Vehicle Platooning using Adaptive Control

Vaidhyanathan Chandramouli
Bradley Dept. of Electrical and Computer Engineering
Virginia Tech
vaidhyanathan@vt.edu

Abstract— Vehicle platooning has emerged as a forward-thinking strategy to increase road capacity, enhance traffic efficiency, and reduce both fuel consumption and emissions through closely spaced vehicular formations. This paper outlines the development and simulation of a vehicle platooning system implemented using MATLAB Simulink. The core objective is the design of a robust control system aimed at maintaining stable inter-vehicle distances and ensuring safe maneuverability under various traffic conditions. Our methodology utilizes an iterative approach, adjusting control parameters to optimize system response and stability. Through a series of simulated scenarios, we assess the efficacy of our control algorithms, demonstrating their capability to adapt to different dynamic environments. The results showcase the significant potential of vehicle platooning within the realm of intelligent transportation systems, underscoring its benefits in real-world traffic scenarios.

Keywords **—** *Vehicle Platooning, Control, Adaptive, MATLAB, Simulink.*

1. INTRODUCTION

Vehicle platooning [1] is about smart cooperation between vehicles. By using technology that allows cars to share information in real time, platooning reduces the space needed between vehicles on the highway. This close formation cuts down on air resistance, which in turn reduces fuel consumption and emissions. It's a win-win: less gas used and smoother traffic flow.

But the benefits don't stop there. Platooning can make driving safer and more predictable by reducing human error, which is often the cause of accidents on the road. With automated systems managing the speed and spacing of the cars, we can expect fewer sudden stops and starts, which often lead to collisions [2][3].

However, perfecting vehicle platooning presents its own set of challenges. It requires a robust control system that can keep cars safely in line, no matter what the traffic conditions are. Plus, these systems need to communicate flawlessly in real time to adjust to quick changes on the road—like a sudden slowdown. In this paper, we're diving into how a vehicle platooning system can be designed and simulated using MATLAB Simulink. We focus on developing a control system that's reliable enough to handle different driving scenarios safely. Through our simulations, we'll show just how effective these systems can be, suggesting a not-so-distant future where vehicle platooning is part of our everyday driving experience.

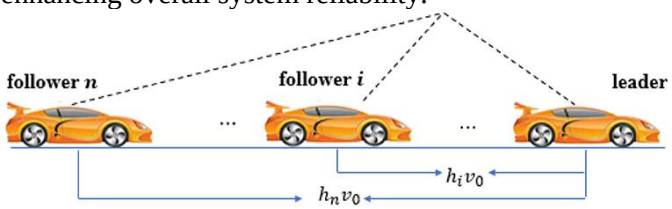
2. LITERATURE REVIEW

The concept of vehicle platooning has been explored for decades, primarily driven by the potential for improved traffic management and reduced energy consumption through enhanced aerodynamic properties[4]. As platooning technology has evolved from theoretical models to applied technologies, the focus has shifted towards practical implementation issues, especially those related to safety and system reliability.

This research[5] points out that while the benefits of platooning are well-documented, the safety concerns are profound, given the close operating distances between vehicles. The primary safety challenge in platooning revolves around maintaining system integrity during normal operations and in the presence of unexpected disturbances or component failures.

Effective communication protocols are essential for seamless vehicle interaction within platoons[6]. Additionally, the integration of sophisticated sensor-based detection technologies[7] plays a crucial role in monitoring and responding to dynamic driving environments. To ensure smooth operation and safety, the development of fault-tolerant adaptive control algorithms is critical. These algorithms manage acceleration, braking, and steering to maintain consistent platoon formations[8]. Moreover, employing failure prevention strategies such as Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) is vital. These methodologies systematically identify and mitigate

potential failure points within the platooning system[9], enhancing overall system reliability.



3. Methodology

The Vehicle Platooning System is designed to maintain a specified target velocity and spacing between multiple vehicles using a combination of velocity and spacing control strategies. The system uses a leader-follower architecture where the lead vehicle's acceleration is determined by a target velocity, and the following vehicles adjust their acceleration based on both the velocity error and the spacing error with respect to the vehicle ahead.

