Oakland University - Zeus

Intelligent Ground Vehicle Competition 2007

Submitted by: Brian Clark, Micho Radovnikovich, Gaurav Saxena,
Phil Stene, Pavan Vempaty, Dhanasekaran Venugopal,

1. Introduction

Zeus, the Greek god of thunder and the ruler of Mount Olympus, is the metaphorical representation of our design; Zeus watches over and controls his realm using lightning, leaving bystanders awed by thunder.

Our version of Zeus is an AI, which utilizes a combination of C, C++, MatLab, and Simulink, running on a 2.8 GHz laptop and an 8 MHz Motorola HCS12 Microcontroller. The AI communicates with sensors (Web cams, Laser Range Finder, Inertial Measurement Unit, GPS, Digital Compass) using RS232 and USB. The actuators are controlled with I/O logic built into the HCS12 Microcontroller. Zeus is powered by two lead acid 12 Volt batteries in parallel.

The frame is constructed of one inch extruded aluminum covered with sheet aluminum. Zeus is steered by turning the drive shaft, which has a drive axle with two fixed wheels and a drive motor attached. The other two wheels are fixed in position located in the rear of the robot, but rotate independently of each other.

1.2 Team Description

A small team of student robotic enthusiasts at Oakland University produces Zeus. The majority of the hours worked has been as a hobby, with only a few subsystems completed for class credit. This has been a challenging year for us, as we have merged 3 previous independent teams together into a single team.

In 2005-2006 school year, Oakland University had 3 teams working on individual robots. With several individuals graduating from Oakland, all of the teams recognized to need to consolidate work into one project. After much planning and debating, we have integrated and expanded upon our past designs, producing some unique ideas. As a result, we have had 6 students shown in Table 1.2.1 working towards this year's entry.

Name	Major	Class	Hours	Related Interests
Brian Clark	Computer Science	Senior	350	Generic Programming, OSS, AI
Phil Stene	Computer Science	Senior	250	Logic Programming, Fabrication
Pavan Vempaty	Systems Engineering	Ph. D	100	AI, Vision, Navigation
Micho Radovnikovich	Electrical Engineering	Senior	200	Electronics
Dhanasekaran Venugopal	Systems Engineering	Masters	150	Obstacle detection analysis
Gaurav Saxena	Embedded Systems	Masters	100	GPS navigation

Table 1.2.1 – List of participating

1.3 Innovations

This year has brought forth many innovations, which we highlight here and describe later in detail. One significant change that we have had has been the merging of different software programs, so that we use both C/C++ with MatLab/Simulink working to compliment one another, providing a more main table program. Another innovation with software is a new technique we are using to handle vision software, were we run different algorithms in conjunction, so we cat produce more types of data, along with data compatibility checking, to gain an improvement in vision performance. The last major innovation with the software is the addition of a JAUS module, which we will use for the JAUS challenge.

We have made several physical improvements in our design this year. First, we were able to reorganize the physical design of the robot to allow us to mount sensors on the side with the drive motor, allowing us to effectively operate in front wheel drive. Next, we have created different power circuits to isolate the motors from the rest of the electronic devices on the robot. Lastly, we decided that there needed to be a better interface allow basic control the robot, and such we have implemented a User Interface Control Box, which allows a user to set a new operating mode, and then initiate a run.

2. Design Process

In developing Zeus several questions were asked. We discussed these questions and organized the answers into the following: essential features, useful features, and nice features. Once our priority list was created we incorporated it into the design process illustrated in figure 2.2.1.

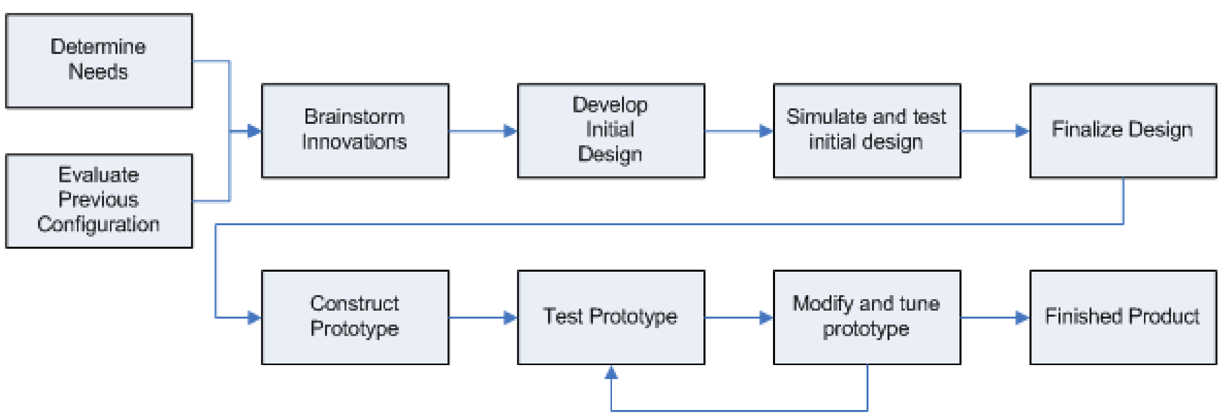


Figure 2.2.1

Naturally, the first questions were; what are the requirements needed for the competition? How does the previous prototype configuration satisfy those needs? How can we improve the prototype to better fulfill those needs? What is a reasonable cost for the vehicle? To answer these questions we examined the current industry/consumer products currently available, examined our previous configuration, and proceeded to brainstorm appropriate innovations to improve our design. Once we had finished with our initial design we began testing and simulating the design in a laboratory environment. The team agreed upon a final design and a prototype was constructed. The prototype was put through a vigorous testing phase to find bugs and otherwise tune the design. Once the design seemed to operate within tolerances we began to field test the vehicle. Field-testing was used to determine durability, run time, usability, and overall performance.

