Safety and stability analysis of FollowerStopper

Chris Kreienkamp
Aerospace and Mechanical Engineering
University of Notre Dame
Notre Dame, IN, United States
ckreienk@nd.edu

Daniel Fishbein
Physics, Astronomy, and Materials Science Department
Missouri State University
Springfield, MO, United States
fishbein486@live.missouristate.edu

Rahul Bhadani
Electrical and Computer Engineering
University of Arizona
Tucson, AZ, United States
rahulbhadani@email.arizona.edu

Dr. Jonathon Sprinkle
Electrical and Computer Engineering
University of Arizona
Tucson, AZ, United States
sprinklejm@email.arizona.edu

***Abstract*—In this paper we demonstrate that the velocity controller, FollowerStopper, is safe and string unstable. FollowerStopper is a controller that is meant to be implemented on an autonomous vehicle or in an adaptive cruise control (ACC) system. Through mathematical proof, simulation in Simulink, and hardware in the loop implementation on a real autonomous vehicle through Robot Operating System (ROS) and Gazebo, several results are achieved. It is found that an autonomous vehicle controlled by FollowerStopper will never crash. FollowerStopper will dissipate larger traffic waves from human-driven vehicles but will amplify smaller velocity perturbations that are created within the controller. Given the maximum LiDAR range of 81 m, FollowerStopper will never command a velocity greater than 13.69 m/s.**

I. Introduction and Related Work

In order to maintain safety while driving, humans need a certain amount of safe distance between their car and the car directly in front of them so that if the car in front stops abruptly, they can react and stop in time to prevent a crash. From here forward the car which is of interest will be termed the autonomous vehicle (AV) and the car directly in front will be termed the lead vehicle. If the cars are traveling at a slow speed, the AV can follow at a closer distance because the car will not travel as far during the time it takes to react to the lead and to brake. When the number of cars on a section of highway increases, the car density increases, often termed as congestion. When highways are congested, cars must travel closer together than they normally would, so drivers must drive slower than the speed limit to maintain safety.

It was shown that humans, once the congestion reaches a certain threshold, will inevitably cause traffic jams. This will occur even if there are no traffic triggers, termed bottlenecks, such as lane changes, merges, tunnels, or other physical hindrances [1]. The reason for the formation of these ”phantom traffic jams” is that humans are only concerned with maintaining safety, but are typically not concerned about dissipating traffic. When a driver brakes, for example, the driver behind will often brake harder, and this chain of events will continue until cars must come to a complete stop. It is even proposed that bottlenecks lead to traffic jams because such events will cause the car density to exceed the threshold [1].

Throughput is the number of cars that pass through a given area over a certain time. The best alternative to traffic jams, meaning the situation which will allow for the greatest throughput, is for all of the cars to follow the same optimal velocity. In doing so, there will be no hard braking or quick acceleration, further providing benefits of improved fuel economy, less wear on the brake pads and engine, and preventing the frustration and stress that accompanies road rage. The throughput is highest when setting the optimal velocity of the autonomous vehicle to be the average speed of the traffic wave ahead [2].

With the development of new ACC and autonomous systems, a larger percentage of cars on the road will have some degree of automation. It will take many decades for all cars on the road to be autonomous, so it is critical to inspect the impact that just a few autonomous vehicles will have on the overall traffic flow. Stern et al. demonstrated that even a small percentage of autonomous vehicles ($< 5\%$) could have a substantial effect in reducing traffic [2]. Some researchers suspect that between 2020 and 2040 there could be the development of highway lanes solely for autonomous vehicles [3], at which point designers could rely on vehicle-to-vehicle communication with the formation of high-density platoons. Until then, however, it is necessary to design autonomous vehicles and ACC systems with the ability to safely interact with imperfect and unpredictable human drivers. To optimize the situation, the autonomous systems should be safer than human drivers and should do as much as they can to reduce traffic.

One such velocity controller is FollowerStopper, developed and tested in a variety of papers and experiments predominantly by the University of Arizona CAT Vehicle team [2], [4], [5]. [5] merely demonstrates that FollowerStopper can avoid collision and dampen traffic waves, but it does not assure safety or prove string stability.

II. Setup

Suppose there is a line of vehicles in a straight infinite lane that never has any lane changes or other bottlenecks, as pictured in Figure 1. In Figure 1, the middle car is vehicle i. The vehicle to the right is vehicle $i-1$, and the trend continues so that the vehicle at the very front of the line of vehicles will be vehicle 1. The vehicle to the left is vehicle $i+1$, and the trend continues so that the vehicle at the very end of the line of n vehicles will be vehicle n. The x-position of 0 is an arbitrary location, but for ease it can be thought of as the starting position of the last vehicle in the line such that at all times $t \geq 0$, the position of every vehicle $x_i \geq 0$.

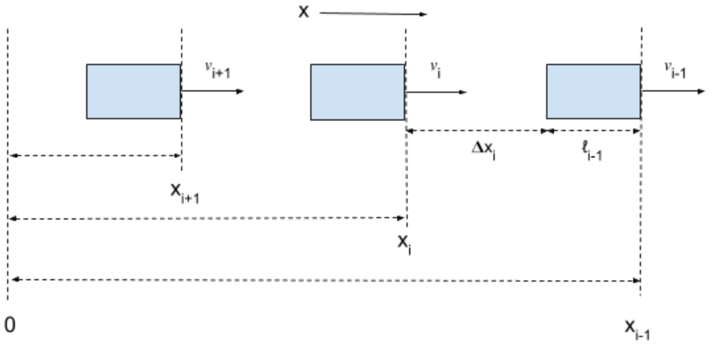


Fig. 1. Line of vehicles in a straight lane.

A. Variables

1) Relative distance, Δx_i: The relative distance for vehicle i is the distance between the front bumper of vehicle i and the back bumper of vehicle $i-1$. That is,

$$\Delta x_i = x_{i-1} - x_i - l_{i-1}, \tag{1}$$

where x_{i-1} is the position of the front bumper of vehicle $i-1$, x_i is the position of the front bumper of vehicle i, and l_{i-1} is the length of vehicle $i-1$.

2) Relative velocity, Δv_i: The relative velocity for vehicle i is the velocity difference between vehicle $i-1$ and vehicle i. That is,

$$\Delta v_i = v_{i-1} - v_i, \tag{2}$$

where v_{i-1} is the velocity of vehicle $i-1$ and v_i is the velocity of vehicle i.

3) Reference velocity, r: Also known as the desired velocity or optimal velocity, the reference velocity is the velocity at which the autonomous vehicle desires to travel. It is typically the average velocity over the length of a traffic wave, and is found by dividing the total distance traveled by a vehicle in front of the AV by the time. That is,

$$r = \frac{d_{total}}{t_{total}} \tag{3}$$

The reference velocity is typically sent to the AV by means of a roadside controller.

4) Spacing error, ϵ_i: The spacing error is the difference between the desired relative distance and the actual relative distance, giving,

$$\epsilon_i = \Delta x_{ides} - \Delta x_i. \tag{4}$$

B. Definitions

1) Safe: A scenario is safe if at all times for every vehicle i such that $1 < i \leq n$, then $\Delta x_i > 0$ m. This means that the relative distance will always be greater than 0 for every vehicle.

2) Individual vehicle stable: A velocity control law is considered to be individual vehicle stable if the spacing error of the AV approaches 0 with time if the lead vehicle travels at a constant velocity.

3) String stable: When a lead vehicle accelerates or decelerates in front of an individual vehicle stable AV, the spacing error will momentarily be nonzero. Suppose there is an infinite line of autonomous vehicles with one lead vehicle in front. The velocity control law is considered to be string stable if, during acceleration or deceleration of the lead vehicle, the spacing error decreases with each successive vehicle as they react to the change in velocity [6]. Additionally, all spacing errors must be in the same direction, either all negative or all positive for a particular change in velocity. [7] displays a theoretical plot of the spacing error of a string unstable and string stable group of vehicles, reproduced in Figure 2.

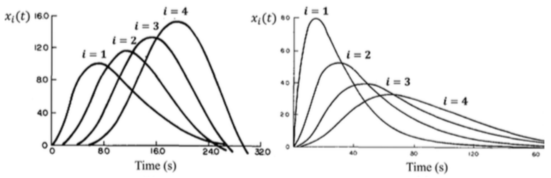


Fig. 2. String unstable and string stable spacing error vs time representation.

C. Infrastructure

The FollowerStopper velocity controller is first modeled in Simulink and then used in conjunction with ROS to test in the physics-based simulation engine, Gazebo. After demonstrating success in Gazebo, the velocity controller is implemented onto the Cognitive and Autonomous Test (CAT) Vehicle at the University of Arizona using a hardware in the loop (HIL) configuration as described in [8].

The CAT Vehicle is a modified Ford Hybrid Escape with a SICK LMS 291 Front Laser Rangefinder, a Velodyne HDL-64E S2 LiDAR, two Pointgrey Firefly MV FFMV-03M2C cameras, and a Novatel VPS/IMU. The FollowerStopper velocity controller uses distance data from the LiDAR to determine the relative distance to the car in front Δx_i. Using the relative distance data, the AV can determine the relative velocity [4]. Assuming that the AV knows its own velocity at all times, it can use the relative velocity data to determine the velocity of the lead vehicle.

III. Parameters

In the development of the FollowerStopper controller, several parameters are often used. They are grouped here to allow for easy reference. Some of the parameters involve acceleration due to gravity, which will be taken as $G = 9.80665\frac{m}{s^2}$ for our purposes. In choosing the parameters, safety is the priority. If safety is not a concern, keeping a small distance between the AV and the lead is desirable in order to accommodate for a higher density of cars on the road and to prevent lane changes of human drivers in front of the AV. Table I at the end of the section summarizes the findings.

A. Minimum relative distance, ψ

Minimum relative distance refers to the minimum acceptable distance between the AV and the lead. While stopped, it is expected that $\Delta x = \psi$. We chose an arbitrary value of $\psi = 1$ because we thought that if the AV was any closer to the lead, the passengers of the AV might be uncomfortable because they do not have control of the car.

B. Comfortable acceleration, a_{cmft}

The comfortable acceleration limits the maximum acceleration of the AV. When the reference velocity increases, the AV will not immediately travel at the reference velocity, but will increase its velocity at an acceleration of a_{cmft}. Particularly in congested traffic conditions, it is undesirable for the AV to have jerky velocity changes. Though it is essential to safety to leave the deceleration of the AV unrestrained, limiting the acceleration will provide for greater passenger comfort and expectability. comfortable acceleration details comfortable accelerations in the range of $0.11G$ to $0.15G$ for public mass transportation systems. Because passengers will be much less affected by a car's acceleration due to comfortable seating and expectability than a public mass transport acceleration, we chose $0.15G$ a reasonable value.

C. Comfortable deceleration, a_{dcmft}

The comfortable deceleration defines the maximum amount by which the reference velocity can change. In cases where the reference velocity drops to a much lower value, it is not required for the AV to immediately travel at that value in order to remain safe. a_{dcmft} ensures a smooth transition when safety is not of concern. [9] details $0.266G$ as a deceleration that is comfortable to passengers and is preferred by a driver.

D. Maximum acceleration, a_{max}

The maximum acceleration refers to the maximum possible acceleration of the AV. Fiver different websites [10]–[14] listed different maximum acceleration data for the Ford Escape Hybrid, which is the vehicle under consideration. The data was based off an experiment where the vehicle would start at a stop and would accelerate as quickly as possible to 60 mph. For safety purposes, the largest acceleration was chosen as the maximum acceleration for the Ford Escape Hybrid, that is $3.53\frac{m}{s^2}$ [11]. If the maximum acceleration of the AV is not known, [9] details $3.34\frac{m}{s^2}$ as a reasonable maximum acceleration. This can be used if a more accurate estimation cannot be found.

E. Maximum deceleration, a_{dmax}

The maximum deceleration refers to the maximum possible deceleration of the AV. Five different websites [10]–[14] listed different maximum deceleration data for the Ford Escape Hybrid. The data was based off of an experiment where the vehicle would travel at a constant velocity before braking as hard as possible, and then reporting the stopping distance. For safety purposes, the deceleration with the lowest absolute value was chosen as the maximum deceleration for the Ford Escape Hybrid, that is $-7.66\frac{m}{s^2}$ [12]. If the maximum deceleration of the AV is not known, [15] details $-3.99\frac{m}{s^2}$ as the maximum deceleration of a passenger car traveling on wet pavement. This can be used if a more accurate estimation cannot be found.

F. Deceleration ratio, k

The deceleration ratio is the ratio between the maximum deceleration of the lead vehicle and the maximum deceleration of the AV. For FollowerStopper to be safe, as is shown in the next section, there must be some knowledge of how the lead and AV accelerations relate. A deceleration ratio less than unity, that is $k < 1$, means that the AV can decelerate to a stop faster than the lead, meaning that it can travel closer than normal to the lead and still be safe. A deceleration ratio greater than unity, that is $k > 1$, means that the lead can decelerate to a stop faster than the AV, meaning that the AV must follow at a greater relative distance in order to have a greater stopping distance. To be safe, assume that the lead vehicle can decelerate as fast as physically possible. From section A of the Appendix, the deceleration of a vehicle is $a = -G\mu$, where μ is the coefficient of friction. [16] concludes that the maximum tire-road friction coefficient in normal driving on a dry road is $\mu_{max} = 1$, so the maximum deceleration of the lead vehicle is $-G$.