3.1. Vehicle dynamics control

Each vehicle in the platoon can be modelled using basic kinematic equations.

$$v(t) = dx/dt$$
$$a(t) = dv/dt$$

Where a(t), v(t) and x(t) are the acceleration, velocity and position of the vehicle at time t.

3.2 Velocity Control

The Velocity Control System (VCS) is responsible for maintaining the target velocity for each vehicle in the platoon. The system's transfer function, denoted as $G_v(s)$ processes the velocity error $e_v(s)$

$$e_v(s) = v_{tar}(s) - v(s)$$

where $v_{tar}(s)$ is the Laplace transform of the target velocity and $v(s)$ is the Laplace transform of the actual velocity.

$$G_v(s) = \frac{K_v}{s}$$

Here Kv represents the proportional gain of the controller which modulates vehicle's response to deviations from the target velocity.

The desired acceleration a_{des} required to necessarily align with the target is given by

$$a_{des,v} = K_v * (v_{tar} - v(t))$$

Where K_v is the velocity gain.

3.3 Spacing Control System

The Spacing Control System (SCS) ensures that a safe and consistent distance is maintained between consecutive vehicles. It employs a control law that adjusts vehicle acceleration based on the spacing error, which is the difference between the desired and the actual inter-vehicle distances.

The spacing error is given by

$$e_d(s) = D_{tar}(s) - D(s)$$
$$d_{err} = D_{tar} - D$$

Where $D_{tar}(s)$ and D(s) are the Laplace transforms of desired and actual spacing.

Transfer function that governs the vehicle's response to spacing errors is given by

$$G_{sc}(s) = K_{derr} + sK_{v_r}$$

In these equations, K_{derr} and K_{v_r} represent proportional and derivative gains respectively.

$$a_{des,s} = G_{sc}(s) * e_d(s)$$

This acceleration output prescribes the required changes in acceleration to maintain the desired spacing, taking into account both the magnitude of the spacing error and its rate of change.

3.4 Integration and Adaptive Control

The overall dynamics of each vehicle are influenced by these control systems through the integration of their outputs. The combined acceleration command A(s) that inputs into the vehicle's dynamic model is the sum of the outputs from both control systems

$$A(s) = a_{des,v}(s) + a_{des,sc}(s)$$

The vehicle's actual Velocity V(s) is then obtained by the integral of combined acceleration A(s) and the vehicle's inherent dynamics as given by the Vehicle gain $G_{vehicle}(s)$.

$$G_{vehicle(s)} = \frac{2}{(s+2)}$$

(τ the time constant which is usually used in modelling vehicle gain is assumed to be equal to 2.

Overall vehicle dynamics given velocity is

$$V(s) = G_{vehicle}(s) * A(s)$$

4. System Modelling using Simulink

In order to model this above described system and run the algorithms as a proof of concept, we will be using MATLAB and Simulink to model this vehicle platooning system.

In Simulink, each component of the vehicle system—such as velocity and spacing control systems—can be modeled as distinct blocks, with transfer functions

