RHINO

Intelligent Ground Vehicle Competition 2009

Team members: Allen Akroush, Aamer Ahmad, Derek Bray, Lawrance Chan, Fadie Ghraib, Josiah Hudsun, Rick Johnson, Mona Kadouh, Gierad Laput, Raymond Llonillo, Anthony Lucente, Lucian Popa, Kirit Patel, Stephen Spuehler, Ziad Yousif

Faculty Advisor: Dr. Narasimhamurthi Natarajan

<u>Outline:</u>

1. **<u>Introduction:</u>**

Rhino is a new robot designed by *The University of Michigan- Dearborn Intelligent Systems Club*. It is an intelligent, unmanned, and autonomous ground vehicle designed to participate in, and meet the requirements of the Intelligent Ground Vehicle Competition. In our report, we will discuss the layout of Rhino's platform highlighting the innovations throughout our design process.

2. **<u>Planning:</u>**

A. Team Formation:

Rhino is the third robot that will represent the University. The decision to enter the third robot was largely influenced by two factors. The more experienced students who had participated in earlier competitions were active in improving robots from the last year and also, they will all be graduating at the end of this term. Consequently it was decided to form a third team consisting of novices, most of whom are sophomores and juniors who have not had any prior experience. It was felt that this would be a valuable learning experience and will provide experienced members for the next year. Thus, an initial team was formed to discuss basic strategies.

B. Design Planning Process:

Given the inexperience of the team members, it was decided to keep the design simple. Also, as many of the team members have not had any advanced courses in engineering, the design had to be simple enough so that *all* the team members could understand and participate fully.

3. **<u>Hardware:</u>**

A. Chassis:

Rhino has an all-new chassis for the 2009 competition. The design process began upon the acquisition of an old wheelchair. It was disassembled and the core frame that held the front casters and motors was retained. A battery holder was fabricated and bolted to the frame as low to the ground as possible to lower the center of gravity. A large box provides a sheltered compartment to house the electrical components. Moreover, removable shelf inside the box provides a place to mount the components such that removing the shelf will provide the necessary accessibility to service the robot. Also, a door at the front of the robot provides additional accessibility. In addition, a camera mount using large diameter steel pipe is located at the front of the robot and holds a video camera that will be used for image processing. The result is an optimized agile robot platform with a zero degree turning radius.

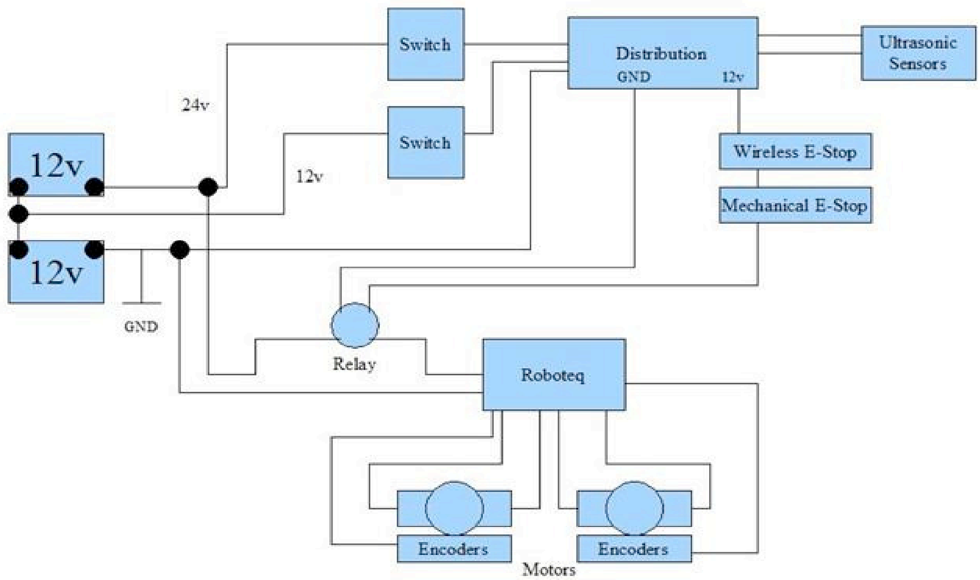
B. Electronics:

I. Computer System:

A Compaq Presario CQ60-206US notebook PC running a dual core Athlon X2 processor is the central processing unit of Rhino. Based on budget constraints, we bought a low budget computer costing $400. Due to the fact that MATLAB is processor-intensive, our main focus was the speed of the processor and battery life. Since the camera did not have drivers for Windows Vista, Windows XP was chosen to be the operating system.

