Launchpad McQuack Autonomous Driving

Team Members: Khurram Malik, Lucas Vanslette, Patrick Donelan

Table of Contents

Summary

Launchpad McQuack is Scrooge McDuck's personal driver/pilot, so our aim is to create an autonomous driving McQuack that would make Scrooge and Launchpad proud. Our base goal is to achieve lane tracking using the camera module on the robot. While Launchpad is known for crashing, our additional target goal is to avoid collisions by using the camera module for obstacle avoidance. We hope to have different avoidance schemes for different object scenarios. Our reach goal is to achieve autonomous reactions based on colors seen by the camera module. We plan to make use of the Duckietown simulation when we cannot meet in-person to test our ideas on the physical hardware. In terms of software, we plan on using python with ROS. As a group, we are most familiar with the python language and we are pursuing ROS over Robot Raconteur due to the greater quantity of online resources that we will be able to access for help. For ROS, we will be using the client-service model for data exchange.

Background and Motivation (include references)

LITEC was the first course to introduce the concept of control for many students. Many of the labs revolved around the use of a microcontroller car. The car could be programmed with PID control to follow a desired heading from an electronic compass, vary its speed based on the distance to an object read from an ultrasonic range sensor, etc. We believe lane tracking and obstacle avoidance through the use of computer vision is the next step up in the study of controls. With the addition of computer vision, the tasks that a robot can accomplish increases vastly. There is no reliance on the physical hardware that a robot has. For example, a line follower robot would require many sensors to accomplish its task, but a robot using computer vision will drastically reduce the number of sensors that are needed. Additionally, we are motivated to explore automated guided vehicles (AGVs) due to their increasing use in industries such as fulfillment (Amazon Robotics [1] is one such user of mobile robots). AGVs have existed in many forms over the decades, but recently with the advancements of computer vision, these robotics are more flexible than ever while at the same time requiring less infrastructural changes for them to work in a warehouse environment [2].

Goals (Duckiebot)

- **Baseline**: Lane tracking
- **Target**: Object avoidance
- **Reach**: Color-based actions

Approach

- **Step 1** : ROS Integration (C.a)
- Step 2 : Get output from camera (A.a-b)
- Step 3 : Basic user controlled movements (B.a)
- **Step 4** : Camera and Image Processing (A.c-d)
- Step 5 : Basic camera controlled movements (A & B.b)
- Step 6 : PID Control based on lane information (A & B.c)
- **Step 7** : Object Detection (A)
- Step 8 : React on object Detection (A & B.d)
- **Step 9** : Simple Actions with colors (A & B.e)
- Step 10 : Tie in obstacle avoidance with color detection. (A & B.d-e)
- Step 11 : Write Report

*__Bolded Steps are key milestones__

A. Camera and Image Processing
    a. Take snapshot using camera
    b. Stream video using camera
    c. Basic image recognition
        i. Ex01: blocked camera vs unblocked camera
    d. Image processing
        i. Obtaining measurements using camera
        ii. Thresholding, finding line contours, etc
        iii. Recognize different colors
B. Motor Control
    a. Basic user-controlled movements using keyboard
    b. Basic camera-controlled movements
        i. Ex01: move forward if camera is not blocked, stop if camera is blocked
    c. PID lane tracking
        i. PID control based on lane data received from camera
    d. React to object detection
        i. Ex01: stop immediately if an object is in the way
        ii. Ex02: slowly decelerate as distance to object decreases
        iii. Ex03: drive around object

iv. Ex04: drive around object with respect to constraints
   1. Constraints could be lane markings, other vehicles, pedestrians, etc

e. React to colors
   i. Ex01: stop if red, go if green
   ii. Ex02: add different obstacle avoidance schemes for different colored objects

C. ROS Integration
   a. Basic skeleton framework flow diagram
      i. Defining nodes
      ii. Labeled arrows between nodes for requests and responses

Roles

In order to maximize learning through this project, each team member will be completing tasks within the different categories. Limiting individuals to certain categories (i.e. having one person handle computer vision, one person handle motor control, etc.) may be more efficient, but it would prevent us from learning about all of the fields. However, if time becomes too heavy of a constraint, we may have to resort to individual-category pairing. Roles will naturally form as the project commences.

Preliminary Roles(Subject to change)*:

- Khurram
  - Step 1
  - TBD
- Patrick
  - Step 1
  - TBD
- Lucas
  - Step 1
  - TBD

***Note : Tasks are assigned in Trello Board (Linked below). Roles will be defined more clearly once the project difficulty is assessed.**

Steps and Timelines

Milestone Dates

1. Project Proposal: **9/24**
2. Decide on RR vs ROS: **9/24**
3. Pickup robot, track squares, and colored tape from lab: **9/25**
4. Assign Trello tasks and labels: **9/25**
5. Completion of Step 1: **9/27**
6. Completion of Step 4: **10/11**
7. Completion of Step 7: **11/1**
8. Completion of Step 9: **11/8**
9. Project Presentation: **12/7 - 12/9**
10. Project Report/video/software : **12/18**

Timeline

Weekending	Step Completion Goal	Actual
9/20	Project Planning	
9/27	Step 1	
10/4	Step 2 & 3	
10/11	Step 4	
10/18	Step 5 & 6	
10/25	Step 7 & Step 8	
11/1	Step 7 & 8	
11/8	Step 9	
11/15	Step 10	
11/22	Step 10	
11/29	Step 10	
12/6 * Presentation Due 12/7	Step 11	

Key Links

---

1. GitHub : https://github.com/kmalik97/Launchpad-McDuck
2. Trello Board : https://trello.com/b/ipIkX2sY/launchpad-mcquack
3. Duckiebot : https://www.duckietown.org/
4. Duckiebot simulation ROS and RR wrappers: https://github.com/burakaksoy/Gym-Duckietown-DuckiebotSimulation-RR-ROS-wrappers

References

---

1. US About Amazon. 2020. *What Robots Do (And Don't Do) At Amazon Fulfillment Centers*. [online] Available at: <https://www.aboutamazon.com/amazon-fulfillment/our-innovation/what-robots-do-and-dont-do-at-amazon-fulfillment-centers/> [Accessed 23 September 2020].
2. Elias, H. and Elias, H., 2020. *Why AGV Robots Are Taking Over Our Warehouse Floors*. [online] TechHQ. Available at: <https://techhq.com/2020/09/why-agv-robots-are-taking-over-our-warehouse-floors/> [Accessed 23 September 2020].