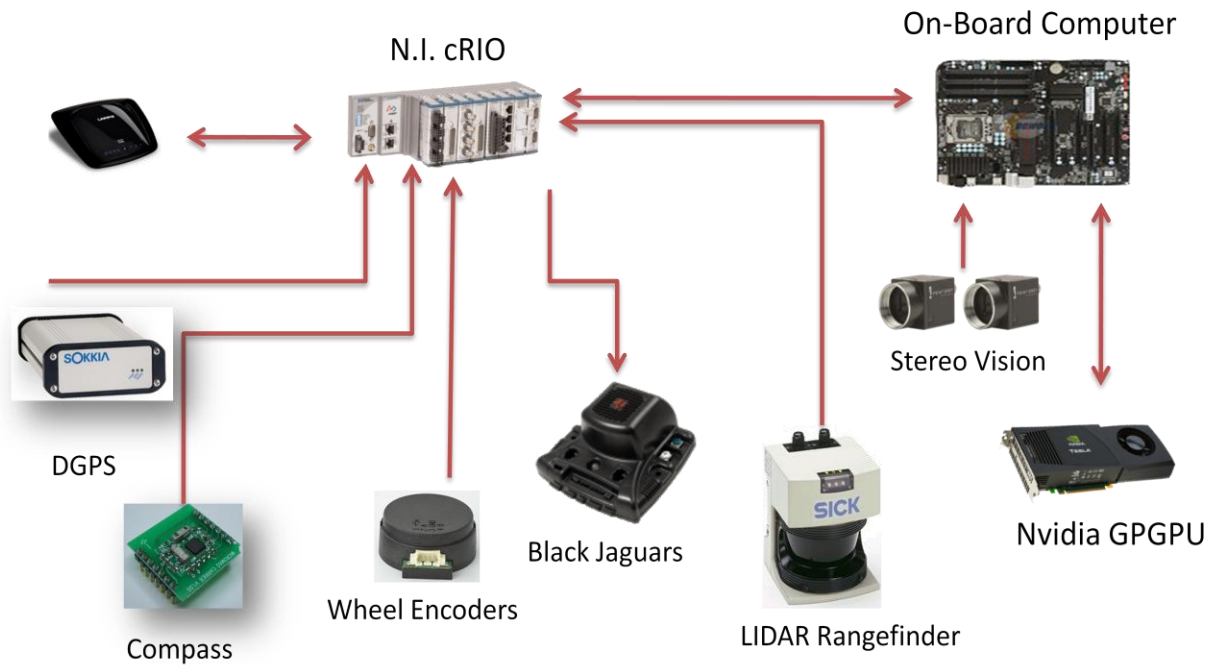


Figure 67: Vehicle Control System Diagram

Since the control system hardware is distributed the software also needs to be. Figure 68 shows the logical separation of the control system and tasks performed by each sub-part.

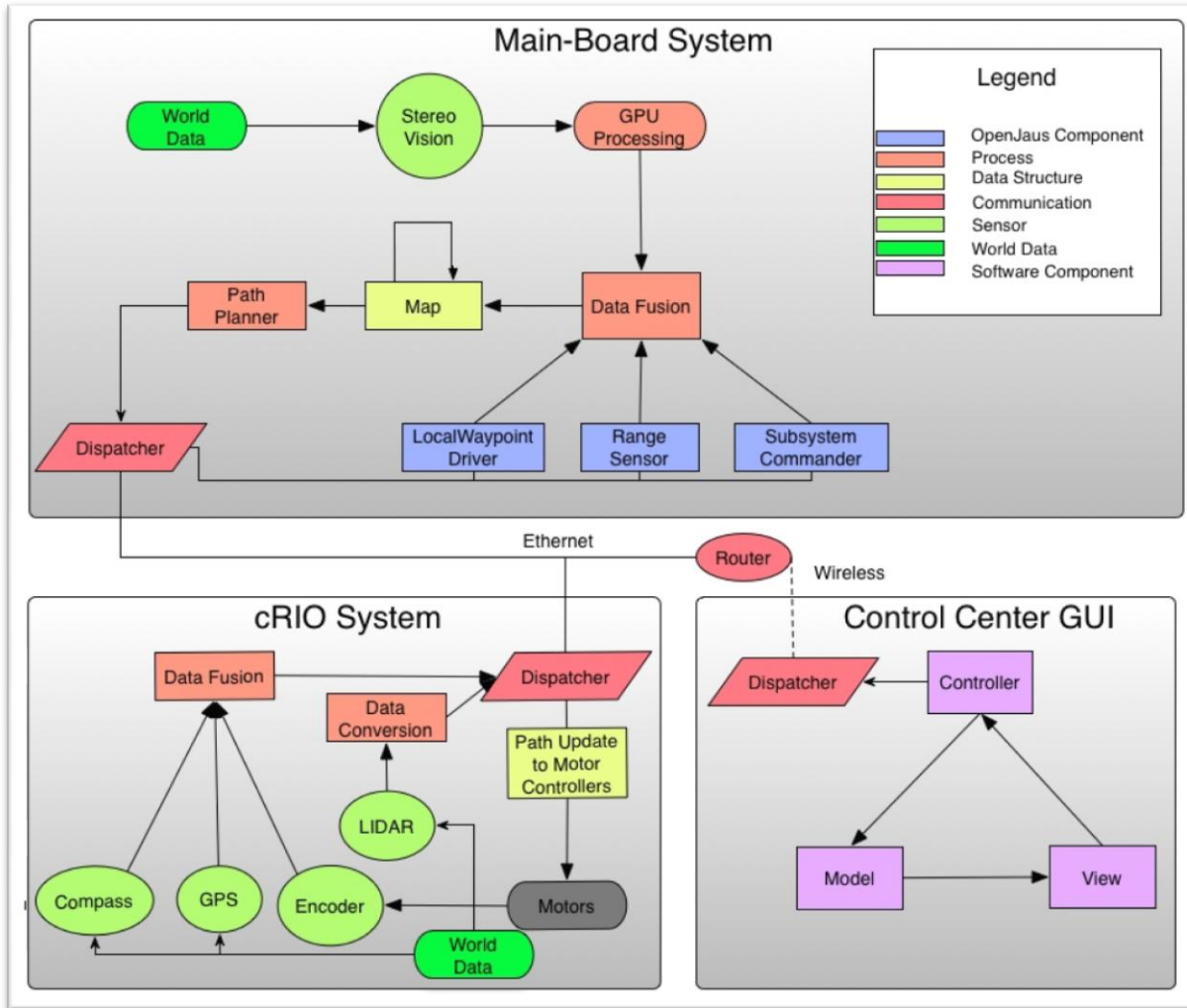


Figure 68: Logical separation of the control system

Splitting the control system into two Main-Board and cRIO Systems increases the performance because motor controls and path finding algorithms are run simultaneously. Addition of the Control center allows faster testing environment.

5.2.1 NI-cRIO

The control system is divided into two parts, the onboard computer and the National Instruments cRIO. The cRIO is responsible for interfacing with all of the vehicle's sensors and the motor controllers.

5.2.1.1 Hardware

The unit the team is using is the NI cRIO-9074. This unit has a 400MHz processor that runs the National Instruments' real-time operating system. It also has a 2 Million Gate Field Programmable Gate

Array (FPGA). The interfacing options built-in to the cRIO are two 10/100Mbps Ethernet ports and an RS-232 serial port. In order to add more input/output capability, the cRIO has modules that can be easily connected to the FPGA. The team is utilizing one 24V digital output module, two +/- 10V analog input modules, two 32 port 5V digital I/O modules, and one four port RS-232 module. An image of the cRIO configuration can be seen in Figure 69. A more detailed description of the National Instruments components used can be found in Appendix B: NI cRIO Specifications.



Figure 69: cRIO Unit

The 24V digital output module is used for controlling the power system relays. The 5V digital input/output modules are used for reading encoder information. The analog input modules read in the potentiometer position on the steer wheel, monitor system voltages, and read in the signal from the wireless E-Stop. The RS-232 module connects to the GPS unit, the three motor Controllers, and the two axis compass. One of the cRIO Ethernet ports connects to the vehicle's onboard wireless router. The RS-232 port on the cRIO itself is used to connect with the LIDAR unit.

5.2.1.2 Main Software Program

The software running on the cRIO is divided into two main sections, the CPU program and the FPGA configuration. Both of these programs are written entirely using the National Instruments LabView graphical programming language. The overall project structure along with the CPU program can be seen in Figure 70.

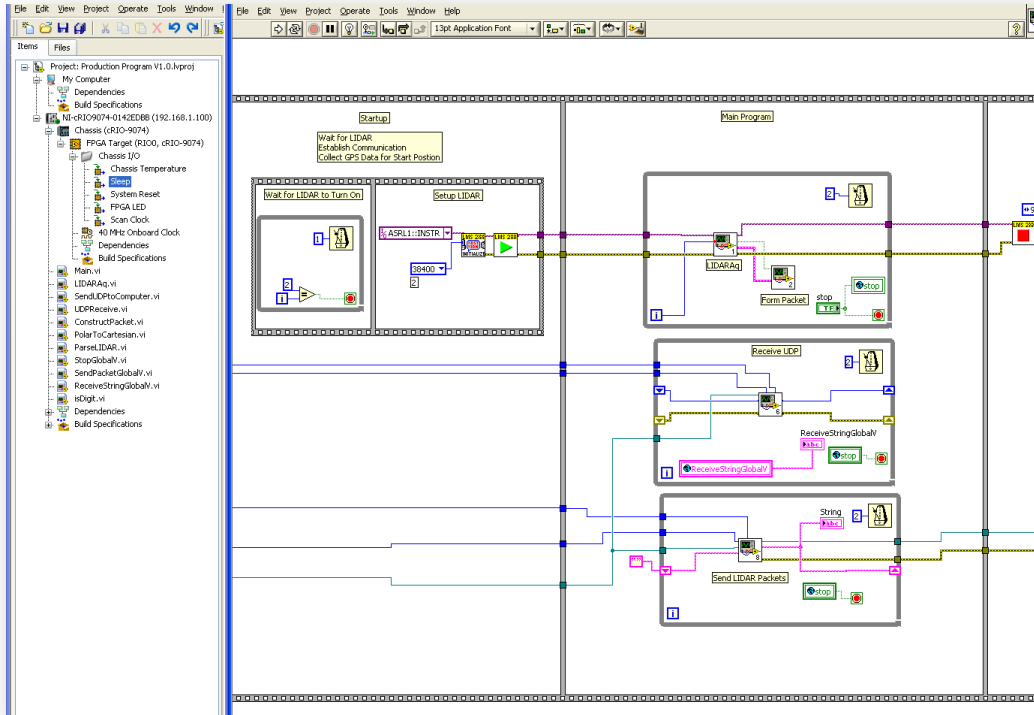


Figure 70: cRIO CPU Program

The CPU program is divided into four stages, initialization, startup, main, and shutdown. During initialization all of the proper startup values and ports are assigned. During startup the cRIO waits for the LIDAR to boot and waits for a handshake to occur between itself and the computer. In order for the handshake to be successful, the cRIO and computer both send UDP packets over Ethernet to each other. When the packets are received by both parties the handshake is complete. The cRIO waits for the information regarding the vehicle's current state from the control GUI. It then moves into the next stage of the program. In the main program, the vehicle has two states. These states are manual and autonomous. Manual mode is used for driving the vehicle with a joystick and autonomous mode is used for vehicle navigation where the vehicle decides its own drive path. In manual mode, the vehicle receives motion control packets from the control GUI, processes them, and drives the vehicle accordingly. When in autonomous mode, the cRIO receives and transmits LIDAR data to the computer. It also uses GPS data and encoder data to determine the vehicle's position and sends that information to the computer. In addition to updating information to the computer, the cRIO receives the calculated path position from the computer and translates it into motor moves. These motor moves get sent to the drive motor controllers.

The FPGA program is responsible for bridging the connection between the CPU and the cRIO modules. It also reduces the load on the CPU by performing calculations. The FPGA handles the PID feedback control loop that controls the position of the front steered wheel. When using the RS-232 module, the FPGA acts as the memory buffer for writing to and reading from the serial ports. The FPGA is also used to read in the quadrature encoders that are used to measure the wheel speeds. An image of the FPGA program can be seen in Figure 71.

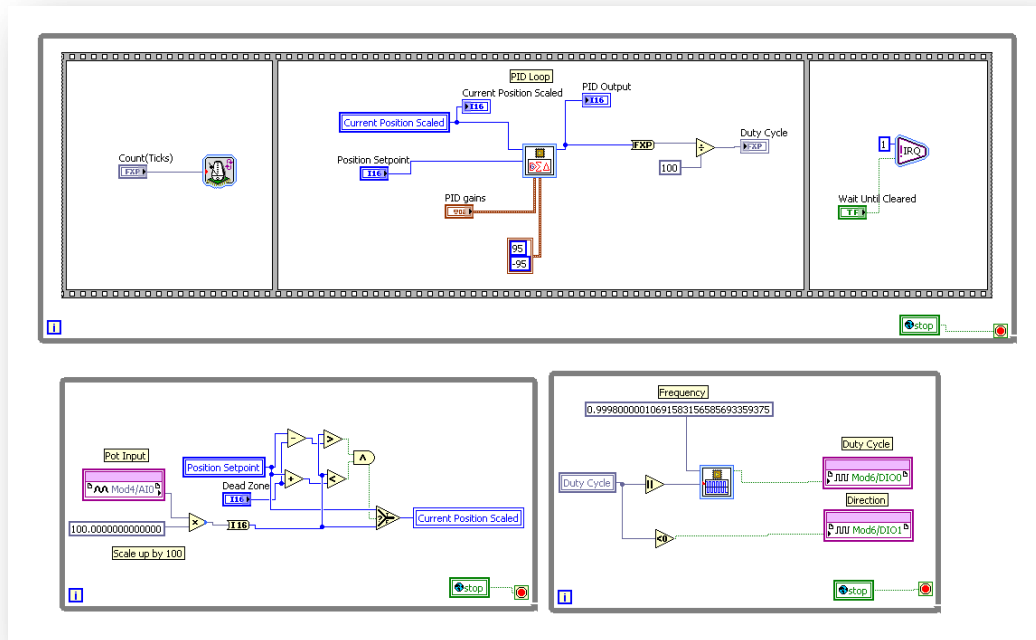


Figure 71: cRIO FPGA Program

5.2.1.3 Sensor and Software Component Integration

The process used for program development was to create separate LabView projects for each major development piece. These pieces include LIDAR interfacing, PID motor control, reading GPS information, network communication, Jaguar motor controller communication, compass, encoder readings, autonomous path calculations, and manual vehicle control. Programming in this manner allowed multiple people to work on separate parts without interfering with each other's work. The disadvantage to the divided development approach is integrating the separate parts together. The team handled this by designating one person to handle the integration. In this way all of the components were integrated while creating minimal problems in the process. A diagram of the cRIO program flow can be seen in Figure 72.

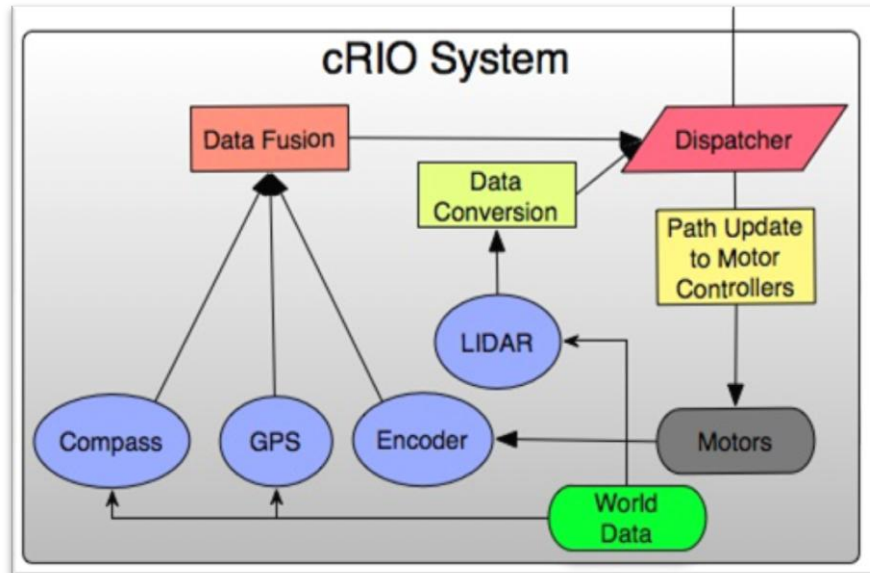


Figure 72: cRIO System Program Flow Diagram

In this diagram, the blue ellipses represent the sensors that return world data to the cRIO. For localization, the compass, GPS, and encoder data is fused to create one vehicle location. The dispatcher in the diagram handles sending and receiving UDP packets to the computer. It sends the information from the sensors and receives path commands that are translated into motor moves. Each of the system interfaces is described in more detail below.

LIDAR

The first development piece worked on was the LIDAR communication. Programming for the LIDAR was simplified significantly since National Instruments provides instrument drivers for the model of LIDAR that the team is using. Using this driver and an example program provided, obtaining data from the LIDAR was achieved in a couple of days. The LIDAR driver establishes a connection with the unit through RS-232 and retrieves data packets. The packets are parsed and converted from polar coordinates into Cartesian coordinate form. The Cartesian coordinates are sent to the onboard computer so that they can be used to update the map.

PID Motor Control

PID motor control was implemented on the cRIO FPGA to control the front steered wheel of the vehicle. Position control was implemented using potentiometer feedback. The angle of the front steer wheel is based directly off of the current speed of the drive wheels. These speeds are determined by the encoder feedback. Since the PID is running on the FPGA, the load on the processor is significantly reduced and the CPU is only required to give the FPGA the wheel position set point.

GPS Localization

The Sokkia Axis GPS receiver is interfaced through a serial port on the RS-232 module. The incoming GPS data is collected and converted to the UTM grid format so that it can be used for vehicle localization. The first GPS coordinate received is saved and called the origin of the vehicle. All coordinates after that have the origin subtracted from them so that the local vehicle position is always known. The GPS data is also combined with encoder and compass data that is read into the cRIO. The compass data provides two axis magnetic heading information and is interfaced through an RS-232 port. The quadrature encoders are read in through the cRIO's FPGA. The local position calculated from this data is sent to the computer so that the local map can be updated.

Networking

The cRIO and main computer maintain communication with one another by sending User Datagram Protocol (UDP) type Ethernet packets to one-another. The cRIO is responsible for updating the computer with position, heading, speed, battery voltage, and LIDAR data. All of this data is sent asynchronously at different time intervals. The packets are sent as soon as new data is available and the computer uses them to update its map.

Motor Controllers

The Jaguar motor controllers utilized by Prometheus employ a CANbus communications protocol to allow the host system to access the features of the Jaguar, such as closed loop speed/position control. The CANbus also allows Jaguars to be daisy chained together; meaning that only one connection to the host system is needed to drive up to 64 Jaguars, as each Jaguar is assigned a unique 6bit device ID. On Prometheus, the Jaguars are connected to the cRIO, which handles generating command messages to send to the Jaguars and parsing any data sent back.

The command structure for the Jaguars utilizes a 29bit message header split into 4 bytes. There are then up to 8 optional bytes of data, the contents of which is defined by the specific command. Messages returned from the Jaguars follow the same structure. The message header contains the following information:

- 5bit device type identifier
- 8bit manufacturer identifier
- 6bit API class identifier
- 4bit API command identifier
- 6bit Device ID

This data is split into the 4 bytes using the structure:

- Byte3:
 - Bits7:6 – Padding
 - Bits5:0 – Device Type ID
- Byte2:
 - Bits7:0 – Manufacturer ID
- Byte1:
 - Bits7:2 – API Class ID
 - Bits1:0 – API Command ID Bits3:2
- Byte0:
 - Bits7:6 – API Command ID Bits1:0
 - Bits 5:0 – Device ID

The device type and manufacturer ID are always 0x02 and 0x02 for the Jaguars (In CANbus, this represents a motor controller made by Texas Instruments). API Class is an identifier which defines the 'type' of command that is being issued. The Jaguars define 7 API classes: Voltage Control, Speed Control, Position Control, Current Control, Status, Configuration, and Acknowledge, numbered 0x00 through 0x07 respectively. API command identifies which command within an API class the message is calling. Each class has its own list of commands unique to that class, such as Voltage Set (0x02) or Speed Enable(0x00). In addition to the information contained in the message header, each message can contain up to 8 bytes of data, the size and content of which is defined by the individual command.

However, Prometheus' cRIO does not have CANbus capabilities, and must communicate to the Jaguars using a standard RS232 serial communication bus. The Jaguars provide an accommodation for this situation by allowing the first Jaguar in the daisy chain to act as a bridge between the serial bus and CANbus. Messages are still sent using the same format as when communicating over CANbus, which some important format changes. First, there are two additional bytes placed at the start of the message header: a 'start byte' which always contains 0xFF, and a message length byte, which contains a number representing the length, in bytes, of the message including the 4 byte CANbus message header and all data bytes, but not the start or length bytes (so valid numbers are always between 0x04 and 0x0C). Also, the data is all sent in little endian format, which must be taken into account as the cRIO stores data in big endian format.