II. Power Distribution:

Two 12 volt 75 amp-hour marine deep cycle batteries are connected in series to provide 24 volt DC power to all robot subsystems. All wires use Anderson connectors with adequate current rate. The Anderson connectors provide easy connect/disconnect capability. The robot is activated by switching on power to the wireless E-Stop by pressing the one-channel that is connected to the main robot solenoid. The solenoid, on the other hand, powers the robots Roboteq AX2850 motor controller, activating the motors on the robot. There is a power distribution box that is powered on by two switches on the side of the robot. The power distribution box supplies both 12volts and 24 volts to all the systems inside the robot. The robots main batteries are charged by a competition grade 2-bank battery charger.
The following is the power distribution panel diagram:



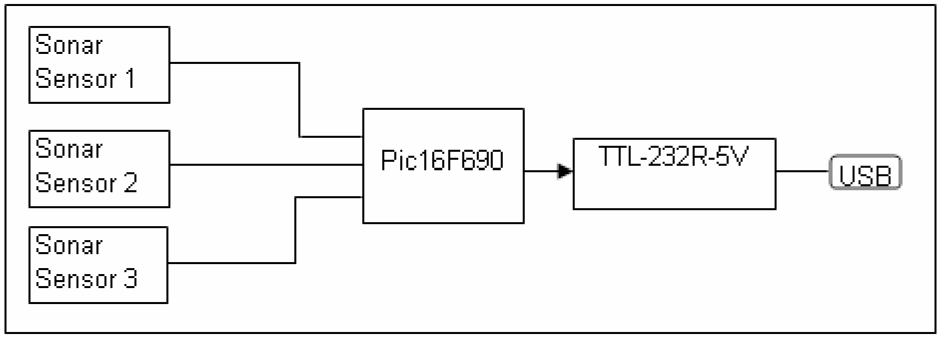
III. Motors:

Motors run at approximately 140 RPM and have a 12.5- inch wheel connected to them, allowing Rhino to achieve the IGVC's 5 MPH speed limit. Both right and left motors are built to be opposite to each other so that one motor can be mounted on the left side and the other can be mounted on the right side. The RPM is about 130 RPM with a load of 60 in-lb at 7.9 amps.

IV. Obstacle Detection:

Detecting obstacles such as barrels is performed by inexpensive sonar sensors from Maxbotics. The sensors are monitored by a PIC microcontroller. Rhino has five such sensors, three looking in front and two to the sides to look for openings in fences (navigation challenge) and to navigate out of traps and switchbacks (autonomous challenge). The sensors measures the distance to the nearest obstacle. The Microprocessor polls the sensors and transmits the measured distances to the main computer. The decision to use this design was motivated by the fact that this topic is covered in one of the senior level courses and one of the team members was familiar with interfacing to the sensors and inter- process communications. In order to detect obstacles on the course the robot was fitted with a sonar system. This subsystem consists of a number of ultrasonic rangefinders, connected to a Pic16F690 microcontroller that sends the data to the computer via USB.

The following is the block diagram for proximity sensors subsystem:



II. **Software:**

A. Programming Environment:

It was decided to use MATLAB as the programming environment. Although this meant that there could be performance degradation, all the members of the team have a good working knowledge of MATLAB and can participate in and learn from this project.

B. Signal processing:

I. Vision capture:

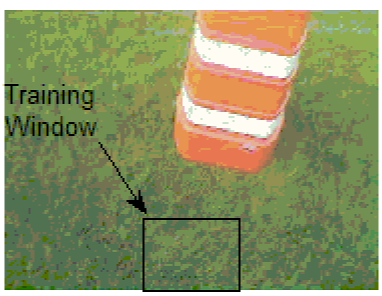
For Vision processing it was decided to use a Sony Handycam as it was readly available. Image capture is performed using a DLL (Vcapg2) written by Kazuyuki Kobayashi of Hosei University.

II. Image processing:

A very simple image segmentation approach was taken for detecting lanes and obstacles. A small patch of each image directly in front of the robot was considered “safe”. Each pixel in the image was then classified as being either safe or unsafe depending on how similar the color at that pixel is compared to the pixels in the safe region. Similarity was calculated using the standard Mahalonobis distance measure. After the classification was completed, the program checks if the path in front of the robot is safe or not. If it is not safe it navigates around the unsafe region by comparing how unsafe the left and the right sides are and decides to turn left or right based on this. Points closer to the robot are given greater weight than points further away. This design approach has several limitations some of which will be addressed during testing:

- If the robot gets too close to an obstacle, then it may misclassify the obstacle as safe. However, this should never happen if our code works as designed. Also, the robot will have proximity sensors that will prevent this from happening in the case of solid obstacles.
- The robot could try to go through a gap in broken lane markings. At present, there is no well defined strategy to prevent this. However, a simple Hough transform technique will be used to fill the gaps in the lane. This module is currently not been fully tested.

