Road Runner

University of Michigan-Dearborn

Anthony Lucente, Ross Marten, Jonathan Hyland, Hachem Nader, Ebrahim Nasser,

Vin Varghese

Department of Electrical and Computer Engineering

University of Michigan-Dearborn

May 27th, 2005

Introduction

Road Runner is an autonomous vehicle that was designed to participate in the 13th Annual IGVC of 2005 in the following events: the Autonomous Challenge Competition, Navigation Challenge Competition. In Autonomous Challenge Competition, the vehicle has to complete a course marked by two lane markers and on the course will be obstacles such as construction barrels as well as potholes (simulated using white paint). Also the terrain consists of grass, sand, and wooden ramps. In the *Navigation Challenge Competition* the vehicle is expected to autonomously visit waypoints in a parking lot specified by their latitude and longitude while avoiding obstacles. The design team consists of students from the department of Electrical and Computer Engineering at University of Michigan-Dearborn. All members are part of University of Michigan-Dearborn's Robotic Club-Intelligent Systems.

A Matlab interface using vcapg (developed by, Kazuyuki Kobayashi[1]) was developed for performing real time video processing using a webcam and a wide-angle lens. This report describes the steps we took, the testing we performed and the capabilities of the resulting systems.

Team Organization

Each member of this design team was assigned both a primary and secondary task to complete on the vehicle. However, our team frequently worked together when we encountered serious problems. As a result, the entire team, rather than just a certain individual made all critical design decisions.

Design Process

Design and construction of the vehicles were an incremental process. After each step, the design was tested and we made sure that the system performed as required. We often had to develop new techniques, such as internet based controlling of the vehicle, so we could test each module. The following describes the steps we took to complete the design:

[1] http://www.ikko.k.hosei.ac.jp/~matlab/matkatuyo/vcapg2.htm

1. Starting point for our design

Chassis: For Road Runner, our team build a chassis capable of differential steering. That gave the robot small turning radius, which allows the vehicle to easily maneuver in extremely small spaces. Two wheel chair motors were used for the drive and they are controlled independently. The chassis design was done first on CAD software and then buillt according to specification.

Vision system: A Logitech Pro 4000 webcam and a wide angle lens to process images. A Matlab interface was developed to grab images and process lane information. The control data was sent to the microprocessor using the serial port.

Processors: An OOPic microprocessor was used to do motor control. Another OOPic controller processed sensor data and send the information back to the PC through the serial port..

2. Understanding the vehicle

Road Runner: To understand the power requirements of the vehicle and to determine the strategy to control the vehicle, we replaced the joystick that came with the wheel chair with electronic circuits that would simulate the movement of the joystick. The 12V DC motor drew about 3 A with regular load and a max of 10 A when stalled. A 30 A fuse was put in series with the power source to prevent any over drive damage to the driver FET’s and the motor.

3. Special Sensors and electronics

Road Runner has a speed sensor to sense the speed of the vehicle and to maintain that speed. An RC car was used as a fifth wheel. The voltage generated by the motor on the RC car was used similar to a tachometer, and the OOPic microcontroller kept in check of the speed by regulating the speed when the voltage increased or decreased.

4. Designing the control circuit

Road Runner: We next replaced the original controller with the simulated joystick by a power electronics circuit that can generate pulse width modulated voltages under the control of a microprocessor. This circuit has an optical isolator to protect the microprocessor from high power motor drive circuits as well as comparators to increase the noise immunity. We selected components to handle up to 30 amps, although the wheel chair motors typically draw 3 to 5 amps and about 10 amps when climbing the ramp. We feel that this is a sufficient safety margin. An H-Bridge circuit was designed using two relays that were controlled by the OOPic microprocessor. The H-Bridge helped to improve the turning radius and reverse capability.

We tested the electronics manually by entering the desired turning value in the microprocessor and verifying that the vehicle turned as desired. We next tested the communication between the main PC and the microprocessor by manually entering the desired turning angle in the PC and then transmitting to the microprocessor using the serial port. Once we were satisfied that the electronics and the control algorithms were correct, we obtained a joystick that can be connected to an offvehicle computer. We designed software so this computer read the joystick and transmitted the information using the Internet to the computer on the vehicle. At this stage we could remotely control the direction of the vehicle and we were able to calibrate the steering system of the vehicle as well as test our ability to control the speed of the vehicle.

5. Emergency Stop circuit

Before testing the vehicle further, to ensure the safety, manual emergency stop button was added to stop the vehicle in an emergency.

6. Speed Control

Once we were able to run the vehicle at a constant speed and control the turning angle, we added additional features to the control algorithms and the electronics. We first designed a sensor circuit to monitor the power to the motor. The control algorithm was modified so that the microprocessor would wait for the power to be turned on and then gradually increase the speed to the desired value, thus eliminating jerks when starting. We designed a speed sensor using an RC car that would send a voltage back to the microcontroller which kept in check of the speed which was proportional to the voltage send back.