To facilitate communications between the cRIO, four primary VIs were created. The first generates a properly formatted CANbus message header when passed the relevant data fields. The second VI takes a CANbus message header and the number of bytes of data the message will contain as the inputs, and generates a properly formatted Jaguar serial message header. With these two components a library of commands utilized by Prometheus was generated which take only the device ID and relevant data as inputs. New commands can also be readily added. To parse the commands returned by the Jaguars, a message decoder VI, performs the reverse operation of the CANbus header generator, extracting the relevant data. This is passed to a command parsing VI that uses a case statement based on the API class and command to determine how to format the returned data and where to send it. Using this system, parsing code for any class/command combination can be easily added.

Autonomous Control

The vehicle will automatically switch into autonomous mode when it starts receiving path packets from the main computer. These path packets are in the form of a motor speed and the radius of the path the vehicle should take. From this, the left and right wheel speeds are calculated and sent to the motor controllers. These speeds are updating at approximately 15Hz so the path can be consistently followed.

The main computer sends path information to the cRIO in the form of a UDP packet with the following syntax: 'UDP PTH 0.000 0.000 0.000'. The packet contains information on the current tentacle to be driven upon: its radius of curvature, the desired linear velocity, and its drive length. This packet is converted into a string on the cRIO and parsed.

Once this information is parsed into a usable format, it is passed to a VI which calculates the left and right motor velocities needed to follow the path. The logic in the VI operates under the assumption that a tentacle is essentially a segment of a circle. Therefore, the following equations hold true:

$$v = \frac{V_R + V_L}{2}$$

And

$$R = \frac{L V_R + V_L}{2 V_R - V_L}$$

Where v is the linear velocity of the robot, R is the radius of curvature that the robot will traverse, V_R/V_L are the right and left wheel velocities, and L is the separation between wheels. By manipulating these two equations, the left and right wheel velocities can be solved in terms of v , R and K , where:

$$K = \frac{\frac{2R}{L} + 1}{\frac{2R}{L} - 1}$$

$$V_R = \frac{v}{K} \text{ if } (R > 1), \text{ else } v$$

$$V_L = v \text{ if } (R > 1), \text{ else } v * K$$

Once the two velocities are calculated, they are passed onto the velocity controller which updates the current motor speeds on the motor controllers.

Manual Control

A manual control program was implemented in the cRIO so that the vehicle could be controlled at any time with a joystick. The program was designed with two states, one of them being the manual state. In order to drive the vehicle in manual mode, a user can connect a joystick to a laptop computer running the Prometheus Control Center. If the laptop is connected to the wireless network on the vehicle, it will send the joystick information to the cRIO. When these packets are received, the vehicle will automatically switch to manual mode and sets the motor speed according to the joystick position. Manual mode also implements a watchdog timer. The timer monitors the incoming joystick packets and if the packets stop coming in for 25ms, the motor speeds get set to zero. This functionality is important for the case where the robot is driven out of the range of the wireless signal. Without the watchdog timer, the motor speeds would remain at whatever their value was the last time a packet was received. The program is also configured such that any point in time the vehicle state can switch between manual, autonomous, and navigation based upon the type of packets it is currently receiving.

5.2.2 Main Board Computer Software Architecture

The Software Architecture for the vehicle was an essential part for the software development cycle. The architecture was constructed using the OpenJaus library which provided a base implementation of the JAUS architecture. All other functionality that was not provided by the OpenJaus was created and integrated into the library. Extending OpenJaus included adding communication with the CRIO and Command Center, path planning and the stereovision code. The following diagram shows the package architecture that was designed and implemented.

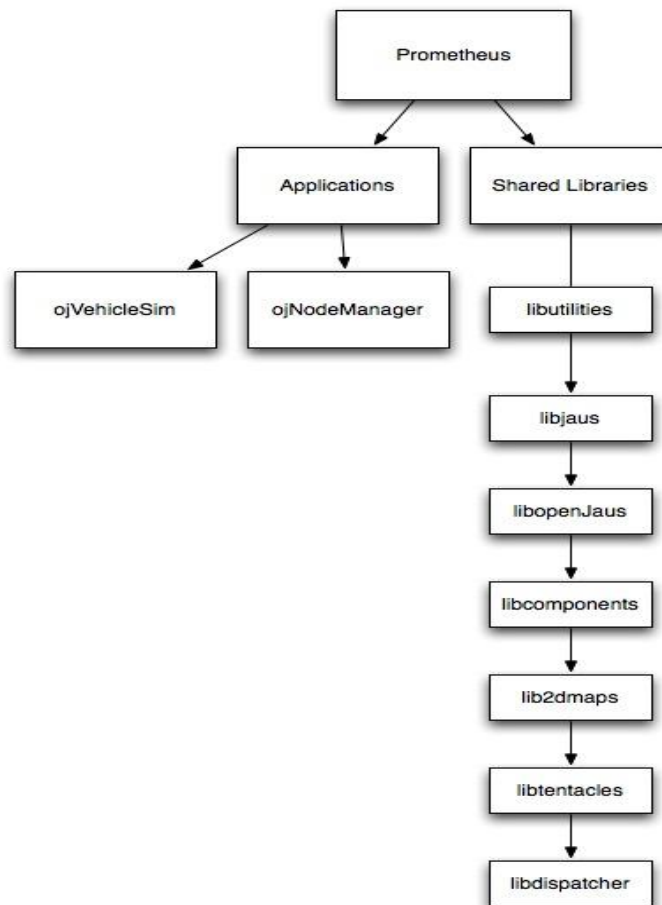


Figure 73: Software package diagram

The ojVehicleSim is the main program in which the vehicle depends on. The vehicle simulator is the application that implements all of the shared libraries listed in Figure 74. The shared libraries listed above, except for libjaus and libOpenJaus, were created by the team to handle the necessary cross-dependencies amongst the other libraries. Figure 74 below contains the list of libraries and their dependencies to other libraries.

Library	Description	Dependencies to other libraries
libjaus	J AUS standard framework	None
libOpenJaus	Base J AUS implementation	libjaus
libcomponents	Set of currently used J AUS components	libjaus,libopenjaus
lib2dmaps	Collection of map objects	libjaus,libcomponents
libtentacles	Path finding and tentacle navigation	libjaus
libdispatcher	Networking objects	libjaus,libopenjaus,libcomponents lib2dmaps,libtentacles
libutilities	All other objects	None
libstereovision	CUDA image processing algorithms	Libdc1394,libraw1394,

Figure 74: Libraries and their dependencies

It was important to factor out the necessary features into multiple libraries to achieve system functionality and ease of integration. Using the shared libraries also allowed the software development team to create specific parts of the software and test them independently before integrating with the rest of the architecture.

5.3.2 Internal System Network

The communication between the Main-Board computer, the cRIO and the Command Center was achieved by using a WIFI router and a WIFI adapter for the main-board computer. Having a router on the system allows the vehicle to broadcast its own network, thus enabling wireless communication from the Command Center to the vehicle. The router is needed to wirelessly login into the vehicles Main-Board computer for necessary modifications. The WIFI adapter allows the Main-Board computer to be connected to two networks simultaneously. This feature was necessary to take part in the J AUS challenge, for which the vehicle is required to communicate with an external wireless network. A small script was written to manage the IPTABLES to rout any internal network messages on the on board router to an outside network. The entire networking software was compiled into a dispatcher library.

To send/receive messages that can be supported in the cRIO, Main-Board computer and the Control Center, the team had to define a custom messaging protocol. The message protocol provides a standard set of header tags that are used to easily identify the data being transferred. Each packet contains a sender, command and information tags described in Figure 75.

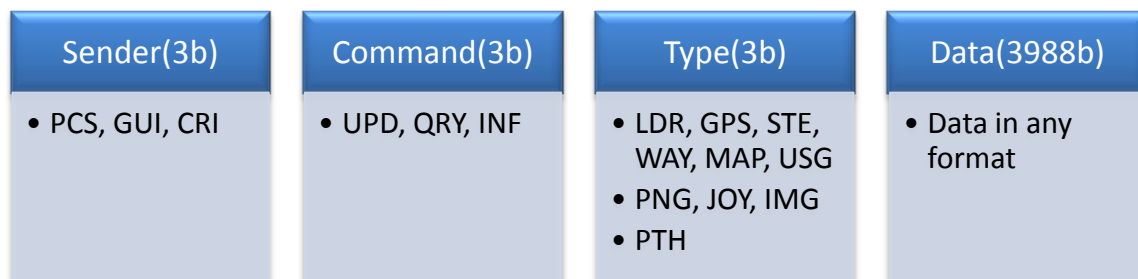


Figure 75: Network packet diagram

The sender tag is used in order to identify the sender of the message. Using the sender tag allows the software to not worry about keeping track of the specific IP Addresses. The command tag lets the dispatcher to know which thread will handle the message. The type describes what type of data will be in the data portion of the message. This message protocol provides a flexible standard on how to interpret messages and makes it simple to extend the dispatcher and add new messages.

The message dispatcher was designed to be multi-threaded using POSIX threads. In total there are three threads that are essential for the dispatcher to provide the appropriate functionality; the receiving thread, update thread and the query thread. The receiver thread is created after initialization of the main program. Its main purpose is to handle all incoming messages without stalling the rest of the system. It also reduces the probability of network buffer overflows because the receiver does not do any parsing, interpretation or updating of data. Upon receiving a message, the receiver thread inspects the command tag. If the tag is “UPD” then the receiving thread spawns the update thread in order to update the proper JAUS components with the newest data. Otherwise, the receiving thread hands the message to the query handler thread. The query handler thread was created to handle all non-JAUS specific network operations. This thread includes the handling of pings⁵⁴, data requests and updates to the Command Center. In Figure 77Figure 75, the flow diagram of the dispatcher’s message handling system is shown.

⁵⁴ Ping is a computer network administration utility used to test whether a particular host is reachable within the network.

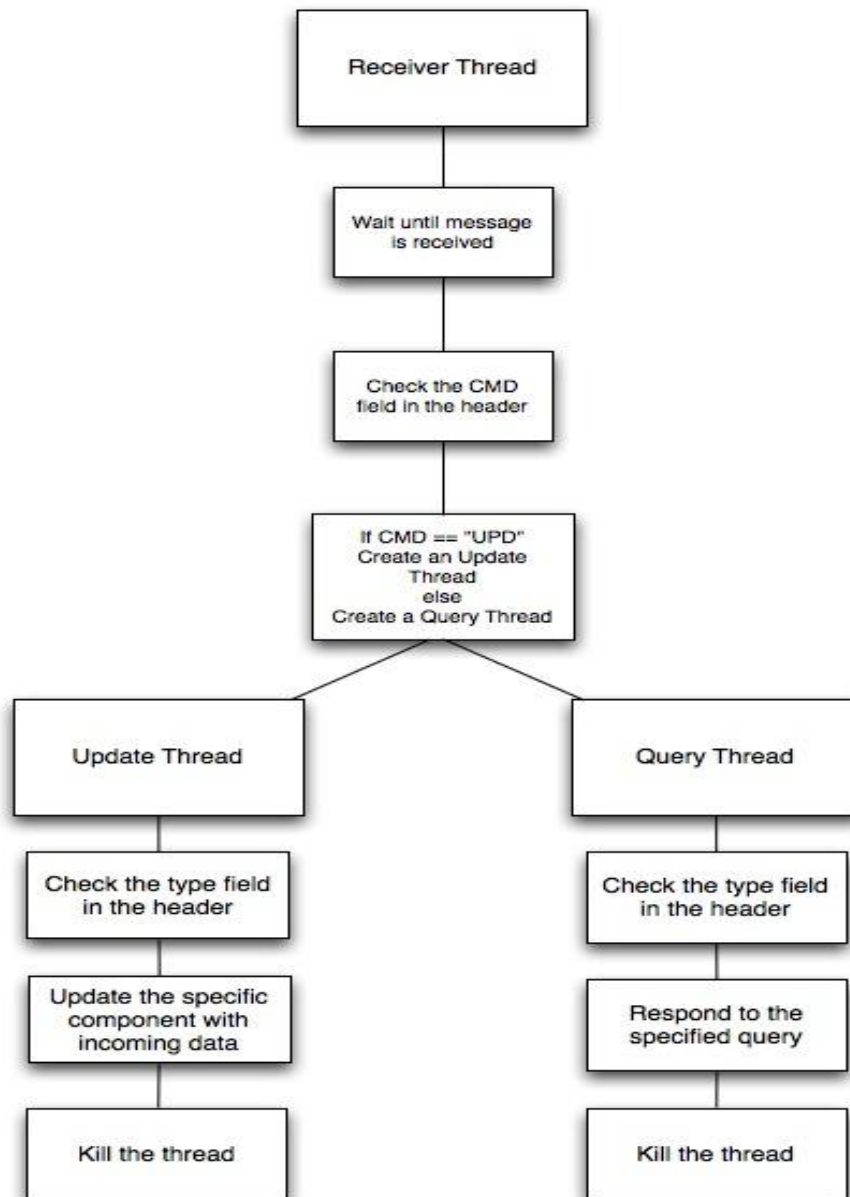


Figure 76: Network Thread Diagram

5.3.3 Navigation

As mentioned in the introduction, in order for Prometheus to qualify for the competition it must be able to detect and avoid obstacles as well as follow lines. Navigation is the main component of the artificial intelligence of the vehicle and it must be designed and implemented and tested comprehensively. It consists of mapping, localization and path planning.

5.3.3.1 Mapping

Since path planning is based on the map it is important to have the most accurate representation of the surrounding terrain and obstacles. The map used for path planning is a top view 2D probability map that was implemented using a 201x201 two dimensional array of grid cells where the location and the heading of the vehicle is always the same. Each cell covers 10x10cm of the terrain and contains information about the area of the ground it covers including the probability of an obstacle being at each grid cell, the type of the obstacle (cone, line, pot hole, etc.) and whether the cell has been updated since the last map update. Using this probability map greatly reduces the complexity of incorporating both LIDAR data and stereo vision data. Since the vision and LIDAR systems run at different speeds it is essential that the map updates can be asynchronous.

Initially, each cell is assigned a probability of 0.5, since we don't know whether it contains an obstacle or not. Once the first LIDAR or stereovision data reading is taken the probabilities of the obstacle containing cells and their neighbor cells are recalculated using Gaussian distribution described below.

$$e^{-\frac{(x-h)^2}{2\sigma^2} - \frac{(y-k)^2}{2\sigma^2}}$$

Where x and y are XY coordinates of the neighbor cell, h and k are the XY coordinates of the cell obstacle containing cell and σ represents the spread of the distribution which is 2 in this case. The probability of cells in between the vehicle and the cell containing an obstacle increases less than probability of the cells hidden by the obstacle. Table 14 shows an example of the probability map where red indicates the point of obstacle and green is the vehicle.

0.5	0.5	0.5	0.5	.75	.75	.75
0.5	0.5	0.5	0.5	.75	1.0	.75
0.5	0.5	0.5	0.5	.3	.3	.3
0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	0.5	0.5	0.5	0.5
0.5	0.5	0.5	R	0.5	0.5	0.5

Table 14: Example of a Probability map after an object has been detected

LIDAR data is significantly more accurate than that from the stereovision system therefore it has a greater effect on the probability map. Stereovision data is primarily used for detection of potholes and lines and has a little effect on probabilities of obstacles that LIDAR detects.

Once the initial probabilities are calculated, smoothing is performed over the LIDAR sweep area (180degrees) and probabilities of cells that did not contain an obstacle are reduced. Such smoothing provides error correction for LIDAR and stereo vision data as well as rotation error.

5.3.3.2 Localization using Kalman Filter

The current location of the vehicle was calculated using information returned from the motor encoders, the DGPS receiver, and a digital compass. This information was adjusted and fused together using a Kalman filter, and then sent to the main board computer to be used in map updates and path calculations.

The Kalman filter is a recursive filter that continually adjusts its internal gain to minimize the covariance of the error between what the actual state of the system is and what the Kalman filter predicts the state of the system to be. Over time, the estimations produced by Kalman filter become increasingly more representative of what the actual state of the system is. However, due to the recursive nature of the filter, the accuracy of the estimations is poor during the early iterations of the filter. In the vehicle's navigation system, the multiple streams of data coming in from the vehicle's navigation sensors are combined into one state vector defining the velocity and position state of the vehicle. This state vector is then optimized and smoothed using the Kalman filter to produce a more accurate velocity and position state for the vehicle. The main advantage of using a Kalman filter in the navigation system is that unwanted errors in the vehicle's state vector due to sensor noise and other factors are smoothed out and have a much smaller distorting effect on the desired sensor information. This results in a more accurate measurement of the vehicle's current location, which in turn results in more accuracy in the vehicle's ability to navigate.

As mentioned previously the location and heading of the vehicle on the map is always the same, and as the vehicle moves the map slides and rotates accordingly. The main reason for implementing the map in this way is the path calculation algorithm described next.

5.3.3.3 Path Planning using Tentacles

A tentacle is a line of certain length along a circle of certain diameter that varies based on a vehicle's speed, see Figure 78.

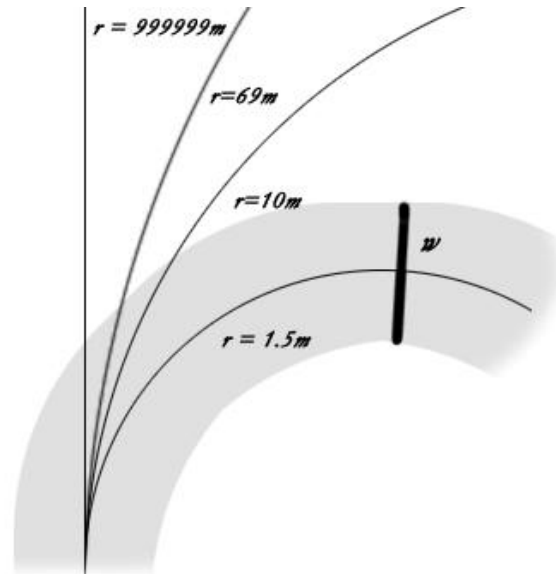


Figure 77: Example of three tentacles and their radii. Support area is in gray.

This method implements a very simple method of path calculation by creating 81 tentacles extending out from the center of the vehicle. Each tentacle has a different length and curvature based on the travel speed of the vehicle. Each tentacle also has a support area which represents the area that the vehicle will be taking as it drives along the tentacle. Travel speeds are grouped into 5 speed sets, each set containing different series of tentacles shown in Figure 78.

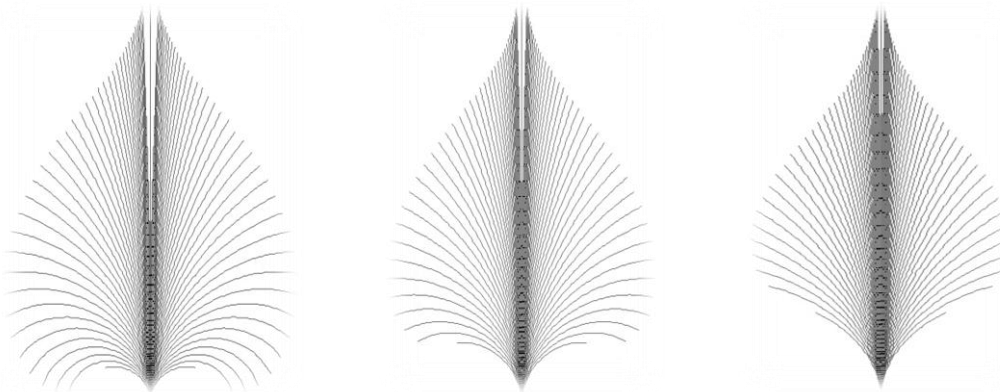


Figure 78: Tentacles of speed sets 1, 3 and 5

With each increasing speed set tentacles curve less hence avoiding rolling at higher speeds. Although calculations show that even driving at maximum speed (2.2m/s) the maximum turning radius (0.77m) is still less than that of the slowest speed set shortest tentacle (1.5m). "Our approach does not accumulate data and uses a local ego-centered grid, thereby avoiding the SLAM problem."⁵¹ One of the reasons this method was selected for path calculation is because it is innovative and was proven to work.

The procedure for creating tentacles was implemented strictly following the paper mentioned above. It provided necessary equations to calculate the curvature and length of each tentacle. However, the support area calculation for each tentacle was done by the team.

Here is the procedure needed to calculate a set of tentacles and their classification area for a given speed set:

1. Calculate length of the shortest tentacle (most curved one) L

$$L = 1.5 + 1 q^{0.1}$$

The reason for exponential factor $q^{0.1}$ is to sample low speeds more frequently.

