Verified Hybrid Controllers for Automated Vehicles

John Lygeros, *Member, IEEE*, Datta N. Godbole, *Member, IEEE*, and Shankar Sastry, *Fellow, IEEE*

***Abstract*— The objective of an Automated Highway System (AHS) is to increase the safety and throughput of the existing highway infrastructure by introducing traffic automation. AHS is an example of a large scale, multiagent complex dynamical system and is ideally suited for a hierarchical hybrid controller. We discuss the design of safe and efficient hybrid controllers for regulation of vehicles on an AHS. We use game theoretic techniques to deal with the multiagent and multiobjective nature of the problem. The result is a hybrid controller that by design guarantees safety, without the need for further verification. The calculations also provide an upper bound on the performance that can be expected in terms of throughput at various levels of centralization.**

***Index Terms*— Automated highway systems, game theory, safety.**

I. Introduction

HYBRID systems, i.e., systems involving the interaction of discrete and continuous dynamics, pose challenging problems which have attracted the attention of researchers from a number of diverse fields. Considerable research effort has been devoted to fundamental topics such as *modeling* [1]–[6] and *simulation* [7]–[9] of hybrid systems. The modeling effort has produced formalisms that allow one to synthesize controllers and verify properties of the closed-loop system performance. For *controller synthesis* the tools that have been proposed extend techniques from discrete-event control [10]–[13] and optimal control [5], [14], [15]. For *verification*, on the other hand, some progress was made in extending conventional analysis tools such as Lyapunov methods for stability analysis [5], [16]; however, most research has concentrated on extending discrete methodologies, in particular model checking [1], [2], [17], [18] and deductive [6], [19] techniques. For model checking, the emphasis has been on systems and properties that can be algorithmically verified. A number of computer-aided tools have been developed for systems where the model checking approach is applicable [20], [21]. For deductive reasoning, the emphasis has been on developing models that provide formal semantics for composition and abstraction and support proofs based on induction on the length of the system executions, invariant assertions, and simulation relations. Semi-automated tools (using, for example, automatic theorem provers) have been implemented to support this approach [22].

Much of the research on hybrid systems has been motivated by applications such as automotive electronics, real-time software, communication protocols, and transportation. One application that has attracted considerable attention is Automated Highway Systems (AHS). The goal is to improve the throughput of the highway system while maintaining (at least) the same level of safety and passenger comfort as the current system. This is to be achieved by automating traffic without building new highways. Clearly, an AHS poses a very complex control problem. Part of the complication comes from its distributed nature. The system consists of a large number of agents (the vehicles) equipped with sensing, communication, and control capabilities that are trying to make efficient use of a common, scarce resource (the highway). Difficulties also arise from the multiple control requirements. The controller has to strive to satisfy many specifications, some of them conflicting. For example, the requirement for increased throughput implies that the vehicles should be traveling fast and close to one another, while the requirement for safety implies that they should be traveling slowly and with large spacings.

In [15] we introduced a methodology for dealing with such multiagent, multiobjective problems. Our method assumes that the requirements are prioritized; in the AHS case safety should be more important than throughput, for example. Then, assumptions are made about the flow of information between the agents. Typically the system is initially assumed to be fully decentralized, i.e., each agent has only access to local information, available through its sensors. The controller design is then seen as a game between the controller of each agent and the disturbance generated by the environment, which includes the actions of other agents. The design process establishes the limits of performance that can be achieved in "safety"; the lower priority requirements are then optimized within those limits. The resulting *emergent behavior* of the system (in the AHS case the throughput) is then extracted. If the level of performance is unsatisfactory, some centralization needs to be introduced. This is achieved by communication of information between the agents. Our calculations suggest what the most important pieces of information are. The effect of this more global knowledge is to reduce the set of possible disturbances generated by other agents. The process can be repeated: the new "biased" game is solved to determine the limits of safety, the low priority requirements are optimized within those limits, and the new emergent behavior is extracted.

In this paper we apply this methodology to the AHS problem. We first consider a decentralized design involving completely autonomous vehicles and then see how performance

Manuscript received August 2, 1996. Recommended by Associate Editor, P. J. Antsaklis. This work was supported in part by the Army Research Office under Grants DAAH 04-95-1-0588 and DAAH 04-96-1-0341 and by the California PATH program, Institute of Transportation Studies, University of California, Berkeley, under Grant MOU-238.

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720-1770 USA (e-mail: lygeros@eecs.berkeley.edu).

Publisher Item Identifier S 0018-9286(98)02662-2.

can be improved by allowing semi-autonomous vehicles that can form platoons. In either case our game theoretic approach is used to design controllers that are guaranteed to satisfy the safety requirements. The calculations also provide upper bounds on the throughput that can be expected in each case. Here we mainly address the problem of maintaining safety in this multiagent setting. The multiobjective nature of the system is also briefly discussed, but for more details the reader is referred to [23]. For application of similar techniques to Air Traffic Management Systems see [24].

The question of safety is one of the most important issues for AHS and has therefore attracted considerable attention. In the context of platooning, the issue of controller design and safety for platoon followers has been addressed extensively in [25]. For autonomous vehicles and platoon leaders a number of controllers have been proposed [26]–[28], some without explicit safety guarantees. Reference [29] highlighted problems, due primarily to the hybrid nature of the system, that may arise if such guarantees are not provided. The work of [30] and [31] explicitly address the issue of safety in terms of the continuous dynamics, assuming the deceleration capability of all vehicles is uniform and without taking into account the discrete disturbances generated by possible vehicle collisions. Similar calculations for different deceleration capabilities can be found in [28] and [32]. The work presented here extends preliminary results reported in [33].

In Section II, we give a brief overview of the modeling formalism introduced in [34] and use it to model the vehicles. In Section III our design methodology is applied to the case of autonomous vehicles. We derive conditions for safety and upper bounds on the throughput that can be expected. In Section IV we show how the throughput can be increased by introducing partial centralization and discuss the concept of platooning. Sufficient conditions that guarantee the safety of an AHS that supports platooning are given in Section V. In the concluding Section VI we discuss some of the key points of our design methodology and some directions for future work. To maintain the flow of the paper some of the more technical proofs are given in Appendix B.

II. Problem Formulation

A. Modeling Formalism and Dynamic Games

We briefly present the hybrid system modeling formalism that will be used to describe various components of the system. For more details the reader is referred to [34].

Definition 1: A *hybrid automaton* H is a collection (X, U, Y, I, f, E, h), with $X = X_D \times X_C$, $U = U_D \times U_C$, $Y = Y_D \times Y_C$, $I \subset X$, $f : X \times U \to TX_C$, $E \subset X \times U \times X$, and $h : X \times U \to Y$, where X_C, U_C, Y_C are, respectively, open subsets of R^n, R^m, R^p for some finite values of n, m, p, and X_D, U_D, Y_D are countable sets.

X, U, and Y are referred to as the state, input, and output spaces. TX_C represents the tangent space of X_C. We use $u \in U$ to denote the input variables, $y \in Y$ to denote the output variables, and $(q, x) \in X_D \times X_C$ to denote the state variables. We assume that discrete sets are given the discrete topology while continuous sets are given the Euclidean topology. Without loss of generality we assume that f is time invariant.

A hybrid system describes the evolution of the variables over time. We assume a set of times of interest of the form $T = [t_i, t_f] \subset R$. The variables will evolve either continuously or in instantaneous jumps. Therefore, the evolution of the system will be over sets of the form $[\tau_0', \tau_1][\tau_1', \tau_2] \cdots [\tau_{n-1}', \tau_n]$ with $\tau_j \in T$ for all j, $\tau_0' = t_i, \tau_n = t_f$, and $\tau_j = \tau_j' \leq \tau_{j+1}$ for all $j = 1, 2, \cdots, n-1$. The implication is that τ_j are the times where discrete jumps occur. We will use $\mathcal{T}$ to denote the set of all such "super-dense" time trajectories and τ to denote an element of $\mathcal{T}$.

Definition 2: A *run* of the hybrid dynamical system H over an interval $T = [t_i, t_f]$ is a collection (τ, q, x, u, y) with $\tau \in \mathcal{T}$, $q : \tau \to X_D$, $x : \tau \to X_C$, $u : \tau \to U$, and $y : \tau \to Y$ satisfying the following.

1) *Initial Condition*: $(q(\tau_0'), x(\tau_0')) \in I$.
2) *Discrete Evolution*: for all i either $(q(\tau_i), x(\tau_i), u(\tau_i), q(\tau_i'), x(\tau_i')) \in E$ and $(q(\tau_i), x(\tau_i)) \neq (q(\tau_i'), x(\tau_i'))$ or $u(\tau_i') \neq u(\tau_i)$.
3) *Continuous Evolution*: for all i with $\tau_i' < \tau_{i+1}$ and for all $t \in [\tau_i', \tau_{i+1}]$, $\dot{x}(t) = f(q(t), x(t), u(t))$, $q(t) = q(\tau_i')$ and $(q(t), x(t), u(t), q(t), x(t)) \in E$.
4) *Output Evolution*: for all $t \in \tau$, $y(t) = h(q(t), x(t), u(t))$.

It should be noted that existence and/or uniqueness of runs cannot be guaranteed. In fact, one can very easily construct hybrid automata that for some initial conditions and/or inputs accept multiple runs or no runs at all. These issues are of theoretical interest but need not concern us in this paper; the automata used to model the vehicles will accept unique runs.

Let $\{H_i\}_{i=1}^N$ be a collection of hybrid automata $H_i = (X_i, U_i, Y_i, I_i, f_i, E_i, h_i)$. We can write the inputs and outputs in vector form as $u_i = [u_{i,1} \ \cdots \ u_{i,m_i}]^T \in U_i$ and $y_i = [y_{i,1} \ \cdots \ y_{i,p_i}]^T \in Y_i$. Let

$$\hat{U} = \{(1,1), (1,2), \cdots, (1,m_1), (2,1), \cdots, (2,m_2), \cdots, (N,1), \cdots, (N,m_N)\}$$

$$\hat{Y} = \{(1,1), (1,2), \cdots, (1,p_1), (2,1), \cdots, (2,p_2), \cdots, (N,1), \cdots, (N,p_N)\}.$$

Definition 3: An *interconnection* of a collection $\{H_i\}$ of automata is a partial map $\mathcal{I} : \hat{U} \longrightarrow \hat{Y}$.

The interpretation is that an interconnection constrains some input variables to have the same value as some output variables. It can be shown that under some mild consistency assumptions an interconnection of hybrid automata defines a new hybrid automaton whose state space is the product of the state spaces of the original automata, its output space is the product of the output spaces, and its input space is the product of the input spaces minus the pre-image of $\mathcal{I}$.

To simplify the subsequent discussion we introduce a special kind of boolean variable, called an *event*. The event is said to *occur* whenever its value changes from "True" to "False" or vice versa. In the computer science literature events are typically associated with discrete transitions in the state of an automaton. Here events are treated as discrete input and output

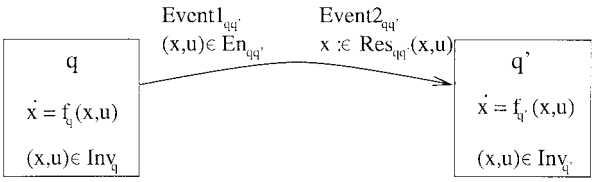

Fig. 1. Typical transition of a hybrid automaton.

variables as in [35] and play the role of the input and output "actions" of [6]. It is easy to show that for a given hybrid automaton H, one can construct a new automaton $\hat{H}$ whose evolution is identical to that of H with the exception that an event occurs upon particular transitions. This can be done by adding a boolean state and a boolean output variable for each event. Similarly, by adding boolean input and a boolean state variable an automaton can be forced to take an enabled transition upon the occurrence of an event. Note that the addition of each event doubles the cardinality of X_D.

If X_D is finite, the hybrid automaton is typically presented as a directed graph [1], [2] whose nodes are indexed by the elements of X_D. While the discrete state is "in a node" q, the continuous state evolves along a vector field, $f_q : X_C \times U \to TX_C$ with $f_q(x,u) = f(q,x,u)$. To each node q we associate an invariant $\mathrm{Inv}_q = \{(x,u) \in X_C \times U \mid (q,x,u,q,x) \in E\}$; the interpretation is that the system can remain in node q if and only if $(x,u) \in \mathrm{Inv}_q$. The discrete evolution is captured by transitions between the nodes. To the transition from node q to node q' we associate a guard $\mathrm{En}_{qq'} = \{(x,u) \in X_C \times U \mid \exists x' \in X_C, (q,x,u,q',x') \in E\}$; the interpretation is that the transition can take place if and only if $(x,u) \in \mathrm{En}_{qq'}$. To each transition we also associate a set valued map $\mathrm{Res}_{qq'}(x,u) = \{x' \in X_C \mid (q,x,u,q',x') \in E\}$; the interpretation is that if the transition takes place from (x,u), then after the transition the state can find itself in any (q',x') with $x' \in \mathrm{Res}_{qq'}(x,u)$. A typical edge of the graph is shown in Fig. 1. By convention, the guard appears first and the reset relation (denoted by $:\in$ or $:=$) appears second. To simplify the figures we list an input event $(\mathrm{Event1}_{qq'})$ that triggers a transition (if any) together with the transition guard and an output event $(\mathrm{Event2}_{qq'})$ triggered by a transition together with the transition reset map. Figures like these will be used subsequently as formal definitions of hybrid automata. On some occasions we will use similar figures informally to convey an idea rather than define a system. In this case we will use circles instead of squares to distinguish the informal automata from the formal ones.

Motivated by [36] we consider the following type of two-player zero sum games.