3. Zeus Design

3.1 Zeus Overview

The current Zeus design comes from a search for a more efficient Energy/Weight/Drive combination. Both mechanical, electrical and software are designed to synchronize with respect to the control signals and maintain harmony through out. The whole design process is described in the following sessions.

3.2 Mechanical Design

3.2.1 Frame Design

In order to maintain a design that is light, strong, and flexible, 1 inch extruded aluminum for the base frame was used. Aluminum sheet metal was used to cover the extruded aluminum, along with some sections of Plexiglas to provide viewing windows. The use of Plexiglas was minimized in order to reduce the internal temperature. Even after the integration of all the teams we did not have any Mechanical Engineering majors. As such, we did not wish to undertake a massive redesign of the frame, as it is already working sufficiently.

3.2.2 Front Wheel Drive

One of the improvements that we decided to make this year was to move from a rear wheel drive system to a front wheel drive system. In order to do this, we had to reposition the steering motor, and then modify and relocate the mounting supports for the SICK laser range finder.

3.2.3 Drive/Steering Method

The downside to our steering mechanism is that one or both of the drive wheels must slip a little in order to turn. However, this is alleviated by the fact that in normal operation, we only have slight changes in steering, and the slipping is mitigated while driving. We feel that this design is beneficial, as the steering and drive mechanisms are straightforward, and so is the control logic. We also gain a very small turning radius, about 1 foot from center, giving us good maneuverability.

3.2.4 Motors/Wheels

Two motors are used to control the motion of the vehicle. One is used to control the angle of the front wheels, enabling the vehicle to turn, and another is used to turn the wheels to make the vehicle move. The wheels on Zeus are 10 inches in diameter.

The maximum speed of the vehicle was measured by observing how fast the wheels spin at maximum motor speed with no load. Over several samples, the average angular velocity of the wheels was found to be 164 RPM. Converting this to the corresponding translational speed of the vehicle brings us to 4.88 mph.

3.2.5 Sensor Mounts

This robot can easily handle eight sensor mounts in their prescribed locations. This robot is also equipped with a stabilization platform for the cameras to have level view regardless of terrain disturbances.

3.2.6 Stability

The spacing between the wheels is sufficient to keep the robot stable and balanced. To avoid roll-over, the main battery was placed in the middle of the frame such that the center of gravity of the vehicle is close to the center as well. The vehicle is will not roll over under normal operating conditions.

3.2.8 Weather Resistance

To help maintain the ability to operate in light rain, two layers of weatherproofing are used. First, the overall case is sealed, so that water cannot enter directly into the frame. Second, all individual electronics are sealed into separate boxes, so that a small leak will not affect sensitive electronics.

3.2.9 Maintenance, Repair, and Physical Access

The doors on the side of the frame allow easy access to the various components on the inside of the vehicle. Everything is packaged such that each component can be moved, replaced or fixed without greatly disturbing the rest of the layout. This greatly helps during the testing process, when many changes need to be made.

3.3 Electrical & Electronic Distribution

3.3.1 Overview

The entire system is powered with two 12-volt batteries. One of the batteries has a 75 Ah capacity and is used to power the motors. The other battery has a 10 Ah capacity and is used to power the on-board electronics, including the SICK, compass, GPS unit, IMU, microcontroller and e-stop. However, the SICK requires 24 volts to operate, so a 12V – 24V converter is used to provide the proper voltage. All power is routed through a central power box, and each component in the system has its own power switch on the box. The computer uses its own battery, and also powers the webcams and joystick. If necessary, a power inverter can be used to convert the 12 VDC of one of the batteries into 110 VAC to power the computer for long runs. Figure

3.2.1a shows a block diagram of the power system with solid orange lines for power flow and dashed green lines for data flow.

