WE Bots Project CAR

Competative Autonomus Racer

Darryl Murray, Shawn Wiegers, Colton Nicotera, Patrick Egan, Elizabeth Hollander, Andrew Simon, Vaughan Murphy

Engineering Department. The University of Western Ontario
London, Canada
webots@eng.uwo.ca

I. Introduction

Project CAR (Competitive Autonomous Racer) is an autonomous RC-sized car developed by the WE Bots robotics club at The University of Western Ontario. This project is the result of a team of students working outside of school hours. Project CAR is composed of several teams, each team building a component of the car and working together to integrate all components into this year's competition submission.

This year's submission focused on safety, modularity, and performance. Key challenges addressed include:

- High-speed vehicle localization
- High-speed vehicle control (acceleration and braking) on different surfaces
- Stoplight and roadway detection
- Collision avoidance with static objects along boundaries of course
- Collision avoidance with other competing robots

The result of these efforts is a fully autonomous vehicle capable of navigating a racecourse, avoiding other vehicles, and responding to traffic lights.

II. High-Level Design

Project CAR is composed of six different systems. Each system provides specific functions that power the racer.

Motor control powers our racer. Its main responsibility is to provide controlled vehicle acceleration, braking, and steering of the chassis propulsion hardware under varying surface conditions. As well, it is responsible for preventing unsafe maneuvers, and stopping the car in the event of danger.

The sensor array is the danger detection and the first half of the localization system. The array and its controller is responsible for collecting and processing the raw sensor data into usable measurements for motor control and the navigation algorithms, and warning of upcoming obstacles.

Computer Vision and Navigation forms the intelligence of the vehicle, and the second half of the localization system. It is responsible for determining the location of the car, navigating the roadway, and intelligently avoiding both static and dynamic obstacles. As well, it is responsible for the interpretation of traffic lights.

The communications backbone is core of the car. It is responsible for routing information quickly and reliably between the appropriate components – both on and off the car. In addition, the communications backbone is responsible for monitoring the health of all subsystems.

The electrical system powers all onboard components. It is responsible for the safe power distribution to all of the car's components, as well as the monitoring and protection of the car's batteries. In addition, the electrical system is responsible for powering-down the vehicle in the event of an emergency.

The chassis forms the substrate of the car. It is responsible for providing high-acceleration propulsion, wide-angle steering, and high traction. In addition, the chassis is securely holding all of the preceding car components, and protecting them from damage.

All systems have dedicated processors, and are separated by standardized connections and APIs, to enable rapid iteration and testing of each component individually. In addition, this modularity allows system failures to be contained to its respective system, enabling the remaining components to respond to the situation and ensure overall safety.

III. Chassis Design and Hardware

The chassis was made of two distinct sections: a purchased base and a manufactured frame. This combination of parts allows for a unique design that provides both reliability and customized functionality.

The base comes from an RC car – included in the RC base is a suspension system, steering system, and a mechanical drivetrain. Professionally manufactured systems such as these will allow the car to maintain control and traction at high speeds. The steering system is an Ackermann linkage that allows the steering wheels to be at different angles. This allows for true circular steering. Along with this, the differentials of the mechanical drivetrain provide more power to the outer wheels to allow for a faster turn. The power for the wheels comes from a single 2200kV brushless DC motor that provides 4-wheel drive through the drivetrain. Finally, the suspension system contains individual springs and dampers for each wheel allowing each wheel to maintain contact with the ground over rough terrain.

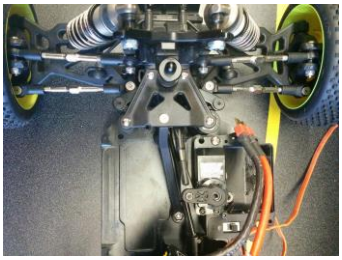

Figure 1 – Drive System Steering

The manufactured frame on top of the RC car base is constructed from bent aluminum sheet metal. Aluminum allows the added frame to be light, while providing safety and structural rigidity. This frame provides additional places to mount components within the car, along with additional safety measures to protect the car. These safety components include bumpers around the car, metal housing around the battery to protect it from impact, and compartments within the car to protect the other systems from the mechanical components.

IV. Vehicle Control and Surface Compensation

Both the driving motor and the steering motor are controlled through a dedicated microcontroller, which is receiving data from the other controllers in the robot. The dsPIC33FJ128MC802 (PIC) was chosen for this task because it has two pulse width modulation (PWM) modules to control the two motors, it has a CAN module to communicate with the other microcontrollers, and it has sufficient memory space.