Definition 4: A *two-player zero sum dynamic game* $(T, \mathcal{X}, \mathcal{U}, \mathcal{D}, \mathcal{Y}, H, J)$ consists of a time interval $T = [t_i, t_f]$, a trajectory space $\mathcal{X}$, an input space $\mathcal{U}$, a disturbance space $\mathcal{D}$, an output space $\mathcal{Y}$, a hybrid automaton H, and a cost function $J : \mathcal{Y} \times \mathcal{U} \times \mathcal{D} \to R$. Player 1 is said to control the input $u \in \mathcal{U}$ while player 2 is said to control the disturbance $d \in \mathcal{D}$. A collection of $(x,u,d,y) \in \mathcal{X} \times \mathcal{U} \times \mathcal{D} \times \mathcal{Y}$ is an *admissible play* if there exists $\tau \in \mathcal{T}$ for which $(\tau, x, (u,d), y)$ is a run of H.

We will assume that the *reward* of player 1 for a given play is $-J(y,u,d)$, while the reward of player 2 is $J(y,u,d)$; player 1 is trying to minimize J while player 2 is trying to maximize it. We will also assume a *closed-loop perfect state information structure*; when called upon to decide their strategy at time t, both players have access to the entire state trajectory up to that point. For simplicity we therefore set $\mathcal{Y} = \mathcal{X}$.

If the hybrid automaton H admits a unique run for every initial condition, input, and disturbance trajectory, one can write the cost function as simply

$$J : X \times \mathcal{U} \times \mathcal{D} \longrightarrow R.$$

For this class of games we will be interested in the following type of solutions.

Definition 5: A *global saddle solution* to the two-player zero sum game is a pair of input and disturbance trajectories $(u^*, d^*) \in \mathcal{U} \times \mathcal{D}$ such that for all $(q^0, x^0) \in X$, all $u \in \mathcal{U}$, and all $d \in \mathcal{D}$

$$J((q^0,x^0),u^*,d) \le J((q^0,x^0),u^*,d^*) \le J((q^0,x^0),u,d^*).$$

A saddle solution is such that any unilateral deviation from it leaves the player who decided to deviate in worse condition. The games considered in this paper turn out to have unique saddle solutions. Existence and uniqueness of solutions cannot be guaranteed in general, however.

B. Vehicle Model

Consider a number of vehicles moving on an AHS (Fig. 2).[1] We try to capture the evolution of the pair $A - B$ and use C and D to isolate them from the rest of the lane. Vehicles in adjacent lanes (E,H) will only come into play during lane changes. The model we develop is hybrid: the continuous dynamics reflect the movement of the vehicles, while the discrete dynamics reflect possible collisions between them. In our safety calculations we impose the following restrictions.

Assumption 1: The operation of all vehicles is normal, all vehicle parameters are known, and lateral control is perfect. Each vehicle is equipped with sensors that provide noiseless continuous measurements of its velocity and acceleration as well as distance and relative velocity with respect to vehicles in front, both in the same and in adjacent lanes. Sensing and actuation delays are negligible.

The assumption of normal operation is crucial in proving safety. If faults or adverse environmental conditions are allowed, the design of safe controllers becomes substantially more complicated [37]. Knowledge of the vehicle parameters (aerodynamic coefficients, mass, etc.) is needed for feedback linearization of the vehicles dynamics (see [38]). This assumption can be relaxed by an adaptive [38], robust, or sliding mode controller [39]. Under normal conditions near-perfect lateral control has been shown to be possible both for lane keeping and for lane changing in [40] and [41]. We further assume that perfect lateral operation can be maintained even in the presence of collisions. Finally, the assumed sensor arrangement is realistic for current technology.

[1] We will keep referring to the notation of Fig. 2 throughout the paper.

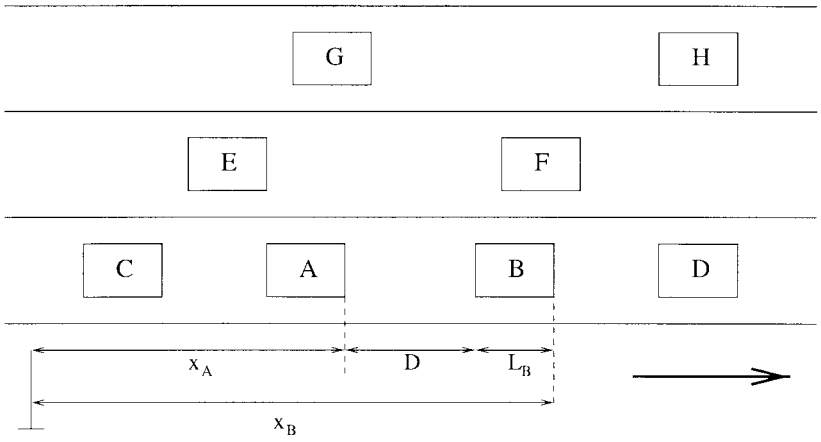

Fig. 2. The model setup.

The calculations presented here easily generalize to vehicle models with sensing and actuation delays, lags, and small additive noise [42]. We assume that the sensor ranges are large enough to maintain safety.

The automaton describing the evolution of the pair $A-B$ has three discrete states $q_A \in X_D = \{q_1, q_2, q_3\}$ and four continuous states. In q_1 vehicle A is moving without touching vehicle B, in q_2 vehicles A and B are moving while pushing against one another, and in q_3 vehicle A has stopped. Let L_i denote the length of vehicle i and x_i its position from a fixed roadside reference. The dynamics and the safety requirements do not depend on the absolute position of the vehicles. Therefore, to reduce the size of the continuous state space we introduce a variable $D_{AB} = x_B - x_A - L_B$ to keep track of the spacing between vehicles A and B. Following [27], we assume that (after feedback linearization) the controller of vehicle A can directly set the *jerk* $u_A = \dddot{x}_A$ through brakes and accelerator actuators. By Assumption 1 vehicle A is equipped with sensors to measure its own velocity and acceleration and the spacing and relative velocity with respect to vehicle B. The acceleration of vehicle B is assumed to be unknown to vehicle A and is treated as a disturbance. The continuous dynamics in states q_1 and q_3 can now be described by a state vector $x_{AB} = [\dot{x}_A \;\; \ddot{x}_A \;\; D_{AB} \;\; \dot{D}_{AB}] \in R^4$ with

$$\dot{x}_{AB} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 \end{bmatrix} x_{AB} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} u_A + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \ddot{x}_B \triangleq Ax_{AB} + Bu_A + D\ddot{x}_B. \quad (1)$$

In state q_2, the continuous dynamics can be more complicated. While vehicles A and B are touching, the acceleration of vehicle A depends not only on the jerk commanded by its controller, but also on the contact forces exerted by vehicle B. State q_2 is not safety critical however, as it is impossible for vehicles A and B to collide while they are touching each other. We will therefore not model the continuous dynamics in state q_2 in detail, we will just assume that they are of the form $\dot{x}_{AB} = f(x_{AB}, u_A, \ddot{x}_B)$ for some function f which is Lipschitz continuous in x_{AB} and continuous in u_A and $\ddot{x}_B$. To simplify the notation, we drop the subscripts whenever it is clear we are referring to the pair $A-B$ and use q and $x = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T$ to denote the continuous state of the pair and u to denote the input of vehicle A. Physical considerations impose constraints on the continuous state and inputs

$$x \in X_C = \left\{ x \in R^4 \mid x_2 \in \left[a_A^{\min}, a_A^{\max}\right] \right\}$$
$$u \in U = \left[j_A^{\min}, j_A^{\max}\right], \quad \ddot{x}_B \in D = \left[a_B^{\min}, a_B^{\max}\right].$$

To ensure our model is realistic, we impose some minimal assumptions on the constraints.

Assumption 2: $a_A^{\min} < 0 < a_A^{\max}$, $a_B^{\min} < 0 < a_B^{\max}$, and $j_A^{\min} < 0 < j_A^{\max}$.

The transitions of the discrete state are governed by the obvious invariants, guards, and reset relations. Additional discrete dynamics arise because of possible collisions between the vehicles. For simplicity we assume the following.

Assumption 3: All collisions are elastic and all vehicles have equal masses.

The case of partially inelastic collisions and unequal masses can be included by introducing some additional notation and minor changes in the subsequent calculations. Three kinds of collisions affect the pair $A-B$. Collisions between vehicles A and B take place whenever $(x_3 = 0) \wedge (x_4 < 0)$ and will be used to determine the safety of the system. Upon their occurrence an output event $\mathrm{Coll_A}$ is issued, x_1 is reset to $x_1 + x_4$, and x_4 is reset to $-x_4$. Collisions between vehicles B and D, and between vehicles C and A, are treated as disturbances by vehicle A. They are modeled by an input event $\mathrm{Coll_B}$ (respectively, $\mathrm{Coll_C}$) and a continuous input $\delta v_B \geq 0$ (respectively, $\delta v_C \geq 0$). Upon their occurrence, x_4 is reset to $x_4 - \delta v_B$ (respectively, x_4 is reset to $x_4 - \delta v_C$ and x_1 is reset to $x_1 + \delta v_C$). Clearly the value of δv_B (δv_C) is of interest only at the times when $\mathrm{Coll_B}$ ($\mathrm{Coll_C}$) occur. We use $d_{AB} = (\ddot{x}_B, \mathrm{Coll_B}, \delta \mathrm{v_B}, \mathrm{Coll_C}, \delta \mathrm{v_C})$ to denote the overall disturbance trajectory for vehicle A. The subscript of d will again be omitted when the context is clear.

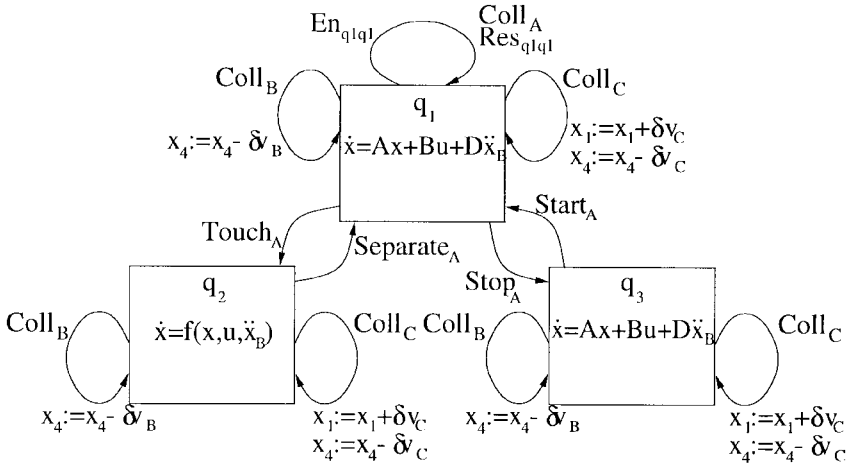

Fig. 3. The plant automaton.

Without loss of generality we assume that a trajectory starts at time $t_i = 0$ from an initial condition $(q^0, x^0) \in X$.

Proposition 1: For every $(q^0, x^0) \in X$ and every u and d piecewise continuous the plant automaton accepts a unique trajectory.

Note that piecewise continuous is equivalent to piecewise constant under the discrete topology.

A standard requirement for automated vehicles is that they will not be allowed to drive in reverse. This requirement can easily be added to our model.

Lemma 1: If for all vehicles $x_1^0 \geq 0$ and $q = q_3$ implies $u \geq 0$, then $x_1(t) \geq 0$ for all $t \geq 0$. In this case $\mathrm{Coll_A}$ can never occur in states q_2 and q_3 and the set of disturbances generated by vehicles B and C is contained in $\mathcal{D} = \{d \mid d$ piecewise continuous, $\forall t : \ddot{x}_B(t) \in [a_B^{\min}, a_B^{\max}], x_1(t) + x_4(t) \geq 0, \delta v_B(t) \in [0, x_1(t) + x_4(t)], \delta v_C(t) \geq 0\}$.

We use $\mathcal{U}$ to denote the class of controllers that apply $u \geq 0$ whenever $q = q_3$. All controllers designed in this paper will belong to the class $\mathcal{U}$. Under the conditions of Lemma 1 the dynamics of the pair $A - B$ are summarized in Fig. 3. The invariants, guards, and reset relations omitted in the figure are given in Appendix A. Our analysis is primarily concerned with safety, therefore (unless otherwise stated) we will assume $q^0 = q_1$ in subsequent analysis, as the vehicles cannot collide otherwise.

Lemma 1 suggests that $\mathcal{D}$ is a conservative estimate of the disturbances that vehicle A may face. Therefore, if controllers are shown to be safe with respect to $\mathcal{D}$, they are guaranteed to be safe for the real system. Further restrictions can be imposed on d through interconnections of the plant automaton with the outputs of appropriate automata that model the behavior of the controllers of vehicles B and C. In our design we will make use of a restriction that allows a single $\mathrm{Coll_B}$ and a single $\mathrm{Coll_C}$ event and limits the relative velocity of the corresponding collisions to v_B and v_C, respectively. In this case the overall disturbance can be compactly parameterized by $d = (\ddot{x}_B, T_B, \delta v_B(T_B), T_C, \delta v_C(T_C))$, where T_B (T_C) is the time $\mathrm{Coll_B}$ $(\mathrm{Coll_C})$ occurs. The class of allowable disturbances has the form

$$\begin{aligned}\mathcal{D}_{v_B v_C} = \{d \mid \ddot{x}_B \text{ piecewise continuous}, \forall t : \ddot{x}_B(t) \\ \in [a_B^{\min}, a_B^{\max}], x_1(t) + x_4(t) \geq 0 \\ T_B \geq 0, T_C \geq 0, \delta v_B(T_B) \in [0, \min\{v_B, x_1(t) \\ + x_4(t)\}], \delta v_C(t) \in [0, v_C]\}. \end{aligned} \quad (2)$$

C. Design Specifications

Ideally, we want no collisions to occur on the AHS. Our analysis will indicate that this requirement may be too restrictive in terms of throughput. Previous studies [43] suggest that collisions with small relative velocities are not likely to result in serious damage to the vehicles or injury to the passengers. This motivates the following definition.

