Verification of a One-Dimensional Adaptive Cruise Control Model With Applications in Vehicle Platooning

Ayana A. Wild, Nathaniel Hamilton, and Tianshu Bao

***Abstract*—Traffic is a growing problem in the US and the number of *phantom traffic jams* is growing in tandem. One solution to reducing their number and their lasting effects involves introducing autonomous vehicle platoons to our roadways. However, autonomous platoons are complicated systems being used in a high risk situation so their dynamics need to be formally verified as safe in order to be used on the roads. In this paper we build a platoon model by stacking Adaptive Cruise Control models and use *nuXmv* to formally verify safety and liveness specifications. While the two car model is verifiably safe, issues occur when the number of vehicles is increased. We use a simulator to visualize and analyze how these issues arise and future solutions to address them.**

***Index Terms*—Verification, nuXmv, Adaptive Cruise Control, Car Following, Platoon**

I. Introduction

Traffic is a growing problem in the US [1], [2], which has led to more research into one traffic phenomena referred to as *phantom traffic jams*. These *phantom traffic jams* are an experimentally reproducible phenomenon, as demonstrated in different experiments [3]–[7]. Common wave triggers include lane changing [8]–[10], but they can even be generated in the absence of any lane changes, bottlenecks, merges, or changes in grade. In such cases, they are caused by the collective dynamics of the drivers on the road [3], [4], [7].

One such solution for resolving these phantom traffic jams is vehicle platoons. A vehicle platoon is a group of vehicles that safely travel close together, usually at high speeds, as shown in Figure 1. It is one of many ideas being discussed for optimal control with self-driving cars for its various benefits including reduced emissions, increased fuel efficiency, and more efficient traffic flow [11]. One way of implementing this beneficial platoon structure in cars today is through the use of Adaptive Cruise Control (ACC). ACC, a commonly-used control technique in current commercial vehicles, is used to increase the safety of vehicles by modifying the velocity control already developed for cruise control [12], [13]. It can be thought of as a simplified version of car following, where steering is not involved.

Car following and variations of it have become a popular method for constructing autonomous ground vehicle control, as shown in [14]–[16]. Car following is the idea that a leader car, car_l, is in front of a follower car, car_f, and there

Department of Electrical and Computer Engineering, Vanderbilt University, Nashville, TN, 37235 USA. e-mail: ayana.a.wild@vanderbilt.edu, nathaniel.p.hamilton@vanderbilt.edu, and tianshu.bao@vanderbilt.edu.

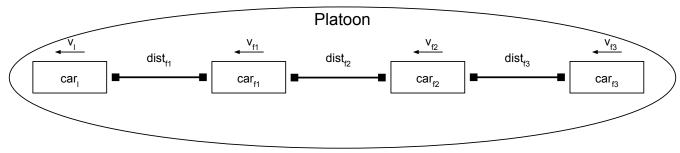

Fig. 1. Platoon illustration.

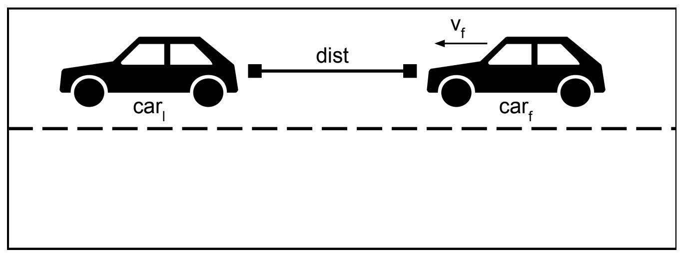

Fig. 2. Car following illustration.

is a measurable distance between them represented by the value $dist$. The distance affects the behavior of the controller based on a set of n thresholds, $thresh$ such that $thresh_0 > thresh_1 > thresh_2 > thresh_3 > ... > thresh_n > 0$, within which the distance can waver. The behavior of car_f is modified through the control of its velocity, v_f.

