RoboJackets 2014 IARRC

Joseph Hickey, Eric Huang, Ben Nuttle, Alberto Sains de la Pena, Alexander Trimm
RoboJackets
George W. Woodruff School of Mechanical Engineering
Georgia Institute of Technology
Atlanta, GA 30332
United States of America
www.robojackets.org

***Abstract*—The RoboJackets joined the International Autonomous Robot Racing Challenge with the intention of recruiting and training new members. A heavily modified Traxxas Slash 4x4 hosts a variety of electrical components including a camera, LIDAR, and single board Linux computer. 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 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, Intelligent Ground Vehicle Competition, and a large outreach team. While chartered in the 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.

For the first time since 2007, the RoboJackets introduced a new team. The International Autonomous Robot Racing Challenge team strives to create a self driving race car built off of an RC car chassis.

A. Motivation

A significant challenge facing the RoboJackets is the recruitment and retention of new members. Many of the existing teams are so well established that they seem daunting or intimidating for new students to attempt to join while still very early in their curricula.

This team addresses the steep learning curve by matching primarily underclassmen members to upperclassmen mentors with the intention of helping new students jump-start their college experience. This team provides a first experience with collegiate competitive robotics and will give younger students the skills necessary to succeed as members of more demanding project teams while building bonds within the RoboJackets community.

B. Objectives

This team began with two stated primary objectives and an additional secondary objective beyond the competition's stated objectives.

- **Primary Objective 1**: Recruit new members and train in systematic design processes and subject specific technical matters including CAD, machining, circuits, soldering, C++, among other areas.
- **Primary Objective 2**: Design and build an autonomous car to race in the 2014 IARRC.
- **Secondary Objective 1**: Recruit members at the end of the year to take on IARRC project management roles. Remaining members work on IGVC and RoboCup.

The remainder of this paper will serve to address the mechanical, electrical, and software components of the RoboJackets' entry into the 2014 International Autonomous Robot Racing Challenge.

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 prcoess 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 selected components without modifying the basic architecture of the vehicle. The basic structure of the car, including suspension geometry and driveline, would remain unmodified throughtout the entire process. Nearly every other compent would be modified, replaced, or relocated to enhance the performance of the vehicle

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

Other components, such as the camera mount in Figure 2, were designed with robustness 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.

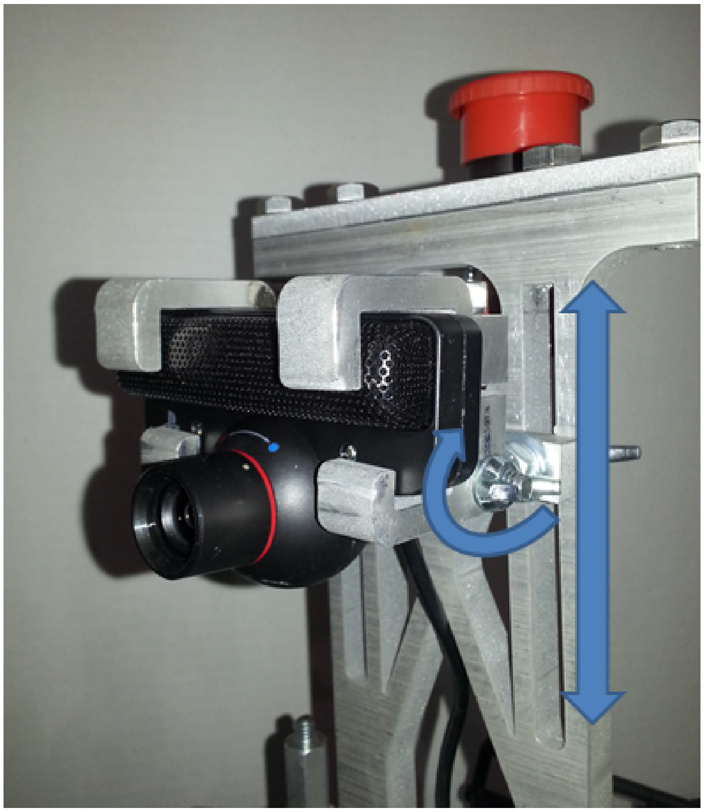


Fig. 2. Adjustable Height and Tilt Camera Mount

Furthering the concept of robustness from robust to changes in the design environment to a concept of durability necessitates components such as soft, deflective bumpers and a roll cage as illustrated in Figure 3. Every major and expensive electrical component is covered by a roll cage to provide protection in the event of a collision with a static or moving object as well as a complete rollover of the vehicle.