TABLE I
FollowerStopper Parameters

Name	Symbol	Calculation	Value
minimum relative distance	ψ	—	1
comfortable acceleration	a_{cmft}	$0.15 \cdot G$	1.47 $\frac{m}{s^2}$
comfortable deceleration	a_{dcmft}	$-0.266 \cdot G$	-2.61 $\frac{m}{s^2}$
maximum acceleration (Ford Escape Hybrid)	a_{max}	—	3.53 $\frac{m}{s^2}$
maximum deceleration (Ford Escape Hybrid)	a_{dmax}	—	-7.66 $\frac{m}{s^2}$
deceleration ratio (Ford Escape Hybrid)	k	$\frac{-G}{a_{dmax}}$	1.28

IV. FollowerStopper Description

A. Classification

The premise of FollowerStopper is to command exactly the reference velocity r whenever safe because this is the velocity that could dissipate already formed traffic jams and could prevent new traffic jams from forming. Assuming that the lead starts a safe distance ahead of the AV, when $r \leq v_{lead}$, FollowerStopper will command the reference velocity, that is $v_{cmd} = r$. FollowerStopper takes the AV's velocity, relative velocity to the lead vehicle, and relative distance to the lead vehicle as inputs to determine three relative distances ξ_1, ξ_2, and ξ_3, in order of increasing magnitude.

Suppose now that $r > v_{lead}$, so the relative distance Δx will be decreasing. Once the AV gets within the relative distance $\Delta x \leq \xi_3$, FollowerStopper will slow down the vehicle by commanding a velocity less than r. If the AV must continue to travel at a value lower than the relative velocity r, FollowerStopper, by slowly decreasing the command velocity, will cause the AV to travel at the lead velocity v_{lead}, following at a relative distance of $\Delta x = \xi_2$. If the lead vehicle decelerates unexpectedly or the relative distance decreases further to $\Delta x = \xi_1$, FollowerStopper will command the AV to stop.

The mathematical representation of the FollowerStopper controller from [4] is reproduced below.

$$v_{cmd} = \begin{cases} 0 & \Delta x \leq \xi_1 \\ v\dfrac{\Delta x - \xi_1}{\xi_2 - \xi_1} & \xi_1 < \Delta x \leq \xi_2 \\ v + (r - v)\dfrac{\Delta x - \xi_2}{\xi_3 - \xi_2} & \xi_2 < \Delta x \leq \xi_3 \\ r & \xi_3 < \Delta x \end{cases} \tag{5}$$

$$\xi_j(\Delta v) = \omega_j + \frac{1}{2\alpha_j}(\Delta v^*)^2 \quad \text{for} \quad \text{j} = 1, 2, 3 \tag{6}$$

where $v = min(max(v_{lead}, 0), r)$. r is the reference velocity as taken from the output of smoothUpParams and v_{cmd} is the command velocity that is sent to the AV. The ω_j and α_j values in the equation for ξ_j are distance and deceleration parameters, respectively, and $\Delta v^* = min(\Delta v, 0)$. [4] suggests for future work to optimize the distance and deceleration parameters, which were previously set to $\omega_1 = 4.5$ m, $\omega_2 = 5.25$ m, $\omega_3 = 6.0$ m, $\alpha_1 = 1.5\frac{\text{m}}{\text{s}^2}$, $\alpha_2 = 1.0\frac{\text{m}}{\text{s}^2}$, and $\alpha_3 = 0.5\frac{\text{m}}{\text{s}^2}$. The following sections seek to accomplish this task of finding the ideal ξ_j values.

B. SmoothUpParams

Theoretically, the reference velocity will be given to the AV by means of a roadside controller which takes traffic data and determines the optimal constant velocity. SmoothUpParams is a function of the new reference velocity r, the autonomous vehicle velocity v_{AV}, the maximum comfortable acceleration of the autonomous vehicle a_{cmft}, and the maximum comfortable deceleration of the autonomous vehicle a_{dcmft}. Using these inputs, SmoothUpParams edits the reference velocity so that it is a reasonable value when it is sent to the FollowerStopper controller. Suppose, for example, the speed limit and consequently the reference velocity were to change from 10 m/s to 15 m/s and no car was in front of the AV. Without SmoothUpParams, FollowerStopper would command the AV to travel at the reference velocity of 15 m/s immediately, but this will result in an unnecessarily large acceleration. SmoothUpParams will cause the reference velocity to increase at a slow rate so that FollowerStopper can comfortably transition to the new reference velocity without accelerating faster than a_{cmft} or decelerating faster than a_{dcmft}.

C. Designing ξ_1 for safety

According to FollowerStopper, if $\Delta x \leq \xi_1$, then $v_{cmd} = 0$. That is, if the relative distance between the AV and the lead is less than or equal to ξ_1, the AV should brake as hard as possible or remain stopped if already at a stop. If FollowerStopper commands $v_{cmd} = 0$, the AV will not initiate the emergency braking command until δ seconds after the FollowerStopper command due to system delay. To prove that FollowerStopper is safe, it must be shown that if $\Delta x = \xi_1$, braking as hard as possible after a delay of δ seconds will never result in a crash. This assumes that no unexpected obstacle comes between the AV and the lead. The metric is computed as:

$$\xi_1 = \psi + \Delta v^{**} + v_{AV}(1 - \frac{a_{max}}{a_{dmax}})\delta + \frac{a_{max}}{2}(1 - \frac{a_{max}}{a_{dmax}})\delta^2 \tag{7}$$

where $\Delta v^{**} = max(0, \frac{1}{2ka_{dmax}}(v_{lead}^2 - kv_{AV}^2))$. The derivation can be found in section B the Appendix.

D. Designing ξ_2 for string stability

When $v_{AV} > v_{lead}$, the relative distance Δx will decrease as the AV approaches the lead. When Δx drops below ξ_3, FollowerStopper is designed to incrementally command a smaller velocity until the command velocity is equal to the velocity of the lead $v_{cmd} = v_{lead}$. Once $v_{AV} = v_{lead}$, the AV will maintain the distance $\Delta x = \xi_2$. If the relative distance drops below this equilibrium distance, that is $\Delta x < \xi_2$, the velocity of the AV will decrease below v_{lead} in an attempt to recover this distance.

If the distance is greater than the equilibrium distance, that is $\Delta x > \xi_2$, the velocity of the AV will increase in an attempt to recover this distance unless v_{lead} exceeds the reference velocity, at which point $v_{cmd} = r$. Therefore, the AV will spend the majority of its functioning at or above the distance ξ_2. It is important, then, that ξ_2 is a string stable distance. According to [6], string stability is maintained when the time-gap h between two cars is such that $h \geq 2\tau$, where τ is the time constant of any lags in tracking the command velocity, or in other words, it is the delay of the system δ. It is reasonable to treat the relative distance ξ_1 as the point from which to maintain a time-gap greater than 2δ because at $\Delta x = \xi_1$, the AV must stop. With this knowledge the metric is computed as:

$$\xi_2 = \xi_1 + 2v_{AV}\delta \tag{8}$$

E. Designing ξ_3 for comfort

If $\Delta x > \xi_3$, then $v_{cmd} = r$. When $\Delta x \leq \xi_3$, the AV will begin a comfortable deceleration from r to v_{lead}, maintaining the distance ξ_2 while it travels at v_{lead}. In order to preserve an equal spacing between all of the relative distance parameters ξ_3, ξ_2, and ξ_1, FollowerStopper is designed such that ξ_2 is the average of ξ_1 and ξ_3. A brief amount of algebra reveals the calculation for ξ_3. The metric is computed as:

$$\xi_3 = 2\xi_2 - \xi_1 \tag{9}$$

F. Maximum FollowerStopper speed

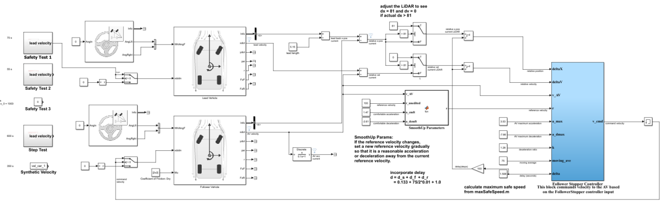
Due to the limited sight of the LiDAR, the AV can only detect relative distance data up to 81 m ahead. Therefore, if there is no obstacle within 81 m, FollowerStopper will assume that there is a vehicle at $\Delta x = 81$ m traveling at the same speed as the AV. In such a case, the AV will continue to accelerate until $\xi_2 = 81$ m because at this point, the AV will no longer accelerate or decelerate unless the lead changes its velocity, which will never happen. $\xi_2 = 81$ m when $v_{AV} = 13.69$ m/s, so the current implementation of FollowerStopper makes it impossible for the AV to travel faster than that speed. The maximum speed would increase if either there was a new formula for ξ_2 that would make it smaller, or if the range of the sensors were to increase, for example with the use of radar instead of LiDAR. In our testing we only incorporated LiDAR, however, so we used 81 m as the maximum distance of a vehicle ahead.

G. Further considerations

FollowerStopper is designed such that the AV will not accelerate faster than a_{max} and will not decelerate faster than a_{dmax}. This is necessary because even if FollowerStopper commands an acceleration outside of this range, it is physically impossible for the AV to execute the command. If FollowerStopper were to command an unrealistic velocity, it could cause irreparable wear on the tires or and excessive load to the engine. Additionally, in order to reduce oscillatory noise in the command velocity data, the FollowerStopper controller, after determining a new command velocity, will take a moving average of the new value and a set number of previous v_{cmd} values. The current controller uses a moving average of 75 steps, meaning that the command velocity will be the average of the newly calculated v_{cmd} and the 74 previous command velocities. The reason for this choice is detailed in the section 6.

V. Simulink Model

To compare the old and new versions of FollowerStopper, a Simulink model was developed incorporating the Vehicle Body 3DOF model. The model incorporates expected sensor and actuation delays and can display output such as the lead and AV velocities, positions, and accelerations. It is not completely accurate, as when sending a delayed v_{AV} to the Vehicle Body 3DOF block, the simulation produces inaccurate data. Therefore, the v_{AV} data could not be delayed. The final representation of the model is displayed below for a lead vehicle and AV. The model is computed using a fixed time-step of 0.01 s, meaning that it is updated every 0.01 s.



adjust the LiDAR to see dx = 81 and dv = 0 if actual dx > 81
lead velocity
Safety Test 1
lead velocity
Safety Test 2
Safety Test 3
lead velocity
Step Test
vel_car_1
Synthetic Velocity
Lead Vehicle
Follower Vehicle
lead length
SmoothUp Parameters
SmoothUp Params: If the reference velocity changes, set a new reference velocity gradually so that it is a reasonable acceleration or deceleration away from the current reference velocity.
incorporate delay d = d_s + d_f + d_r = 0.133 + 75/2*0.01 + 1.0
calculate maximum safe speed from maxSafeSpeed.m
deltaX
deltaV
v_AV
r
a_max
a_dmax
k
moving_ave
delta
v_cmd
Follower Stopper Controller
This block commands velocity to the AV based on the FollowerStopper controller input

Fig. 3. Simulink model.

A. Vehicle Body 3DOF

The Vehicle Body 3DOF Dual Track block in Simulink models the lead vehicle and the AV. This particular block is ideal for creating virtual sensor data, which can then be used by FollowerStopper. Further information about the model can be found in [17]. By inputing a velocity ”xdot”, steering angle, and initial longitudinal position ”X_o” to the lead vehicle, the block artificially creates vehicle data including the instantaneous position and instantaneous velocity of the lead. Another Vehicle Body 3DOF Dual Track block models the AV, using the same steering angle, but starting behind the lead at a lesser longitudinal position ”X_o”. The input velocity for the AV comes from the command velocity v_{cmd} output of FollowerStopper.

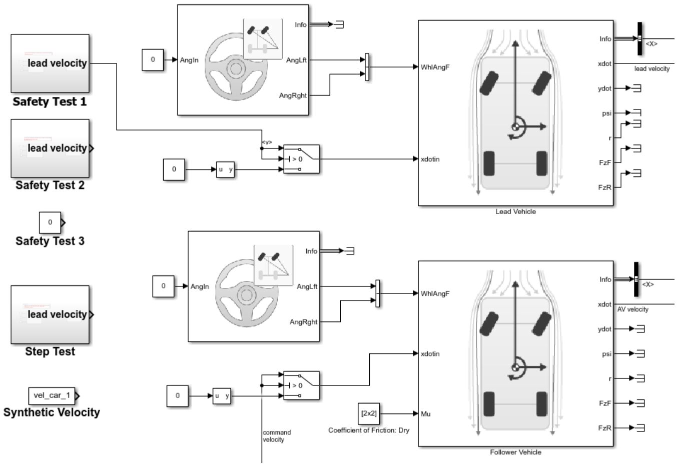


Fig. 4. Lead vehicle and AV model in Simulink using the Vehicle Body 3DOF Dual Track block.

Several velocity data files were created for the lead to follow to test the safety, string stability, and overall response of the AV to different velocities and accelerations. One particular velocity data file used the exact velocity of a human-driven car from [18]. Experiment E, car 1 was chosen as the vehicle off of which to obtain a synthetic lead vehicle velocity because Experiment E features the most oscillatory traffic without intervention by the CAT Vehicle to modify the flow of cars. Information about the other experiments in the dataset can be found in [19].

