CADD-E

CAmera Directed Driving Endeavor

Team Six: Alex Gisi, Musau Wambua, Justin Sizoo, Caden Kroonenberg, Joe Nordling

1 Synopsis

An autonomous golf cart capable of driving the cart path for 18 holes of golf with minimal input from the driver.

2 Architecture

Self-driving car technology has been the focus of intense research over the last three decades due to its potential to alleviate road congestion, increase passenger comfort, and make driving safer. While self-driving cars are still far from being common on the roads, the same technology can be effectively adapted to smaller vehicles like golf carts. Implementing self-driving technology on a golf cart obviates the most challenging parts of making self-driving cars, namely safety in urban areas and navigating poor weather conditions [1], while maintaining the goals of making transportation safer and more accessible.

Our design adapts the self-driving car architecture presented in [2]. We present our system architecture in Figure 1. At the high level, we distinguish between the perception and decision making systems. The perception system receives information via the cart's sensors (camera, radar, and GPS) in order to estimate the car's state and build a representation of its surroundings. The decision-making system plans and carries out actions via the car's actuators (throttle, brake, and steering) in order to navigate between a fixed initial point and destination while satisfying certain constraints, like passenger comfort and avoiding obstacles. Within those systems a number of subsystems can be identified.

2.1 Offline Map

Holds static information about the environment, including drivable routes and available behavior at intersections and in unstructured environments. We plan to use OpenStreetMaps for the offline maps, like Mercedes' Bertha self-driving car [3].

2.2 Localizer

Identifies the cart's location within the offline map using GPS data. Since sub-5cm precision is required for respecting cart path boundaries, we will use a GPS equiped with Real Time Kinematics (RTK) adjustment.

2.3 Collision Space Identifier

Identifies regions the cart should avoid based on analysis of present and past camera and radar data. A computer vision system to detect objects and do semantic segmentation will be required for obstacle avoidance and motion planning. Our current plan of action is to mount a front-facing camera to the golf cart in order to process the environment in front of the golf cart.

Our object detection model will likely piggyback off of an open-source model (likely a YOLO model) which may then be further trained as needed to suit our needs. After receiving the detections of the objects, we may then further classify them in order to get a better understanding of their orientation. From an object's class and orientation we will be able to predict potential areas that the object is likely move to in the future. This