Definition 6: A *safe collision* between two vehicles is one where the relative velocity at impact is less than a threshold v_a. An Automated Highway System where vehicles experience only safe collisions is called a *safe AHS*.

Based on the analysis of [43], we set $v_a = 3$ ms^{-1} in our calculations.

For each vehicle we treat the controller design as a two-player zero sum game between the vehicle controller u and the disturbance d. The game is over a number of cost functions that reflect the various requirements imposed on the controller. The top priority requirement is, of course, safety. Whenever possible, we would like the vehicles not to collide at all. This requirement can be encoded by a cost function

$$J_1(q^0, x^0, u, d) = -\inf_{t \geq 0} x_3(t). \quad (3)$$

A controller is safe with respect to J_1 if it can guarantee that $J_1(q^0, x^0, u, d) \leq C_1 \triangleq 0$ m. The limiting case of "no collision" corresponds to two vehicles touching each other with zero relative velocity.

In our calculations we will also have to deal with situations where a collision is unavoidable. In this case, the goal of the controller is to minimize the relative velocity at impact. This alternative safety requirement can be encoded by

$$J_1'(q^0, x^0, u, d) = |x_4(t')| \quad (4)$$

where $x_3(t') = 0$. A controller will be safe with respect to J_1' if it can guarantee that $J_1'(q^0, x^0, u, d) \leq C_1' \triangleq 3\ \text{ms}^{-1}$.

Whenever safety is not an issue, the controller is required to provide a comfortable ride for the passengers. We assume that comfort implies small control inputs.[2] We encode comfort using the cost function

$$J_2(q^0, x^0, u, d) = \sup_{t \geq 0} |u(t)|. \tag{5}$$

A maneuver will be comfortable if $J_2(q^0, x^0, u, d) \leq C_2 \triangleq 2.5\ \text{ms}^{-3}$.

If neither safety nor comfort are at stake, we would like the controller to converge to a steady-state spacing from the vehicle in front and a desired velocity v_H. These performance requirements can be encoded by an *efficiency* cost function

$$J_3(q^0, x^0, u, d) = \int_0^\infty (x(t) - x_d)^T P (x(t) - x_d)\, dt \tag{6}$$

where x_d reflects the desired spacing and velocity and P is a positive semidefinite matrix.

The design methodology of [15] allows us to deal with all the cost functions discussed above by solving a sequence of nested games. In this paper, we are primarily concerned with the issue of safety and will therefore discuss the issues of passenger comfort and efficiency only briefly.

III. Autonomous Vehicles

We first derive safety conditions for the simplest possible automated highway system, a decentralized design where no collisions take place. We refer to this design as an *autonomous vehicle AHS*. We start with a single lane highway and concentrate on an arbitrary vehicle A. Our goal is to avoid collisions altogether, therefore we treat the safety problem as a game between the control (action of vehicle A) and the disturbance (actions of vehicles B and C) over cost function J_1. The safety calculation proceeds by assuming no collisions are possible, designing safe controllers under this assumption, and then verifying that if the resulting controller is used, the "no collision" assumption is indeed satisfied.

A. Autonomous Vehicle Safety

Consider an arbitrary vehicle A. If no collisions take place, the only disturbance to vehicle A comes from $\ddot{x}_B$, i.e., $d \in \mathcal{D}_{00}$. Formally this is equivalent to an interconnection of the Coll_B and Coll_C input events with the outputs of an automaton that never generates Coll_B and Coll_C. Consider the candidate feedback saddle solution $(u_1^*, d_{1_{00}}^*)$ given by

$$u_1^*(q, x) = \begin{cases} j_A^{\min}, & \text{if } \left(x_2 > a_A^{\min}\right) \wedge (q \neq q_3) \\ 0, & \text{if } \left(x_2 = a_A^{\min}\right) \vee (q = q_3) \end{cases}$$

and

$$\ddot{x}_{B1}^*(q, x) = \begin{cases} a_B^{\min}, & \text{if } x_1 + x_4 > 0 \\ 0, & \text{if } x_1 + x_4 = 0. \end{cases} \tag{7}$$

Clearly $(u_1^*, d_{1_{00}}^*) \in \mathcal{U} \times \mathcal{D}_{00}$. Let $(q^*(t), x^*(t))$ denote the value of the state at time t under $(u_1^*, d_{1_{00}}^*)$ starting at (q^0, x^0).

[2]This is a necessary but not sufficient condition for comfort, as low-frequency speed variations can also result in an uncomfortable ride.

Proposition 2 (Saddle Solution for Autonomous Operation): If $x^0 \in X_C$, then $x^*(t) \in X_C$ for all $t \geq 0$. $(u_1^*, d_{1_{00}}^*)$ is a global saddle solution for cost $J_1(q^0, x^0, u, d)$.

Proof: Verify that for all u and d, $J_1(q^0, x^0, u_1^*, d) \leq J_1(q^0, x^0, u_1^*, d_{1_{00}}^*) \leq J_1(q^0, x^0, u, d_{1_{00}}^*)$ [32]. □

The saddle solution allows us to determine the set of *safe states* and classify all the *safe controls*. Let $\text{int}(V)$ denote the interior and ∂V the boundary of a subset V of X and define

$$V_1 \triangleq \left\{(q, x) \in X \mid J_1\left(q, x, u_1^*, d_{1_{00}}^*\right) \leq C_1\right\} \subset X \tag{8}$$

$$\mathcal{U}_1(q, x) \triangleq \left\{ \begin{cases} \emptyset, & \text{if } x \in X \backslash V_1 \\ u_1^*(q, x), & \text{if } x \in \partial V_1 \\ \left[j_A^{\min}, j_A^{\max}\right], & \text{if } x \in \text{int}(V_1) \wedge q \neq q_3 \\ \left[0, j_A^{\max}\right], & \text{if } x \in \text{int}(V_1) \wedge q = q_3 \end{cases} \right\} \subset \mathcal{U}. \tag{9}$$

Lemma 2 (Safe States and Safe Controls for Autonomous Operation): If $(q^0, x^0) \in V_1$, $d \in \mathcal{D}_{00}$, and for all $t' \in [0, t]$, $u(t') \in \mathcal{U}_1(q(t'), x(t'))$, then $x(t) \in V_1$ and vehicles A and B will not collide. If $(q^0, x^0) \notin V_1$, then for any u there exists a trajectory $d \in \mathcal{D}_{00}$ that results in a collision.

Proof: The first part follows from the properties of the saddle solution and the definitions of V_1 and $\mathcal{U}_1$. For the second part, assume $(q^0, x^0) \notin V_1$ and for all $t \geq 0$ consider $\ddot{x}_B(t) = \ddot{x}_{B1}^*(x(t))$. The conclusion follows by the properties of the saddle solution and the definition of V_1. □

V_1 is what is known as a *control invariant set*, a set that can be rendered invariant by u, despite the actions of d. Lemma 2 indicates that V_1 is the largest control invariant set for which safety can be guaranteed. $\mathcal{U}_1$ is the *least restrictive class of controls* that renders V_1 invariant.

The saddle solution calculations allow us to algebraically determine the set of safe states V_1 (and hence the set of safe controls $\mathcal{U}_1$). For $q^0 = q_1$, ∂V_1 can be encoded by a function

$$\begin{aligned} s &: R^3 \longrightarrow R \\ \left(x_1^0, x_2^0, x_4^0\right) &\longmapsto x_3^0 = s\left(x_1^0, x_2^0, x_4^0\right) \end{aligned}$$

that returns the minimum spacing required for no collision for a given choice of initial velocity, acceleration, and relative velocity. s is parameterized by $a_A^{\min}$, $a_B^{\min}$, and $j_A^{\min}$.

Theorem 1 (Autonomous Vehicle Safety): A single lane AHS populated by a finite number of vehicles all of which satisfy $(q^0, x^0) \in V_1$ and $u(t) \in \mathcal{U}_1(q(t), x(t))$ for all $t \geq 0$ is safe.

Proof: From Lemma 2 and an induction argument on the number of vehicles it follows that no collisions are obtained on such an AHS. Therefore, the AHS is safe. □

The analysis generalizes to a multilane highway by adding simple lane change protocols and control laws. Section V-A3 describes how V_1 can be used to derive conditions under which a lane change can be executed in safety for a more complicated AHS design that involves platooning. The results trivially generalize to the autonomous vehicle case.

B. Passenger Comfort and Efficiency

Having established conditions for safety, we now turn our attention to passenger comfort and efficiency. Formally, the

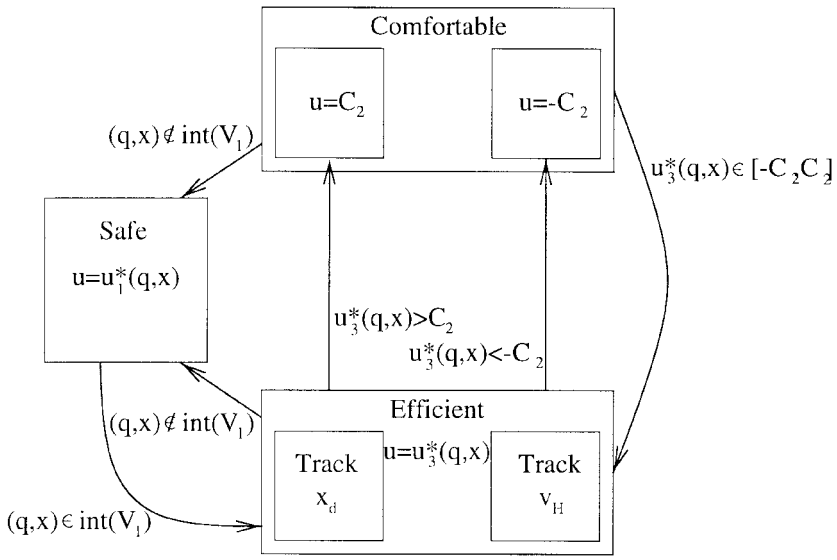

Fig. 4. Hybrid controller automaton.

comfort requirement can be treated as a game between u and d over cost function J_2. We seek a saddle solution (u_2^*, d_2^*)

$$J_2(q^0, x^0, u_2^*, d_2^*) = \max_{d \in \mathcal{D}_{00}} \min_{u \in \mathcal{U}_1(q,x)} J_2(q^0, x^0, u, d) = \min_{u \in \mathcal{U}_1(q,x)} \max_{d \in \mathcal{D}_{00}} J_2(q^0, x^0, u, d).$$

The requirement that $u \in \mathcal{U}_1(q,x)$ implies that such a saddle solution is safe but can only exist if $(q,x) \in V_1$. The saddle solution itself is not very important in this case, all we need is the set of safe and comfortable controls, given by

$$\mathcal{U}_2(q,x) \triangleq \begin{cases} \emptyset, & \text{if } x \in X \backslash V_1 \\ u_1^*(q,x), & \text{if } x \in \partial V_1 \\ [-C_2, C_2], & \text{if } x \in \text{int}(V_1) \wedge q \neq q_3 \\ [0, C_2], & \text{if } x \in \text{int}(V_1) \wedge q = q_3. \end{cases} \tag{10}$$

To complete the design, the requirement for efficiency should also be addressed. We will not go into the details of the efficient design. This problem can be approached in a number of ways and the solution does not affect safety in any way. One possible design could make use of the saddle solution (u_3^*, d_3^*), for cost function J_3 with $u_3^* \in \mathcal{U}_2(q,x)$; other designs like the ones in [27] and [28] may be even more appropriate. To make sure the design is reasonable, we will only assume that the desired steady-state spacing satisfies $(q_1, x_d) \in \text{int}(V_1)$ and that the desired velocity v_H is never exceeded. These objectives can be achieved by a controller that switches between tracking spacing and tracking v_H, based on state information [27]. The resulting safe, comfortable, and efficient controller can be implemented by a hybrid automaton (Fig. 4). The controller automaton has no continuous state. Its input (q,x) and its output u are provided from/to the plant automaton through the obvious interconnection.

C. Autonomous Vehicle Throughput

Calculating the expected throughput for a proposed AHS design is a very challenging task. Our analysis can be used to obtain an upper bound on the steady-state throughput. Lemma 2 provides the minimum spacing required for safety if a steady-state velocity x_1^0 is maintained. This quantity gives rise to what is known in the transportation literature as the *pipeline throughput*, the steady state per lane capacity of the highway in the absence of lateral flow (lane changes, entry, and exit). Assuming all vehicles have the same length $L_i = L$, the pipeline throughput Q (in vehicles per lane per unit time) is given by

$$Q = \frac{x_1^0}{s(x_1^0, 0, 0) + L}. \tag{11}$$

This is clearly a very optimistic estimate. Even if the demand is high enough, the actual value of the throughput is likely to be much smaller because of safety margins (added to s to account for delays, sensor noise, etc.) and lateral flow disturbances. Pipeline throughput can nonetheless provide useful information. Assume that the vehicles have no knowledge of their deceleration capability $a_A^{\min}$ but know that it is bounded in an interval $[\underline{a}, \overline{a}]$. Then, to maintain safety, they need to follow each other at a spacing

$$s^{\max} = \max_{a_A^{\min}, a_B^{\min} \in [\underline{a}, \overline{a}]} s(x_1^0, 0, 0)$$

(recall that s is parameterized by the model constants $a_A^{\min}, a_B^{\min}, j_A^{\min}$). The resulting throughput for $\underline{a} = -9.3$ ms^{-2}, $\overline{a} = -4.9$ ms^2, $j_A^{\min} = -25$, and $L = 5$ m is shown in Fig. 8 (case $N = 1$). With minimal changes the estimate can be refined to allow for sensing and actuation delays, safety margins, and different information structures (e.g., vehicles estimating $a_A^{\min}$ on line) [42]. The results of

our calculations can also directly be used in more elaborate throughput estimates that take into account lateral flow (see [44]).

