Safer Adaptive Cruise Control for Traffic Wave Dampening

Emily Baschab*
University of Alabama
efbaschab@crimson.ua.edu

Savannah Ball
Monmouth College
sball@monmouthcollege.edu

Audrey Vazzana
Rose-Hulman Institute of Technology
vazzanam@rose-hulman.edu

Jonathan Sprinkle
University of Arizona
sprinkjm@arizona.edu

ABSTRACT

This project aims to develop an adaptive cruise controller for vehicles at low speeds in stop-and-go traffic. Current adaptive cruise controllers can use RADAR sensors to follow a vehicle at high speeds (greater than 18 mph), but reach their limits if the lead vehicle's velocity dips below threshold, requiring the driver of the host vehicle to resume control over the car's speed. Some cruise controllers adapt to stop-and-go traffic, but these are mostly experimental and have yet to see widespread commercial implementation. These experimental models often have issues because of their limited data; consequently, the acceleration and deceleration can be jarring and uncomfortable to passengers. In contrast, because of our reliable sensor data, and the sensor configuration unique to the CAT Vehicle, our cruise controller will be capable of following cars at low speeds and functioning continuously, even when the car is stopped.

CCS CONCEPTS

• **Computer systems organization** → **Robotic autonomy**; • **Applied computing** → **Engineering**.

KEYWORDS

Adaptive Cruise Control, Vehicle Autonomy, stop-and-go

ACM Reference Format:
Emily Baschab, Savannah Ball, Audrey Vazzana, and Jonathan Sprinkle. 2021. Safer Adaptive Cruise Control for Traffic Wave Dampening. In *ACM/IEEE 12th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2021) (ICCPS '21), May 19–21, 2021, Nashville, TN, USA*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3450267.3452003

1 INTRODUCTION

Adaptive Cruise Control (ACC) works by taking in data from sensors and running that sensor data through an algorithm that calculates the optimal distance, velocity, and acceleration. Based on these optimal calculations, the host vehicle's computer can determine how much to increase the throttle, and what action the engine of the vehicle should take, simulating that same decision a driver makes in pressing either the throttle or the brake.

The current state of commercially available cruise control, though far more advanced than in the past, is still surprisingly limited. Adaptive cruise control in modern cars either does not operate under a certain speed threshold (less than 18 mph) or, if it does, operates poorly, causing a jarring experience for the driver and allowing cut-ins from other cars. This project aims to create a model which improves functionality under this threshold speed.

2 BACKGROUND

There are two general categories of spacing policies used in ACC. These are Constant Spacing Policies (CSP) and Variable Spacing Policies (VSP). Constant Spacing Policies are relatively simple: the desired distance is assigned as a designated constant. Variable Spacing Policies encompass any algorithm under which desired vehicle distance varies based on external and/or internal inputs. In most VSPs, a quantity known as Time Headway is utilized. Time Headway is defined as the time elapsed before the front bumper of the following car will pass the current position of the front bumper of the lead car at its current speed [3]. In many models, the idea of using Constant Time Headway (CTH), which means that the velocity is calculated to ensure the Time Headway remains at a constant value, is used.

Typical ACC spacing policies tend to work very well at high speeds where margins between cars are high and car behavior is predictable. However, at lower speeds, these techniques tend to result in a degradation in driving quality and experience for passengers.

2.1 The CAT Vehicle

This project utilizes the CAT Vehicle which is ideal for the testing of an ACC controller. The CAT vehicle testbed contains three types of simulated sensors: Front Laser Rangefinder, Velodyne LIDAR, and two side cameras mounted on the left and right side of the vehicle. The physical platform for the CAT Vehicle Hardware in Loop (HIL) is a Ford Hybrid Escape vehicle mounted with a Front Laser Rangefinder, Velodyne LIDAR, two side cameras and a GPS[1]. Further, the CAT vehicle testbed at University of Arizona provides a unique integrated simulation and physical platform for design, intensive testing, and real vehicle simulation to effectively verify and validate design ideas in a seamless workflow. It features transfer of controller design from simulated environment to physical platform without rewriting any component of the controller. The CAT Vehicle test bed comprises the Robot Operating System (ROS)

ICCPS '21, May 19–21, 2021, Nashville, TN, USA

ACM ISBN 978-1-4503-8353-0/21/05…$15.00
https://doi.org/10.1145/3450267.3452003

based simulator that runs on the ODE physics engine and the virtual environment using the Gazebo simulator to simulate vehicle to vehicle interaction and traffic like situations.

3 DESIGN DEVELOPMENT

To overcome the limitations of typical ACC, we created a model specifically designed to function at low speeds.

To do this, we created two distinct processes for computing the desired velocity for our follower car. The first process is the distance estimator and filter which interprets the raw data from the sensors and outputs relative distance and velocity of the lead car. The second process, the ACC model, takes these inputs and computes desired velocity.