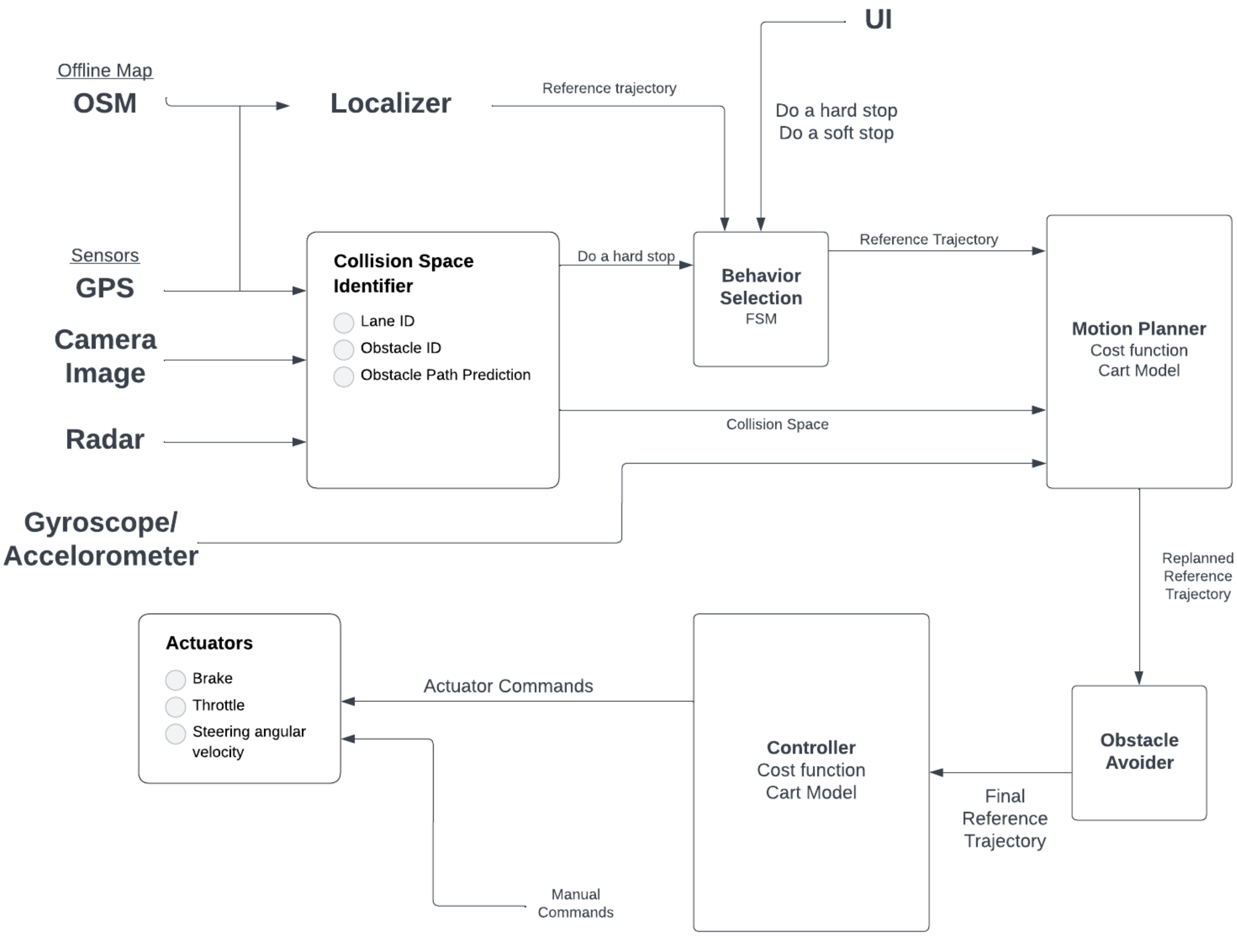

Figure 1: CADD-E Architecture

will allow us to stop when a person is walking across the cart path or slow down and pull over if another golf cart is headed towards our cart.

Semantic segmentation will be used in order to tell us where the cart path is located and more importantly if there are unexpected obstacles in the cart path. For example, if a tree branch has fallen in the path the segmentation model will understand that that branch is not the path. We may take this information and then proceed to avoid the obstacle.

2.4 Behavior Selector

Determines the currect driving behavior by choosing a reference path to follow and a goal speed. The reference path extends a few seconds ahead of the cart, and the goal speed is the desired speed at the end of the reference path. A trajectory is a path associated with a goal speed. The available reference trajectories are

- cartPath: Follows the cart path as stored in the offline maps. Computed offline by modeling the path.
- softStop: Pull off the path and slow to a stop. Useful for stopping at a tee box or to play a ball.
- hardStop: Stop on the path. Parametrized by the distance to stop. Useful for avoiding pop-up obstacles.

Usually, the reference path will be the pre-planned cart path stored in the offline maps. But if the user inputs a stop command in the user interface, for example, the behavior selector will switch to the softStop

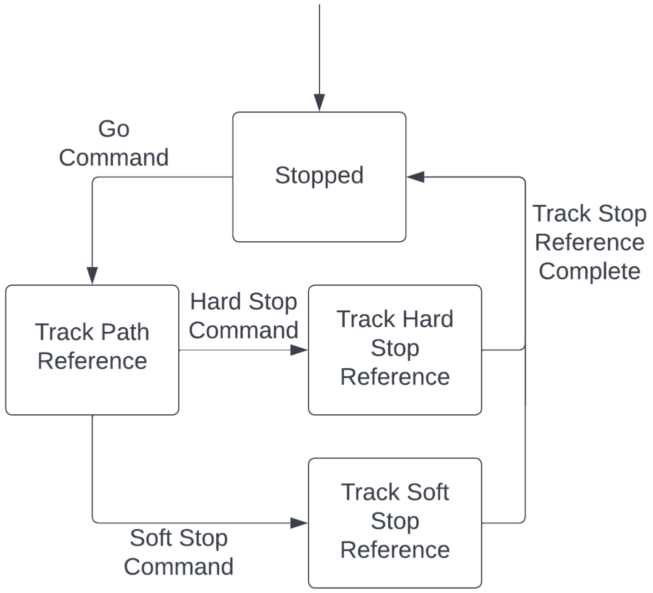

Figure 2: Finite State Machine for Cart Behavior

trajectory which pulls off the path and slows to a stop. Such a stop trajectory will be a pre-defined curve. Once the user wants to continue along the path, the behavior selector will switch back to following the path reference at the path speed limit. The behavior selector can be programmed as a finite state machine, for example in [4][5][3].