B. Delay

The delay δ of the system is of supreme importance because at high speeds, the AV can travel a considerable distance before receiving and implementing a new velocity command. Contributions to the delay include the LiDAR sensor delay (0.133 s), the moving average filter delay $(\frac{75 \text{ moving steps}}{2}(0.01\frac{\text{s}}{\text{step}}))$, and a fixed delay (1.0 s) due to friction, actuator dynamics, controller update rate, etc. From [4], considering that the position sampling frequency F_s and velocity sampling frequency F_v are both 75 Hz,

$$\begin{aligned}\delta &= \frac{\delta_f F_s}{F_v} + \delta_r = \delta_f + \delta_r \\ &= 0.133 + \frac{75}{2}(0.01) + 1 \\ &= 1.508 \text{ s}\end{aligned}$$

where δ_f is the filter delay and δ_r is the fixed delay. It is desirable to replicate the actual CAT Vehicle in the Simulink model as nearly as possible. As mentioned above, the LiDAR takes relative distance data, which needs to be filtered to become relative velocity data, adding the 0.133 s sensor delay. Assuming that the AV velocity is known at all times, the AV velocity, relative distance, and relative velocity go as inputs into FollowerStopper. The command velocity v_{cmd} is modified with a moving average filter, adding the 0.375 s filter delay. Once there is a v_{cmd} value, it is sent to the actuators, adding the 1.0 s delay.

It is obvious that it would be inaccurate to use the current values of the relative distance, relative velocity, and AV velocity in the Simulink model. Though the AV knows its velocity at the time the LiDAR data is processed, it should use the velocity from 0.133 s beforehand, when it received the LiDAR data. It is currently unclear as to which value of the AV velocity FollowerStopper uses in the CAT Vehicle.

Given this knowledge, the Simulink model could be considered reasonably accurate if the relative distance, relative velocity, and AV velocity were all delayed by 1.508 s. Unfortunately, when the AV velocity is delayed, the simulation updates v_{cmd} once every 1.508 s, slowing the simulation from its normal update rate of 0.01 s. The problem is related to the Vehicle Body block rather than FollowerStopper, and a solution could not be found that would not compromise more important parts of the model. The model therefore uses the delayed relative distance and relative velocity values, but it uses the current AV velocity, deeming the model inaccurate but still an acceptable platform from which to test FollowerStopper before implementation into Gazebo or the actual CAT Vehicle.

C. SmoothUpParams

The SmoothUpParams block, as described above, edits the reference velocity before sending it to FollowerStopper. If the reference velocity changes, most likely due to a road-side controller sending information to the AV, SmoothUpParams will slowly and continually update the reference velocity sent to FollowerStopper. SmoothUpParams will ensure that, if FollowerStopper is commanding exactly the reference velocity, the changing reference velocity will not cause the AV to exceed a maximum comfortable acceleration or deceleration. Once SmoothUpParams reaches the actual reference velocity, it will simply continue to send the current reference velocity until it receives a new one. Our tests never include a changing reference velocity, so SmoothUpParams could technically be removed from the model and nothing would change. However, if further work were to be done in optimizing the reference velocity, SmoothUpParams would be essential.

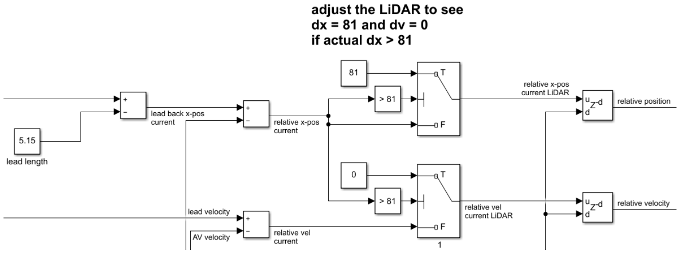


Fig. 5. Delay in Simulink.

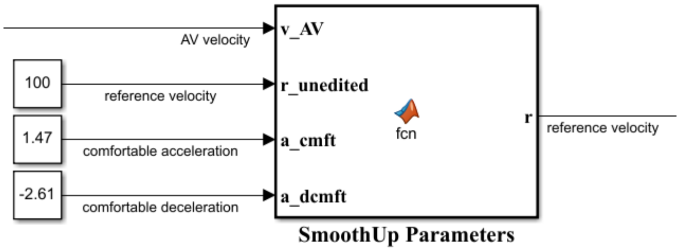


Fig. 6. SmoothUpParams in Simulink.

D. FollowerStopper

The FollowerStopper block takes Δx, Δv, r, a_{max}, a_{dmax}, k, the moving average, and δ as inputs and uses a MATLAB Function to calculate ξ_1, ξ_2, ξ_3, and v_{cmd} as described in section 4 above. The FollowerStopper block outputs the new v_{cmd}, and then that value is used as the input velocity to the AV for the next time step.

E. Multi-car model

The descriptions thus far have focused on the development of a two-car model, with one lead vehicle and one AV. This is sufficient for building the model, implementing in Gazebo, and testing with the CAT Vehicle. Additionally it is sufficient for determining if FollowerStopper is safe because any velocity can be input to the lead, and the AV will follow. When determining string stability, however, a multi-car model is required in order to see how each successive car reacts to a change in velocity by the lead vehicle. To test for string stability, the AV Vehicle Body 3DOF block, delay blocks, SmoothUpParams, and FollowerStopper were combined into one block to represent one AV, as pictured in Figure 8. The one combined block could take the lead position, moving

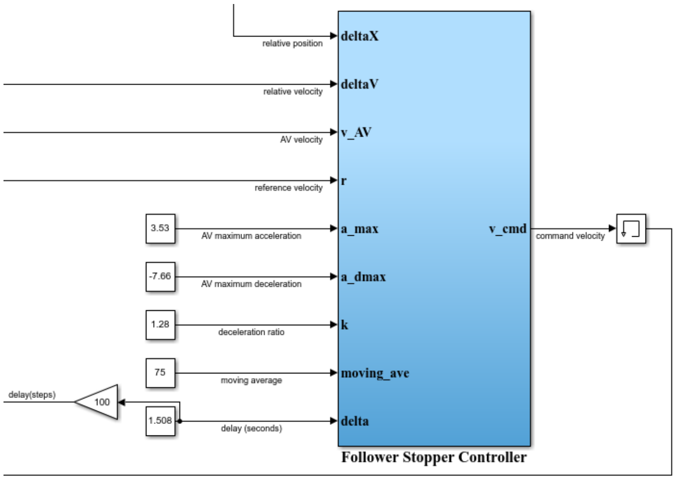


Fig. 7. FollowerStopper in Simulink.

average, reference velocity, and lead velocity as inputs and would output the AV position, spacing error, and velocity. By connecting several blocks in a line and editing the initial longitudinal position of each vehicle, a multi-car model could be developed and analyzed, as seen in Figure 9.

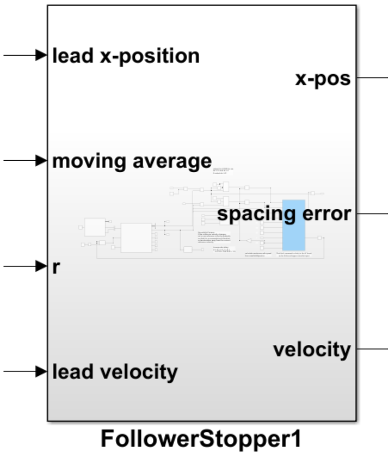


Fig. 8. Combined AV and FollowerStopper block in Simulink.

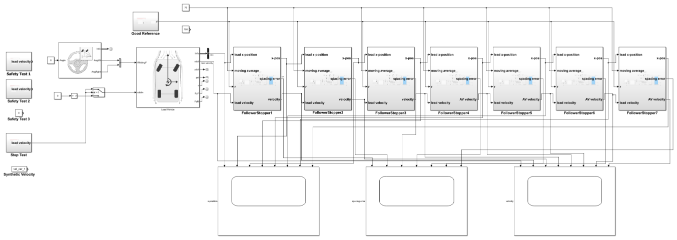


Fig. 9. Multi-car model in Simulink.

VI. Safety and String Stability Verification

A. *Safety and string stability tests*

Three tests were developed to test the safety of the original FollowerStopper and the new version of FollowerStopper, based on the optimized ξ_j values. When designing the tests, the maximum speed of the AV from previous calculations was considered so that the tests would not operate above this velocity. The safety of a controller is determined by its minimum relative distance. In a worst-case scenario, FollowerStopper would have an incredible high reference velocity, so all safety tests were recorded using a relative velocity of $r = 100$ m/s. Step Test was performed with a good and a bad reference velocity, as the reference velocity has an effect on the level of string stability. The safety tests used the two-car Simulink model with the original and new version of FollowerStopper while the string stability tests required a seven-car Simulink model and used only the new version of FollowerStopper.

1) Safety Test 1: The front of the lead vehicle starts 10 m in front of the AV. At the start of the simulation it accelerates from 0 to 12 m/s at an acceleration of a_{max}. It continues traveling at 12 m/s for 40 seconds. It then decelerates to 0 at a deceleration of G.

2) Safety Test 2: The front of the lead vehicle starts 10 m in front of the AV. At the start of the simulation it accelerates from 0 to 10 m/s at an acceleration of a_{max}. It continues traveling at 10 m/s for 25 seconds. It then accelerates at an acceleration of a_{max} for $\delta = 1.508$ seconds. It then immediately decelerates to 0 at a deceleration of G.

3) Safety Test 3: The front of the lead vehicle starts 1000 m in front of the AV. The lead vehicle remains stationary through the entire test.

4) Step Test: The front of the lead vehicle starts 10 m in front of the AV. At the start of the simulation it accelerates to 10 m/s at an acceleration of G. The lead continues traveling at 10 m/s for 175 seconds. It then decelerates to 2 m/s at a deceleration of G. It continues traveling at 2 m/s for 150 seconds. It then accelerates to 10 m/s and continues traveling at 10 m/s for the remainder of the test.

B. *Determination of the moving average*

Before performing the safety and string stability tests, an ideal moving average value needed to be chosen for the new FollowerStopper. To decide, Step Test was implemented using the seven-car model. The moving average was modified in each run, using no moving average, a moving average of 5, 25, 50, 75, and 100. The plots of each are detailed in Figures 12–17, respectively, in section C of the Appendix.

In order to have individual vehicle stability, the actual relative distance must approach the desired relative distance as time approaches infinity if the lead travels at a constant velocity. Because FollowerStopper is operating at a velocity lower than the reference velocity, the desired relative distance is ξ_2. The plots of FollowerStopper with no moving average, a moving average of 5, 25, and 50 reveal that the velocities never reach a steady value, meaning that the relative distance would not be constant, so FollowerStopper in such cases would not be individual vehicle stable.

The plots of the moving average of 75 and 100 both reveal individual vehicle stability, but the moving average of 100 results in slight oscillations in the last vehicle at around 400 seconds, so a moving average of 75 was chosen.

C. Safety verification of the original FollowerStopper

The original ξ_j values for FollowerStopper were not optimized but were merely chosen to achieve some desired behavior [4]. Though the original FollowerStopper is safe in some instances, it is not always safe, making it a non-viable option for commercial use. In Safety Test 1, the relative distance drops to -0.7 m. In Safety Test 2, the relative distance drops to -5.7 m. In Safety Test 3, the relative distance drops to -162.3 m. In all three tests, the relative distance is negative, meaning that implementation of the original FollowerStopper results in a crash and is unsafe. The plots of the three tests are detailed Figures 18–20 in section D of the Appendix.

D. Safety verification of the new FollowerStopper

Due to the derivations for ξ_j as detailed in section 4, the new FollowerStopper is designed for safety. In Safety Test 1, the relative distance drops to 6.6 m. In Safety Test 2, the relative distance drops to 5.6 m. In Safety Test 3, the relative distance drops to 5.0 m. In all three tests, the minimum relative distance $\Delta x > 0$, so the new implementation of FollowerStopper will not result in a crash and is safe. The plots of the three tests are detailed in Figures 21–23 in section D of the Appendix.

E. String stability verification

Based on the description of string stability in the definitions section, we can test for string stability based on the AV velocity and spacing error. To test using velocities, during acceleration or deceleration or the lead, the acceleration of each successive FollowerStopper vehicle should be less extreme in order to be string stable, and the velocities should not overshoot. Additionally, velocity perturbations should dissipate with each successive car. This means that if there is a deceleration from 10 m/s to 5 m/s, each vehicle should decelerate at values decreasing in magnitude, and no vehicle should travel slower than 5 m/s at any point.

Another way to test for string stability is based on spacing error, or the difference between the actual relative distance and desired relative distance. Because FollowerStopper is designed to follow at a distance of ξ_2 when the reference velocity is greater than the lead velocity, we can treat ξ_2 as the desired relative distance. During acceleration or deceleration, the maximum spacing error should decrease as each vehicle reacts to the change in velocity. The Step Test was performed using a bad reference velocity of $r = 100$ m/s and a good reference velocity. In a real traffic-wave, a lead vehicle would slow down from its original velocity, and after exiting the wave, it would speed up back to its original velocity. The good reference velocity is designed to dissipate the traffic-wave of the Step Test. Therefore, the reference velocity is 6.1 m/s for 327 s, at which point the first AV behind the lead will catch up to the lead. At exactly this instant, the lead begins to accelerate up to 10 m/s, so the reference velocity increases to 10 m/s. In such a case, the autonomous vehicles travel at a constant velocity and never have to decelerate.