IV. Platoons of Vehicles

A. The Platooning Concept

Can we somehow exceed the maximum throughput of the autonomous AHS? The calculations of Section III indicate that this cannot be done unless either the vehicles are allowed to collide or some centralization is introduced (more global information is made available to them). We now try to derive conditions for a safe AHS on which collisions are possible; it will soon become apparent that some degree of centralization is needed in this case to guarantee that all collisions are safe.

Consider again the pair $A - B$ and assume that initially $x_4^0 = 0$. A crucial observation [45] (which can also be inferred from the calculations in this paper) is that in the presence of differences in deceleration capability and/or sensing and actuation delays, collisions will be at low relative velocities if either the vehicles are far enough from one another (in which case they can come to a stop before colliding) or they are close enough (in which case a collision happens almost instantaneously and therefore the relative velocity is low). This observation gives rise to the *platooning concept*, where it is envisioned that vehicles will travel in platoons, following each other very closely with spacings of the order of 2–4 m. Platoons are separated by large spacings to avoid interplatoon collisions. In addition to safety, the large interplatoon spacing helps attenuate disturbances as they propagate down a string of platoons.

In [46] a control hierarchy was proposed to organize traffic in platoons. The hierarchy involves controllers both on the vehicles and on the roadside. The goal of the roadside controllers is to maximize the flow of the highway system by appropriately routing traffic. A number of designs have been proposed for the roadside controller, making use of traffic flow models [47], [48], the concept of highway work [49], and the concept of highway space-time [50].

The hierarchy of [46] calls for an on-board vehicle controller organized in two layers: the *coordination layer* and the *regulation layer*. It is envisioned that the regulation layer will consist of primarily continuous controllers for maintaining safe spacing between the vehicles, tracking desired velocities, etc. The coordination layer on the other hand will be primarily discrete and will be responsible for the intervehicle communication necessary to ensure safety. Our goal in the remainder of this paper is to produce regulation layer controllers and specifications for the coordination layer so that the hybrid on-board controller is guaranteed to be safe. We then use our results to obtain an upper bound on the throughput that can be achieved in safety on a platoon-based AHS.

B. Discrete Aspects

To motivate the design process we first describe informally the operation of a vehicle on an AHS that supports platooning.

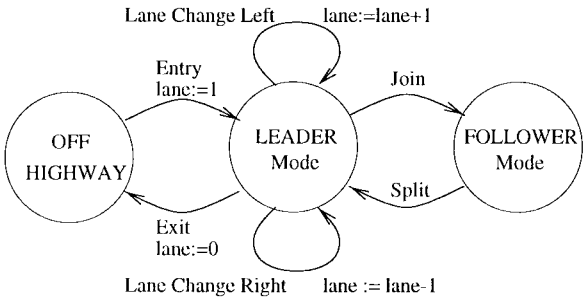

Fig. 5. Automated vehicle operation.

The vehicle can find itself in one of two discrete modes: the *leader mode* and the *follower mode*.[3] The state of the vehicle is also indexed by its discrete lane. Assume that the automated lanes are assigned numbers, increasing from right to left, with one representing the right-most lane, and let lane number zero indicate that the vehicle is off the AHS. The discrete state of an automated vehicle can be summarized by a pair (*mode, lane*) with *mode* $\in \{$leader, follower$\}$ and *lane* $= 0, 1, \cdots$. This is a somewhat coarse description of the discrete state (for example we do not keep track of the exact position of a vehicle in the platoon), but it will suffice for our purposes.

The evolution of the discrete state of an automated vehicle can be represented informally by the automaton of Fig. 5. The figure suggests the tasks that the regulation layer controller will be called upon to perform: tracking the desired spacing for the two modes and executing the transitions. The transitions between the discrete states are achieved by means of five maneuvers: *join*, *split*, *lane change*, *entry*, and *exit*. The join maneuver is used to join two platoons in a single lane to form a bigger platoon. The trailing platoon accelerates to catch up with the leading platoon. The split maneuver is used to break a platoon into two smaller platoons. The trailing platoon decelerates to safe interplatoon distance. The lane change maneuver is used to move a vehicle from one lane to another. Finally, the entry and exit maneuvers allow the vehicle to get on/off the automated highway.

In [51] communication protocols were designed to implement the discrete aspects of these maneuvers. The protocols are in the form of *finite state machines*, a special class of hybrid automata with no continuous state. Some simplifying assumptions were made; for example, it was assumed that a platoon can only be engaged in one maneuver at a time and that only single vehicles (*free agents*) can change lanes. The performance of the proposed protocols was verified using COSPAN [52], a finite-state machine model checker. The verification concerned only discrete aspects of the protocol operation, such as absence of deadlocks. Here we establish conditions on the protocol transitions that allow us to extend the verification to continuous aspects of the system operation, in particular safety with respect to collisions.

[3]The ensuing discussion will reveal that besides the obvious difference in desired spacing, the two modes also differ in the information needed for control.

C. The Follower Mode

Some of the issues that arise in the regulation layer design are not addressed here. Besides the limitations of scope introduced in Assumption 1, we will not address the design of control laws for the follower mode. This problem is particularly challenging and has received considerable attention in recent years [25], [38]. The most obvious difficulty is regulating to the constant follower spacing at all speeds, without running into the preceding vehicle. In [25] it was shown that to achieve this objective a vehicle needs information about the acceleration of the preceding vehicle, in addition to the spacing and relative velocity information available through the sensors. It is envisioned that this information will be communicated among adjacent vehicles in a platoon using a point-to-point infrared link. Another problem is that small disturbances (e.g., a transient change in the velocity of the platoon leader) may get amplified as they propagate from one follower to the next, possibly resulting in collisions down the string. In [25] it was shown that to avoid this *slinky effect*, followers also need information about the state of the platoon leader in addition to information about the vehicle immediately ahead of them. Because the delays of point-to-point communication increase with the size of the platoon, this information has to be broadcast by the leader (using a radio link for example). Finally, in [25] it was shown that the follower control law will result in amplification of the control effort exerted by followers further back in the platoon. This imposes restrictions on the operation of the leader. If, for example, the braking force is amplified by a factor of $\gamma \geq 1$, then, in a platoon of identical vehicles with deceleration capability $a^{\min}$, the deceleration of the lead vehicle should not exceed $a^{\min}/\gamma$ in order to avoid actuator saturation and the possibility of collisions within the platoon.

Here we circumvent these complications by assuming that a follower control law u^F, a safe set of initial conditions in the follower state space V^F, and a time T^F (possibly dependent on the size of the platoon) have been determined, such that we have the following assumption.

Assumption 4 (Platoon String Stability Under Collisions): Assume that all followers apply u^F, that at time t_i the state of all but one of the followers is in V^F, and that a safe collision occurs between the exceptional follower and the vehicle ahead of it. For any $t_f \geq t_i + T^F$, if no interplatoon collisions occur in the interval $[t_i, t_f]$, then at time t_f the state of all followers is in V^F, no intraplatoon collisions occur in $(t_i + T^F, t_f]$, while in $[t_i, t_i + T^F]$ all intraplatoon collisions are safe and the first and last vehicles experience at most one collision.

The assumption is intended to capture the situation of Fig. 6. Two platoons moving at slightly different speeds collide and become one platoon. If the intraplatoon spacing is small and Assumption 3 is satisfied, after a finite number of intraplatoon collisions the vehicle velocities will still be close to the initial velocities, but arranged in decreasing order. The work of [25] and [53] and intuition from basic collision dynamics suggest that this is not an unrealistic assumption. We are currently working on casting the follower control problem in the design framework discussed here and proving Assumption 4 as a theorem.

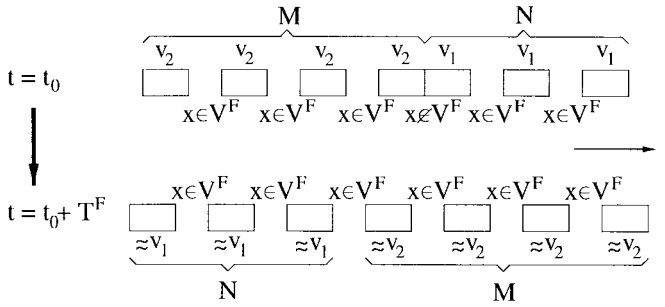

Fig. 6. A safe platoon collision ($v_2 \in [v_1, v_1 + v_a]$).

V. Platooning in Safety

A. Regulation Layer Design

To implement an AHS that supports platooning, continuous control laws are needed for the leader and follower modes and for the join, split, lane change, entry, and exit maneuvers. The design of the follower law was discussed in Section IV-C. Here we describe the design of the remaining controllers. We restrict our attention to safety; passenger comfort and efficiency are assumed to be treated as in Section III-B. The controllers for the lead mode and the various maneuvers differ from one another in the cost function used to encode safety and in the disturbances that they have to guard against. The reason for these differences will become apparent in the next section, when the coordination protocols are discussed. The controllers for the leader mode and the lane change, entry, and exit maneuvers will be required to prevent collisions altogether; therefore, the cost function used in their design will be J_1. The join and split controllers on the other hand will allow safe collisions; therefore, the cost function used in their design will be J_1'. The disturbances experienced by each controller will differ in the severity of $\mathrm{Coll_B}$ and $\mathrm{Coll_C}$. The coordination protocols will be such that a vehicle (A) in the leader mode may face $d \in \mathcal{D}_{v_a v_a}$ (one safe collision between B and D and one safe collision between C and A), a vehicle executing a join or a split may face $d \in \mathcal{D}_{00}$ (no collision by either B or C), while during a lane change, entry, or exit vehicles may face either $d \in \mathcal{D}_{00}$ or $d \in \mathcal{D}_{v_a 0}$.

1) Leader Law: We treat the design of the controller for the lead mode as a game between u and $d \in \mathcal{D}_{v_a v_a}$ over cost function J_1. Consider the candidate saddle solution (u_1^*, d_1^*) where $d_1^* = (\ddot{x}_{B1}^*, T_{B1}^*, \delta v_{B1}^*, T_{C1}^*, \delta v_{C1}^*)$, u_1^*, and $\ddot{x}_{B1}^*$ are given by (7) and $T_{B1}^* = 0$, $\delta v_{B1}^* = \min\{v_a, x_4^0 + x_1^0\}$, $T_{C1}^* = 0$, and $\delta v_{C1}^* = v_a$. Clearly $(u_1^*, d_1^*) \in \mathcal{U} \times \mathcal{D}_{v_a v_a}$. Let $(q^*(t), x^*(t))$ denote the state at time t under the inputs (u_1^*, d_1^*) starting at (q^0, x^0).

Proposition 3 (Leader Saddle Solution): If $x^0 \in X_C$, then $x^*(t) \in X_C$ for all $t \geq 0$. (u_1^*, d_1^*) is a global saddle solution for cost $J_1(q^0, x^0, u, d)$.

As for the autonomous vehicle case, the saddle solution allows us to calculate the set of safe states and classify the

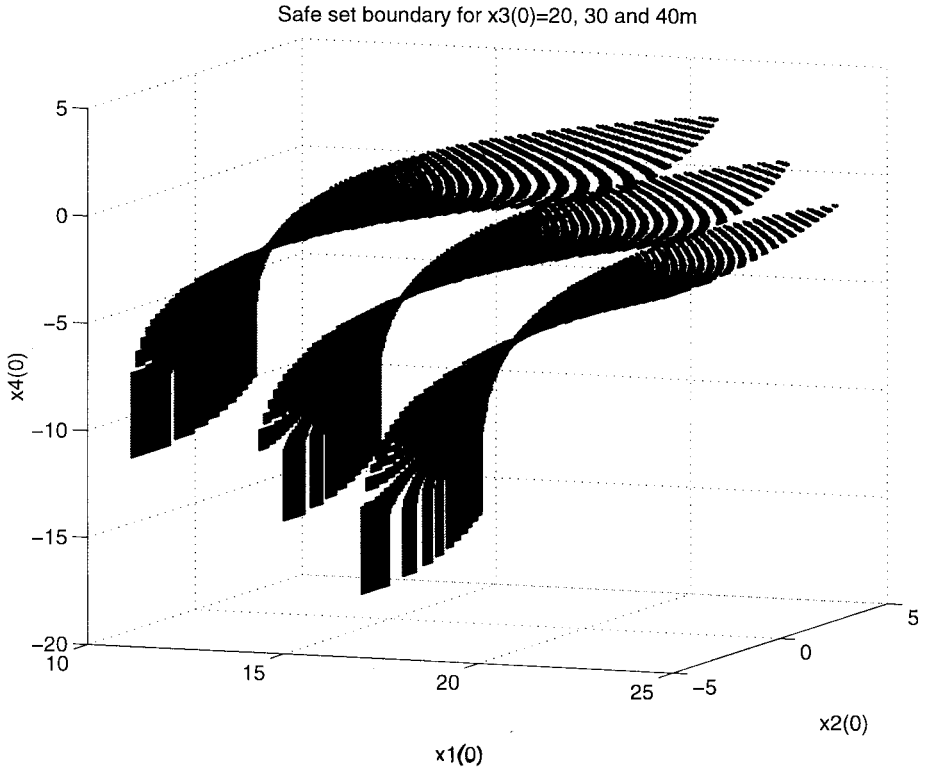

Fig. 7. Level sets of s^L for $q^0 = q_1$ and $x_3^0 = 20, 30,$ and 40 m.

safe controls. Define

$$V^L \triangleq \{(q,x) \in X \mid J_1(q,x,u_1^*,d_1^*) \leq C_1\} \subset X \quad (12)$$

$$\mathcal{U}^L(q,x) \triangleq \begin{Bmatrix} u_1^*(q,x), & \text{if } (q,x) \in X \backslash \text{int}(V^L) \\ [j_A^{\min}, j_A^{\max}], & \text{if } (q,x) \in \text{int}(V^L) \wedge q \neq q_3 \\ [0, j_A^{\max}], & \text{if } (q,x) \in \text{int}(V^L) \wedge q = q_3 \end{Bmatrix} \subset \mathcal{U}. \quad (13)$$

As before, for $q^0 = q_1$, ∂V^L can be encoded by a function $s^L : R^3 \to R$. Three level sets of this function are shown in Fig. 7.