In order to control braking, the microcontroller only uses one intensity. That is to say, the brakes are either on - at full - or off. This is due to the most-likely use cases of the brakes: in an emergency stop, and at the end of a race. For the emergency stop, full brake application is necessary in order to stop as soon as possible. Meanwhile, when the race is finished, the vehicle should stop soon after - applying full brakes will not have negative consequences. Lastly, having only one intensity results in a simpler design, which is less prone to errors.

In order to ensure high-speed vehicle control, Proportional Integral Derivative (PID) control is used in both the motor speed and the steering angle. Feedback comes from the sensor data, and the navigation algorithm sets the target speed and angle. The PID response will aim to be over-damped, set as close as possible to critically-damped.

The PID controller was tested on multiple different surfaces. The test conditions are designed to handle all use cases: no load (on a mount), indoor flooring, and outdoor pavement in dry conditions. After testing, PID values were optimized for each test condition. Additionally, the PIC checks the road surface (friction) and the requested angle/speed combination to make sure the values will not cause the robot to tip or roll over. In the event of a problematic angle/speed request, the steering angle is prioritized over the speed. The robot may move slower, but will still travel in the desired direction. This will assist other systems in collision avoidance.

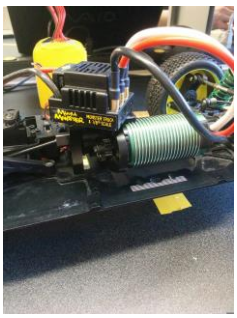

Figure 2 – Drive Motor

A model of the motor control was created in Matlab's Simulink add-on to test and anticipate the robot's response. A first-order linear model was assumed to provide a starting point in experimental analysis. The time constant (τ) was derived by testing the motor's time response to a step input, and measuring

the time at 63% of full response. To get a critically-damped response, the PID constants was tuned in Simulink.

V. Hall-Effect Encoders

Incremental encoders are installed in all four wheels of the car. By lining the inside of the wheels with magnetic strips and placing a hall-effect sensor in the hub of the wheel, the car can determine the rate of revolution of the wheels by detecting the change in magnetic polarity along the magnetic strips as they revolve with the wheels. This was done with an A1230 hall-effect sensor, and the raw data was processed and interpreted by an Arduino Mega 2560. After interpretation, the information was sent over the CAN network to the motor system for PID control and navigation systems.

VI. Power Distribution and Cutoff

A Rhino 4000 mAh 6-cell lithium polymer battery (Figure 3) powers Project CAR. This battery has a high power density, allowing long runtimes without excessive weight. A custom low power monitoring and shutdown circuit, controlled by a PIC16F917, monitors the battery. Each cell is checked to ensure that the voltage across each cell is within the recommended operating voltage of 3V – 4.2V. The circuit also tracks the total current supplied by the battery to ensure that it is less than 100A – the maximum recommended discharge rating. Readings are sent to a mobile app via the CAN network.

Figure 3 – Rhino 4000mAh 6-cell lithium polymer battery

To ensure that the monitoring system is low power, a switchable voltage divider was implemented (Figure 4). This allows the circuit to lower the cell voltage to a level that the controller can measure, while only consuming power when a measurement is in-progress.

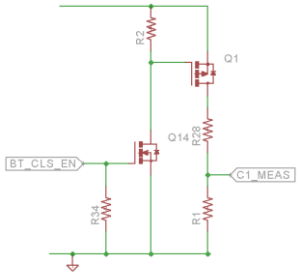

Figure 4 – Switching Voltage Divider

If a monitored value is outside the acceptable range, the controller disconnects the battery from the car by pulling up the MOSFET gate in the switching circuitry (Figure 5) to prevent further battery discharge. As a convenience, the switching circuitry also supports a power-on push button to connect the battery for easy use.

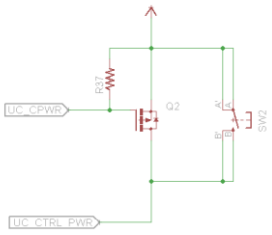

Figure 5 – Switching Circuitry

VII. Communications and System Health

The CAR is composed of multiple discrete microcontrollers, each in control of a separate system within the CAR. As the needs of these systems are widely varying (i.e. certain systems require lower latency, others require higher processing power) almost all individual systems are coded in different programming languages and environments. Due to this, a standard communications protocol would be required. Several were considered, but CAN stood out, as it is decentralized, reducing the possibility of one microcontroller being overwhelmed with requests.