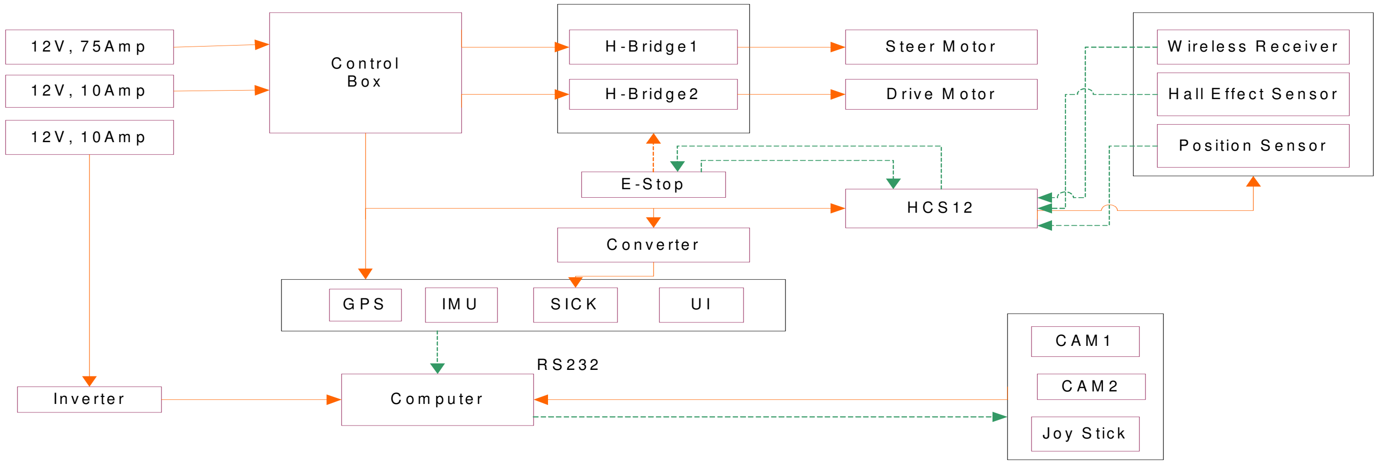


Figure 3.2.1a

3.3.2 Safety

The separation of the motors from the on-board electronics eliminates the chance of damage to the electronics from stray voltage spikes caused by the rapidly changing magnetic fields from the motors. Also, each component on the electronics circuit has its own switch on the master power box. This avoids possible damage caused by all components turning on at the same time. With these things considered, the chances of electrical system failure or overload are fairly low.

3.2.3 Scalability

If more sensors need to be added to the system, the power system could easily handle it. The microcontroller has many unused ports to handle I/O with more sensors, and the batteries have good enough current capacity to handle many more sensors as well.

3.3.4 Battery Life

The motors use the 75 Ah battery for themselves, and the on-board electronics will not subtract from the motor run time, which will allow us for about 90 minutes. The 10 Ah battery for the on-board electronics will last for about 3 hours of runtime.

3.3.5 Maintenance

If anything needs to be moved or replaced regarding the power system, it is fairly straightforward to make changes. Quick disconnecting power cables are implemented for all components, and each major component is packaged separately. This avoidance of spliced wires greatly simplifies the system, as well as provides flexibility. Spotting the cause of problems is easy, and fixing the problems is manageable.

3.3.6 E-Stop

The E-stop system is a relay circuit that breaks the connection of the PWM signals to the motors when either the physical switch on the vehicle is depressed or the wireless signal is telling the vehicle to stop. The advantage to this method is that the system is not dependent on software to stop the vehicle in an emergency, preventing faulty E-stop control in the case of severe software problems.

The physical E-stop switch is mounted on the top of the vehicle's frame, and is routed directly to the base of a BJT to control the current through two relays, one for each PWM signal. This is done in such a way that current flows through the relays and connects the PWM signals to the motors when the switch is not depressed. The wireless signal drives the base of another BJT, which controls the current through another pair of relays. The pairs of relays are put in series so the motors cannot move until both E-stop systems are disengaged.

3.4 Sensors Units

Sensor	**Description**
Web Cam - Logitech Quick Cam pro 5000 	This can find intangible obstacles, such as potholes or lines. It also views a large area.
LADAR – SICK LMS 200 	This is useful for locating physical obstacles, by angle and distance. Scans only as one arc.
GPS - Novatel Propak 4E 	Used to establish current location of the robot.
Compass – Honeywell HMR3200 	Used to establish the heading of the robot.
Sonar - Devantech SRF04 	This is a "rear view" sensor for Zeus. We already have information as to where we just were, and need to verify that nothing has moved behind us.

Table 3.4.a – Sensor descriptions

We use several sensors to allow the AI a chance to perceive Zeus's relationship with the world. Listed below in the table 3.4.a is a description of what each sensor does.

3.5 Software and Hardware Implementation

3.5.1 Software Strategy

One of our design challenges were the integration of the software. One of the previous teams had an extensive C++ code base, while another team had an extensive code base set up in MatLab and Simulink.

After much deliberation, it was decided to use a mix between the two code bases. C++ offers rich control over algorithms and data structures. MATLAB/Simulink offered the ability to quickly organize mathematical processes, and in an understandable model. By using the combination of the two, complementing each other, we were able to construct a more robust AI, with the overall design complexities laid out. Ultimately, we were able to turn one of our greater problems into a benefit for our team.

3.5.2 Hardware & Software Integration

The data flow between hardware and software is shown in figure 3.4.2a. The whole process of communication between hardware and software is discussed as follows:

Image data is grabbed from the webcams to detect lanes and obstacles. Lines are drawn corresponding to the detected lane lines and the endpoints of these virtual lines are calculated. The coordinates of the locations of any detected obstacles are also calculated. This raw data is then fed to the path-planning algorithm, which processes it and generates a series of waypoints for the robot to follow, such that the robot does not veer off course or run into anything. The waypoints are generated by creating a Bezier curve with some points from the processed data coming from the vision block. The waypoints are then converted to motor speed control signals in the vehicle control block. These control signals are then sent to the microcontroller, which converts them into the appropriate PWM