Proposition 4: V^L is monotone with respect to x_3, in the sense that if $(q, [x_1 \ x_2 \ x_3 \ x_4]^T) \in V^L$ then $(q, [x_1 \ x_2 \ x_3' \ x_4]^T) \in V^L$ for all $x_3' \geq x_3$.

Proof: The proof follows by the fact that x_3^* is linear in x_3^0. □

Let the controller for the leader mode u^L be any feedback controller satisfying the $u^L(q,x) \in \mathcal{U}(q,x)$.

Lemma 3 (Leader Safety): If $(q^0, x^0) \in V^L$, $d \in \mathcal{D}_{v_a v_a}$, and for all $t \geq 0$, $u(t) = u^L(q(t), x(t))$, then vehicles A and B will not collide.

Proof: The proof follows by Proposition 3 and the properties of the saddle solution. □

Lemma 3 requires that $d \in \mathcal{D}_{v_a v_a}$, therefore at most one Coll_B and at most one Coll_C events are allowed. This requirement is imposed because these events can push the state outside V^L. If more events of the same kind occur while $x \notin V^L$, a collision may be possible. The requirement can be relaxed to allow more Coll_B and/or Coll_C events once x returns to V^L.

Lemma 4: There exists a finite $T^L > 0$ such that if $(q(T), x(T)) \notin V^L$ for some $T \geq 0$, no Coll_B or Coll_C events occur in $[T, T + T^L]$, and for all $t \in [T, T + T^L]$, $u(t) = u^L(q(t), x(t))$, then either $(q(T + T^L), x(T + T^L)) \in V^L$ or $q(T + T^L) = q_3$.

Lemma 3 provides safety guarantees for the leader mode, while Lemma 4 provides "liveness" guarantees. If the disturbance is such that any two Coll_B or Coll_C events are separated by T^L, then a vehicle under u^L will not collide with the vehicle ahead of it, and it will either stop or its state will return to V^L infinitely often.

2) Join and Split Laws: We treat the design of the controllers for the join and split maneuvers as a game between u and $d \in \mathcal{D}_{00}$ over cost function J_1'.

Proposition 5 (Join/Split Saddle Solution): At any time the saddle inputs $u_{1'}^* \in \mathcal{U}$ and $\ddot{x}_{B1'}^* \in \mathcal{D}_{00}$ take on either the minimum or the maximum value allowed by the constraints. Each trajectory may involve at most one switching from the maximum to the minimum and at the time of impact both $u_{1'}^*$ and $\ddot{x}_{B1'}^*$ are at their minimum values. $u_{1'}^*$ can take on its maximum value only when $\ddot{x}_{B1'}^*$ does.

The proof of Proposition 5 (Appendix B) suggests that if vehicle B can decelerate harder than vehicle A and a collision takes place before vehicle B stops, then the worst disturbance vehicle B can generate may involve hard acceleration (in an attempt to increase the distance to vehicle A) followed by hard braking. In extreme cases the best response for vehicle A may also involve hard acceleration (in an attempt to keep up with vehicle B) followed by hard braking. These predictions were verified in [54] by numerical experiments.

Unlike the leader and autonomous cases, the calculation of the safe sets and controls has to be done numerically in this case, as it is difficult to express the saddle solution in feedback form and solve the corresponding equations.[4] From the point of view of safety, the join and split controllers can be treated very similarly. Clearly the rest of the controller design will have to be different, as the objectives of the two maneuvers differ. The design can be completed using the techniques outlined in Section III or by using other controllers proposed for this purpose in the literature [27], [28]. As before, define

$$V^J \triangleq V^S \triangleq \{(q,x) \in X \mid J_1'(q,x,u_{1'}^*, d_{1'}^*) \leq C_1'\} \subset X \quad (14)$$

$$\mathcal{U}^J(q,x) \triangleq \mathcal{U}^S(q,x) = \begin{Bmatrix} u_{1'}^*(q,x), & \text{if } (q,x) \in X \backslash \text{int}(V^J) \\ [j_A^{\min}, j_A^{\max}], & \text{if } x \in \text{int}(V^J) \wedge q \neq q_3 \\ [0, j_A^{\max}], & \text{if } x \in \text{int}(V^J) \wedge q = q_3 \end{Bmatrix} \subset \mathcal{U}. \quad (15)$$

Let the controllers for the join and split maneuvers, u^J and u^S, be feedback controllers satisfying $u^J(q,x) \in \mathcal{U}^J(q,x)$ and $u^S(q,x) \in \mathcal{U}^S(q,x)$.

Lemma 5 (Join/Split Safety): If $(q^0,x^0) \in V^J$ ($(q^0,x^0) \in V^S$), $d \in \mathcal{D}_{00}$ and for all $t' \in [0,t]$ $u(t') = u^J(q(t'),x(t'))$ ($u(t') = u^S(q(t'),x(t'))$), then $(q(t),x(t)) \in V^J$ ($(q(t),x(t)) \in V^S$) and vehicles A and B will either not collide or will experience a safe collision.

Proof: By the properties of the saddle solution and the definitions of safe states and controls. □

3) Lane Change, Entry, and Exit Laws: We assume the lane change policy of [51], where a vehicle is a free agent both before and after the lane change. Assume free agent A wishes to move to the lane where platoons E and F are moving. The lane change can be decomposed in two phases. In the first phase A and/or E decelerate so that A is aligned with an appropriate gap in the target lane. In the second phase the actual move of A from the origin to the target lane takes place. For the purpose of safety, entry and exit from the automated highway will be treated as a simplified version of a lane change. In particular, entry will be modeled as a lane change from the on-ramp to an automated lane, with vehicles B, C, and D missing. Exit will be modeled as a lane change from an automated lane to an off ramp, with vehicles E, F, G, and H missing.

Specialized controllers are needed for vehicles A and E (u^{CA} and u^{CE}, respectively) for the lane change maneuver. During the alignment phase u^{CA} makes use of x_{AB} and x_{AF} to keep A safe with respect to B while bringing it sufficiently far away from F for the lane change to take place. During the move phase u^{CA} ensures that A remains safe with respect to both B and F. Likewise, during the alignment phase u^{CE} makes use of x_{EA} and x_{EF} to keep E safe from F while creating a gap for A. During the move phase, u^{CE} ensures that E remains safe with respect to both A and F. The coordination protocols will guarantee that while a vehicle is applying u^{CA} it will not be hit from behind (no Coll_C disturbance) and that while it is applying u^{CE} it will not be hit from behind, and the lane changing vehicle A will not experience a collision (no Coll_B or Coll_C disturbance). The protocols will also require that a vehicle should not collide with the vehicle ahead of it while applying u^{CA} or u^{CE}. We therefore treat the design of u^{CA} as a game between $u \in \mathcal{U}$ and $d \in \mathcal{D}_{v_a 0}$ and the design of u^{CE} as a game between $u \in \mathcal{U}$ and $d \in \mathcal{D}_{00}$, both over cost function J_1. By the proof of Proposition 3, it is immediately apparent that the saddle solution in the first case is the same as (u_1^*, d_1^*) with $\delta v_{C1}^* = 0$, and in the second case it is the same as (u_1^*, d_1^*) with $\delta v_{B1}^* = \delta v_{C1}^* = 0$. Let $(u_1^*, d_{1_{v_a 0}}^*)$ and $(u_1^*, d_{1_{00}}^*)$ denote these saddle solutions and define

$$V^{CA} \triangleq \left\{(q,x) \in X \,\middle|\, J_1\left(q,x,u_1^*, d_{1_{v_a 0}}^*\right) \leq C_1\right\}$$

and

$$V^{CE} \triangleq \left\{(q,x) \in X \,\middle|\, J_1\left(q,x,u_1^*, d_{1_{00}}^*\right) \leq C_1\right\}.$$

Consider the two classes of controllers shown at the bottom of the page. Note that u_1^* depends only on x_2 and q; therefore, $u_1^*(q_A, x_{AB}) = u_1^*(q_A, x_{AE})$ and $u_1^*(q_E, x_{EA}) = u_1^*(q_E, x_{EF})$. Let u^{CA} and u^{CE} be feedback controllers satisfying $u^{CA}(q, x_{AB}, x_{AF}) \in \mathcal{U}^{CA}(q, x_{AB}, x_{AF})$ and $u^{CE}(q, x_{EA}, x_{EF}) \in \mathcal{U}^{CE}(q, x_{EA}, x_{EF})$.

Lemma 6 (Lane Change Safety): If $(q_A^0, x_{AB}^0) \in V^{CA}$ ($(q_E^0, x_{EF}^0) \in V^{CA}$), $d_{AB} \in \mathcal{D}_{v_a 0}$ ($d_{EF} \in \mathcal{D}_{v_a 0}$) and for all $t \geq 0$, $u_A(t) = u^{CA}(q_A(t), x_{AB}(t), x_{AF}(t))$ ($u_E(t) = u^{CE}(q_E(t), x_{EA}(t), x_{EF}(t))$), then vehicles A and B (E and F) will not collide. If in addition $(q_A^0, x_{AF}^0) \in V^{CA}$ ($(q_E^0, x_{EA}^0) \in V^{CE}$), $d_{AF} \in \mathcal{D}_{v_a 0}$ ($d_{EA} \in \mathcal{D}_{00}$), and vehicle A changes lane, then vehicles A and F (E and A) will not collide.

Proof: Assume $(q_A^0, x_{AB}^0) \in V^{CA}$. Note that u_1^* is applied whenever $(q_A, x_{AB}) \notin \text{int}(V^{CA})$. The definition of V^{CA} and the properties of the saddle solution imply that vehicles A and B will not collide. The proof for x_{EF}, x_{AF}, and x_{EA} is similar. □

The first sentence of Lemma 6 provides conditions under which the alignment phase of the lane change is safe, while

[4] Analytical calculation is possible in the join/split case, under slightly different assumptions [28].

$$\mathcal{U}^{CA}(q_A, x_{AB}, x_{AF}) \triangleq \begin{cases} u_1^*(q_A, x_{AB}), & \text{if } (q_A, x_{AB}) \in X \backslash \text{int}(V^{CA}) \vee (q_A, x_{AF}) \in X \backslash \text{int}(V^{CA}) \\ [j_A^{\min}, j_A^{\max}], & \text{if } (q_A, x_{AB}) \in \text{int}(V^{CA}) \wedge (q_A, x_{AF}) \in \text{int}(V^{CA}) \wedge q \neq q_3 \\ [0, j_A^{\max}], & \text{if } (q_A, x_{AB}) \in \text{int}(V^{CA}) \wedge (q_A, x_{AF}) \in \text{int}(V^{CA}) \wedge q = q_3 \end{cases}$$

$$\mathcal{U}^{CE}(q_E, x_{EA}, x_{EF}) \triangleq \begin{cases} u_1^*(q_E, x_{EA}), & \text{if } (q_E, x_{EA}) \in X \backslash \text{int}(V^{CE}) \vee (q_E, x_{EF}) \in X \backslash \text{int}(V^{CA}) \\ [j_A^{\min}, j_A^{\max}], & \text{if } (q_E, x_{EA}) \in \text{int}(V^{CE}) \wedge (q_E, x_{EF}) \in \text{int}(V^{CA}) \wedge q \neq q_3 \\ [0, j_A^{\max}], & \text{if } (q_E, x_{EA}) \in \text{int}(V^{CE}) \wedge (q_E, x_{EF}) \in \text{int}(V^{CA}) \wedge q = q_3 \end{cases}$$

the second sentence provides conditions under which the move phase is safe. During the alignment phase A need not worry about colliding with F, and E need not worry about colliding with A. Therefore, u^{CA} and u^{CE} can apply a milder control input instead of u_1^* when $x_{AF} \notin \text{int}(V^{CA})$ and when $x_{EA} \notin \text{int}(V^{CE})$, respectively. During the move phase, however, u_1^* is needed in both these cases to guarantee safety. The protocol design will be such that during the lane change vehicle C will be applying $u^L(q_C, x_{CA})$. Therefore, throughout the move phase A is safe with respect to both B and F (by Lemma 6), E is safe with respect to both the A and F (by Lemma 6), and C is safe with respect to both A and B (by Lemma 3 and Proposition 4). Hence, the lane change can be safely aborted at any stage. This is a very desirable property as it allows us to decouple the lanes. Proposition 4 also implies that occlusion of vehicles during a lane change does not affect safety. For example, vehicle C can remain safe with respect to vehicle B even though it has sensory information about it only after vehicle A has effectively left the lane.