Figure 24 displays the velocities of the lead and six FollowerStopper followers with a bad reference velocity. Each successive car accelerates and decelerates with a smaller magnitude, and no car overshoots 10 m/s or 2 m/s, suggesting string stability. Figure 25 displays the velocities of the lead and six FollowerStopper followers with a good reference velocity. Though there is no overshoot, small perturbations in the first couple of FollowerStopper vehicles amplifies so that the last FollowerStopper has considerable oscillations in its velocity, suggesting string instability.

Figure 26 displays the spacing errors of the lead and six FollowerStopper followers with a bad reference velocity. Particularly in the middle region, each successive vehicle. has a decreasing spacing error. Though there are some wayward data points at the beginning of the data corresponding to the second acceleration in the Step Test, the data shows a similar trend of a decreasing spacing error with each successive vehicle, suggesting string stability. The regions on the left and right of the plot are concerning because a string stable controller should have either all positive or all negative spacing errors for a specific acceleration or deceleration. This is not the case in the left and right regions, suggesting string instability.

Figure 27 displays the spacing errors of the lead and six FollowerStopper followers with the good reference velocity. The left region is similar to the left region in the bad reference velocity graph, but it is smoother. The middle region, as expected, displays no deviation from the desired relative distance, as expected. The right region, however, exemplifies the magnification of small velocity perturbations that was apparent in Figure 25 at around 350 s, suggesting string instability.

Though FollowerStopper demonstrates some qualities of string stability, the current version cannot be considered string stable because of its magnification of small perturbations of velocity. Though it seems to dissipate larger velocity changes such as from the lead, it fails to dissipate smaller changes, demonstrated particularly well by the oscillations that grow at around 50 seconds in Figure 28, which implements Safety Test 1 on the seven-car model. If the small perturbations were to be dissipated, a more convincing case could be made for the string stability of FollowerStopper. This could be achieved through the design of a more successful command velocity filter than the current moving average filter, but this could be developed in future work.

VII. Results

A. Gazebo implementation

Using Robot Operating Systems (ROS), the Simulink model could be downloaded as C/C++ code to be implemented in the physics-based simulator Gazebo. Gazebo operates using a hardware in the loop (HIL) principle with the CAT Vehicle, allowing for accurate estimation of implementation of FollowerStopper onto the real vehicle. When converting the Simulink model to C/C++ code, the vehicle dynamics blocks and all artificial delays were removed because the CAT Vehicle will introduce its own delays.

B. CAT Vehicle testing

Using the CAT Vehicle, we tested the new FollowerStopper controller in a parking lot owned by the University of Arizona with minimal pedestrian or vehicle traffic. A human-driven vehicle traveled at a slow velocity, acceleration, and deceleration, and the CAT Vehicle successfully and safely reacted to the input. Subsequently, a human moved in front of the vehicle at a walking pace while making several stops along the way, and the vehicle safely followed without any driver input to the gas or brake pedals. No conclusions about string stability could be made during the physical tests because such conclusions can only be tested with several autonomous vehicles with FollowerStopper. Unfortunately, there is only one CAT Vehicle, limiting string stability testing.

VIII. Conclusion

In this paper we present an optimized version of the velocity controller FollowerStopper using theoretical proof, Simulink simulation, and implementation onto the CAT Vehicle autonomous vehicle. The development of the optimization and development of the Simulink model, which served as the primary platform for testing, were discussed rigorously. The original version of FollowerStopper proved useful in avoiding crashes and dissipating traffic waves in previous experiments setup through the University of Arizona. We provide a formal analysis of the safety and string stability of the velocity controller.

The physics-based proof demonstrates that FollowerStopper should never crash theoretically, and a variety of tests with the Simulink model asserts this claim. Some tests in Simulink demonstrate string stability, while others provide contradictions. FollowerStopper possesses qualities of string stability through its dissipation of the synthetically-created vehicle velocity data, but it amplifies small velocity perturbations that it creates on its own, resulting in a conclusive determination that the optimized version is not string stable.

A. Future work

Of upmost concern is modifying FollowerStopper to maintain its safety but also to be string stable. Development of a more extensive command velocity filter than just a moving average could dissipate the small perturbations, resulting in a string stable controller.

Little work was done to optimize the reference velocity r. An idea could be to use smart phone traffic congestion data to calculate the length of a traffic wave and average speed of the vehicles in the wave as a reference velocity for the autonomous vehicle.

FollowerStopper was proven to be safe given dry, sunny conditions when the car is traveling straight with no incline. Future research could be done to edit the acceleration parameters based on the incline of the road, the weather conditions, and the wear of the tire. [16] shows that different weather conditions dramatically affect the coefficient of tire-road friction.

The current Simulink model has several limitations. The most concerning limitation is its difficulty in handling the AV velocity delay. Future work could attempt to use a different vehicle dynamics model or an entire new platform that could better incorporate delay.

Use of LiDAR as the main sensor input is limiting due to the short range of LiDAR. Future work could test the accuracy and usability of relying on radar or some other longer-range sensor so that the vehicle can safely travel faster than the current maximum speed of $v_{AVsafe} = 13.69$ m/s. A redevelopment of ξ_2 to a lesser value could also increase the maximum speed.

References

[1] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S.-i. Tadaki, and S. Yukawa, “Traffic jams without bottlenecks–experimental evidence for the physical mechanism of the formation of a jam,” *New journal of physics*, vol. 10, no. 3, p. 033001, 2008.

[2] R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, H. Pohlmann, F. Wu, B. Piccoli, *et al.*, “Dissipation of stop-and-go waves via control of autonomous vehicles: Field experiments,” *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 205–221, 2018.

[3] T. Litman, *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada, 2017.

[4] R. Bhadani, M. Bunting, B. Seibold, R. Stern, S. Cui, J. Sprinkle, B. Piccoli, and D. B. Work, “Real-time distance estimation and filtering of vehicle headways for smoothing of traffic waves,” in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 280–290, ACM, 2019.

[5] R. K. Bhadani, B. Piccoli, B. Seibold, J. Sprinkle, and D. Work, “Dissipation of emergent traffic waves in stop-and-go traffic using a supervisory controller,” in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 3628–3633, IEEE, 2018.

[6] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[7] S. Feng, Y. Zhang, S. E. Li, Z. Cao, H. X. Liu, and L. Li, “String stability for vehicular platoon control: Definitions and analysis methods,” *Annual Reviews in Control*, 2019.

[8] R. K. Bhadani, J. Sprinkle, and M. Bunting, “The cat vehicle testbed: A simulator with hardware in the loop for autonomous vehicle applications,” *arXiv preprint arXiv:1804.04347*, 2018.

[9] L. L. Hoberock, “A survey of longitudinal acceleration comfort studies in ground transportation vehicles,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 99, no. 2, pp. 76–84, 1977.

[10] CarSort, “2012 ford escape hybrid.”

[11] MotorTrend, “2012 ford escape reviews.”

[12] D. Vanderwerp, “Ford escape hybrid 4wd,” December 2004.

[13] Edmunds, “Used 2012 ford escape hybrid hybrid review.”

[14] “Vehicle test book,” 2007.

[15] D. Fambro, K. Fitzpatrick, and R. Koppa, “Nchrp report 400: Determination of stopping sight distances,” *TRB, National Research Council, Washington, DC*, 1997.

[16] S. Muller, M. Uchanski, and K. Hedrick, “Estimation of the maximum tire-road friction coefficient,” *Journal of dynamic systems, measurement, and control*, vol. 125, no. 4, pp. 607–617, 2003.

[17] MathWorks, 1 Apple Hill Drive, Natick, MA 01760-2098, *Vehicle Dynamics Blockset: Reference*, version 1.2 ed., March 2019.

[18] F. Wu, R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, B. Piccoli, B. Seibold, *et al.*, “The arizona ring experiments dataset (ared),” 2018.

[19] F. Wu, R. E. Stern, S. Cui, M. L. Delle Monache, R. Bhadani, M. Bunting, M. Churchill, N. Hamilton, B. Piccoli, B. Seibold, *et al.*, “Tracking vehicle trajectories and fuel rates in phantom traffic jams: Methodology and data,” *Transportation Research Part C: Emerging Technologies*, vol. 99, pp. 82–109, 2019.

APPENDIX

A. Determination of the maximum deceleration of the lead vehicle

To determine the maximum deceleration of the lead vehicle, consider Figure 10.

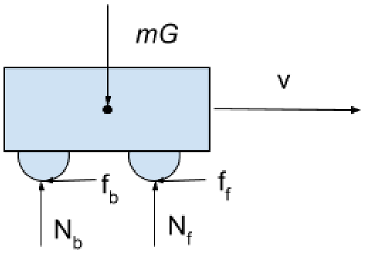


Fig. 10. Free-body diagram of a four-wheeled vehicle during deceleration.

Given the mass, m, of the vehicle and acceleration due to gravity, G, the normal force acting on the front and back tires can be expressed

$$mG = N_f + N_b = N.$$

The force of friction is proportional to the normal force by means of the coefficient of friction, μ. Assuming that the force of friction on the front and back tires is equal,

$$f_f + f_b = N_f\mu + N_b\mu = (N_f + N_b)\mu = N\mu = mG\mu$$

According to Newton's second law, the sum of the forces is proportional to the mass and the acceleration of an object. If we only consider the car's longitudinal direction, or the direction in which it is traveling at velocity, v,

$$\begin{aligned} \sum F &= ma \\ -f_f - f_b &= ma \\ -mG\mu &= ma \\ a &= -G\mu \end{aligned}$$

B. Derivation of ξ_1

Consider Figure 11, depicting two cars at times t_0, t_1, and t_2. t_0 is the time at which the relative distance is equal to the emergency breaking distance, that is, $\Delta x(t_0) = \xi_1$. t_1 occurs δ seconds after reaching ξ_1, at which point the AV will first be able to react to being within the emergency breaking distance due to the delay. t_2 is the time at which both vehicles will be stopped and the AV will be a distance ψ behind the lead vehicle. The vehicle to the left is the AV and the vehicle to the right is the human-driven lead. All positions and velocities have been labeled as functions of time, and the relative positions and relative velocities at each time can be computed according to the equations in section II.A. First we will consider the motion of the two vehicles from time t_1 to time t_2. Within this time region in a worst-case scenario, both the AV and lead will constantly decelerate at their maximum decelerations, and in the derivation we will express the lead vehicle maximum deceleration as proportional to the AV maximum deceleration with proportionality constant k. That is, suppose ka_{dmax} is the maximum deceleration of the lead. The minimum distance $\Delta x(t_1)$ which can be considered safe will assume that the AV will end up at a relative distance $\Delta x(t_2) = \psi$ from the lead vehicle. The following derivation uses this idea with the equations of motion to derive a value for $\Delta x(t_1)$.

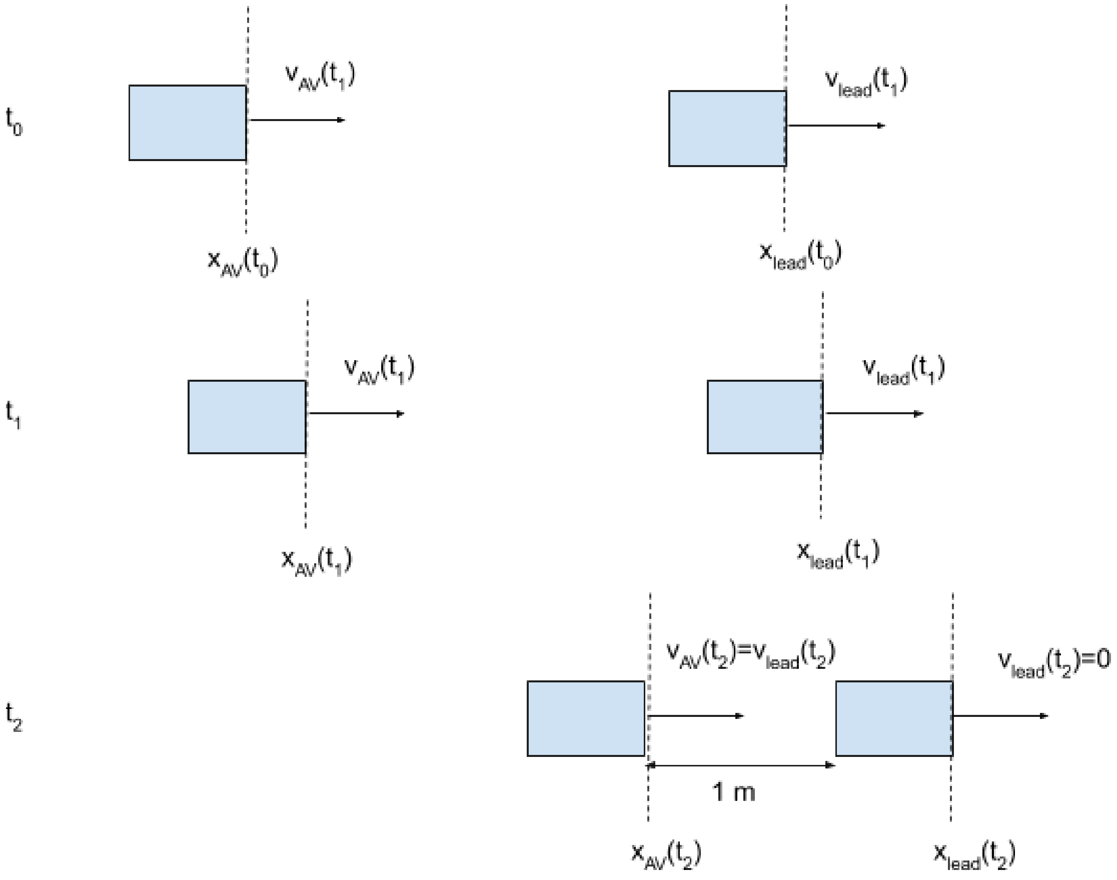


Fig. 11. Emergency breaking progression.