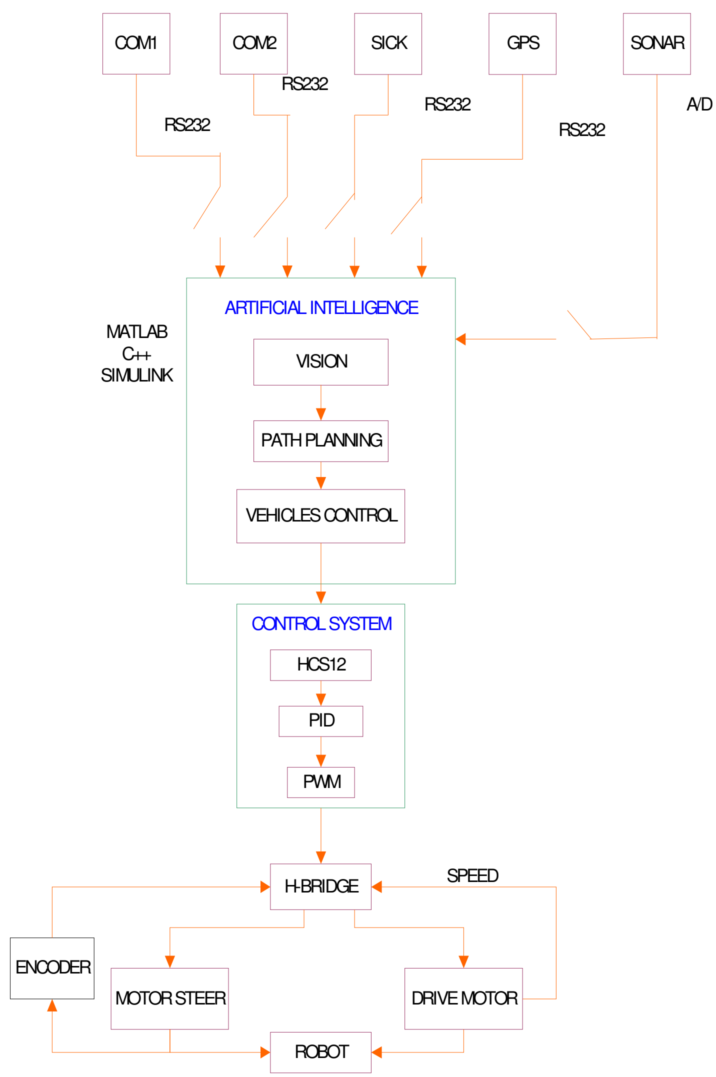


Figure 3.4.2a

signals using the PID controller implemented on it. The PWM signals are sent to the motors via H-Bridge to make the robot follow the desired path precisely with help encoder as feedback.

Sonar detectors are used to detect obstacles behind the robot. This way, it will still be able to avoid obstacles when moving backwards. Finally GPS is used for the path planning in the GPS challenge of the competition.

3.5.3 Computer

We used a laptop computer to run the Zeus AI. We use USB and RS232 to communicate with sensors. For the JAUS challenge, we utilize a built in 802.11g interface. There is a button on the laptop which can be used to enable and disable the radio, along with a LED that shows when the radio is active. This allows us to keep the radio on only for the JAUS challenge.

The built in battery has a run time of about 45 minutes. For extended runs, we use a 12 Volt lead acid battery in conjunction with an AC Inverter to provide extra power for long runs. We view this primarily as a facility in to assist us debugging.

3.6 Artificial Vision

3.6.1 Overview

Zeus uses two low cost web cameras in conjunction with variable polarizing filters to perceive its surroundings and navigate through the specified course. The vision algorithm is robust enough to detect not only the lanes but also it is designed to detect the obstacles that are in the course.

3.6.2 Lane Detection

The artificial vision software consists of two algorithms working simultaneously to detect lines. The first algorithm is based on a Hough Transform and the second algorithm is based on Pattern Recognition. The output lines from these two methods are fused using the Kalman filter. With some precisely calculated Fuzzy Logic rules, weights are determined to each algorithm. Depending upon the weights, Kalman Filter will output the estimated line coordinates.

3.6.3 Obstacle Detection

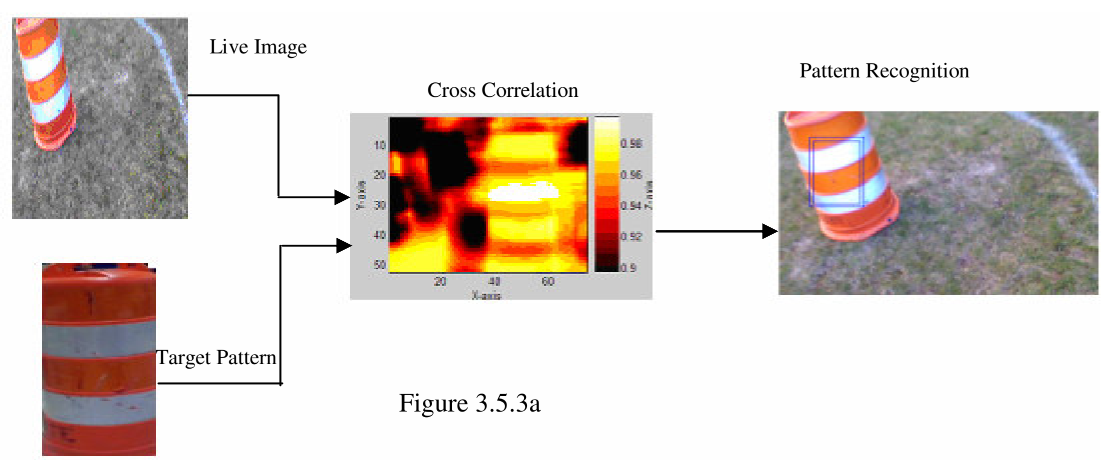


Figure 3.5.3a

This model uses the same Pattern Recognition algorithm that is discussed above to identify the locations of the obstacles. In order to recognize the obstacles such as barrels and pot- holes, their pictures are taken and provided for the algorithm to detect and locate its coordinates. The following example, figure 3.5.3a gives a better understanding of applying Pattern Recognition algorithm.

The following block diagram represents the Artificial Vision flow chart.