A major goal for this year was to ensure the reliability and the safety of the car's communications system. The CAN protocol inherently provides bit checking, and the nature of the signal (two wires, initially both at 2.5V, but one pushed to 0V and the other pushed to 5V) provides protection against electrical interference. Satisfied with the hardware, the next step was to focus on software reliability. Heartbeat signals are sent out periodically to each microcontroller, ensuring that all systems remain both connected to the car and able to respond in a timely manner. A rolling average of these heartbeat response times is recorded, with the car entering an emergency stop mode if any one subsystem fails. The same applies to the Bluetooth connection between the mobile E-Stop application and the car.

VIII. Static and Dynamic Collision Sensors

In order to avoid obstacles, ultrasonic proximity sensors were arrayed around the perimeter of the car. A total of eight HCSR04 ultrasonic sensors were used to detect oncoming obstacles from all directions by situating two at each corner, facing perpendicularly from each other in alignment with the chassis walls. By sending out sonic waves and timing their flight time when rebounding from an object, the ultrasonic sensors determine how close obstacles are. The sensor data is interpreted by an Arduino Mega 2560, and sent over the CAN network to the navigation system for navigating around the detected obstacle.

IX. Vehicle Localization Sensors

Vehicle localization was accomplished using a variety of sensors to gather absolute and relative position data. The inertial measurement unit (IMU) and GPS were used in tandem to localize and assist with navigation in the car. Each sensor handled its own data collection, with their data interpreted by an Arduino Mega 2560. The IMU – a MPU 6050 – fed roll, pitch, and yaw data to the Arduino over I^2C, which was then passed to the motor control and navigation systems using the CAN network. The GPS – a NEO6MV2 – similarly fed longitudinal and latitudinal information to the Arduino over an I^2C connection, before being broadcast over the CAN network.

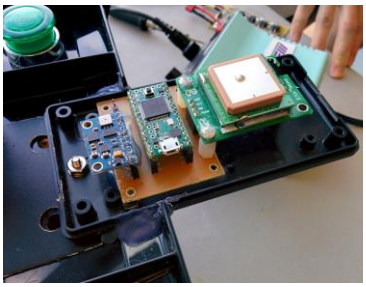

Figure 6 – IMU/GPS Development Setup

To determine the reliability of the GPS measurements for use in navigation, the GPS module was tested outdoors to measure the margin of error. It was found that the GPS' margin of error was too large to be used as the sole source of position data. The GPS data currently is used as a reference to notice gross errors in the car's localization, while the IMU data is used to assist in estimating the car's new location from previously determined positions.

In order to avoid leaving the track area, eight LM393 ambient light sensing photoresistors were used to interpret markings on the course. Arrayed in the same manner as the ultrasonic sensors, but pointed at the ground, the photoresistors change resistance based on the light reflecting off the pavement the car is running over. Because different colours reflect varying intensities of light, by measuring the voltage across the photoresistor, one can determine whether there are lines painted on the pavement, and use this knowledge to steer the car to remain between the course lines. All data interpretation for these devices was done through an Arduino Mega 2560, and was relayed using the CAN network to the navigation system.

X. Vision System Hardware

In this iteration of the Project CAR vision system, the hardware consisted of the Nvidia Jetson TX1, a Connect Tech Orbitty carrier board for the Jetson, and a Stereolabs ZED camera. The Nvidia Jetson TX1 provides a powerful embedded development platform, which is used to run the computer vision system in its entirety. The Jetson also allows for optimization of the computer vision algorithms through parallel processing using CUDA. The Orbitty carrier board reduced the footprint of the Jetson from a mITX motherboard to roughly a credit card, while still maintaining the necessary IO interface options. The ZED camera introduces a cost-efficient stereo vision camera with a pre-built SDK featuring camera calibration and passive depth mapping functionality, which is crucial to the navigation algorithms.

XI. Traffic Light Detection

Traffic light detection takes advantage of the change in brightness in the image as the lights transition. In detection, two images are subtracted from each other, giving an image of the differences between them. A low-pass filter removes noise, and an OpenCV circle detector looks for the bright circle of the traffic lights as their intensities change. A detected traffic light signal is then sent into the ROS network to start the configured navigation algorithm.

XII. Circuit Race Navigation System

The circuit-race navigation system has two components – line-based navigation, and pylon-based navigation. Line detection takes a simplistic approach to determining steering vectors and throttle values by following the right guiding line on the road. This is achieved by first retrieving pre-processed image frames and depth mapped frames from the ROS network. Each frame is then split into a set number of evenly distributed rows where each row is used to find a tangent line. Each tangent line is combined with the stereo camera depth map to determine the 3D real world points desired for navigating to. Next, these points are used to calculate a series of vectors, which are weighted and finally combined into a single navigation vector.

