RoboJackets IARRC 2016

George Smirnov, Ransomed Adebayo, Blake Lash, Yuki
RoboJackets
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332 United States of America
www.robojackets.org

Abstract

This paper addresses the mechanical, electrical, and software components of the RoboJackets' entry into the 2016 International Autonomous Robot Racing Challenge. A heavily modified Traxxas Slash 4x4 chassis hosts a variety of electrical components including a camera, LIDAR, and an Intel NUC running Ubuntu. The Robot Operating System (ROS) meta-operating system provides much of the framework for the software of this autonomous system.

I. INTRODUCTION

RoboJackets is the student-run competitive robotics organization at the Georgia Institute of Technology. Founded in 1999 as a BattleBots team, the organization has since grown to include RoboCup Small Size League, the Intelligent Ground Vehicle Competition, the International Autonomous Robot Racing Challenge, and a large outreach team. All teams at RoboJackets compete in competitions both locally in the United States and Internationally. While chartered in the Woodruff School of Mechanical Engineering, members come from all departments (predominately computer science, mechanical engineering, aerospace engineering, and electrical engineering) to participate in these extracurricular activities.

II. MECHANICAL

A. *Design Principles*

The vast majority of new members on this team have had no previous experience in the engineering design process. As such, the year began with an overview of material that the

students would likely see much later in their college careers. Multiple design methods were studied including the methods popularized by Dr. Singhose in the undergraduate course ME2110: Creative Decisions and Design, the more classical Pahl and Beitz design method, and the more contemporary Open Engineering System concepts.

Ultimately these methods were fused to give the most exposure to the engineering design process. Design tools the students would use in their ME2110 class work, such as Objective and Function Trees, Morphological Charts, and Concept Evaluation Matrices were integrated into the structure of the Pahl and Beitz method leading from Task Clarification to Conceptual Design, Embodiment Design, and as much Detailed Design as possible without introducing more advanced engineering concepts such as deformable bodies. This process was guided by the principles of Modularity, Mutability, and Robustness explored in Open Engineering Systems.

Due to observations from the competition in 2015, several changes were made in preparation for the 2016 edition of the competition. It was decided to build a new version of the racecar without the destruction of the old racecar in order to have a platform for software testing during build time. The basic structure of the new car was derived from the previous car with changes in areas that lacked competence. The embodiment and initial detailed design was completed in Solidworks. Following a brief introduction to CAD, the members were trained on using Inventor and set out to reverse engineer the existing components of the selected donor RC car to develop a baseline for further modification.

B. *Platform Overview*

A Traxxas Slash 4x4 was selected to be the donor vehicle for this project due to the high modularity of the design. Figure on the right illustrates this strength, demonstrating how the entire center of the car may be simply replaced while maintaining the integrity of the suspension and driveline. This substantial modularity would

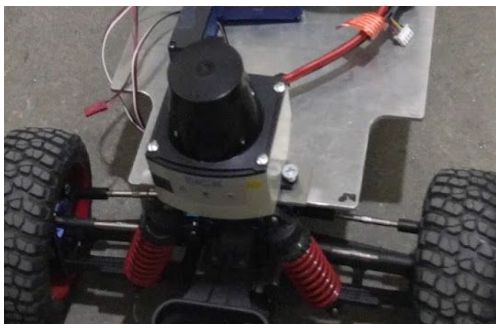
easily enable modification or replacement of selected components without modifying the basic architecture of the vehicle. The basic structure of the car, including suspension geometry and driveline, would remain unmodified throughout the entire process. Nearly every other component would be modified, replaced, or relocated to enhance the performance of the vehicle.

C. *Camera Mount*

Other components, such as the camera mount, were designed with flexibility in mind. The design of the new camera mount is not completely different from the old one. Notably, as a result of the reduction of overall size of the car, the camera has a higher view than previous models due to the relative size of the car and height of the camera. The mount also allows for easy height and tilt adjustment. While a fixed mount would be an acceptable solution, a mutable mount is a more robust solution in that it enables camera adjustment quickly and easily. Such a feature becomes a necessity when testing and modifying software as well as replacing many of the other component modules on the physical car that may occlude the camera's view.

