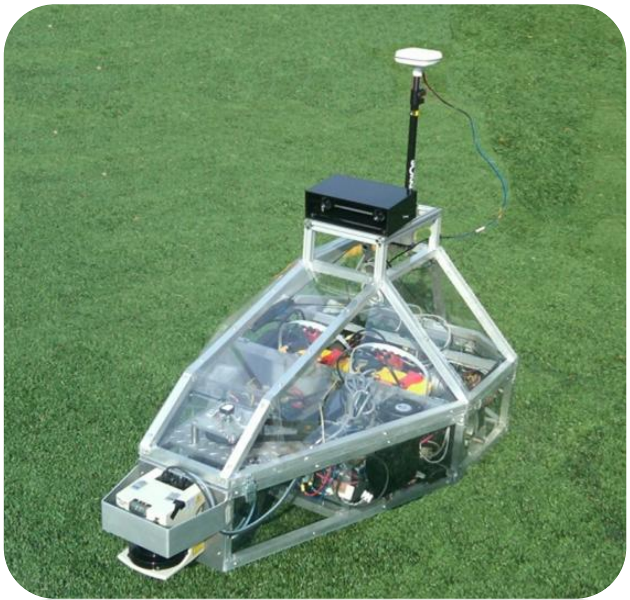
Prometheus

Worcester Polytechnic Institute



Team Prometheus:
Justin Barrett
Rob Fitzpatrick
Chris Gamache
Ricardo Madera
Adam Panzica
Benjamin Roy
Daniel Sacco
Viktoras Truchanovicius
Bohua (Jack) Wang

Faculty Advisor Statement:
I, professor Taskin Padir of the Robotics Engineering Program and Electrical Computer Engineering Department at Worcester Polytechnic Institute, Worcester do certify that the design and implementation of this vehicle has been credited to each team member for their work.

Date:__________ Signature:________________________
Professor Taskin Padir

Contents

1. Introduction

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 the successful completion of the project. To effectively distribute knowledge across the team and work in parallel, team Prometheus is organized into four subgroups: Sensors, Controls, Chassis and Power. Table 1 shows all the team members, their majors/minors, the year of graduation and the primary subgroup they belong to. More information about each team member can also be found on the team's website www.igvc-wpi.org.

