Literature Review

Autonomous Golf Cart Project

Eduarda Farias

Florida Polytechnic University

June 2020

Contents

1 Abstract

Vehicle automation has become an inevitable phenomenon and increasingly close to a worldwide reality. Defined and divided into six different levels by the Society of Automotive Engineers (SAE), ranging from “no automation” (0) to “full automation”(5), driving automation and its applications are already available and are expected to have a significant presence in the future. Aiming to be a contribution to the development of this industry, the goal of the Autonomous Golf Cart Project is to modify a standard golf cart to implement a fully autonomous driving system. This document provides an overview of relevant articles and has the goal to compare the different autonomous driving systems designed, analyzing methods used, favorable outcomes and limitations.

2 ROS Implementation

Performed by a majority of the population on a daily basis, driving seems a reasonably simple task and the series of actions involved can be mindlessly disregarded. However, the complexity of this process becomes clear when trying to implement driving automation and the existence of a network that allows free and organized flow of relevant data through every part of the system unveils as an absolute necessity.

Among several operating systems, ROS (Robot Operating System) particularly excels in fulfilling this need and is frequently used in driving automation projects. ROS provides several services and is a “distributed framework” that enables not only the design of executables, inside nodes that can be grouped and divided into stacks and packages, but also their communication. In ”Autonomous Golf Cart Navigation Using ROS” [1] the operating system’s unique way of providing a nodal-based system, in which programmers can benefit from a high level of modularity, is described as a keen advantage. It allows parts to be developed and tested independently, making the process more efficient and less time consuming. In addition, Marin-Plaza, P., Ahmed, H., Martin, D., Moreno, J. & de la Escalera, A.[5] highlight the possibility of creating a network within the vehicle, as well as a network of different vehicles, as a reason for ROS-based architecture outperforming former works in autonomous vehicles.

A common trend observed, and explicitly defined in “ROS-based Architecture for Autonomous Intelligent Campus Automobile (iCab)”[5], consists of dividing the vehicle architecture and categorizing network nodes into three levels: high (deliberative), middle (sequencing or hybrid) and low (reactive). These layers are responsible for, respectively: path planning and decision making, sensor data processing and determination of which actions are required to execute a decision and, lastly, sensor data acquisition along with action execution. This segmentation contributes to an effective organization of the network, with a clear distinction of hierarchy and task delegation.

Also vital for ROS applicability and success, its language independence allows its implementation in several modern programming languages and an extensive number of possibilities when choosing or creating a higher level application and a graphical user interface (GUI). The GUI is the bridge between the driver and the vehicle system, commonly used to display relevant information as vehicle speed and map and route visualization. It can be used for passengers to choose their final destination and, during initial test phase, it can be particularly useful to display any information for further analysis and diagnosis. Java, Python (PyQt) and C++ implementations were observed in “ Autonomous Golf Cart” [3], “Autonomous Golf Cart Navigation Using ROS” [1] and “ROS-based Architecture for Autonomous Intelligent Campus Automobile (iCab)” [5], respectively.

3 Navigation

Two of the most imperative factors in allowing the completely autonomous operation of a vehicle are effective localization and navigation, in other words, the vehicle being capable of calculating and recognizing its position within the surrounding environment and capable of choosing and following a path, determining its subsequent steps based on predetermined and/or generated maps, its current localization and acquired sensor data.

To provide the vehicle with the necessary information, commonly used equipment includes LIDAR, stereo camera, GPS unit, wheel encoders, and inertial measurement units (IMUs). Data provided by GPS units is reported as insufficiently accurate, therefore creating a need for additional sensors, localization approaches or even navigation systems . In "Golf cart Prototype Development and Navigation Simulation Using ROS and Gazebo" [2], a RT3002 navigation system connected to the NI CompactRIO is used, publishing position and velocity values to the higher level controller application with an accuracy of two centimeters, for example.

The Particle Filter Algorithm is also frequently implemented, which consists of using odometry data to estimate a series of probable immediate vehicle positions while analyzing data from additional sensors, eliminating the positions which present any inconsistency when compared to the data acquired and providing a more precise assessment of the vehicle's location [1].

