

UMD Raptor

University of Michigan-Dearborn

Anthony Lucente, Jonathan Hyland, Michael Kinnell, Siby Varughese



Department of Electrical and Computer Engineering
University of Michigan-Dearborn
June 10-12th, 2006

Introduction

This document describes the design of RAPTOR, an autonomous ground vehicle intended for participation at the 14th Annual Intelligent Ground Vehicle Challenge. This vehicle was designed and built from scratch by the Intelligent Systems Club, a student organization at the University of Michigan-Dearborn. The design requirements and the budget for the robot were given to the team and the various members participated in the design and construction of the vehicle.

Requirements

The vehicle had to be designed to meet the following:

1. Take part in the Autonomous Challenge
2. Take part in the Navigation Challenge
3. Total budget under \$6000
4. Design and built in less than 4 months using all-voluntary participation

Design Team

The design team for the project consists of Anthony Lucente, the President of the Club, Jonathan Hyland, the Treasurer of the Club, Shibu Varghese and Mike Kinnel, two additional members of the team. Of the four, all except Varghese took part in the 2005 IGVC and were instrumental in making the key decisions based on the previous year's experience.

Team Organization

Name	Responsibility	Class Standing	Hour Spent
Jonathan Hyland	System Design, Software Design, Vision System, Chassis Fabrication, Motor Control, GPS Software Design, Electrical System, Software Integration	Undergraduate, EE/CE	375
Anthony Lucente	System Design, Chassis Fabrication, Motor Control, Electrical System	Undergraduate, EE/CE	375
Michael Kinnel	System Design, Electrical System, Software Integration	Undergraduate, EE/CE	100
Sibu Varughese	Chassis Fabrication, Electrical System	Undergraduate, EE/CE	50

Design Decisions

Past Experience

Since three of the team members had participated in the last year's challenge, one of the first steps taken by the team was to go over the problems they faced and the lessons learnt.

1. Speed control: Last year's design had a very aggressive speed control strategy. As a result, during one of the test runs, the vehicle when coming down the ramp came down too fast and the rapid deceleration that was induced by the speed control algorithm caused the vehicle to flip over.
2. Communication with peripheral devices: All the communications used RS232 serial protocol. However, the primary control computer was a laptop which had no RS232 serial ports and the communications required USB to serial adapters. This resulted in several system crashes and an unreliable performance.
3. Control protocol: The control protocol between the main computer and the motor controller (an auxiliary microprocessor) consisted of sending the steering

command, with the microprocessor controlling the speed to a preset desired value.

This in turn limited the controllability of the vehicle.

4. Vision System: The vision system consisted of a Logitech web cam. The quality of the resulting image as well as the dynamic range of the camera limited its usefulness in varying lighting conditions. This resulted in unpredictable behavior in the presence of shadows.
5. The GPS: Last year's design used GPS to get the heading/orientation of the vehicle. This meant that the vehicle had to move some distance before accurate heading information was available. This resulted in a very poor performance.

Design Choices

1. Choice of Sensors

The first decision to be made was the choice of sensors. Based on prior experience, budget, and time constraints, it was decided that the vision system be used and avoid using proximity/range sensors such as the SICK (?) laser range finder. Instead of using a web cam with custom rigged wide-angle lens attachment, a mini-DV camera was used with automatic AGC to compensate for the changing lighting conditions and a fire-wire digital interface to the PC. For the Navigation Challenge, the team decided to use a digital compass in addition to the GPS unit to obtain precise heading information.

2. Onboard Computer

It was decided to use a desktop computer instead of a laptop to avoid using USB to serial converters. Instead, it was decided to build a desktop computer with standard RS232 serial ports. A secondary, but equally important reason for not using a laptop is that laptops tend to be removed and used for other projects.

3. Motor Control

Last year's vehicle had a simple power electronics circuit designed and built by the team. The circuit had several serious limitations. It was not possible to reverse the polarity of the motors and the only way to slow the vehicle down was to use uncontrolled braking. Based on the experience mentioned earlier, it was decided to incorporate four quadrant controls using the standard H-Bridge configuration. It was decided to use the open source design for the H-Bridge to be controlled by a custom built microprocessor system. The OOPIC development system was utilized because of the familiarity with the system. An alternate approach using a PIC micro was attempted but not pursued fully due to time and manpower limitations.

4. Chassis

The choice was between using the chassis from last year and building a new one. An attempt was made to use the chassis from last year; however it was found that the space was not sufficient to meet all our needs. Hence, it was decided that a custom design be planned for the chassis. To accommodate all the additional components, the size of the vehicle had to be increased substantially from last year's vehicle. This also meant that we could no longer use casters because of the increased size and had to use skid-steering. Skid-steering requires considerably more torque when turning and the vehicle had to meet certain geometric conditions. Based on this, it was decided to use six wheels, three on each side. In a very early design, it was decided that six wheel chair motors were to be used to drive the vehicle. The overall dimension of the vehicle was dictated by the constraints imposed by mounting three motors on each side. A CAD drawing of the

vehicle was created with wheel to wheel clearance of 1 cm. Since the vehicle would be hand built, it was decided not to reduce the tolerances below the 1 cm.