However, car following is complex and these complex models can have bugs, which result in automobile crashes. As a result, formal verification of car following models is necessary to ensure the safety of passengers. In this project, we aim to model the problem of car following, simplified to the Adaptive Cruise Control application, apply it to a platoon, and verify that our model satisfies the following safety and liveness specifications using *nuXmv*.

$$\text{Safety: } \Box\neg(mode = Danger)$$
$$\text{Liveness: } \Box\neg(mode = Danger) \rightarrow \big(\Box\Diamond(mode = Keeping) \vee \Diamond\Box(mode = Chasing)\big)$$

The safety specification states that the vehicle is always not in the *Danger* mode, meaning the system is always operating outside of a crashing scenario, i.e. *safe*. The liveness specification states that as long as the system remains safe, then the vehicle is either repeatedly in the *Keeping* mode or persistently in the *Chasing* mode. The caveat of remaining safe is necessary because the *Danger* mode stops all motion and acts as a trap.

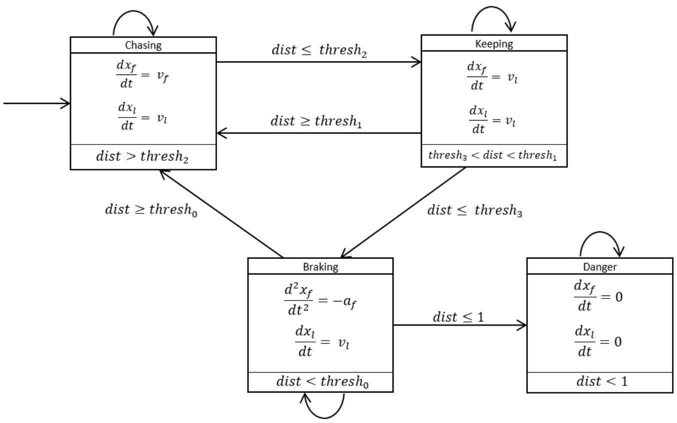

Fig. 3. Diagram illustrating our hybrid system model for car following.

So, if the system ever entered the *Danger* mode, it would not be able to leave, making the other statements false.

II. Mathematical Model

For this paper, we chose to augment the Adaptive Cruise Control model presented in [17] by adding one extra mode to model what happens if a collision should occur. Therefore, our model is a hybrid automaton. A hybrid automaton is a *generalized finite-state automaton* that is equipped with continuous variables. The syntax of hybrid automata is defined as follows.

Definition II.1 (Hybrid Automaton [18]). A hybrid automaton is a tuple $\langle Q, E, X, Init, I, f, G, R \rangle$ where:

- Q is a set of finite q_is that represents modes.
- $E \subseteq Q \times Q$ is a finite set of edges that represent changes of control modes.
- X is the set of variables with $X = \mathbb{R}^n$.
- $Init$ is the initial set.
- I is a set of invariants defined for each mode q_i.
- f is the flow function.
- G is the guard predicates.
- R is reset map.

A. Single Follower Model

In their example, the adaptive cruise control can be divided into 3 stages(modes): *Chasing*, *Keeping*, and *Braking*. Ours adds one extra mode, *Danger* that stops motion once a collision occurs. Each mode is governed by a set of differential equations and the system keeps moving between these modes. Speed of the leader vehicle and the following vehicle will be considered and also the distance between them $dist$. The diagram for the hybrid system model we created is shown in Figure 3.

The adaptive cruise control is composed of three stages:

1) **Chasing:** In this stage, $dist > thresh_2$, and the follower car, car_f, will try to catch the leader car, car_l, at speed $v_f > v_l$. So the perturbed motion of car_f is governed by $\frac{dx_f}{dt} = v_f$. When $dist \leq thresh_2$, the system transitions to the *Keeping* mode.
2) **Keeping:** In this stage $thresh_3 \leq dist \leq thresh_1$ and car_f will try to maintain its velocity at v_l. When $dist \leq thresh_3$, the system transitions to the *Braking* mode. When $dist \geq thresh_1$, the system cycles back to the *Chasing* mode.
3) **Braking:** If $dist < thresh_0$, then car_f will decelerate according to some prescribed procedure until $dist \geq thresh_0$. We take into consideration the addition of noise even in the braking phase. Once $dist \geq thresh_0$, the braking transitions to the mode of chasing the leader car.
4) **Danger:** If $dist \leq 1$, the system will enter the danger mode and car_f and car_l will stop immediately. The system will stay in the mode forever and never leave.

