Strings of Vehicles: Modeling & Safety Conditions[*]

John Lygeros[†] and Nancy Lynch[‡]

[†]Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, CA 94720-1770
lygeros@eecs.berkeley.edu

[‡]Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139
lynch@lcs.mit.edu

Abstract. Motivated by our work on Automated Highway Systems (AHS), we consider a physical system, the *string of vehicles* and construct a natural model for it in the Hybrid Input/Output Automaton formalism. We describe a special maneuver that may have to be executed by the system, the *emergency deceleration maneuver*, and derive necessary and sufficient conditions on the system parameters under which this maneuver can be executed in safety. We conclude by giving a brief discussion of the implications of our results for the design of an AHS that allows the formation of platoons of vehicles.

1 Introduction

Hybrid systems have attracted the attention of both computer theorists and control engineers. Our work ultimately aims at a rapprochement of these two perspectives. Here we use a combination of techniques from the two areas to address a specific problem in transportation. This is the problem of the safety of a collection of vehicles traveling one behind the other in a single lane; we refer to such a collection as a *string of vehicles*. The problem is hybrid as it involves both continuous vehicle motion and (possibly) collisions, which in our setting are treated as discrete velocity changes. We try to establish conditions under which a string of vehicles will be safe while executing a particular maneuver.

We start by developing a detailed model for the system in the *Hybrid Input/Output Automaton* modeling framework (Section 2). Modest extensions of the original framework of [1] are needed to capture all the phenomena of interest for this problem. Then, in Section 3 we introduce the emergency deceleration

[*] Research supported by California PATH under MOU-238 and MOU 288, ARPA contract F19628-95-C-0118, AFOSR contract F49620-97-1-0337, AFOSR contract F49620-94-1-0199 and the U.S. Department of Transportation DTRS95G-0001.

maneuver, whose safety analysis is the primary focus of this paper. We give some necessary and some sufficient conditions under which the safety of the maneuver can be guaranteed. Finally, in Section 4, we discuss the implications of our results in the context of platooning of vehicles.

We believe our work is potentially of both theoretical and practical importance. On the theoretical side we hope that the results presented here will be extended to a general methodology for dealing with hybrid systems, one where continuous and discrete techniques are combined in a coherent framework. The practical implications of our work are more immediate. Our results indicate that the design of specialized emergency maneuvers may be crucial to the success of an automated highway system that allows for the formation of platoons.

2 Vehicle String Model

2.1 Overview of the Modeling Formalism

Based on the work of [1], we consider a hybrid automaton, A, as a dynamical system that describes the evolution of a finite collection of variables, V_A. Variables are typed; for each $v \in V_A$ let *type*(v) denote the type of v. For each $Z \subseteq V_A$, a *valuation* of Z is a function that to each $v \in Z$ assigns a value in *type*(v). Let $\mathbf{Z}$ denote the set of valuations of Z; we refer to $s \in \mathbf{V_A}$ as a system *state.* In this paper we assume that the evolution of the variables is over the set $T^{\geq 0} = \{t \in \mathbb{R} | t \geq 0\}$. The evolution of the variables involves both continuous and discrete dynamics. Continuous dynamics are encoded in terms of *trajectories* over V_A, that is functions that map intervals of $T^{\geq 0}$ to $\mathbf{V_A}$. Discrete dynamics are encoded by *actions.* Upon the occurrence of an action the system state instantaneously "jumps" to a new value. We use Σ_A to denote the set of actions that affect the evolution of A.

More formally, a *hybrid automaton*, A is a collection (U_A, X_A, Y_A, Σ_A^{in}, Σ_A^{int}, Σ_A^{out}, Θ_A, $\mathcal{D}_A$, $\mathcal{W}_A$) consisting of:

- Three disjoint sets U_A, X_A, and Y_A of variables, called *input*, *internal*, and *output variables*, respectively. We set $V_A = U_A \cup X_A \cup Y_A$.
- Three disjoint sets Σ_A^{in}, Σ_A^{int}, and Σ_A^{out} of actions, called *input*, *internal*, and *output actions*, respectively. We set $\Sigma_A = \Sigma_A^{in} \cup \Sigma_A^{int} \cup \Sigma_A^{out}$.
- A non-empty set $\Theta_A \subseteq \mathbf{V_A}$ of *initial states.*
- A set $\mathcal{D}_A \subseteq \mathbf{V}_A \times \Sigma_A \times \mathbf{V}_A$ of *discrete transitions.*
- A set $\mathcal{W}_A$ of *trajectories* over V_A.

Some technical axioms are imposed on the above sets to guarantee that the definitions are consistent. The axioms introduced in [1] are too restrictive for the application considered here; fortunately the extensions needed are fairly straightforward.

An *execution*, α, of A is an alternating sequence $\alpha = w_0 a_1 w_1 a_2 w_2 \cdots$, finite or infinite, where for all i, $a_i \in \Sigma_A$, $w_i \in \mathcal{W}_A$ defined over a left closed time interval and *fstate*$(w_0) \in \Theta_A$. In addition, if α is a finite sequence then it ends with a trajectory and if w_i is not the last trajectory its domain is right-closed and $(\mathit{lstate}(w_i), a_{i+1}, \mathit{fstate}(w_{i+1})) \in \mathcal{D}_A$. Here $\mathit{fstate}(w)$ and $\mathit{lstate}(w)$ denote the initial and final states of a trajectory w. An execution is called *finite* if it is a finite sequence and the domain of its final trajectory is a right-closed interval. A state $s \in \mathbf{V_A}$ is called *reachable* if it is the last state of a finite execution.

Hybrid automata "communicate" through shared variables and shared actions. Consider two automata A and B with $X_A \cap V_B = X_B \cap V_A = Y_B \cap Y_A = \emptyset$ and $\Sigma_B^{int} \cap \Sigma_A = \Sigma_A^{int} \cap \Sigma_B = \Sigma_A^{out} \cap \Sigma_B^{out} = \emptyset$. Under some mild technical assumptions, the *composition*, $A \times B$, of A and B can be defined as a new hybrid automaton with $U_{A\times B} = (U_A \cup U_B) \setminus (Y_A \cup Y_B)$, $X_{A\times B} = X_A \cup X_B$, $Y_{A\times B} = Y_A \cup Y_B$ and similarly for the actions. $\Theta_{A\times B}$, $\mathcal{D}_{A\times B}$ and $\mathcal{W}_{A\times B}$ are such that the executions of $A \times B$ are also executions of each automaton when restricted to the corresponding variables and actions.