If the line-based navigation system fails to determine a navigation vector, the circuit race navigation will resort to pylon-based navigation. Since pylons are irregularly shaped objects and are not continuous like lines are, the problem of steering a car through them autonomously required a creative solution. To solve this problem, an algorithm was laid out in multiple steps. The first step is detecting the pylons in the image. To do this, the image was pre-processed, and the colour orange was filtered out to isolate the pylons. Once the pylons were isolated in the image, the approximate centers of the pylons were retrieved. The second step is to track the centers of the pylons as the car moves through them. Because detecting pylons is computationally expensive, it was decided that the pylons would be tracked using a tracking algorithm as they moved, and the pylons would only be re-detected occasionally. The third step is to form the steering vector. The stereo camera makes it possible to find the location of each pylon in 3D space. By forming a vector joining two pylons on the same side of the course, a steering vector could be obtained by projecting this vector on the ground plane. Finally, a throttle value was created by the sharpness of the steering vector. If the turn is sharper, the throttle value is decreased.

At this point, sensor data is used to make course and throttle adjustments, such as decreasing the throttle in response to an impending collision alert. This adjusted vector is then passed to the ROS network to be sent to the motor system via the CAN network.

XIII. Drag Race Navigation System

When the traffic light detector has detected the start of the drag race, the drag race navigation system instructs the motor system to proceed. GPS readings, IMU acceleration readings, and the accumulated Hall Effect sensor distance (based on the 60m known distance) are used to approximate when the car is approaching the end of the drag race, while IMU readings and ultrasonic collision detection is used to make course corrections to counteract car course drift.

When approaching the finish line, by using the known RGB value of magenta and tuning the individual values, a filter was determined. This filter is used to separate the magenta lines from the background and reduce noise when checking for magenta inside the region of interest. The throttle is reduced to zero when the magenta of the finish line is detected and then lost under the car. Ultrasonics are also used to apply the brakes before hitting the end barriers, as a fail-safe.

XIV. Safety and Testing

Project CAR uses a Bluetooth module connected to the communications microcontroller for emergency stop control, navigation mode selection, and the collection of diagnostic data via the Bluetooth-connected Android app. For redundancy, in addition to the communications heartbeat checks, an emergency stop button is hardwired into the power system in case the wireless E-Stop fails. In addition, both the power and motor systems are programmed to respond to emergency stop requests, to provide redundant stopping mechanisms should one of the systems fail to act. Lastly, both of these safety-critical systems also track received heartbeat requests from the CAN controller, and will shut down if a heartbeat has not been received recently. Both of these systems have written this code independently from the others, to avoid programming errors from crippling both systems' ability to respond.

To test the car reliably, a test rig was built so that the car could be run on a tabletop without moving. The test rig consists of rollers mounted on a board with wooden L-brackets placed at each side of the car (Figure 7). The rollers allow the

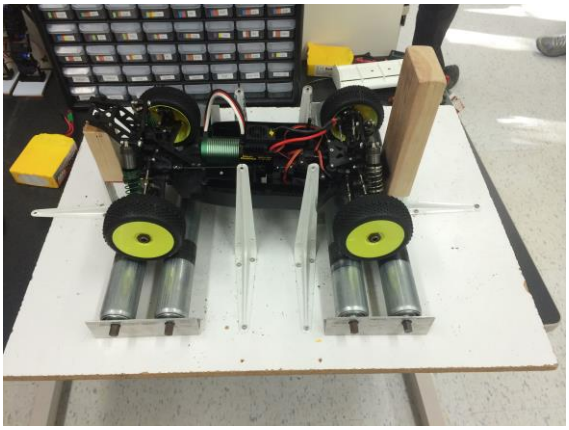

Figure 7 – Laboratory Test Fixture

car's wheels to spin as if it were on the road, while the brackets keep the car in place. This has allowed easy testing of the car while running in a controlled environment. For larger tests, a test track was set up to mimic the conditions of the real race.

XV. Summary

With distributed processing as our guiding principle, Project CAR has been working all year on this year's IARRC submission. This submission is the result of a team of students working outside of school hours to make each year a better year, and every new car iteration a better car. This year's submission focused on improving our safety, modularity, and performance, while getting an amazing learning opportunity for the WE Bots Project CAR team beyond the classroom. Designing systems from scratch offers a unique learning experience to everyone involved. We at WE Bots hope to continue and improve on the tradition of a modular, distributed processing system and excellent learning opportunities in the years to come.