Instead of using a determinate function to describe the behavior of vehicles, adding a random element is also a feasible way. In diagram (3), the ODE dynamic could be replaced by a stochastic equation dynamic which brings more possibilities and varieties. A possible framework that could contain the stochastic dynamic is a stochastic hybrid system framework as shown in [17]. This can be a potential work in the future direction.

B. Platoon Model

The model described in Figure 3 only defines 2 vehicles. If we consider more vehicles on the road, the same strategy can be applied to 3^{rd} and 4^{th} vehicle. One possible way is to add more instances of the hybrid automaton. A system consisting of n vehicles has $n-1$ distance variables and 5 thresholds associated with each distance variable. Therefore we have 4^n different mode combinations in this system which means the number grows rapidly. This typical phenomenon is called *State Space Explosion problem.*

Table I shows the number of distinct mode combinations as we consider a platoon.

TABLE I
Relation between vehicles and mode combinations.

Number of Vehicles	Number of Mode Combinations
2	4
3	16
4	64
5	256

In the model we assume the leading vehicle maintains a constant speed at each stage. In real situation, the front vehicle can brake and cause the following vehicle to decelerate. Therefore, a necessary extension is required to accommodate this kind of situation.

A possible way to fix this is to connect the behavior of each consecutive vehicle pairs. Given the scenario that the leading vehicle is braking, the following vehicle start to brake until the value $dist$ is large enough. For simplicity, we replaced the velocity of leading vehicle by a changing number and provide the corresponding simulation.

In braking mode, if the speed of leading vehicle is less than following vehicle, the variable $dist$ can reach a even smaller number and cause the system to enter danger mode. In danger mode, both vehicle stop immediately and the system malfunctions. We should aim to avoid this type of situation.

The changing speed strategy can also be applied to the platoon scenario. The leading vehicle of the platoon moves at an inconsistent speed and cause the following vehicle adjusting their speed in order to maintain a safer distance. We have to emphasize that if one pair of vehicle enters the danger mode, all other vehicle behind them will stop eventually because the distance measured are getting smaller.

III. Implementation

We chose to use *nuXmv* to verify our model. *nuXmv* is a symbolic model checker for analyzing both finite-state and infinite-state synchronous systems [19].

Because *nuXmv* cannot handle hybrid automata, we encoded the model as a Mealy machine with some additional assumptions. In our future work, we plan to implement the model in a tool that can handle hybrid automata such as *Flow** or *SpaceEx*, but for the purpose of this paper, we thought the modifications needed to accommodate *nuXmv* would not negate any properties we found to be true. Instead of using differential equations, each transition represents 1 second of simulated time. For example, the distance between vehicles is calculated as $next(dist) = dist + next(v_l) - next(v_f)$.

A. Single Car Follower

In the *nuXmv* model, all velocities are held to be strictly non-negative, i.e. greater than or equal to 0, and less than $100m/s \approx 224mph$. Holding the velocities to be non-negative guarantees the vehicles cannot go in reverse which the model was not designed to handle. The upper bound was selected arbitrarily and changing it to a smaller value could reduce the state space. The rest of the model is defined as a series of transitions between modes with corresponding actions. Each transition in Figure 3 is represented as a transition statement with the desired dynamical change occurring after the new mode has been determined. For example, the transition from *Chasing* to *Keeping* is written out as:

$$\begin{aligned}&(mode = Chasing \land dist \le thresh_2) \rightarrow \\ &\quad (\circ(mode) = Keeping \\ &\quad \land \circ(v_l) = v_l \\ &\quad \land \circ(v_f) = v_l \\ &\quad \land \circ(dist) = dist + \circ(v_l) - \circ(v_f) \\ &\quad)\end{aligned}$$

For further illustration, the transition from *Braking* to *Braking* is written out as:

$$\begin{aligned}&(mode = Braking \land 1 < dist < thresh_0) \rightarrow \\ &\quad (\circ(mode) = Braking \\ &\quad \land \circ(v_l) = v_l \\ &\quad \land \circ(v_f) = v_f - a_f \\ &\quad \land \circ(dist) = dist + \circ(v_l) - \circ(v_f) \\ &\quad)\end{aligned}$$

The transition space for the model is written out as an intersection of each of these transition statements, requiring all of them to hold true at all times. The initial state is also defined in the nuXmv file. The initial state is written out as the initial mode must be *Chasing* and the initial distance between vehicles must be greater than or equal to $thresh_0$ to ensure the vehicles really are starting in the *Chasing* mode. All other variables are undefined, allowing for a more robust analysis of the system.