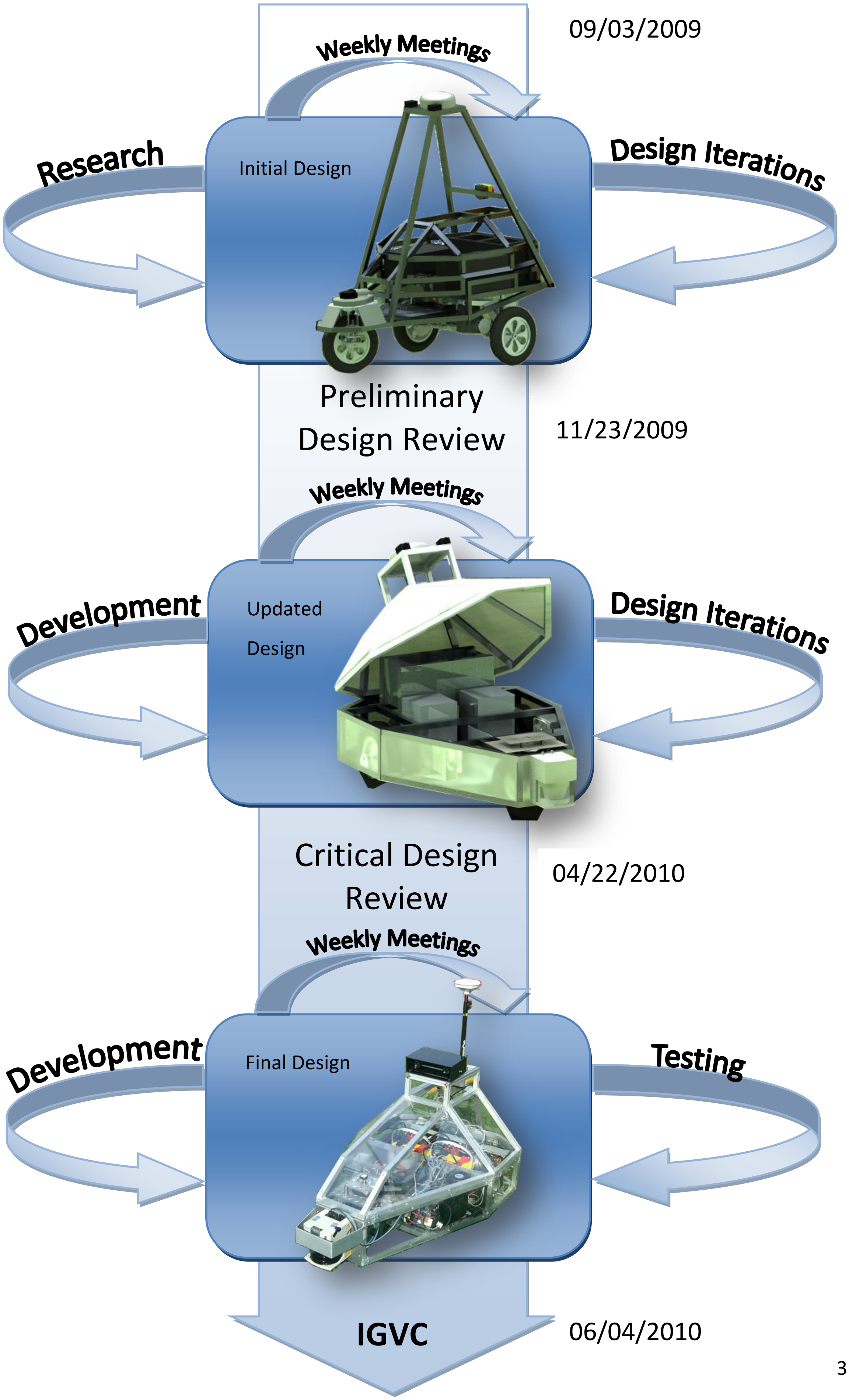
Name	Major	Primary Subgroup
Justin Barrett	Robotics Engineering 2010	Sensors
Rob Fitzpatrick	Mechanical Engineering 2010	Chassis
Chris Gamache	Robotics Engineering 2010	Sensors
Ricardo Madera	Computer Science 2010	Controls
Adam Panzica	Robotics/Mechanical Engineering 2010	Chassis
Benjamin Roy	Electrical and Computer Engineering 2010	Power
Daniel Sacco	Electrical and Computer Engineering 2010	Power
Viktoras Truchanovicius	Computer Science 2010	Controls
Bohua "Jack" Wang	Mechanical Engineering 2010	Chassis

Table 1: Team Prometheus

In order to ensure the proper communication amongst the subgroups there is an overlap of members in each subgroup. Every person in the team belonged to at least two subgroups: primary and secondary.

2. Design and Build Process

This project was completed by the team members as the senior capstone design requirement in their respective majors. The project timeline spanned throughout the entire 2009-2010 academic year. The team first established a set of design specifications based on the IGVC rules. Using an iterative development process the team and each subgroup conducted weekly meetings with project advisors. In each meeting the team presented weekly accomplishments, revised the timeline and reviewed the budget. On November 23rd 2009, the team held a preliminary design review (PDR) attended by the WPI faculty and students. Based on the feedback received from the PDR, the team redesigned the chassis of the vehicle. During the spring semester the team worked on building the vehicle, systems integration and the development of controls. On April 22nd 2010, Prometheus was presented to WPI community during the critical design review (CDR). The diagram below summarizes the design process followed by the team throughout the academic year.


09/03/2009
Weekly Meetings
Research
Initial Design
Design Iterations
Preliminary Design Review
11/23/2009
Weekly Meetings
Development
Updated Design
Design Iterations
Critical Design Review
04/22/2010
Weekly Meetings
Development
Final Design
Testing
IGVC
06/04/2010

3. Design Overview

Figure 1: Prometheus and its major components

Before discussing the subsystems that comprise Prometheus in detail, it is necessary to provide an overview of the vehicle. 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

The components and features of the vehicle are summarized in Table 2.

