Vision Based Autonomous Vehicle

Rami Awar, Tarek Tohme
American University of Beirut
Beirut, Lebanon
rba13@mail.aub.edu, tpt00@mail.aub.edu

I. Introduction

A. Problem Statement

The goal behind this robot is overcoming two challenges set by the 2nd Annual AUB Robotics Club Engineering Design Challenge. The first is navigating a road marked by lanes, and parking the car at some point. The second challenge is more evolved and consists of making deliveries to predefined locations on a predefined map. The map will be provided as a scaled image and nothing more.

II. Design Process

A. Requirements

- The robot should be able to travel at a speed of 0.4m/s to 1m/s, decided in reference to challenge maximum time and path lengths.
- The robot height should have around 3 cm ground clearance, decided in reference to wheel availability
- The robot should not cost more than 300$ in total, determined by our team's maximum willingness to invest.
- The robot's camera mount should be at least 15cm above the ground as to get a wide viewing angle.
- The robot should be able to carry extra loads up to 2.5kg more than its own weight.
- The robot length and width must be less than 30 cm to be cut out of already available plexi glass.

B. Drive Mechanism

We chose differential drive for its cost effectiveness and simplicity. The differential drive robot consists of two wheels mounted an a common axis controlled by two seperate motors. Two other supporting or caster wheels are mounted on a perpendicular axis for equal weight distribution and stability.

C. Selection of Motors and Wheels

To choose motors we need to get an estimate of the needed specifications. We chose 9cm diameter wheels to get a clearance of 3cm at least.

$$T = h2 + hk + k2T = h^2 + h \cdot k + k^2$$

$$RPM = \frac{60 \cdot speed}{\pi \cdot diameter} = \frac{60 * 0.5}{\pi * 0.09} = 110\ r.p.m.$$

We've got 2 motors, a total mass of around 4.5kg, and 4 wheels to divide the weight upon.

$$Weight \approx 45N = 2w_1 + 2w_2 \Rightarrow w_1 \approx 8N$$

Where w_1, w_2 are the weights acting on each drive wheel and each caster wheel, and more than half of the weight is on the caster wheels gives us the earlier result. Considering a low friction surface, a friction coefficient of $\mu = 0.85$ would be a safe assumption. Maximum torque is needed at the beginning of movement to overcome static friction, which can be calculated as follows:

$$T_{max} = \mu \cdot w_1 \cdot radius = 0.85 * 8 * 0.045 \approx 0.3\ N \cdot m$$

In summary,

- $Motor\ RPM = 110$
- $Motor\ torque = 0.3\ N \cdot m$
- $Wheel\ diameter = 9cm$

We found just the right motor for our specs, which is the FIT0441 brushless dc motor from DFRobot.

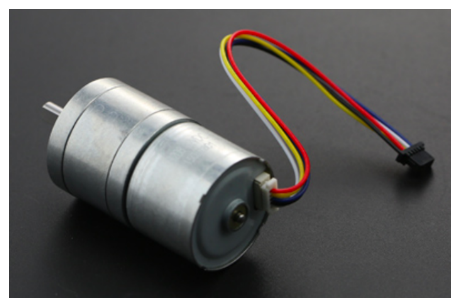

Fig. 1. FIT0441 Brushless DC Motor

It operates at 12V, has an internal encoder, and has the following specifications:

- Motor Rated Speed: $7100 - 7300 rpm$
- Motor Torque: $2.4\ kg \cdot cm \approx 0.24\ N \cdot m$
- Motor Shaft Speed: $159\ rpm$
- Reduction Ratio: $45 : 1$

Notice that the torque provided is less than that needed, and the speed is more than that required. For this adjustments for decreasing the maximum allowed weight were made, and no problems were faced whatsoever.

D. Chassis Design

Fig. 2. Advantage of cylindrical chassis

As a general shape, we chose a cylindrical chassis for its increased maneuverability and slightly easier obstacle avoidance as shown in Figure 2. To fit several components, we have gone with a stacking approach. Motor controllers, sensors, and PCBs are on the bottom-most layer, while the higher layers will hold more high level controllers such as a mini PCand a touchscreen, along with a camera mount. The design was prototyped in Autodesk Fusion 360, and iterated to preference. We added an extra layer to have ample storage space, for the current and future projects.

Fig. 3. Car CAD Model

E. Hardware Components

The following block diagram represents the connections between different parts of our robot. We will be using an arduino to read sensor inputs and control the motors, and a serial connection between the arduino and the raspberry pi to send and receive commands. The raspberry pi will be performing the image processing part to highlight goals and travel inside the provided path, as well as constantly map the traveled path and display sensor data and outputs on the screen.

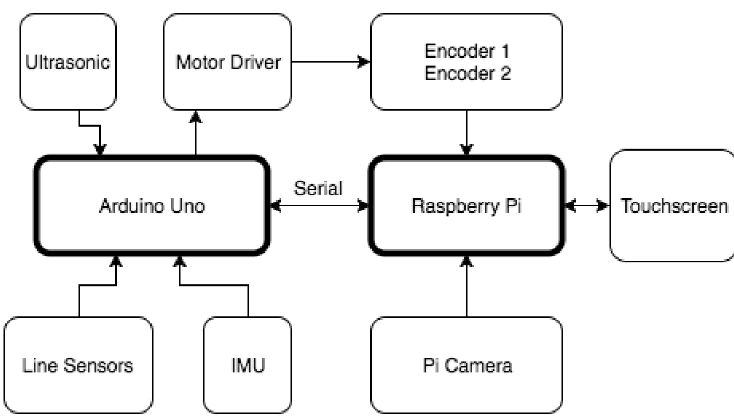

Fig. 4. Car Hardware Block Diagram

Maximum voltage and current distributions are listed in the following table.