$$
\begin{aligned}
\text{Equation of motion}: v^2 - v_0^2 = 2a\Delta x & \\
\text{Lead vehicle}: & \\
v_{lead}(t_2)^2 - v_{lead}(t_1)^2 &= 2ka_{dmax}(x_{lead}(t_2) - x_{lead}(t_1)) \\
0 - v_{lead}(t_1)^2 &= 2ka_{dmax}x_{lead}(t_2) - 2ka_{dmax}x_{lead}(t_1)) \\
2a_{dmax}x_{lead}(t_1) - \frac{v_{lead}(t_1)^2}{k} &= 2a_{dmax}x_{lead}(t_2) \\
\text{Autonomous vehicle}: & \\
v_{AV}(t_2)^2 - v_{AV}(t_1)^2 &= 2a_{dmax}(x_{AV}(t_2) - x_{AV}(t_1)) \\
0 - v_{AV}(t_1)^2 &= 2a_{dmax}(x_{lead}(t_2) - l_{lead} - \psi - x_{AV}(t_1)) \\
2a_{dmax}(l_{lead} + \psi + x_{AV}(t_1)) - v_{AV}(t_1)^2 &= 2a_{dmax}x_{lead}(t_2) \\
\text{Setting the equations equal}: & \\
2a_{dmax}x_{lead}(t_1) - \frac{v_{lead}(t_1)^2}{k} &= 2a_{dmax}(l_{lead} + \psi + x_{AV}(t_1)) - v_{AV}(t_1)^2 \\
2a_{dmax}(x_{lead}(t_1) - x_{AV}(t_1) - l_{lead}) &= 2a_{dmax}\psi + \frac{v_{lead}(t_1)^2}{k} - v_{AV}(t_1)^2 \\
\Delta x(t_1) &= \psi + \frac{1}{2a_{dmax}}(\frac{v_{lead}(t_1)^2}{k} - v_{AV}(t_1)^2)
\end{aligned}
$$

Delay is an essential consideration because if the AV is currently at $\Delta x = \Delta x(t_1)$, it needs to initiate emergency breaking, but it will not do as such until δ seconds have passed, where δ is the delay of the system. Consider the time t_0 at which point the AV recognizes that it must send an emergency braking command in order to initiate emergency braking at the instant it reaches t_1 from the above example. When considering the emergency stopping distance ξ_1, we must plan for the worst case scenario. The worst case scenario is that at time t_0 the lead car begins to accelerate before coming to an immediate stop because the AV might think that it can begin to accelerate as well. Though from time t_0 to time t_1 the accelerations might fluctuate, a worst-case scenario would involve maximum acceleration of the AV and maximum deceleration of the lead. The following derivation, based off of a derivation in [4], incorporates the delay to determine what a safe value of ξ_1 will equal such that if $\Delta x = \xi_1$, the AV will initiate hard braking at exactly $\Delta x = \Delta x(t_1)$. In this setup, $\xi_1 = \Delta x(t_0)$.

$$\begin{aligned}
\text{Equation of motion}: x &= x_0 + v_0 t + \frac{1}{2}at^2 \\
\text{Distance traveled during delay}: & \\
x_{lead}(t_1) &= x_{lead}(t_0) + v_{lead}(t_0)\delta + \frac{1}{2}a_{lead}(t_0)\delta^2 \\
x_{AV}(t_1) &= x_{AV}(t_0) + v_{AV}(t_0)\delta + \frac{1}{2}a_{AV}(t_0)\delta^2
\end{aligned}$$

Worst-case scenario, $a_{lead}(t_0) = a_{dmaxLEAD} = ka_{dmax}$, $a_{AV}(t_0) = a_{max}$

$$\begin{aligned}
\text{Incorporating relative distance}: & \\
\Delta x(t_1) &= \Delta x(t_0) + (v_{lead}(t_0) - v_{AV}(t_0))\delta + \frac{1}{2}(ka_{dmax} - a_{max})\delta^2 \\
\xi_1 = \Delta x(t_0) &= \Delta x(t_1) - (v_{lead}(t_0) - v_{AV}(t_0))\delta - \frac{1}{2}(ka_{dmax} - a_{max})\delta^2 \\
&= \psi + \frac{1}{2a_{dmax}}(\frac{v_{lead}(t_1)^2}{k} - v_{AV}(t_1)^2) \\
&\quad -(v_{lead}(t_0) - v_{AV}(t_0))\delta - \frac{1}{2}(ka_{dmax} - a_{max})\delta^2
\end{aligned}$$

The above equation for ξ_1 guarantees that a crash is impossible and that the AV will never come within 1 meter of the lead, assuming that there are no unexpected obstacles that appear between the AV and the lead. However, when the AV arrives at time t_0, it will not know $v_{lead}(t_1)$ or $v_{AV}(t_1)$. Both are best approximated using the equation of motion, $v = v_o + at$, with a set to a safe value. In a worst-case scenario, the v_{lead} will be slower than expected and v_{AV} will be faster than expected. Therefore safe acceleration values will be the same as indicated in the previous derivation. Because all time-dependent values will be expressed at the same time t_0, the notation can be dropped, giving

$$\begin{aligned}
\xi_1 &= \psi + \frac{1}{2ka_{dmax}}((v_{lead} + ka_{dmax}\delta)^2 - k(v_{AV} + a_{max}\delta)^2) - (v_{lead} - v_{AV})\delta - \frac{1}{2}(ka_{dmax} - a_{max})\delta^2 \\
&= \psi + \frac{1}{2ka_{dmax}}(v_{lead}^2 + 2v_{lead}ka_{dmax}\delta + k^2a_{dmax}^2\delta^2 - kv_{AV}^2 - 2kv_{AV}a_{cmft}\delta - ka_{cmft}^2\delta^2) \\
&\quad -v_{lead}\delta + v_{AV}\delta - \frac{ka_{dmax}\delta^2}{2} + \frac{a_{max}\delta^2}{2} \\
&= \psi + \frac{v_{lead}^2}{2ka_{dmax}} + v_{lead}\delta + \frac{ka_{dmax}\delta^2}{2} - \frac{v_{AV}^2}{2a_{dmax}} - \frac{v_{AV}a_{cmft}\delta}{a_{dmax}} - \frac{a_{max}^2\delta^2}{2a_{dmax}} - v_{lead}\delta + v_{AV}\delta - \frac{ka_{dmax}\delta^2}{2} + \frac{a_{max}\delta^2}{2} \\
&= \psi + \frac{1}{2ka_{dmax}}(v_{lead}^2 - kv_{AV}^2) + v_{AV}(1 - \frac{a_{max}}{a_{dmax}})\delta + \frac{a_{max}}{2}(1 - \frac{a_{max}}{a_{dmax}})\delta^2.
\end{aligned}$$

Finally, it must be noted that it is never desirable for ξ_1 to drop below ψ, the minimum acceptable relative distance. In the above equation for ξ_1, the first, third, and fourth terms will always be positive, but the second term could be negative. To assure that ξ_1 will remain greater than ψ, the second term must be set equal to 0 if it is negative, giving a final solution

$$\begin{aligned}
\xi_1 &= \psi + \Delta v^{**} + v_{AV}(1 - \frac{a_{max}}{a_{dmax}})\delta + \frac{a_{max}}{2}(1 - \frac{a_{max}}{a_{dmax}})\delta^2 \\
\Delta v^{**} &= max(0, \frac{1}{2ka_{dmax}}(v_{lead}^2 - kv_{AV}^2)).
\end{aligned}$$

C. Moving average plots

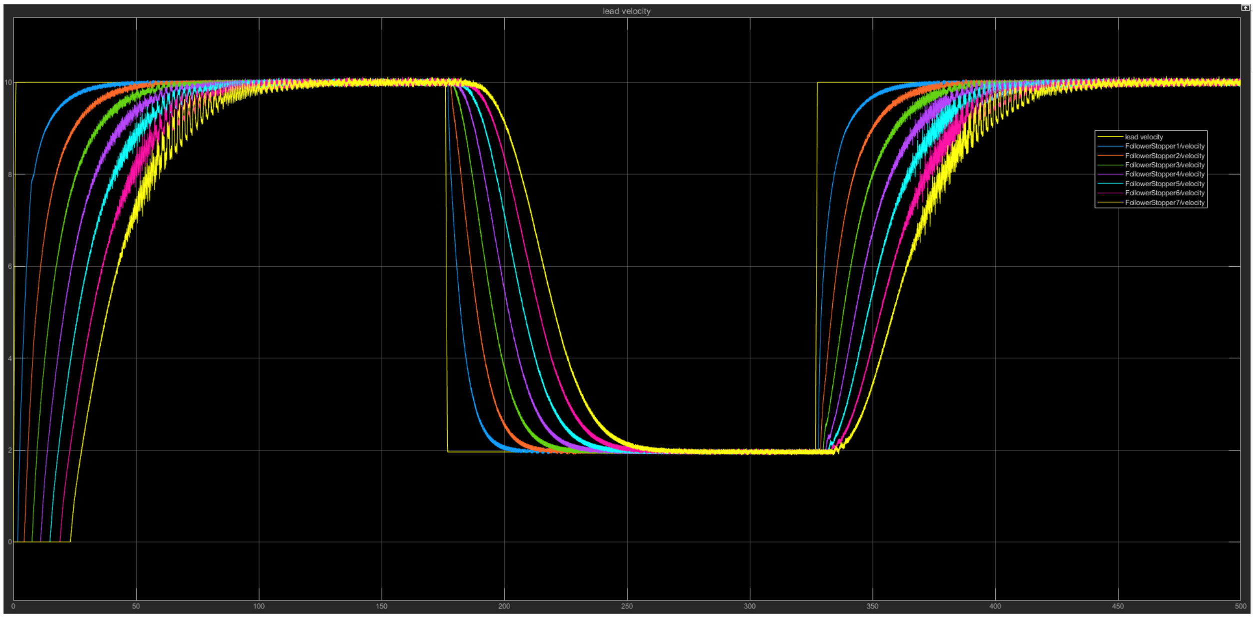


Fig. 12. Step Test with no moving average.

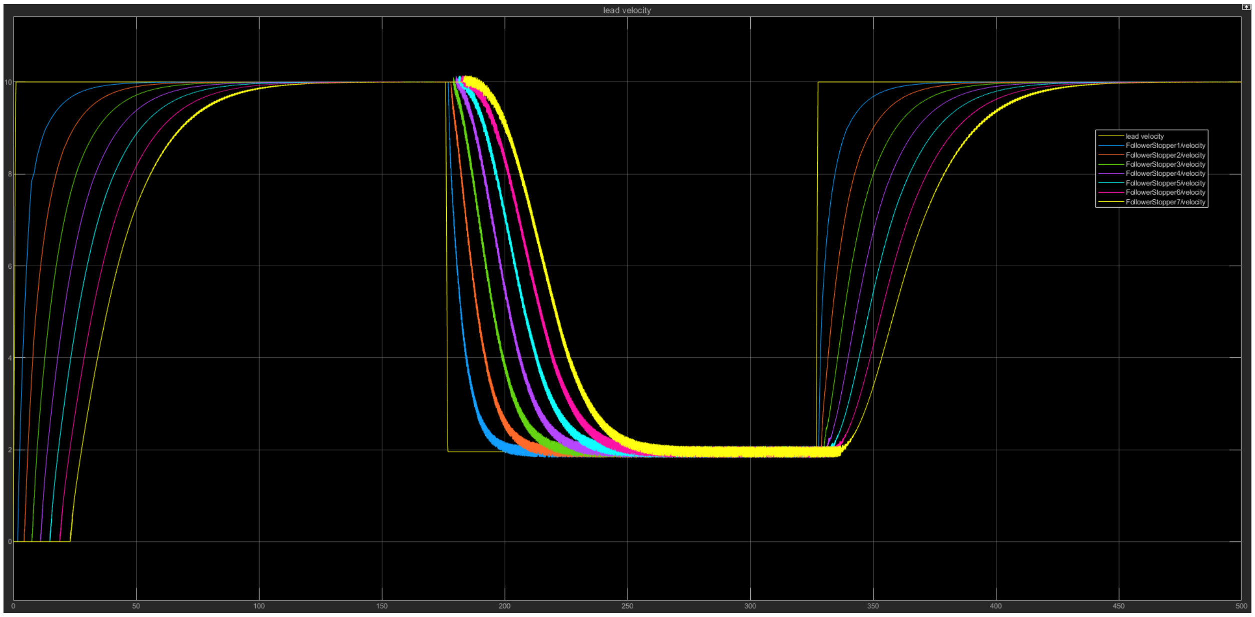


Fig. 13. Step Test with a moving average of 5.