Feature	Component
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
Obstacle Detection	SICK LMS291-S05 LIDAR
Lane Following	Stereo Vision system using 2x FL2G-13S2M/C Cameras
Waypoint Navigation	Sokkia Axis 3 DGPS Receiver PNI V2Xe 2-Axis Compass Module
Localization	2x Quadrature Encoders
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 **Manual Control of the Vehicle** **Ease of Testing**	Control Center - The graphical user interface written using JAVA SWING
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
E-Stops	Wireless: IC Dynamics remote entry system Mechanical: circuit relays

Table 2: Design features of the Prometheus

Table 3 presents predicted and measured performance of the vehicle.

Attribute	Design Prediction	Measured Value/Approach
Speed	5.0mph	5.0mph
Ramp climbing ability	45 degrees	45 degrees
Reaction times	0.25s	0.3s
Battery life	40min	2h
Distance at which obstacles are detected	~10cm	~10cm
How vehicle deals with complex obstacles	Driving with Tentacles	Driving with Tentacles
Accuracy of arrival at navigation waypoints	~2m	To be tested
Nominal power consumption	1600 Watts	To be measured

Table 3: Predicted and measured vehicle’s performance

The predicted performance values presented above are based on the analysis carried out during the design process. The measured values reflect the current performance of the vehicle.

3.1 Design Innovations

From the very beginning team Prometheus kept innovation in mind. This is WPI's first time competing at the IGVC which was a driving force for the team to be innovative wherever the opportunity presented itself. Intelligent ground vehicle Prometheus has a number of unique features:

- A 3-wheel custom weatherproof chassis with a steered front wheel and differential drive.
- Path planning using *Driving with Tentacles* approach[1].
- Distributed hardware architecture for the control system.
- NVIDIA graphics processor used for image processing.

4. Electrical System

The intelligent ground vehicle Prometheus is powered by two 12V sealed lead acid batteries from Optima Batteries used in series to implement a 24V system. They have a capacity of 55Ah and are Optima's high performance batteries which will provide approximately 40 min of run time.

Prometheus utilizes two NPC T-64 series brushed DC motors on its drive wheels. The T-64 motors provide a solid compromise between output power, generating over 1kW at its peak value, and efficiency, drawing less than 300W each during normal operating conditions. They allow Prometheus to reach its ungoverned maximum speed of 3.6 m/s in less than two seconds. The motors are driven by Black Jaguar motor controllers by Texas Instruments running on 24V.

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 the vehicle subsystems via two fuse blocks that use auto-resettable fuses and relays for e-stops.

[1] 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.

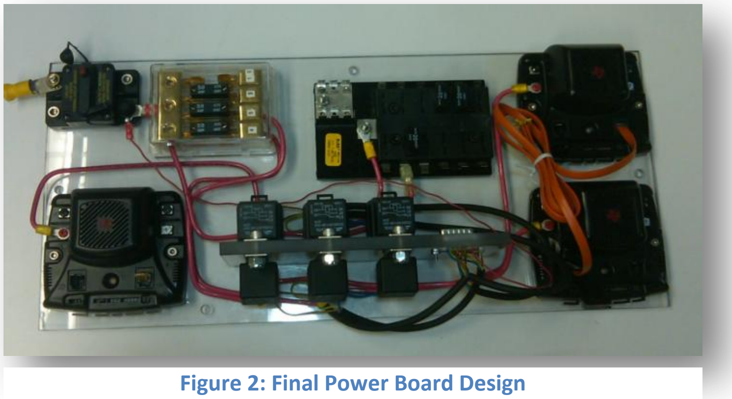


Figure 2: Final Power Board Design

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. The final power board is shows in Figure 2.

5. Sensor System

In view of the overview provided in Section 3, the sensors and their integration to the vehicle will be discussed next.

5.1 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. The SICK LIDAR system being used on the vehicle was received as a donation from a WPI alumnus.