Embedded Software

The embedded software had one main goal, to take data strings in and translate those strings into real world events. This goal was looked at further and then divided into five models that were given to new students to get them started. The Five models were:

1. H-bridge speed and direction control
2. Speed feedback control and correction
3. Compass oration
4. String reader and translator
5. Main linker

Image Processing

Images are acquired for a Cannon digital video camcorder by using Matlab's Image Acquisition toolbox. These images are then processed in Matlab to determine the relative orientation of the lanes in the course, as well as the location of all obstacles. The controls algorithm uses this information to plan the route though the course. The image-processing hardware is composed of an AMD64 PC and a digital camcorder. The camera is mounted in a way that allows for height and tilt adjustments to maximize coverage of the vehicles image plan. The image processing software module can be broken down into two sub modules lane finder and obstacle detection.

Lane Detection

The lane finder has the purpose of estimating vehicle orientation in the course with respect to the heading direction of the vehicle, as well as the offset between the lane center and the center point of the vehicle's wheelbase. The first step in this process is to acquire the image from the camera, which is done through the Image Acquisition

toolbox. Next, the image is applied to a filter with products intensity image. This is done to decrease processing time. Finally, the image is passed through the Hough Transform, which will detect the lines. For line detection, Hough Transform is the best way to go because it is not affected by noise or lighting conditions. This information is then passed to the control module which determines the best direction for the vehicle to move towards.

Obstacle Detection

The second sub module in the image-processing unit is to detect obstacles and hazards that may obstruct RAPTOR path. Obstacle detection is done through the digital camcorder. Once the lane markings are detected, the vehicle's path will be delimited within their boundaries. Thus, obstacle detection is performed only within a Region of Interest (ROI), delimited by the lanes and extending up to a critical distance, (approximately 10 feet). Since all obstacles are to be avoided, identification of the obstacles is not needed; only their detection is important. Obstacle detection uses intensity and color information of all pixels within the ROI to plan the path of the vehicle. The algorithm used is as follows:

1. A small region of the image immediately in front of the vehicle is chosen as a representative of an obstacle free region.
2. The color image in this region is extracted and the covariance of the color information is obtained.
3. An approximation to the eigenvector corresponding to the smallest Eigen value of the covariance (least dispersion measure) is computed.
4. This eigenvector is used as weights to be used to convert the color image to modified intensity image.

By our choice of weights, the intensity of the part of the image that corresponds to the chosen region would be nearly constant while parts of the image that are different would have intensity different from that of the region. If the intensity at any part of the image differs from that of the chosen region by more than three standard deviations of the

intensity in the region (this value is easily computed from the covariance matrix), then it is assumed that the difference is due to obstacles. Obstacles are typically detected when they are 10 feet from the vehicle by the vision system.

Direction Control

The lateral control of the vehicle is governed by the location of the lanes and the location of the obstacles. First a heading direction is determined based on the lanes that are detected. The control is based on a target point determined by where the vehicle should be at a look-ahead distance D as shown in figure 4. Based on straightforward calculations, it can be shown that a smooth control can be obtained if the input is obtained by multiplying f , the offset of the desired trajectory at the look-ahead distance D , by a gain. The vision system first computes the offset f using the lanes. If there are any obstacles in front of the vehicle, they act as a repulsive force on the target point and moving it laterally. The magnitude of the repulsive force is made inversely proportional to the distance of the obstacle from the car, and its distance from the desired direction. The image processing program also provides a visual display of the lanes and the target point.

Navigation

Navigation data is acquired for a Garmin 16a GPS and compass. The software Module for Navigation control interfaces through a serial port (RS232) with Matlab. Matlab decodes the GPS signal and applies a Kalman filter to improve accuracy. The compass is then used to determine the heading of the vehicle. The obstacle detection and avoidance system uses the same image processing software as stated above. The data obtained from GPS unit is used to send steering control signals to the main microprocessor to navigate from one waypoint to other. When an obstacle is detected, the obstacle avoidance system will take over the control until the obstacle is avoided.

Safety Systems

Manual emergency stop

The manual emergency stop consists of a red push button to stop the vehicle immediately. Pushing the stop button cuts off the power to the motors, which in turn brings the vehicle to an immediate stop.

Wireless emergency stop

A wireless remote keyless entry unit with a range of 100ft (30 meters) was modified to stop the vehicle remotely. When the remote button is pressed, it cuts off the power to the motor, which stops the motor immediately.

Description	Component Cost	Sub-Category Total
<u>Chassis Construction-Mechanical Design</u> Square Metal Tubing and Sheet Metal Wheels: (6)-12" Wheels Motors: (6) 24V 115rpm DC Wheelchair Gear Motors	450.00 72.00 600.00	1122.00
<u>Motor Control Circuitry</u> (2) 400 AMP H-Bridge	2 x 340.00	680.00
<u>Robot Computer</u> AMD Athlon 3200 Antec ARIA Aluminum MicroATX Cube Computer 80 GIG Hardrive 802.11g Wireless Card NEC 15" Touch Screen	554.82 45.99 23.99 209.00	833.80

<u>Motor Microprocessor</u> OOPIC Controller	79.00	79.00
<u>GPS</u> -Gamin 16A 5HZ	300.00	300.00
<u>Camera</u> Cannon ZR500	250.00	250.00
<u>R/C Transmitter and Receiver</u> Laser 6 75MHz System: R/C Brain:	138.95 123.00	261.95
<u>Accessories</u> -Wireless Emergency Stop -Manual Emergency Stop -(4) Deep Cycle Batteries 12V	89.00 80.00 600.00	769.00
	Total: \$	4295.75

Acknowledgments

The team would like to thank John Lucente for his help and support for the project.

Faculty Advisor Certification

I hereby certify that the engineering design in the vehicle by the current student team has been significant and equivalent to what might be awarded credit in a senior design course.

N. Natarajan
Associate Professor,
Department of Electrical and Computer Engineering
University of Michigan-Dearborn