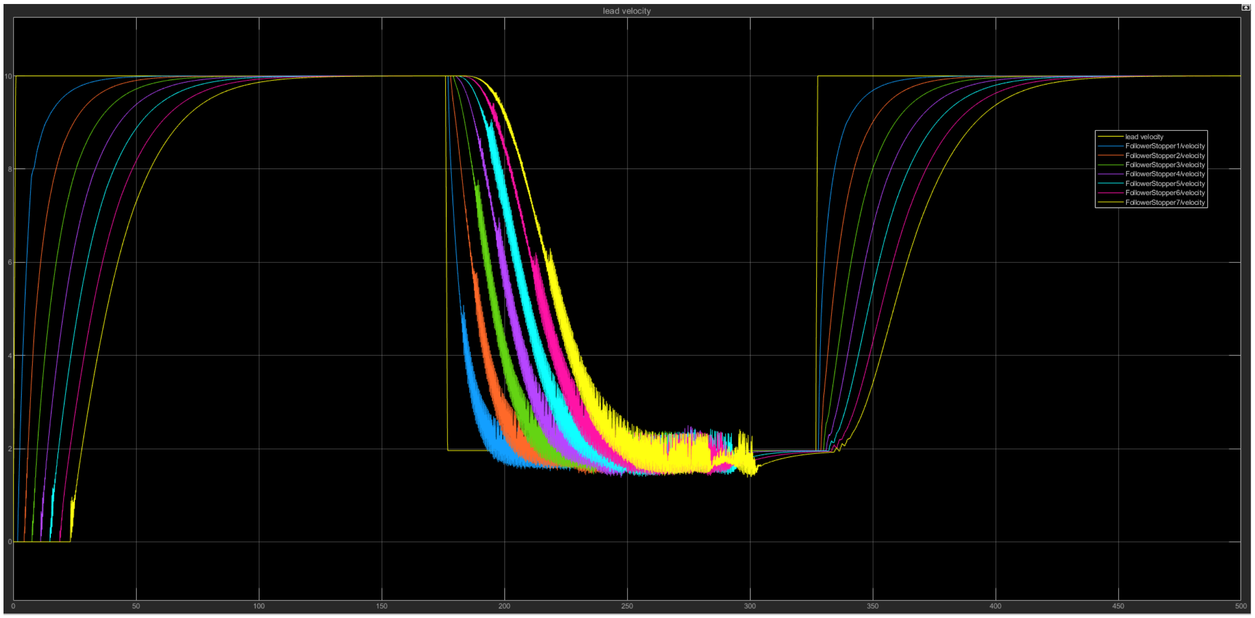


Fig. 14. Step Test with a moving average of 25.

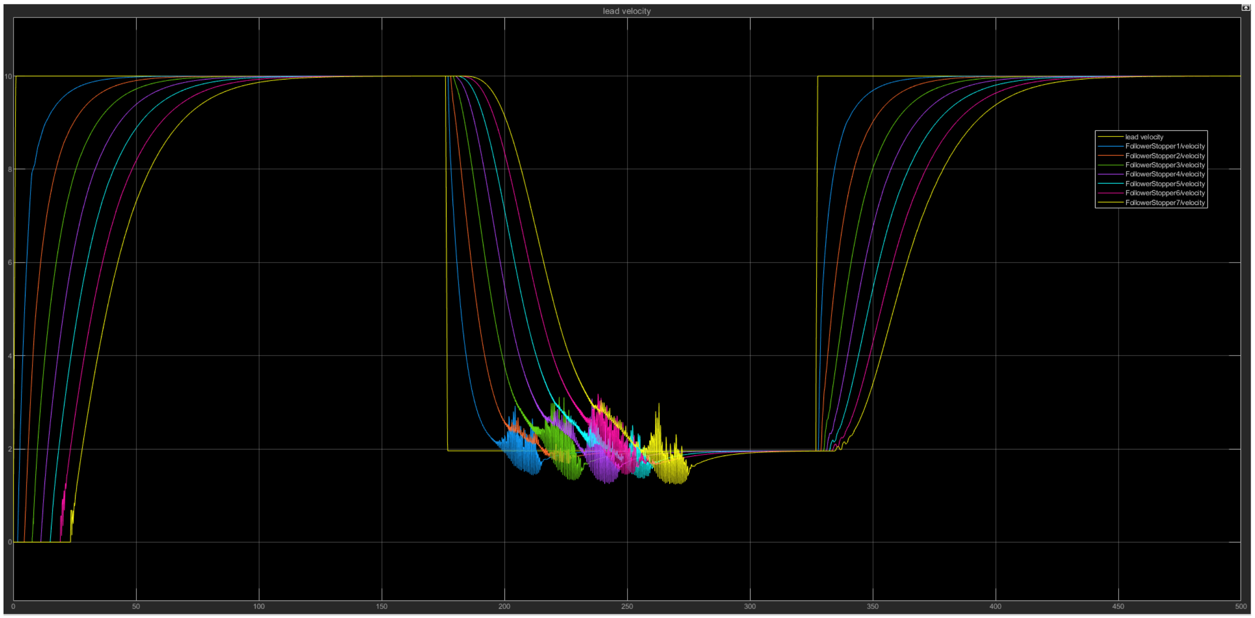


Fig. 15. Step Test with a moving average of 50.

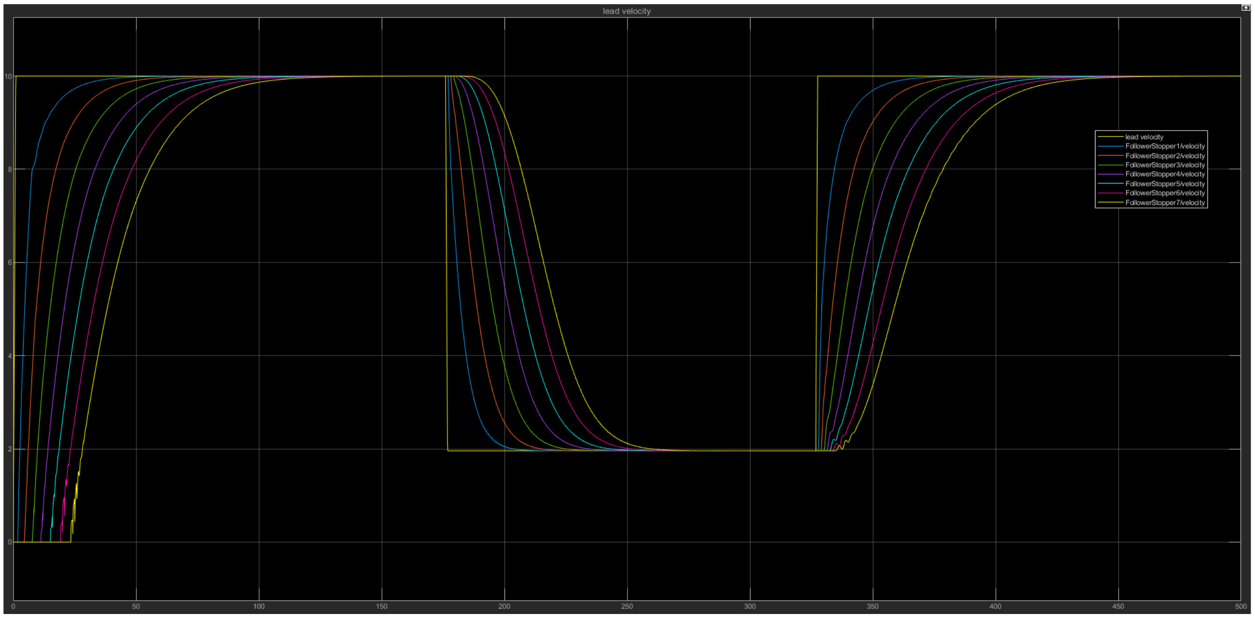


Fig. 16. Step Test with a moving average of 75.

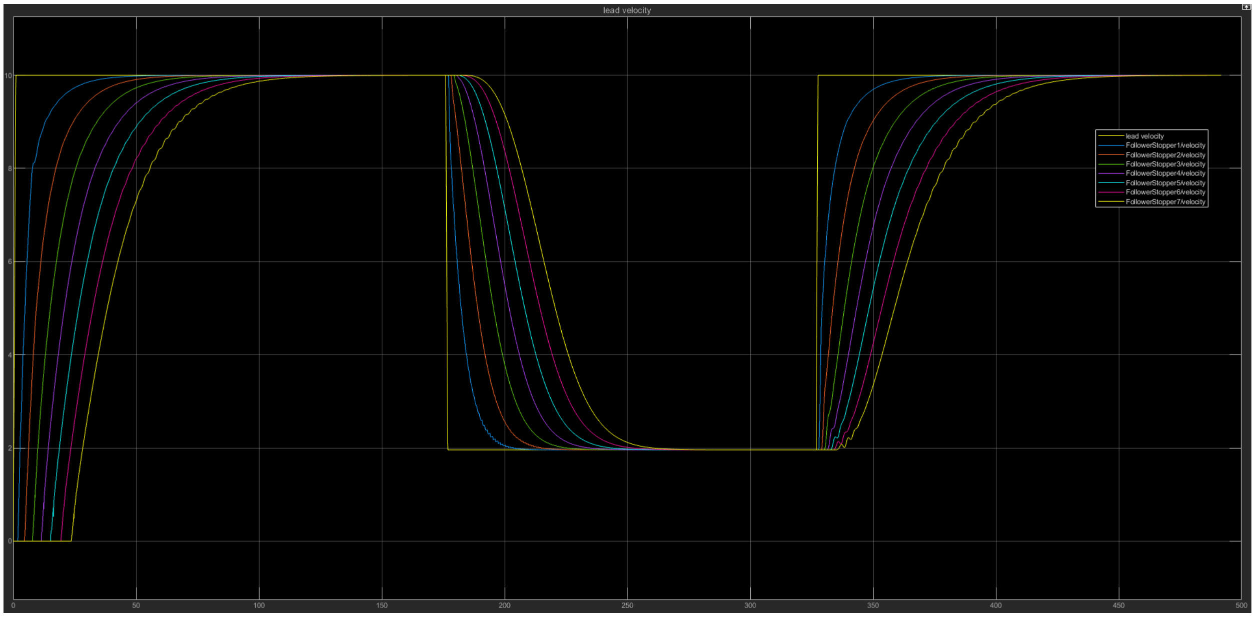


Fig. 17. Step Test with a moving average of 100.

D. *FollowerStopper safety plots*

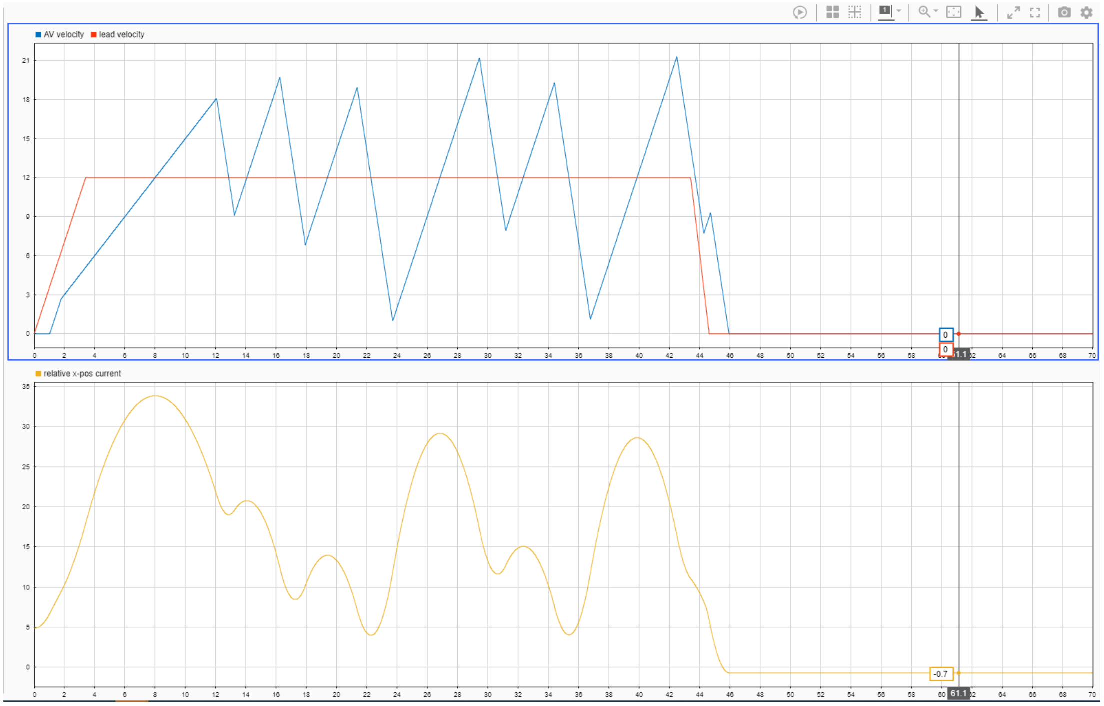


Fig. 18. Original FollowerStopper in Safety Test 1.