5.2 DGPS

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 strings. 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.

5.3 Compass

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 modules.

5.4 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. Two FL2G-13S2M/C cameras from Point Grey have been selected for the stereo vision system due to their low cost.

5.5 Quadrature Encoders

The encoders being used on the vehicle are two US Digital E8P Optical Encoders. They have a range of 180 to 512 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 wheels.

6. Control System

Prometheus is designed to perform multiple challenging tasks like object detection, path finding, 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.

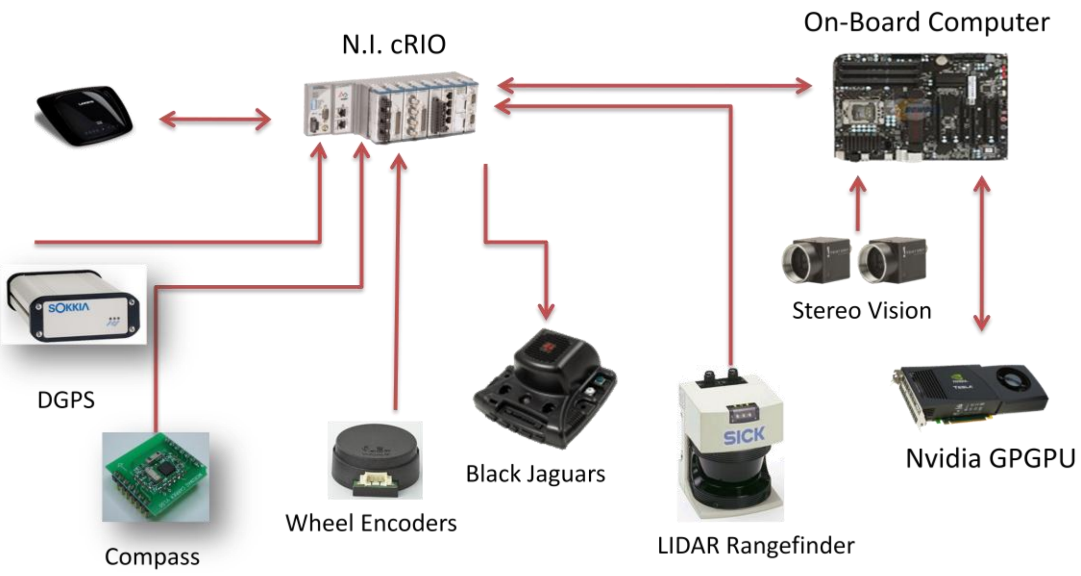


Figure 3: Vehicle Control System Diagram

6.1 Configuration

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 3. In the distributed system the cRIO is responsible for sensor interfacing and sending motor commands to the motor controllers. Two-way communication occurs between the cRIO and the main-board computer. The computer is responsible for image processing, path calculations, and maintaining the vehicle's map.

6.2 NI-cRIO Controller

The vehicle is equipped with a NI cRIO-9074 system. 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 as well as modules providing additional communication options. The cRIO is responsible for processing LIDAR, motor encoder, DGPS receiver, compass data, sending feedback to the main-board computer and executing path commands.

6.2.1 Controlling Motors

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

The main-board computer sends path information to the cRIO and once this information is parsed into a usable format, it is used to calculate the left and right motor velocities needed to follow the path. A manual control program is also implemented on the cRIO so that the vehicle can be controlled by a joystick.

6.2.2 PID Motor Control

PID motor control is implemented on the cRIO FPGA to control the orientation of the front wheel of the vehicle using encoder feedback. The desired orientation of the front wheel is based on the current speed of the back differential wheels which are measured by quadrature encoders. 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.

6.2.3 Processing LIDAR

The LIDAR driver establishes a connection with the cRIO through RS-232 and retrieves data packets containing an array of distance information. The packets are parsed and converted from polar coordinates into Cartesian coordinates which are then sent to the main-board computer for the map update.