D. *Lidar*

This year's robot still includes the SICK Tim 55x LIDAR, which is mounted in the front of the car. Unlike the previous car, the lidar is mounted upright because the car is low enough that accurate and competent sensing can be made normally.



E. *Steering and Suspension*

The steering and suspension of the vehicle has been modified to account for a heavier payload. Stiffer springs were added to the front and rear shock towers to remedy body sag. The

single steering servo was replaced with Traxxas Waterproof 2075 High Torque servo, allowing for stronger and more responsive turning. Custom linkages were made to connect the servo to the steering column and a servo saver was installed to protect the servos in the event of a hard impact on the front wheels.

III. ELECTRONICS

The electrical tasks for this project were largely relating to distribution. This work is naturally divided into two groups, determined by whether the signal distributed was power or information.

A. *Power Distribution*

One of the major tasks faced by the electrical team was power distribution within the robot. Sensors, actuators, and processing nodes each had to be provided with a power signal capable of appropriate current and voltage.

B. *Signal Distribution*

A second task faced by the electrical team was providing the physical integration of the different informational systems that formed the robot. In addition to providing connectivity between sensors and processing nodes, the formerly remote controlled vehicle was retrofitted to be capable of autonomous operation. This involved two primary pieces of work: determining the electrical signals necessary to control the vehicle itself as well a deciding upon the hardware needed to do so easily and reliably. Since no electrical documentation was provided with the vehicle, an oscilloscope was used to determine the protocol used by the robot to control its motors. Once this was determined, it was necessary to decide on a system for delivering these signals. To ensure predictable receipt of motor commands, an Arduino was selected to interface with the motor controllers. Due to its lack of operating system resource management, the Arduino is able to deliver commands at exact known intervals.

C. *Safety*

Several design features were added to ensure the safety of both the robot itself and the humans around it. For the safety of the robot itself, steps were taken to electrically protect the most valuable parts of the system which primarily comprised the sensors of the robot.

The power distribution system also used a number of PTC fuses and polarized connectors. The fuses enable the team to safely leverage the power distribution circuitry while ensuring that the aggregate demand of the electronics did not damage the board. Additionally, the use of polarized connectors through the design insured that no power signal would be accidentally reverse, thus removing one of the major potential causes of electrical damage.

IV. SOFTWARE AND PROGRAMMING

A. *System Architecture*

Due in part to its common usage in the field of robotics, the software team chose to design the autonomous car solution on the pre-existing Robot Operating System (ROS) platform.

ROS is an open-source, meta-operating system for robots. It provides services you would expect from an operating system, such as hardware abstraction, device drivers, libraries, visualizers, an inter-process communication protocol (ICP), package management, and more. The ROS communication architecture allows ROS to operate during runtime as a ”graph” or peer-to-peer network of processes (potentially distributed across machines).

ROS was a natural choice for our software team because it conferred several key advantages, listed below.

- C++, Python, and Java bindings.
- Pre-existing drivers for Sick and USB 2.0 cameras.
- A modifiable graphical frontend, rviz.
- OpenCV integration.
- Multi-process architecture.
- Fast iterations between code and results.

B. *Cone Detection*

A Sick TIM551 laser range finder was used to perform cone detection. The Sick laser collects a dense array of range data points at known angles. The laser measures range by sending out a pulse of light and recording its elapsed time of flight. A pinning mirror within the TIM55x points the laser across a 270 degree arc.

To detect cones, the points from the laser scan are clustered and then noise is eliminated by removing clusters that are too close to each other (for example when the robot is near a wall) and the resulting clusters are plotted and then analyzed further to detect features on the road.

C. *Lane Detection*

During the race new frames from the PS3-Eye are continually captured and searched for boundary lines. Each time a new frame is received, the following process takes place.

1) A region of interest representing the observable ground is taken.

2) The image is resized such that the boundary lines are a known width.

3) 16 kernel filters are passed over the image, each looking for line edges at different angles, offset from the center.

4) Complementary line edge pair results are elementwise multiplied together.

5) Orthogonal line edge pair results are subtracted from each other.

6) The resulting image is smeared left for white and right for yellow to create weighted ”drive” and ”no-drive” regions.

