

RoboJackets 2015 IARRC

Casey Barnette, Alex Gurney, Joshua Morton, Alberto Sainz de la Pena, Matthew Barulic

RoboJackets

George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology

Atlanta, GA 30332

United States of America

www.robjackets.org

***Abstract*—This paper addresses the mechanical, electrical, and software components of the RoboJackets’ entry into the 2015 International Autonomous Robot Racing Challenge. A heavily modified Traxxas Slash 4x4 chassis hosts a variety of electrical components including a camera, LIDAR, and Lenovo Thinkpad running Linux. 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. 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 classwork, 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.

The embodiment and initial detailed design was completed in Autodesk Inventor. While this is the CAD package taught in the undergraduate course ME1770: Introduction to Engineering Graphics and Visualization and used on the other RoboJackets projects, the vast majority of our members had never used CAD software before. Following a brief introduction to CAD, its history and its uses, and its fundamental concepts, 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 1 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



Fig. 1. Modularity of the Traxxas Slash 4x4 Design [1]

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

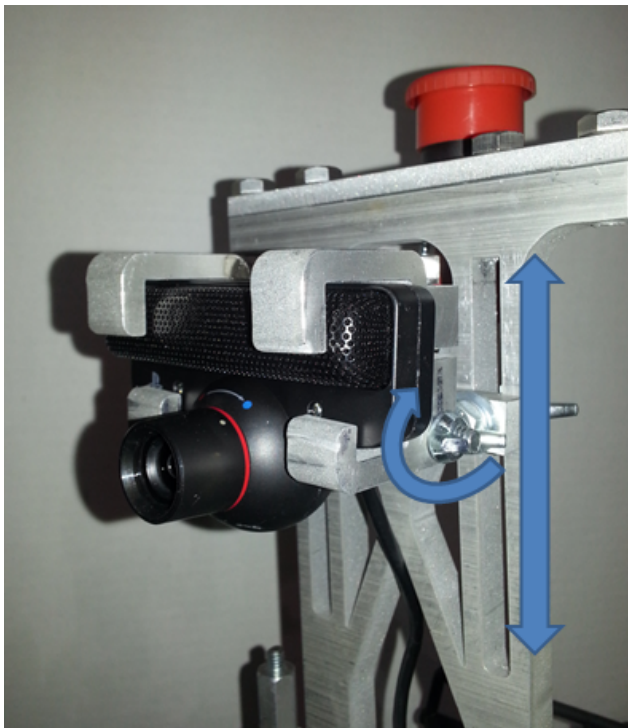


Fig. 2. Adjustable Height and Tilt Camera Mount

Other components, such as the camera mount in Figure 2, were designed with flexibility in mind. This mount 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. Roll Cage

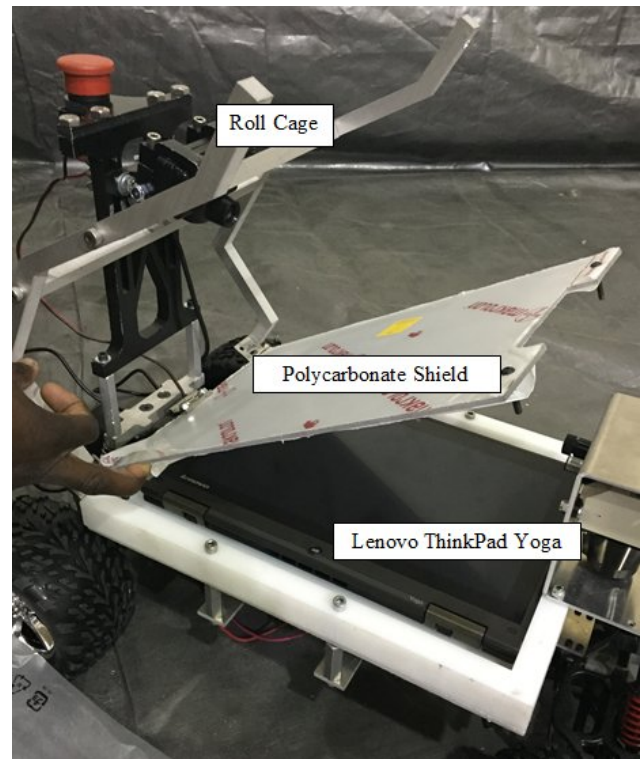


Fig. 3. Roll Cage Protecting Major Electrical Components

The laptop tray and roll cage (as seen in Figure 3) were designed to securely carry a Lenovo ThinkPad Yoga in its tablet orientation. The clear polycarbonate shield snugly keeps the laptop in place, while allowing the screen to be easily viewed for real-time debugging and code initialization. The roll cage

was designed to avoid large forces from potential roll-overs. All components of these two assemblies were manufactured using a Maxièm 1515 WaterJet.

E. LIDAR



Fig. 4. LIDAR mount

This year's robot includes the addition of the SICK Tim 55x LIDAR, and therefore a LIDAR mount (as seen in Figure 4). The mount was designed not only to safely secure the LIDAR to the robot, but also to protect the sensor from impact, debris, and inclement weather. It is made of a single sheet of aluminum 6061, which was cut using a Maxièm 1515 WaterJet and formed with a sheet metal brake. After the initial mount was made and attached, it was determined that the upright orientation of the LIDAR produced a scanning plane too high above the orange cones of the race course. To remedy this, a new mount was designed in which the orientation of the LIDAR was flipped vertically. This change in orientation allows the robot to effectively sense the cones.