2. Calculate the radius of the shortest tentacle using the L calculated above.

$$R_j = \frac{L}{\varphi(1 - q^3)}$$

$$q = j/(n - 1) \text{ and } \varphi = 1.2 (\pi / 2)$$

3. For every tentacle (81 in this case) calculate its length l_k and radius r_j for a speed set j and tentacle k .

$$l_k = \begin{cases} l + 20 \text{ m} \sqrt{\frac{k}{40}} & | k = 0, \dots, 40 \\ l + 20 \text{ m} \sqrt{\frac{k-40}{40}} & | k = 41, \dots, 80 \end{cases} .$$

$$r_k = \begin{cases} \rho^k R_j & | k = 0, \dots, 39 \\ \infty & | k = 40 \\ -\rho^{k-41} R_j & | k = 41, \dots, 80 \end{cases} ,$$

4. Calculate number of segments (bins) in each tentacle. Each segment has a certain length along the tentacle (5cm in this case)
5. Calculate coordinates of each bin with regards to global map mentioned above.
6. Calculate support area for each bin calculated above. Support area is just an array of coordinates that represent the width of the tentacle in each bin. The width is greater than the width of the vehicle and that is 4 feet in this case.

Such set of tentacles is created for each speed. Prometheus has five speed sets each with an increasing velocity of about 0.3m/s for each speed set where the slowest speed is 0.4m/s.

5.3.3.4 Finding the Best Tentacle

Every time the map is updated each speed set of tentacles is traversed to find the most preferable tentacle. Initially, each tentacle has a preference of 1. Preference of each tentacle is calculated using one of the two heuristics depending on the mode (the navigation challenge or the autonomous challenge). Before tentacle preference is calculated the drivable length of the tentacle must be found.

Each bin in a tentacle has a weight attribute which is calculated by looking at the support area of the bin. Maximum probability is chosen and assigned to the weight of the bin. If the weight of the bin passes the threshold then the bin is considered good and the vehicle would be allowed to move up until that bin on the tentacle. If the weight is greater than the threshold the bin calculation stops at that point and the next tentacle is then looked at. This process is repeated for every tentacle in each speed set.

Heuristic for finding the best tentacle in any mode of operation takes into consideration the length of the tentacle (tL), the drivable length of the tentacle (dL). Product of the equation $\frac{dL}{tL} * 2 * dL$ is then multiplied with the current preference of the tentacle. The tentacle with the highest preference is picked to be driven next.

Heuristic used in the navigation challenge takes into consideration the distance from each drivable bin (bD) and the distance from the vehicle (wD) to the waypoint. Ratio of the wD/bD is multiplied with the current preference of the tentacle. The higher the ration the more preferable the tentacle is. The most preferable tentacle is picked to be driven next. The latter heuristic was calculated empirically through testing and can be modified if needed.

Such implementation of tentacles and the global map allows quick access (under 0.02s) to a known location in memory to find the best tentacle to be driven. Once the best tentacle is found it is

sent to the cRIO to be converted into motor commands to drive the vehicle accordingly. The speed of the platform can be calculated using this formula and knowing the speed set j the tentacle came from:

$$v_j = v_s + q \cdot 1.2 (v_e - v_s),$$

where v_s is the lowest speed of the platform (0.4m/s) and v_e is the maximum speed of the vehicle (2.2m/s).

5.3.4 Control Center: The Graphical User Interface

To ease the development and testing process of the platform a graphical user interface was written using JAVA SWING technology. It is the Control Center of the vehicle and is capable of sending query commands, mentioned previously. It was important to factor out the necessary features into multiple libraries to achieve system functionality and ease of integration. Using the shared libraries also allowed the software development team to create specific parts of the software and test them independently before integrating with the rest of the architecture to the Main-Board computer as well as control the vehicle using a joystick. It receives vehicle LIDAR data, next drivable path, state updates, computer usage information and camera view which helps monitoring the status of the vehicle during testing and the competition. Control Center can also pause, start and shutdown the main program on the Main-Board computer and change between the different modes of running (autonomous, navigation and manual). It has the ability to configure network port information and send a list of waypoints to be navigated to the Main-Board computer for the navigation challenge. Figure 79 shows a screenshot of the Control Center.

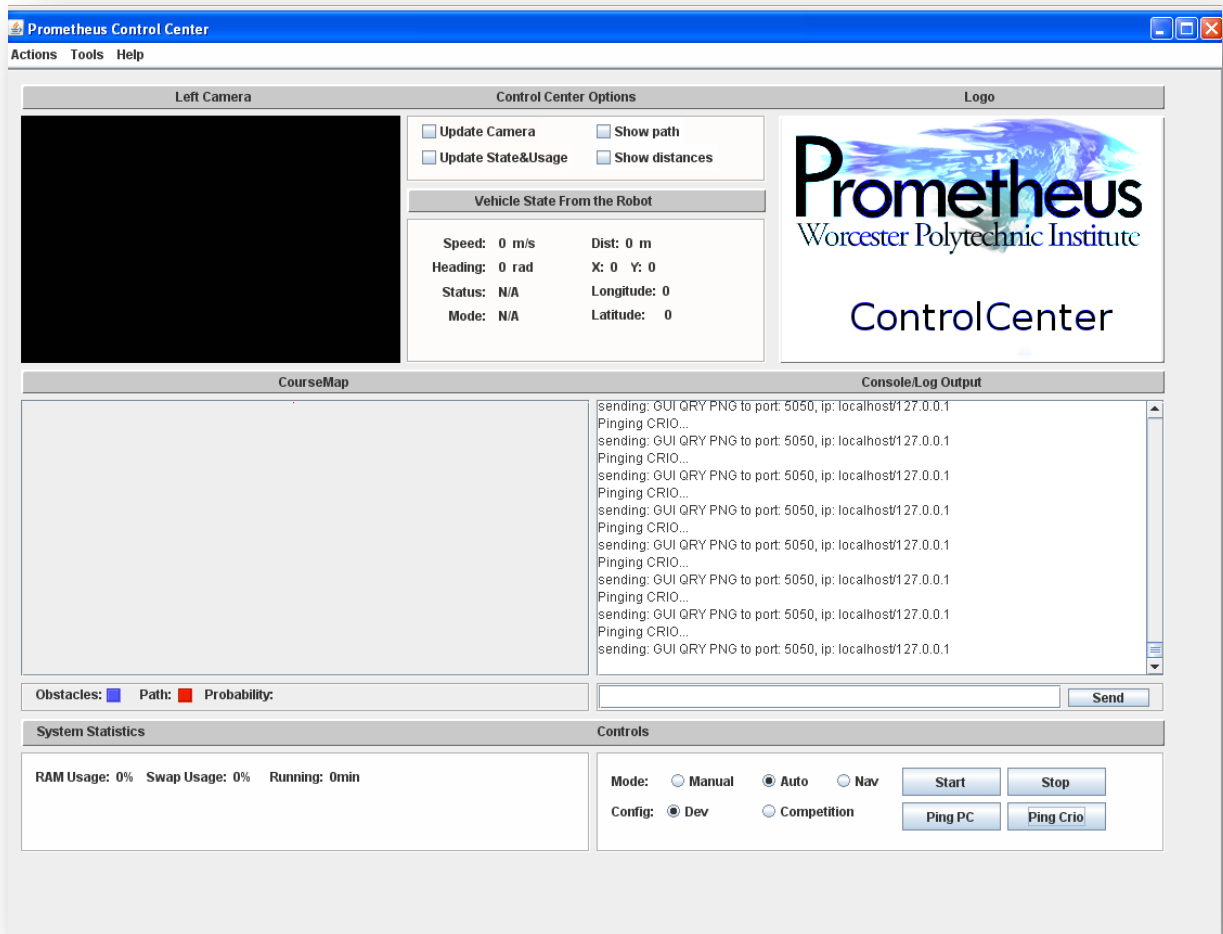


Figure 79: Control Center

5.4 Results

Controls system of the vehicle consists of multiple components each contributing to the system in a specific manner. The next several subsections will go through the results of each of those components.

5.4.1 cRIO

The final cRIO system is functional with all of the basic operations of the vehicle outlined below.

- UDP communication
- Manual vehicle control
- Autonomous vehicle control
- LIDAR data collection, conversion and transmission
- Jaguar motor control

- PID Motor Control
- Reading Encoders
- Reading GPS data and calculating position

Full UDP communication was established with a custom packet structure to allow communication between the cRIO and the on-board computer. The cRIO is able to receive joystick commands from the graphical control center to maintain control of the vehicle during manual operation. It also receives path packets from the computer and calculates what path it should drive along. It communicates with the LIDAR, converts the LIDAR data to Cartesian coordinates, and sends the data back to the computer. It successfully sends motor commands via an RS232 serial connection through the FPGA to the Jaguar motor controllers to set their mode, ramp, and speed. It runs a PID algorithm on the FPGA to control the steered wheel. It successfully reads in the rear wheel encoders and calculates the angle of the steered wheel based on the speed of the rear wheels. It also reads in the GPS signal, parses the packets, converts to UTM grid and determines the local position of the vehicle.

Implementing all of these components used up a lot of resources on the cRIO. The main cRIO program uses 40% of the available program memory and the FPGA utilization can be seen in Table 15.

FPGA Component	Percent Utilization
Slices	17.2%
Flip Flops	15.5%
Total Look-Up Tables	49.5%
Block RAMs	10.0%

Table 15: FPGA Utilization

5.4.2 Networking

The inter-system communication which was set up between the main-board computer, the cRIO and the Control Center turned out to be reliable. Due to having an onboard router with only three clients, packet loss was minimal throughout the system. Since the main-board computer's UDP receiver ran on its own thread, the cRIO was never able to effectively overflow the main-board's buffer. The wireless router also gave the team the capability to remote-desktop into the main-board computer. This allowed the software developers to leave the main-board computer inside the vehicle permanently and continue programming from a remote machine.

5.4.3 Software Architecture

The software architecture was implemented using the OpenJaus library. OpenJaus provided several of the components that the system needed for the competition while other components were created or

edited in order to handle the system specific requirements. In the final software build, the following components were used:

- The Range Sensor (rsensor) component was created to effectively handle the LIDAR sensor data that is returned from the cRIO.
- The Subsystem Commander (subsysCommander) holds the state of the vehicle including its running mode (auto, navigation, manual).
- The Local Waypoint Driver (lwd) is responsible for handling position data sent from the cRIO, keeping track of waypoints and local position information of the platform.

Each of these components were running in parallel and being updated using system's dispatcher. It provided the ability to have asynchronous map updates as well as gave the infrastructure to control the state and mode of the vehicle.

5.4.4 Mapping

The data structure that was used to store the world state captured by the vehicles sensor systems was a 2-d probability map. This map was constructed using a 2-dimensional array of grid cells. Each cell in the map represents a 10cm^2 area of the field. Every cell contains a probability, which is set by the map update process. This probability is represented by a value between 0 and 1, which allows the navigation algorithm to estimate whether an object occupies that certain cell. Initially, all of the probabilities in the map are set to 50%. Once a map update is received, all of the probabilities are updated. The LIDAR operates at 4 Hz and the stereovision system at 15 Hz hence the frequency of the update was set to 15 Hz. This frequency marked the maximum amount of data transfer that could be achieved by the sensor system as well as the minimum frequency before slowing down the fastest sensor.

5.4.5 Path-finding

The path finding with tentacles algorithm, which was described earlier, was tested in the lab environment and in the hallways. There were a few problems with the way the best tentacle was picked. Since the LIDAR is in the front of the vehicle the tentacle did not take the rest of the platform into account. Because of that, the vehicle turned earlier than it was possible for it to fit through. That problem was solved by shifting the probability map up by one meter so that the tentacles are drawn from the back of the vehicle. Waypoint navigation was not tested outside and the GPS signal in the lab is not strong enough.

5.5 Discussion

The distributed system approach using the cRIO and the on-board computer proved to add complexity to the vehicle but it provided many benefits. It allowed parallel development for six people. The computer system allows for high speed video processing with the GPU. The cRIO allows for easy LIDAR communication, easy system interfacing with the removable modules, high speed encoder readings on the FPGA, and a high speed PID controller on the FPGA. The overall performance of the system is satisfactory and could be improved by increasing the updated rate of the LIDAR, faster image processing algorithms and better heuristic for path calculation. Also, it would be beneficial to have a simulator implemented so that mapping and path planning could be done without the vehicle.

5.5.1 cRIO Discussion

Still to be done on the cRIO is using feedback control on the drive motors. The ideal method for doing this is using the built-in functionality of the Jaguar motor controllers. If the encoders are connected directly to the Jaguars, they will do their own PID algorithm for speed control. If this is done, then the encoders will no longer be read into the FPGA and the speed of the motors will need to be retrieved over RS232 from the Jaguars. A Kalman filter for determining the vehicle's exact location needs to be finished and tested. Once this is done, getting the vehicle to navigate to provided GPS coordinates needs to be tested. A simple program also needs to be created for the wireless vehicle E-Stop. This program will read from analog inputs and turn off power to the motors when the appropriate signal is received.

5.5.2 Networking

Many potential issues were avoided by using multithreading on the main-board computer's dispatcher. Although much testing was done, there were many cases which were not covered. After some research, the team became aware that the GPS receiver might interfere with the wireless router's broadcast signal. So far, the network system has proven to be quite robust and flexible. Packet loss has been minimal and total transfers do not exceed any of the hardware capabilities.

5.5.3 Software Architecture

The use of OpenJaus for the system architecture proved to be useful because it provided flexible and easy to extend infrastructure of the system. The use of the dispatcher enhanced the performance and capabilities of the system even more. The only drawback is that OpenJaus has an overhead that is not necessary and adds complexity to the system. In the future, the OpenJaus library could be modified to eliminate the extra load on the system.

5.5.4 Mapping

The 2-Dimensional probability map has been tested in the lab. The distances of objects can be accurately represented within the data structure. One of the improvements regarding mapping and localization is to have a full course map and a list of actions taken by the vehicle so that any run can be simulated and reviewed.

5.5.5 Path-finding

Path-finding using tentacles work well but can still be improved by using different heuristics for preference calculation. Since the system architecture allows modifications to any of its parts other path planning algorithms should be explored.

5.5.7 Control Center

Control Center proved to be very useful in the development and testing phase of the project. Ability to see the LIDAR data and best tentacle really helped the team in debugging. However, it can still be improved by adding the ability to see the whole course map and top 5 best tentacles as supposed to just the most preferable. Adding various sensor readings like battery life, left right motor speeds etc. would help even more. Also, being able to see stereo vision process and images would significantly improve the development and testing of the stereo vision.

6: Power System Design

The design of the vehicle's power system is dictated by the design choices made for the drive, sensor and control systems. The main power draws in the system come from the onboard computer and the three drive motors. The requirements of the power system are to provide the appropriate level of electrical voltage and current for all of the onboard systems such that the vehicle can run for 6 minutes at a time at the competition and for a suitable amount of time for testing. As part of the competition requirements, the power system must also support a manual and wireless emergency stop (E-Stop). The manual E-Stop must be able to kill power on the entire vehicle by pressing a button located on the outside of the vehicle. The wireless E-Stop must kill power to the vehicle's drive motors when activated.

6.1 Research

Before designing the power distribution and emergency stop system, the team performed some research and analysis. A worst case and nominal power consumption analysis was performed to determine the battery capacity and output current. Different battery technologies were researched to determine the best type for the system, and different distribution and emergency stop solutions were investigated.

6.1.1 Power Estimates

The vehicle uses a variety of sensors and motors as well as embedded and non-embedded control and processing systems. Each of these systems needs a power source to operate. The power source will be a DC battery at 24 volts (either two 12 batteries in series or one larger 24 volt battery). An estimate of power use is presented in Table 16.

Item	MAX Power(W)	Nominal Power(W)
Computer	367.8	344
GPS Reviever+ Antenna	1	1
Drive motor 1	1200	600
Drive motor 2	1200	600
Steering Motor	34.7	15
cRIO-Chassis	20	20
cRIO- Modules	28	20
LIDAR	30	20
Camera X2	7.6	7.6
TOTALS	2891.1	1629.6
TOTAL CURRENT	120.4625	67.9

Table 16: Estimated power usage

This power estimate shows that if all of the systems consume their rated maximum power, the vehicle will draw approximately 2kW of power. When translated to batteries, a 24 volt battery capable of delivering 2kW of power for an hour would have a rating of over 80Amp-hours.

The vehicle's power system must also include a means of conditioning power so that, even with large current spikes from the motors, the voltage delivered to other systems remains constant to prevent logic errors. The 24V from the battery will be fed into an uninterruptable power source that so that the power to the computer will be constant and reliable. Once conditioned, the power must be delivered to each specific system while also protecting against short circuits. There will be a main power switch as well as an e-stop that cuts the power to the mechanical systems and sensors to prevent anyone from getting hurt or from the systems damaging itself or its surroundings.

6.1.2 Battery Options

The electric power demands of the system (current, voltage, capacity) are what determine the correct battery technology to use. The system has a few demands that dictate the correct battery technology. Because the vehicle is mobile, weight and subsequently the power density of each battery is important. However more important than a battery's weight and volume is its capacity. Having many systems with rather substantial power draws, the vehicle will need a large battery capable of a large burst of power for a short time.

Through the research several battery technologies were identified such as, Nickel Cadmium, Nickel Metal-Hydrde, Lithium-ion, Sealed Lead Acid and Lithium Polymer. Each technology has its

advantages and disadvantages. Each battery technology is discussed in detail in Appendix E: Battery Type Research.

6.1.3 Emergency Stops

According to the competition rules, the manual emergency stop system must kill all power to the vehicle, be hardware based, and be activated through the push of a physical button on the vehicle. The team looked at two possible solutions to for implementing the manual E-Stop. The first possible solution is using a high current emergency stop button. A unit like this would carry all of the electrical current traveling from the batteries into the vehicle. Pressing the E-Stop would cause a direct break in the power circuit and shut off power to the vehicle. The second manual system looked at was a relay based system. Using this method, a low current emergency stop button is used and is connected to relays on each of the power circuits. Pressing the button results in a signal being sent to all of the relays causing them to turn off and cut power to the system.

The wireless E-Stop must be held by a judge on the field and when activated must send a signal to the vehicle and cut power to the drive motors to bring the vehicle to a quick stop. The team looked at two possible implementations for a wireless E-Stop. The first system is using a small microcontroller and a Wireless internet chip. These units could be placed in a box with a button on the outside. When the button is pressed, the signal is received by a wireless internet router on the vehicle. Once the signal is received, the onboard computer can cut power to the drive motors on the vehicle. The second option looked at is using a remote unlocking system for a car. Using this system, pressing a button on a car unlocking device sends an RF signal to a receiver unit in the vehicle. When the receiver unit gets the signal, it sends a signal to unlock a car door. This signal can be connected the vehicle's onboard computer and can be interpreted as an E-Stop. Then the computer can cut power to the drive motors.

6.2 Power System Development

The battery technology that is implemented in the vehicle is Sealed Lead Acid (SLA). There are several deciding factors in this decision. The first factor is that the SLA technology provides very high currents (hundreds of amps) without the use of battery protection circuitry like in lithium ion and lithium polymer batteries need. Second is that the SLA batteries can be fully discharged without destroying or diminishing their capacity. Since lithium ion and lithium polymer batteries require circuit protection, they are not an economical choice for the vehicle. The type of Sealed Lead Acid battery we are using is Absorbed Glass Matt (AGM) formed in cylinders from Optima Batteries. These batteries can be compared to a typical car battery in size and weight. They have a capacity of 55AH and are Optima's

high performance batteries which will give approximately 40 min of run time. Two of the 12V batteries will be used in series to implement a 24V system. An image of one of the batteries is shown in Figure 80.



Figure 80: Optima Yellow Top Battery⁵⁵

For the power distribution system, the team chose to implement the majority of the power components on a single board that is located in the rear of the vehicle. The batteries are connected in series and the terminals are connected to the distribution board. The incoming power to the board goes through a 150 amp circuit breaker. The power goes to two fuse blocks that use auto-resettable fuses. Power then goes through relays if appropriate and to the vehicle subsystems. The electrical current in all of the circuits is monitored so that the power usage of the system can be known. A diagram of the power system can be seen Figure 81.