B. Three-Vehicle Platoon

The original model was applied to each successive pair of vehicles in the platoon. A platoon of three vehicles $(lead, 1, 2)$ was implemented. Each vehicle has an independent velocity (v_{lead}, v_1, v_2). Both pairs of vehicles (v_{lead}, v_1) and (v_1, v_2) each have a mode $(mode_1, mode_2)$ and a distance $(distance_1, distance_2)$. The transitions are similar to those for a single pair just adapted to two pairs of vehicles. The safety specifications for the three vehicle platoon are:

$$\Box\neg(mode_1 = Danger)$$
$$\Box\neg(mode_2 = Danger)$$

The liveness specifications are:

$$\begin{aligned}\Box\neg(mode_1 = Danger) \rightarrow \big(&\Box\Diamond(mode_1 = Keeping) \\ &\lor \Diamond\Box(mode_1 = Chasing)\big)\end{aligned}$$

$$\begin{aligned}\Box\neg(mode_2 = Danger) \rightarrow \big(&\Box\Diamond(mode_2 = Keeping) \\ &\lor \Diamond\Box(mode_2 = Chasing)\big)\end{aligned}$$

C. Four-Vehicle Platoon

Additionally, we implemented a platoon consisting of four vehicles $(lead, 1, 2, 3)$, each with their own velocity $(v_{lead}, v_1, v_2, v_3)$. Each pair of vehicles $\{(v_{lead}, v_1), (v_1, v_2), (v_2, v_3)\}$ have individual modes $(mode_1, mode_2, mode_3)$ and distances $(distance_1, distance_2, distance_3)$. Transitions for each vehicle pair are defined as shown in Figure 3. The safety specifications for each pair are extended to be:

$$\Box\neg(mode_1 = Danger)$$
$$\Box\neg(mode_2 = Danger)$$
$$\Box\neg(mode_3 = Danger)$$

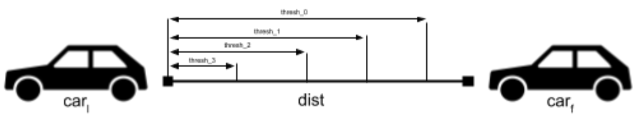

Fig. 4. The relationship of the thresholds for the car following implementation.

TABLE II
THRESHOLD VALUES

Trial #	Thresh$_0$	Thresh$_1$	Thresh$_2$	Thresh$_3$	Safety	Liveness	Sanity
1	10	8	6	4	True	True	False
2	100	80	60	40	True	True	True
3	8	7	6	5	True	True	False
4	15	7	3	2	True	True	True
5	30	15	8	4	True	True	True

The liveness specifications for each pair are similarly extended to be:

$$\square\neg(mode_1 = Danger) \rightarrow \big(\square\lozenge(mode_1 = Keeping) \\ \vee \lozenge\square(mode_1 = Chasing)\big)$$

$$\square\neg(mode_2 = Danger) \rightarrow \big(\square\lozenge(mode_2 = Keeping) \\ \vee \lozenge\square(mode_2 = Chasing)\big)$$

$$\square\neg(mode_3 = Danger) \rightarrow \big(\square\lozenge(mode_3 = Keeping) \\ \vee \lozenge\square(mode_3 = Chasing)\big)$$

IV. ANALYSIS AND RESULTS

Through utilizing the interactive mode of *nuXmv*, the model was examined with respect to the aforementioned safety and liveness specifications for both the single car follower and the platoon.

A. Single Car Follower

Both the safety and liveness requirements returned true meaning that the system never enters the $Danger$ state and as long as the system is safe the vehicles are either repeatedly $Keeping$ or persistently $Chasing$.