2.5 Motion Planner

Calculates an obstacle-free trajectory considering the reference path and desired speed which accounts for the cart's dynamic constraints and passenger comfort. We assume the motion planner receives a representation of the online map which encodes the (safety-expanded) location of obstacles. The motion planner adjusts the reference path to maintain a safe distance from obstacle regions. This problem reduces to a nonlinear optimization problem which can be solved by applying nonlinear model predictive control (NMPC). MPC is an optimal control method which uses a model of the plant, in this case the golf cart, to optimize control inputs in a receding horizon fashion. Due to the driving dynamics, the optimization problem is nonlinear, which increases the computational demands. Regardless, NMPC is well-suited for autonomous vehicle planning in real time, as shown in [6][7]. Significant MPC design decisions are the choice of plant model and cost function, which we will determine from literature review and simulation.

2.6 Obstacle Avoider

Slows the target speed to behave more cautiously around potential obstacles. Due to variations between the plant model and reality combined with real-time optimization constraints, the motion planner may compute a suboptimal trajectory with respect to the distance from potential obstacles. We will implement a final layer of obstacle avoidance to allow more cautious behavior by integrating probabilistic obstacle regions calculated by the collision space identifier. For example, we can simulate the potential paths of obstacles and slow down accordingly, as in [8], a test-car implementation which avoided all accidents in a year of operation.

2.7 Controller

Generates actuator (brake, throttle, steering) commands to implement a given trajectory. Again, the controller can be framed as an optimization problem and solved using an NMPC approach. By decomposing the MPC controllers into a high-level path replanning module (the motion planner) and a low-level path following module (the controller), we are basically following [7]'s hierarchical controller model, which demonstrated real-time performance on icy roads when implemented on a test car. The controller will use a higher-fidelity vehicle model than the motion planner to fully account for dynamic constraints.

2.8 User Interface

Lets the user start and stop the golf cart. The golf cart will feature an easily accessible touch screen that will serve as a User Interface. The UI will be a simple display split into two sections.

The left section will always provide any relevant information about the cart's motion such as vehicle speed. Below this telemetry data will be a set of buttons that are shown depending on the state of the vehicle. If the vehicle is currently stopped, the user will be presented with a large "Go" button followed by two buttons that allow the user to choose where they would like to go. Once the user has selected either the "Next TeeBox" button or the "Next Green" button, they can press the "Go" button and the cart will switch states and begin transit. Once the golf cart is in transit, the user will be presented with a simple "Stop" button. Once pressed, this button will apply the brakes until the cart is back to the stop state. This simple system of stopping the cart will be necessary in case user input is needed to stop the cart. Similarly, if the user needs to apply corrective steering, they can do this directly through the steering wheel rather than interacting with the UI.

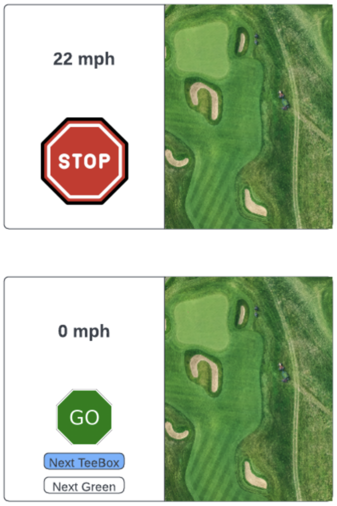

Figure 3: Potential UI design

The right section of the UI will display a map of the golf course that the cart is currently navigating. This map will provide the user with a simple view of how the cart will navigate to its next destination according to the online map of the cart.

References

[1] H. Surden and M.-A. Williams, "How self-driving cars work," *Available at SSRN 2784465*, 2016.

[2] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113 816, 2021.

[3] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, *et al.*, "Making bertha drive—an autonomous journey on a historic route," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[4] K. Jo, J. Kim, D. Kim, C. Jang, and M. Sunwoo, "Development of autonomous car—part ii: A case study on the implementation of an autonomous driving system based on distributed architecture," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 8, pp. 5119–5132, 2015.

[5] R. Guidolini, L. G. Scart, L. F. Jesus, V. B. Cardoso, C. Badue, and T. Oliveira-Santos, "Handling pedestrians in crosswalks using deep neural networks in the iara autonomous car," in *2018 international joint conference on neural networks (ijcnn)*, IEEE, 2018, pp. 1–8.

[6] M. A. Abbas, R. Milman, and J. M. Eklund, "Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles," *Canadian journal of electrical and computer engineering*, vol. 40, no. 1, pp. 12–22, 2017.

[7] Y. Gao, *Model predictive control for autonomous and semiautonomous vehicles*. University of California, Berkeley, 2014.

[8] R. Guidolini, C. Badue, M. Berger, L. de Paula Veronese, and A. F. De Souza, "A simple yet effective obstacle avoider for the iara autonomous car," in *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)*, IEEE, 2016, pp. 1914–1919.