EE 483 Final Report — Group 11

Project: Avoiding Ducks and Other Obstacles
Members: Hamad Alkooheji · Tazwar Siddique · Hariz Razali

Project description

Autonomous vehicles need the ability to cross safely through environments, especially ones that contain unexpected static obstacles. In Duckietown, rubber ducks and other Duckiebots play the role of pedestrians and vehicles, and being able to safely avoid these obstacles is important to show reliable and safe autonomous driving. In this project the focus will be on specifically static vehicles that are positioned on the road as hurdles, and it will avoid and maintain its lane after avoiding the obstacle.

Our project uses both the lane keeping system developed on both labs 2 and 3 by integrating them with an obstacle perception as well as a closed loop avoidance controller. The Duckiebot uses its camera and Time of Flight range sensor to detect the obstacles that are placed in its front. When reaching the specified distance, the Duckiebot will then stop and execute an avoidance maneuver then recenter itself back to the center of the lane.

The main goal of this project is to develop a system that can perceive on lane hurdles and be able to safely steer away from them while remaining on the road. Along with qualitative demonstrations, we aimed for consistent detection in indoor lighting and safe behavior after each avoidance maneuver.

Implemented System

The system has three main parts: lane detection, obstacle detection, and the avoidance controller.

Lane detection

The camera node publishes the image from the front camera. The line_detector node resizes and then crops the image just enough for just the road to be visible**.** First the image is converted to HSV, and then two-color filters are applied, one for white lane marking and another for yellow lane markings. After the image filters are applied, Canny edge detection and a Hough transform are used to take out line segments. The resulting segments are then published to the Duckietown lane filter, which in turn calculates the lane pose (d, ϕ). The heading angle ϕ is used as a feedback signal for lane keeping.

Obstacle detection

For Detecting other vehicles, the system integrates the already existing Duckietown vehicle detection node, which looks for the circular pattern on the back of the Duckiebot and outputs a detection flag. In addition, a ToF sensor measures the distance to objects in front of the robot. When the detection is true and the ToF sensor calculates the distance to be within the chosen threshold which is 0.2 – 0.3 m the system then draws up the conclusion that there is a static vehicle blocking its path.

Avoidance controller

The avoidance_control node subscribes to the lane pose, the vehicle detection flag, and the ToF range, and sends velocity commands to the wheels. During normal behavior it runs a PID controller on ϕ to keep the robot centered in the lane. When a hurdle is detected, then node is then switched from the normal lane following operation to an avoidance sequence that is governed by a small state machine:

1. The robot first stops in front of the obstacle.
2. It then performs a left turn to move around the obstacle.
3. Next, it executes a right turn to return to the lane.
4. Finally, it resumes nominal lane-following control.

The PID gains and the duration of each state are specifically tuned so that the Duckiebot remains consistent and able to return to the lane without large fluctuations.

Participation of Team Members

Work on the project was divided so that each member took one main subsystem, all three members helped in both integration and testing:

- **Hamad Alkooheji** - Designed and implemented the avoidance controller, tuned the PID gains, and ran most of the experiments and data collection.
- **Tazwar Siddique** - Set up and tested the obstacle detection, integrated the vehicle detection node, and helped tune the controller.
- **Hariz Razali** - Set up the lane detection, tuned the HSV and Hough parameters, and handled ROS integration and launch files.

All three team members shared the responsibility for debugging, running test scenarios on the Duckietown track, and preparing the written report and presentation.

Results and Discussion

Lane Keeping Behaviour

The heading angle ϕ from the lane_filter_node/lane_pose displayed how effectively the Duckiebot was able to align itself with the lane. In early testing, ϕ was changed many times depending on how the robot was disturbed or turned sharply. After all the tuning the PID gains and the durations of both the TURN_LEFT and TURN_RIGHT state, ϕ stayed closer to zero which showed that the Duckiebot remained roughly centered in the lane. This indicated that Lab 3 lane-keeping controller can be extended with the additional state while remaining stable.