TABLE I
CAR HARDWARE COMPONENTS

Components	Rated Voltage	Rated Current
DC Brushless Motors	12V	2A
Raspberry Pi 3	5V	1A
Raspberry Pi Cam	5V	500 mA
Touchscreen	5V	500 mA
Line Sensor	5V	10 mA
Ultrasonic Sensor	5V	20 mA
HC-05 BT Module	3.3V	10mA

With this knowledge we chose a 12V lithium polymer battery with 1000mAh and 10C discharge rate to power both motors, and a portable power bank with 10,000mAh and a 1A and 2A port to power the arduino, rapsberry pi, touchscreen and sensors.

III. SOFTWARE

A. Controller Description

The robot depends solely on computer vision, and this approach was chosen for future projects using this robot. A PI controller was chosen due to high noise caused by track glossiness, and tuned manually by testing (A derivative term was not tolerable due to the mentioned noise).

B. Lane Tracking Algorithm Iterations

As a first attempt, we started with a very primal algorithm that takes a large region of interest, extracts contours, and finds the center of gravity of all the contours combined through averaging. This puts weights on white areas and discards black areas. We chose our error to be how much this center of gravity is offset from the camera's center of view, and passed this error into a Kalman filter and the new smoothed value into our controller. The problem with this attempt was that the pi was unneccessarily processing the mid parts and edges of the image that carry no significant data.

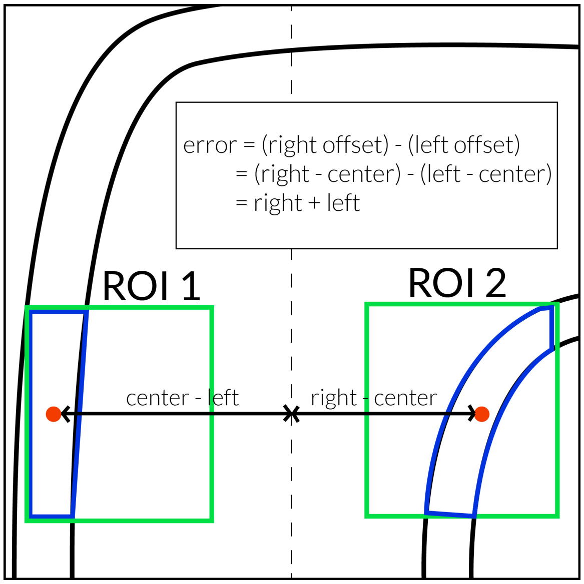

Fig. 5. Lane Tracking Error

As a second attempt, 2 regions of interest were chosen, symmetric with respect to the center of the view. With the same operations and some adjustments such as rejecting circular contours to account for specular highlights, the new error becomes (right offset)+(left offset) as demonstrated in the above Figure 5. This provides a lot more control since we have two inputs instead of one, and the number of possible states that can be detected increased exponentially. Actions such as maintaining the same distance from one lane alone when the other is missing become possible with such an arrangement. This algorithm seemed to be working perfectly on its own, with no sensors other than the pi-cam itself.
Although the most prevalent methods for lane tracking are based on line fitting such as the hough line transform or RANSAC, we decided to go with a less computationally expensive approach because we did not need more than that to solve our challenge. It would have been more optimal to implement such algorithms, but much more time consuming since neural networks would have to be trained to map the line angles to steering angles. Any other approach would not cover several cases and result in failure during the competition.

C. Parking Logic

To park the robot in the parking areas, we used an ultrasonic sensor to detect when an area becomes within view, and upon that turned 90 degrees in the corresponding direction and parked.

D. Delivering to Predefined Locations

Our delivery method consisted of mapping the track beforehand using odometry, and then using this map along with our encoders and IMU to localize ourselves within the track based solely on distances travelled and current heading. The brushless DC motors have built in quadrature encoders which can read off the pi, and the IMU calibration and filter were part of the Mahogany library. As to the automated delivery method, we found that it would be better to make the deliveries manually as we plan to use the robot for future projects, and implementing such a system will alter the topmost layer physically and permanently.

IV. Conclusion

In conclusion, building this robot was an introduction to computer vision and its capabilities from visual odometry to physical distance measurement which is all impressive when seen from the perspective of pixel manipulation. This modular robot will serve us in future projects that are more realistic than our current challenge. It was a fruitful experience overall, and we were glad to participate in this last EDC for us before becoming organizers!

1) G. Singhpannu, M. D. Ansari, and P. Gupta, Design and Implementation of Autonomous Car using Raspberry Pi, International Journal of Computer Applications, vol. 113, no. 9, pp. 2229, 2015.
2) M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, Junior: The Stanford Entry in the Urban Challenge, Springer Tracts in Advanced Robotics The DARPA Urban Challenge, pp. 91123, 2009.
3) A. Steinfeld, Interface lessons for fully and semi-autonomous mobile robots, IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, 2004.
4) W. Wang, S. Yang, Y. Li, and W. Ding, A rough vehicle distance measurement method using monocular vision and license plate, 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015.
5) R. Munguia and A. Grau, Monocular SLAM for Visual Odometry, 2007 IEEE International Symposium on Intelligent Signal Processing, 2007.