6.2.4 Processing DGPS

The Sokkia Axis 3 DGPS receiver is interfaced through a serial port on the cRIO's RS-232 module. The incoming DGPS data is collected and converted from latitude and longitude into the Universal Transverse Mercator (UTM) coordinate system so that it can be used for vehicle localization. The first DGPS coordinate received is saved and becomes the origin of the vehicle's map. All subsequent DGPS coordinates have the origin subtracted from them so that the local vehicle position is always known with respect to its starting position. The DGPS data is also combined with encoder and compass data that is read into the cRIO. The compass provides two-axis (pitch and yaw) magnetic heading information and communicates over an RS-232 connection. The quadrature encoders are read in through the cRIO's

FPGA. The local position calculated from this data is sent to the main-board computer so that the local map can be updated.

6.2.5 Sensor Data Fusion

The current location of the vehicle is calculated using information returned from the motor encoders, the DGPS receiver, and the digital compass. This information is adjusted and fused together using an Extended Kalman Filter, and then sent to the main-board computer to be used in map updates and path calculations. In its most general terms, the Extended Kalman Filter uses the motor encoders to predict the next state (x, y positions and heading) of the robot and verifies and corrects the predicted state using the data provided by the DGPS receiver and the compass.

6.2.6 Communicating with the Main-Board Computer

The cRIO and main-board computer maintain communication by sending Universal Datagram Protocol (UDP) type Ethernet packets to each other. The cRIO is responsible for updating the main-board computer with position, heading, vehicle 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.

6.3 Software Architecture

The team chose two languages, C/C++ for the development of the main board software and Java for the Control Center implementation. The Software Architecture for the vehicle was an essential part for the software development cycle (Figure 4). Software architecture was constructed extending the OpenJaus library which provided a base implementation of parallel software architecture for an unmanned system. All other functionality that was not provided by the OpenJaus was created and integrated into the rest of the system. Extending OpenJaus included adding communication with the CRIO and Command Center, path planning and the stereo vision code. OpenJaus was not used for the implementation of JAUS protocol because it uses a reference architecture which is no longer supported by the competition.