Note that once (q_E, x_{EA}) enters V^{CE}, it remains there throughout the lane change. However, (q_A, x_{AB}), (q_A, x_{AF}), and (q_E, x_{EF}) can leave the set V^{CA} due to collisions of vehicles B or F. A "liveness" guarantee, similar to Lemma 4, can be also be given for the lane change.

Lemma 7: If $(q_A(T), x_{AB}(T)) \notin V^{CA}$ for some $T \geq 0$, no Coll_B or Coll_C events occur in $[T, T+T^L]$, and for all $t \in [T, T+T^L]$, $u_A(t) = u^{CA}(q_A(t), x_{AB}(t), x_{AF}(t))$, then either $(q_A(T+T^L), x_{AB}(T+T^L)) \in V^{CA}$ or $q_A(T+T^L) = q_3$. A similar claim holds for $(q_A, x_{AF}) \notin V^{CA}$ and for $(q_E, x_{EB}) \notin V^{CA}$ under $u_E(t) = u^{CE}(q_E(t), x_{EA}(t), x_{EF}(t))$.

Proof: The proof is virtually identical to the proof of Lemma 4. □

B. Coordination Layer Specifications

We describe a set of requirements imposed on the coordination layer so that its interaction with the regulation layer control laws is guaranteed to be safe. The discussion in this section is informal and provides only the information necessary for stating and proving the platooning safety theorem. The discrete steps we assume are consistent with the protocols specified in [51], while the discrete/continuous interface aspects are consistent with the design of [55]. The formal analysis carried out in these two papers provides performance guarantees for the discrete dynamics of the system.

All followers are assumed to apply u^F and, unless otherwise stated, the leaders are assumed to apply u^L. Assumption 4 allows us to ignore the follower dynamics for the most part. By abuse of notation, let A and B (Fig. 2) represent platoons of M and N vehicles, respectively, and let x_{AB} denote the distance between the leader of A and the last follower of B. Following [51] we assume all communication and coordination is handled by the platoon leader.

First consider what happens when A joins B. The maneuver is initiated by A, whose leader checks that the platoon is not already engaged in another maneuver and that $x_{AB} \in V^J$. If this is the case, the leader declares the platoon engaged in a join maneuver and sends a join request to the leader of B. Upon receiving the maneuver request, the leader of platoon B checks if it is already engaged in another maneuver. If this is the case, the request is declined. A message is sent to the leader of A who aborts the join (is no longer engaged in the maneuver and remains a leader). Otherwise, the leader of B switches to the set V^L, corresponding to the deceleration capability it would have if A and B were to join successfully (recall that the deceleration of a leader is limited by the deceleration capabilities of all the followers). Even if initially $x_{BD} \notin V^L$ for the new V^L, the design of u^L is such that after a finite amount of time T^L, either B stops or $x_{BD} \in V^L$. In the former case the join request is declined, while in the latter the maneuver is accepted by B. An appropriate message is sent to A which accordingly either aborts the maneuver or checks if still $x_{AB} \in V^J$. If this is the case, the leader of A applies u^J until either $x_{AB} \in V^F$ or the leader of A collides with the last vehicle of B. In either case, intraplatoon coordination is established, the leader of A starts applying u^F, and A and B begin operating as a single $N+M$ vehicle platoon. If there was no collision the new platoon declares the maneuver complete and is no longer engaged in it. Otherwise, the new platoon waits until $x \in V^F$ for all followers (guaranteed to be the case by at most T^F time units) and then a further T^L in time units before declaring the maneuver complete. The extra time is added to allow x_{CA} to return to V^L, which it may have left as a result of the $A-B$ collision. Note that there is no guarantee that the join will successfully terminate in a finite amount of time (though this is likely to be the case in most realistic situations).

Now assume A represents the last M vehicles and B the first N vehicles of a platoon of size $N+M$, and consider what happens if A wants to split from B. The leader checks that the platoon is not already engaged in another maneuver, that $x_{AB} \in V^S$, and that $x_{BD} \in V^L$. If this is the case it declares the platoon as engaged in a split maneuver and the "leader" of A applies u^S. If A collides with B, the maneuver is aborted following the same process as a join termination. If at some point $x_{AB} \in V^L$ or if A stops the maneuver is completed, the leader of A starts applying u^L and the leader of B switches to the set V^L corresponding to the new platoon size. Both A and B declare themselves as no longer engaged in the split maneuver.

Finally, consider a free agent A wishing to move to the lane where platoons E and F are moving. A first ensures that it is not already engaged in another maneuver and that $x_{AB} \in V^{CA}$. If this is the case it sends a lane change request to vehicles E, F, and G. If platoon E is engaged in another maneuver or if $x_{EF} \notin V^{CA}$, the lane change is aborted. Otherwise, G commits to not change lanes in between E and F and a message is sent to A to proceed. A and E apply u^{CA} and u^{CE} until either A stops or $x_{AB} \in V^{CA}$, $x_{AF} \in V^{CA}$, $x_{EA} \in V^{CE}$, and $x_{EF} \in V^{CA}$ (one of the two is bound to happen by T^L time units). If A stops, the maneuver is aborted. Otherwise, A starts moving to the new lane, while it applies u^{CA} and E applies u^{CE}. When A reaches the target lane the lane change is declared complete and A and E revert to u^L.

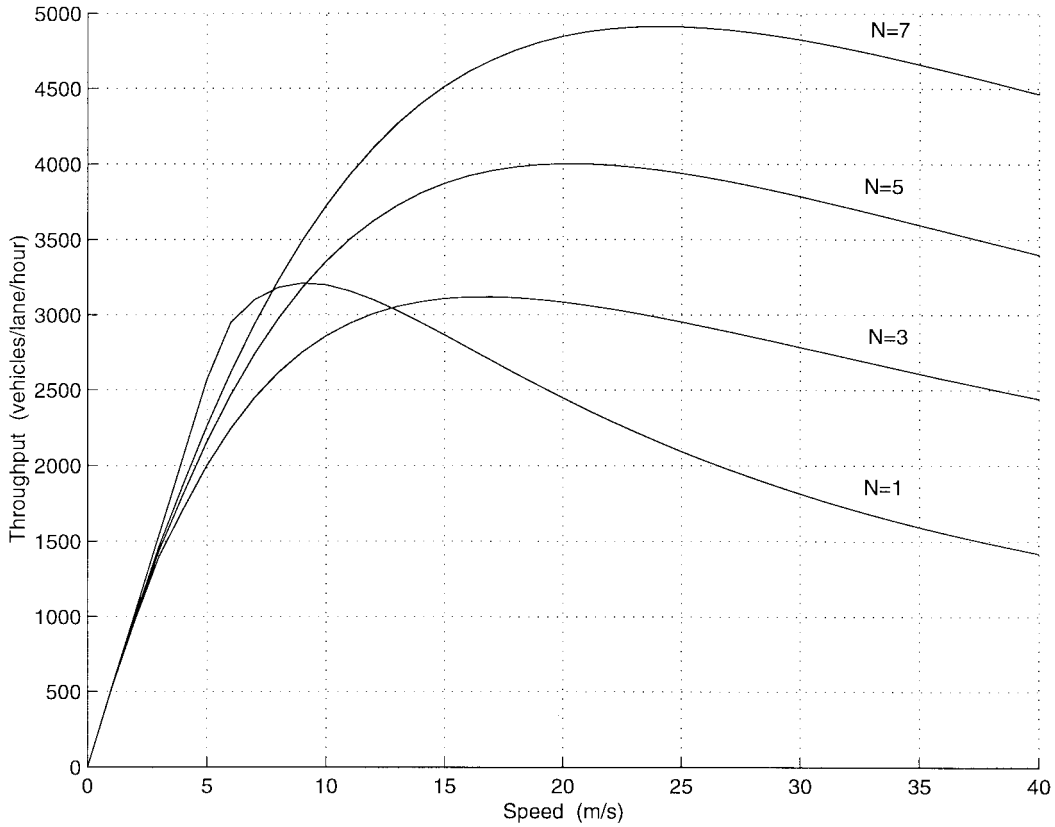

Fig. 8. Pipeline throughput for platoons of size N ($N = 1$ autonomous vehicles).

C. Platooning Safety

Theorem 2 (Platoon Safety): An AHS populated by a finite number of vehicles following the proposed coordination and regulation requirements will be safe.

Proof: The proof involves the lemmas of Section V-A, the coordination requirements of Section V-B, and an induction argument on the number of vehicles. Consider again the vehicles of Fig. 2.[5] First consider A joining or splitting from B. In this case the coordination requirements ensure that u_D and u_C belong to $\{u^L, u^F, u^{CA}, u^{CE}\}$. Therefore, Assumption 4 and Lemmas 3 and 6 ensure that vehicle A will not experience any collision disturbances. According to Lemma 5, platoon A may experience a safe collision with platoon B. After the collision, all vehicles in platoon A start applying u^F. By Assumption 4, all followers return to V^F after a finite number of safe collisions and the first vehicle of platoon B and the last vehicle of platoon A experiencing at most one collision.

Now consider A applying the leader control law. A join/split maneuver between vehicles B and D may produce one collision disturbance for vehicle A ($\mathrm{Coll_B}$). The coordination specification of Section V-B also permits C to join/split from A. This maneuver can also potentially result in safe collisions. Still, Lemma 3 guarantees that vehicle A will not collide with vehicle B. When $\mathrm{Coll_B}$ and/or $\mathrm{Coll_C}$ occur, the state of vehicle A may leave the set V^L. According to Lemma 4, T^L will be the maximum time it takes for the state to return to V^L in this case, i.e., a lower bound on the time separation between two successive collisions of the same kind that can be tolerated by u^L. The coordination requirement on the join and split maneuvers guarantees that if the maneuver results in collision, a new join or split will not be initiated before all intraplatoon collisions subside and an extra interval T^L elapses. This timing requirement on join/split guarantees that the assumptions of Lemma 3 are satisfied, implying that vehicle A will not collide with vehicle B under u^L. A similar argument indicates that a vehicle executing a lane change maneuver (and consequently an entry or an exit) will not collide with the vehicle ahead of it. □

Note that the limitation of "one maneuver per vehicle at a time" imposed by the coordination layer [51] ensures that the front vehicle of a joining or splitting pair will be in the leader mode, while the trailing vehicle will either be followed by a vehicle in the leader mode or by one executing a lane change. The additional timing constraint ensures that the platoons have enough time to recover from the collisions before a new join or split is initiated. Overall, collisions due to joining and splitting are guaranteed not to propagate beyond the platoons engaged in the maneuver.

Corollary 1 (Individual Maneuver Safety): Any collisions that take place during the join or split maneuvers or in the follower mode will be safe. A vehicle will never collide while executing a lane change (and hence an entry or an exit). A vehicle in the leader mode will never collide with the vehicle ahead of it.

[5] By Assumption 5 $A, \cdots, H$ may in fact represent entire platoons.

TABLE I

$En_{q_1q_1} = \neg Coll_B \wedge \neg Coll_C \wedge x_3 = 0 \wedge x_4 < 0 \wedge x_1 \neq 0$ $Res_{q_1q_1} = [x_1 + x_4 \;\; x_2 \;\; x_3 \;\; -x_4]^T$ $Inv_{q_1} = \neg Coll_B \wedge \neg Coll_C \wedge x_1 \neq 0 \wedge x_3 \neq 0$	$En_{q_1q_2} = \neg Coll_B \wedge \neg Coll_C \wedge x_3 = 0 \wedge x_4 = 0 \wedge x_1 \neq 0$ $Res_{q_1q_2} = [x_1 \;\; \ddot{x}_B \;\; x_3 \;\; x_4]^T$	$En_{q_1q_3} = \neg Coll_B \wedge \neg Coll_C \wedge x_1 = 0$ $Res_{q_1q_3} = [x_1 \;\; 0 \;\; x_3 \;\; x_4]^T$
$En_{q_2q_1} = \neg Coll_B \wedge \neg Coll_C \wedge (x_2 < \ddot{x}_B \vee x_1 = 0 \vee x_4 \neq 0)$ $Res_{q_2q_1} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T$	$Inv_{q_2} = \neg Coll_B \wedge \neg Coll_C \wedge x_1 \neq 0 \wedge x_2 \geq \ddot{x}_B \wedge x_4 = 0$	
$En_{q_3q_1} = \neg Coll_B \wedge \neg Coll_C \wedge x_1 \neq 0$ $Res_{q_3q_1} = [x_1 \;\; x_2 \;\; x_3 \;\; x_4]^T$		$Inv_{q_3} = \neg Coll_B \wedge \neg Coll_C \wedge x_1 = 0$

D. Platooning Throughput

In addition to the lane change, entry, and exit disturbances, the calculation of the throughput for an AHS that supports platoons is further complicated by the presence of disturbances due to the formation and dissipation of platoons. Pipeline throughput estimates can again provide an upper bound. Assume an AHS is populated entirely by platoons of size N moving at velocity x_1^0 at steady state. Let F denote the follower spacing. It is easy to see that the throughput of such an AHS in vehicles per lane per unit time is given by

$$Q = \frac{Nx_1^0}{s^L(x_1^0, 0, 0) + NL + (N-1)F}. \tag{16}$$

As for the autonomous vehicle case, assume that vehicles do not know their deceleration capability but know that it is bounded in a range $[\underline{a}, \overline{a}]$. Following [25], we can model the fact that the deceleration of the leader is limited by that of the followers by assuming that platoons follow each other at spacings of

$$s^{\max} = \max_{\gamma a_A^{\min}, a_B^{\min} \in [\underline{a}, \overline{a}]} s^L(x_1^0, 0, 0)$$

where $\gamma = 1,\ 1.05,\ 1.1,\ 1.15,\ 1.2$, for $N = 1, 2, 3, 4$, and $N \geq 5$, respectively. The resulting throughput for $F = 2m$ is shown in Fig. 8. Note that as platoon leaders have to be able to tolerate $d \in \mathcal{D}_{v_a v_a}$, whereas autonomous vehicles only have to deal with $d \in \mathcal{D}_{00}$, the platooning throughput can be smaller at low speeds.