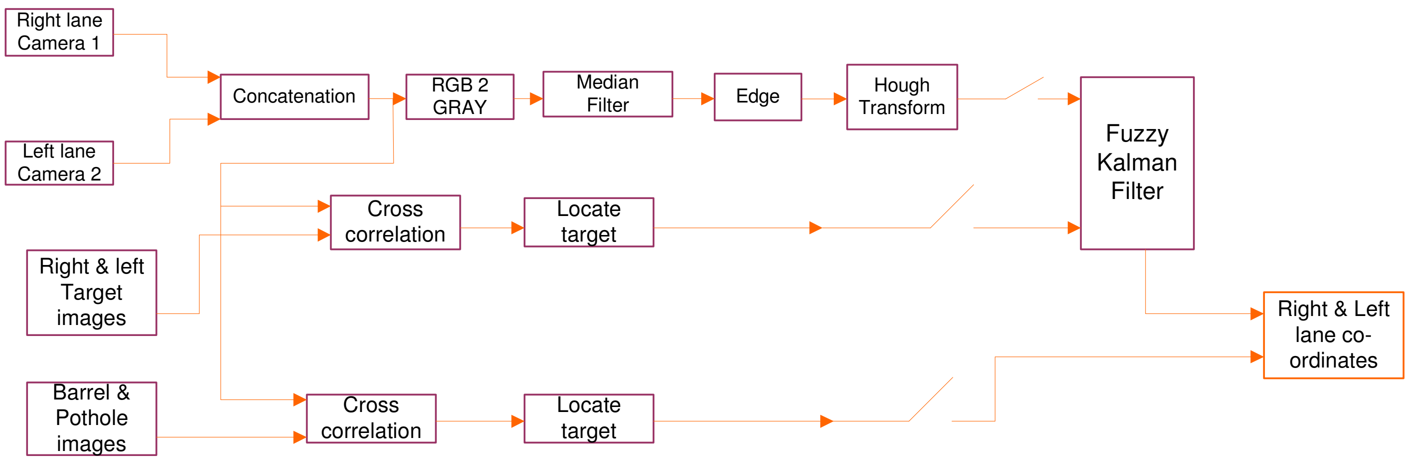
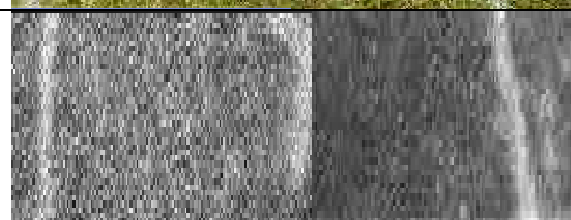
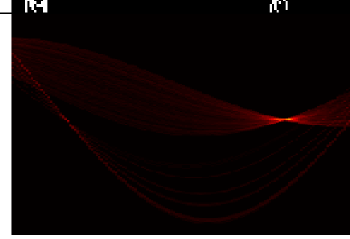
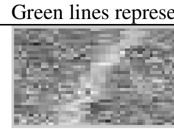


Figure 3.5.3b

Table 3.5.3c explains the purpose of each block in the Artificial Vision algorithm.

Block	Purpose	Output
Camera 1 & Camera 2	Live images are grabbed by camera 1 and camera 2 with 120X160 each.	 Left Lane Camera  Right Lane Camera
Concatenation	This block concatenates the images to form the resultant image of size 120X320.	
RGB to Gray	The image is then transformed from RGB to Grayscale.	
Median Filter	This Median filter block is used to filter out the salt and pepper noise associated with the input images.	
Edge	In this block edges are acquired for both lanes isolating from their background.	
Hough Transform	This well-known technique is applied to isolate features of the lines from the edges.	 Junction forms when the Hough Transform identifies the lines
Detect Lines	Once the target lines are identified, their locations are retrieved.	 Green lines represent the detected left and right lanes
Target Image	This block specifies the lane image to be identified for the Pattern Recognition algorithm.	

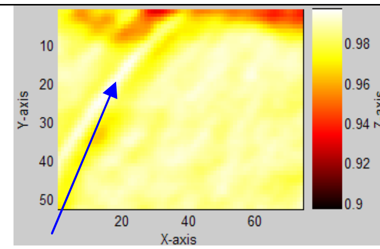
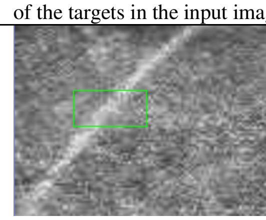
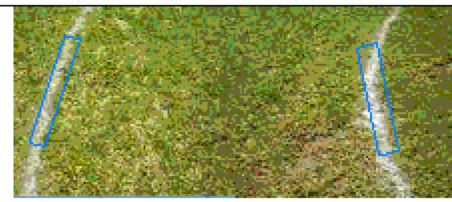
Cross Correlation	This method compares the target image and live image (from ‘left lane camera’) and outputs the correlated image.	 Large values in this window correspond to the locations of the targets in the input image.
Detect Lines	Once the target lines are identified, their locations are retrieved.	 Green rectangle represents detected left lane.
Fuzzy Logic Kalman Filter	This method estimates the locations of the right lane and the left lane based on the Hough Transform & Pattern Recognition methods. These methods are weighted with respect to the Fuzzy Logic rules that will allow Kalman Filter to select the algorithm that matches with the Fuzzy Logic rules.	 Blue rectangles represent the detected left and right lanes based on the Kalman Filter estimation.

Table 3.5.3c

3.6.4 World View

As lines and other obstacles are detected, they are registered with the world map data structure. Upon registering a scan, the world map check to see if the obstacle is already known, and if so, updates the known position with the added data. (This would be the case if two scans showed barrels with a center of 2 cm apart, for instance.)