Along with the above mentioned information, the LIDAR (Light Detection and Ranging) sensor emits a series of light pulses (fanned into a maximum range of 180 degrees) that are reflected into the sensor housing after reaching any surrounding obstacles or objects. Using the time taken for each pulse to be reflected, the sensor calculates the distance traveled, detecting objects and enabling the creation of a real-time 3D map of the environment. Although it has a different operation approach, a stereo camera can be used with the same purpose, or added to the system in order to provide an even more factual analysis of the vehicle's surroundings. The map produced is then fed to the higher level nodes along with other relevant data.

The higher level nodes (also called deliberative nodes) are usually divided into global and local planners, respectively responsible for finding the optimal global route using a complete map of the environment and for modifying the chosen route locally, accounting for dynamic obstacles and the vehicle's constraints [4].

The Dijkstra method is frequently implemented in the global planner and is based on assigning values for the neighbor regions in a map (the farther the region, the larger is the value assigned) and calculating the shortest path respective to the minimum sum of values. For local planning, the Time Elastic Band method is used in "Golf cart Prototype Development and Navigation Simulation Using ROS and Gazebo" [2] and "Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles" [4], deforming the initial path chosen by the global planner to account for the vehicle's kinematic model and continuously updating the local path to avoid dynamic obstacles, being described as a good choice independently of minor model adjustments or path following inconsistencies. Additional methods mentioned are the Branch-and-Bound algorithm to choose the optimal global path and the Look-ahead Interpolation method to improve steering and avoid sharp turns.

4 Safety Systems

A feature present in a great majority of designs analyzed is a safety system, specially dedicated to manage emergency situations, usually allowing the driver (or passengers) to override the autonomous mode. An approach described involves the use of an electromagnetic clutch to enable and disable autonomous steering (not powered while in manual mode and cutting power to the steering servo) and kill switches along with impact sensors, with the purpose of cutting power to the engine and forcing a stop if pressed. Another safety strategy is to determine maximum velocities according to the vehicle's distance from surrounding objects and reverting the polarity of the motor, causing the vehicle to immediately slow down if an obstacle is identified within a close range [2]. Additional strategies include the use of sounds and visuals to alert the driver, and potentially passengers, of an emergency or malfunction.

5 Conclusion

The purpose of this review is to provide insight into current works in implementing driving automation. It is clear that ROS will be a keen tool in not only effectively integrating the vehicle's hardware and software, but also in programming and controlling it. In addition, it will be necessary to add several devices, including but not limited to servo motors, encoders, a LIDAR sensor, and impact sensors. To organize the node network and make the flow of data through the entire system more efficient, it is advisable to divide the vehicle architecture and the main nodes into high, middle and low levels as mentioned above. Similarly, the higher level path planner can also be divided into global and local planner and some of the algorithms and methods that can be implemented by these planners include Dijkstra, Time Elastic Band and Particle Filter. Lastly, it is imperative to have safety systems in place, therefore it is ideal to consider a design for the adaptation of the golf cart that does not compromise the possibility of manually driving the vehicle, so the driver can assume control in case of any emergencies or malfunction of the autonomous features.

6 References

1. Allender, D., Kuerhn, J., Miller, A., Wellman, B. & Young, J. (2011). Autonomous Golf Cart Navigation Using ROS. CalPoly.

2. Afanasyev, I., Magid, E., Matsuno, F., Sargitov, A. & Shimchik, I. (2016). Golf cart Prototype Development and Navigation Simulation Using ROS and Gazebo. MATEC Web of Conferences 75, 09005.

3. Gaynor, D., Latham, T., Anderson, I. & Johnson, C. (2013). Autonomous Golf Cart. ASEE North-Central Section Conference.

4. Marin-Plaza, P., Ahmed, H., Martin, D. & de la Escalera, A. (2018). Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles. Journal of Advanced Transportation, Volume 2018.

5. Marin-Plaza, P., Ahmed, H., Martin, D., Moreno, J. & de la Escalera, A. (2018). ROS-based Architecture for Autonomous Intelligent Campus Automobile (iCab). UNED Plasencia Revista de Investigación Universitaria, Vol. 12.