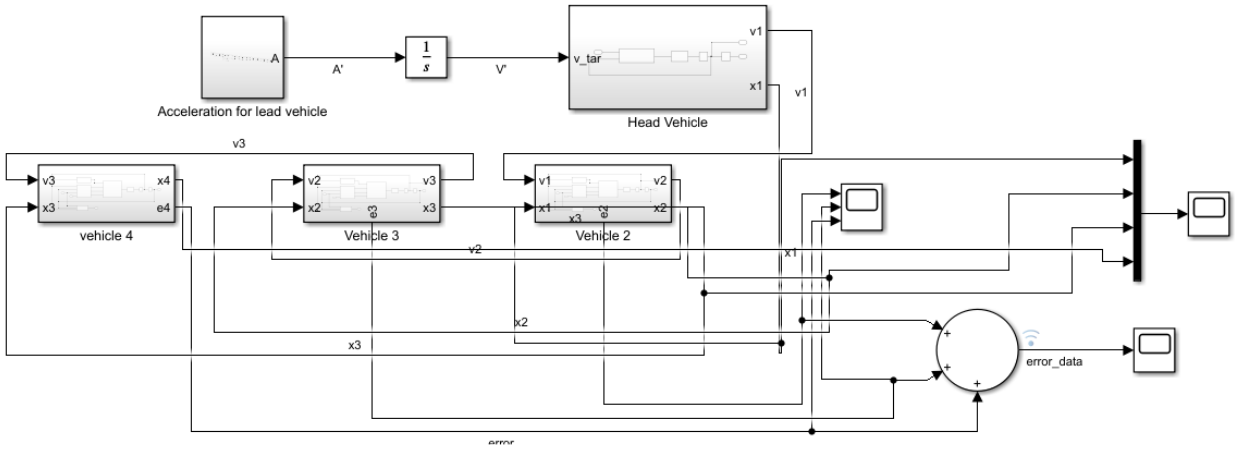


Figure 2: Vehicle Platooning System Design

$G_v(s)$nd $G_{sc}(s)$represented using the Transfer Function blocks. These can be straightforwardly connected to other system components that simulate vehicle dynamics, which might include integrating blocks or custom S-function blocks to represent the vehicle's response to acceleration commands, $G_{vehicle}(s)$.

The below block diagram represents the system modelling of the follower vehicles with a vehicle control block along with a spacing control block integrated with the vehicle dynamic function $G_{vehicle}(s)$.

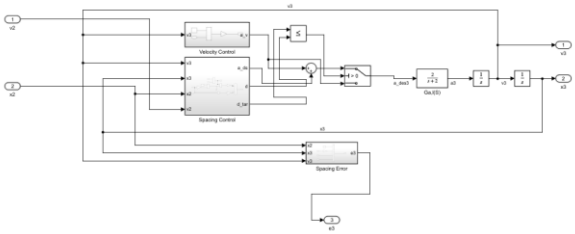


Figure 3: Block diagram for a follower truck

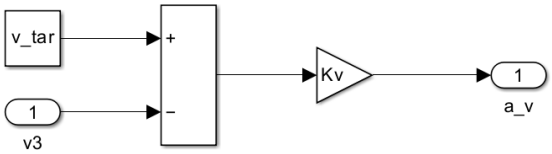


Figure 4: Block diagram for velocity control

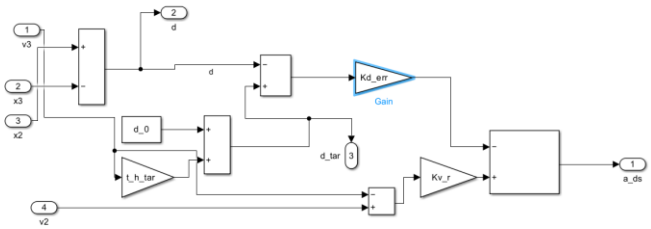


Figure 5: Block diagram for spacing control

Given that we have modelled the system in simulink as given by these diagrams, we can now analyze the system model through different experiments.

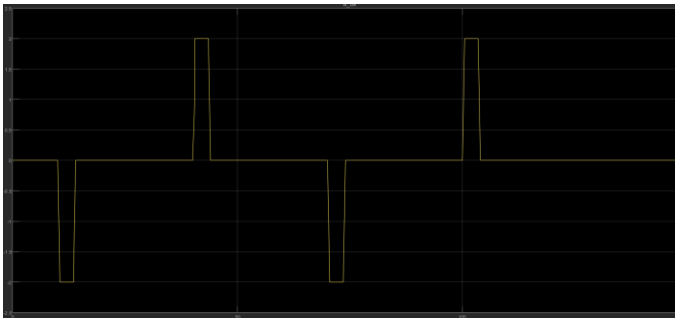
5. Assumptions and Experiment Parameters

We will be using 4 trucks with a leader-follower system architecture as discussed in the methodology. The leading truck would deal with the changes with acceleration and the other trucks adjust to these abrupt changes manually.

The trucks start of 50 meters apart from each other and have varying velocities of 20, 24, 28, and 27 meters per second respectively(first is the leading truck). The target velocity is set to 20 for this experiment.

Though these will be the parameters for discussing most of the experiments, We will also include one graph to see how the system varies if the initial velocities differ.