3.7 Path Planning

3.7.1 Overview

The purpose of the Path Tracking module is to ensure that Zeus is following the designed path set forth by the path-planning module. The Path Tracking module will steer the Zeus so that it is not only in the proper position, but also in the proper orientation.

3.7.2 Path Tracking

Zeus path planning is based on the mid-line calculation relative to the right and left lanes. The heading angle of the robot is derived with respect to the midpoint of the net camera X-axis frame. From the figure 3.7.2a it is observed that from the LIDAR, obstacle information is acquired and mid-line is shifted relative to the obstacles. The safe distance of the robot to the obstacle is around 0.75mts to 1.5mts. For guiding the path of the robot, a 2^{nd} order Bezier trajectory is designed. Bezier curve in this application needs at least three points to generate the optimal path for the robot.

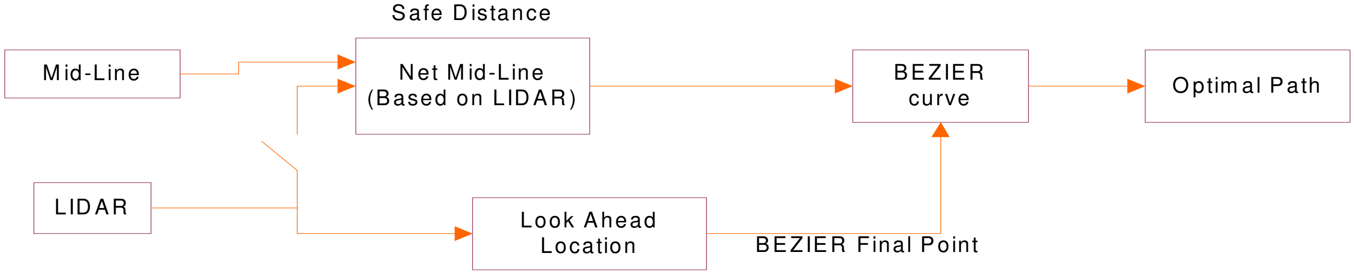


Figure 3.7.2a

Figure 3.7.2b shows the simulation of the 2^{nd} order Bezier curve with the blue as the Bezier trajectory and yellow and red representing the safe and critical distances and black representing obstacles.

Selecting the points for the Bezier curve is important for maneuvering the robot to an optimal path. Zeus, with the help of LIDAR looks ahead beyond its safe distance defined by the red and yellow squares shown in the picture to get the information of the obstacles that could occur in the near future. Based on this looking ahead point, Bezier curve is generated as shown in the figure 3.7.2c, where the blue point is the looking ahead point, which acts like the pivot. This concept helps the robot to avoid dead ends and traps, since the robot always tries to move as well as orient itself with respect looking ahead point.

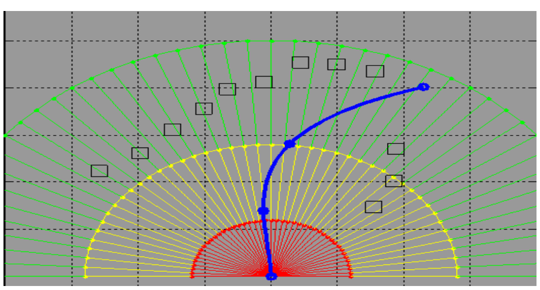


Figure 3.7.2b

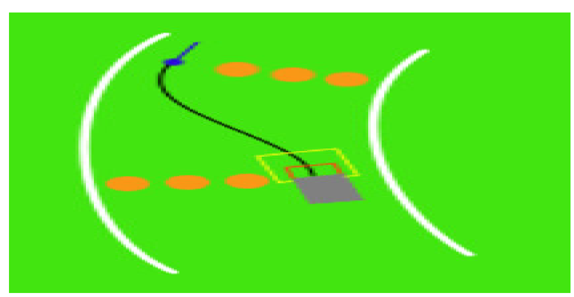


Figure 3.7.2c

Figure 3.7.2c depicts using a Bezier curve to calculate a path (black) to destination point and direction (blue). Barrels (orange) and lines (white) are shown for reference). The robot (gray) is shown, with safety zones (red and yellow). The path is calculated from the current location to the destination, using both points and first derivative (direction) as references. The tangent of the black line indicates the desired direction of the robot at the point the tangent is taken from along the path.

3.7.3 Proactive Path Planning

In order to avoid various forms of traps, we utilize the World View module to map out known obstacles. This allows us to run a deep search across the map's data structures, searching out possible paths, as a proactive method of finding direction. These deep searches are CPU intensive to solve, so this mechanism can only be used sparingly. By running periodic updates, we use the deep search algorithm to find long-term waypoints, and use quicker reactive routines to navigate towards the long-term waypoints. This allows us to identify traps such as switchbacks, dead ends, or other complicate structures while still operating responsively.

3.7.4 Vehicle Control System

Figure 3.7.4a shows a block diagram of the vehicle control system.