⁵⁵ http://www.fourleafbatteries.co.uk/images/YT_D34%20Right.jpg

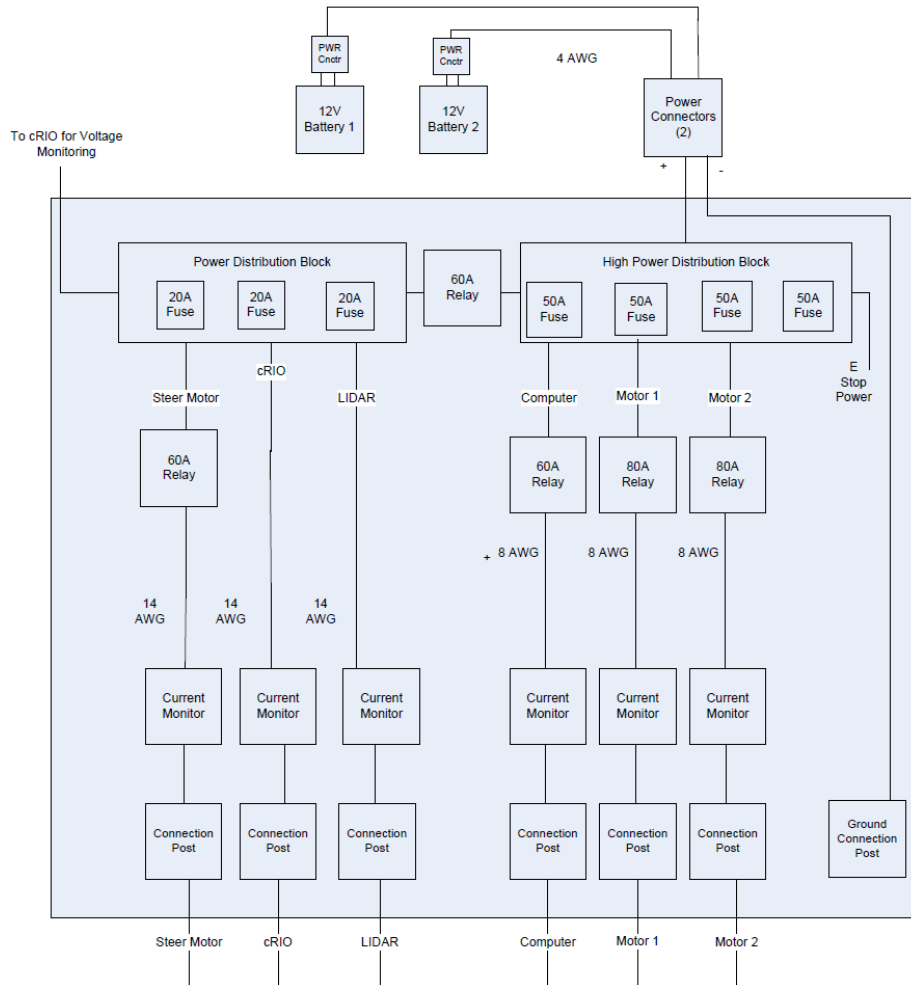


Figure 81: Power Distribution Diagram

The method chosen for the manual E-Stop is a relay system with a low current E-Stop button. The E-Stop button is connected to the ground line of the relays. When pressed, the connection to ground is cut and the relays release thus breaking the current traveling through the relays. The positive voltage to the relays is controlled by the cRIO control unit. This way, power to the drive motors can be removed when the cRIO receives the wireless E-Stop signal. The wireless E-Stop signal is provided by an RF car unlocking system. When the keychain button is pressed, the control unit on the vehicle detects the signal and sends an electrical pulse that is received by the cRIO.

As part of the team’s design strategy, a temporary mounting board was implemented before the final power board was built. This purpose of the mounting board was to get the vehicle driving as quickly as possible in order to identify potential system problems early in the building stage of the project. An image of the temporary mounting board can be seen in Figure 82.

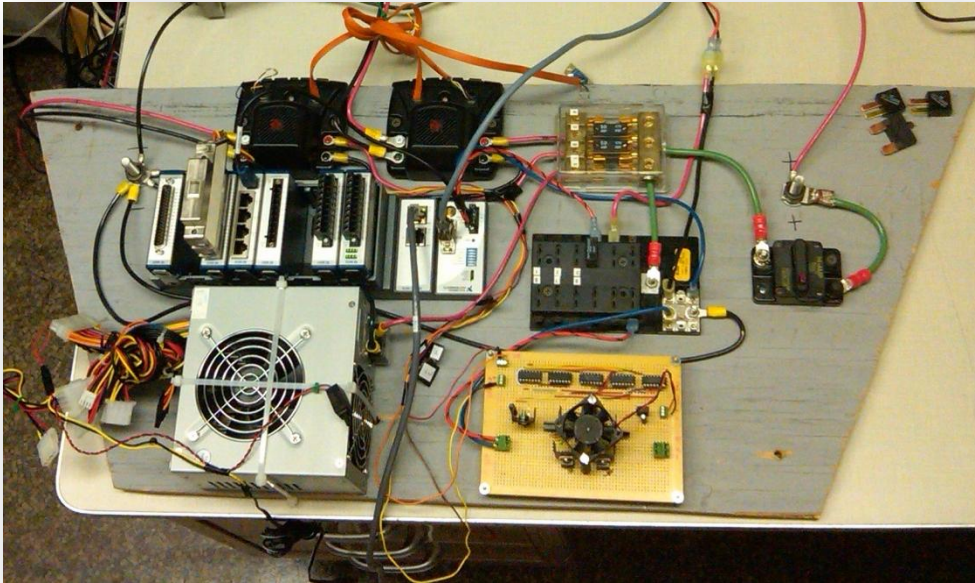


Figure 82: Temporary Mounting Board

This mounting board includes components from the power distribution board and components that will be mounted in the computer compartment. In this phase of the vehicle the emergency stops have not been implemented yet and the custom motor controller for the steered wheel was still in use.

6.3 Results

The final power distribution board was built on a sheet of ¼" polycarbonate. This board includes the master power breaker, three Jaguar motor controllers, two fuse blocks, and the emergency stop relays. A DB15 connector was added to the board so that the signals to the relays from the cRIO can be connected easily. The board is operating well with the three drive motors, cRIO, computer, router and LIDAR connected. This final power board design before it was finished being wired can be seen in Figure 83.



Figure 83: Final Power Board Design

6.4 Discussion

The team did not have time to implement electrical current monitors on all of the power circuits. This is something that is beneficial to the system because it can determine exactly what the vehicle's power usage distribution is and how the power usage varies in different operating conditions. However, it is not something that is necessary so it was dropped so that more time could be allocated to more important areas of the vehicle. The full vehicle with all electronics attached has not been completed yet so the actual runtime that can be achieved with one battery charge has not been determined yet.

7: Results

Results for the individual subsystems of the vehicle are given in their respective sections. Table 17 depicts the status of each of the project requirements that were set up initially. As of today the team has successfully met only six out of nine. However, although the project has come to an end the team will continue working on the Prometheus for another month until the time of the competition.

Requirement	Status
Length < 7 feet	Completed
Width < 5 feet	Completed
Height < 5 feet*	Completed
Mechanical E-stop	In progress
Wireless E-stop	In progress
Max Speed < 5mph	Completed
Lane Following	In progress
Obstacle Avoidance	Completed
Waypoint Navigation	Completed

Table 17: List of project requirements and their status

Besides the requirements above the team has rated the overall Prometheus' design in several categories which will be judged in the competition as well, show in Table 18. It can be seen that the vehicle is outstanding in all categories except for serviceability because to generate an efficient use of space while being rugged, safe and original some access to systems had to be sacrificed. One example of poor serviceability is that the power board is very difficult to get to in its current configuration, however in the coming few weeks its orientation will be reversed so that it will be easily accessible through a back frame panel.

Criteria	Good	Fair	Poor
Packaging neatness, efficient use of space	X		
Serviceability		X	
Ruggedness	X		
Safety	X		
Degree of original content in the vehicle	X		
Style	X		

Table 18: Vehicle design evaluation

In conclusion the team has built and realized an autonomous ground vehicle Prometheus featuring:

- Rear differential drive using two motors and a steered front wheel.
- Custom weatherproof aluminum chassis.
- Two lead acid batteries.
- National Instruments cRIO Controller.
- Main board computer equipped with NVIDIA's Tesla 1060c GPGPU for image processing.
- Path calculation using *Driving with Tentacles* approach.

The vehicle is capable of obstacle avoidance, lane following, waypoint navigation and meets the qualification requirements for the IGVC. Although the project has come to an end, the team will continue testing and improving the performance of the vehicle for another month.



Figure 84: An autonomous intelligent ground vehicle Prometheus

8: Budget Estimate

Part Name	Price	Cost/Donation
NI cRIO Controller	\$5,600	Donation
SICK LMS291-S05	\$6,000	Donation
NVIDIA Tesla C1060	\$1,500	Donation
Optima Yellow Top Batteries (4)	\$900	Donation
Sokkia Axis 3 DGPS Receiver	\$2,600	Donation
Point Grey FL2G-13S2M/C Cameras (2)	\$1,600	Cost
PENTAX 4.8mm f/1.8 C - 2/3"/REG (2)	\$275	Cost
US Digital Quad Encoders	\$175	Cost
PNI V2xe 2-axis Compass	\$250	Cost
Main Computer System	\$1,400	Cost
Motors (2 NPC-T64, 1, ML42-24)	\$700	Cost
Motor Controllers (3)	\$300	Cost
Chassis Components (Aluminum Tubing, Plastic, Metal Plates, nuts and bolts, etc.)	\$1,750	Cost
Power Electronics (Circuit Breakers, Relays, Cables, Battery Chargers)	\$1,250	Cost
Total Donations	\$ 16,600	
Total Cost	\$7,700	
Total	\$ 24,300	

9: Sponsors

This requires a large budget for all the equipment like sensors, control systems, drive trains and materials. Unfortunately, the school is only able to contribute a \$1000 for each department involved in the project. Therefore the contribution from the sponsors is essential in order for this project to be completed. We are grateful to everyone who is contributing to this project in any way. Below you will find information about each of the sponsors and their contributions.

6.1 WPI

We would like to thank WPI and the following departments for accommodating the team with lab space and computer hardware needed for this project as well as helping with project funds.

- Robotics Engineering Program
- CS Department
- ECE Department
- ME Department



6.2 The NVIDIA Corporation

NVIDIA is a multinational corporation which specializes in the development of graphics processing units and chipset technologies for workstations, personal computers, and mobile devices. Based in Santa Clara, California, the company has become a major supplier of integrated circuits (ICs) such as graphics processing units (GPUs) and chipsets used in graphics cards, and of video-game consoles and personal-computer motherboards.



Figure 85: Tesla C1060 GPGPU

Donation: Tesla C1060, see Figure 85.

6.3 The National Instruments

National Instruments specializes in designing hardware for embedded systems and test equipment. They also utilize custom software to speed up the hardware programming process.



Donations:

- Compact RIO embedded control system
- 4 LabView Student Edition Licenses
- Professional Support

10: Bibliography

- Akroush, Allen, et al. "university of Michigan-Dearborn - Wolf." *Intelligent Ground Vehicle Competition*. 2008. <http://www.igvc.org/design/reports/dr233.pdf> (accessed 10 12, 2009).
- ASD. *Advanced Safety Devices*. 2010. http://www.safety-devices.com/how_compass_works.htm (accessed 2010).
- ASIMO. *ASIMO*. 2010. <http://asimo.honda.com/> (accessed 2010).
- Barry, Kevin, et al. "μCeratops Design Report IGVC 2008." *Intelligent Ground Vehicle Competition*. 2008. <http://www.igvc.org/design/reports/dr229.pdf> (accessed 10 12, 2009).
- Cardwell, James, et al. "Bluefield State College - Anassa IV." *Intelligent Ground Vehicle Competition*. May 2009. <http://www.igvc.org/design/reports/dr197.pdf> (accessed April 10, 2010).
- Cenek, Peter D., Neil J. Jamieson, and Maurice W. McLarin. "Frictional Characteristics of Roadside Grass Types." *Thames Valley Safer Roads Partnership*. http://www.saferroads.org.uk/Papers_PDFs/Peter%20Cenek%20-%20Frictional-Characteristics-Roadside-Grass-Types.pdf (accessed April 6, 2010).
- Corporation, PNI. "PNI V2Xe 2-Axis Compass Module Datasheet." October 2005. [cplusplus.com. *The C++ Resource Network*](http://www.cplusplus.com/). <http://www.cplusplus.com/> (accessed 2010).
- Digital, US. "US Digital E8P OEM Optical Kit Encoder Datasheet."
- Divya, K.C., and Jacob Ostergaard. "Battery energy storage technology for power systems—An overview." *Electrical Power Systems Research*, 2009: 511.
- Dong, Guo, and Ming Xie. "Color Clustering and Learning for Image Segmentation Based on Neural Networks." *IEEE Explore*, 2005: 928.
- Dutch, Steven. *Converting UTM to Latitude and Longitude (Or Vice Versa)*. 12 14, 2009. <http://www.uwgb.edu/dutchs/usefuldata/UTMFormulas.HTM>.
- Eclipse. *About the Eclipse Foundation*. 2010. <http://www.eclipse.org/org/> (accessed 2010).
- FARS. *Fatal Analysis Reporting System Encyclopedia*. 2008. <http://www-fars.nhtsa.dot.gov/Main/index.aspx> (accessed 2010).
- Felix von Hundelshausen, Michael Himmelsbach, Falk Hecker, Andre Mueller, and Hans-Joachim Wuensche. "Driving with Tentacles: Integral Structures for Sensing and Motion." *Journal of Field Robotics*, 2008.
- Forsyth, David A, and Jean Ponce. "Computer Vision A modern Approach." 29. Pearson Education, 2003.
- Garcia, Auelio, and Francisco Zabala. "IONO - Cal State University, Fullerton." *Intelligent Ground Vehicle Design*. 2006. <http://www.igvc.org/design/reports/dr159.pdf> (accessed April 29, 2010).

Gavrila, Dr. Dariu M. *Dr. Dariu Gavrilă Computer Vision Image Processing Technology Site*. 2010. http://www.gavrila.net/Research/Smart_Vehicles/smart_vehicles.html.

Gerrits, Mark, and Philippe Bekaert. "Local Stereo Matching with Segmentation-based Outlier Rejection." *Hasselt University*, 2006.

Givental, Gary, et al. "Lawrence Technological University - Viper." *Intelligent Ground Vehicle Competition*. May 2008. <http://www.igvc.org/design/reports/dr211.pdf> (accessed April 10, 2010).

Hokuyo Automatic Co., LTD. "Hokuyo URG-04LX Specifications Document." 07 22, 2005.

IGVC. *IGVC*. 2010. <http://www.igvc.org> (accessed 2010).

Inc., DEEM Controls. *How do Absolute Encoders work?* 02 2009. <http://www.deemencoders.com/Howitworks.html#2> (accessed 2010).

Keith, Sevcik. *Interfacing with the Sick LMS-200*. <http://www.pages.drexel.edu/~kws23/tutorials/sick/sick.html> (accessed 2010).

Keller, John. *Unmanned vehicle spending in the 2010 DOD budget to reach \$5.4 billion*. 05 29, 2009. <http://www.militaryaerospace.com/index/display/article-display/363553/articles/military-aerospace-electronics/executive-watch/unmanned-vehicle-spending-in-the-2010-dod-budget-to-reach-54-billion.html> (accessed 2010).

kent, Danny. *OpenJaus - JAUS Robotics Software Development Kit (SDK)*. 2010. <http://openjaus.com/> (accessed 2010).

Ltd, Sokkia Co. "Sokkia Axis Operations Manual." 1999.

Maxim Integrated Products. *Glossary*. <http://www.maxim-ic.com/glossary/> (accessed April 29, 2010).

Mehaffey, Joseph. *Error Measures - what do they mean?* <http://gpsinformation.net/main/errors.htm> (accessed 2010).

Murray, Don, and James J. Little. "Using Real-Time Stereo Vision for Mobile Robot Navigation." *Autonomous Robots*, 2000.

NetBeans. *NetBeans IDE 6.8 Release Information*. 2010. <http://netbeans.org/community/releases/68/> (accessed 2010).

NPC Robotics. "NPC-T64." *NPC Robotics*. 2009. <http://www.npcrobotics.com/products/viewprod.asp?prod=42&cat=20&mode=gfx> (accessed 10 13, 2009).

Oracle. *What is Java and why do I need it?* 2010. http://www.java.com/en/download/faq/whatis_java.xml (accessed 2010).

Roboteq INC. *Roboteq Motor Controllers Technology*. 2009. <http://www.roboteq.com/technology/roboteq-ax-family-motor-control-technology.html> (accessed October 12, 2009).

SICK, Inc. "SICK LMS200/291 Technical Information Document."

Stam, Joe. *Stereo Imaging with CUDA*. January 2008.

The C++ Resource Network. <http://www.cplusplus.com> (accessed 03 01, 2010).

Travis, Detle. *Hokuyo URG-04LX Laser RangeFinder (LIDAR)*. 02 23, 2009. <http://www.hizook.com/projects/hokuyo-urg-04lx-laser-rangefinder-lidar>.

Trimble GPS. *Trimble GPS Tutorial*. <http://www.trimble.com/gps/index.shtml> (accessed October 3, 2009).

USARIC. *Unmanned Systems & Robotics Interoperability Center*. 2010. <http://www.usaric.org/> (accessed 2010).

Wikipedia. *Image Rectification*. February 6, 2010. http://en.wikipedia.org/wiki/Image_rectification (accessed April 26, 2010).

"Y-clops Robot." *Y-clops*. 2005. <http://yclops.ece.byu.edu/yclops.php> (accessed April 29, 2010).

Appendix A: On-board computer specifications

Video	http://www.newegg.com/Product/Product.aspx?Item=N82E16814150445	XFX GM210XYNF2 GeForce 210 512MB 64-bit DDR2 PCI Express 2.0 x16 HDCP Ready Video Card - Retail	49.99
Motherboard	http://www.newegg.com/Product/Product.aspx?Item=N82E16813188046	EVGA E758-TR 3-Way SLI (x16/x16/x8) LGA 1366 Intel X58 ATX Intel Motherboard - Retail	\$269.99
CPU	http://www.newegg.com/Product/Product.aspx?Item=N82E16819115202	Intel Core i7-920 Bloomfield 2.66GHz LGA 1366 130W Quad-Core Processor Model BX80601920 - Retail	\$288.99
RAM	http://www.newegg.com/Product/Product.aspx?Item=N82E16820231225	G.SKILL 6GB (3 x 2GB) 240-Pin DDR3 SDRAM DDR3 1600 (PC3 12800) Triple Channel Kit Desktop Memory Model F3-12800CL9T-6GBNQ - Retail	\$149.99
PSW	http://www.newegg.com/Product/Product.aspx?Item=N82E16817121045	KINGWIN ABT-730MM 730W ATX 12V Ver.2.91 SLI Ready CrossFire Ready Active PFC Power Supply - Retail	\$76.99

HDD	http://www.newegg.com /Product/Product.aspx?Item=N82E16822136074	Western Digital Caviar Blue WD3200AAKS 320GB 7200 RPM SATA 3.0Gb/s 3.5" Internal Hard Drive -Bare Drive	49.99
	http://www.newegg.com /Product/Product.aspx?Item=N82E16820220455	Patriot PS-100 PS32GS25SSDR 2.5" 32GB SATA I/II MLC Internal Solid State Drive (SSD)	99.99

Appendix B: NI cRIO Specifications

Component	Model	Description
Main cRIO Unit	NI-9074	Rugged, embedded control and monitoring system, 400 MHz real-time processor, data logging, 2M gate 8-slot FPGA chassis for custom I/O timing, control, and processing Two 10/100BASE-T Ethernet ports; RS232 serial port for connection to peripherals
24V Digital Output Module	NI-9472	8-Channel 24 V Logic, 100 μ s, Sourcing Digital Output Module
Analog Input Module (x2)	NI-9201	8-Ch, \pm 10 V, 500 kS/s, 12-Bit Analog Input Module, C Series
RS-232 Serial Interface Module	NI-9870	4 RS232 (TIA/EIA-232) serial ports, Baud rates from 14 b/s to 921.6 kb/s
Digital Input/Output Module (x2)	NI-9403	C Series 32-Ch, 5 V/TTL Bidirectional Digital I/O Module

Appendix C: Hardware Control System Research

Mini ITX

Mini ITX motherboards are low power, small form factor, boards. Of the motherboards described here, this board is the smallest with dimensions of 6.7 x 6.7 inches. These boards come with the processor as part of the board and usually do not include a fan since the processor power consumption is very low. These boards do include PCI expansion slots and connectors on the back to connect to peripheral devices. The assortment of connectors usually includes USB, Ethernet, audio, serial, PS/2 and VGA. Mini ITX boards alone would not be good at image processing but there is potential to use a graphics card in conjunction with the board. Mini ITX boards also do not include as many possibilities for interfacing with peripherals as some embedded systems described later in this section do, but generally they do have more than a laptop. An image of a general Mini ITX board can be seen below.



Figure 86: Mini ITX Motherboard⁵⁶

Mini ITX System

Mini ITX systems are pre-built computers with Mini ITX motherboards. These have similar specifications to a Mini ITX motherboard. The advantages of using a Mini ITX system are its low cost, low power consumption, and small form factor. Buying a full system with all parts included is less expensive than buying all of the individual parts. The disadvantage to this method is customizability. When buying a pre-built system, there will be fewer options than when building a custom system. An image of a Mini ITX system can be seen in the figure below.