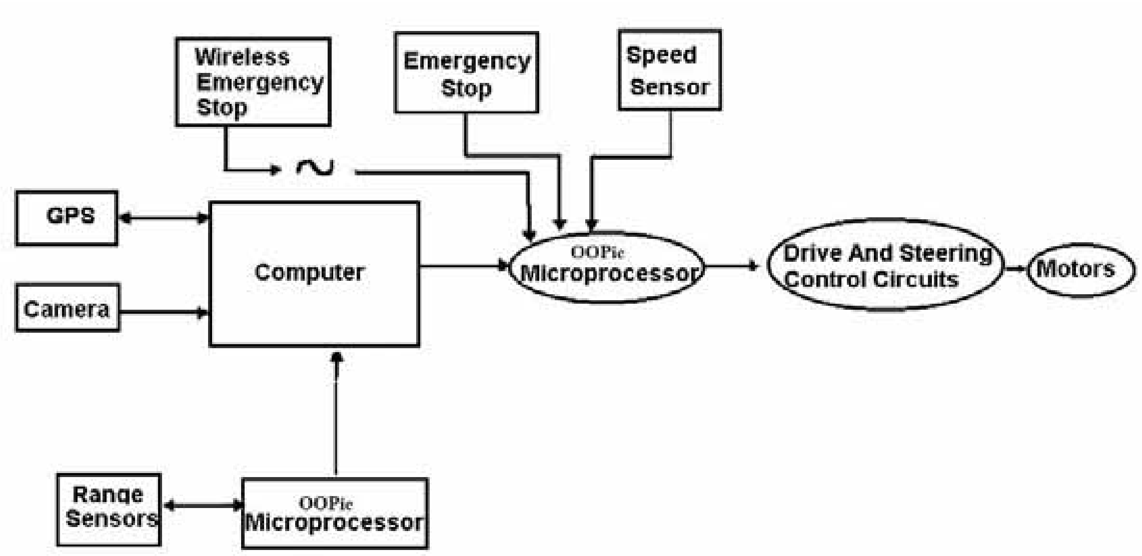
7. Obstacle Detection

Once we were able to run the car with the speed sensor, an ultrasonic sensor (SRF08) was added to detect obstacles. This sensor works on the I2C bus on the OOPic microcontroller. We interfaced the sensor to an OOpic microcontroller and the data was send back to the PC through the serial port. The PC made control decisions and sent control data accordingly to the motor controller microprocessor through the serial port. The control algorithm was modified to stop the vehicle if the obstacle was too close, or else take evasive action for obstacles that are not too close. Once testing was done using one sensor, we added two more sensors to increase the field of view to 180 degrees

8. Lane following and Obstacle detection and avoidance

A camera Logitech Pro 4000 webcam and a wide-angle lens is used to detect the lane markers and obstacles. The control of the vehicle is performed in two steps. First a small region is selected in front of the vehicle. This region is treated as prototypical of the pathway. Any region whose color is statistically different from this region is considered an obstacle (lane markers, potholes etc.) A desired direction is then chosen to avoid these obstacles, with obstacles closer to the vehicle receiving higher precedence.

System Block diagram



Software strategy Autonomous challenge:

PC control software consists of Matlab API for lane tracking, and obstacle detection and avoidance system, and steering control. The obstacle detection and avoidance system consists the image processing software and OOPic microcontroller with SRF04 sensors. Main microprocessor has the software for motor control.

Navigation challenge:

PC control software has a Matlab interface to decode GPS signal information. The obstacle detection and avoidance system consists the image processing software and OOPic controller with SRF08 sensor system. 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.

GPS

A Garmin GPS unit was used because of budget restrictions. The GPS receiver was interfaced to a serial port (RS232) of the computer. A serial interface program, using Matlab was developed to parse the GPS data coming through the serial port. We implemented a Kalman filter to increase the accuracy of the data. Also, we maintain a history of past velocities to determine the heading of the vehicle accurately. Initially, one point was chosen as a target. The vehicle was tested to see if it would repeatedly come back to the same point. As a result we decided to decrease the speed of the vehicle when it gets closer to the target. By slowing down the vehicle and filtering the data, we were able to come within 20 centimeters of the target with very high probability (we had less than 2 failures in 25 attempts). We decided to increase the probability of success by making 4 attempts at a target by moving away, turning around and making another pass. Theoretically, we can calculate the failure to be less than 10^{-3}. The next test we performed was the ability to go from one target to the next and eventually return to the starting point. For this 3 points were selected with one of them the starting point and the other two acting as waypoints.

Safety, Reliability and Durability

Our team wanted to ensure the safety, durability and reliability of the vehicle and as a result, the following methods were implemented into the design:

Manual emergency 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, 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.

Acknowledgments

The team would like to thank Elizabeth Tharakan and Sibu Verghese for their 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