In order to verify that the specifications were actually reachable and satisfied, as opposed to vacuously true, we also analyzed the following specifications:

$$\square\neg(mode = Braking),$$

$$\square\neg(mode = Chasing),$$

$$\text{and } \square\neg(mode = Keeping).$$

Each of the above specifications should return false if the *Braking*, *Chasing*, and *Keeping* modes are reachable.

The threshold values were adjusted to make the system work properly. The tested thresholds are shown in Table II. The values that were shown to avoid collisions in the model implementation were $Thresh_0 = 10$, $Thresh_1 = 8$, $Thresh_2 = 6$, and $Thresh_3 = 4$. *nuXmv* demonstrated that the state space explosion problem applies to this model. For this version of the model, there are 928699 reachable states out of the total 4.16201×10^6 possible states.

As shown in Table II, both the liveness and safety specifications returned true for the previously mentioned thresholds. The results indicate that the system never reaches the *Danger* mode and that eventually the system will either be repeatedly *Keeping* or persistently *Chasing*. Another tuple of thresholds that also satisfies the specifications is shown as Trial 3 in Table II.

B. Platoon

The model is applied to each consecutive pair of vehicles in the platoon. As shown in Figure 1, each vehicle follows the vehicle preceding it, except for the front-most vehicle. When implementing a three vehicle platoon, the state space explosion problem is exemplified as the possible states increases from 4.16201×10^6 for two vehicles to 1.71508×10^{11} for three vehicles. When trying to implement a four vehicle platoon, the state space explosion problem is exemplified through the number of possible states increasing from 4.16201×10^6 for two vehicles to 7.0675×10^{15} for four vehicles.

We implemented a three vehicle platoon utilizing the same thresholds as mentioned in Section IV-A. The safety, liveness, and verification specifications were analyzed. Verifying that the safety and liveness specifications are not vacuously true was implemented through the following specifications:

$$\square\neg(mode_1 = Braking)$$
$$\square\neg(mode_2 = Braking)$$
$$\square\neg(mode_1 = Chasing)$$
$$\square\neg(mode_2 = Chasing)$$
$$\square\neg(mode_1 = Keeping)$$
$$\square\neg(mode_2 = Keeping)$$

The three vehicle platoon implementation took about ten hours to terminate. The system returned that the safety specification for $mode_1$ was satisfied, while the safety specification for $mode_2$ was not satisfied. Similarly, the liveness specification for $mode_1$ was satisfied, while the liveness specification for $mode_2$ was not satisfied. The verification specification $\square\neg(mode_1 = Braking)$ returns true, which indicates that the mode *Braking* is not reachable for vehicle 1. The same verification specification returns false for $mode_2$, meaning the mode is reachable for vehicle 2. The verification specification $\square\neg(mode_1 = Chasing)$ returns false, which indicates that the mode *Chasing* is reachable for vehicle 1. The same verification specification for $mode_2$ returns false, meaning the mode is also reachable for vehicle 2. The verification specification $\square\neg(mode_1 = Keeping)$ returns false, which indicates that the mode *Keeping* is reachable for vehicle 1. The same verification specification for $mode_2$ returns false, meaning the mode is also reachable for vehicle 2.

Using the same thresholds as used in Section IV-A for the single car follower model, we implemented an attempt for four vehicle platooning. The implementation also checked the safety and liveness specification as well as the verification specifications. However, the implementation did not terminate within a day of running, but did return that the safety specifications for $mode_1$ and $mode_3$ were satisfied. The safety specification for $mode_2$, however, was not satisfied, meaning that vehicle 2 can crash into vehicle 1. The liveness specifications for both $mode_1$ and $mode_2$ were satisfied, whereas the liveness specification for $mode_3$ was not satisfied.