⁵⁶ http://www.wired.com/images_blogs/gadgetlab/images/2007/09/17/picture_1.jpg



Figure 87: Mini ITX System⁵⁷

Micro ATX

Micro ATX motherboards are slightly larger than Mini ITX boards, they measure 9.6 x 9.6 inches. These boards have almost all of the capabilities of a full size ATX board. Usually Micro ATX boards have less PCI slots for peripheral connections. Most of them have USB, audio, Ethernet, PS/2, and video connections. The advantage to these boards is smaller size. Using this board would require buying a separate processor. An advantage to using a micro ATX board is that it has the ability to interface with a graphics card for increased image processing power. Some disadvantages are that micro ATX boards are not as power efficient as mini ITX boards and they do not provide as much interfacing capability as some of the embedded systems discussed later in this section, and when choosing a board there is less selection than an ATX board since ATX is more popular. An image of a Micro ATX board can be seen below.

⁵⁷ <http://www.ewayco.com/22-low-cost-embedded-systems-epia-mini-itx-pc/GX-low-cost-embedded-systems-VIA-EPIA-Mini-ITX-large.jpg>



Figure 88: Micro ATX Motherboard⁵⁸

ATX

ATX boards are full size desktop computer motherboards. At 12 x 9.6 inches, the ATX board is the largest of the motherboards explored for the project. This board has similar interfacing connections as a micro ATX board. It does however include more expansion slots which could be used to increase interfacing capabilities through PCI cards. It can also be used with a graphics card to increase image processing capabilities. Using either the micro ATX or ATX boards also requires purchasing a CPU, heat-sink, RAM, and a storage drive. ATX motherboards also use more power than typical embedded systems. An image of a typical ATX motherboard can be seen below.

⁵⁸ <http://wafflesandpirates.files.wordpress.com/2007/05/micro-atx-overhead.jpg>



Figure 89: ATX Motherboard⁵⁹

Laptop

Using a laptop as the primary control system for the vehicle is also an option. Laptops have the benefits of being small, low powered, including their own battery power source, including a screen, and being relatively inexpensive. Laptops are also a modular unit that can easily be removed from the vehicle which allows for programming development without requiring the vehicle to be present. For the JAUS⁶⁰ competition, wireless communication is required. If a laptop is used as part of the control system, the built-in laptop wireless card can be used for JAUS communication. In the area of processing power, a laptop can provide more processing capabilities than some embedded systems but not as much as a desktop computer system. The main disadvantage of using a laptop for the control system is that it lacks peripheral connections and expandability. Most laptops are limited to a few USB ports and one PCMCIA slot. This would prevent us from using video cards and we would most likely require additional control hardware in order to interface with all of the sensors and motor controllers. Another downside to using a laptop is that they are not designed for embedded applications and as a result they are not as rugged as embedded systems. An example of the type of laptop we were considering for the vehicle can be seen below.

⁵⁹http://img.alibaba.com/photo/106025539/Asus_P6TSE_Core_i7_Intel_X58_DDR3_CrossFireX_A_GbE_ATX_Motherboard.jpg

⁶⁰ Joint Architecture for Unmanned Systems



Figure 90: Laptop⁶¹

PC104 Single Board Computer

The PC104 is a certain form factor of Single Board Computers (SBC's). The term PC104 refers to the unique standard protocol used for the computer bus. This bus allows for easy expandability by allowing multiple boards to be stacked onto the main board and connected through the computer bus. An example of a PC104 system can be seen in the figure below. Every PC104 system must include a CPU board, often referred to as the SBC, and can include any number of expansion modules. By stacking, custom configurations can be easily made to meet the specific needs of an embedded system. Possible expansion modules include USB, RS-232, Ethernet, GPS, Analog I/O, Digital I/O, wireless modems, and FPGA boards. There is no specified limit on how many expansion modules may be used but after a certain number, signal strength and processing power become limitations on the system.

⁶¹ <http://www.geeksquad.com/intelligence/wp-content/uploads/2009/12/laptop.jpg>



Figure 91: PC104 Single Board Computer⁶²

PC104 systems are commonly used for embedded applications because they are reliable for data acquisition, rugged, relatively inexpensive, and have low power consumption. The expansion modules include mounting holes and stand-offs to firmly secure them together making the system more rugged. For our application, the downside to this system is that it may not be able to provide the level of processing power we need for image processing. Generally, PC104 CPU boards range from 200MHZ to 1 GHZ.

National Instruments Compact RIO

The Compact RIO (cRIO) is a real-time embedded controller developed by National Instruments. This system includes a processor and an FPGA backplane that integrates with reconfigurable modules that can be added to the base system. The cRIO can be seen in the figure below. The main advantage of the Compact RIO is its re-configurability. It is also designed to be an embedded control module and as such is very rugged, a feature desirable for an unmanned ground vehicle.



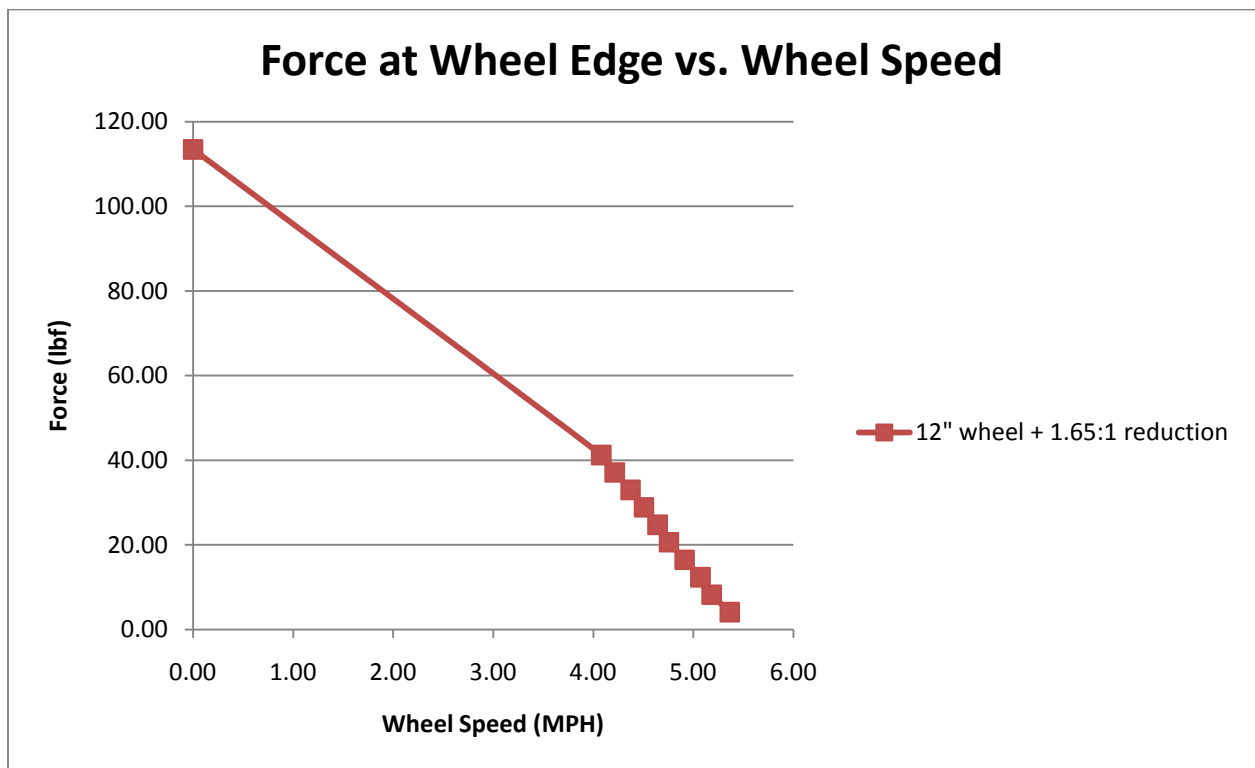
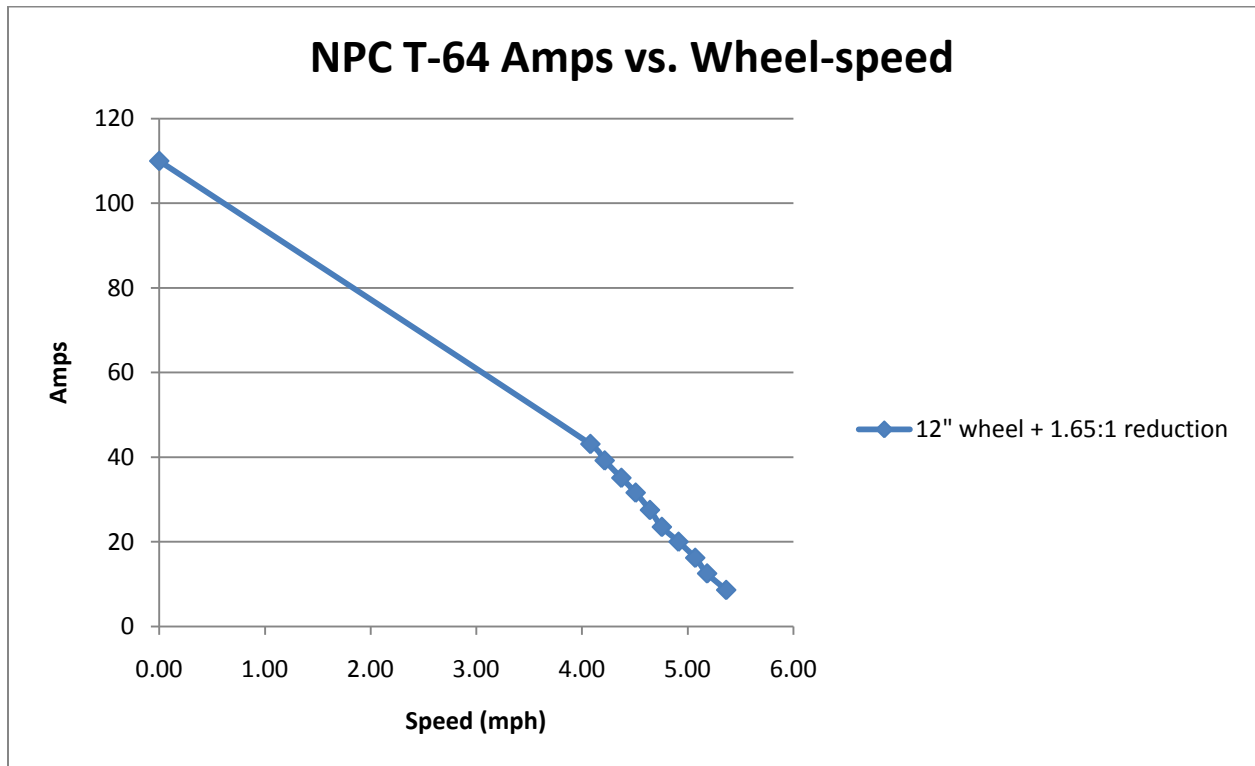
⁶² <http://www.signal11.us/io.html>

Figure 92: cRIO System⁶³

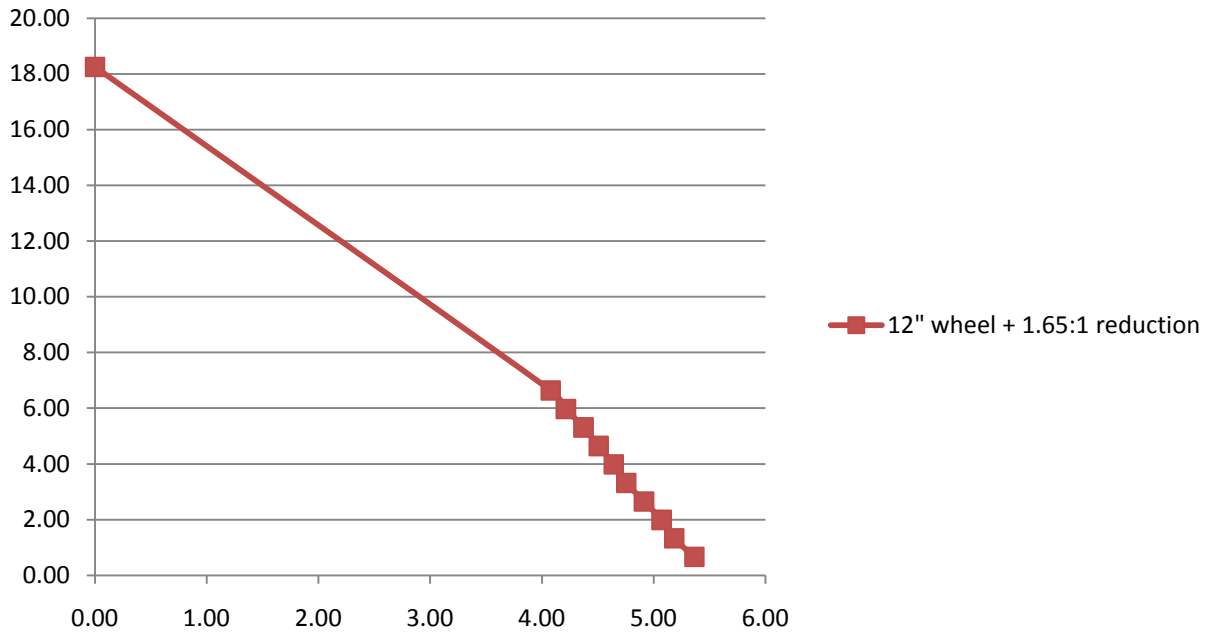
The cRIO is composed of three main parts, controller, backplane, and modules. The controller houses a processor for executing code in real-time and communicating with the backplane and devices to which it reports data. The backplane consists of an FPGA that can be configured by the user. The backplane provides all of the necessary connections for the expansion modules and it communicates with the controller processor. Integrated controller and chassis units can have up to 2 million gate FPGAs and 400 MHz processors. Expansion modules can be manufactured by National Instruments or by third party vendors. These modules include Analog I/O, Digital I/O, GPS, CAN Controllers, RS-232, motion controllers, and others. Using these modules, a system can be constructed to fit the specific requirements of an embedded system. The downside to the cRIO is that it is not designed for image processing. Prometheus needs to be able to process streaming images from the stereo vision system in real time. It also needs to correlate that information with the range-finding data in real time. The cRIO would not be capable of accomplishing this.

⁶³ <http://www.ni.com/>

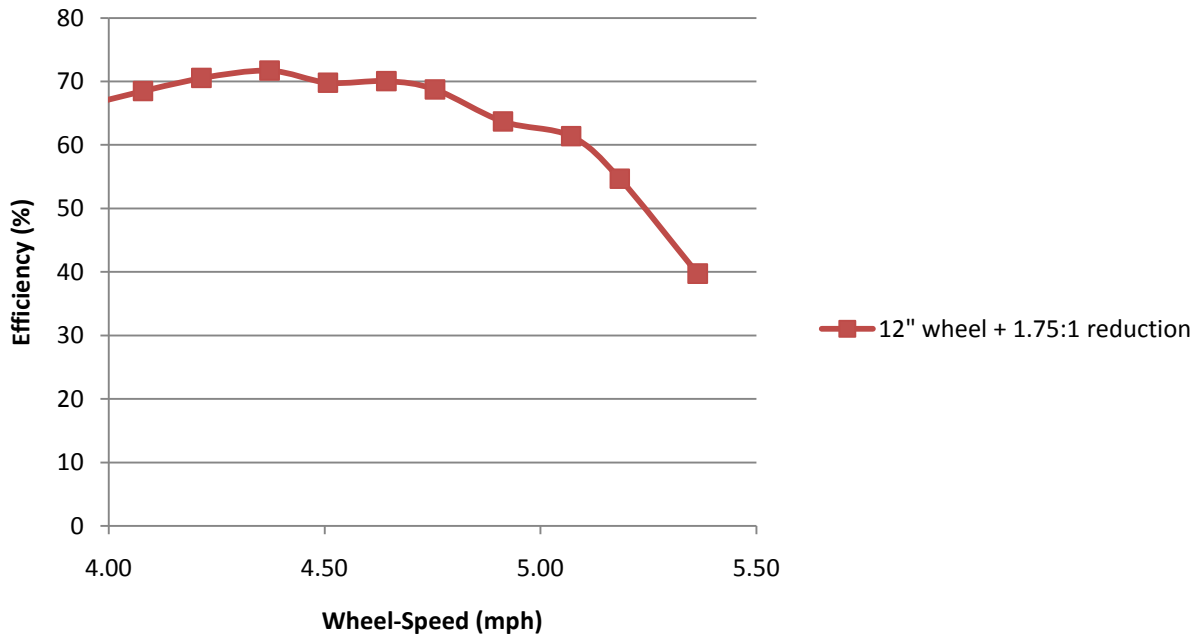
Appendix D: Motor Performance Data



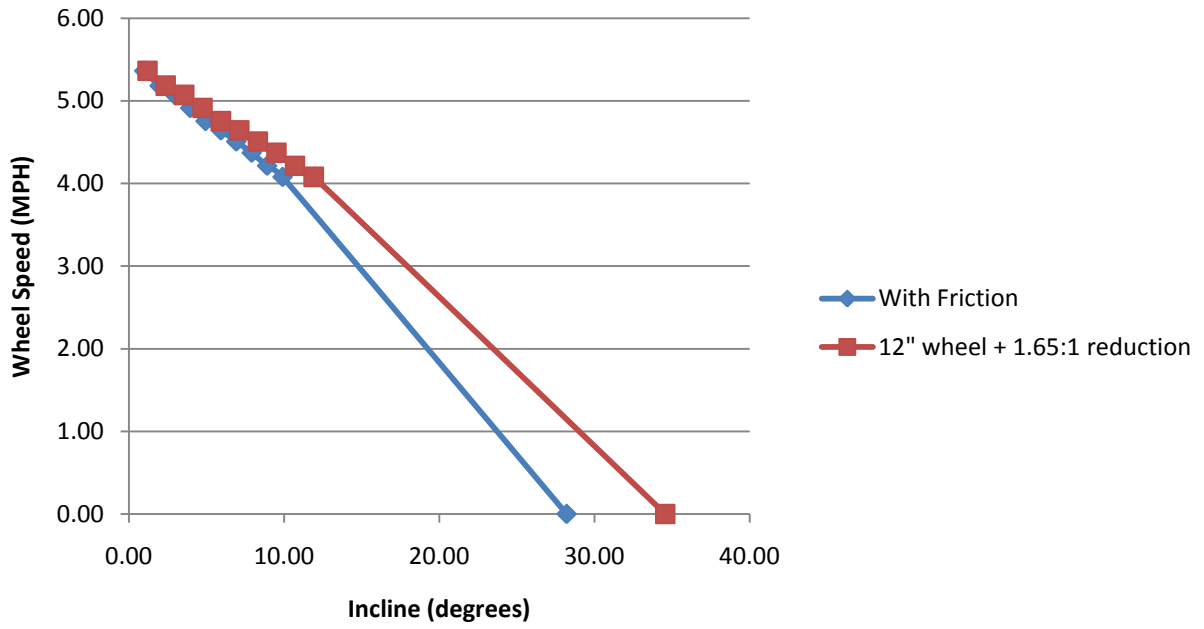
Acceleration vs Wheel Speed



NPC T-64 Efficiency vs. Wheel-Speed



Wheel Speed vs. Incline



Appendix E: Battery Type Research

In a Nickel Cadmium (or NiCd) battery, the electrodes of the battery use Nickel Oxide Hydroxide and metallic Cadmium as the two electrodes. There are two types of batteries sealed and vented, however we would need a sealed battery because the orientation of the battery cannot be guaranteed the correct orientation for operation. Though this battery technology is one of the oldest it is still used today in many applications because it has several factors that other technologies can't replace yet. The NiCd battery has a very low equivalent series resistance meaning that the battery is capable of very high surge currents making it a good choice for any mobile application that need high amounts of current for short periods of time. These batteries can also tolerate almost complete discharge without any adverse side effects. One of the major drawbacks of the battery is that the battery has a lower volumetric energy capacity than some newer battery technologies. Another problem with the NiCd battery is that the battery is hazardous so when the battery is no longer needed, it is hazardous waste.

The Nickel Metal Hydride or NiMH battery technology has begun to replace the NiCd battery because it solves a few of the problems with the NiCd battery. With two to three times the volumetric energy storage capacity and different materials, the NiMH battery not only better the energy density of the NiCd battery, but also provides a rechargeable energy source that is non-toxic. One drawback of the NiMH technology is that the self discharge of the battery is slightly more than the NiCd. However the energy source we are looking to use will be charged and discharged very frequently and thus the energy losses that would, over long periods of time, drain the battery, will not affect the design.

Lithium ion batteries are common rechargeable batteries used for consumer electronics. This is because they have a very high energy to size ratio. Care needs to be taken when handling lithium ion batteries since they can explode if treated improperly due to a highly flammable electrolyte used between the cathode and the anode. Lithium ion batteries are more expensive than other forms of rechargeable batteries but their high energy density, lack of memory effect from discharging and slow discharge during non-use have made them very desirable. A new version of lithium ion batteries is lithium polymer batteries. These batteries use the same flammable electrolyte but it is contained within a polymer composite. This makes the batteries easier to manufacture thus making them less expensive. The fact that the flammable electrolyte is contained in a polymer composite also makes them more robust to physical damage.