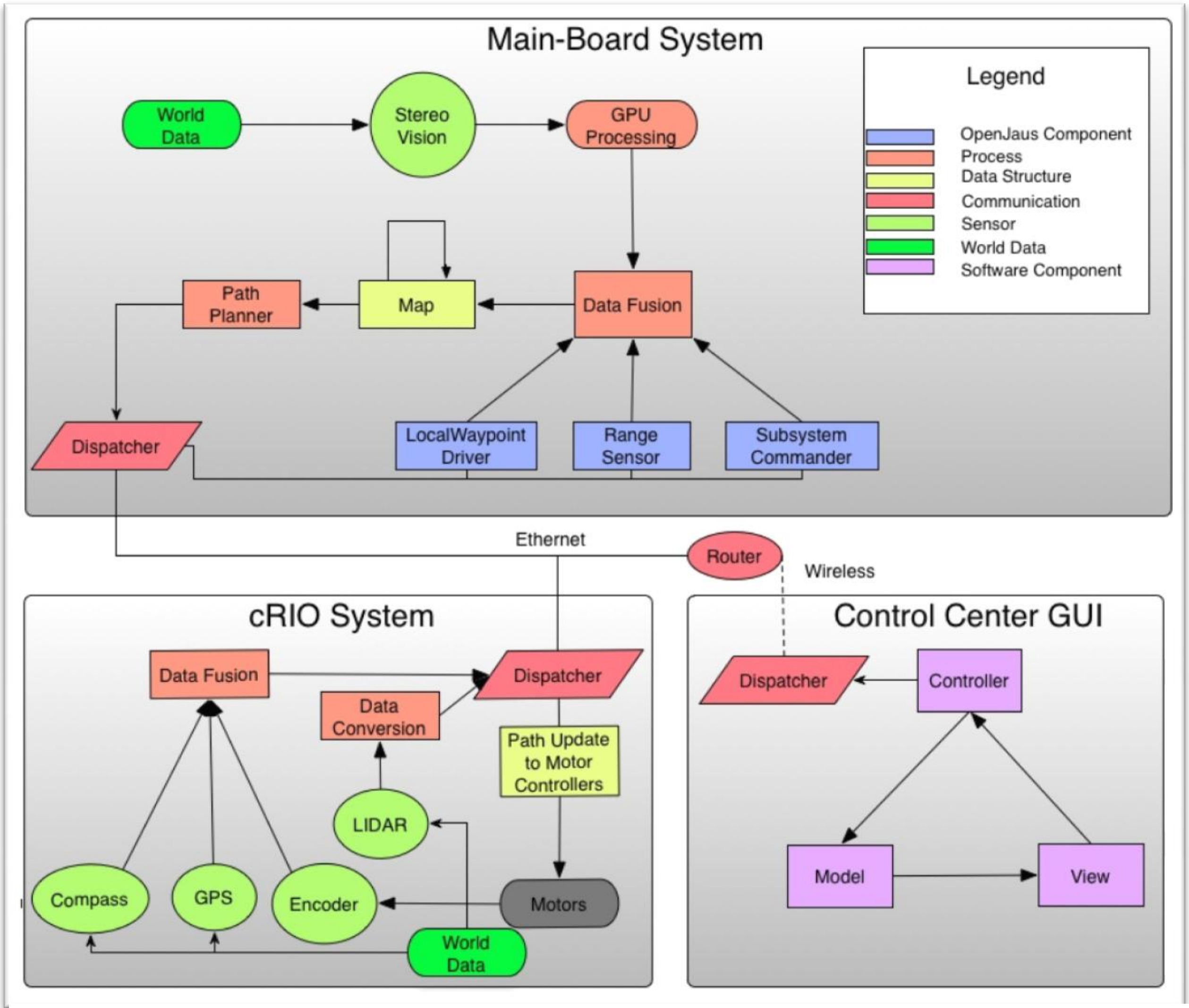


Figure 4: Control system architecture

6.3.1 Stereo Vision

Team Prometheus elected to use a stereo vision system to detect lines, potholes and to supplement the LIDAR for obstacle detection. To handle the immense amount of image processing required by the system, team Prometheus utilized NVIDIA's Tesla C1060 graphics processing unit (GPU). Since general purpose graphic processing is a relatively new concept most of the stereo vision functionality on the GPU was programmed by the team.

The process consists of three steps; rectification, segmentation and pixel matching. Image segmentation uses a neural network based approach to effectively segment the image based on color and is used to identify objects. Pixel matching creates a disparity map, which is used to calculate distances to the objects. The disparity map that results from pixel matching is then used to place objects on the vehicle's 2D probability map. Examples of image segmentation and disparity mapping are shown in Figure 5 and Figure 6.

