Simulation Results

Controller 2 Updates:

The updated controller includes a new function block to calculate buffer_time, along with modifications to the final output function block. This improvement addresses situations where the ego car is far from the lead car but has a high relative velocity. In the previous model, the controller would accelerate the ego car, risking a crash due to the rapid approach to the lead car. The new function first calculates the distance required for the ego car to match the lead car's speed while decelerating at a balanced acceleration rate. It then determines the time needed by dividing this distance by the ego car's speed, ensuring safer and more controlled behavior in such scenarios.

Key updates include:

(1) Coasting instead of braking for a smoother ride and improved fuel economy.
(2) Restricting acceleration when the ego car is significantly faster than the lead car, even if the space gap is large.
(3) Limiting acceleration to prioritize fuel efficiency.
(4) Implementing dual-mode acceleration to balance safety with fuel economy.

These changes aim to create a smoother driving experience while optimizing energy use.

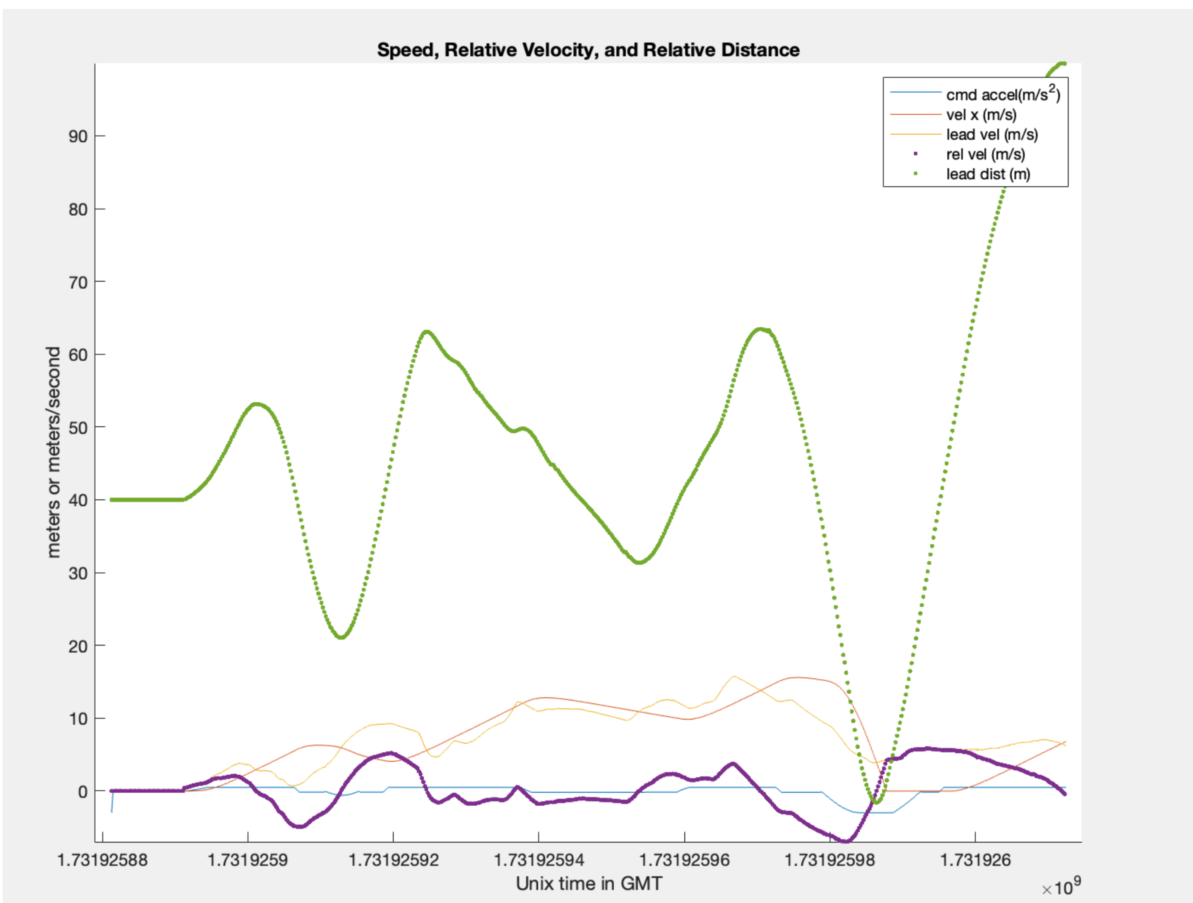

Figure 1: Acceleration, Speed, Relative Velocity, and Lead Distance for ACC Controller 2

