NATCAR: Autonomous Vehicle Project

Student Team: Christian Setiawan - Electrical Engineering
Vikram Sai Kishan Sriram - Computer Engineering
Sankar Srinivasan - Electrical Engineering
Ashwin Sambathkumar - Electrical Engineering

UC DAVIS
UNIVERSITY OF CALIFORNIA
UC DAVIS
COLLEGE OF ENGINEERING

Hardware

Printed Circuit Board

- VNH5019A-E H-Bridge IC for DC Motor
- Used an OpenMV M7 Microcontroller to detect line and generate PWM signals
- Three separate ground planes to isolate high motor currents from the circuit
- Headers pins for Bluetooth, battery, DC motor, servo motor, and microcontroller

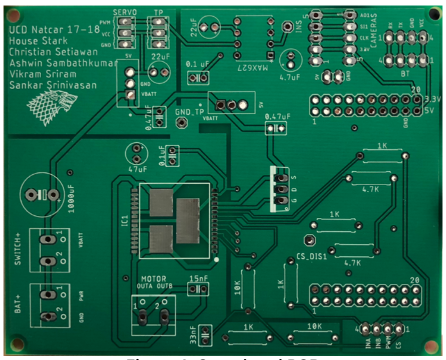

Figure 1. Completed PCB

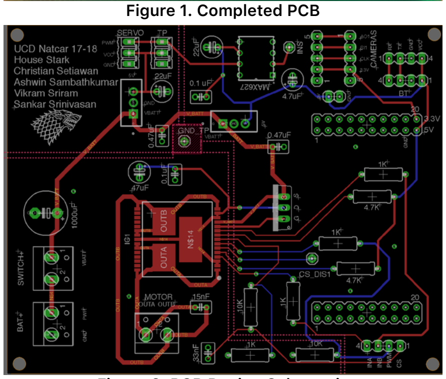

Figure 2. PCB Design Schematic

Car Construction

- Used a Schumacher SupaStox ATOM 1/12th GT12 Circuit Car Kit
- Mounted PCB onto car using PlexiGlass by drilling holes onto our chassis
- 3D printed a mount for our OpenMV M7 microcontroller in the ESDC

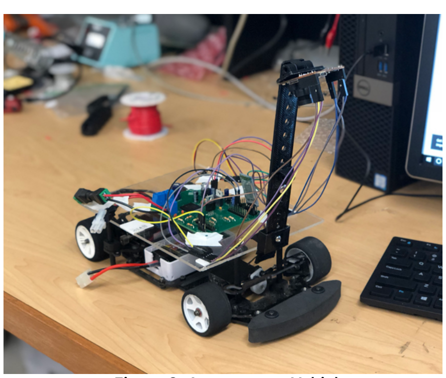

Figure 3. Autonomous Vehicle

Software

Steering Control

Algorithm 1 (Blob Detection):

- Used blob detection in order to read race track and anticipate new line curve by scanning and merging blobs in our Region Of Interest (ROI)
- Using these Blobs, we added up the centroid sum times the weight and divided by weighted sum to get the center position of the line.
- Then, using the center of the screen and this center value, we computed the deflection angle, which gave us an error function output that we could use for the PID control loop

Algorithm 2 (Linear Regression):

- Used a linear combination of the deflection angle from the line and the deflection angle from normal to line to set a desired output for PID
- Deflection angle from line forces the car to align with the line
- Deflection angle from normal forces car to intercept the line if straying too far from the line
- Normal deflection term is weighted by distance from line

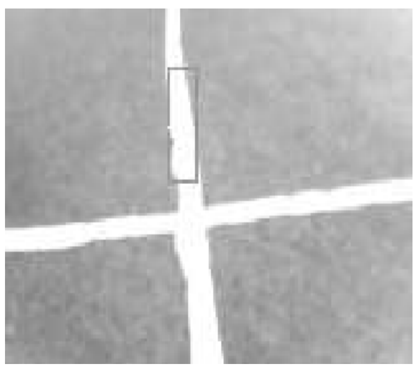

Figure 4. Camera Blob Detection

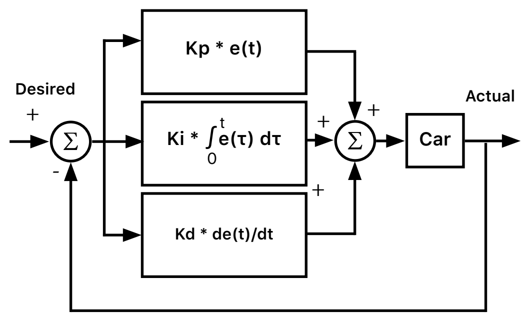

Figure 5. PID Steering Control System

Speed Control

- Utilized a Hall Effect sensor for speed detection by counting spinning tire revolutions
- Controlled the speed by calculating Error = (Speed from Hall Effect) - (Desired Speed)
- Desired speed is a linear function of the deflection angle constrained to our desired outputs at straights and curves
- Output was compared with the current speed to determine whether to brake or accelerate
- Accelerating was achieved by rapidly changing the pulse width percentage of the DC motor until desired RPS
- If the throttle error was low then the car would coast, if error was high then it would brake
- Breaking was done by reversing the INA and INB to put the car in reverse, this method ensured hard braking.

Bluetooth Connectivity (Debugging and Safety)

- Used an HC-05 Bluetooth module to wirelessly debug and tune our car
- Allowed us to easily change speed and PID constants while car was actively driving
- Able to implement a hard stop in case our car veered off course to prevent crashing into other individuals, other cars, or into the wall
- Connected using Blueterm 2 on Android and over Tera Term on Windows OS