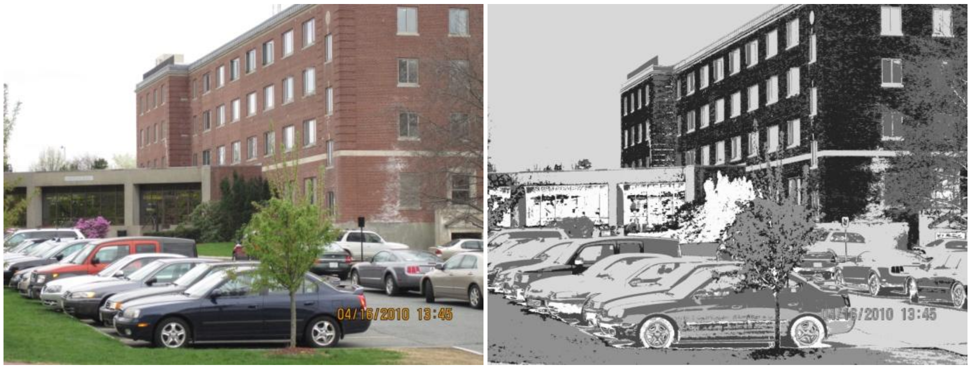


Figure 5: Color segmentation on outdoor environments

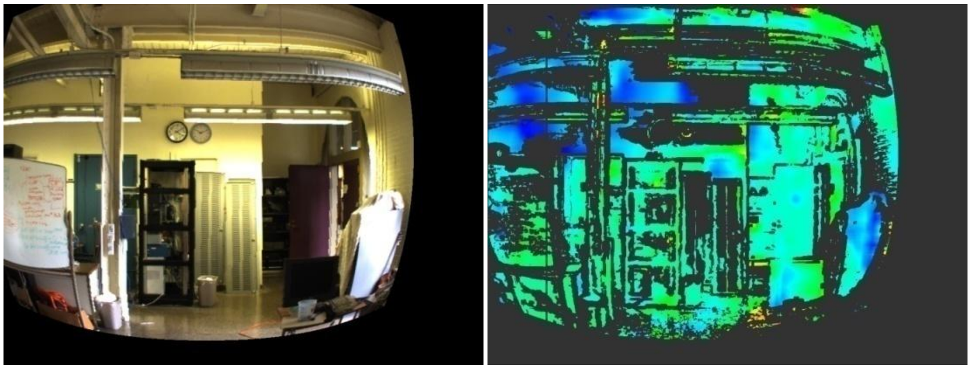


Figure 6: Original rectified image and resultant disparity map

6.3.2 Mapping and Path Planning

Since the path planning used in our approach is based on a world map, it is important to have the most accurate representation of the terrain and obstacles surrounding the vehicle. The map used for path planning is a top view 2D ego-centered probability map that is implemented using a 201x201 dimensional array of grid cells where the location and the heading of the vehicle is always the same. Each cell covers a 10x10cm area of the terrain and contains information including the probability of an obstacle being at the corresponding grid cell, the type of the obstacle (cone, line, pothole, etc.) and whether the cell has been updated since the last map change. Using this probability map greatly reduces the complexity of incorporating both LIDAR and stereo vision data. Since the vision and LIDAR systems run at different speeds it is essential that the map updates can be asynchronous. After each map update, a smoothing is performed over the LIDAR sweep area (180°) to reduce the probability of cells that did not get updated. Such smoothing provides error correction for LIDAR and stereo vision data.

Path planning is accomplished using *Driving with Tentacles* approach which is a method of path calculation implemented by creating 81 tentacles extending out from the turning center of the vehicle (Figure 7). Each tentacle is an arc along a circle with a specified diameter and 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 containing a different series of tentacles. As the speed increases, tentacles in the set curve less hence avoiding rolling at higher speeds.

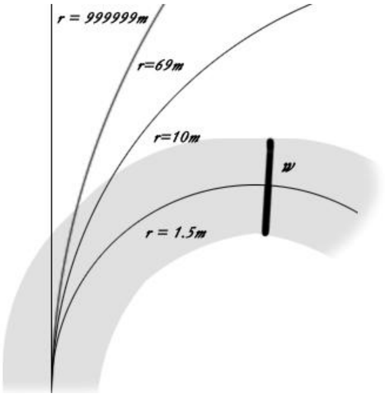


Figure 7: Example of three tentacles and their radii. Support area is in gray.

In order to select the best tentacle to be driven, several heuristics are employed. The heuristic for finding the best tentacle in any mode of operation (autonomous or navigation) takes into consideration the length of the tentacle, and the drivable length of the tentacle. The heuristic used in the navigation challenge takes into consideration the distance from each drivable bin (a 5cm segment on a tentacle) and the distance from the vehicle to the waypoint. The most preferable tentacle is picked empirically through testing and can be modified if needed.

6.3.3 JAUS

The integration of JAUS required research of the protocol and involved reading JAUS standards provided by the IGVC. After the team got familiar with the architecture the development and testing of the protocol began. Open source Junior Middleware library was used to implement JAUS compliant interface into the vehicle and JAUS Verification Tool (JVT) was used to verify and test the implementation. The team was able to integrate all of the required services. However, the JVT is not capable of testing them all so the verification and correction cannot be confirmed until the completion.

7. Conclusion

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.
- Distributed controls approach using 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. The team will continue with testing and improving the performance of the vehicle until the IGVC.

8. Budget

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**	