A *derived variable* of A is a function on $\mathbf{V_A}$. Derived variables will be used to simplify the system description, but also to facilitate the analysis. A *property* of A is a boolean derived variable. A property is *stable* if whenever it is true at some state it is also true at all states reachable from that state. A property is *invariant* if it is true at all reachable states. Typically properties will be shown to be stable or invariant by an induction argument on the length of an execution. It is easy to show that:

Lemma 1 *Assume that for all reachable states s of A, P true at s implies P true at s' for all s' such that either there exists $w \in \mathcal{W}_A$ with right closed domain and* $\mathrm{fstate}(w) = s$ *and* $\mathrm{lstate}(w) = s'$, *or, there exists $a \in \Sigma_A$ with $(s, a, s') \in \mathcal{D}_A$. Then P is a stable property. If further P is true at all $s \in \Theta_A$, then P is an invariant property.*

In some places differential equations will be used to simplify the description of the set $\mathcal{W}_A$. In such cases $\mathcal{W}_A$ is assumed to be populated by all trajectories generated by the differential equation in the usual way. To simplify the description of $\mathcal{D}_A$, we will assign a *precondition* and an *effect* to each action. The precondition is a predicate on $\mathbf{V_A}$ while the effect is a predicate on $\mathbf{V_A} \times \mathbf{V_A}$. The action can take place only from states that satisfy the precondition; moreover, the states before and after the transition should be such that the effect is satisfied. When no confusion can arise we use v' to denote the value of variable v after an action.

2.2 String Model

Consider a string of N vehicles (Figure 1) moving one behind the other in a single lane, with vehicle 0 coming first. The overall model will be the composition of a number of automata (Figure 2). The *plant* will be a hybrid automaton containing the dynamics of all the vehicles in the string. Each vehicle is equipped with *sensors* and *controllers*. The sensor automaton S_i reads the values of the plant output variables as inputs and produces real valued output variables. The controller automaton, C_i, reads the corresponding sensor output variables and

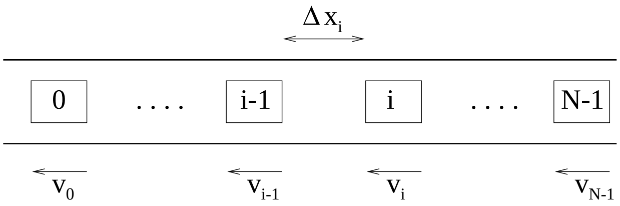

Fig. 1. A string of vehicles

uses them to generate the input variables of the plant. The S_i and C_i may have internal variables and actions. In this paper we assume that the sensor and controller automata are simple input/output maps and concentrate on the development of a realistic plant model.

The plant is modeled by an automaton $P = (U_P, X_P, Y_P, \Sigma_P^{in}, \Sigma_P^{int}, \Sigma_P^{out}, \Theta_P, \mathcal{D}_P, \mathcal{W}_P)$. P has no input and no output actions, hence $\Sigma_P^{in} = \Sigma_P^{out} = \emptyset$. Here we are only interested in answering questions of "safety", encoded in terms of possible collisions among the vehicles of the string. The answers to these questions will depend on the relative spacing and the velocities of the vehicles, but not their absolute position on the road. Let Δx_i denote the spacing between vehicle i and $i-1$, v_i the speed of vehicle i, acc_i its acceleration and u_i its commanded acceleration[2] and define $x_i = [\Delta x_i \;\; v_i] \in \mathbb{R}^2$, $x = [x_0 \; \ldots \; x_{N-1}] \in \mathbb{R}^{2N}$ and $u = [u_0 \; \ldots \; u_{N-1}] \in \mathbb{R}^N$. Also consider a collection of boolean variables $Touching = \{Touching_1, \ldots Touching_{N-1}\}$; the evolution of these variables (Section 2.2) will be such that $Touching_i$ is true whenever vehicle i is touching vehicle $i-1$. Define the internal and input variables as $X_P = \{x, acc, Touching\}$ and $U_P = \{u\}$ respectively. Physical limitations constrain the valuations of the input variables to lie in a rectangular compact set, i.e. $u_i(t) \in [a_i^{min}, a_i^{max}]$ for all i and for all t. The values of a_i^{min} and a_i^{max} are determined by the vehicle characteristics (engine, brakes, tires, etc.). To ensure that the model is realistic we impose the following assumption on Θ_P and the input constraints.

Assumption 1 *For all i, $\Delta x_i(0) \geq 0$, $v_i(0) \geq 0$, $Touching_i(0) = \text{False}$ and $a_i^{min} < 0 < a_i^{max}$.*

Discrete Dynamics The continuous system evolution can be interrupted by three classes of internal actions: collisions, vehicles touching with zero relative velocity (and subsequently "pushing" against one another) and vehicles moving apart (after having touched). We assume that the continuous evolution stops as soon as the precondition of an action becomes true, to allow the action to take place. All variables not explicitly mentioned in the effect are assumed to be unaffected by the action.

[2] As discussed in Section 2.2, the commanded and actual acceleration may differ when vehicles are touching and pushing each other.

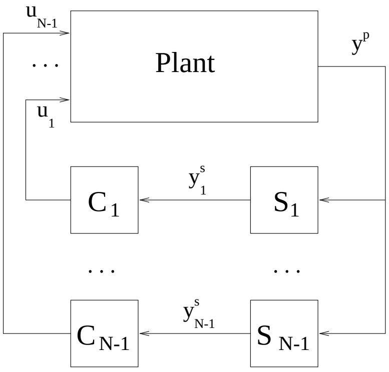

Fig. 2. System modules

Consider first the case of collisions. Let $\mathit{Collision}_i$ be an internal action that takes place whenever vehicle i collides with vehicle $i-1$. The precondition for $\mathit{Collision}_i$ is:

$$(\Delta x_i = 0) \wedge (v_i > v_{i-1}) \tag{1}$$

To determine the effect of the action we use a simple collision model. To determine v_i and v_{i-1} after the collision we solve a pair of equations:

$$M_i v_i' + M_{i-1} v_{i-1}' = M_i v_i + M_{i-1} v_{i-1} \tag{2}$$

$$v_{i-1}' - v_i' = (v_i - v_{i-1})\alpha_i \tag{3}$$

where M_i is the mass of vehicle i and α_i is the coefficient of restitution, a measure of the energy lost in the collision. Equation (2) is the *conservation of momentum equation* while Equation (3) is referred to as the *restitution equation.* By appropriate choice of α (possibly as a function of the speeds) this collision model can capture a wide range of collision scenarios. To maintain a certain level of generality in the subsequent discussion we will typically assume that the coefficient of restitution is a function of the relative velocity $v_{i-1} - v_i$ at impact and will denote it by $\alpha_i(\cdot)$. To ensure that the model is realistic we impose the following assumption:

Assumption 2 *For all i, $M_i > 0$ and $\alpha_i(v) \in [0,1]$ for all $v > 0$.*

Note that in general vehicles may end up going backwards as a result of collisions if, for example, a light vehicle elastically hits a slowly moving heavy vehicle (i.e. $M_i \ll M_{i-1}$, $\alpha_i \approx 1$ and $v_{i-1} \approx 0$).

Multiple instantaneous collisions are also possible in this setting. These are situations where there exist N_1 and N_2 with $0 \leq N_1 < N_2 < N$ such that $\Delta x_{N_1} \neq 0$, $\Delta x_{N_2+1} \neq 0$ (if any) and for all i with $N_1 < i \leq N_2$, $\Delta x_i = 0$ and $v_i > v_{i-1}$. The value, x', of the state after the collision again satisfies $\Delta x_i' = \Delta x_i$ for all i and $v_i' = v_i$ for all $i < N_1$ or $i > N_2$. To determine the values of v_i for $N_1 \leq i \leq N_2$ we resolve the multiple collision as a sequence of pairwise collisions, according to equations (2) and (3). The pairwise resolutions will keep taking place as long as there exists a j with $N_1 < j \leq N_2$ such that $v_j > v_{j-1}$. When this condition is violated we will say that the multiple collision has been resolved. It turns out that, if the masses of the vehicles are unequal or the restitution coefficients α_i are not identically equal to 1, one can construct scenarios where the velocities of the vehicles after the multiple collision has been resolved depend on the order in which the pairwise resolutions were executed. To circumvent this problem we state our theorems and proofs in a way that the results hold for *all possible* orderings of the pairwise resolutions.

Next, let Touch_i be an internal action that takes place whenever vehicle i touches vehicle $i-1$ with zero relative velocity. The precondition for Touch_i is:

$$(\mathit{Touching}_i = \text{False}) \wedge (\Delta x_i = 0) \wedge (v_i = v_{i-1}) \wedge (\mathit{acc}_i \geq \mathit{acc}_{i-1}) \tag{4}$$

The effect of Touch_i is to declare the two vehicles as touching, i.e. $\mathit{Touching}_i' =$ True.

Finally, consider what happens when vehicles that are touching start moving away from one another. Let $\mathit{Separate}_i$ be an internal action that takes place whenever vehicle i is already touching vehicle $i-1$ and starts to move away. The precondition for $\mathit{Separate}_i$ is:

$$(\mathit{Touching}_i = \text{True}) \wedge [(\mathit{acc}_i < \mathit{acc}_{i-1}) \vee (v_i < v_{i-1})] \tag{5}$$

The effect of $\mathit{Separate}_i$ is to declare the two vehicles as no longer touching, i.e. $\mathit{Touching}_i' =$ False.

Continuous Dynamics The set of trajectories $\mathcal{W}_P$ will be generated by a dynamical system. Assume there are no vehicles ahead of the string and set $\Delta x_0 \equiv \infty$. Then, for $i = 1, \ldots, N-1$ the laws of motion imply that:

$$\begin{aligned} \dot{\Delta x}_i(t) &= v_{i-1}(t) - v_i(t) \\ \dot{v}_i(t) &= \mathit{acc}_i(t) \end{aligned}$$

The value of the actual acceleration, acc_i, of vehicle i depends on the acceleration commanded by the controller of that vehicle, u_i, and on whether the vehicle is touching vehicle $i-1$ or vehicle $i+1$. In the case when the vehicles are not touching we simply set the actual acceleration equal to the commanded acceleration. The case where vehicles are touching is more complicated. The reason is that when vehicles are pushing against one another, there are forces exerted from one vehicle to the other. Therefore, the actual acceleration of a vehicle depends not only on the acceleration commanded by its own controller, but also on the accelerations commanded by the controllers of the neighboring vehicles that are pushing against it.

To resolve this issue we first introduce some abstract definitions. Consider a nonempty finite subset of the natural numbers $S \subset \mathbb{N}$. S is a *segment* if it consists of consecutive numbers. A *subsegment* of a segment S is any subset of S that is also a segment. For segments S_1 and S_2 with $\min(S_2) = \max(S_1) + 1$

we define their *concatenation* (denoted by S_1S_2) as the segment $S_1 \cup S_2$. A *weighted average function on* S is any function $a : 2^S \to \mathbb{R}$ such that for all L, R subsegments of S:

$$\min\{a(L), a(R)\} \leq a(LR) \leq \max\{a(L), a(R)\} \tag{6}$$

whenever the concatenation LR is defined. A segment S with a weighted average function a is *unsplitable* if:

$$S = LR \Rightarrow a(L) \leq a(R)$$

A *partition of* S is a finite collection $S_1, \ldots, S_n$ where $S = \cup_{k=1}^{n} S_k$ and for all k, S_k is a segment and $S_k \cap S_l = \emptyset$ for $l \neq k$. Without loss of generality assume that $\min(S) = \min(S_1)$ and for all $1 < k \leq n$, $\min(S_k) = \max(S_{k-1}) + 1$ and write $S = S_1S_2 \ldots S_n$. A partition of $S_1 \ldots S_n$ of S is called a *maximal partition* if for all $k = 1, \ldots, n$, S_k is unsplitable and either $n = 1$ or for all $k = 2, \ldots, n$, $a(S_{k-1}) > a(S_k)$.

Theorem 1 *For every segment, S, and every weighted average function, a, on S there exists a unique maximal partition.*

Though interesting, the proof of Theorem 1 is omitted here as it is not necessary for the safety results. An algorithm to construct the maximal partition has also been developed.

Intuitively (returning to the vehicle example) a maximal partition is such that vehicles in an element of the partition are pushing against one another while vehicles in different elements of the partition are moving away from one another. Assume there exist i, j with $0 < i < j < N$ such that vehicles i to j are touching each other. Define the segment $S = \{i, \ldots, j\}$ and for every subset $S' \subseteq S$ consider the function:

$$a(S') = \frac{\sum_{k \in S'} M_k u_k}{\sum_{k \in S'} M_k} \tag{7}$$

One can show that a is a weighted average function on S. To determine the acceleration of the vehicles in this collection at a given instant, let $S_1 \ldots S_n$ be the maximal partition of S at that instant and for all $k = 1, \ldots, n$ set:

$$acc_l = a(S_k) \text{ for all } l \in S_k \tag{8}$$

If one assumes that the force exerted on a vehicle by the road depends only on the commanded acceleration of that vehicle (and not on whether the vehicle is touching other vehicles), then this choice is what one would expect from physical intuition. The total force commanded by all the vehicles determines the acceleration of their combined mass.

2.3 Output Evolution

The output evolution is determined as a function of the evolution of the inputs and states. We assume that in principle all the internal variables can be made available to the controllers. Limitations imposed by current sensing and communication technology should be incorporated in the sensor automata. Therefore the information made available by vehicle i is $y_i^p(t) = [x_i(t) \;\; acc_i(t)]$. Let $y^p = \left[y_0^p \dots y_{N-1}^p\right] \in \mathbb{R}^{3N}$ and define the output variables as $Y_P = \{y^p\}$.

2.4 Model Consistency & Safety Requirements

The following lemma suggests the proposed plant model agrees with basic physical intuition:

Lemma 2 *The plant automaton is such that:*

1. *If E and E' are the kinetic energy before and after Collision$_i$, then $E' \leq E$.*
2. $\wedge_{i=0}^{N-1} [\Delta x_i \geq 0]$ *is an invariant property of the plant.*
3. $\wedge_{i=1}^{N-1} [(\text{Touching}_i = \textit{True}) \Rightarrow (\Delta x_i = 0)]$ *is an invariant property of the plant.*

The kinetic energy of the string is defined as:

$$E = \sum_{i=0}^{N-1} \frac{1}{2} M_i v_i$$

The first property shows that (as expected) no energy is generated as a result of the collisions. The second property shows that the model does not allow vehicles to run over one another (a physical impossibility). The last property shows that vehicles are declared as touching by the model only when they are physically touching.

We are interested in defining the system performance in terms of the severity of the collisions experienced by the vehicles. Following [2], we assume that a collision is safe if the relative velocity at impact is below a certain threshold, v_A. A commonly cited threshold is $v_A = 3ms^{-1}$ [2].

Definition 1 *A string is* **safe** *if* $\bigwedge_{i=1}^{N-1} [(\Delta x_i = 0) \Rightarrow (v_i \leq v_{i-1} + v_A)]$ *is an invariant property.*

The main limitation of our model is that is does not account for the lateral motion of the vehicles. We assume that all vehicles effectively move along a straight line. This assumption may be unrealistic, especially in the presence of collisions when large forces and moments can be exerted from one vehicle to another. The situation will be even worse when the vehicles move along a curved road.

3 Safety Conditions for Emergency Deceleration

3.1 Background

The *emergency deceleration maneuver* is a situation where the first vehicle in the string applies maximum deceleration until it comes to a stop, thus endangering the remaining vehicles of the string. It is assumed that the emergency deceleration of vehicle 0 is caused by some abnormal condition, such as a mechanical malfunction or an obstacle. We would like to determine the conditions under which the remaining vehicles can maintain their safety despite this "malicious" behavior of the leader.

The safety of general strings of vehicles has been analyzed using a number of techniques. Most results in the literature start by partly characterizing the string by determining "automata" for the sensors and controllers and then trying to establish the range of initial conditions and parameters for which the string is safe. This type of analysis has led to conditions under which pairs of vehicles are guaranteed not to collide [3, 4] or experience safe collisions [4, 5]. In some cases the conditions have also been extended to longer or even infinite strings [6, 7].

Perhaps the most challenging problem in this area has been the design of controllers for platoons of vehicles. A *platoon* is a string of very tightly spaced vehicles. Typically intra-platoon spacings are of the order of 1-2 meters. The safety of the intra-platoon controllers [6] relies on the assumption that the behavior of the first vehicle is in some sense "reasonable". This means that the controller C_0 takes into account the limitations of the rest of the vehicles in the string when calculating u_0. This requirement is clearly violated in the case of the emergency deceleration maneuver. It is conjectured however that the platoon is going to be safe even in this case [8]. The justification is that collisions are going to take place in rapid succession, because the vehicles are all close to one another. Therefore, if the speeds of all vehicles are initially the same, the relative velocity at the time of collision is going to be small. We attempt to establish conditions under which this conjecture is true.

The safety of the string under an emergency deceleration maneuver depends on the response of the remaining vehicles of the string to the deceleration of the leader. Here we consider a very simple *default deceleration strategy.* Assume that at time $t = 0$ the leading vehicle applies maximum deceleration, a_0^{min}, until it stops at which point its commanded acceleration becomes 0. After a delay d_i vehicle i also applies a_i^{min} until it comes to a stop. This scenario can be easily encoded in the model discussed above by simple sensor and controller automata. The results discussed here refer to the case $d_i = 0$; some of them directly extend to the more general case.

3.2 Safety Conditions For Strings of Length N=2

We first develop conditions for a string of two vehicles to be safe under the default deceleration strategy. These conditions will form the basis of safety results for longer strings. We refer to a two vehicle string as a *pair.* One can easily show that:

Proposition 1 *$(v_0 \geq 0)$ and $(v_1 \leq 0)$ are stable properties for a pair. If $(v_1 \leq 0)$ the pair is safe (in particular $Collision_1$ cannot occur).*

To derive more meaningful safety properties consider the derived variables:

$$C_1(\Delta x_1, v_1, v_0) = (a_1^{min} + a_0^{min})v_0^2 - 2a_0^{min}v_0v_1 - 2(a_0^{min})^2\Delta x_1 \tag{9}$$

$$C_2(\Delta x_1, v_0, v_1) = \frac{a_1^{min}}{a_0^{min}}v_0 - v_1 \tag{10}$$

$$P_1(\Delta x_1, v_0, v_1) = (v_0 - v_1)^2 - 2(a_0^{min} - a_1^{min})\Delta x_1 - v_A^2 \tag{11}$$

$$P_2(\Delta x_1, v_0, v_1) = v_1^2 - \frac{a_1^{min}}{a_0^{min}}v_0^2 + 2a_1^{min}\Delta x_1 - v_A^2 \tag{12}$$

To simplify the notation we will explicitly mention the function arguments only when necessary. We also introduce a derived boolean variable C given by the expression:

$$C = \big[(C_1 \leq 0) \wedge (a_0^{min} \leq a_1^{min})\big] \vee \big[(C_2 \leq 0) \wedge (a_0^{min} \geq a_1^{min})\big] \vee \big[(v_0 = 0)\big] \tag{13}$$

P_1, P_2 and C are used to construct safety invariants. A collision can take place either while both vehicles are moving or while vehicle 1 is moving and vehicle 0 has stopped (by Proposition 1 collisions cannot take place once vehicle 1 stops). The property $(P_1 \leq 0)$ will encode conditions that guarantee safety if a collision takes place while both vehicles are still moving. $(P_2 \leq 0)$ will encode conditions that guarantee that either no collision takes place or a safe collision takes place after vehicle 0 has stopped. The predicate C will be used to distinguish the two cases.

Lemma 3 *$(P_1 \leq 0) \vee (v_1 \leq 0)$ is a stable property of the pair.*

Proof. $(P_1 \leq 0) \vee (v_1 \leq 0)$ is preserved by $Touch_1$ and $Separate_1$, as both these actions leave $\Delta x_1, v_0$ and v_1 unaffected. Assume $(P_1 \leq 0) \vee (v_1 \leq 0)$ is true when $Collision_1$ occurs. By Proposition 1 $(v_1 \leq 0)$ can not be true in this case. Therefore $(P_1 \leq 0)$ is true, i.e. $P_1(\Delta x_1, v_0, v_1) = P_1(0, v_0, v_1) \leq 0$. Hence, by the restitution equation (3), $(v_0' - v_1')^2 = (v_0 - v_1)^2\alpha_1^2 \leq (v_0 - v_1)^2 \leq v_A^2$, as $\alpha_1 \in [0, 1]$ by Assumption 2. Therefore, $P_1(\Delta x_1', v_0', v_1') = P_1(0, v_0', v_1') \leq 0$ and $(P_1 \leq 0) \vee (v_1 \leq 0)$ is again true after $Collision_1$.

Assume at some state, s, $(P_1 \leq 0) \vee (v_1 \leq 0)$ is true and consider all trajectories that start at s. If $(v_1 \leq 0)$ is true at s it will also be true at the last state of the trajectory by Proposition 1. If $(P_1 \leq 0) \wedge (v_1 > 0)$ is true at s, consider the variation of P_1 along a trajectory:

$$\begin{aligned}\frac{d}{dt}P_1 &= 2(v_0 - v_1)(acc_0 - acc_1) - 2(a_0^{min} - a_1^{min})(v_0 - v_1)\\ &= \begin{cases} 0 & \text{if } (v_0 > 0) \wedge (v_1 > 0) \wedge \neg Touching_1 \\ 2a_0^{min}v_1 & \text{if } (v_0 = 0) \wedge (v_1 > 0) \wedge \neg Touching_1 \\ -2(a_0^{min} - a_1^{min})(v_0 - v_1) & \text{if } Touching_1 \end{cases}\end{aligned}$$

In the cases where *Touching*$_1$ = False, $\dot{P}_1 \leq 0$, therefore $(P_1 \leq 0)$ will be true at least until $(v_1 \leq 0)$ becomes true. If *Touching*$_1$ = True and $v_0 < v_1$ (resp. $v_0 > v_1$) action *Collision*$_1$ (resp. *Separate*$_1$) occurs and the trajectory stops. If *Touching*$_1$ = True and $v_0 = v_1$, then $\dot{P}_1 = 0$. Overall, $(P_1 \leq 0) \vee (v_1 \leq 0)$ will be true at the last state of the trajectory. ■

Lemma 4 *If* $(P_1 \leq 0) \vee (v_1 \leq 0)$ *is true then the pair is safe.*

Proof. If $(v_1 \leq 0)$ is true the pair is safe by Proposition 1. If $(P_1 \leq 0)$, at the time when $\Delta x_1 = 0$, $P_1(\Delta x_1, v_0, v_1) = P_1(0, v_0, v_1) \leq 0$, therefore $(v_0 - v_1)^2 \leq v_A^2$. Hence, $v_1 \leq v_0 + v_A$ and the pair is safe. ■

The conditions of Lemma 4 can be relaxed by introducing P_2. Consider:

$$I = [P_1 \leq 0] \vee [C \wedge (P_2 \leq 0)] \tag{14}$$

Lemma 5 $I \vee (v_1 \leq 0)$ *is a stable property of the pair.*

Proof. If $(P_1 \leq 0) \vee [C \wedge (P_2 \leq 0)] \vee (v_1 \leq 0)$ is true at the pre-state of *Touch*$_1$ or *Separate*$_1$ it will also be true at the post-state as both actions leave $\Delta x_1, v_0$ and v_1 unaffected. Assume $(P_1 \leq 0) \vee [C \wedge (P_2 \leq 0)] \vee (v_1 \leq 0)$ is true when *Collision*$_1$ occurs. If $(P_1 \leq 0) \vee (v_1 \leq 0)$ is true, it will also be true after *Collision*$_1$ by Lemma 3. Assume *Collision*$_1$ occurs while $C \wedge (P_2 \leq 0)$ is true. We distinguish the following cases:
Case 1: $(v_0 = 0) \wedge (P_2 \leq 0)$ is true. Then, at $\Delta x_1 = 0$, $v_1^2 - v_A^2 \leq 0$, therefore $v_1 = v_1 - v_0 \leq v_A$.
Case 2: $(C_1 \leq 0) \wedge (a_0^{min} \leq a_1^{min}) \wedge (P_2 \leq 0)$ is true. Then, $0 < \frac{a_0^{min} + a_1^{min}}{2a_0^{min}} \leq 1$ and at $\Delta x_1 = 0$, $\frac{a_0^{min} + a_1^{min}}{2a_0^{min}} v_0 \geq v_1$. Therefore, $v_0 \geq v_1$ and hence $(C_1 \leq 0) \wedge (a_0^{min} \leq a_1^{min}) \wedge (P_2 \leq 0)$ cannot be true when *Collision*$_1$ occurs.
Case 3: $(C_2 \leq 0) \wedge (a_0^{min} \geq a_1^{min}) \wedge (P_2 \leq 0)$ is true. This implies that $\frac{a_1^{min}}{a_0^{min}} \geq 1$, $\frac{a_1^{min}}{a_0^{min}} v_0 \leq v_1$ and, at $\Delta x_1 = 0$, $v_1^2 - \frac{a_1^{min}}{a_0^{min}} v_0^2 - v_A^2 \leq 0$. These three inequalities imply that $(v_0 - v_1)^2 - v_A^2 \leq 0$.

In all cases where *Collision*$_1$ is possible, $0 < v_1 - v_0 \leq v_A$. Therefore $(v_0 - v_1)^2 \leq v_A^2$ and hence $(v_0' - v_1')^2 \leq v_A^2$ (by equation (3) and Assumption 2). Therefore, if *Collision*$_1$ occurs while $C \wedge (P_2 \leq 0)$ is true, $(P_1 \leq 0)$ will be true after the collision. Overall, if $(P_1 \leq 0) \vee [C \wedge (P_2 \leq 0)] \vee (v_1 \leq 0)$ is true when *Collision*$_1$ occurs it will also be true afterwards.

Assume at some state, s, $(P_1 \leq 0) \vee [C \wedge (P_2 \leq 0)] \vee (v_1 \leq 0)$ is true and consider the trajectories that start at this state. If $(P_1 \leq 0) \vee (v_1 \leq 0)$ is true at s it will also be true at the last state of the trajectory, by Lemma 3. If $C \wedge (P_2 \leq 0) \wedge (v_1 > 0)$ is true at s, consider the derivatives of the functions C_1, C_2 and P_2 along the trajectory:

$$\frac{d}{dt} C_1 = 2(a_0^{min} + a_1^{min}) v_0 \, acc_0 - 2a_0^{min} acc_0 v_1 - 2a_0^{min} v_0 \, acc_1 - 2(a_0^{min})^2 (v_0 - v_1)$$

$$= \begin{cases} 0 & \text{if } (v_0 > 0) \wedge \neg \mathit{Touching}_1 \\ 2(a_0^{min})^2 v_1 & \text{if } (v_0 = 0) \wedge \neg \mathit{Touching}_1 \\ 2(a_1^{min} v_0 - a_0^{min} v_1)\, acc_0 - 2(a_0^{min})^2 (v_0 - v_1) & \text{if } \mathit{Touching}_1 \end{cases}$$

$$\frac{d}{dt} C_2 = \frac{a_1^{min}}{a_0^{min}} acc_0 - acc_1$$

$$= \begin{cases} 0 & \text{if } (v_0 > 0) \wedge \neg \mathit{Touching}_1 \\ -a_1^{min} & \text{if } (v_0 = 0) \wedge \neg \mathit{Touching}_1 \\ \left(\frac{a_1^{min}}{a_0^{min}} - 1 \right) acc_0 & \text{if } \mathit{Touching}_1 \end{cases}$$

$$\frac{d}{dt} P_2 = 2 v_1\, acc_1 - 2 \frac{a_1^{min}}{a_0^{min}} v_0\, acc_0 + 2 a_1^{min} (v_0 - v_1)$$

$$= \begin{cases} 0 & \text{if } \neg \mathit{Touching}_1 \\ 2 \frac{a_0^{min} v_1 - a_1^{min} v_0}{a_0^{min}} acc_0 + 2 a_1^{min} (v_0 - v_1) & \text{if } \mathit{Touching}_1 \end{cases}$$

Consider first the variation of P_2. If $\mathit{Touching}_1 =$ False and as long as $v_1 > 0$, $\dot{P}_2 = 0$. Therefore, if $(P_2 \leq 0)$ is true at s, $(P_2 \leq 0) \vee (v_1 \leq 0)$ will be true at the last state of the trajectory. If $\mathit{Touching}_1 =$ True and $v_1 \neq v_0$ the trajectory stops (as the precondition of either $\mathit{Collision}_1$ or $\mathit{Separate}_1$ is satisfied). If $\mathit{Touching}_1 =$ True and $v_1 = v_0$ then $\dot{P}_2 = 2(a_0^{min} - a_1^{min}) v_0\, acc_0 / a_0^{min}$. If $a_0^{min} > a_1^{min}$ the trajectory stops and action $\mathit{Separate}_1$ occurs. Otherwise, $\dot{P}_2 \leq 0$, therefore $(P_2 \leq 0)$ will be true at the last state of the trajectory.

Now consider the variation of C. Recall that $C \wedge (v_1 > 0)$ is assumed to be true at s. Distinguish two cases:

Case A: $(C_1 \leq 0) \wedge (a_0^{min} \leq a_1^{min})$ is true at s. If $\mathit{Touching}_1 =$ False and as long as $v_1 > 0$ and $v_0 > 0$, $\dot{C}_1 = 0$. If $\mathit{Touching}_1 =$ True and $v_1 \neq v_0$ the trajectory stops (as the precondition of either $\mathit{Collision}_1$ or $\mathit{Separate}_1$ is satisfied). If $\mathit{Touching}_1 =$ True and $v_1 = v_0$ then $\dot{C}_1 = 2(a_1^{min} - a_0^{min}) v_0\, acc_0 \leq 0$ as $a_0^{min} \leq a_1^{min}$. Overall, $[(C_1 \leq 0) \wedge (a_0^{min} \leq a_1^{min})] \vee (v_0 = 0) \vee (v_1 \leq 0)$ will be true at the final state of the trajectory.

Case B: $(C_2 \leq 0) \wedge (a_0^{min} \geq a_1^{min})$ is true at s. If $\mathit{Touching}_1 =$ False and as long as $v_1 > 0$ and $v_0 > 0$, $\dot{C}_1 = 0$. If $\mathit{Touching}_1 =$ True and $v_1 \neq v_0$ the trajectory stops (as the precondition of either $\mathit{Collision}_1$ or $\mathit{Separate}_1$ is satisfied). If $\mathit{Touching}_1 =$ True and $v_1 = v_0$ then $\dot{C}_2 = (a_1^{min} - a_0^{min}) acc_0 / a_0^{min} \leq 0$, as $a_0^{min} \geq a_1^{min}$. Therefore, $[(C_2 \leq 0) \wedge (a_0^{min} \geq a_1^{min})] \vee (v_0 = 0) \vee (v_1 \leq 0)$ will be true at the final state of the trajectory.