The controller dynamically adapts to changes in the lead vehicle's behavior, as demonstrated in the plot. Gradual changes in cmd_accel and the use of coasting reflect improvements aimed at achieving smoother driving and better fuel economy. The system appears responsive, seen in the synchronization of the ego vehicle's velocity with the lead vehicle's behavior. The sharp increase in lead distance and the decrease in ego velocity (vel x) at the end suggest that the controller prioritizes safety during significant changes in the lead vehicle's behavior. Before the second lead distance peak, the ego vehicle is coasting as intended to improve fuel economy.

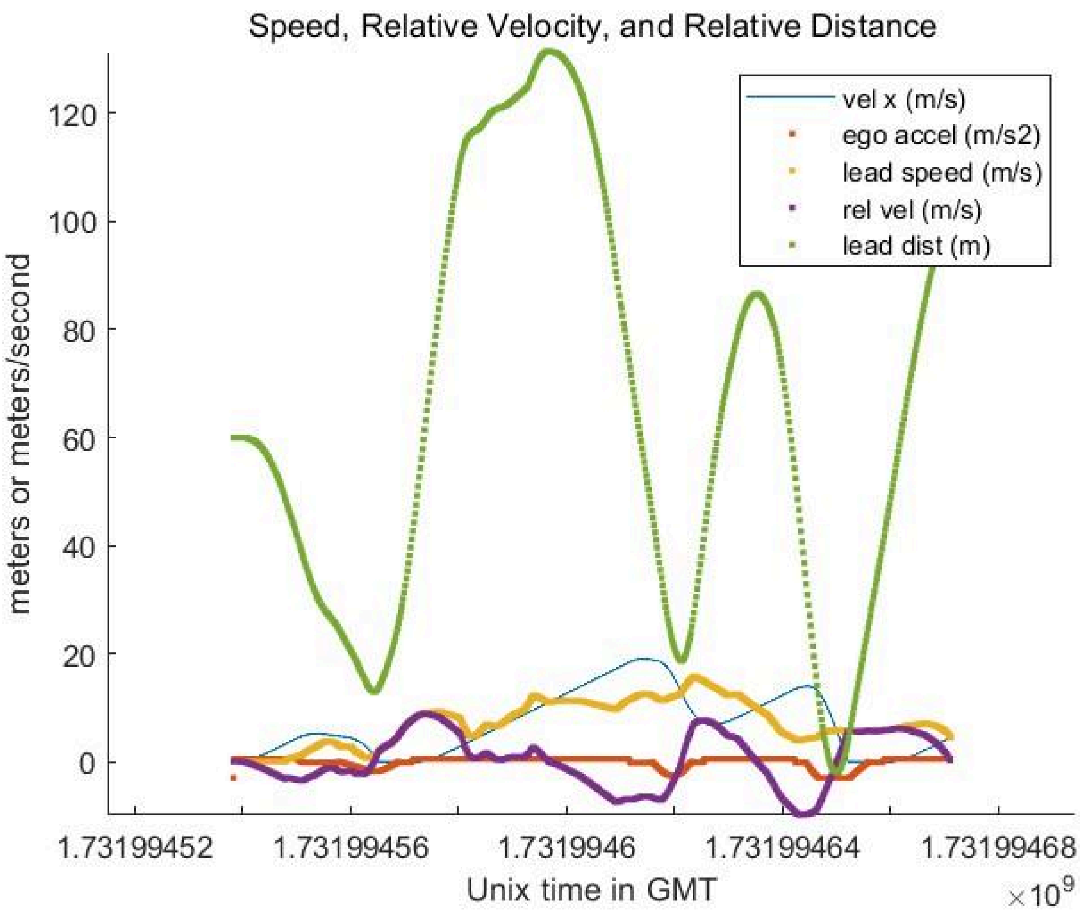

Figure 1.1: ACC Controller 2 with Increased Space Gap (60m from 40m)

NOTE: Max desired speed is **different in multi_vehicle.slx and skynetmini_controller2.slx (35 vs 20 m/s)**