A pure filtering approach was chosen over more involved methods due to time constraints in the update loop, caused by high speed driving and high camera frame rate.

D. *Stoplight Detection*

The race begins when a stoplight changes in color from red to green. Our Stoplight Detection ROS node retains the 5 most recent frames captured by the PS3-Eye in a buffer. Each time this buffer updates, a results value r is calculated as seen below, and if greater than an experimentally derived trigger threshold, the stoplight was observed switching from red to green, and the Stoplight Detection node broadcasts a go event and ceases all further computation for the remaining life of the program.

Let I0 be the oldest image in the buffer, and In,red be the red channel in the nth oldest image. Let d be the distance in pixels between the red and green light centers of the stoplight. Let R be the result of the element-wise multiplication of the decrease of redness and the offset increase of greenness between the most and least recent images in the buffer, calculated according to the following equation:

$$R[r, c] = \{(I0,red[r, c] - I0,green[r, c] + I0,blue[r, c]\ 2\ -\ I4,red[r, c] - I4,green[r, c] + I4,blue[r,c])\} \otimes (I4,blue[r + d, c] - I4,red[r + d, c]) - (I0,blue[r + d, c] - I0,red[r + d, c])$$

Let $K = 1/\ n2 * \ (1 \ldots 1 \ldots\ldots\ldots 1 \ldots 1)$ be an $n \times n$ image kernel where n is the approximate width and height of the observed light.

Let $r = \max(R,K)$ be the maximum result of convolving R with K. The value of r then represents the maximum average transition from red to green in a stoplight shaped area of the image. If this value is low, it could be due to random noise. If this value is high, it is a light change.

E. *Vehicle Control*

The Traxxas Slash 4x4 is normally controlled by a handheld joystick with two analog inputs. The first input controls the position of the front servo drive, effectively steering the car. The second input controls the velocity of the rear wheels.

To control the motors from software, an Arduino replaced the radio module onboard the Traxxas. Error checking and safety timeout behaviors are coded into the Arduino motor controller. The Traxxas motors had a peculiar failure mode where if no command was received during a cycle, the motors would then spin out of control. Consequently, it is necessary to run the motor controller off the Arduino clock to avoid missing a control loop window.

F. *Autonomous Driving*

Driving is done via Reactive Control. We chose this method due both to its relative simplicity and its robustness. We first aggregate sensor data from our camera and LIDAR and the additional processors into a set of scaled images. These images represent the relative utility of each area of our world, for example road lines are low utility, but the area between them has high utility. OpenCV images were chosen for this because they are easy to view and show to human users, easy to compare to raw sensor data, easy to modify in code, and easy to process by our driving nodes.

The images are then aggregated by taking the average of each grayscale value at each point in each image. After aggregating all of these images, we choose the path (angle) with the

highest total utility and provide that angle to the wheel controller to steer in that direction. This is done via a simple linear scan over all angles in the aggregate image.

V. CONCLUSION

The hands on experience of designing, building, and programming an autonomous robot is invaluable for learning and internalizing the engineering design process. While class lessons may provide the technical expertise required to be an engineer, they do not teach the skills necessary for product development or success in a professional environment.

Our competition entry focused on the core concepts of modularity, flexibility, and robustness enabling a rapidly deployable design. The combination of off the shelf components as well as custom designed modules provided for fast prototyping as well as rolling upgrades that did not require a full rebuild.

We hope our entry into the 2016 competition will be a strong competitor, provide a solid platform for future competition entries, and encourage the promotion and education of robotics to beginning engineering students.

VI. ACKNOWLEDGMENT

The RoboJackets would like to thank our sponsors. Without their generosity, none of our work would be possible.

- George W. Woodruff School of Mechanical Engineering
- College of Computing, Georgia Institute of Technology
- Institute for Robotics and Intelligent Machines
- Student Government Association, Georgia Institute of Technology
- Caterpillar • General Motors
- General Motors Foundation
- United Technologies Corporation
- National Instruments
- MSC Industrial Supply Co.
- MathWorks

VII. REFERENCES

[1] [Online]. Available: http://traxxas.com/sites/default/files/6807L-LCGmodular-assembly.jpg