Overall, if $(P_1 \leq 0) \vee [C \wedge (P_2 \leq 0)] \vee (v_1 \leq 0)$ is true at the first state of a trajectory, it will also be true at the last state. ■

Theorem 2 (Sufficient Condition for Pair Safety) *If I is initially true the pair is safe.*

Proof. I initially true and Lemma 5 imply $[P_1 \leq 0] \vee [C \wedge (P_2 \leq 0)] \vee (v_1 \leq 0)$ is an invariant property of the pair. If $(P_1 \leq 0) \vee (v_1 \leq 0)$ is true safety is guaranteed by Lemma 4. If $C \wedge (P_2 \leq 0)$ is true, the proof of Lemma 5 indicates that at $\Delta x_1 = 0$, $v_1 - v_0 \leq v_A$, which again implies safety. ■

Conditions under which the string is unsafe can be obtained in a similar way. Consider a derived boolean variable *Collided* which is initially false and becomes true when the actions $Collision_1$ occurs. Let:

$$C' = (C_1 \leq 0) \tag{15}$$

$$I' = [\neg C' \wedge (P_1 > 0)] \vee [(C' \vee (v_0 = 0)) \wedge (P_2 > 0)] \tag{16}$$

Theorem 3 (Necessary Condition for Pair Safety) *If* $I' \wedge (v_1 > 0) \wedge \neg\text{Collided}$ *is true initially then the pair is unsafe.*

The proof involves an argument similar to the one used for Theorem 2. The proof of Theorem 2 indicates that if the first collision is safe, all subsequent collisions will also be safe. The condition of Theorem 3 is therefore such that the *first* collision between the two vehicles is unsafe. More unsafe collisions may follow.

3.3 Safety Conditions for Strings of Length $N > 2$

Next, we derive a very simple sufficient condition for a string of arbitrary length to be safe. Even though the condition is conservative, interesting conclusions about the safety of platoons of vehicles can be derived from it (see Section 4). A string is *near uniform mass* if $\alpha_i(v) \equiv \alpha$ and $\alpha M_{k-1} \leq M_k \leq M_{k-1}/\alpha$. The near uniform mass condition allows us to put some bounds on the change of speed that a collision can induce. For example, it can be shown that:

Proposition 2 $\bigwedge_{i=0}^{N-1}(v_i \geq 0)$ *is an invariant property of a near uniform mass string.*

Recall that in general vehicles may end up going backwards due to a collision.

We construct invariant properties that allow us to characterize the safety of such a string. Let $\hat{a}_{min} = \min_{0\leq k<N} a_k^{min}$ and $\hat{a}_{max} = \max_{0\leq k<N} a_k^{min}$ and for $0 \leq i < j \leq N-1$ define $\Delta x_{ij} = \sum_{k=i+1}^{j} \Delta x_k$. For any pair of vehicles $i < j$, consider the function:

$$P(\Delta x_{ij}, v_i, v_j) = v_j - \frac{\hat{a}_{max}}{\hat{a}_{min}} v_i - v_A \tag{17}$$

Theorem 4 (Sufficient Condition for String Safety) *A near uniform mass string of* N *vehicles is safe if initially* $P(\Delta x_{ij}, v_i, v_j) \leq 0$ *for all* i, j *with* $0 \leq i < j \leq N-1$.

The proof is again by induction. Note that the conditions of Theorem 4 involve *all pairs in the string* and not just adjacent vehicles.

Finally, we establish conditions such that any string formed by a collection of vehicles satisfying:

$$a_i^{min} \in [\underline{a}, \overline{a}], \quad M_i \in [\underline{M}, \overline{M}], \quad \alpha_i(v) \equiv 1 \tag{18}$$

is guaranteed to be safe. Assume that all vehicles in the string are initially moving with velocity v.

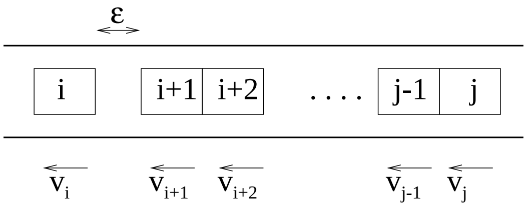

Fig. 3. Final configuration for theorem proof

Theorem 5 (Necessary Condition for String Safety) *All strings of N vehicles satisfying (18) are safe under the default deceleration strategy only if initially $(P_1(\mathit{\Delta} x_{ij}, v, v) \leq 0) \vee (P_2(\mathit{\Delta} x_{ij}, v, v) \leq 0)$ is true for all i, j with $0 \leq i < j \leq N-1$ and for all $a_i^{min}, a_j^{min} \in [\underline{a}, \overline{a}]$.*

Theorem 5 effectively states that a string may be unsafe if *any two vehicles in it are unsafe*. The proof is constructive: we show that, if two vehicles i and j violate the conditions of the theorem, one can chose the deceleration capabilities, a_k^{min}, and the masses, M_k, of vehicles $k = i+1, \ldots, j-1$ so that the string exhibits unsafe collisions. The idea of the construction is to bring the vehicles from their initial arrangement to the final arrangement of Figure 3, without any collisions taking place. The construction will be such that after resolving the multiple collision between vehicles $i+1, \ldots, j$ the velocity of vehicle $i+1$ will be the same as the velocity of vehicle j before the collision. For ϵ small enough, the next collision will be between vehicles $i+1$ and i and the relative velocity will be ϵ close to the relative velocity with which vehicles j and i would have collided if vehicles $i+1, \ldots, j-1$ were not there.

4 Implications for Platooning

We establish bounds on the system parameters (in particular the difference in deceleration capability between the vehicles) for a string to be safe. We start with the sufficient condition of Section 3.3. Consider a near uniform mass string and let $\overline{a} - \underline{a} = \epsilon$. Then, all strings whose vehicles satisfy (18) are guaranteed to be safe under the default deceleration strategy if $\left(1 - \frac{\overline{a}}{\underline{a}}\right) v - v_A \leq 0$ or equivalently:

$$\epsilon \leq -\frac{\underline{a} v_A}{v} \tag{19}$$

Substituting "typical" values of $\underline{a} = -9ms^{-2}$ and $v_A = 3ms^{-1}$ leads to $\epsilon \leq 1.08$ for $v = 25ms^{-1}$ and $\epsilon \leq 0.9$ for $v = 30ms^{-1}$.