Fig. 3. Roll Cage Protecting Major Electrical Components

C. *Sensory Equipment*

The Slash 4x4 includes a rotary encoder that functions based upon the Hall effect to provide a measure of the speed of the vehicle. Additional components added to the vehicle include a camera for an RGB view of the world and a LIDAR for high precision laser scanning.

A PS3 Eye was selected for vision due to its affordability and purpose built design for machine vision. A Hokuyo URG-04LX remaining from previous RoboJackets projects was included due to its high accuracy and small size.

III. ELECTRICAL

The electrical tasks for this project were largely tasks of 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 having appropriate specifications. Figure 4 shows the connectivity between each of the components in the vehicle.

In order to provide the necessary connectivity, the team produced a custom power distribution PCB, shown in Figure 5, which provided a 5V, 2.5A bus which was then branched off to provide power for all of the processing and sensing nodes of the robot. This design leveraged the inboard distribution of the PandaBoard, which provided both 5v and 3.3V outputs, in order to simply the task of integrating the necessary sensors. In addition to the provided level shifting and voltage regulation, the power distribution board also provided for some more practical considerations of the development process. One such design feature was circuitry to allow the electronics to be powered off of a ubiquitous wall wart power supply in addition to the battery to be used in the competition. This feature enabled the team to test the platform indefinitely without the need to either purchase redundant batteries or suffer mandatory downtime while allowing them to recharge.

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 systems. To overcome the hardware and software limitations of the PandaBoard, the decision was made to use offload the lower level control of the robot onto an ATMEGA 328 based Arduino microprocessor and provide a higher level interface from the central processing unit, the PandaBoard. The

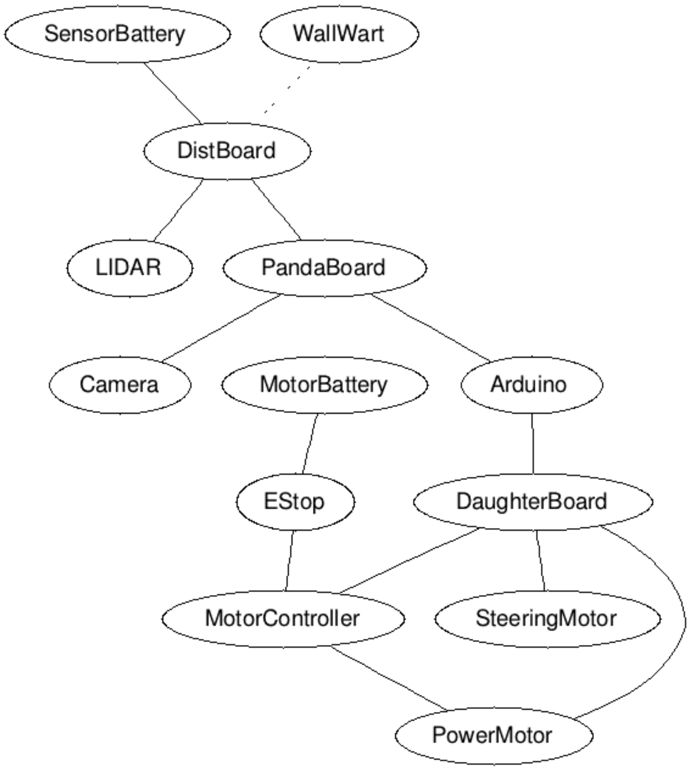


Fig. 4. Electrical Connectivity Chart

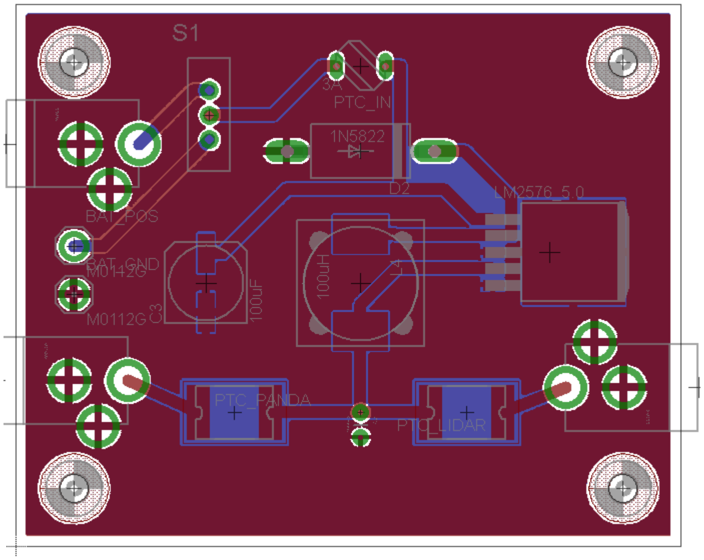


Fig. 5. Power Distribution PCB

Pandaboard does not feature hardware to generate the servo-like signals required by the motors. While providing these signals by means of software was theoretically possible, it proved inefficient to do so. Additionally, the PandaBoards full operating system made control timing stochastic since other tasks would often share processing time with the robots control logic. This provides large potential complications for real-time systems.

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 causes by the motors powering the RC vehicle, two separate batteries were used within the robot. 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 board in Figure 5 also used a number of PTC fuses and polarized connectors. The fuses enable the team to safely leverage the power distribution circuitry built into the PandaBoard 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

A. System Architecture

Due in part to the introductory nature of our IARRC team, 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 protol (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 confered several key advantages, listed below.

- C++, Python, and Java bindings.
- Pre-existing drivers for Hokuyo 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 Hokuyo URG-04LX, a 1D laser range finder, was used to perform cone detection. The Hokuyo collects a dense array of range data points at known angles in a 180 degree field of view. (Futhermore, it has excellent performance in outdoor sunlight.) The Hokuyo measures range by sending out a pulse of light and recording its elapsed time of flight. A spinning mirror within the Hokuyo points the laser across an 180 degree arc.

To extract the cones, the laser scan points were clustered using the metric distance between consecutive points. The width of each cluster is computed and then thresholded to select the cones. The locations of the nearest cones are used to determine the current heading of the vehicle. Figure 7 shows a visualization of this clustering algorithm.

C. Lane Detection

The PS3-Eye is mounted on a fixture located at the back of the Traxxas. The fixture is capable of varying camera height and tilt. Using this setup, the camera is positioned to capture the stoplight at the top of the frame and the lanes in as much of the bottom of the frame as possible.

The lane detection algorithm works in the following way. First, any image above the calibrated horizon is discarded. In addition to that, any car chassis in the image is zeroed out. OpenCV is then used to run the standard edge detection pipeline. The image is then smoothed with a Gaussian kernel and detect edges using the Canny algorithm. Before running contour detection on the edge image, dilation and erosion operations remove spots and join noisy edges into connected paths. On the post processed image, the contours detected arise at the granularity of lanes and cone outlines.

D. Stoplight Detection

Canny edges with Hough Circles run on the results was used for stoplight detection. To remove false positives from the output of Hough Circles, the results were filtered by checking the spread of the pixel intensity values within the circle. If the pixels were of a reasonably consistent intensity (as one would expect of a stoplight) the circle was taken as a valid stoplight, if not, then it was rejected. To detect the change of the stoplight from red to green, the vertical displacement between detected circles must be greater than the radius of the last detected circle.

E. Vehicle Control

The Traxxas Slash 4x4 is nominally 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 all 4 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 recieved during a cycle, the motors would then spin out of control. Consequently, it was important to run the motor controller off the Arduino clock instead of the PandaBoard. If the PandaBoard GPIO pins had been used, we ran the risk of missing the control loop window for the Traxxas motors.

To communicate between the Arduino and the PandaBoard, we wrote a simple message serialization protocol. Message packets were comprised of an unique header byte, data bytes, a checksum, and an unique footer byte. When the PandaBoard and Arduino open the serial over USB connection, random bytes are written to the serial stream. We included the unique header/footer bytes and checksum to prevent the motor controller from mistakingly driving random commands through the motors. In practice, however, faulty electrical connections formed the root of our RC motor controller problems.

F. Autonomous Driving

We integrate the outputs of our spotlight, lane, and cone detection to generate vehicle steering and motor velocity commands. We employ a reactive control approach for completing the circuit race. The objective of our circuit racing controller is to keep the lanes (or nearest cones) equidistance from the image center (or laser scan origin) while maintaining a forward velocity. We handle the edge case where only one lane (or line of cones) is visible by creating an imaginary lane (or cone) a calibrated distance opposite the visible feature. Because the circuit has independent runs of pure lines or pure cones, the algorithm picks it's heading from the feature nearest and foward from the vehicle.

V. CONCLUSION

The RoboJackets had a successful year, completing each of our stated objectives. This new team provided many opportunities in a sandboxed environment to test new training and teaching techniques as well as refresh the way our organization approaches competitions. Many of the newly recruited members will matriculate into our more advanced teams while some will continue on the IARRC team in leadership capacities.