The following picture shows how the image processing finds a safe path through the obstacles:



III. Global Positioning System:

The main GPS unit used is an inexpensive Wintec G-Rays I unit capable of up to 10Hz refresh rate. However, the accuracy is comparable to other such inexpensive units. Standard curve-fitting technique taught in basic calculus is used to improve the accuracy of the GPS measurements. While this approach does not have the full power of a Kalman filter, it has the advantage of being understood by undergraduate students. A second GPS unit will be used to provide heading information and to improve the GPS accuracy. The GPS unit is attached to the PC. The GPS unit sends position and heading data to the PC at a rate of 10Hz in the NMEA message format (http://www.gpsinformation.org/dale/nmea.html). The messages are essentially several ASCII coded string fields, comma separated, with a trailing checksum. Messages are terminated with a CR/LF pair. The messages are parsed and processed in MATLAB to determine the robot's position and heading. Since the messages are not synchronous to MATLAB, we use MATLAB's callback feature to capture the data sent by the GPS unit. The callback feature in MATLAB in essence designates a function that is to be called whenever a given event occurs. In our case, we designate that the function 'GPS_callback_handler' be called whenever the CR/LF pair is received. In MATLAB code, the current latitude, longitude, distance to target point and heading toward target point of the robot are kept in global variables. This allows the values to be read by sections of code other than the callback. Whenever a CR/LF combination is received, the callback function is called and it reads the sentence out of the input buffer, examines it, and, if the message is legal and valid, it updates these global variables. This update occurs automatically and independent of whatever other code MATLAB is running at the time the CR/LF terminator is received. The function 'GPS_callback_handler' reads the NMEA sentence from the input buffer, and does the following:

- Examines the checksum of the newly received message. If the checksum test fails, the message is ignored and the global variables are not updated.
- Checks to see that the message is a 'GPRMC' message. (The GPS sender sends several messages, each with different information. The information needed to update the global variables is in this 'GPRMC' message) If it is not, the message is ignored and the global variables are not updated.
- Reads the fields in the message that indicate current latitude and longitude of the robot.
- From these fields, MATLAB calculates distance and heading to target and updates those global variables.

4. Safety:

A. Hard E-stop:

The manual emergency stop consists of red push button to stop the vehicle immediately. Pushing the stop button cuts off the power to the motor and locks the wheels by turning on the electronic brakes, bringing the vehicle to an immediate stop.

B. Wireless E-stop:

A wireless remote keyless entry unit with a range of 650ft was modified to stop the vehicle remotely. When the remote button is pressed, power is cut to the motor, stopping the motor immediately.

5. Performance:

Attribute	**Design Prediction**
Maximum Speed	5.0 mph
Climbing Ability	30 degree ramp
Nominal Power Consumption (Watts = Amps x Volts)	500 watts
Battery Operating Time (24v 75AH Battery System)	4 hours
Distances at which objects can be detected	20 ft
Waypoint accuracy	3 meters
Reaction Times	15 Hz
How vehicle deals with complex obstacles	Reactive Fuzzy Logic approach

6. Cost Analysis:

Part	**Cost**
Chassis Parts	$100.00
Motor Controller	$620.00
Laptop	$549.00
Ultrasonic Sensors	$139.75

Microcontroller and Programmer	$49.99
Camcorder	$0.00 (Donated)
Two Bank Battery Charger	$189.99
Wheel Chair	$0.00 (donated)
Wheel Encoders	$239.00
Electronic Connectors	$130.00
GPS	$49.99
Batteries	$130.00
Wireless E-stop	$49.99
Hard E-stop	$38.00
6% Sales Tax	**$ 137.08**
Total	**$ 2421.80**

7. Conclusion:

The fact that this year's team was made up of students from different engineering backgrounds (Electrical, Mechanical, Computer and Software Engineering) solidified our steps and sharpened our determination. Implementation and design were performed with fortitude and excitement. Although Rhino is not as advanced as Wolf and Raptor (the other two robots), it has provided the team members with a cutting-edge learning opportunity.

8. Acknowledgements:

The team would like to thank Dr. Narasimhurthi Natarajan for his assistance throughout Rhino's implementation process. The team would also like to thank Prof. Kazuyuki Kobayashi of Hosei university for the image capturing program.

9. Signature:

I certify that the design and creation of Raptor has been significant and is equivalent to what might be awarded credit in a senior design course.

Dr. Narasimhurthi Natarajan
Professor of the Department of Electrical and Computer Engineering, University of Michigan – Dearborn

Date: