Roadrunner

Southern Illinois University Edwardsville
2014-2015

I certify that the design and engineering of the vehicle “Roadrunner” by the SIUE Team Roadrunner has been significant and equivalent to what might be awarded credit in a senior design course.

Gary Mayer
Department of Computer Science
Southern Illinois University Edwardsville

I. Introduction

The Roadrunner is a collaboration of digital systems, power distribution, mechanical systems, and artificial intelligence. The intelligent ground vehicle (IGV) is built on the platform of a small form factor, single-rider golf-cart. The cart was transformed into an autonomous vehicle by modification of the steering, acceleration, and braking systems, and by the addition of various environmental sensors, processing components, artificial intelligence software, and power sources. Some of the key features of the Roadrunner are its differential GPS system, stereo vision, and its ability to carry a human passenger.

II. Organization

Team "Roadrunner" is a multidisciplinary team from Southern Illinois University Edwardsville (SIUE). The team is formed under a larger group known as Special Interest Group – Robotics (SIG-R). The team is comprised of three Computer scientist, three Electrical/Computer Engineers, and two Mechanical Engineers. The team formed in the summer of 2014 and has progressively working on the Roadrunner throughout the 2014-2015 school year.

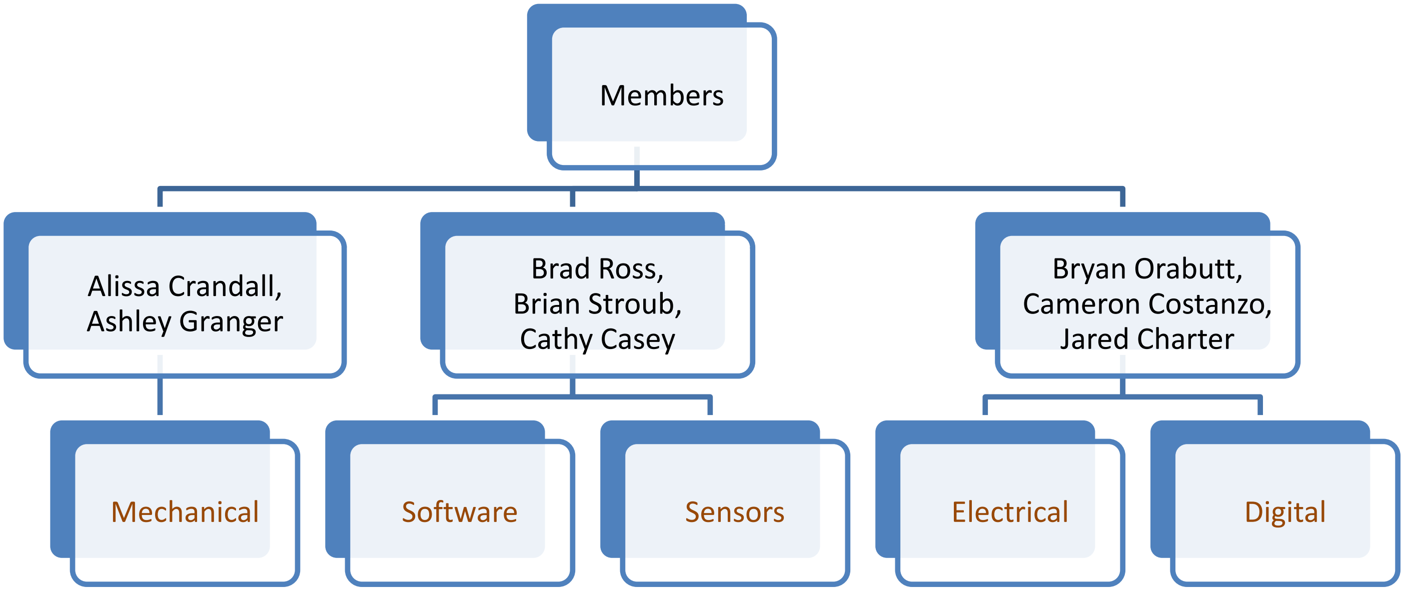


Figure 1: Systems delegation breakdown

III. Systems

1. Sensors

The Roadrunner uses a variety of sensors to sense its environment.

For orientation and waypoint navigation, two BU-353S4 GPS receivers are used. One GPS is placed in the front while the other is placed towards the rear of the vehicle. By keeping the identity of each GPS, the orientation of the vehicle relative to the course can be determined. Using the front of the Roadrunner as the origin, the coordinate of the vehicle is determined which thus allows the Roadrunner to determine a heading to the specified GPS waypoint on the course.

Figure 1: USB GPS antenna/receiver

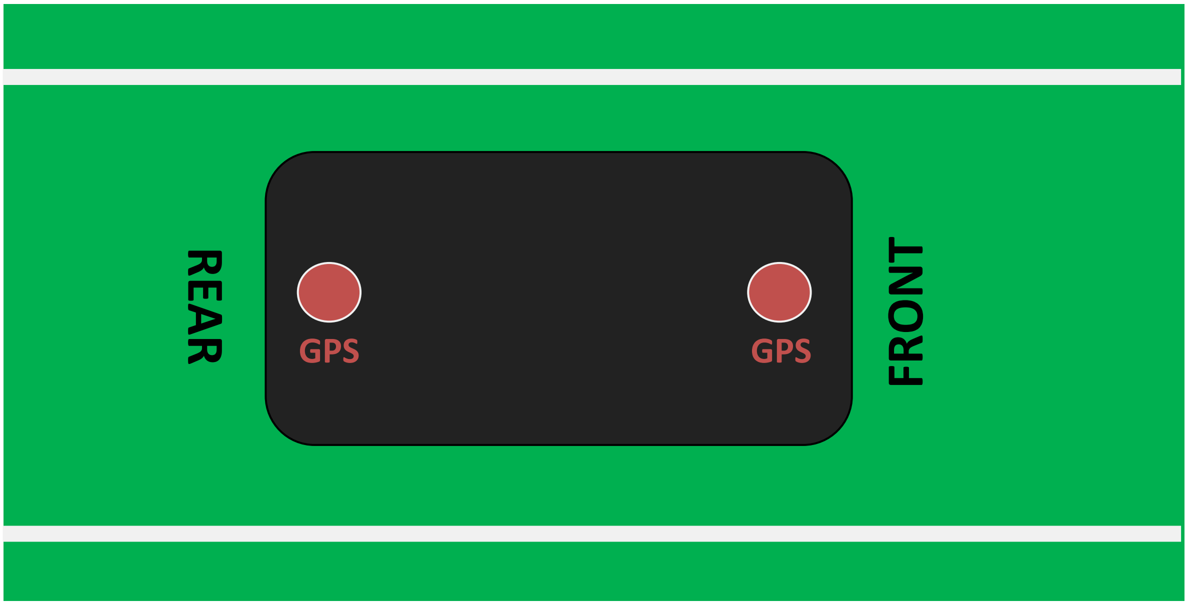


Figure 2: Model of differential GPS module placement (top view)

For line tracking and pothole detection, the Roadrunner is equipped with two Logitech webcams which are configured for stereoscopy. By using stereo vision, the viewing angle of the vehicle is increased for line-tracking on each side of the Roadrunner's path. The stereo vision also allows for distance to detected potholes to be calculated using triangulation with the measured distance between the two cameras, and the distances from each camera to the pothole.

Figure 3: USB Camera Module

Figure 4: Picture of the camera placement on the vehicle

A SICK LMS100 laser range finder (LRF) is the primary sensor for obstacle detection. This LRF emits a planar laser at a sweeping angle angle of 270˚. The maximum range of the LRF is 50m with a minimum of 0.5m. By filtering out the obstacles detected beyond the Roadrunner’s set detection threshold, obstacles such as barrels can be detected by the grouping of points of similar distance. This process is described in further detail in the AI Plans section of this report.

Figure 5: SICK LMS100 Laser Range Finder

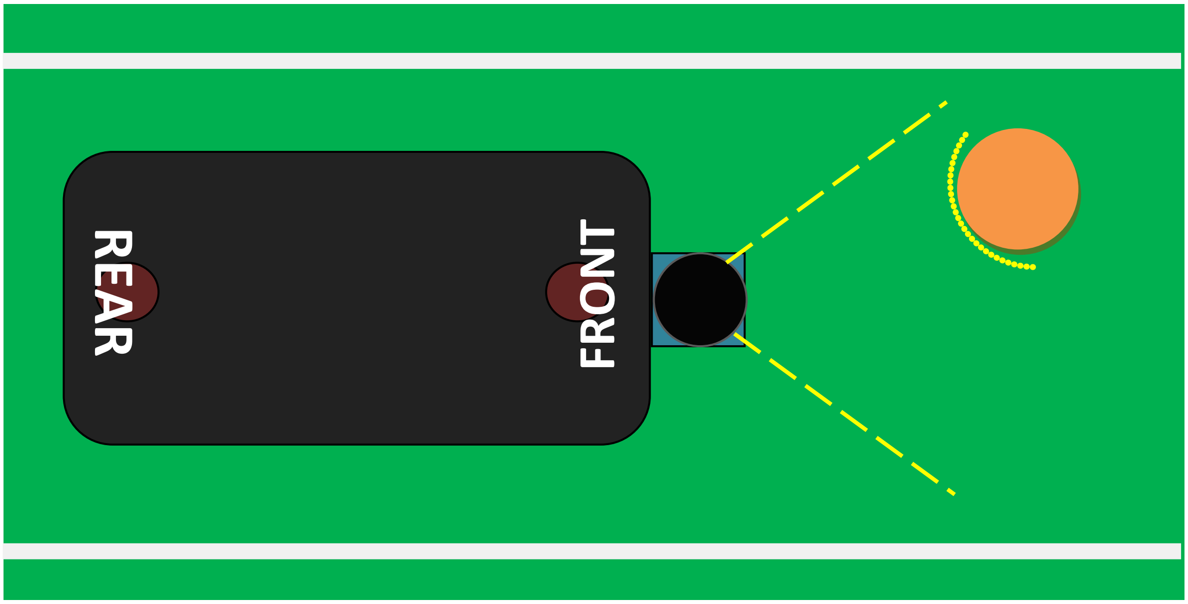


Figure 6: Model of LRF placement (top view)

An US Digital E6 encoder is located on both the drive shaft and the steering shaft of the Roadrunner. These encoders allow for speed, direction, and total distance travelled by the shaft to be calculated and determined. On the steering shaft, the encoder is primarily used to determine the angle which the wheels are turned to by keeping track of a count of the encoder signals. On the drive shaft, the encoder is used to maintain the speed the Roadrunner travels at and also can be used to keep track of the linear distance the vehicle has travelled in the course.

Figure 7: US Digital E6 Optical Encoder

A Vishay spectrol precision potentiometer (pot) is located at the top of the steering shaft. By capturing the voltage value of the pot as the wheels are facing straight forward, this value can then be used to compare with when the wheels need to be returned to straight forward. This is particularly useful when the vehicle needs to go in a straight line reliably and is also used to center the wheels on startup.

Figure 8: Vishay spectrol precision potentiometer

2. Digital Systems

In order for the Roadrunner's high-level laptop to interface with its low-level components it uses multiple microcontrollers and motor controllers.

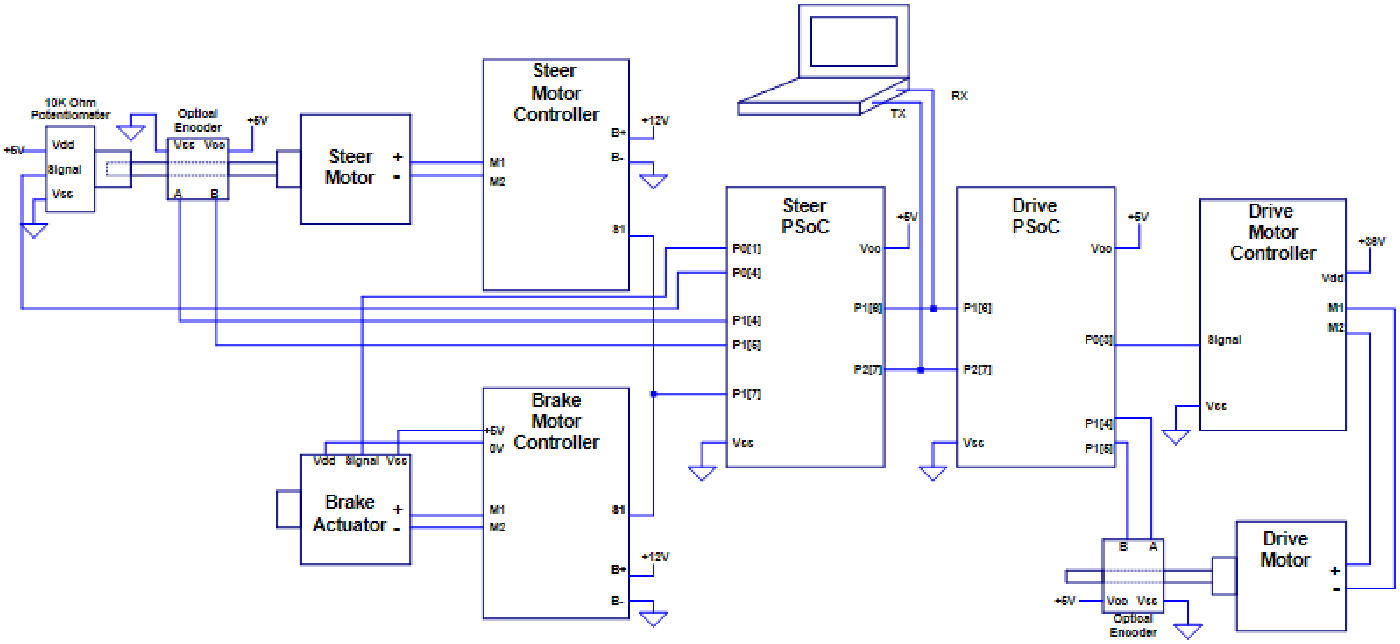


Figure 9: Roadrunner digital systems schematic

Two PSoC 1 microcontrollers along with their included evaluation development kit boards are used as the primary driving and steering controllers. Each PSoC interprets commands sent from the laptop, sends data back to the laptop, interprets signals from sensors, sends signals to low level components, and sends signals to the motor controllers.

A set of commands are used to tell the PSoC what to do. Each command sent to the PSoC's UART is parsed accordingly and the appropriate operation is performed. These operations include turning to an angle, getting an encoder count, driving forward at a speed, driving backwards at a speed, and pinging each controller.

The encoder count and potentiometer values are interpreted and stored on the PSoCs. These values are can then be sent to the AI on the laptop or used in calculations and comparisons directly on the PSoC.

The PSoC also sends signals to two 2N2222 transistors used for flipping a relay to put the Roadrunner in reverse, as well as powering a motor for the safety light in autonomous mode.

Motor controller signals to the steering and brake motor controllers are sent as 4-byte serial packages. These packages are calculated and sent from the PSoC on a single bus to these two Dimension Engineering Syren 25A controllers operating in a byte addressable, packet serial mode.

3. Electrical and Power Systems

The Roadrunner is powered by four 12VDC 28AmpHr. lead-acid batteries. Three of these batteries are in series to form a 36VDC supply. This supply powers the drive motor, through the drive motor controller, and supplies power to the 36V-12VDC inverter. The other motor controllers, motors, and the reverse relay are powered by this inverter. The fourth battery supplies power to the PSoC 5VDC regulator, the LMS100 LRF, and the E-stop receiver. Low-level logic components are supplied power by their controllers or by the USB ports of the laptop. A schematic of the power system breakdown is shown here:

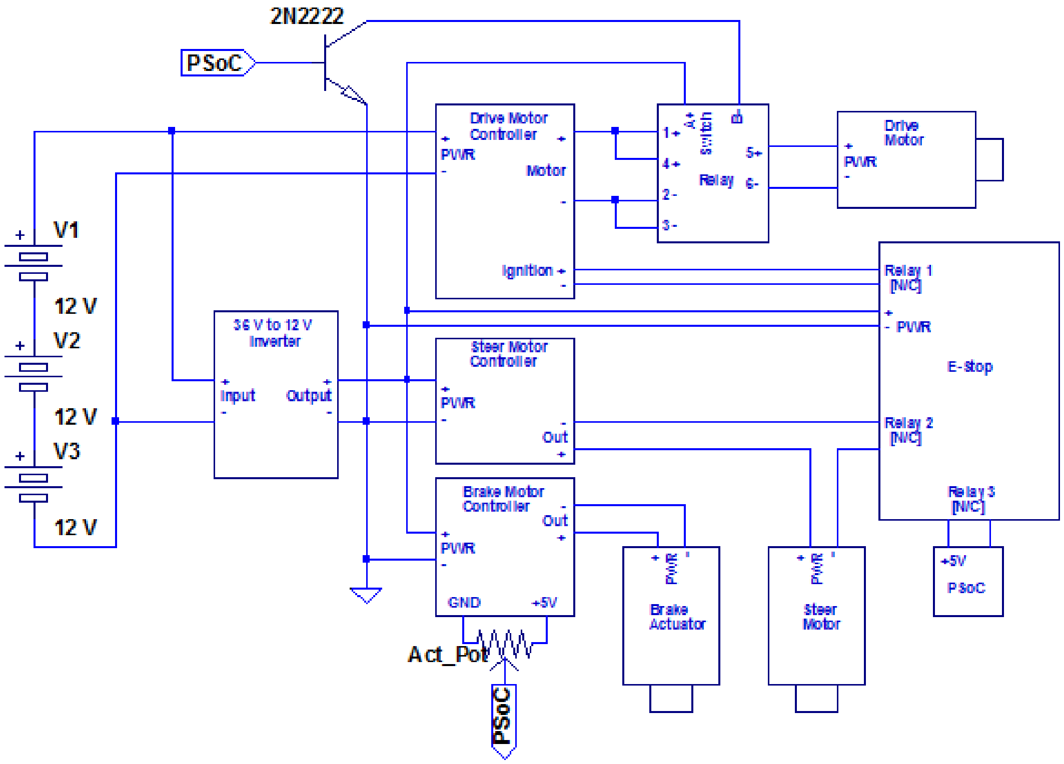


Figure 10(a): Roadrunner Power and Electrical System (3-battery configuration)

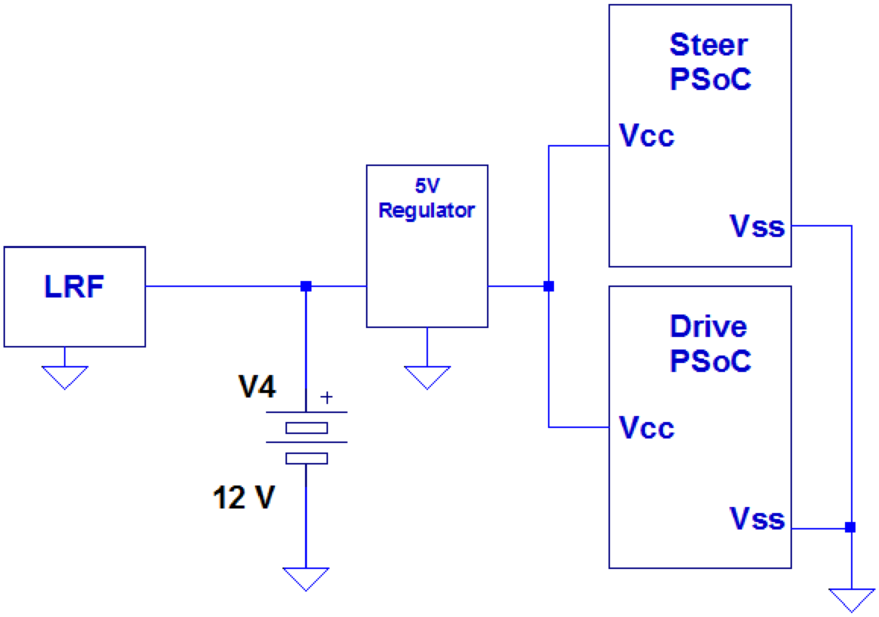


Figure 10(b): Roadrunner Power and Electrical System (separate battery)

4. Mechanical

The gear system for the Roadrunner's steering had been implemented by a previous team which worked with the golf cart [1]. The mechanical brake system had also been implemented by this team. However, both of these systems were modified and updated for this year's competition. Beyond the adjustments to these two systems, multiple 3D printed mounts and casings have also been added to the vehicle this year.

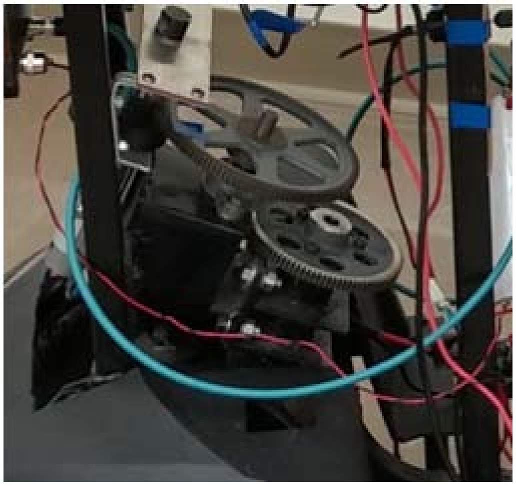


Figure 11: Image of the gear system for steering

---

[1] - See *References*, (1.)

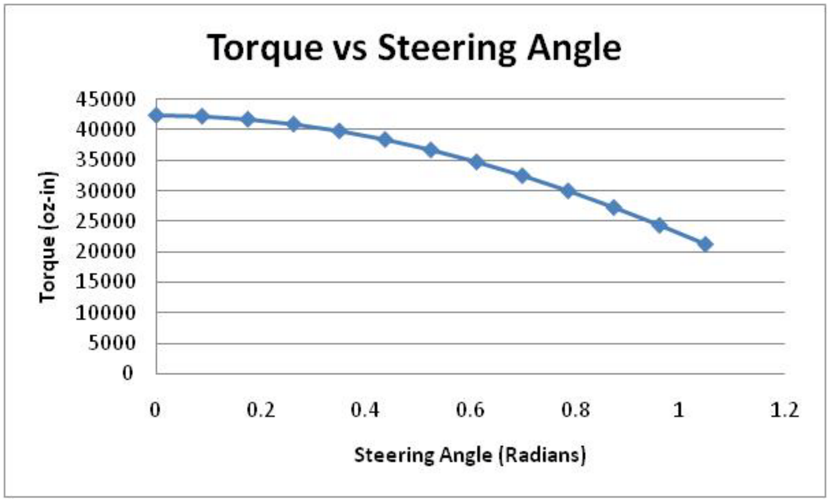


Figure 12: A tourque versus steering angle plot for the steering system

5. Software System

The Roadrunner AI is running on an 64-bit Ubuntu 12.04 distribution operating system. A version of Robot Operating System (ROS), ROS Hydro, is used as an operating environment for the software and AI. ROS allows for easier collaboration between sensor data, programs, and low-level components by modularizing each aspect to be its own node. These nodes then are able to use a common form of “talking” with each other.

Figure 13: Ubuntu 12.04 and ROS Hydro Medusa logos

IV. Core AI Plans

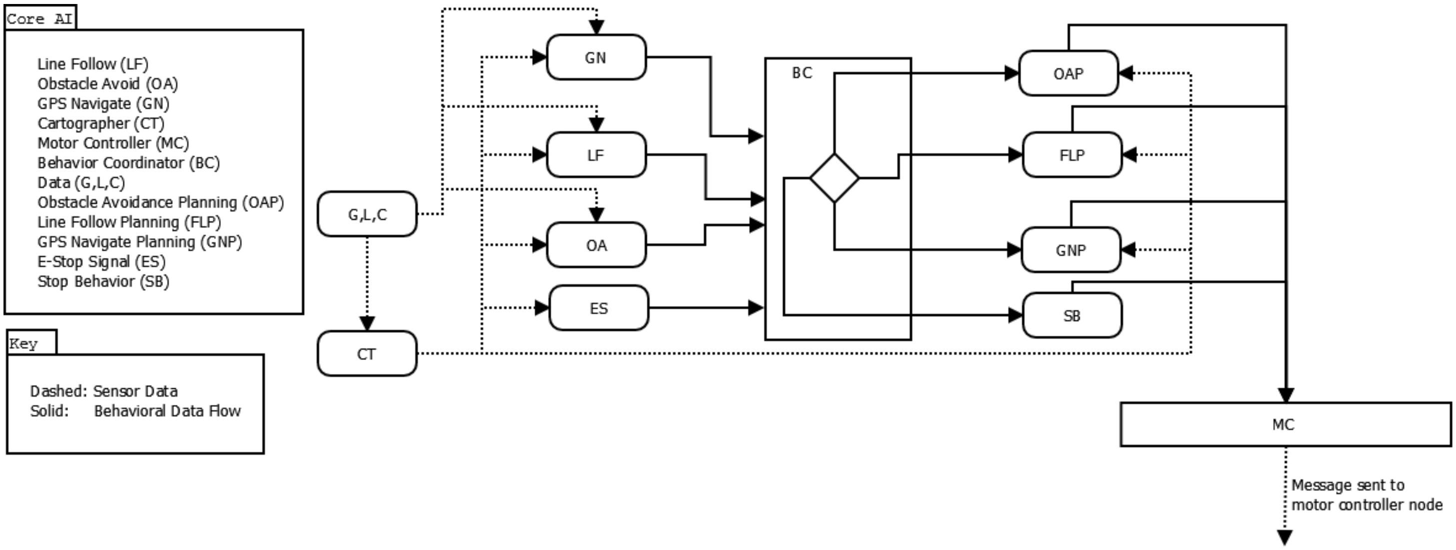


Figure 14: Diagram of Core AI architecture

Line Detection and Following

The process of detecting potholes and bounding lines involves using blob detection, a form of grouping pixels based on color to determine the characteristics of an object. Roadrunner uses its two USB cameras to capture images, and then those images are checked for certain colors. In the case of this competition, our cameras are programmed to detect and blob the color white within a certain threshold. By grouping pixels of the same color as blobs, we can then determine the size and location of these blobs and use that information to plan our route.

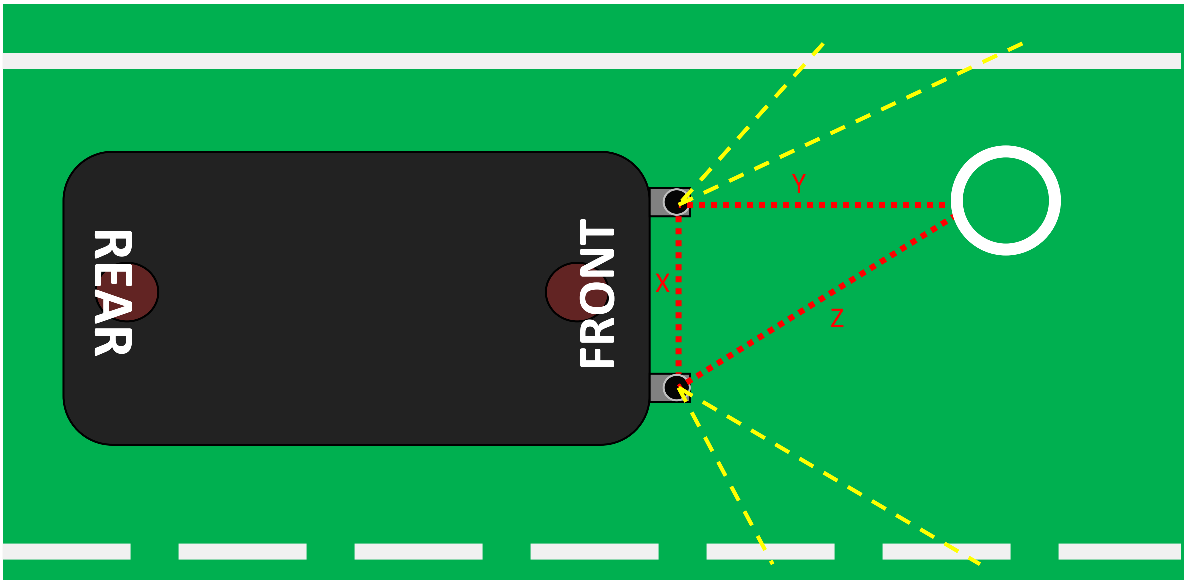


Figure 15: Model of camera blob detection and stereoscopy

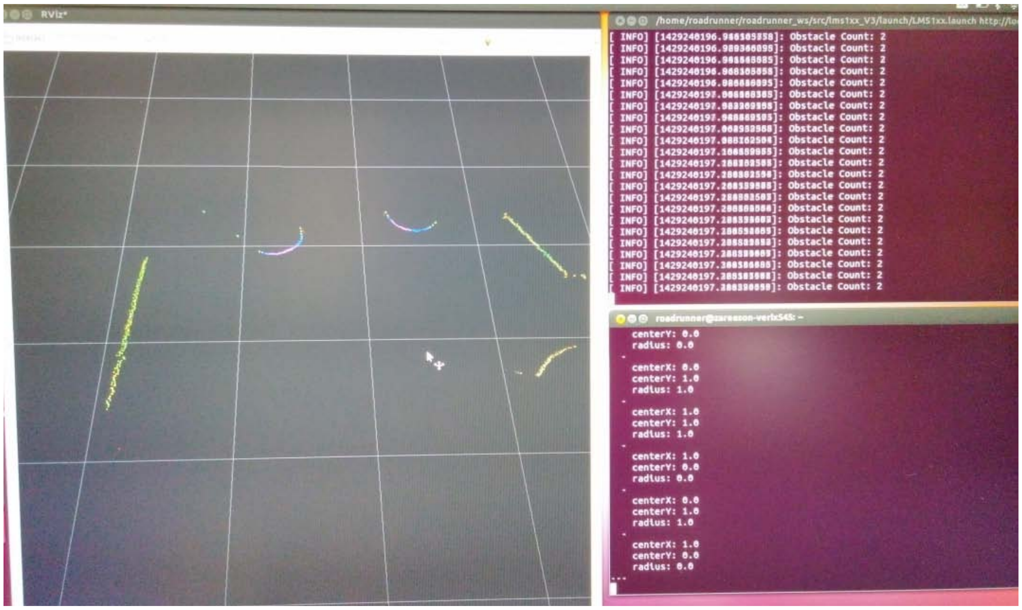
Using the blobs detected by the USB cameras, we can use the locations of these blobs to determine the distance from each blob on the left and right side of the picture. We then find the distance between the closest blobs on each side and use that value to get the midpoint. The Roadrunner then uses this value to center itself properly within the lines.

If a line is only detected on one side (due to an obstacle obstructing one side or being in no-man's land), the Roadrunner will prioritize another behavior (such as obstacle avoidance or waypoint navigation) based on the reasoning for not seeing two lines while also avoiding the line as best as possible.

Figure 16: Example of blob detection of white objects

Obstacle Detection and Avoidance

The Roadrunner uses its SICK LMS100 LRF to detect obstacles within a certain range. The shape, size, and location of the object is determined. Using these parameters, a circle can be inscribed if the object is determined to be barrel. The angle which the Roadrunner needs to turn to get around this object is then calculated and sent to the motor control node.



```
[ INFO] [1429240196.[illegible]]: Obstacle Count: 2
[ INFO] [1429240196.[illegible]]: Obstacle Count: 2
[ INFO] [1429240196.[illegible]]: Obstacle Count: 2
[ INFO] [1429240196.[illegible]]: Obstacle Count: 2
[ INFO] [1429240196.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
[ INFO] [1429240197.[illegible]]: Obstacle Count: 2
```

```
    centerY: 0.0
    radius: 0.0
  -
    centerX: 0.0
    centerY: 1.0
    radius: 1.0
  -
    centerX: 1.0
    centerY: 1.0
    radius: 1.0
  -
    centerX: 1.0
    centerY: 0.0
    radius: 0.0
  -
    centerX: 0.0
    centerY: 1.0
    radius: 1.0
  -
    centerX: 1.0
    centerY: 0.0
    radius: 0.0
---
```

Figure 17: Rviz capture of LRF barrel detection

Waypoint Detection and Navigation

Using two USB GPS units, our algorithm takes an average between the two units and uses that as the coordinate of each position throughout the course. The waypoint navigation algorithm will simply traverse from waypoint to waypoint using the shortest possible path (typically being a straight line.) However, if a white line or obstacle is impeding the shortest path, our behavioral coordinator will prioritize navigating around those impediments.

Stitching of the sensor data is done in a program named the cartographer. The inputs into this node are laser data, camera data, and the GPS data. Matrices are used to translate data from the location of each sensor reading and translate it to correspondence to the center of the front axle. The cartographers' data is used by the path finding functions as well as the methods that check for objects and set flags. The camera data will be rotated and translated to create a GPS location to travel as waypoints while line following. Laser data is translated to make center of detected obstacles as points in the world map that cannot be traversed. GPS points are used as goal points and the current point is assumed to be the front axle center due to the area covered by the GPS radius. The output from the cartographer is the location to the world map, which will be in relation to GPS location, and a virtual map