Verifying that the safety and liveness specifications are not vacuously true was implemented through the following specifications:

$$\square\neg(mode_1 = Braking)$$
$$\square\neg(mode_2 = Braking)$$
$$\square\neg(mode_3 = Braking)$$
$$\square\neg(mode_1 = Chasing)$$
$$\square\neg(mode_2 = Chasing)$$
$$\square\neg(mode_3 = Chasing)$$
$$\square\neg(mode_1 = Keeping)$$
$$\square\neg(mode_2 = Keeping)$$
$$\square\neg(mode_3 = Keeping)$$

The verification specification $\square\neg(mode_1 = Braking)$ returns true, which indicates that the mode *Braking* is not reachable for vehicle 1. The same verification specification returns false for $mode_2$, meaning the mode is reachable for vehicle 2. Unfortunately, the implementation did not terminate to determine the validity of the remaining verification specifications.

V. Simulated Platoon Visualization

To demonstrate our model is applicable to platooning applications and help us visualize how the model works, we wrote a simulator in MATLAB. The simulator allows us to test different scenarios for the ACC model and see what transitions the model makes and how it affects other following cars. These simulations cannot prove or disprove any behavior. However, it is helpful to see what is going on to help us further understand how things work.

In the simulator, the lead vehicle follows an independent velocity profile while the remaining cars follow the dynamics proposed in the model from Section 2. The value for a_f in the $Braking$ mode is set to $-3m/s^2$, which was the value specified in the model because it is a quick deceleration, but not so fast that people choke on their seat belts. The thresholds are $thresh_0 = 10m$, $thresh_1 = 8m$, $thresh_2 = 6m$, and $thresh_3 = 4m$, which are the smallest values we found to work while testing our model. At initialization, cars are spaced $11m$ apart with velocities equal to the lead car's initial velocity. This ensures each car starts in the $Chasing$ mode. The simulation works in 1 second intervals, to match the assumptions made for the *nuXmv* implementation.

A. Constant Lead Car Velocity

In the first simulation, the lead vehicle follows a constant velocity profile where $v_l = 8m/s \approx 18mph$. The maximum velocity any car can reach is set to $11m/s \approx 25mph$. The results of the simulation are shown in Figure 5.

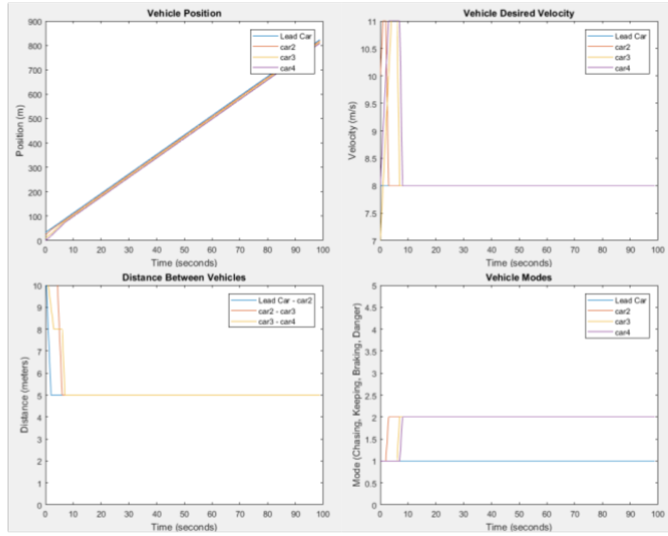

Fig. 5. Simulation results from following a constant velocity profile.

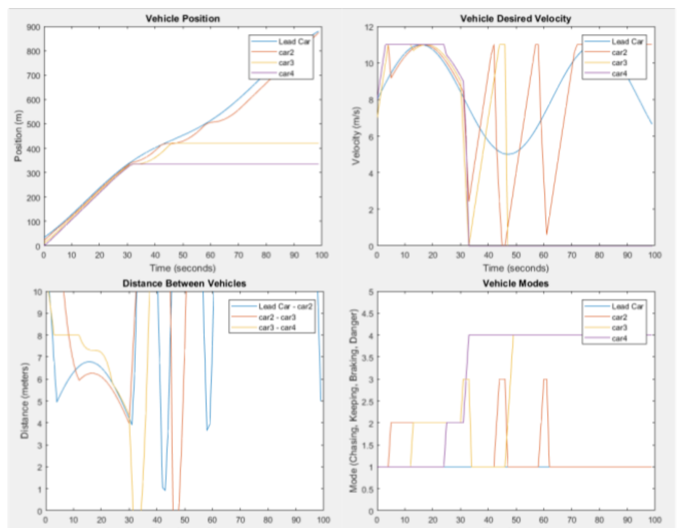

Fig. 6. Simulation results from a platoon following a sinusoidal velocity profile.