As the last alternative for the on board power source, the Lead Acid battery has been considered. They have been around for a very long time; and it is still used in hundreds of applications from standard car, boat and motorcycles to computer server power backup systems. The Lead Acid

battery that generated electricity when aqueous Lead Dioxide transforms to Lead Oxide and then reacts with Sulfuric acid to transform to Lead sulfate which generates Lead ions thus producing electricity. During the charging process the opposite process happens (Divya and Ostergaard 2009). The Lead Acid battery also has a low equivalent series which makes this battery a good choice for applications requiring short but high current bursts. Also because these batteries have been around for a long time they are produced relatively inexpensively when compared with the others.

Appendix F: The Software Architecture

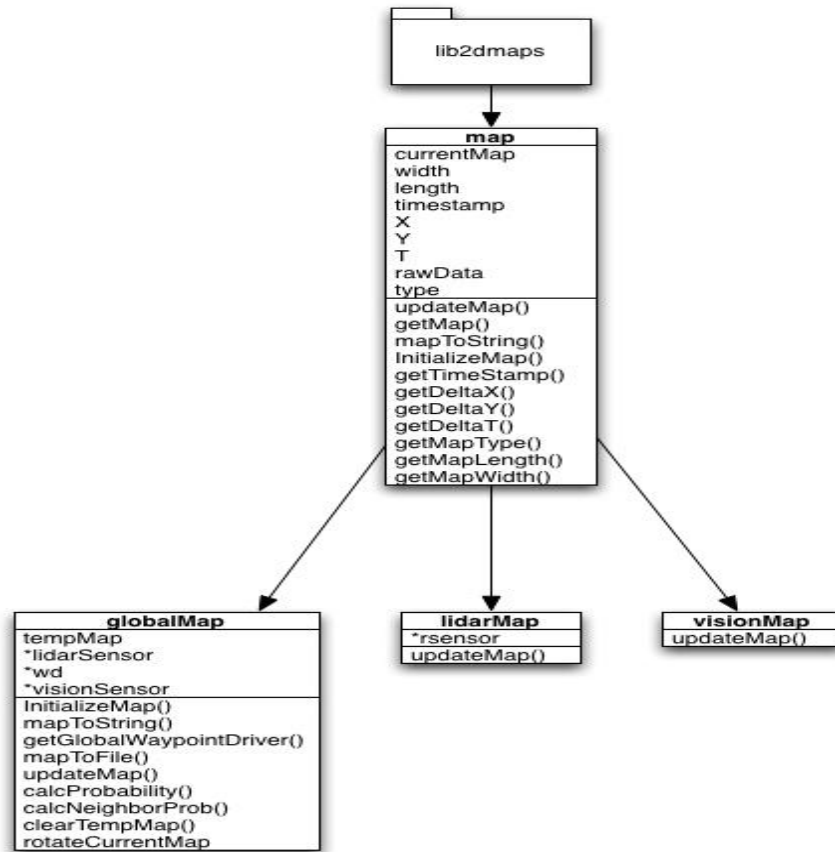


Figure 93: Lib2dmaps diagram

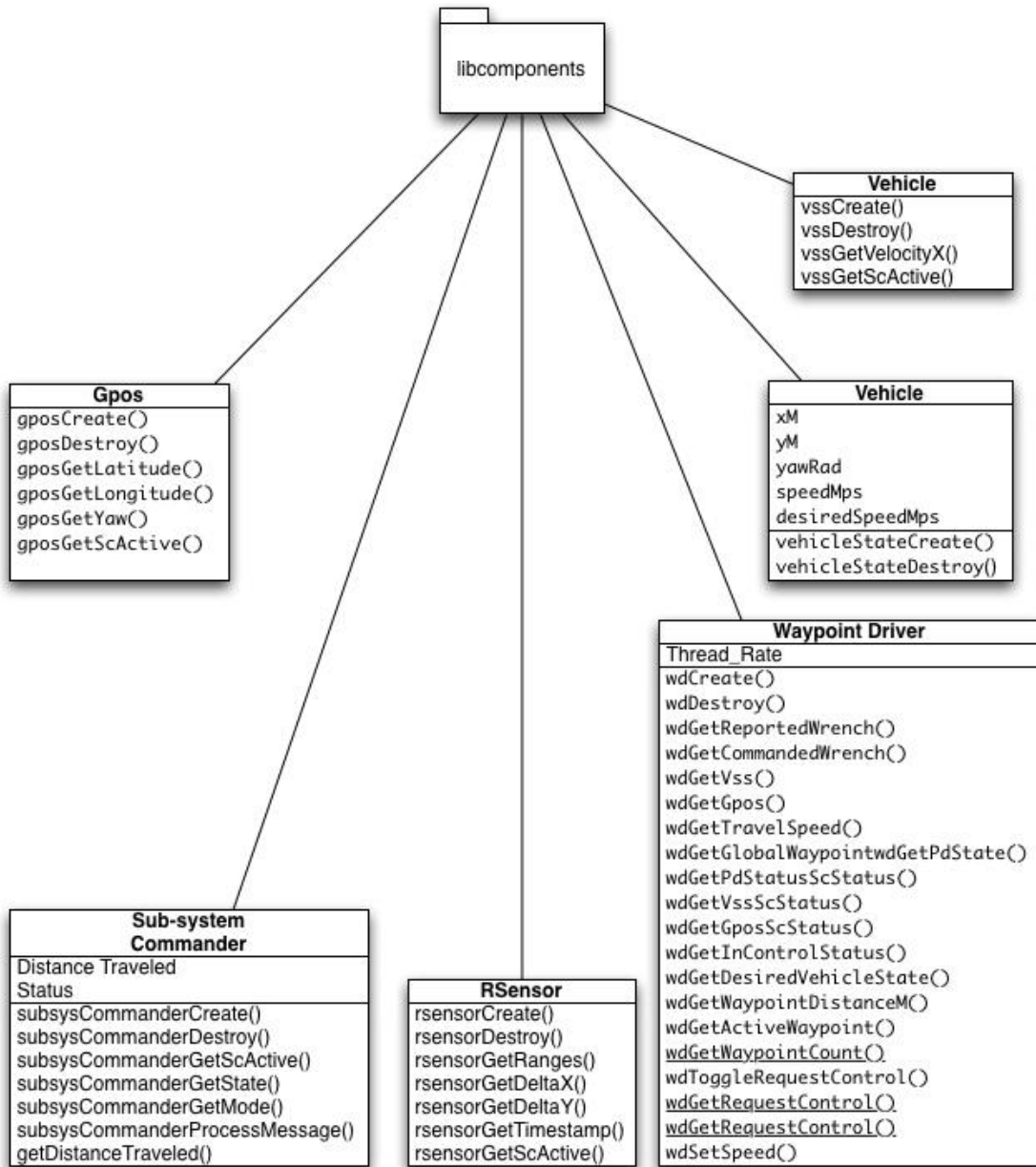


Figure 94: libcomponents diagram

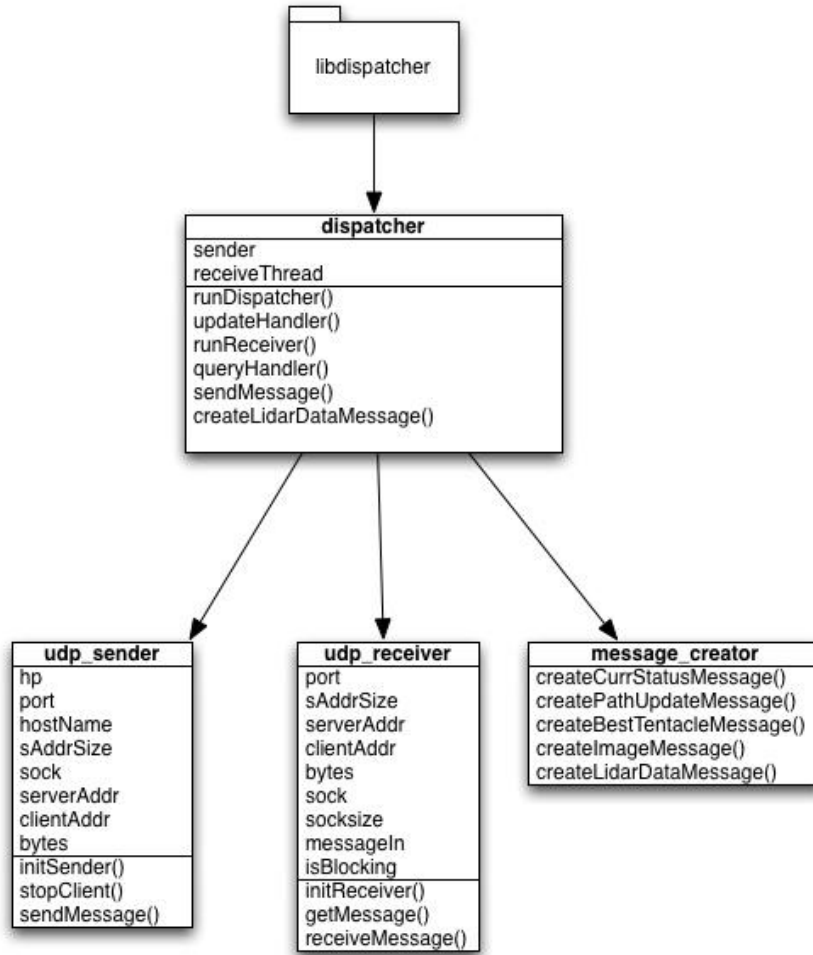


Figure 95: libdispatcher diagram

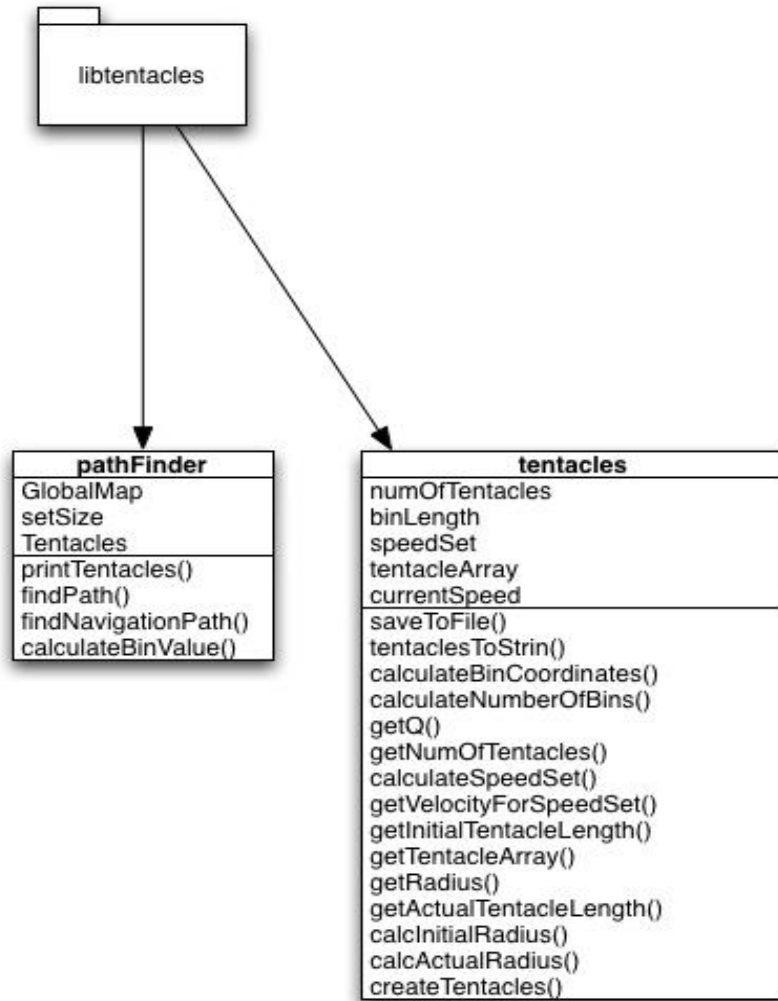


Figure 96: libtentacles diagram

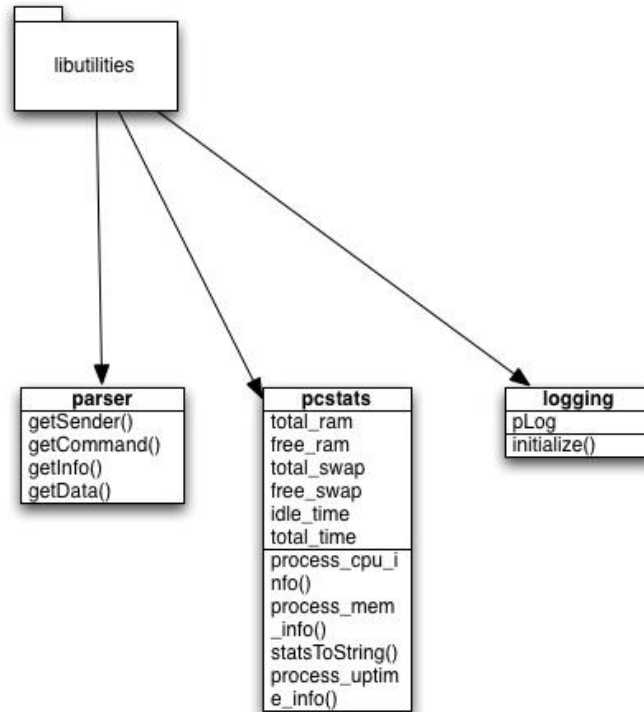


Figure 97: libutilities diagram

Appendix G: Competition Rules

I. COMPETITION INFORMATION

I.1 TEAM ENTRIES

Teams may be comprised of undergraduate and graduate students, and must be supervised by at least one faculty advisor. Interdisciplinary teams are encouraged (EE, ME, CS, etc.). Students must staff each team. Only the student component of each team will be eligible for the awards. Faculty supervisor will certify that all team members are bonafide students on application form and will also provide contact information (telephone number and e-mail address) for himself and the student team leader on the form. Business/Non-Engineering students are encouraged to join teams to promote marketing, sponsorships, and other program management functions. For a student to be eligible to compete as a team member, they are required to have attended at least one semester of school as a registered student between June 2009 and June 2010.

Team sponsors are encouraged. Sponsors' participation will be limited to hardware donation and/or funding support. Sponsors logos may be placed on the vehicle and may be displayed inside of the team maintenance area. Teams should encourage sponsor attendance at the IGVC.

Schools are encouraged to have more than one entry; but are limited to a maximum of three per school, and each vehicle must have a separate team of students and a distinct design report. Each entry must be based on a different chassis and software and must be documented by a separate application form and design report, submitted in accordance with all deadlines. All entries must have a team name and each application form must be **TYPED** and accompanied with a \$250.00 non-refundable registration fee made payable to *Oakland University*. Intention to compete must be received no later than **February 28, 2010**, by mailing your application form to:

Gerald C. Lane
C/O Dr. Ka C. Cheok
102G SEB
SECS-ESE Dept.
Oakland University
Rochester, MI 48309-4478

If you have any questions, please contact Bernard Theisen by telephone at (586) 574-8750, fax: (586) 574-8684 or e-mail: bernard.theisen@us.army.mil.

I.2 VEHICLE CONFIGURATION

The competition is designed for a small semi-rugged outdoor vehicle. Vehicle chassis can be fabricated from scratch or commercially bought. Entries must conform to the following specifications:

- **Design:** Must be a ground vehicle (propelled by direct mechanical contact to the ground such as wheels, tracks, pods, etc or hovercraft).
- **Length:** Minimum length three feet, maximum length seven feet.
- **Width:** Minimum width two feet, maximum width five feet.
- **Height:** Not to exceed 6 feet (excluding emergency stop antenna).
- **Propulsion:** Vehicle power must be generated onboard. Fuel storage or running of internal combustion engines and fuel cells are not permitted in the team maintenance area (tent/building).
- **Speed:** For safety, a maximum vehicle speed of five miles per hour (5 mph) will be enforced. All vehicles must be hardware governed not to exceed this maximum speed. No changes to maximum speed control hardware are allowed after the vehicle passes Qualification.
- **Mechanical E-stop location:** The E-stop button must be a push to stop, red in color and a minimum of one inch in diameter. It must be easy to identify and activate safely, even if the vehicle is moving. It must be located in the center rear of vehicle at least two feet from ground, not to exceed four feet above ground. Vehicle E-stops must be hardware based and not controlled through software. Activating the E-Stop must bring the vehicle to a quick and complete stop.
- **Wireless E-Stop:** The wireless E-Stop must be effective for a minimum of 50 feet. Vehicle E-stops must be hardware based and not controlled through software. Activating the E-Stop must bring the vehicle to a quick and complete stop. During the competition performance events (Autonomous Challenge and Navigation Challenge) the wireless E-stop will be held by the Judges.
- **Payload:** Each vehicle will be required to carry a 20-pound payload. The shape and size is approximately that of an 18" x 8" x 8" cinder block. Refer to section I.3 Payload.
- **A priori Data:** The intent is to compete without a priori or memorized data. Course position data should not be mapped/stored. This is difficult to enforce, each team is expected to comply with the intent. Both the Autonomous Challenge and Navigation Challenge courses will be changed after each heat and between runs to negate any memorization or course familiarization techniques.

I.3 PAYLOAD

The payload must be securely mounted on the vehicle. If the payload falls off the vehicle during a run, the run will be terminated. The payload specifications are as follows: 18 inches long, 8 inches wide, 8 inches high and a weight of 20 pounds.

I.4 QUALIFICATION

All vehicles must pass Qualification to receive standard award money in the Design Competition and compete in the performance events (Autonomous Challenge and Navigation Challenge). To complete Qualification the vehicle must pass/perform the following eight criteria.

- **Length:** The vehicle will be measured to ensure that it is over the minimum of three feet long and under the maximum of seven feet long.
- **Width:** The vehicle will be measured to ensure that it is over the minimum of two feet wide and under the maximum of five feet wide.

- **Height:** The vehicle will be measured to ensure that it does not to exceed six feet high; this excludes emergency stop antennas.
- **Mechanical E-stop:** The mechanical E-stop will be checked for location to ensure it is located on the center rear of vehicle a minimum of two feet high and a maximum of four feet high and for functionality.
- **Wireless E-Stop:** The wireless E-Stop will be checked to ensure that it is effective for a minimum of 50 feet. During the performance events the wireless E-stop will be held by the Judges.
- **Max Speed:** The vehicle will have to drive at full speed over a prescribed distance where its speed will be determined. The vehicle must not exceed the maximum speed of five miles per hour. No change to maximum speed control hardware is allowed after qualification. If the vehicle completes a performance event at a speed faster then the one it passed Qualification at, that run will not be counted.
- **Lane Following:** The vehicle must demonstrate that it can detect and follow lanes.
- **Obstacle Avoidance:** The vehicle must demonstrate that it can detect and avoid obstacles.
- **Waypoint Navigation:** Vehicle must prove it can find a path to a single 2 meter navigation waypoint.

During the Qualification the vehicle must be put in autonomous mode to verify the mechanical and wireless E-stops and to verify lane following and obstacle avoidance. The vehicle software can be reconfigured for waypoint navigation qualification. For the max speed run the vehicle may be in autonomous mode or joystick/remote controlled. Judges will not qualify vehicles that fail to meet these requirements. Teams may fine tune their vehicles and resubmit for Qualification. There is no penalty for not qualifying the first time. Vehicles that are judged to be unsafe will not be allowed to compete. In the event of any conflict, the judges' decision will be final.

I.5 INDEMNIFICATION AND INSURANCE

Teams will be required to sign an application form prior to **February 28, 2010**. Along with the application form, there will be a Waivers of Claims that will need to be signed by each individual who will be participating at the competition.

Additionally, the Team's sponsoring institution will also be required to supply AUVSI with a Certificate of Insurance at the time the Application Form is submitted. The certificate is to show commercial general liability coverage in an amount not less than \$1 million.

NOTE: The IGVC Committee and Officials will adhere to the above official competition details, rules and format as much as possible. However, they reserves the right to change or modify the competition where deemed necessary for preserving fairness of the competition. Modifications, if any, will be announced prior to the competition as early as possible.

II AUTONOMOUS CHALLENGE COMPETITION

All teams must pass Qualification to participate in this event.

II.1 OBJECTIVE

A fully autonomous unmanned ground robotic vehicle must negotiate around an outdoor obstacle course under a prescribed time while staying within the 5 mph speed limit, and avoiding the obstacles on the track.

Judges will rank the entries that complete the course based on shortest adjusted time taken. In the event that a vehicle does not finish the course, the judges will rank the entry based on longest adjusted distance traveled. Adjusted time and distance are the net scores given by judges after taking penalties, incurred from obstacle collisions, pothole hits, and boundary crossings, into consideration.

II.2 VEHICLE CONTROL

Vehicles must be unmanned and autonomous. They must compete based on their ability to perceive the course environment and avoid obstacles. Vehicles cannot be remotely controlled by a human operator during competition. All computational power, sensing and control equipment must be carried on board the vehicle.