VI. Concluding Remarks

We presented a methodology for the design of hybrid onboard controllers for automated vehicles. The methodology was based on techniques from game theory and optimal control. Assuming a particular policy for intervehicle information exchange, these techniques allow us to quantify the safe behaviors of the system. Moreover, the calculations suggest ways in which the intervehicle information exchange can be modified to improve the system performance. The safety conditions obtained by solving the gaming problems are clearly sufficient. They are also necessary in the sense of being "tight." The properties of the saddle solution imply that if the safety conditions are violated (e.g., an autonomous vehicle finds itself closer than s from the preceding vehicle), there exist disturbance trajectories (e.g., acceleration trajectories of the preceding vehicle) that will result in a collision. We demonstrated our approach by designing safe controllers for an AHS based on autonomous vehicles and an AHS that allows intervehicle cooperation and supports platooning.

One advantage of the proposed methodology is that it provides considerable insight into the system that can be useful in contexts other than controller design. We showed how limits on pipeline throughput can directly be inferred from our calculations. The calculations can also be used to infer technological requirements such as sensor ranges. A vehicle needs to be safe even if a stopped vehicle appears at the limit of its sensor range; therefore, the minimum sensor range needed for safe steady-state operation with speed x_1^0 is $s(x_1^0, 0, -x_1^0)$. Our results also indicate what pieces of information would improve performance. For example, it is easy to see that knowledge of the vehicles deceleration capability can greatly reduce the average following distance and hence increase throughput [42].

In our framework, the role of interagent coordination is to reduce the set of allowable disturbances. This has the effect of "biasing" the game in favor of the controller and hence increasing the range of conditions under which game winning controls exist and improving the overall system performance. The intervehicle coordination was used to limit both the discrete (collisions) and continuous (acceleration) disturbances to produce safe join and split controllers.

Our calculations deal with only a class of safety problems of importance for AHS. There are still a number of questions to be answered. For example, the safety of the follower operation

was taken for granted here, even in the presence of collisions. More work is needed to show Assumption 4 can indeed be satisfied and to relax it if possible. An even more challenging problem is relaxing the "normal operation" assumption to allow faults and adverse environmental conditions. This may require not only extending the architecture with fault detection capabilities [37] and new controllers [56] but also changing the methodology of the verification. Here, our objective was to show that no unsafe collisions are possible. In the presence of faults it is easy to construct scenarios where unsafe collisions are possible whatever the controller does [56]. A possible solution would be to introduce probabilistic analysis and require that the probability of unsafe collisions is small. Formulating such proofs will require considerable extension of our design methodology.

Probabilistic analysis can also be used to produce less conservative "normal mode" controllers. The design methodology presented in this paper assumes that the *worst disturbance* may be produced by neighboring vehicles at any time, thereby leading to somewhat conservative designs with smaller throughput. If we use parameters corresponding to manual driving in our models, the resulting controllers would yield less throughput than the one observed on existing highways. The fairly small number of collisions on the current highway system indicates that the probability of the worst disturbance arising must be small. A probabilistic analysis methodology that takes into account the process of disturbance generation can be used to further analyze the safety/capacity tradeoff in an AHS. Probabilistic safety analysis also has the potential to provide control designs for partially automated vehicles operating in mixed automated-manual traffic, which is perhaps the most challenging problem in this area.

APPENDIX A
DISCRETE DYNAMICS OF THE PLANT AUTOMATON

See Table I.

APPENDIX B
ADDITIONAL PROOFS

Proof of Proposition 1: Existence and uniqueness during continuous evolution is guaranteed by the linear dynamics in states q_1 and q_3 and the assumptions on f. Existence for the discrete evolution is guaranteed, as for each value of q the disjunction of the invariant and the guards of the outgoing transitions is true. Uniqueness for the discrete evolution is guaranteed, as for each value of q the pairwise conjunction of the guards of the outgoing transitions among themselves and with the invariant is false. □

Proof of Lemma 1: By assumption, no vehicles are moving backward in the beginning of the run. We show that this property is preserved by showing that if it holds in a given state it also holds for all states that can be reached from there either by discrete transitions or by continuous evolution; the claim follows by induction on the length of the run of the automaton.

Assume that no vehicles are moving backward in a given state. Consider first the states that can be reached from that state by discrete transitions. Transition $\mathrm{Coll_C}$ increases x_1, while transitions $\mathrm{Coll_B}$, $\mathrm{Stop_A}$, $\mathrm{Start_A}$, $\mathrm{Touch_A}$, and $\mathrm{Separate_A}$ leave it unaffected. Therefore, if $(x_1 \geq 0)$ is true before any of these transitions, it will also be true afterwards. After transition $\mathrm{Coll_A}$, x_1 is reset to $x_1 + x_4$. This is the value of $\dot{x}_B$ before $\mathrm{Coll_A}$, which satisfies $\dot{x}_B \geq 0$ by the induction hypothesis.

Now consider a continuous trajectory that at time t_1 satisfies $x_1(t_1) \geq 0$, and assume for the sake of contradiction that there exists $t_2 > t_1$ with $x_1(t_2) < 0$. By continuity this is possible only if there exists $t \in [t_1, t_2)$ such that $x_1(t) = 0$. At time t the discrete state transitions to q_3 and x_2 are reset to zero. By assumption, if the system evolves continuously from this state, then $u \geq 0$, and therefore x_1 cannot decrease any further as $\dot{x}_1 = x_2 = 0$ and $\ddot{x}_1 = u \geq 0$. This contradicts the hypothesis that there exists t_2 with $x_1(t_2) < 0$.

The claim that in this case $\mathrm{Coll_A}$ can never occur in states q_2 and q_3 follows (in fact the q_2 part is true even if the lemma assumptions are violated). The set $\mathcal{D}$ simply restricts $\dot{x}_B$ to be nonnegative. It therefore contains all disturbances allowed by the lemma assumptions and possibly more, as $\ddot{x}_B$ is only required to be piecewise continuous whereas in fact it is piecewise differentiable. □

Proof of Proposition 3: The first part is a straightforward calculation. For the second part we show that a unilateral change in strategy leaves the player who decided to change in worse condition. Let T_1 denote the time x_2 reaches $a_A^{\min}$ and T_2 the time $x_1 + x_4$ reaches zero under (u_1^*, d_1^*) and note that

$$x_4^*(t) = [0 \quad -t \quad 0 \quad 1]x^0 - \int_0^t (t-\tau)u_1^*(\tau)\,d\tau + \int_0^t \ddot{x}_{B1}^*(\tau)\,d\tau - (\delta v_{B1}^* + \delta v_{C1}^*).$$

First, fix $d = d_1^*$ and let u vary. Let $x(t)$ denote the state at time t under the inputs (u, d_1^*). Then

$$x_4(t) - x_4^*(t) = \int_0^t (t-\tau)(u_1^*(\tau) - u(\tau))\,d\tau.$$

We need to distinguish two cases.

1) $t \leq T_1$: The bounds on u imply that $u(\tau) \geq u_1^*(\tau)$, therefore $x_4(t) - x_4^*(t) \leq 0$.
2) $t \geq T_1$: Recall that $\ddot{x}_{B1}^*(t)$ is piecewise constant. Therefore $x_4(t)$ and $x_4^*(t)$ are piecewise differentiable, with derivatives $-x_2(t)$ and $-x_2^*(t)$, respectively. By definition of T_1, $x_2^*(t) = a_A^{\min} \leq x_2(t)$. Therefore, as $x_4(0) = x_4^*(0)$, $x_4^*(t) \geq x_4(t)$.

In either case $\dot{x}_3(t) = x_4(t) \leq x_4^*(t) = \dot{x}_3^*(t)$. Using the fact that $x_3^*(0) = x_3(0) = x_3^0$ and integrating leads to $J_1(x^0, u, d_1^*) \geq J_1(x^0, u_1^*, d_1^*)$.

Next, fix u_1^* and allow d to vary. Let $x(t)$ denote the state at time t under the inputs (u_1^*, d). Then

$$x_4(t) - x_4^*(t) = \int_0^t (\ddot{x}_B(\tau) - \ddot{x}_{B1}^*(\tau))\,d\tau - \big(\delta v_B 1_{T_B}(t) - \delta v_{B1}^*\big) - \big(\delta v_C 1_{T_C}(t) - \delta v_{C1}^*\big)$$

where $1_{T_B}()$ $(1_{T_C}())$ is the step function at time T_B (T_C). Note that $\delta v_{C1}^* = v_a$ and $\delta v_C \le v_a$ imply that $\delta v_C 1_{T_C}(t) - \delta v_{C1}^* \le 0$ for all $t \ge 0$ (as $v_a > 0$). If $\delta v_{B1}^* = v_a$, the same is true for the term $(\delta v_B 1_{T_B}(t) - \delta v_{B1}^*)$. If $\delta v_{B1}^* < v_a$, then $x_4^*(0) + x_1^*(0) = 0$ (recall that $T_{B1}^* = 0$), and therefore $x_4^*(t) + x_1^*(t) \equiv 0$ (once vehicle B stops it never starts moving again under d_1^*). But, $x_4(t) + x_1(t) \ge 0$ (as $x(t) \in X$) and $x_1(t) = x_1^*(t)$ (as $x_1(t)$ does not depend on δv_B). Therefore, if $\delta v_{B1}^* < v_a$, $x_4(t) - x_4^*(t) \ge 0$ for all $t \ge 0$.

For the integral term we need to distinguish two cases.

1) $t \le T_2$: The bounds on $\ddot{x}_B$ imply that $\ddot{x}_B(\tau) \ge \ddot{x}_{B1}^*(\tau)$, therefore $\int_0^t (\ddot{x}_B(\tau) - \ddot{x}_{B1}^*(\tau))\, d\tau \ge 0$.
2) $t \ge T_2$: By definition of T_2, $\int_0^t \ddot{x}_{B1}^*(\tau)\, d\tau = -(x_1^0 + x_4^0)$. The state constraints imply that $x_1(t) + x_4(t) \ge 0$ (vehicle B does not go backward), therefore $\int_0^t \ddot{x}_B(\tau)\, d\tau \ge -(x_1^0 + x_4^0)$. Subtracting, $\int_0^t (\ddot{x}_B(\tau) - \ddot{x}_{B1}^*(\tau))\, d\tau \ge 0$.

The stopping time for B under d will always be greater than the stopping time under d_1^*. In both cases, we are able to conclude that $\dot{x}_3(t) = x_4(t) \ge x_4^*(t) = \dot{x}_3^*(t)$. Integrating this inequality and using the fact that $x_3^*(0) = x_3(0) = x_3^0$ gives $J_1(x^0, u_1^*, d) \le J_1(x^0, u_1^*, d_1^*)$.

Overall $J_1(x^0, u_1^*, d) \le J_1^*(x^0) \le J_1(x^0, u, d_1^*)$ for all d and u. By definition, (u_1^*, d_1^*) is globally a saddle solution. □

Proof of Lemma 4: Assume, for the sake of contradiction, that (q, x) can exit V^L and never re-enter, without the vehicle stopping. Note that u_1^* is applied while $(q, x) \in X \backslash V^L$ (i.e., vehicle A decelerates as hard as possible). By Lemma 3 vehicles A and B will not collide. Therefore, if (q, x) remains outside V^L, vehicle A will eventually stop. This contradicts our original assumption. Let T_3 denote the time it takes a vehicle to stop under u_1^*. A simple calculation indicates that T_3 depends on $x_1^0, x_2^0, \delta v_C$ and the model constants. Taking

$$T^L = \max\{T_3 \,\big|\, x_1^0 \in [0, v_H], \; x_2^0 \in [a_A^{\min}, a_A^{\max}], \delta v_C \in [0, v_a]\} \quad (17)$$

provides a (finite) upper bound on the time it takes for the statement $((q, x) \in V^L) \vee (q = q_2)$ to become true. □

Proof of Proposition 5: We write J_1' as an integral cost function, assume that optimal controls exist, and proceed to classify them using the Maximum Principle [57]. At the time of impact t', the relative velocity will be negative, therefore

$$\begin{aligned} J_1'(x^0, u, \ddot{x}_B) &= -x_4(t') \\ &= -x_4^0 + \int_0^{t'} (x_2(t) - \ddot{x}_B(t))\, dt. \end{aligned}$$

Define the Lagrangian as $L(x, u, \ddot{x}_B) = x_2 - \ddot{x}_B$. For a given initial condition, the extrema of $J_1'(x^0, u, \ddot{x}_B)$ occur at the extrema of $\int_0^{t'} L(x(t), u(t), \ddot{x}_B(t))\, dt$. Optimization is subject to input constraints $u \in [j_A^{\min}, j_A^{\max}]$, $\ddot{x}_B \in [a_B^{\min}, a_B^{\max}]$ and state constraints $x_2 \in [a_A^{\min}, a_A^{\max}]$ and $x_4 + x_1 \ge 0$. Define functions $S_1(x) = a_A^{\min} - x_2$, $S_2(x) = x_2 - a_A^{\max}$, and $S_3(x) = -x_1 - x_4$. The state constraints are satisfied as long as $S_i(x) \le 0$ for $i = 1, 2, 3$. Following [57] we differentiate the constraint equations until the inputs appear

$$\begin{aligned} \dot{S}_1(x, u, \ddot{x}_B) &= -u, \qquad \dot{S}_2(x, u, \ddot{x}_B) = u \\ \dot{S}_3(x, u, \ddot{x}_B) &= -\ddot{x}_B. \end{aligned}$$