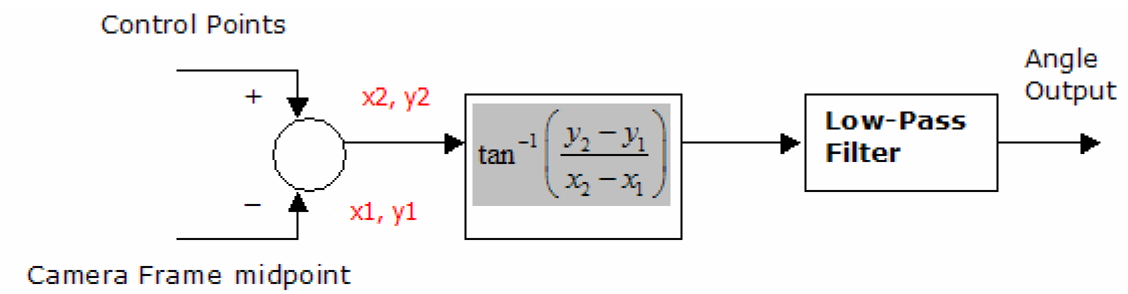


Figure 3.7.4a

This block inputs the coordinates of two control points, finds the angle of the line between them and outputs it as a control signal to the steering motor. The raw coordinates of the points are normalized by the camera frame midpoint, which is a constant. The inverse tangent is calculated from these points, and the low-pass filter gets rid of rapid fluctuations in the output signal due to rapidly changing inputs.

3.7.5 Speed Control

The speed of the robot is controlled with respect to the obstacle distance, which is provided by the LIDAR and the heading angle of the robot. The vehicle moves with its maximum velocity if it specified safe distance frame is free of obstacles and if the heading angle is in and around zero degrees.

3.8 JAUS

In preparing for the JAUS challenge, we reviewed documentation from the JAUS working group site to determine what we would need to implement. We produced several ideas on what software packages we could use to assist us in the implementation. Eventually, we decided upon using an Open Source library called net6, which is essentially a C++ wrapper to the Berkley Socket library. This allowed us to use network sockets in manners natural to C++, such using Resource Acquisition Is Initialization (RAII). We would them implement the subset of JAUS required for the competition from the documentation. We identified the definitions, headers, and messages that were of interest to us in the competition. From this, we were able to construct several modules with which we could use to incorporate JAUS into our AI control software.

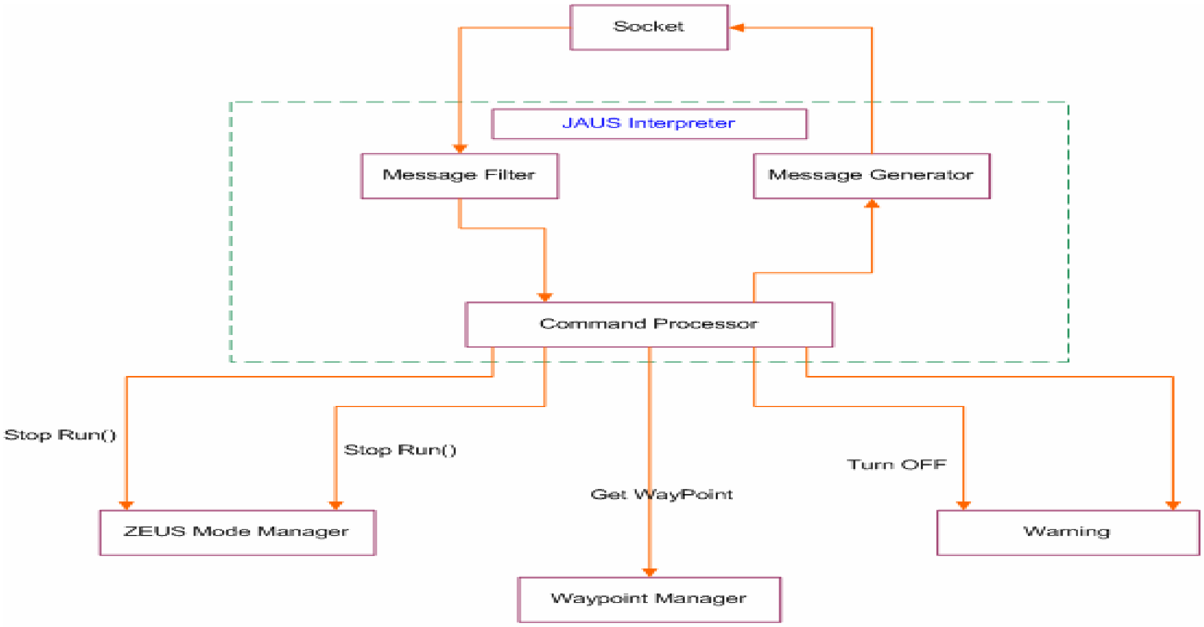


Figure 3.8.1a

From the figure 3.8.1a the center of the integration strategy is the JAUS manager class, which handles communication over interfaces to other classes, and contains several classes that handle internal operations. The other external classes that the JAUS manager class interfaces with are a network socket, the waypoint manager, the warning device controller, and the main Zeus mode manager. Internally, the JAUS manager has classes to handle message filtering, message generating, and command processing. The layout is depicted in figure 3.8.1a.

The message filter is configured with the address information for the Operator Control Unit (OCU), as well as Zeus's local address. The message filter uses this information so that it only accepts packets destined for Zeus, and also implements the required proper header formation checking, discarding malformed packets.

The message generator shares the same address information with the message filter. However, it is used to generate messages on demand and forward them to on the network (or Internet). It is called by the command processor, in response to messages sent to Zeus.

The command processor does the bulk of the work, which is to record a command and act upon it. For the JAUS challenge, we only have to handle a subset of the full JAUS command set, so we only focus on a handful of messages. There are three areas on which we focus, starting/stopping autonomous mode, enabling a warning device, and querying GPS waypoints. The command processor is capable of performing all of those functions, and when JAUS is enabled, it will perform them when the proper command is received.