II.3 OBSTACLE COURSE

The course will be laid out on grass, pavement, simulated pavement, or any combination, over an area of approximately 60 to 120 yards long, by 40 to 60 yards wide and be 700 to 800 feet in length. This distance is identified so teams can set their maximum speed to complete the course pending no prior violations resulting in run termination. The course boundaries will be designated by continuous or dashed white and/or yellow lane markers (lines) approximately three inches wide, painted on the ground. Track width will be approximately ten feet wide with a turning radius not less than five feet. Alternating side-to-side dashes will be 15-20 feet long, with 10-15 feet separation.

Expect natural or artificial inclines with gradients not to exceed 15%, sand pit (sand depth 2 - 3 inches) and randomly placed obstacles along the course. The course will become more difficult to navigate autonomously as vehicle progresses. The sand pit may be simulated with a light beige canvas tarp covering the entire width of the track for ten feet.

Obstacles on the course will consist of various colors (white, orange, brown, green, black, etc.) 5-gallon pails, construction drums, cones, pedestals and barricades that are used on roadways and highways. Natural obstacles such as trees or shrubs and manmade obstacles such as light post or street signs could also appear on the course. The placement of the obstacles may be randomized from left, right, and center placements prior to every run.

Potholes will be two feet in diameter and two inches in depth will be placed on the course. Simulated potholes are two feet diameter white circles, and may also be used on the course (Course width will be adjusted here to insure minimum passage width).

There will be a minimum of six feet clearance, minimum passage width, between the line and the obstacles, i.e. if the obstacle is in the middle of the course then on either side of the obstacle will be six feet of driving space. Or if the obstacle is closer to one side of the lane then the other side of the obstacle must have at least six feet of driving space for the vehicles.

Also in the event will be complex barrel arrangements with switchbacks and center islands. These will be adjusted for location between runs. Direction of the obstacle course may also be changed between heats.



Examples of Obstacle Configurations on the Autonomous Course

II.4 COMPETITION PROCEDURES

- The competition will take place in the event of medium rain but not in heavy rain or lightning.
- Each qualified team will have the up to two runs (time permitting) in each of three heats.

- Judges/officials will assign a designated starting order. Teams will setup on-deck in that order. Failure to be on-deck will place you at the end of the order for the run and may forfeit you final (second) run in a heat based on heat time completion.
- No team participant is allowed on the course before the team's first run, and only one team member is allowed on the course during a run.
- At the designated on-deck time, the competing team will be asked to prepare their vehicle for an attempt. On-deck teams start in the order they arrive in the starting area unless they give way to another team.
- The Starting Official will call teams to the starting line. The Starting Officials' direction is final; the Starting Official may alter the order to enhance the competition flow of entries (e.g. slower vehicles may be grouped together to allow the running of two vehicles on the course simultaneously).
- A team will have one minute in the starting point to prep the vehicle for the judge to start the vehicle.
- The judge will start the vehicle by a one touch motion; i.e. hitting the enter key of a keyboard, a left mouse click, lifting the e-stop up, flipping a toggle switch, etc.
- An attempt will be declared valid when a designated judge gives the start signal at the designated competing time. An attempt will continue until one of the following occurs:
 - The vehicle finishes the course.
 - The vehicle was E-Stopped by a judge's call.
 - The team E-Stops the vehicle.
 - Five minutes have passed after the vehicle run has started.
 - The vehicle has not started after one minute after moving to the start line or at the judges' discretion.
 - Teams will ready the vehicle for start and advice the IGVC Judge to push one button to start the vehicle, the vehicle must be ready for a judges start command in one minute.
 - An official will carry the E-Stop.
- Time for each heat will be strictly observed.
- Tactile sensors will not be allowed.
- Actual potholes (two feet in diameter and two inches deep) or simulated potholes (two feet diameter top soil circle) will introduced on the course.
- Each vehicle will be given 5 minutes per attempt to complete the course, if the vehicle has not completed the course in the 5 minute time period, it will ended by a judge's choice E-stop, with no penalty assigned for that run.
- Each vehicle must navigate the course by remaining inside the course boundaries and navigating around course obstacles. For the following Traffic Violations, the appropriate ticket will be issued and deducted from the overall distance or time score. Refer to section II.5 Traffic Violation Laws.

II.5 Practice Course

All teams that have Qualified will be given six tokens. Each token represent one opportunity to use the Autonomous Challenge Practice Course. The course will be open daily for use from the time a team Qualifies till the start of the third heat of the Autonomous Challenge. The course will be run like the Autonomous Challenge with the same rules and similar obstacles. One token allows a maximum of six minutes (one minute at the start point and five minutes for the run) on the Autonomous Challenge Practice Course. In that time you must position your vehicle at the start, prep the vehicle for the judge to start, and can continue to run as long as you do not break any of the rules of the Autonomous Challenge. If so, your run and remaining time will be ended. All teams will still have unlimited access to the regular practice fields.

II.6 TRAFFIC VIOLATION LAWS

	Traffic Violations	Ticket Value	E-Stop	Measurement
1	Leave the Course/Scene	- 10 Feet	Yes	Yes
2	Crash/Obstacle Displacement	- 10 Feet	Yes	Yes
3	Careless Driving	- 5 Feet	No	No
4	Sideswipe/Obstacle Touch	- 5 Feet	No	No
5	Student's Choice E-Stop	- 5 Feet	Yes	Yes
6	Judge's Choice E-Stop	0 Feet	Yes	Yes
7	Potholes	- 5 Feet	No	No
8	Blocking Traffic	- 5 Feet	Yes	Yes
9	Loss of Payload	0 Feet	Yes	Yes

- **Leave the scene\course:** All portions of the vehicle cross the boundary. The overall distance will be measured from the starting line to the furthest point where the final part of the vehicle crossed the boundary outside edge.
- **Crash:** The overall distance will be measured from the starting line to the collision point with the obstacle.
- **Careless Driving:** Crossing the boundary while at least some part of the vehicle remains in bounds.
- **E-Stop:** The overall distance will be measured from the starting line to the front of the vehicle or where the final/furthest remaining part of vehicle if stopped, crossed the boundary outside edge.
- **Obstacle Displacement:** Defined as displacing permanently the obstacle from its original position. Rocking/Tilting an obstacle with no permanent displacement is not considered obstacle displacement. Rocking/Tilting is a – 5 feet ticket value.
- **Actual and Simulated Potholes:** Hitting or crossing potholes will result in – 5 feet ticket value for each pothole hit or crossed and render run ineligible for standard prize money.
- **Blocking Traffic:** Vehicles stopping on course for over one minute will be stopped and measured.

II.7 HOW COMPETITION WILL BE JUDGED

- A team of judges and officials will determine compliance with all rules.
- Designated competition judges will determine the official times, distances and ticket deductions of each entry. At the end of the competition, those vehicles crossing the finish line will be scored on the time taken to complete the course minus any ticket deductions. Ticket values will be assessed in seconds (one foot = one second) if the vehicle completes the course within the five minute run time.
- The team with the adjusted shortest time will be declared the winner.
- In the event that no vehicle completes the course, the score will be based on the distance traveled by the vehicle minus the ticket deductions. The team with the adjusted longest distance will be declared the winner.
- For standard award money consideration, entry must exhibit sufficient degree of autonomous mobility by passing the money barrel. The money barrel location is determined by the judges

during the final/actual course layout. If a tie is declared between entries, the award money will be split between them.

II.8 GROUNDS FOR DISQUALIFICATION

- Judges will disqualify any vehicle which appears to be a safety hazard or violate the safety requirements during the competition.
 - Intentional interference with another competitor's vehicle and/or data link will result in disqualification of the offending contestant's entry.
 - Damaging the course or deliberate movement of the obstacles or running over the obstacles may result in disqualification.
 - Actions designed to damage or destroy an opponent's vehicle are not in the spirit of the competition and will result in disqualification of the offending contestant's entry.
-

III. DESIGN COMPETITION

All teams must participate in the Design Competition.

III.1 OBJECTIVE

Although the ability of the vehicles to negotiate the competition courses is the ultimate measure of product quality, the officials are also interested in the design strategy and process that engineering teams follow to produce their vehicles. Design judging will be by a panel of expert judges and will be conducted separate from and without regard to vehicle performance on the test course. Judging will be based on a written report, an oral presentation and examination of the vehicle.

Design innovation is a primary objective of this competition and will be given special attention by the judges. Innovation is considered to be a technology (hardware or software) that has not ever been used by this or any other vehicle in this competition. The innovation needs to be documented, as an innovation, clearly in the written report and emphasized in the oral presentation.

III.2 WRITTEN REPORT

The report should not exceed 15 letter-sized pages, including graphic material and all appendices, but not including the title page. Line spacing must be at least 1.5, with at least a 10 point font (12 is preferred). Each vehicle must have a distinct and complete report of its own (a report cannot cover more than one vehicle). Participants are required to submit four hard copies of the report and an electronic copy on a CD; failure to submit either of these will result in ***disqualification***. All reports, both for new vehicles and for earlier vehicles with design changes, must include a statement signed by the faculty advisor certifying that the design and engineering of the vehicle (original or changes) by the current student team has been significant and equivalent to what might be awarded credit in a senior design course. The certification should also include a brief description of the areas in which changes have been made to a vehicle from a previous year. Everything must be mailed so as to arrive by **May 17, 2010**, addressed to:

**Bernard Theisen
21281 Curie Avenue
Warren, MI 48091-4316**

Written reports arriving after that date will lose 10 points in scoring for each business day late, electronic copies arriving after that date will lose 5 points in scoring for each business day late. Teams are encouraged to submit reports even several weeks early to avoid the last minute rush of preparing vehicles for the competition, and there will be no penalty for last minute changes in the vehicle from the

design reported. The electronic copy of the report will be posted on the competition's web site in PDF format after the completion of the competition.

The paper should present the conceptual design of the vehicle and its components. Especially important to highlight are any unique innovative aspects of the design and the intelligence aspects of the vehicle. Also included must be descriptions of:

electronics	design planning process
electrical system	signal processing
actuators	plan for path following
software strategy	(both solid & dashed lines)
sensors	plan for control decisions
computers	system integration plan

Design of the lane following and obstacle detection/avoidance systems must be specifically described. Also, the system used for waypoint navigation should be detailed.

Components acquired ready-made must be identified, but their internal components need not be described in detail. The steps followed during the design process should be described along with any use of Computer-Aided Design. How considerations of safety, reliability, and durability were addressed in the design process should be specifically described, as well as problems encountered in the design process and how they were overcome. The analysis leading to the predicted performance of the vehicle should be documented, specifically:

- Speed
- Ramp climbing ability
- Reaction times
- Battery life
- Distance at which obstacles are detected
- How the vehicle deals with complex obstacles including switchbacks and center islands dead ends, traps, and potholes
- Accuracy of arrival at navigation waypoints
- Comparison of these predictions with actual trial data is desirable.

Although cost itself is not a factor in judging (these are considered research vehicles), the report should include a cost estimate (not counting student labor) for the final product if it were to be duplicated. A breakdown of the cost by component is helpful.

The team organization and the names of all members of the design team, with academic department and class, should be included along with an estimate of the project's total number of person-hours expended.

Vehicles that have been entered in IGVC in earlier years and have not had significant changes in design are ineligible in either the design or performance events. Vehicles that have been changed significantly in design (hardware or software) from an earlier year are eligible, but will require a completely new design report (15 pages or less) treating both the old and new features, thus describing the complete vehicle as if it were all new.

Judges will score the written reports as follows:	Maximum Points
1. Conduct of the design process and team organization (including decision-making & software development)	50
2. Completeness of the documentation	50
3. Quality of documentation (English, grammar, and style)	50
4. Effective innovation represented in the design (as described above)	150
5. Description of electronic design*	100
6. Description of software strategy*	150
7. Description of systems integration*	150

*Descriptions to include: lane following, obstacle detection/avoidance, and waypoint navigation (GPS or other)

8. Efficient use of power and materials	50
9. Attention given to safety, reliability, and durability	50
Total	800

III.3 ORAL PRESENTATION

The technical talk should relate the highlights of the written report described above and include any updates of the design since the written report. Audio or video tape presentations of the text are not allowed, but graphic aids may be presented by video, slide projection, computer projection, overhead transparencies, or easel charts. The presentation must be made by one or more student members of the team to the judges and other interested members of the audience and should last not more than 10 minutes. A penalty of 5 points will be assessed for each minute or fraction thereof over 11 minutes. After the presentation, judges only may ask questions for up to 5 minutes. The audience should be considered as a senior management group of generally knowledgeable engineers upon whom the project is dependent for funding and the team is dependent for their employment. Scoring will be as follows:

Judges will score the oral presentations as follows:	Maximum Points
1. Clear and understandable explanation of the innovations	50
2. Logical organization of the talk	25
3. Effective use of graphic aids	25
4. Articulation	20
5. Demonstrated simulation of vehicle control in performance events	10
6. Response to questions	10
7. Salesmanship	10
Total	150

Effective use of graphic aids includes not blocking the view of the screen by the presenter and simple enough graphics that are large enough to read (block diagrams rather than detailed circuit diagrams). Articulation refers to the clarity and loudness of speaking. Response to questions means short answers that address only the question. Salesmanship refers to the enthusiasm and pride exhibited (why this vehicle is the best).

Participants are responsible for providing their own visual aids and related equipment (the vehicle itself may be displayed). A computer-connected projector will be made available. Projectors may also be supplied by the participants.

III.4 EXAMINATION OF THE VEHICLE

The vehicle must be present and will be examined by the judges preferably immediately after the oral presentation or at another convenient time the time during the competition. Software is not included in this judging. Judging will be as follows:

Judges will score the vehicle examinations as follows:	Maximum Points
1. Packaging neatness, efficient use of space	20
2. Serviceability	20
3. Ruggedness	20
4. Safety	20
5. Degree of original content in the vehicle (as opposed to ready-made)	50
6. Style (overall appearance)	20
Total	150

III.5 FINAL SCORING

The number of points awarded by the individual judges will be averaged for each of the 22 judging areas above, and these results will be offered to each participating team for their edification. The total of the average scores over all 22 areas (max 1100) will be used to determine the award winners.

When two teams of judges are used (due to a large number of entries) each judging team will determine the top three winners in their group, and the resulting six contestants will participate in a runoff of oral presentations and vehicle examinations judged by all judges. Those results will be combined with the original written report scores to determine the final winner.

IV. NAVIGATION CHALLENGE

All teams must pass Qualification to participate in this event.

IV.1 OBJECTIVE

Navigation is a practice that is thousands of years old. It is used on land by hikers and soldiers, on the sea by sailors, and in the air by pilots. Procedures have continuously improved from line-of-sight to moss on trees to dead reckoning to celestial observation to use of the Global Positioning System (GPS). The challenge in this event is for a vehicle to autonomously travel from a starting point to a number of target destinations (waypoints or landmarks) and return to home base, given only the GPS coordinates of the targets in latitude and longitude.

IV.2 ON-BOARD SENSORS

It is expected that most contestants will use Differential GPS, but non-differential GPS is allowed as well as dead reckoning with compasses, gyros, and wheel odometers. Vision systems and/or sonar and laser rangefinders may be used for obstacle detection. There are a number of handheld GPS systems that connect to laptop computers available on the market for under \$200. Garmin, SkyMap/GPS, and Magellan are some; Earthmate even has one for Macintosh. These may not be convenient to integrate in vehicle control programs. However, differential GPS units are available from Hemisphere GPS, Trimble, Thales, Magellan, Garmin, NovAtel, and Starlink (and possibly others). Differential correction signals are available in the Southeast Michigan area from the U.S. Coast Guard. WAAS or any of the commercial suppliers of corrections are also allowed. The use of a base station to supplement the on board GPS unit is not permitted.

IV.3 GPS COURSE

The map in the figure below shows a typical course for the Navigation Challenge. This is a practice map for use by teams during development of their vehicle. Coordinates for the actual navigation course waypoints will be given to the contestants on **June 5, 2010** in degrees latitude and longitude, but no XY coordinates will be provided. There will be three starting boxes (one for each heat) that serve also as waypoints. (To locate the course perimeter, the southwestern corner of the course will be 10 meters west and 30 meters south of the westernmost starting box. The course is oriented to true north.)

The competition course will be run on grass and will be approximately 55 by 65 meters (roughly 0.1 acre), and the total travel distance on the course will be on the order of 220 meters depending on the route chosen for the vehicles. The exact waypoint locations will be marked on the grass for use by the judges, but there will be no standup markers to indicate those positions. Construction barrels, barricades, fences, and certain other obstacles will be located on the course in such positions that they must be circumvented to reach the waypoints. These may be randomly moved between runs.

The course will be divided into two areas by a fence with a 2 meter wide opening located somewhere along it (no coordinates are provided). The opening will be randomly relocated along the fence at the start of each run. Waypoints south of the fence (the Valley) will have 4 meter diameter circles or squares around them (visible only to the judges) and waypoints north of the fence (the Mesa) will have 2 meter circles around them.

No team participant is allowed on the course before the team's first run, and only one team member is allowed on the course during a run.

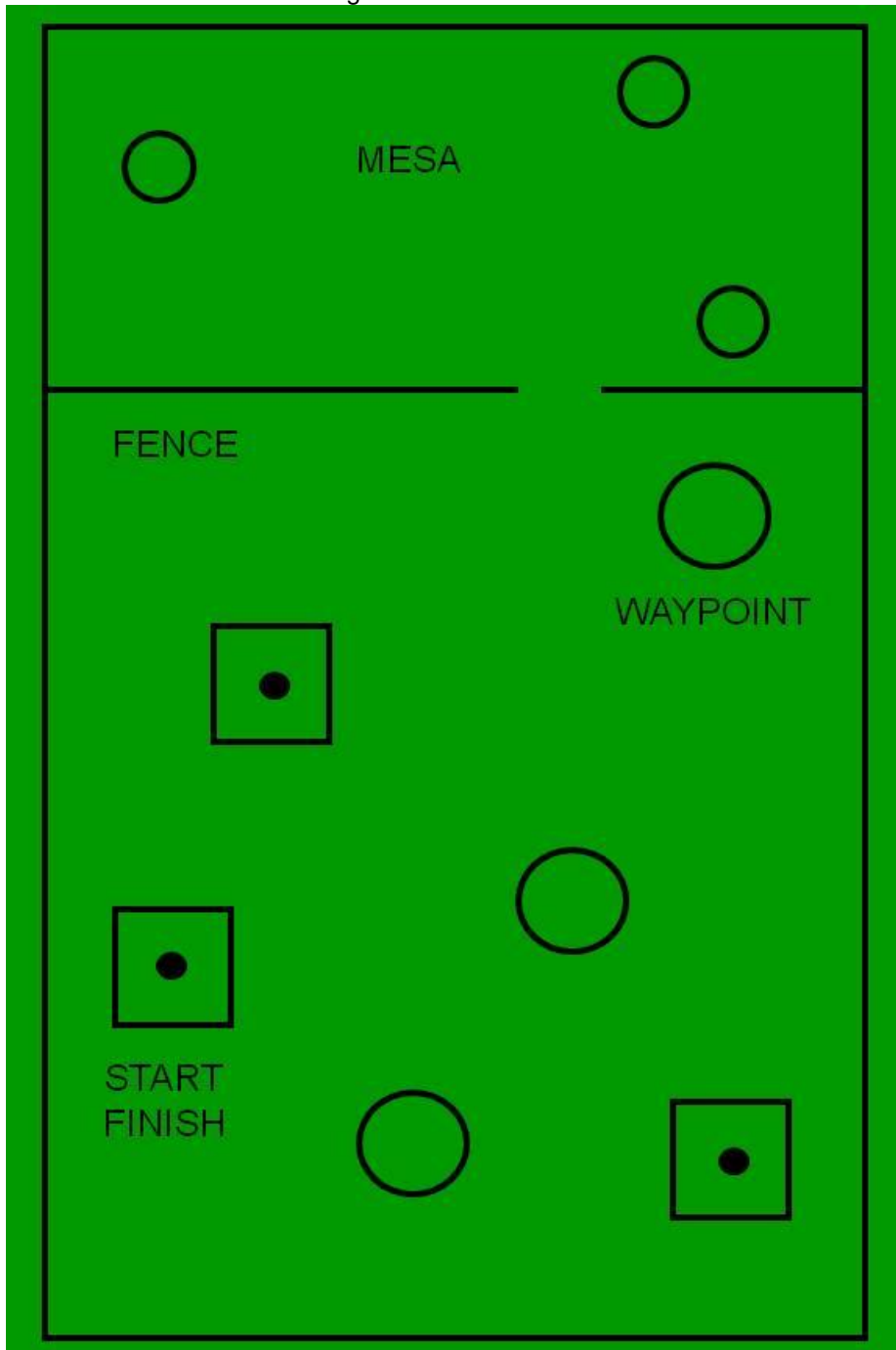


Figure 98: 2010 Practice Map for the Navigation Course

IV.4 RUN PROCEDURE AND SCORING

There will be three heats during the day with start and stop times the same as those in the Autonomous Challenge. It is intended that each team will be allowed up to two runs on the course during each of the three heats. The trial with the best performance will be used for scoring. Starting times will

be first-come-first-served within each heat, except that teams up for their first trial will have priority over those wanting a second trial. It is unlikely that there will be time in the day for all registered teams to get six tries.