From the simulation results, we can see that no crashes occur and all of the cars enter the $Reaching$ mode and remain there until the end of the simulation. This indicates the system can reach a steady, unchanging equilibrium, if the system starts in a safe operating mode and the lead vehicle does not change its velocity.

B. Sinusoidal Lead Car Velocity

In the second simulation, the lead vehicle follows a sinusoidal velocity profile where $v_l = 3 * sin(time/10) + 8m/s$ meaning the velocity oscillates from about $11mph$ to $25mph$. The results of the simulation are shown in Figure 6.

The simulation resulted in two crashes. Both crashes involved the last car in the platoon and happened for the same reason. To understand why these crashes happened, we have

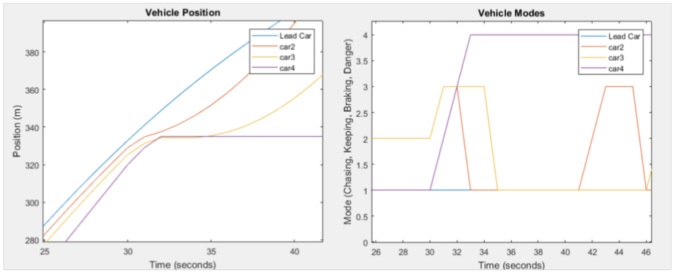

Fig. 7. Simulation results from a platoon following a sinusoidal velocity profile.

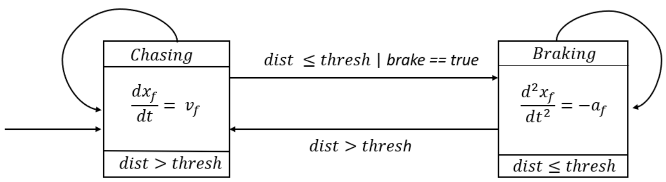

Fig. 8. Diagram illustrating a simpler model.

zoomed in on the positions and modes of all of the cars around the time of the first crash in Figure 7. The crash occurs when $car4$ is in the *Chasing* mode but then $car3$ suddenly changes to *Braking* and the gap closes too fast for $car4$ to switch from *Chasing* to *Keeping* to *Braking* with enough time to reduce speed and avoid the collision. In future work we will focus on finding a way to prevent these kinds of scenarios from occurring.

VI. Future Work

The main direction for the future of this work is to improve the model through the incorporation of physical-world characteristics. Reducing the drastic changes in velocity outside of the $Braking$ state is an area of additional exploration. Another important future direction is implementing the model in SpaceEx or other verification tools such as Flow* and HyComp. The model should be extended to handle larger platoon formations without collisions. Our implementation of both the three vehicle platoon and the four vehicle platoon have their imperfection. For the three vehicle platoon, it took about ten hours to terminate, some modes that should have been reachable were not, and the *Danger* mode was reachable when it should not have been, indicating collisions. For the four vehicle platoon, it ran for about two days never terminated, the *Danger* mode was reachable when it should not have been, indicating collisions, and the liveness specification for $mode_3$ was not satisfied. There are many improvements that could be implemented in the model to fix the aforementioned errors.

Another possibility is to try solving the problem through the use of only two modes, shown in Figure 8, for each pair of vehicles. The following vehicle's velocity is directly related to the leader vehicle's velocity, as shown in Figure 3. In order to deal with the reachable state space explosion problem, a model that is both environmentally and economically inefficient was hypothesized. In the model with two modes, the following vehicle has an initial speed and adjusts its velocity with respect to the distance between it and the leading vehicle. If the distance between them is below a threshold *thresh*, the following vehicle enters the Braking mode. When the distance is greater than the threshold, the following vehicle enters the Chasing mode. The constant acceleration and deceleration of the vehicle is indicative of the inefficiencies mentioned above.