The acceleration given on to the lead vehicle is given by some combination of step and ramp functions and the output looks like this. The acceleration fluctuates between -5 and 5 in the interval of 0 to 150 seconds.



These are the target values for velocities and distance respectively.

$$V_{tar} = 32m/s, d_{tar} = 8m,$$

The gain parameters are given below and we will be varying them to fine tune to find the right parameter.

$$K_{v_r} = 0.7, K_{d_{err}} = 0.5\ , K_v = 0.5,$$

6. Results

6.1 Distance vs time for the trucks

Position of each truck with respect to time for each truck, the trucks start around 50m apart and never intersect despite the wavy nature of the graph. That is where the adaptive control kicks in.

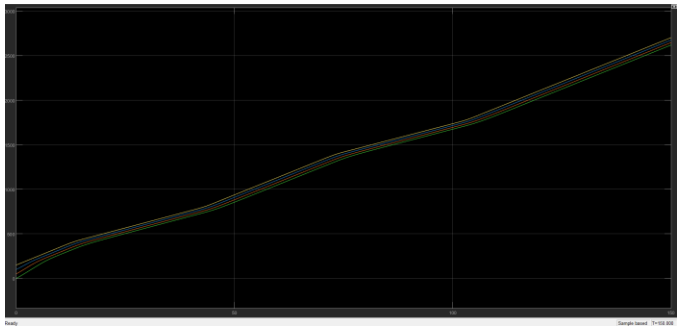


Figure 7: Position of trucks vs time

6.2 Spacing error vs time

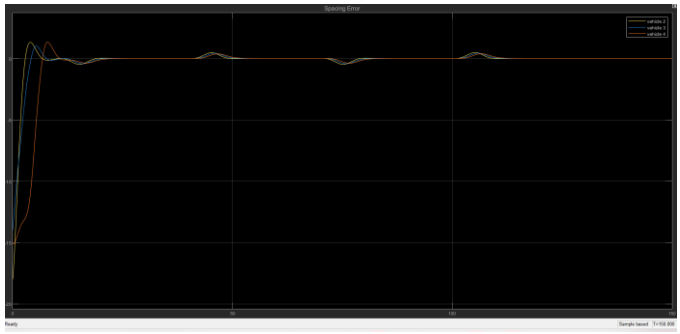


Figure 8: Spacing error over time

The spacing error as defined in section 3.3 is plotted over time. Initially the system tries to offset the relatively huge difference but as time goes on, the model adapts to the rapid changes and quickly its velocity.

6.3 Velocity over time

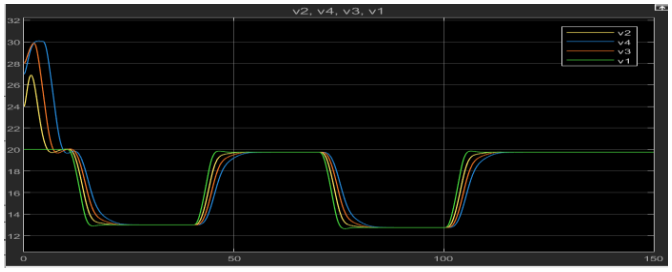


Figure 9: Velocity of Trucks over time

6.4 Hypertuning of parameters: K_v

A value we will be observing over the next few results is the sum_error. This captures the sum of error of all the trucks and they are plotted over time for all different simulations with different parameter values.

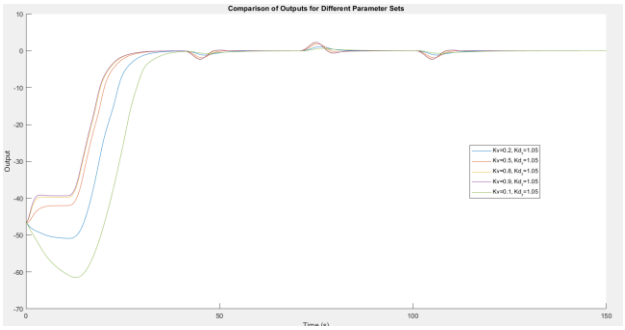


Figure 10: Hyper parameter tuning for K_v

Analysis of $K_v$is interesting since there is no one correct value to assign to the system. As K_v modulates the velocity error of the controller, the higher values of the K_v, tend to converge quickly and reduce the d_{err}. However for subsequent values they tend to perform worse than the lower values of K_v. The right value needed to be assigned to $K_v$depends on the use-case and priority, though we will be taking a value in the middle K_v = 0.8, for further analysis. Having a dynamic $K_v$value could be a matter of future research.