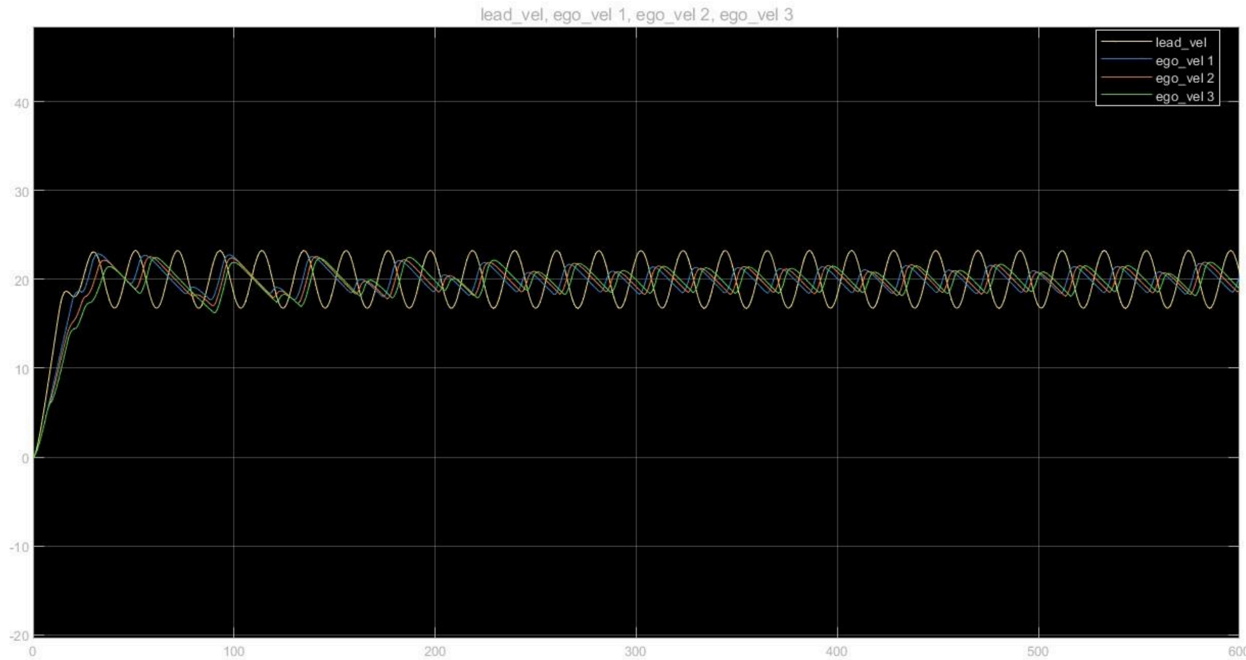

Figure 2: Speed of Each Vehicle in Multi-Vehicle Simulation

The multi-vehicle simulation demonstrates how multiple ego vehicles respond to a lead vehicle's velocity. Despite initial oscillation, the plot shows that all ego vehicles stabilize their velocities to track the lead vehicle.