VII. Conclusions

The results indicate that for the simplified adaptive cruise control model with only two vehicles the safety and liveness specifications are satisfied. When extending the model to a platoon with both three and four vehicles, the results show that there is a possibility of collisions between vehicles. Through this work, we learned a lot about State Space Explorer (SpaceEx), adaptive cruise control (ACC), and car following. Some challenges include initially attempting to implement the model in SpaceEx and modelling a platoon without collisions. The state space explosion problem appeared prominently our implementation of the adaptive cruise control model.

References

[1] G. Saldivia, "Stuck in traffic? youre not alone. new data show american commute times are longer," Sep 2018.

[2] F. Siddiqui, "Americans' commutes keep getting longer, according to survey data," Sep 2018.

[3] Y. Sugiyama, M. Fukui, M. Kikuchi, K. Hasebe, A. Nakayama, K. Nishinari, S. ichi 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, p. 033001, mar 2008.

[4] S. ichi Tadaki, M. Kikuchi, M. Fukui, A. Nakayama, K. Nishinari, A. Shibata, Y. Sugiyama, T. Yosida, and S. Yukawa, "Phase transition in traffic jam experiment on a circuit," *New Journal of Physics*, vol. 15, p. 103034, oct 2013.

[5] R. Jiang, C.-J. Jin, H. Zhang, Y.-X. Huang, J.-F. Tian, W. Wang, M.-B. Hu, H. Wang, and B. Jia, "Experimental and empirical investigations of traffic flow instability," *Transportation Research Procedia*, vol. 23, pp. 157 – 173, 2017. Papers Selected for the 22nd International Symposium on Transportation and Traffic Theory Chicago, Illinois, USA, 24-26 July, 2017.

[6] R. Jiang, M.-B. Hu, H. Zhang, Z.-Y. Gao, B. Jia, Q.-S. Wu, B. Wang, and M. Yang, "Traffic experiment reveals the nature of car-following," *PLoS ONE*, 2014.

[7] R. E. Stern, S. Cui, M. L. D. 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, "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.

[8] J. A. Laval and C. F. Daganzo, "Lane-changing in traffic streams," *Transportation Research Part B: Methodological*, vol. 40, no. 3, pp. 251 – 264, 2006.

[9] J. A. Laval, "Stochastic processes of moving bottlenecks: Approximate formulas for highway capacity," *Transportation Research Record*, vol. 1988, no. 1, pp. 86–91, 2006.

[10] Z. Zheng, S. Ahn, D. Chen, and J. Laval, "Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations," *Transportation Research Part B: Methodological*, vol. 45, no. 2, pp. 372 – 384, 2011.

[11] "Vehicle platooning for safety and efficiency act," Jan 2018.

[12] C. Domonoske, "A safety feature by any other name can really be confusing, AAA finds," *NPR*, Jan 2019.

[13] G. Newell, "A simplified car-following theory: a lower order model," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195 – 205, 2002.

[14] A. Loria, J. Dasdemir, and N. Alvarez Jarquin, "Leader-follower formation and tracking control of mobile robots along straight paths," *IEEE Transactions on Control Systems Technology*, vol. 24, pp. 727–732, March 2016.

[15] J. Chunyu, Z. Qu, E. Pollak, and M. Falash, “A new multi-objective control design for autonomous vehicles,” in *Optimization and Cooperative Control Strategies* (M. J. Hirsch, C. W. Commander, P. M. Pardalos, and R. Murphey, eds.), (Berlin, Heidelberg), pp. 81–102, Springer Berlin Heidelberg, 2009.
[16] A. Soni and H. Hu, “Formation control for a fleet of autonomous ground vehicles: A survey,” *Robotics*, vol. 7, p. 67, 2018.
[17] J. Hu, J. Lygeros, and S. Sastry, “Towards a theory of stochastic hybrid systems,” in *Hybrid Systems: Computation and Control* (N. Lynch and B. H. Krogh, eds.), (Berlin, Heidelberg), pp. 160–173, Springer Berlin Heidelberg, 2000.
[18] J.-F. Raskin, “An introduction to hybrid automata,” in *Handbook of networked and embedded control systems*, pp. 491–517, Springer, 2005.
[19] “The nuxmv model checker.”