6.5 Hypertuning of parameters: K_{v_r}

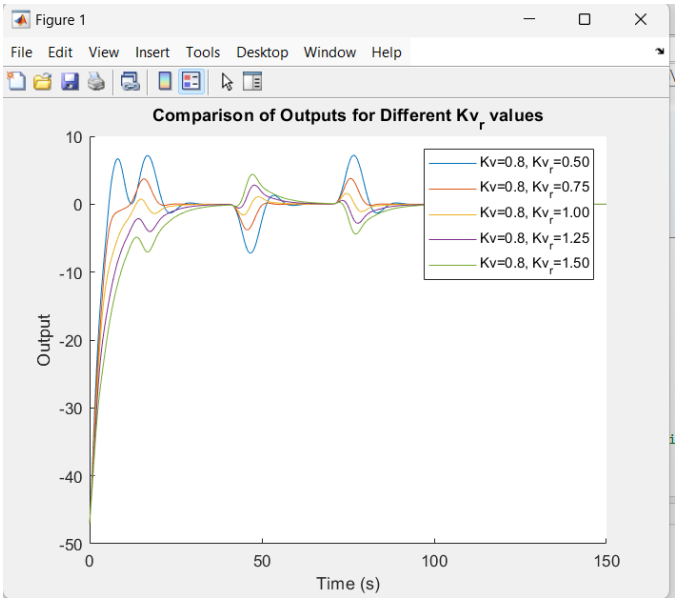


Figure 11: Hyper parameter tuning for K_{v_r}

Keeping the K_v as 0.8, and finding the optimal parameter for K_{v_r}, we find that the proportional gain K_{v_r} achieves desired values for K_{v_r} values around 1.0

6.6 Hypertuning of parameters: $K_{d_{err}}$

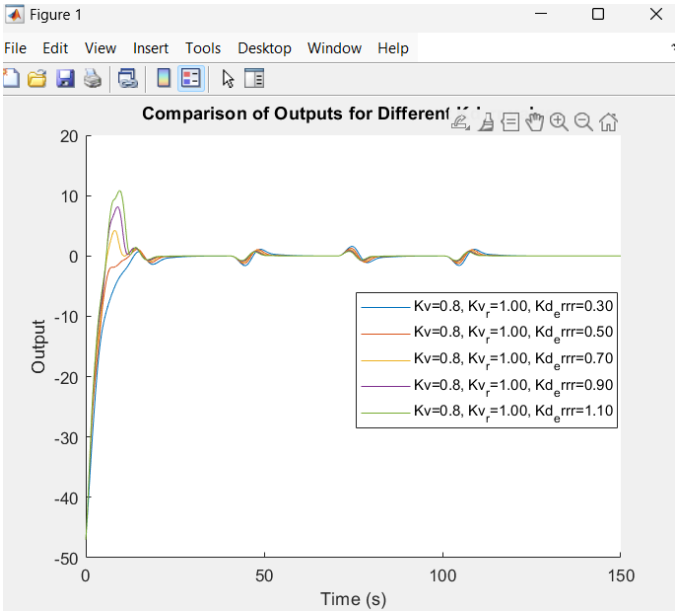


Figure 12: Hyper parameter tuning for $K_{d_{err}}$

As the curve suggests fastest convergence is suggested by proportional spacing error $K_{d_{err}}$= 0.3. As it is visible, the graph for the tuned parameters $K_{d_{err}}$ = 0.3, K_v = 0.8, K_{v_r} = 1.0 result in the minimum deviation from the suggested spacing error and maintains the distance regardless of perturbations to the lead vehicle.

