Self-Driving: A Theoretical Reality

Raghavendra Joshi
raghavendrajoshi5703@gmail.com, +91-9820921210

Abstract - **In an era where even art can be done by computers, why has automation in driving not been perfected? While there is no one line answer to that, this paper is an attempt at understanding the current automatic driving systems. We study Adaptive Cruise Control (ACC), which is a precursor to complete self-driving systems. To understand ACC, we look at how it evolved, and an implementation of the same. Various control strategies, which are an integral aspect of ACC, are also discussed. Lastly, it makes a case for developing a simulation to explore the limitations of the current automatic driving systems.**

Keywords - Self-driving, Adaptive Cruise Control, Control Strategies.

I. INTRODUCTION

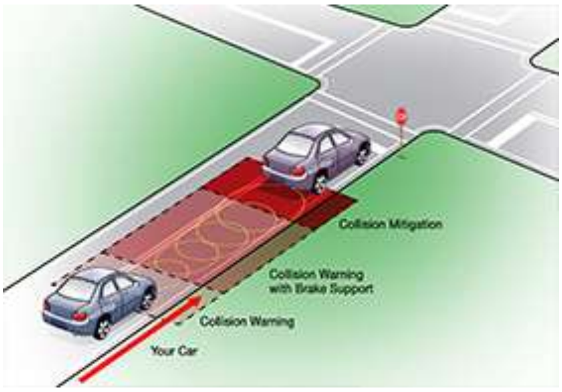
The advancement of technology has been a boon to almost every single industry around the globe, and beyond. The automobile industry is no exception. 'Getting from point A to point B with minimum effort', seems to be an apt way of summarizing the goal that the automobile industry strives to achieve. When humans decided that pushing two pedals to achieve enormous speeds in the matter of seconds was too much work, the need for automation in driving evolved. While the main goal remains just that, there have been a few precursors to self-driving cars. Adaptive Cruise Control is one of them. It is not the most sophisticated form of self-driving, but its simplicity and its potential make it interesting.

Adaptive Cruise Control (ACC) is the real-time distance and velocity management of a given car with respect to nearby entities. ACC can be classified into two categories - high and low-speed ACC. Both have their own set of problems. For instance, in high-speed ACC, distance management when there is sudden braking of the car in front at high speeds. Potential solutions include automatic breaking mechanisms and lane-changing algorithms using fuzzy logic. As for low-speed ACC, identifying vehicular and non-vehicular entities and reacting to them suitably is an interesting scenario. Here solutions range from better sensing techniques to automatic honking systems to even cessation of ACC in some cases.

Just like microprocessors function in 3 basic steps – fetch, decode, and execute, which are repeated until the program is completed. We can also reduce the functioning of Adaptive Cruise Control (ACC) to 3 basic steps – sense, compute, and execute. The various different sensors on the car sense the physical characteristics of nearby vehicles, barriers, and other such entities. The Adaptive Cruise Controller will compute this data, and give suitable commands to the accelerator, breaks, steering systems, indicators, etc. Each aspect of this discussed in detail further in this paper.

II. COLLISION WARNING SYSTEMS / COLLISION AWARENESS SYSTEMS AND DRIVER PREDICTION MODELS

[1] Before we had ACC, we had CWS / CAS. They work on a fundamental concept of 'critical distance' – the distance required to avoid collisions while maintaining a safety margin. This critical distance is calculated on the basis of relative distance, vehicle velocity, and relative velocity. When this critical distance is crossed by the driver, an indication is (like an alarm being sounded). The problem with this is that it could give false alarms, it does not calculate the risk of collision, and does not predict human behaviour. But, by using Kalman filters and Dynamic Bayesian networks, the future behaviour can be recognised on the basis of past observations. These proposed methods even provide a good stop behaviour estimation with less than standard deviations. Even today, cars like the Mercedes E200 use CAS-like systems.



III. ASIAN INSTITUTE OF TECHNOLOGY INTELLIGENT VEHICLE

[2] While ACC has been around since 1991, and has been implemented several times over the years, we are taking this example because of its academic nature, which makes it easier to study. In 2009, some researches from the Asian Institute of Technology, Thailand (AIT), successfully implemented ACC in an Intelligent vehicle.