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, mutability, and robustness enabling a very flexible and 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 2014 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.

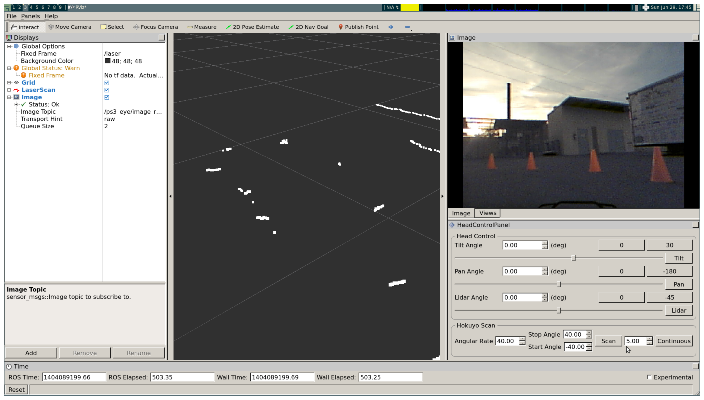
ACKNOWLEDGMENT

The RoboJackets would like to thank our sponsors. Without their benevolence and 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



Global Options
Fixed Frame /laser
Background Color 48; 48; 48
Global Status: Warn
Fixed Frame No tf data. Actual...
Grid ☑
LaserScan ☑
Image ☑
Status: Ok
Image Topic /ps3_eye/image_r...
Transport Hint raw
Queue Size 2

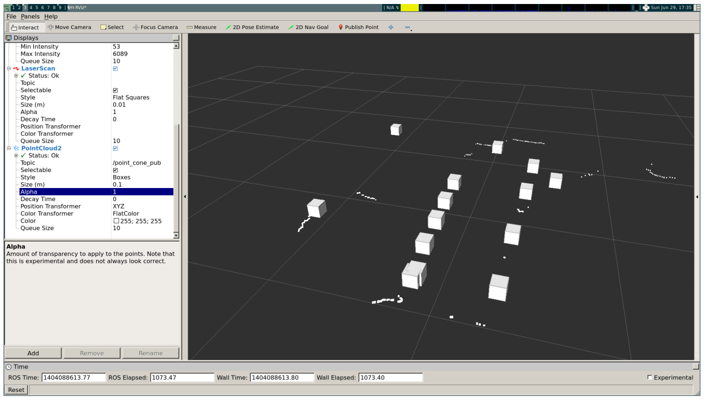
Image Topic
sensor_msgs::Image topic to subscribe to.

Head Control
Tilt Angle 0.00 (deg) 0 30
Pan Angle 0.00 (deg) 0 -180
Lidar Angle 0.00 (deg) 0 -45

Hokuyo Scan
Angular Rate 40.00
Stop Angle 40.00
Start Angle -40.00
5.00

ROS Time: 1404089199.66 ROS Elapsed: 503.35 Wall Time: 1404089199.69 Wall Elapsed: 503.25
☐ Experimental

Fig. 6. IARRC RViz GUI Interface.



Min Intensity 53
Max Intensity 6089
Queue Size 10
LaserScan ☑
Status: Ok
Topic
Selectable ☑
Style Flat Squares
Size (m) 0.01
Alpha 1
Decay Time 0
Position Transformer
Color Transformer
Queue Size 10
PointCloud2 ☑
Status: Ok
Topic /point_cone_pub
Selectable ☑
Style Boxes
Size (m) 0.1
Alpha 1
Decay Time 0
Position Transformer XYZ
Color Transformer FlatColor
Color ☐ 255; 255; 255
Queue Size 10

Alpha
Amount of transparency to apply to the points. Note that this is experimental and does not always look correct.

ROS Time: 1404088613.77 ROS Elapsed: 1073.47 Wall Time: 1404088613.80 Wall Elapsed: 1073.40
☐ Experimental

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

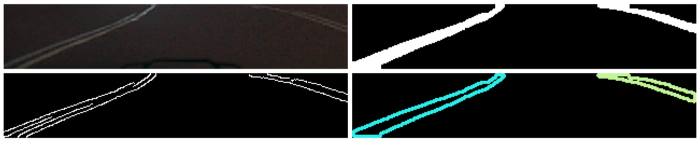


Fig. 8. Lane detection algorithm. Top left: Raw image. Bottom left: Canny edge output. Top right: Erosion and dilation output. Bottom right: Contour output, i.e. detected lanes.