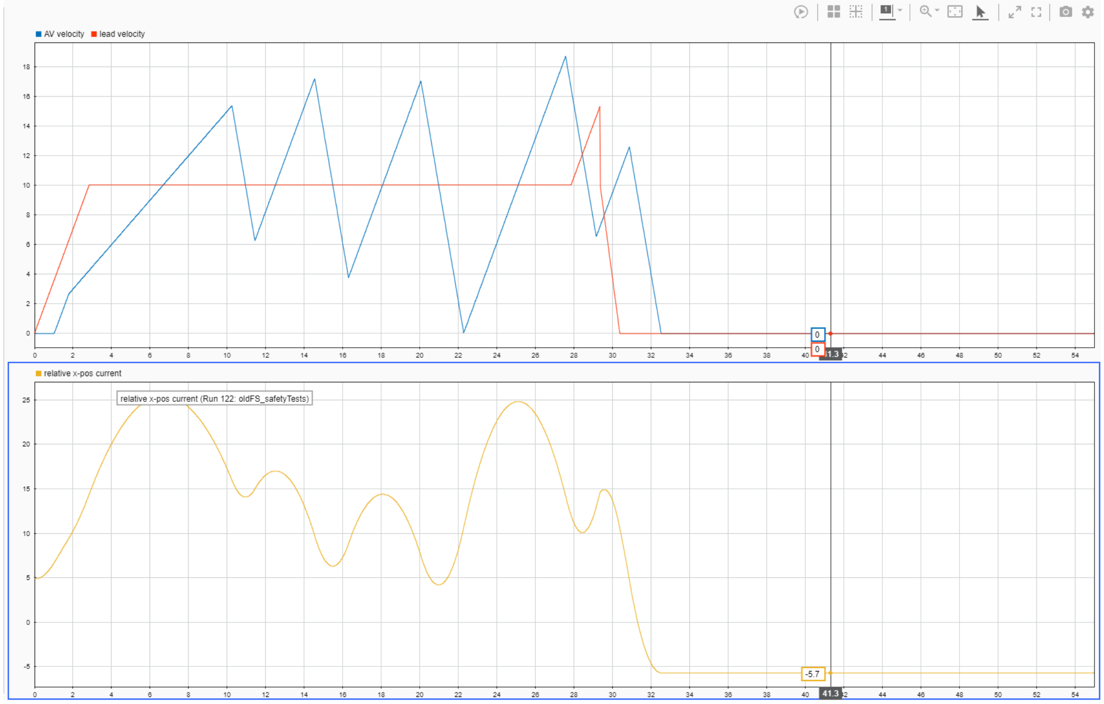


Fig. 19. Original FollowerStopper in Safety Test 2.

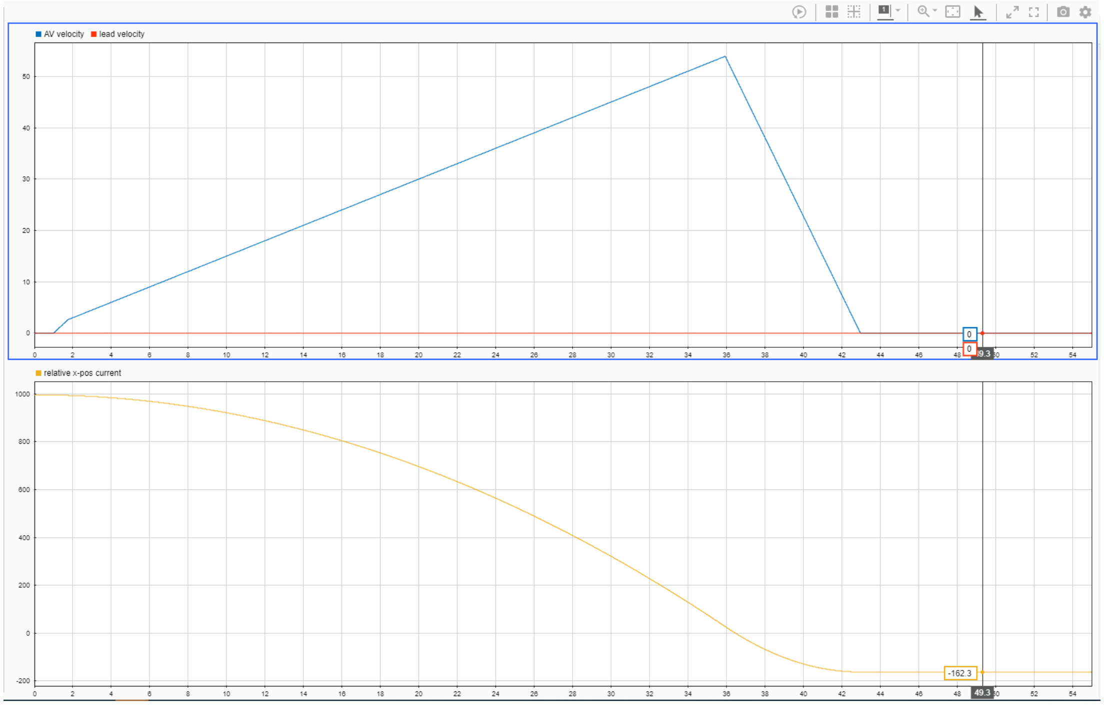


Fig. 20. Original FollowerStopper in Safety Test 3.

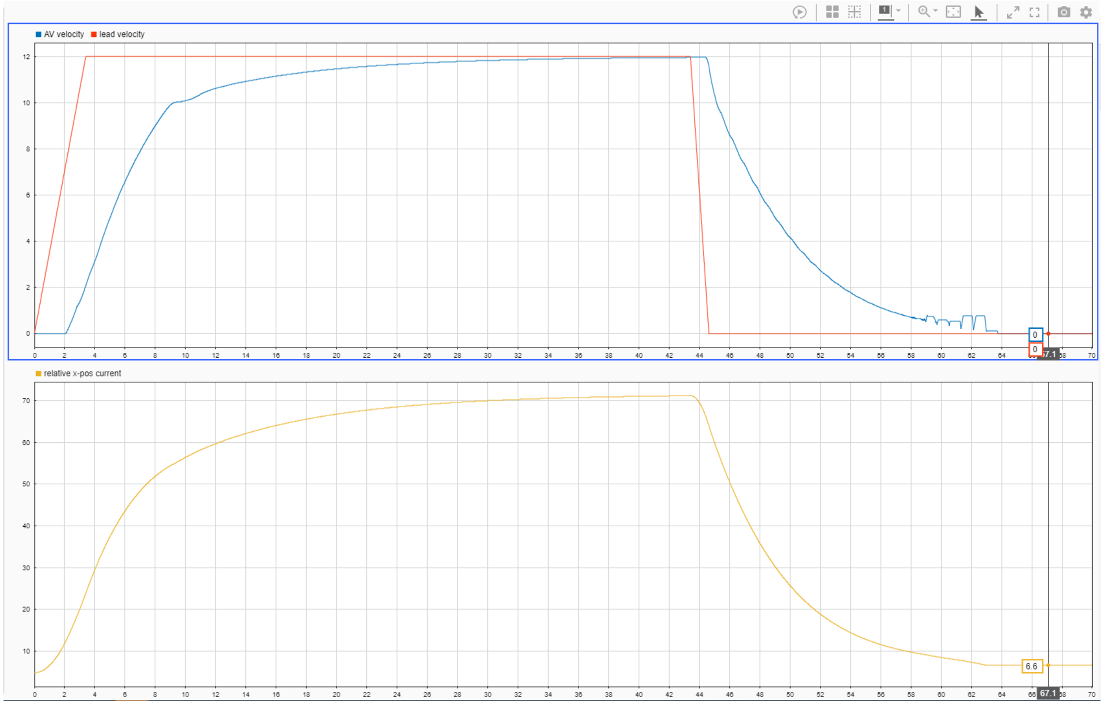


Fig. 21. New FollowerStopper in Safety Test 1.

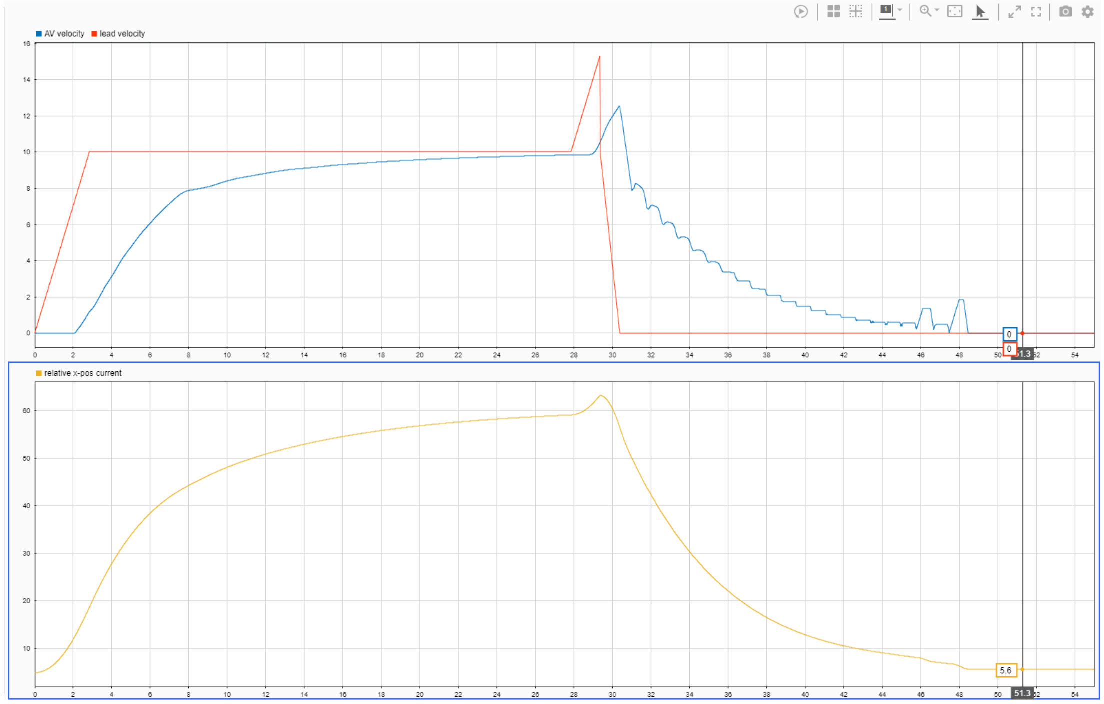


Fig. 22. New FollowerStopper in Safety Test 2.

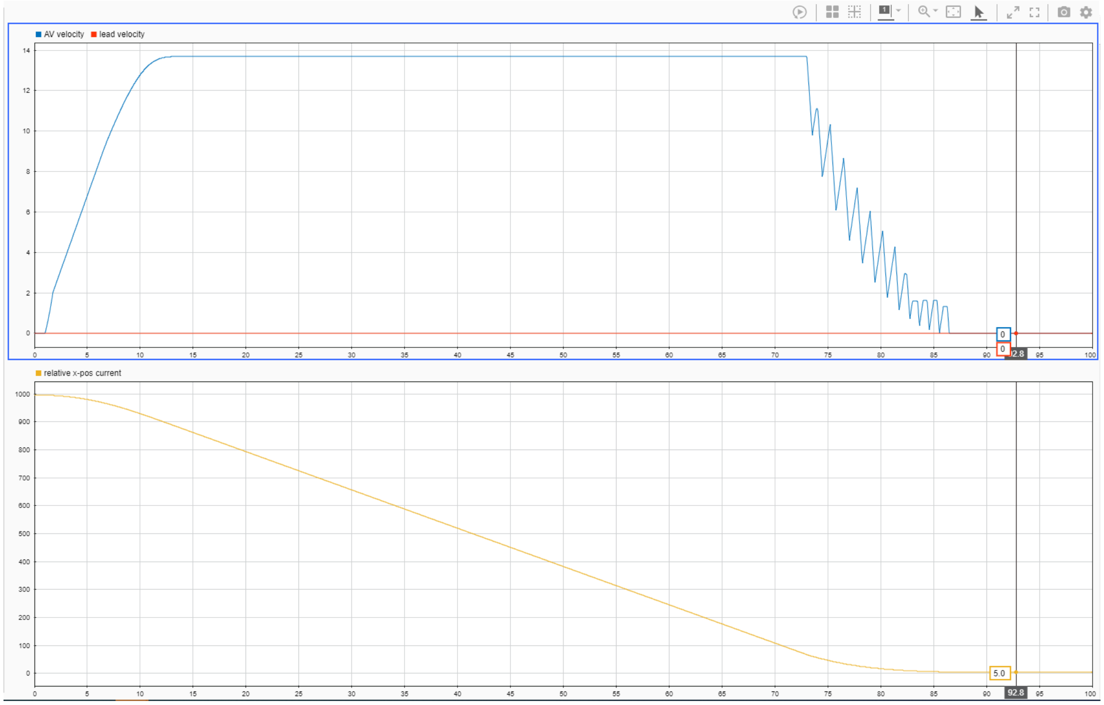


Fig. 23. New FollowerStopper in Safety Test 3.