The ACC model is specifically designed to adapt to low speed, stop-and-go traffic more responsively than commercially available models without compromising passenger safety.

4 FINAL DESIGN

4.1 Distance Estimator

Upon receiving LIDAR data from the CAT Vehicle sensors, the relative distance, velocity, and acceleration are calculated by a distance estimator program and published to a ROS node.

Because there are occasionally small inconsistencies in which part of the lead car is being measured by the sensors, it is necessary to make sure that these erroneous data points are filtered out. This is done by a Distance Filter which operates by the following logic.

$$d_{rel}(t-s) = \frac{d_{rel}(t) - d_{rel}(t-s)}{s} \tag{1}$$

Where s represents the step size (1/75)

4.2 Controller Theory

After being formatted by the Distance Estimator, the data is received by the ACC Controller which computes the desired speed.

In order to follow at an appropriate distance and velocity, we decided to split the domain of our speeds in two. The higher speeds would require the host vehicle to follow at a distance that would increase as the speed of the leading vehicle did. The lower speeds, at which the host vehicle could stop relatively quickly, would require that the host vehicle follow the lead vehicle while maintaining a constant distance far enough away to avoid a crash but close enough not to allow cut-ins.

Using this theory, the controller follows this logic to calculate the desired distance for the car

$$d_{des} = \begin{cases} av + \frac{l_{car}}{k} & \text{if } v \leq d_{rel} \\ C & \text{if } v > d_{rel} \end{cases} \tag{2}$$

where a is the realistic braking acceleration and k and C represent arbitrary constants. In the current model a = 1.7, k = 2, and C = 40.

Subtracting the desired distance from relative distance yields the quantity Distance Gain: the desired change in relative distance. The Distance Gain is added to the relative distance to get the new command velocity for the vehicle such that $v_{des} = 2d_{rel} - d_{des}$. This, in turn, is modified with a PID controller to determine the desired acceleration.

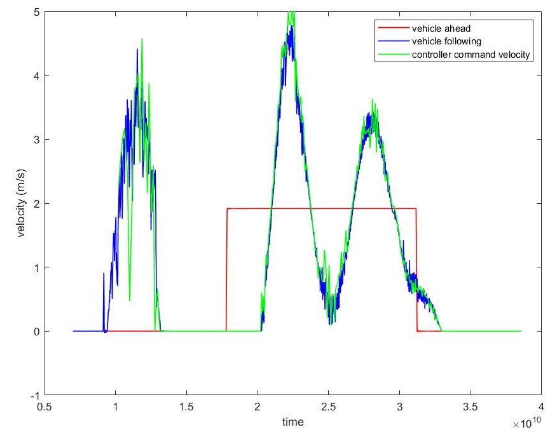

Figure 1: Plot of the lead and following car behavior from Simulink model data.

4.3 Verification and Testing

To test our data, before implementing it in the CAT Vehicle, we used ROS (Robot Operating System). ROS is ideal for this task because it can use previously stored and real-time data. To run our ROS simulations, we took previously gathered test data from oscillatory traffic [2]. This allowed us to eliminate one source of variability in our testing, thus making our results more consistent and reliable than they might be if different datasets were used each time. ROS is also readily integrable with our Simulink models, making it a convenient way to test adjustments to our model.

5 RESULTS

The results produced from Simulink revealed that the following car would emulate the velocity of the lead car while maintaining an acceptable distance at low speeds. A sample of the car's behavior can be found in Figure 1.

ACKNOWLEDGMENTS

This opportunity is supported by the National Science Foundation under Award IIS-1950359, the "CAT Vehicle Research Experience for Undergraduates." Additional thanks are due to the University of Arizona CAT Vehicle Program, especially Nancy Emptage, for making this research possible.

REFERENCES

[1] Rahul Bhadani, Jonathan Sprinkle, and M. Bunting. 2018. The CAT Vehicle Testbed: A Simulator with Hardware in the Loop for Autonomous Vehicle Applications. *2nd International Workshop on Safe Control of Autonomous Vehicles* (2018), 32–47. https://doi.org/10.1109/ACCESS.2020.2978244

[2] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, R. Haulcy, H. Pohlmann, F. Wu, B. Piccoli, B. Seibold, J. Sprinkle, and D. B. Work. 2017. The Arizona Ring Experiments Dataset (ARED). doi.org/10.15695/vudata.cee.2

[3] C. Wu, Z. Xu, Y. Liu, C. Fu, K. Li, and M. Hu. 2020. Spacing Policies for Adaptive Cruise Control: A Survey. *IEEE Access* 8 (2020), 50149–50162. https://doi.org/10.1109/ACCESS.2020.2978244