Significant changes to the hardware of the car were made. For the acceleration system, they changed the traditional mechanical throttle-valve system by an electrical 'drive-by-wire' system, which uses a dc servo motor to control the throttle position. The braking was done by a Cool muscle dc servo motor, which was controlled by an ARM7 microcontroller via serial communication. A 'SICK LMS 291' distance sensor was also installed at the front bumper of the vehicle which operated on the lidar principle.

And as for the software, as discussed before, a ‘drive-by-wire’ system was installed for the throttle valve position control. This receives inputs from sensors and uses algorithms to determine the appropriate control outputs. This software is typically embedded in an Electronic Control Unit (ECU) (which in this case is the Adaptive Cruise Controller) that is connected to the various control systems throughout the vehicle. The use of drive-by-wire technology improves vehicle performance and safety, and enables the integration of advanced features such as adaptive cruise control, lane departure warning, and automatic emergency braking.

The ACC AIT developed has two control modes – distance control mode and velocity control mode. Distance control mode is operated when the host car finds obstacles in front of it, and velocity control mode is used when there are no obstacles. The distance control mode was implemented using the Mamdani’s fuzzy interface method. The velocity control mode was implemented using an ARM7 microcontroller with a proportional derivative (PD) algorithm to avoid the overshoot. Fuzzy logic control strategies are further explained in this paper.

IV. CONTROL STRATEGIES

[3] There are broadly 3 control strategies for ACC – PID feedback / feedforward control, Model Predictive Control, and Fuzzy Logic Control.

A. PID

PID (proportional-integral-derivative) control is a type of control strategy that can be used in adaptive cruise control (ACC) systems to regulate vehicle speed and maintain a safe distance from the vehicle in front. The basic idea behind PID control is to continuously measure the error between the desired speed and the actual speed, and use this error to adjust the control inputs (e.g., throttle or brake) in order to minimize the error. In an ACC system, a sensor such as a radar or lidar sensor is used to detect the distance and relative speed of the vehicle in front. The desired speed and the desired distance from the vehicle in front are set by the driver or by the ACC system itself. The error between the desired speed and the actual speed, and the desired distance and the actual distance, are then used as inputs to the PID controller. The proportional term of the PID controller compares the error to the setpoint and generates a correction that is proportional to the error. The integral term sums the error over time and generates a correction that is proportional to the accumulated error. The derivative term generates a correction that is proportional to the rate of change of the error. In summary, PID control is a control strategy that can be used in adaptive cruise control systems to regulate vehicle speed and maintain a safe distance from the vehicle in front. While the PID control strategy is a robust one that is easy to implement and tune, it is generally yielding suboptimal control in a real-life driving situation.

B. MPC

The Model Predictive Control (MPC) algorithm repeatedly solves an optimization problem that involves the prediction model, the performance criterion, and the constraints. The prediction model is based on the vehicle dynamics and the behaviour of the vehicle in front, and it is used to predict the future behaviour of the system over a finite time horizon. The performance criterion is typically based on the desired speed and the desired distance from the vehicle in front. The constraints are based on the physical limitations of the vehicle and the safety requirements. While MPC gives optimal control with great precision, it has a heavy computational burden and requires a certain level of future trip information which is not always the case.

C. FLC

Fuzzy Logic Control (FLC) is a control strategy that can be used in Adaptive Cruise Control (ACC) systems to regulate vehicle speed and maintain a safe distance from the vehicle in front. Fuzzy logic is a mathematical method for representing and manipulating uncertain or imprecise information. It uses a set of ‘fuzzy rules’ to map the inputs to the outputs, and a fuzzy inference engine to determine the optimal control inputs. It can handle uncertainty and imprecision in the inputs, handle multiple objectives and multiple constraints, and adapt to changing traffic conditions. The fuzzy inference engine uses a set of membership functions to represent the degree of membership of the inputs to various fuzzy sets (such as "close", "medium", or "far"). The membership functions are then used to adjust the vehicle speed and maintain a safe distance from the vehicle in front. While FLC does a good job at understanding and mimicking human behaviours, it still yields suboptimal control in a real-life driving situation, and has too many parameters (membership functions) to be efficiently updated in real time.