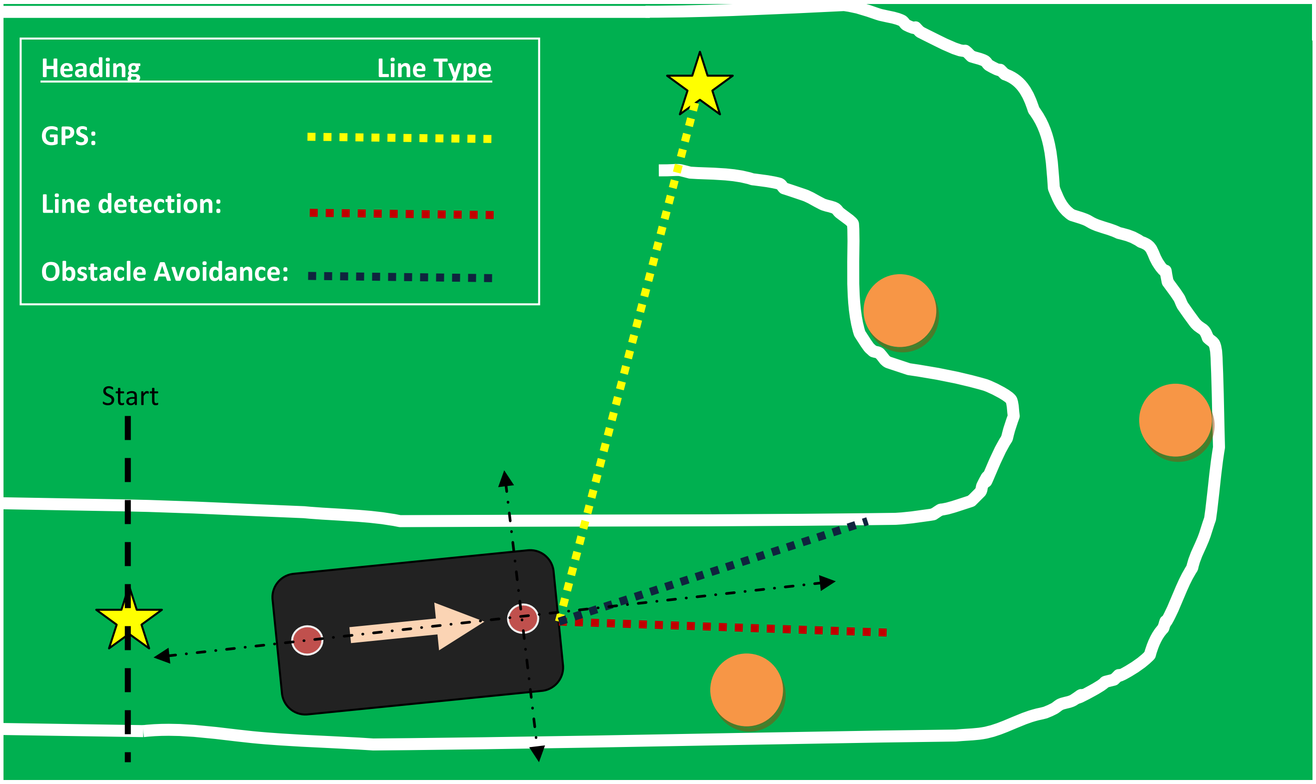


Figure 18: Model of sensor heading determination

Behavioral Coordinator

Our algorithm uses flags to determine when a behavior (line following, obstacle avoidance, or waypoint navigation) should be activated based off of sensor input. A subsumption network (shown below) handles the priority of each behavior and activates each when necessary.

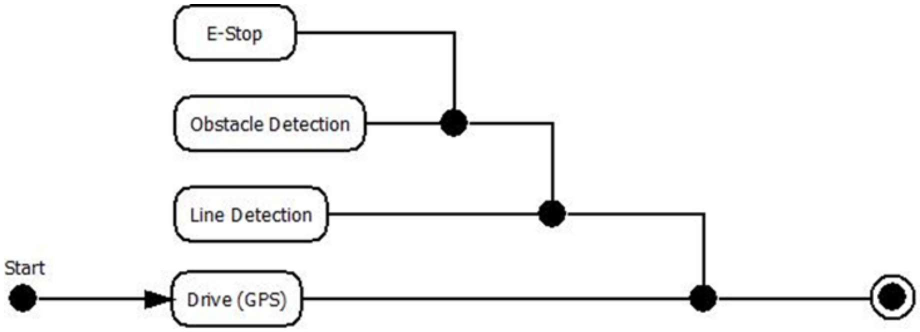


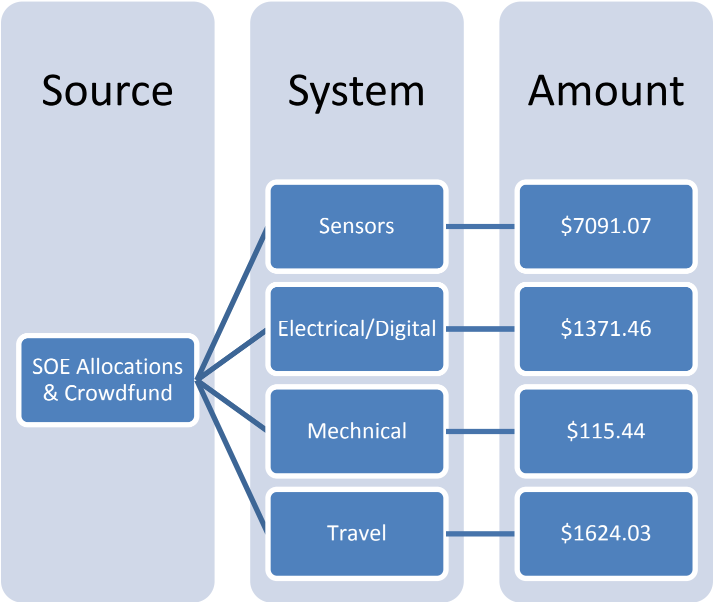
Figure 19: Subsumption network showing sensor priority

Mapping Techniques

Using a database, we store the GPS locations of each obstacle and line. The database also stores the portions of the path that the Roadrunner successfully traversed in order to attempt to recreate that path.

V. Cost

The overall monetary cost of this project was $10,202. These costs included parts, tools, and travel costs. This money was received by a student organizations allocation through the School of Engineering (SOE) and by a crowdfunding fundraiser.



Source	System	Amount
SOE Allocations & Crowdfund	Sensors	$7091.07
	Electrical/Digital	$1371.46
	Mechnical	$115.44
	Travel	$1624.03

Figure 20: Project monetary breakdown

On average, each student spent about 15-20 hours a week on this project throughout the 2014-2015 school years with variances during semester breaks. Therefore each student spent approximately 800 hours total on the Roadrunner project.

VI. Performance and Analysis

The battery life of the Roadrunner has been approximated to be a runtime of three hours with periodic motor control. The optimal range at which obstacles are detected has not yet been determined but has been tested at a range of 2-8 meters. Upon being determined to have entered into a dead end or trap, the Roadrunner will use a simple algorithm to backup to its previous position and try moving forward again with new sensor data.

References

1. Mead, R., J.B. Weinberg, J. Toennies, J.R. Croxell, B. Adams, G. Engel, J. Hiatt, N. Italiano, R. Krauss, A. Backs, and M. Gorlewicz (2008) “Road Runner: An Autonmous Vehicle for HRI Research”, in the Technical Report (WS-08-08) of *The 2008 Association for the Advancement of Artificial Intelligence (AAAI-08) Workshop on Mobile Robotics: Mobility and Manipulation*, Chicago, IL, July 2008, pp. 23-28.