E. FollowerStopper string stability plots

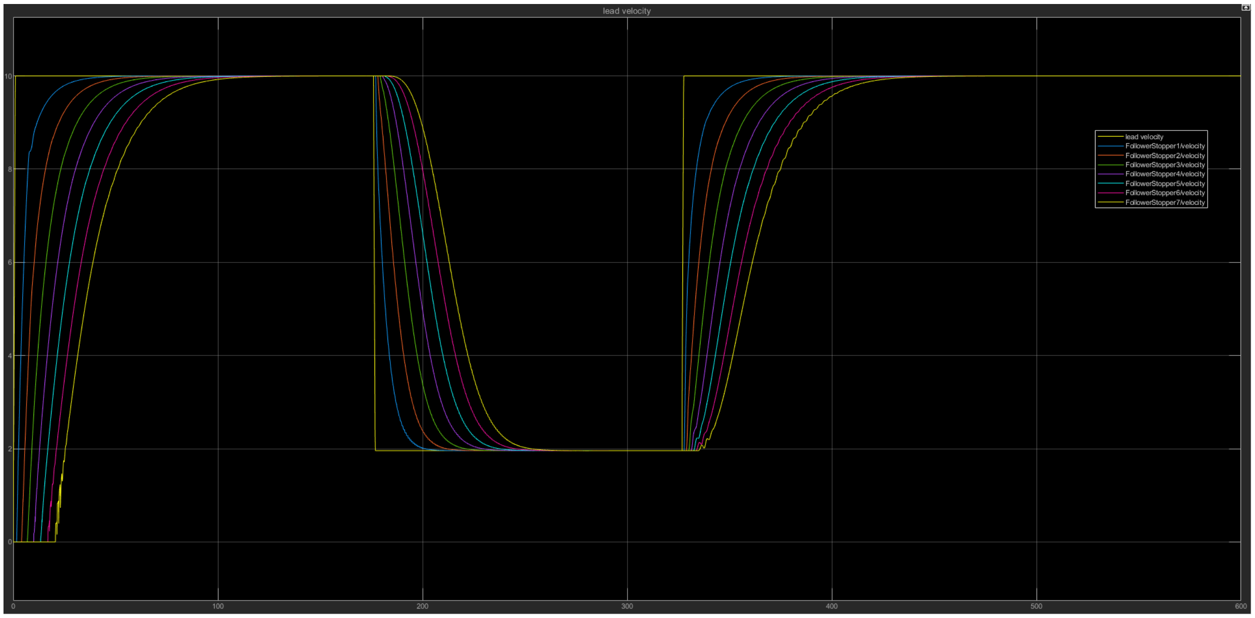


Fig. 24. Velocity vs time of seven-car FollowerStopper model in Step Test with bad reference velocity.

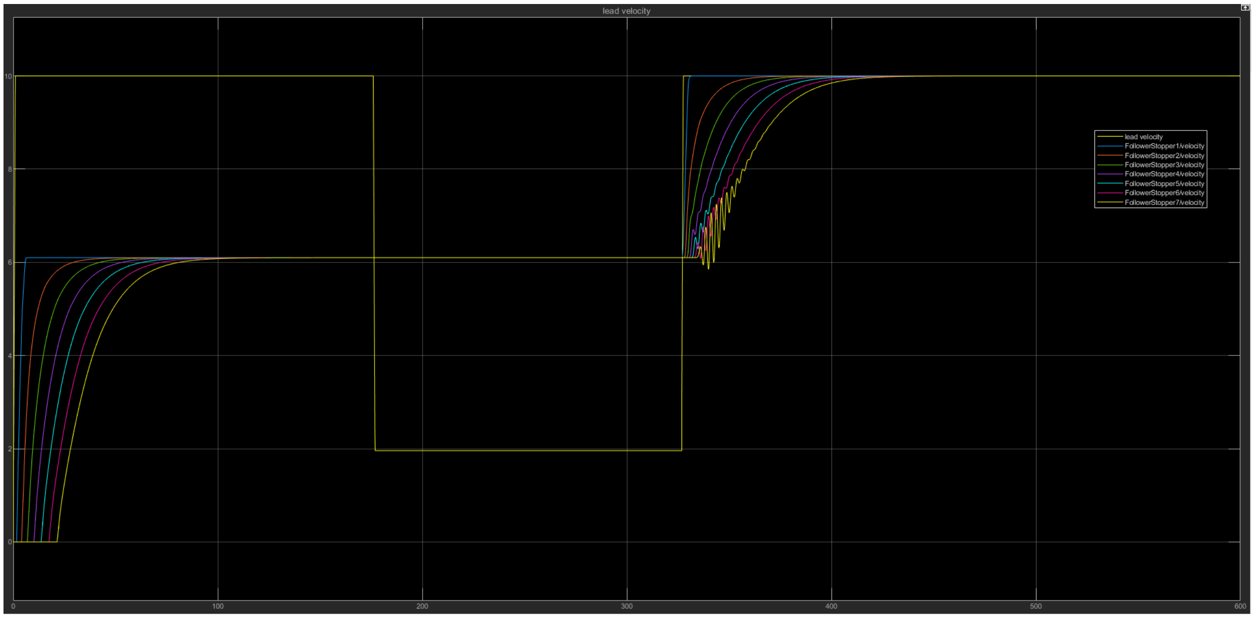


Fig. 25. Velocity vs time of seven-car FollowerStopper model in Step Test with good reference velocity.

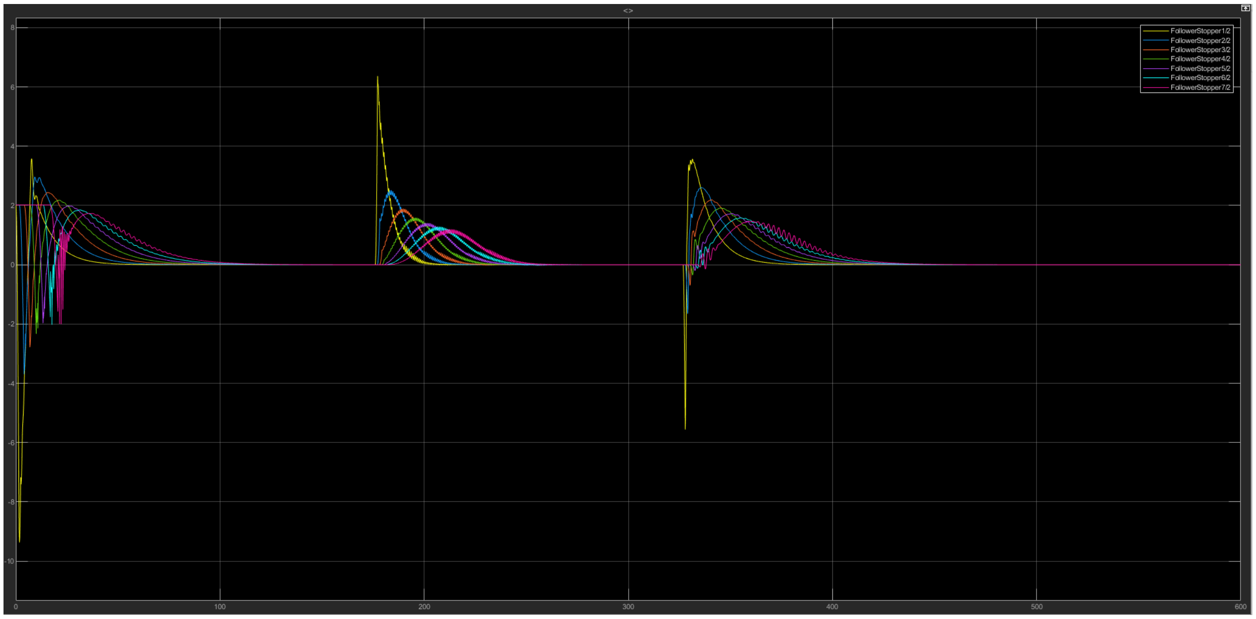


Fig. 26. Spacing error vs time of seven-car FollowerStopper model in Step Test with bad reference velocity.

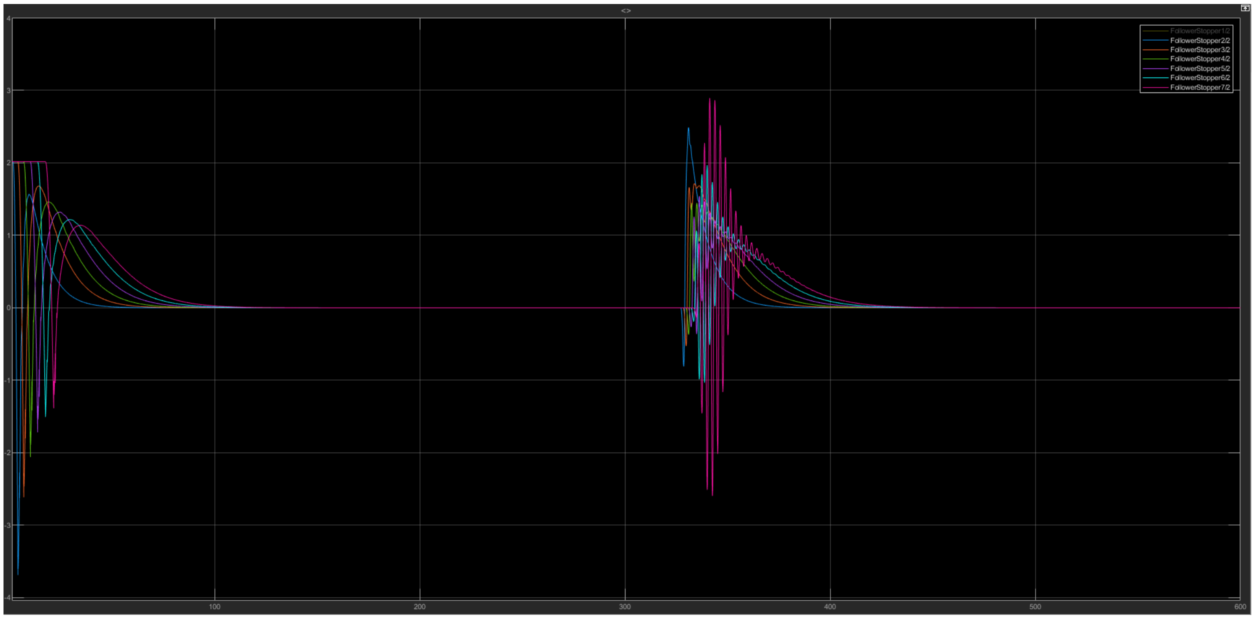


Fig. 27. Spacing error vs time of seven-car FollowerStopper model in Step Test with good reference velocity.

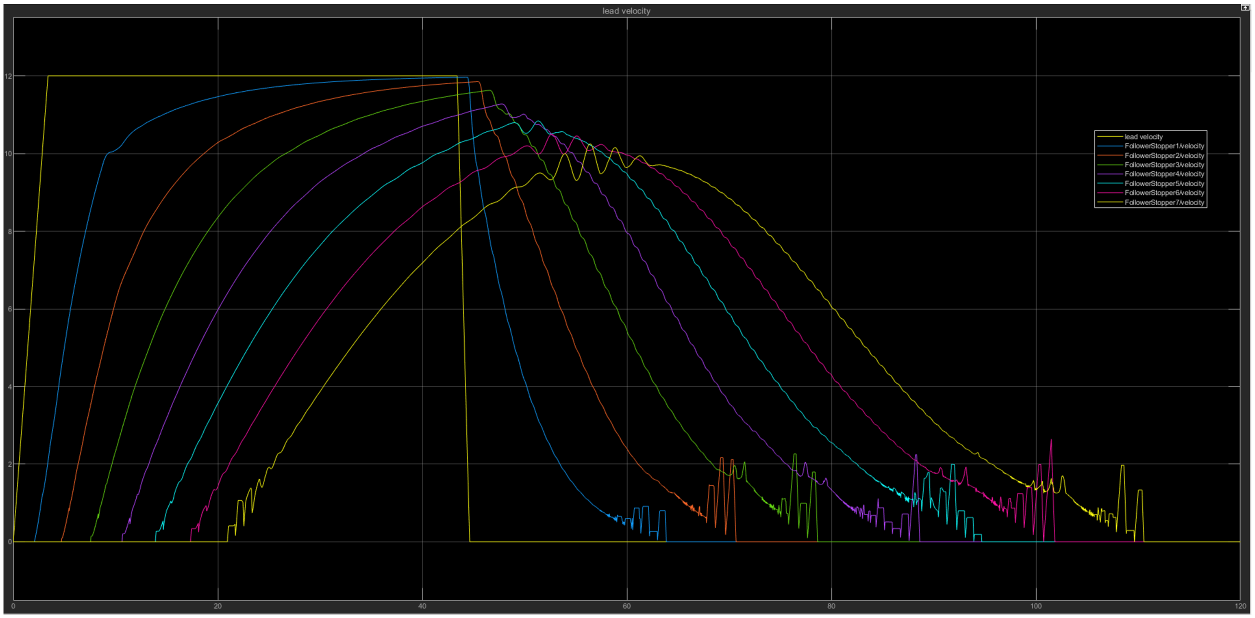


Fig. 28. Velocity vs time of seven-car FollowerStopper in Safety Test 1 with bad reference velocity.