3.9 GPS Navigation Challenge

Given that the location of the robot it is essential for the path planning to take place, the *Location Sensory* system has been designed to maximize accuracy. The GPS can give the robot a raw WAAS-corrected signal which in practice gives us an accuracy on where the robot is in the world by roughly 180 cm. By combining this information, the direction the robot is heading via the digital compass, and the odometer readings from the wheel sensors into the Extended Kalman filter, an accuracy of approximately 20 centimeters has been tested – an order or magnitude more accurate than simply using a GPS system alone.

The control scheme for the GPS navigation is with the block diagram below. Current position of the robot is acquired from the Novital GPS receiver. The noisy latitudes and longitudes are passed through the Extended Kalman Filter to provide better estimates of the position. From the JAUS interpreter module, target waypoints are acquired and robot heading is calculated based on the current position and target position. Camera, LIDAR, IMU and Compass are integrated to provide an optimal and safe path for navigation.

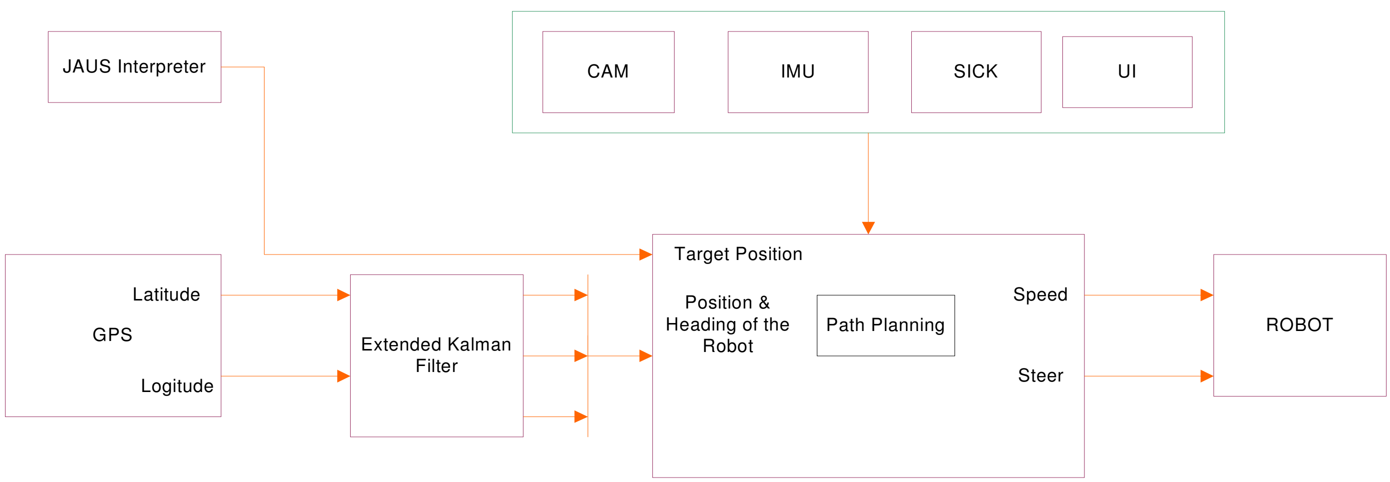


Figure 3.9.1a

3.10 Safety - AI Level

It is always important to keep track of the Artificial Intelligence performance and status, by observing the statistical behavior of the decision-making algorithms. If the decision-making process is chaotic, the AI safety algorithm will terminate the entire software process to avoid un-wanted events.

3.11 User Interface Panel

Zeus features a prominent user interface panel. An aluminum box is fixed to the top of the vehicle at the rear. This is painted safety yellow to make it more visible. This control box features controls to set modes and start runs, as well as hosting the pushbutton E-Stop. The box also features a master power switch for the onboard electronics package. The box houses a Xilinx Spartan 3 Evaluation board. This board connects with the onboard to send/receive control signals. Control signals sent/received by the Spartan 3 Board are in the form of a single ASCII character from 0 to 4. An RS232 interface is used to connect the control box to the onboard computer. All switches and indicators are waterproof. A splash proof USB connector is also provided to allow for joystick control of the vehicle while in Manual Mode.

4. Budget

Item	Quantity	Unit Price	Total Price
SICK LADAR	1	$4,000.00	$4,000.00
Novatel GPS	1	$3,000.00	$3,000.00
Digital Compass	1	$175.00	$175.00
Laptop	1	$900.00	$900.00
MiniDragon HCS12 Microcontroller	1	$85.00	$85.00
Spartan III	1	$99.00	$99.00
Extruded Aluminum	N/A	N/A	$200.00
Aluminum Sheet	N/A	N/A	$220.00
12 V, 75 Ah Marine Battery	1	$251.00	$251.00
12V, 10 Ah Battery	1	$43.00	$43.00
Drive Motor	1	$155.00	$155.00
Steering Motor	1	$35.00	$35.00
Webcams	2	$65.00	$130.00
Victor Motor Controller	2	$115.00	$230.00
Grand Total:			$9,523

5. Conclusion

This year we have undergone a huge restructuring, merging three teams into one single team. We have worked hard to find the best solutions from the independent work that we had previously undergone, and have came up with some creative combinations of ideas. We are looking forward to the competition so we can see how far our teamwork has taken us, and are hope to see in next years design both improvements on our weaknesses and a continued stream of creativity towards recognizing new possible solutions.

Although the team members have worked on this project on a volunteer basis outside of class, we have found it to be an invaluable part of our education, and in just that alone we know success.