For the necessary conditions of Section 3.3, note that:

$$\begin{array}{ll} \frac{\partial P_1}{\partial a_i^{min}} = -2\mathit{\Delta} x_{ij} \leq 0 & \frac{\partial P_1}{\partial a_j^{min}} = 2\mathit{\Delta} x_{ij} \geq 0 \\ \frac{\partial P_2}{\partial a_i^{min}} = \frac{a_j^{min}}{(a_i^{min})^2} v_i^2 \leq 0 & \frac{\partial P_2}{\partial a_j^{min}} = -\frac{v_i^2}{a_i^{min}} + 2\mathit{\Delta} x_{ij} \geq 0 \end{array}$$

N	ϵ (ms^{-2})		
	$v = 25ms^{-1}$, $F = 1m$	$v = 30ms^{-1}$, $F = 1m$	$v = 25ms^{-1}$, $F = 2m$
2	4.5	4.5	2.25
3	2.25	2.25	1.125
4	1.5	1.5	1.125
5	1.125	1.125	1.125
≥ 6	1.125	0.9	1.125

Table 1. Maximum allowable difference in deceleration capability

Therefore, the condition $(P_1(\Delta x_{ij}, v, v) \leq 0) \vee (P_2(\Delta x_{ij}, v, v) \leq 0)$ for all a_i^{min} and $a_j^{min} \in [\underline{a}, \overline{a}]$ is equivalent to $(P_1(\Delta x_{ij}, v, v) \leq 0) \vee (P_2(\Delta x_{ij}, v, v) \leq 0)$ for $a_i^{min} = \underline{a}$ and $a_j^{min} = \overline{a}$. To further simplify the calculation assume that initially the string is uniformly spaced, i.e. $\Delta x_i = F$ for all i. Then the necessary condition for string safety requires that for all $i \leq j$:

$$\epsilon \leq \max \left\{ \frac{v_A^2}{2(j-i)F}, \frac{2(j-i)\underline{a}^2 F - \underline{a} v_A^2}{v^2 - 2(j-i)\underline{a}F} \right\}$$

Table 1 shows the necessary condition for ϵ. The numbers indicate that the sufficient condition is conservative for small strings but approaches the necessary condition as the string size increases (the number for $N = 2$ in Table 1 is both necessary and sufficient).

If the string represents a platoon and based on the characteristics of vehicles on current highways, the bound on ϵ is reasonable for $N = 2$ but rather restrictive for higher platoon sizes (even under perfect road conditions). Note also that the calculation saturates after the first few vehicles; a similar observation was made in [6] about the increase in deceleration effort required along a platoon for "string stability". Overall, The above calculations indicate that the safety of the platooning system under emergency braking can only be guaranteed under rather limited conditions, in particular for small platoons consisting of vehicles of similar deceleration capabilities. This observation is in agreement with the numerical study of [9]. One can improve the situation by modifying the system parameters, by arranging the vehicles in a platoon in a particular order (e.g. in the order of increasing deceleration capability) and by designing better deceleration controllers. All these alternatives are the topic of current research.

5 Concluding Remarks

The string system introduced here is an interesting example for trying out different hybrid systems techniques. The system is simple enough to approach analytically, yet it can produce executions with very complex continuous-discrete interaction, even for string sizes as small as $N = 3$. Here we used induction arguments to answer safety questions; induction proofs are ideally suited to the structure imposed by the HIOA modeling formalism used to encode the system.

We are currently working on extending the results discussed here to account for phenomena like sensing and actuation uncertainties and delays. These extensions are likely to involve the use of simulation relations and abstraction mappings (similar analysis was carried out in [5] for a simpler system). We are also trying to investigate the effect of different deceleration strategies. Allowing different deceleration strategies makes the problem much more challenging; for example more sophisticated analysis techniques may be needed to ensure that the proposed controllers do not resort to "Zeno" executions to ensure the safety of the system[3]. The ultimate goal is of course to construct an optimal deceleration strategy for a each string; powerful optimal control tools are likely to be needed for this purpose. Hopefully solution to these problems will suggest ways in which control theory and computer science techniques can be used in tandem to address complicated questions in hybrid systems.

References

1. N. Lynch, R. Segala, F. Vaandrager, and H. Weinberg, "Hybrid I/O automata," in *Hybrid Systems III*, no. 1066 in LNCS, pp. 496–510, Springer Verlag, 1996.
2. A. Hitchcock, "Casualties in accidents occuring during split and merge maneuvers," tech. rep., PATH Technical Memo 93-9, Institute of Transportation Studies, University of California, Berkeley, 1993.
3. J. Lygeros, D. N. Godbole, and S. Sastry, "A game theoretic approach to hybrid system design," in *Hybrid Systems III*, no. 1066 in LNCS, pp. 1–12, Springer Verlag, 1996.
4. P. Li, L. Alvarez, R. Horowitz, P.-Y. Chen, and J. Carbaugh, "Safe velocity tracking controller for AHS platoon leader," in *IEEE Conference on Decision and Control*, pp. 2283–2288, 1996.
5. E. Dolginova and N. Lynch, "Safety verification for automated platoon maneuvers: a case study," in *Proceedings of HART97* (O. Maler, ed.), no. 1201 in LNCS, pp. 154–170, Berlin: Springer Verlag, 1997.
6. D. Swaroop, *String Stability of Interconnected systems: an application to platooning in automated highway systems.* PhD thesis, Department of Mechanical Engineering, University of California, Berkeley, 1994.
7. J. Lygeros, D. N. Godbole, and S. Sastry, "A verified hybrid controller for automated vehicles," Tech. Rep. UCB-ITS-PRR-97-9, Institute of Transportation Studies, University of California, Berkeley, 1997. (to appear in the Special Issue on Hybrid Systems of the IEEE Transactions on Automatic Control).
8. S. Shladover, "Operation of automated guideway transit vehicles in dynamically reconfigured trains and platoons," Tech. Rep. UMTA-MA-0085-79-3, U.S.Deprtement of Transportation, 1979.
9. D. N. Godbole and J. Lygeros, "Tools for safety and throughput analysis of automated highway systems," in *American Control Conference*, pp. 2031–2035, 1997.

[3] This is not an issue for the default deceleration strategy considered here, as it is easy to show that all vehicles come to a stop in finite time and after a finite number of collisions.