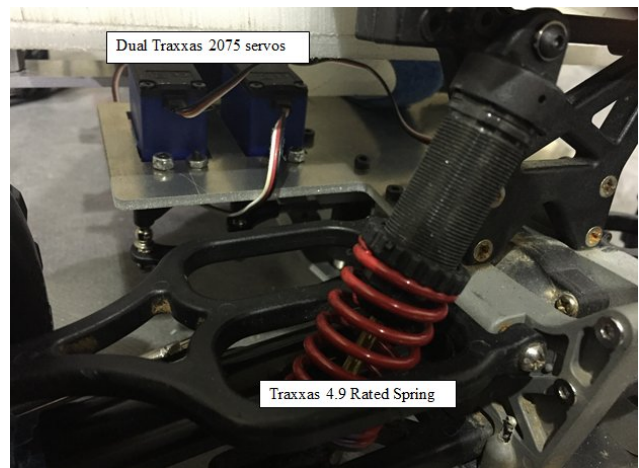


Fig. 5. Steering and Suspension

F. Steering and Suspension

The steering and suspension of the vehicle has been heavily modified to account for a heavier payload. Stiffer springs were added to the front and rear shock towers to remedy body sag. An additional set of springs was also added to the rear shock tower in order to support the added weight of the laptop, laptop tray and roll cage. The single steering servo was replaced with two Traxxas Waterproof 2075 High Torque servos, allowing for stronger and more responsive turning. Custom linkages were made to connect the two servos 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. ELECTRICAL

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 as 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 Nano 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.

To help protect sensitive electronics from the large voltage fluctuations caused by the motors powering the RC vehicle, protective circuits divide the robot's power system into two parts. The first provided power to the drive and steering motors on the robot, while the second provided the power necessary for all sensors and computation.

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.

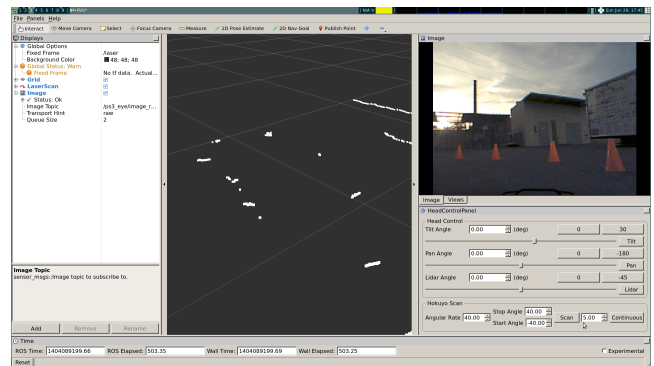


Fig. 6. IARRC RViz GUI Interface.

IV. SOFTWARE

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

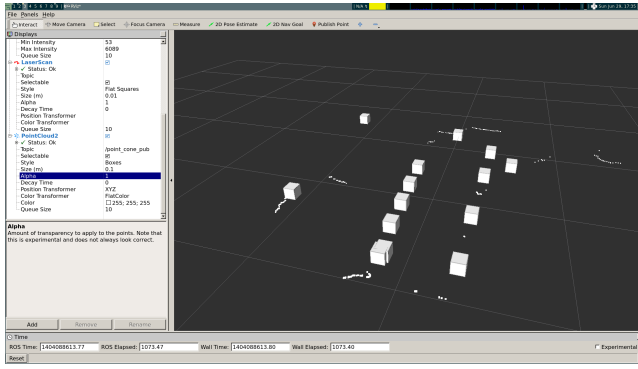


Fig. 7. Cone detection algorithm. The white boxes are the tracked cones.

spinning mirror within the TIM55x points the laser across an 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 kernal 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 element-wise 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 I_0 be the oldest image in the buffer, and $I_{n,red}$ be the red channel in the n th 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 Equation 1

$$\begin{aligned}
 R[r, c] = & \left\{ \left(I_{0,red}[r, c] - \frac{I_{0,green}[r, c] + I_{0,blue}[r, c]}{2} \right) \right. \\
 & \left. - \left(I_{4,red}[r, c] - \frac{I_{4,green}[r, c] + I_{4,blue}[r, c]}{2} \right) \right\} \\
 & \otimes \left\{ \left(I_{4,blue}[r + d, c] - I_{4,red}[r + d, c] \right) \right. \\
 & \left. - \left(I_{0,blue}[r + d, c] - I_{0,red}[r + d, c] \right) \right\} \quad (1)
 \end{aligned}$$

Let $K = \frac{1}{n^2} * \begin{pmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{pmatrix}$ be an $n \times n$ image kernel

where n is the approximate width and height of the observed light

Let $r = \max(R \star K)$ 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 2015 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.

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
- Institute for Robotics and Intelligent Machines
- Student Government Association
- Caterpillar
- General Motors
- General Motors Foundation
- United Technologies Corporation
- National Instruments
- MSC Industrial Supply Co.

- MathWorks

REFERENCES

- [1] [Online]. Available: <http://traxxas.com/sites/default/files/6807L-LCG-modular-assembly.jpg>