6.7: Error for different initial velocities

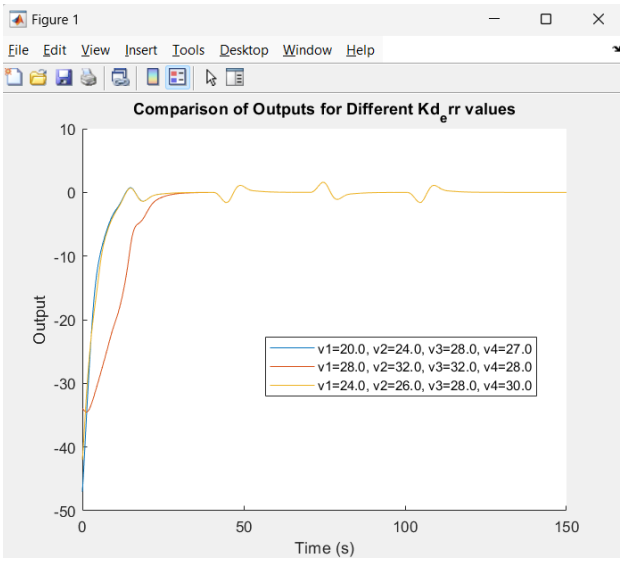


Figure 13: Spacing error vs time for different initial velocities

7. Conclusion

In conclusion, this project has successfully demonstrated the application of adaptive control strategies for vehicle platooning using MATLAB Simulink. The focus was on developing a reliable control system capable of maintaining consistent inter-vehicle distances and managing the collective dynamics of a vehicle platoon under various simulated traffic conditions. Through iterative adjustments and optimizations of control parameters the project illustrated the potential of adaptive control systems to enhance vehicular safety and efficiency within a controlled environment.

8. Future Work

We have modelled the unpredictability only in the lead truck and not in the trucks that follow it. Perhaps a good future direction would be to introduce unpredictability in all parts of the algorithm including the position and the velocity of the vehicles. One other tangent to go would be to properly optimize the control variables and propose an optimization algorithm to analyze the answers.

9. References

[1] Shladover, S. E., Su, D., & Lu, X.-Y. (2015). Impacts of Cooperative Adaptive Cruise Control on Freeway Traffic Flow. Transportation Research Record, 2489(1), 63-70

[2] Bergenhem, C., Shladover, S., Coelingh, E., Englund, C., & Tsugawa, S. (2012). Overview of Platooning Systems. In Proceedings of the 19th ITS World Congress, Vienna, Austria

[3] SAE International. (2017). Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles.

[4] Thrun, Sebastian (2010). "Toward Robotic Cars". Communications of the ACM. 53 (4): 99–106. K. Elissa

[5] Silvia et. Al. Do adaptive cruise control and lane keeping systems make the longitudinal vehicle control safer https://doi.org/10.1016/j.trc.2022.103756..

[6] Carl Bergenhem et. Al. Vehicle-to-Vehicle Communication for a Platooning System

[7] Lazar RG, Pauca O, Maxim A, Caruntu CF. Control Architecture for Connected Vehicle Platoons: From Sensor Data to Controller Design Using Vehicle-to-Everything Communication. Sensors (Basel). 2023 Aug 31;23(17):7576. doi: 10.3390/s23177576. PMID: 37688028; PMCID: PMC10490767.

[8] Lee Y, Ahn T, Lee C, Kim S, Park K. A Novel Path Planning Algorithm for Truck Platooning Using V2V Communication. Sensors (Basel). 2020 Dec 8;20(24):7022. doi: 10.3390/s20247022. PMID: 33302467; PMCID: PMC7764291.

[7] Lazar RG, Pauca O, Maxim A, Caruntu CF. ControlArchitecture for Connected Vehicle Platoons: From Sensor Data to Controller Design Using Vehicle-to-Everything Communication. Sensors (Basel). 2023 Aug 31;23(17):7576. doi: 10.3390/s23177576. PMID: 37688028; PMCID: PMC10490767.

[8] Lee Y, Ahn T, Lee C, Kim S, Park K. A Novel Path Planning Algorithm for Truck Platooning Using V2V Communication. Sensors (Basel). 2020 Dec 8;20(24):7022. doi: 10.3390/s20247022. PMID: 33302467; PMCID: PMC7764291.

[9] https://www.nhtsa.gov/document/safety-analysis-heavy-duty-truck-platooning-syste