V. SENSING SYSTEMS

For ACC to work properly, real time sensing of the surrounding environment is very important. The most important sensor in an ACC system is the Distance sensing camera. Placed all around the car, or just in front (like the Audi A8), these cameras will get an accurate estimation of the distances, absolute and relative velocities and accelerations, of all the entities around the car. With the help of which, the Adaptive Cruise Controller will then generate a mapping of those entities, on the basis of which it will give commands. Object identification is not that important for ACC (it is for CACC). As long as we know at what speed an object is going, and precisely where that object is, ACC can function.

VI. PROBLEMS IN HIGH-SPEED ACC

What good is an ACC that does not have a lane-changing algorithm? When one is going on a highway at high speeds, and the car in front brakes suddenly, the traditional ACCs will simply break, causing the car behind to crash into it. The correct approach, however, is to break a little to make space, and to steer the car away from the line of collision. This can be done by the implementation of a lane-changing algorithm. We can make an efficient lane changing algorithm using fuzzy logic. Factors like distances from our car to the cars in front and behind, their speeds, and the distances and speeds of the cars in the destination lane will come into consideration. While lane changing algorithms exist in certain top-end cars, they are not perfect and are the cause of many complaints. More research can certainly be done here.

VII. PROBLEMS IN LOW-SPEED ACC

Low speed ACC gets much more complex. Under congested traffic situations, we need the car to perform a “stop and go” function. But because of the complexity of traffic patterns and even non-vehicular entities, earlier versions of ACC would perform a “stop and wait” function, in which the driver would have to resume the forward movement of the car when they felt appropriate. This was because car manufacturers were hesitant to offer a system that would resume on its own. While analysing a given traffic situation is possible, it is doing it in real time that is the problem. A well-designed microcontroller might not be enough, fast sensing techniques and execution are needed to support it. Cooperative Adaptive Cruise Control (CACC) [4] is a solution but it is not going to be a reality any time soon. ACC must first be perfected before CACC can become a reality.

VIII. CONCLUSION

We first concluded that just alerting the driver when something comes critically close to the car is not enough. The intention must also be enquired into. This can be done by Kalman filters and Dynamic Bayesian Networks. After looking at the AIT intelligent vehicle, we got to know that at least experimentally, ACC can be successfully executed using various control strategies which were also discussed (PID, MPC and FLC). However, we still do not have a perfectly implemented self-driving car available in the market today. The reasons causing this demand further research. This paper makes a case for a simulation of the various ACC control strategies to explore the limitations of each and hopefully arrive at a system that will give results optimal enough to be implemented safely in cars on a mass scale.

REFERENCES

[1] Kumagai, Toru & Akamatsu, Motoyuki. (2006), “Prediction of Human Driving Behavior Using Dynamic Bayesian Networks,” IEICE Transactions. E89-D. 857-860. 10.1093/ietisy/e89-d.2.857.

[2] W. Pananurak, S. Thanok and M. Parnichkun, "Adaptive cruise control for an intelligent vehicle," 2008 IEEE International Conference on Robotics and Biomimetics, Bangkok, Thailand, 2009, pp. 1794-1799, doi: 10.1109/ROBIO.2009.4913274.

[3] He, Yinglong & Ciuffo, Biagio & Zhou, Quan & Makridis, Michail & Mattas, Konstantinos & Li, Ji & Li, Ziyang & Yan, Fuwu & Xu, Hongming. (2019), “Adaptive Cruise Control Strategies Implemented on Experimental Vehicles: A Review,” IFAC-PapersOnLine. 52. 21-27. 10.1016/j.ifacol.2019.09.004.

[4] K. C. Dey et al., "A Review of Communication, Driver Characteristics, and Controls Aspects of Cooperative Adaptive Cruise Control (CACC)," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 2, pp. 491-509, Feb. 2016, doi: 10.1109/TITS.2015.2483063.