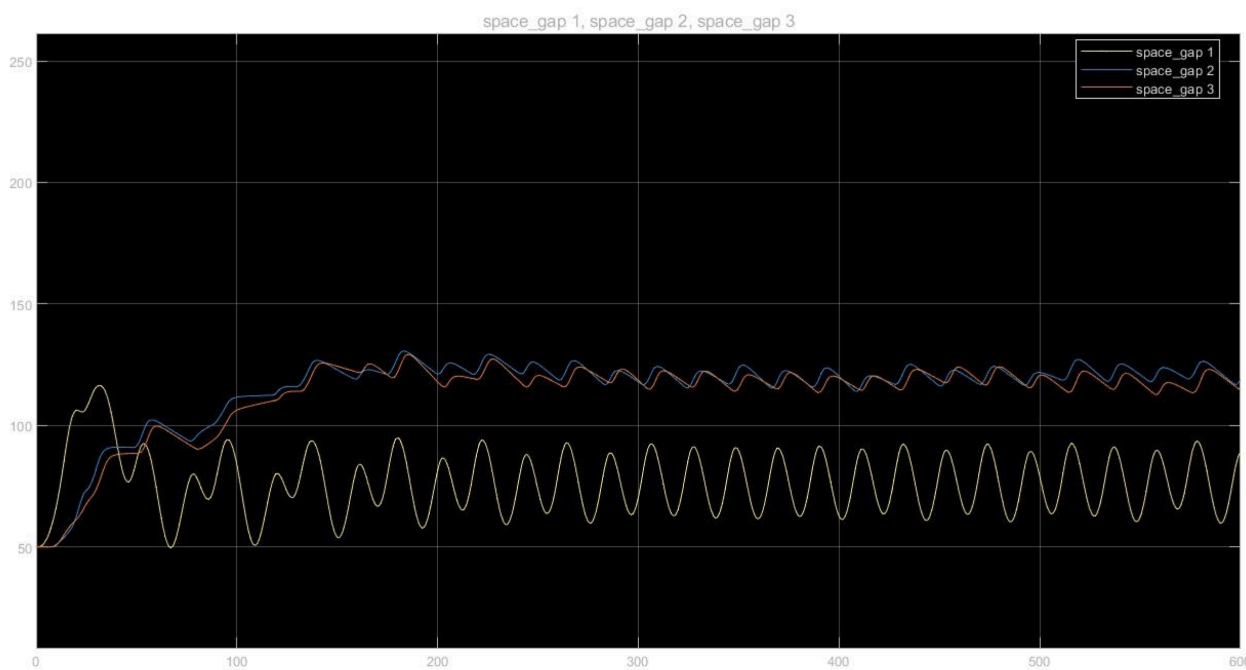

Figure 3: Space Gaps in Multi-Vehicle Simulation

Figure 3 shows the space gaps between multiple ego vehicles and their respective lead vehicles in a multi-vehicle simulation. The space gaps initially oscillate before stabilizing over time, indicating that the controllers are effectively managing safe following behaviors and distances.

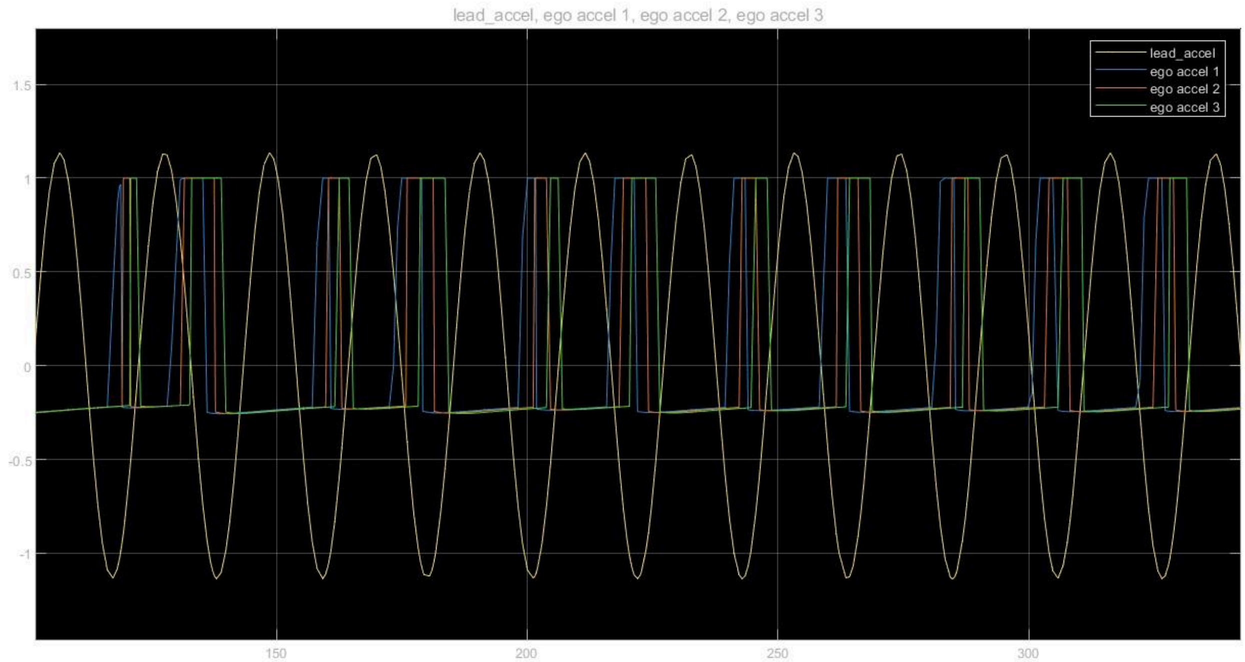

Figure 4: Cmd Accel's in Multi-Vehicle Simulation

This plot illustrates the commanded accelerations of multiple ego vehicles in response to the lead vehicle's acceleration. The ego accelerations generally follow the lead vehicle's behavior, with period oscillations that reflect adjustments to maintain safe behavior.