Vehicles will park in the starting box selected by the judges for each heat and have 5 minutes for final adjustments before starting. Vehicles may seek the waypoints in any order, and the vehicle actually reaching the most waypoints (counting also the Start/Finish boxes) in the allotted six minute run time will be the winner. The vehicle must finish in the same box in which it started. If two or more vehicles reach the same number of waypoints, the vehicle doing so in the least time will be declared the leader. If two or more vehicles reach the same number of waypoints while stopped by the six-minute rule, they will be declared tied and will share any awards.

If a vehicle fails to come within two meters of a target in the southern area of the course or one meter in the northern area, it will not be judged to have reached that target. In order to qualify for standard award money a vehicle must reach at least six waypoints (not counting the start/finish box).

IV.5 RUN TERMINATION

All runs will be terminated by an E-stop (by the students or the judges) signaled by a judge's whistle or bell, either:

- When the vehicle arrives back at the original starting point or enters that starting box any time after first leaving it. If any part of the vehicle leaves the perimeter of the field
- If the vehicle strikes any obstacle.
- If six minutes have elapsed since the start of the run (220 meters in six minutes is 1.4 miles per hour).
- In all cases the judges' call will be final.

V. JAUS CHALLENGE

Participation in the JAUS Challenge is recommended.

V.1 TECHNICAL OVERVIEW

Each entry will interface with the Judge's COP providing information as specified below. The general approach to the JAUS interface will be to respond to a periodic status and position requests from the COP. This requires the support of the JAUS Transport Specification (AS5669A) and the JAUS Core Service Set (AS5710). The JAUS Transport Specification supports several communication protocols, the competition will use only the Ethernet based JUDP. The Core services required for the competition include the discovery, access control, and management services. The JAUS Mobility Service Set (AS6009) or JSS-Mobility defines the messaging to be used for position communications and waypoint based navigation.

V.2 COMMON OPERATING PICTURE

The COP will provide a high level view of the systems in operation that successfully implement the JAUS protocol as described above. This software is a simple validation, reporting and recording tool for the Judges to use while verifying student implementations of the JAUS standard. It provides a graphical display of the operational area in relative coordinates. Primitive graphics are loaded in the display of the COP to add perspective. Each reported status is displayed on the COP user interface and recorded for future reference. For competitions and systems reporting positional data, a 2-D map on the COP display is annotated with the updated position as well as track marks showing the previous position of the system for the current task.

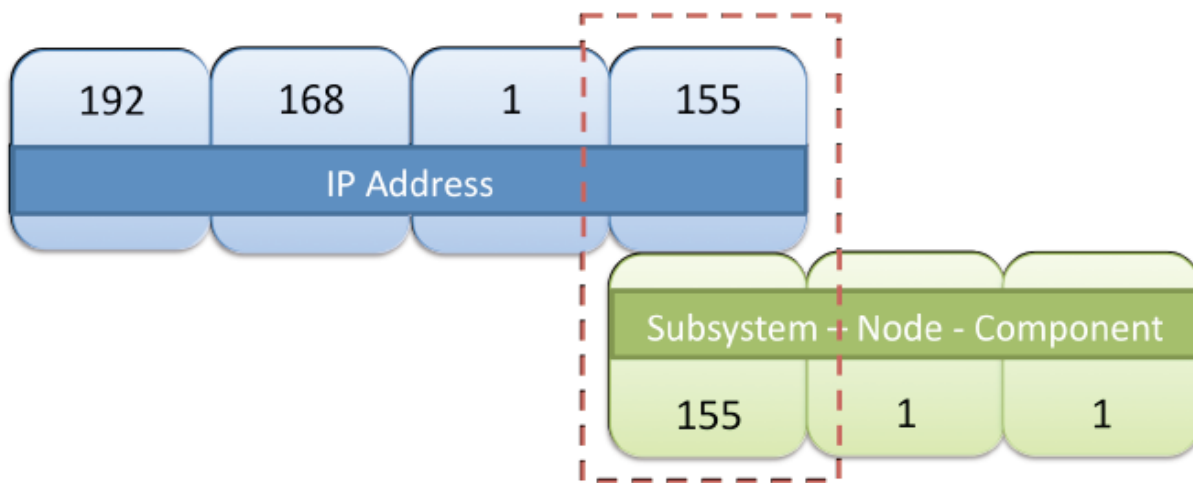
V.3 COMMUNICATIONS PROTOCOLS

The teams will implement a wireless 802.11b/g or hardwired Ethernet (RJ-45) data link. The interface can be implemented at any point in the student team's system including the control station or mobility platform.

The Internet Protocol (IP) address to be used will be provided at the competition. For planning purposes, this address will be in the range of 192.168.1.100 to 192.168.1.200. The Judge's COP will have both hard-wire and 802.11b/g capabilities where the IP address of the COP will be 192.168.1.42. All teams will be provided an IP address to be used during the competition. The last octet of the IP address is significant, as it will also be used as the subsystem identifier in the team's JAUS ID. The port number for all JAUS traffic shall be 3794.

V.4 JAUS SPECIFIC DATA

The JAUS ID mentioned above is a critical piece of data used by a JAUS node to route messages to the correct process or attached device. As indicated above each team will be provided an IP address in which the last octet will be used in their respective JAUS ID. A JAUS ID consists of three elements, a Subsystem ID, a Node ID and a Component ID. The Subsystem ID uniquely identifies a major element that is an unmanned system, an unmanned system controller or some other entity on a network with unmanned systems. A Node ID is unique within a subsystem and identifies a processing element on which JAUS Components can be found. A Component ID is unique within a Node represents an end-point to and from which JAUS messages are sent and received. The last octet of the assigned IP address will be used as the team's JAUS Subsystem ID. So for the team assigned the IP address of 192.168.1.155, the completed JAUS ID of the position-reporting component might be 155-1-1 where the node and component are both assigned the IDs of 1. This is shown in the IP and JAUS ID Assignment Figure below. The Node ID and Component ID are discussed further in the JAUS Service Interface Definition Language standard (AS5684). The COP software will be programmed with the assumption that all services required by the specific competition are implemented on a single component.



IP and JAUS ID Assignment

In summary, each team will be assigned an IP address by the judges. The last octet of that IP address will be the team's subsystem identifier. The COP will be a subsystem as will each team's entry in the competition. The COP will have a JAUS ID of 42:1:1 and an IP address of 192.168.1.42. The port number shall be 3794.

V.5 COMPETITION TASK DESCRIPTION

Messages passed between the COP and the team entries will include data as described in the task descriptions below. The COP will initiate all requests subsequent to the discovery process described as Task 1. A system management component is required of all teams. This interface will implement several of the messages defined by the Management Service defined in the JSS-Core. This service inherits the Access Control, Events and Transport services also defined by the JSS-Core document. The implementation of the Access Control interfaces will be necessary to meet the JAUS Challenge requirements; however no messages from the Events service will be exercised. The sequence diagram in Discovery and System Management Figure shows the required transactions for discovery including the access control setup and system control protocol. This interaction is required for every task.

The judges will evaluate each team's ability to meet the Interoperability Challenge for the tasks described below in accordance with the scoring chart.

Judges will score the task as follows:	Maximum Points
1. Transport Discovery	10
2. Capabilities Discovery	10
3. System Management	10
4. Velocity State Report	10
5. Position and Orientation Report	10
6. Waypoint Navigation	10
Total	60

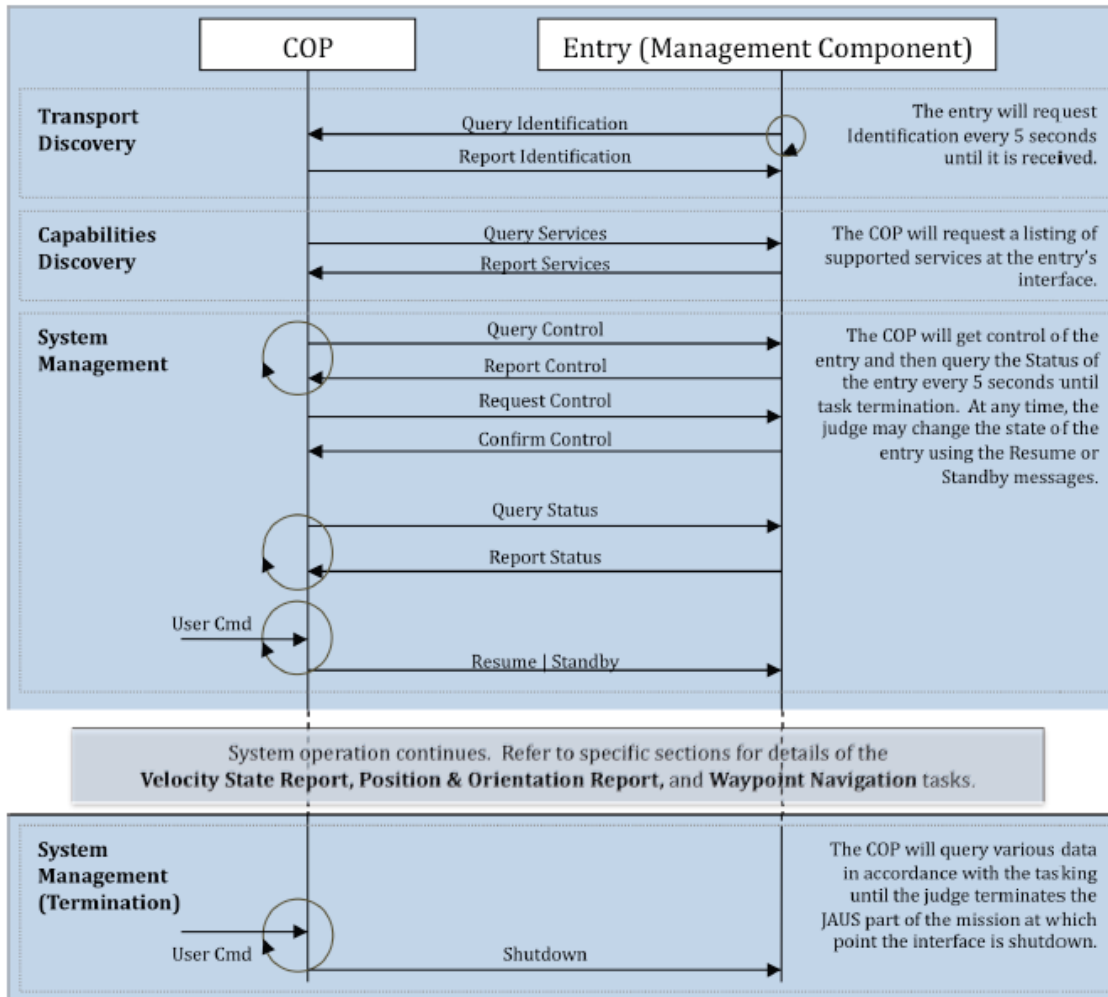
V.6 TRANSPORT DISCOVERY

For any two elements in the system to communicate meaningful data there must first be a handshake to ensure both sides use the same protocols and are willing participants in the interaction. For the sake of simplicity, the team's entry shall initiate the discovery protocol with the Judge's COP, and the IP address and JAUS ID of the COP shall be fixed. The IP address and JAUS ID of the Judge's COP are defined as:

COP IP ADDRESS:	192.168.1.42:3794
COP JAUS ID:	42-1-1 (Subsystem-Node-Component)

The discovery process, in Discovery and System Management Figure, will occur at the application layer. The student team's JAUS element will send a request for identification to the COP once every 5 seconds. The COP will respond with the appropriate informative message and request identification in return from the team's JAUS interface. After the identification report from the COP, the team entry will stop repeating the request. This transaction will serve as the basic discovery between the two elements.

The COP software will be programmed with the assumption that all services required by the specific competition are provided at the single JAUS ID. Furthermore, as per the AS5669A Specification, the team's entry shall receive JUDP traffic at the same IP address and port number that initiated the discovery protocol. Teams should note that this is different from common UDP programming approaches in which the outbound port for sent messages is not bound.



Discovery and System Management

The following table shows the messages sent from the COP to the team's entry, along with the expected response and minimal required fields to be set using the presence vector (PV) if applicable, required to complete this portion of the challenge:

Input Messages	Expected Response	Required Fields (PV)
Query Identification	Report Identification	N/A

V.7 CAPABILITIES DISCOVERY

Following the completion of the Transport Discovery handshake the COP will query the entry for its capabilities. The Query Services message and Report Services message are defined in the AS5710 document and require the inheritance of the Transport service. The COP will send a Query Services message to a student team entry. Upon receipt of the message the student team entry shall respond with a properly formed Report Services message.

The following table shows the messages sent from the COP to the team's entry, along with the expected response and minimal required fields to be set using the presence vector (PV) if applicable, required to complete this portion of the challenge:

Input Messages	Expected Response	Required Fields (PV)
Query Identification	Report Identification	N/A

V.8 SYSTEM MANAGEMENT

The implementation of the status report is required. This interoperability task, like the discovery tasks above, is also a prerequisite for all other tasks. The task begins with the discovery handshake as described above and continues for an indeterminate period of time. The protocol is given in Discovery and System Management Figure. The following table shows the messages sent from the COP to the team's entry, along with the expected response and minimal required fields to be set using the presence vector (PV) if applicable, required to complete this portion of the challenge:

Input Messages	Expected Response	Required Fields (PV)
Query Control	Report Control	N/A
Request Control	Confirm Control	N/A
Query Status	Report Status	N/A
Resume	<none>	N/A
Standby	<none>	N/A
Shutdown	<none>	N/A

V.9 VELOCITY STATE REPORT

In the Velocity State Report task the COP will query the entry for its current velocity state. The COP will send a Query Velocity State message to a student team entry. Upon receipt of the message the student team entry shall respond with a properly formed Report Velocity State message.

The following table shows the messages sent from the COP to the team's entry, along with the expected response and minimal required fields to be set using the presence vector (PV) if applicable, required to complete this portion of the challenge:

Input Messages	Expected Response	Required Fields (PV)
Query Velocity State	Report Velocity State	Velocity X, Yaw Rate & Time Stamp [320 Decimal, 0140h]

V.10 POSITION AND ORIENTATION REPORT

For performing the task Position and Orientation Report, the discovery and status protocols described above are also required. In addition to the COP queries for status, the vehicle systems will also be required to respond correctly to local position queries. The reports will be validated for relative position and with respect to a relative time offset to ensure the time contained within each position report is valid with respect to some timer within the entry's system. In other words, the position reports must show that the travel occurred at a reasonable speed and not instantaneously. Additional variation in the position reporting using the available presence vectors is allowed. Minimally, all entries must report X, Y and Time Stamp.

The following table shows the messages sent from the COP to the team's entry, along with the expected response and minimal required fields to be set using the presence vector (PV) if applicable, required to complete this portion of the challenge:

Input Messages	Expected Response	Required Fields (PV)
Set Local Pose	<none>	X, Y & Yaw [67 Decimal, 0043h]
Query Local Pose	Report Local Pose	X, Y & Time Stamp [259 Decimal, 0103h]

V.11 WAYPOINT NAVIGATION

The team entry shall implement the Local Waypoint List Driver service from the JAUS Mobility Service Set (AS6009). From a starting point in the JAUS challenge test area the student entry will be commanded to traverse, in order, a series of 4 waypoints. Time will be kept and will start at the moment that the student entry exits the designated start box. Upon leaving the start box the student entry will proceed to the first waypoint in the list. Upon satisfactorily achieving each waypoint the team will be credited with 2.5 points. Time is kept for each waypoint achieved. The shortest overall time taken to achieve this task will determine the winner in the event of a tie.

The following table shows the messages sent from the COP to the team's entry, along with the expected response and minimal required fields to be set using the presence vector (PV) if applicable, required to complete this portion of the challenge:

Input Messages	Expected Response	Required Fields (PV)
Set Element	Confirm Element Request	N/A
Query Element List	Report Element List	N/A
Query Element Count	Report Element Count	N/A
Execute List	<none>	N/Speed (value of 1)
Query Active Element	Report Active Element	N/A
Query Travel	Report Travel Speed	N/A
Query Local Waypoint	Report Local Waypoint	X & Y (value of 3)

VI. AWARDS AND RECOGNITION

All schools are only eligible to win award money once per event (Autonomous Challenge, Design Competition, Navigation Challenge and JAUS Challenge); if more than one team from the same school places in the same event, only the highest placing team will be placed in a standing and receive money for that event.

VI.1 AUTONOMOUS CHALLENGE COMPETITION

Autonomous Competition Standard Awards

1 ST Place	\$5,000
2 ND Place	\$4,000
3 RD Place	\$3,000
4 TH Place	\$2,000
5 TH Place	\$1,000
6 TH Place	\$ 750

Nominal Award Money

(Short of money barrel)

1 ST Place	\$1,000
2 ND Place	\$ 800
3 RD Place	\$ 600
4 TH Place	\$ 400
5 TH Place	\$ 200
6 TH Place	\$ 100

VI.2 VEHICLE DESIGN COMPETITION

Design Competition Standard Awards

1 ST Place	\$3,000
2 ND Place	\$2,000
3 RD Place	\$1,000
4 TH Place	\$ 750
5 TH Place	\$ 500
6 TH Place	\$ 250

Nominal Award Money

(Vehicle did not pass Qualification)

1 ST Place	\$ 600
2 ND Place	\$ 500
3 RD Place	\$ 400
4 TH Place	\$ 300
5 TH Place	\$ 200
6 TH Place	\$ 100

VI.3 NAVIGATION CHALLENGE COMPETITION

Navigation Competition Standard Awards

1 ST Place	\$4,000
2 ND Place	\$3,000
3 RD Place	\$2,000
4 TH Place	\$1,000
5 TH Place	\$ 750
6 TH Place	\$ 500

Nominal Award Money

(Did not make 7 waypoints)

1 ST Place	\$ 800
2 ND Place	\$ 600
3 RD Place	\$ 400
4 TH Place	\$ 300
5 TH Place	\$ 200
6 TH Place	\$ 100

VI.5 JAUS CHALLENGE

JAUS Competition Standard Awards

1 ST Place	\$3,000
2 ND Place	\$2,000
3 RD Place	\$1,000
4 TH Place	\$ 750
5 TH Place	\$ 500
6 TH Place	\$ 250

Nominal Award Money

(Vehicle did not pass Qualification)

1 ST Place	\$ 600
2 ND Place	\$ 500
3 RD Place	\$ 400
4 TH Place	\$ 300
5 TH Place	\$ 200
6 TH Place	\$ 100

VI.5 ROOKIE-OF-THE-YEAR AWARD

The Rookie-of-the-Year Award will be given out to a team from a new school competing for the first time ever or a school that has not participated in the last five competitions (for this year the team would be eligible if they haven't competed since the ninth IGVC in 2004). To win the Rookie-of-the-Year Award the team must be the best of the eligible teams competing and perform to the minimum standards of the following events. In the Design Competition you must pass Qualification, in the Autonomous Challenge you must pass the Rookie Barrel and in the Navigation Challenge you must make three waypoints. The winner of the Rookie-of-the-Year Award will receive \$1,000 in award money; in the case the minimum requirements are not met the best of the eligible teams competing will receive \$500.

VI.6 GRAND AWARD

The Grand Award trophies will be, presented to the top three teams that perform the best overall (combined scores per below), in all three competitions. For each competition, points will be awarded to each team, below is a breakdown of the points:

Autonomous Challenge	Passed Money Barrel	Short of Money Barrel
First Place	48	24
Second Place	40	20
Third Place	32	16
Fourth Place	24	12
Fifth Place	16	8
Sixth Place	8	4

Design Competition	Vehicle Qualified	Vehicle Failed to Qualify
First Place	24	12
Second Place	20	10
Third Place	16	8
Fourth Place	12	6
Fifth Place	8	4
Sixth Place	4	2

Navigation Challenge	Completed 7 Waypoints	Short of 7 Waypoints
First Place	36	18
Second Place	30	15
Third Place	24	12
Fourth Place	18	9
Fifth Place	12	6
Sixth Place	6	3

J AUS Competition	Vehicle Qualified	Vehicle Failed to Qualify
First Place	24	12
Second Place	20	10
Third Place	16	8

Fourth Place	12	6
Fifth Place	8	4
Sixth Place	4	2

VI.7 PUBLICATION AND RECOGNITION

International recognition of all participating teams through AUVSI and SAE publications.

Special recognition for the vehicles will be held at the Association for Unmanned Vehicle Systems International's Unmanned Systems North America 2010 Symposium & Exhibition Held at Colorado Convention Center in Denver, Colorado on August 24TH – 27TH, 2010

All teams are invited to display the winning vehicles in the AUVSI exhibit halls.

Videos of the competition event will be distributed to sponsors, media and the public. All design reports, articles, videos and pictures will be post on the IGVC website www.igvc.org.