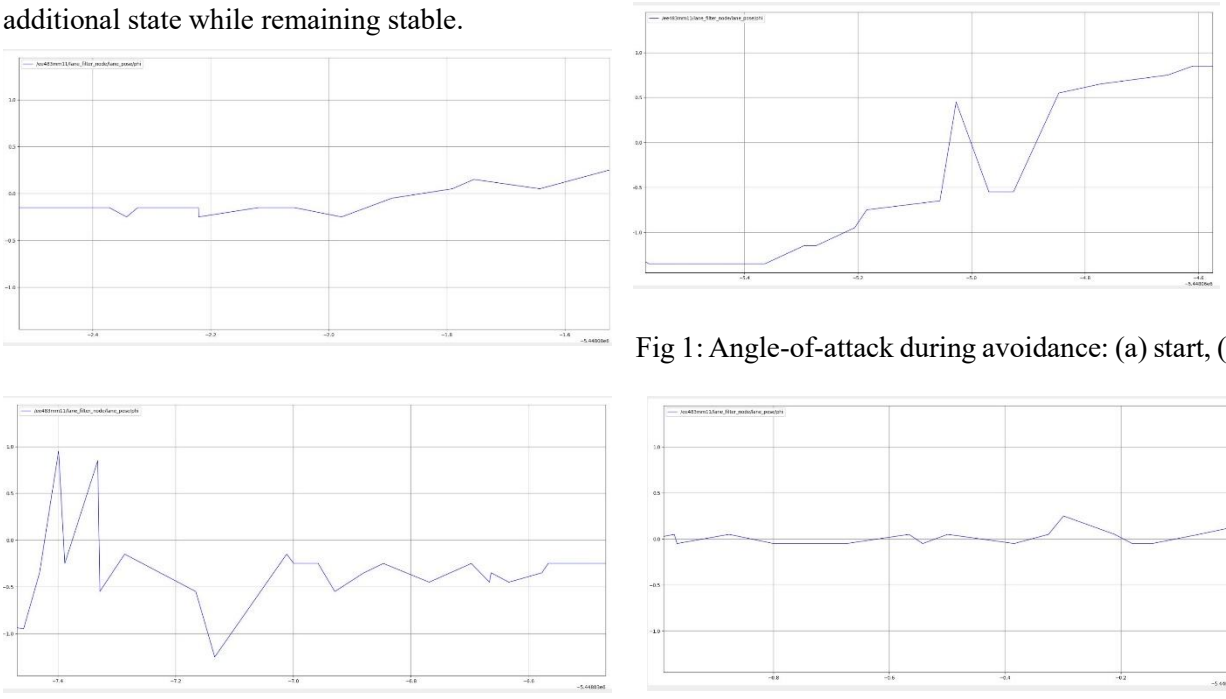

Fig 1: Angle-of-attack during avoidance: (a) start, (b) left turn, (c) right turn, (d) back to lane following.

Obstacle Distance and Avoidance

The ToF readings from the /front_center_tof_driver_node/range were used to measure the distance to a static obstacle in the lane. In the ToF distance plot the distance repeatedly decreased then jumped back up, which showed that the Duckiebot was overshooting the desired stopping point and then moving away again, which indicates ineffective open-loop avoidance.

With the final closed-loop controller, the distance quickly settled near 0.3 m and remained constant as the Duckiebot stopped at a safe distance, executed an avoidance maneuver, and then returned to lane following, although more testing is needed for different track layout and lighting conditions.

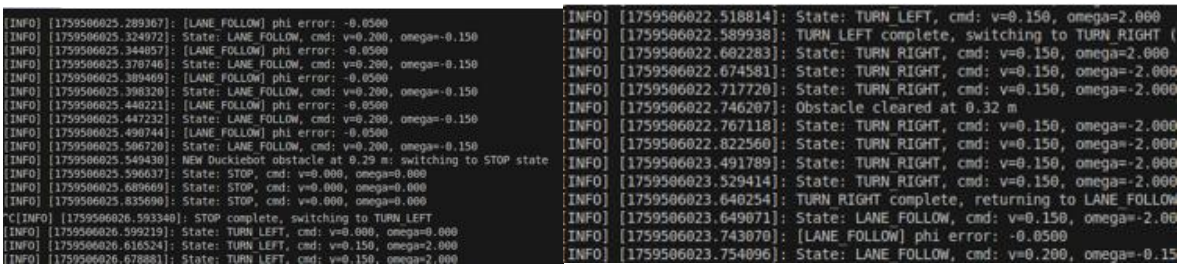

Fig 2: Terminal Output at Initial and Final Run Stage.

Assumptions

The project relied on the following main assumptions:

- **Static, single obstacle:** Only one duck or Duckiebot is in the lane at any time, and it remains stationary during the test.
- **Controlled environment:** All experiments are run on the standard Duckietown track indoors. Shadows, reflections, and outdoor lighting effects are not considered.
- **Flat and straight track:** The controller parameters are tuned for the standard flat Duckietown surface and may not work as well on different surfaces.

Future Work

Several improvements could make the system better and more robust:

- **Multiple and moving obstacles:** handle several ducks or Duckiebots at once
- **More complex tracks:** Test on curved lanes and at intersections, where the avoidance maneuver must be coordinated with turns.
- **Closed-Loop obstacle avoidance:** Change to a sensor-feedback-based control and include wheel odometry for better motion estimation.

Conclusion

In this project we implemented a Duckiebot system that can follow the lane, detect a static obstacle, and avoid it safely. We integrated the lane keeping from Labs 2 and 3 with Duckietown vehicle detection and the ToF range sensor to trigger avoidance when the robot is close to the obstacle. The avoidance controller uses a small FSM (STOP–TURN_LEFT–TURN_RIGHT–LANE_FOLLOW) and runs PID control during lane following to re-center after the maneuver. The results and figures show the robot stopping near the chosen distance, turning around the obstacle, and returning to stable lane following. Since testing was done on a controlled indoor track with a single static obstacle, the next step is to improve robustness for different lighting and track layouts and extend the system to multiple or moving obstacles.