Let $\mu = [\mu_1 \;\; \mu_2 \;\; \mu_3]^T$ and form the Hamiltonian

$$\begin{aligned} &H(x, p, \mu, u, \ddot{x}_B) \\ &\quad = L(x, u, \ddot{x}_B) + p^T(Ax + Bu + D\ddot{x}_B) + \sum_{i=1}^{3} \mu_i \dot{S}_i \\ &\quad = (1 + p_1 - p_4)x_2 + p_3 x_4 + (p_2 - \mu_1 + \mu_2)u \\ &\qquad + (p_4 - 1 - \mu_3)\ddot{x}_B \end{aligned}$$

with $\mu_i = 0$ if $S_i(x) < 0$, $\mu_i > 0$, if $S_i(x) = 0$, and $(u, \ddot{x}_B)$ are chosen so that $\dot{S}_i(x, u, \ddot{x}_B) = 0$, if $S_i(x) = 0$. As the terms in u and $\ddot{x}_B$ decouple, we take the min–max of H to obtain a saddle solution $(u_{1'}^*, \ddot{x}_{B1'}^*)$. Let $H^*(x, p, \mu) = H(x, p, \mu, u_{1'}^*, \ddot{x}_{B1'}^*)$, where

$$u_{1'}^* = \begin{cases} j_A^{\min}, & \text{if } (p_2 > 0) \wedge (S_1(x) < 0) \wedge (S_2(x) < 0) \\ j_A^{\max}, & \text{if } (p_2 < 0) \wedge (S_1(x) < 0) \wedge (S_2(x) < 0) \\ 0, & \text{if } (p_2 = 0) \vee (S_1(x) = 0) \vee (S_2(x) = 0) \end{cases} \quad (18)$$

$$\ddot{x}_{B1'}^* = \begin{cases} a_B^{\max}, & \text{if } (p_4 - 1 > 0) \wedge (S_3(x) < 0) \\ a_B^{\min}, & \text{if } (p_4 - 1 < 0) \wedge (S_3(x) < 0) \\ 0, & \text{if } (p_4 - 1 = 0) \vee (S_3(x) = 0). \end{cases} \quad (19)$$

The Euler–Lagrange equation for p is

$$\dot{p} = -\left[\frac{\partial H^*(x, p, \mu)}{\partial x}\right]^T = \begin{bmatrix} 0 \\ p_4 - p_1 - 1 \\ 0 \\ -p_3 \end{bmatrix}$$

which, with the boundary condition $p(t') = [0 \;\; 0 \;\; c \;\; 0]^T$ for some $c \in R$ leads to $p_1(t) = 0$, $p_2(t) = -\frac{c}{2}(t' - t)^2 + (t' - t)$, $p_3(t) = c$, and $p_4(t) = c(t' - t)$ for $t \in [0, t']$.

The solution is not completely specified at this stage. There are two parameters (c and t') to be determined from the boundary conditions $x_3(t') = 0$ and $H^*(x(t'), p(t'), \mu(t')) = 0$. Here only we try to determine conditions under which the optimal controls take on the maximal value allowed by the constraints. For this purpose we use the condition $H^*(x(t'), p(t'), \mu(t')) = 0$. As $u_{1'}^* = 0$ if $\mu_1 > 0$ or $\mu_2 > 0$ and $\ddot{x}_{B1'}^* = 0$ if $\mu_3 > 0$, the condition simplifies to

$$(x_2(t') - \ddot{x}_{B1'}^*(t')) + c x_4(t') = 0. \quad (20)$$

As the case $x_4(t') = 0$ is covered by Proposition 3, we further restrict our attention to $x_4(t') < 0$. We distinguish two cases. If $x_2(t') \le \ddot{x}_{B1'}^*(t')$, then $c < 0$. Therefore, $p_2(t) \ge 0$ and $p_4(t) \le 0$, and the controls assume the minimum value allowed by the constraints for all $t \in [0, t']$.

If $x_2(t') > \ddot{x}_{B1'}^*(t')$, then $c > 0$. If $0 < ct' \le 1$, $p_2(t) \ge 0$, and $p_4(t) \le 1$; therefore, the controls assume the minimum value allowed by the constraints for all $t \in [0, t']$. If $1 < ct' \le 2$, then $p_2(t) > 0$ for all $t \in [0, t']$ while $p_4(t) > 1$ for $t \in [0, t' - 1/c)$ and $p_4(t) \le 1$ for $t \in [t' - 1/c, t']$. Therefore,

$\ddot{x}^*_{B1'}$ assumes first the maximum and then the minimum value allowed by the constraints, while $u^*_{1'}$ assumes the minimum value allowed by the constraints for all $t \in [0, t']$. Finally, if $ct' > 2$, then $p_2(t) \leq 0$ for $t \in [0, t' - 2/c)$ and $p_2(t) > 0$ for $t \in [t' - 2/c, t']$ while $p_4(t) > 1$ for $t \in [0, t' - 1/c)$ and $p_4(t) \leq 1$ for $t \in [t' - 1/c, t']$. Therefore $u^*_{1'}$ and $\ddot{x}^*_{B1'}$ both assume the maximum value allowed by the constraints first, then $u^*_{1'}$ switches to the minimum value, and finally $\ddot{x}^*_{B1'}$ switches to the minimum value. A nonzero amount of time elapses between the two switches. □

ACKNOWLEDGMENT

The authors would like to thank Dr. P. Varaiya for his invaluable insight into the AHS problem and Dr. J. Ben-Asher for his helpful comments on the proof of Proposition 5.

REFERENCES

[1] X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine, "An approach to the description and analysis of hybrid systems," in *Hybrid System*, Lecture Notes in Computer Science, no. 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 149–178.

[2] R. Alur, C. Courcoubetis, T. A. Henzinger, and P. H. Ho, "Hybrid automaton: An algorithmic approach to the specification and verification of hybrid systems," in *Hybrid System*, Lecture Notes in Computer Science, no. 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 209–229.

[3] R. W. Brockett, "Hybrid models for motion control system," in *Perspectives in Control*, H. Trentelman and J. Willems, Eds. Boston, MA: Birkhäuser, 1993.

[4] A. Nerode and W. Kohn, "Models for hybrid system: Automata, topologies, controllability, observability," in *Hybrid System*, Lecture Notes in Computer Science, no. 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 317–356, 1993.

[5] M. S. Branicky, "Control of hybrid systems," Ph.D. dissertation, MIT, Cambridge, MA, 1994.

[6] N. Lynch, R. Segala, F. Vaandrager, and H. B. Weinberg, "Hybrid I/O automata," in *Hybrid Systems III*, Lecture Notes in Computer Science, no. 1066. New York: Springer Verlag, 1996, pp. 496–510.

[7] L. Tavernini, "Differential automata and their simulators," *Nonlinear Analysis, Theory, Methods and Appl.*, vol. 11, no. 6, pp. 665–683, 1987.

[8] A. Deshpande, A. Gollu, and L. Semenzato, "The SHIFT programming language and run-time system for dynamic networks of hybrid automata," Inst. Transportation Studies, Univ. California, Berkeley, Tech. Rep. UCB-ITS-PRR-97-7, 1997.

[9] M. Anderson, D. Bruck, S. E. Mattsson, and T. Schonthal, "Omsim—An integrated interactive environment for object-oriented modeling and simulation," in *Proc. IEEE/IFAC Joint Symp. Computer Aided Control System Design*, 1994, pp. 285–290.

[10] M. Lemmon, J. A. Stiver, and P. J. Antsaklis, "Event identification and intelligent hybrid control," in *Hybrid System*, Lecture Notes in Computer Science, no. 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 268–296.

[11] O. Maler, A. Pnueli, and J. Sifakis, "On the synthesis of discrete controllers for timed systems," in *Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, no. 900. New York: Springer Verlag, 1995.

[12] M. Heymann, F. Lin, and G. Meyer, "Control synthesis for a class of hybrid systems subject to configuration-based safety constraints," in *Hybrid and Real Time Systems*, Lecture Notes in Computer Science, no. 1201. New York: Springer Verlag, 1997, pp. 376–391.

[13] H. Wong-Toi, "The synthesis of controllers for linear hybrid automata," in *Proc. IEEE Conf. Decision and Control*, 1997.

[14] A. Nerode and W. Kohn, "Multiple agent hybrid control architecture," in *Hybrid System*, Lecture Notes in Computer Science, no. 736, R. L. Grossman, A. Nerode, A. P. Ravn, and H. Rischel, Eds. New York: Springer Verlag, 1993, pp. 297–316.

[15] J. Lygeros, D. N. Godbole, and S. Sastry, "Multiagent hybrid system design using game theory and optimal control," in *Proc. IEEE Conf. Decision and Control*, 1996, pp. 1190–1195.

[16] L. Hou, A. Michel, and H. Ye, "Stability analysis of switched systems," in *IEEE Conf. Decision and Control*, 1996, pp. 1208–1214.

[17] R. Alur and D. Dill, "A theory of timed automata," *Theoretical Computer Sci.*, vol. 126, pp. 183–235, 1994.

[18] T. Henzinger, P. Kopke, A. Puri, and P. Varaiya, "What's decidable about hybrid automata," in *Proc. 27th Annu. Symp. Theory of Computing, STOC'95*. ACM, 1995, pp. 373–382.

[19] Z. Manna and A. Pnueli, *Temporal Verification of Reactive Systems: Safety*. New York: Springer-Verlag, 1995.

[20] C. Daws, A. Olivero, and S. Yovine, "Verifying ET-LOTOS programs with KRONOS," in *Proc. 7th IFIP WG G.1 Int. Conf. Formal Description Techniques, FORTE'94, Formal Description Techniques VII*, D. Hogrefe and S. Leue, Eds. Bern, Switzerland: Chapman & Hall, 1994, pp. 227–242.

[21] T. A. Henzinger, P. H. Ho, and H. W. Toi, "A user guide to HYTECH," in *Proc. TACAS 95: Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science, no. 1019, E. Brinksma, W. Cleaveland, K. Larsen, T. Margaria, and B. Steffen, Eds. New York: Springer Verlag, 1995, pp. 41–71.

[22] Z. Manna and the STep group, "STeP: The Stanford temporal prover," Computer Science Dept., Stanford Univ., Tech. Rep. STAN-CS-TR-94-1518, July 1994.

[23] J. Lygeros, C. Tomlin, and S. Sastry, "Multi-Objective hybrid controller synthesis," in *Proc. HART'97*, Lecture Notes in Computer Science, no. 1201, O. Maler, Ed. Berlin: Springer Verlag, 1997, pp. 109–123.

[24] C. Tomlin, G. Pappas, and S. Sastry, "Conflict resolution for air traffic management: A case study in multi-agent hybrid systems," Electronic Research Laboratory, Univ. California, Berkeley, Tech. Rep. UCB/ERL M96/38, 1996.

[25] D. Swaroop, "String stability of interconnected systems: An application to platooning in automated highway systems," Ph.D. dissertation, Dept. Mechanical Engineering, Univ. California, Berkeley, 1994.

[26] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 657–672, 1993.

[27] D. N. Godbole and J. Lygeros, "Longitudinal control of the lead car of platoon," *IEEE Trans. Veh. Technol.*, vol. 43, pp. 1125–1135, 1994.

[28] P. Li, L. Alvarez, R. Horowitz, P.-Y. Chen, and J. Carbaugh, "Safe velocity tracking controller for AHS platoon leader," in *Proc. IEEE Conf. Decision and Control*, 1996, pp. 2283–2288.

[29] D. N. Godbole, J. Lygeros, and S. Sastry, "Hierarchical hybrid control: An IVHS case study," in *Hybrid Systems II*, Lecture Notes in Computer Science, no. 999, P. Antsaklis, W. Kohn, A. Nerode, and S. Sastry, Eds. New York: Springer Verlag, 1995, pp. 166–190.

[30] J. Frankel, L. Alvarez, R. Horowitz, and P. Li, "Safety oriented maneuvers for IVHS," in *Proc. American Control Conf.*, 1995, pp. 668–672.

[31] E. Dolginova and N. Lynch, "Safety verification for automated platoon maneuvers: A case study," in *Proc. HART'97*, Lecture Notes in Computer Science, no. 1201, O. Maler, Ed. Berlin: Springer Verlag, 1997, pp. 154–170.

[32] J. Lygeros, D. N. Godbole, and S. Sastry, "A game theoretic approach to hybrid system design," Electronic Research Laboratory, Univ. California, Berkeley, Tech. Rep. UCB/ERL-M95/77, Oct. 1995.

[33] ——, "A verified hybrid controller for automated vehicles," in *Proc. IEEE Conf. Decision and Control*, 1996, pp. 2289–2294.

[34] J. Lygeros, "Hierarchical hybrid control of large scale systems," Ph.D. dissertation, Dept. Electrical Eng., Univ. California, Berkeley, 1996.

[35] R. Alur and T. A. Henzinger, "Reactive modules," in *Proc. 11th Annu. Symp. Logic in Computer Science*. IEEE Computer Soc. Press, 1996, pp. 207–218.

[36] T. Başar and G. J. Olsder, *Dynamic Non-Cooperative Game Theory*, 2nd ed. New York: Academic, 1995.

[37] J. Lygeros, D. N. Godbole, and M. E. Broucke, "Extended architecture for degraded modes of operation of IVHS," in *Proc. American Control Conf.*, 1995, pp. 3592–3596.

[38] S. Sheikholeslam, "Control of a class of interconnected nonlinear dynamical systems: The platoon problem," Ph.D. dissertation, Dept. Electrical Eng., Univ. California, Berkeley, 1991.

[39] J. K. Hedrick, D. McMahon, V. Narendran, and D. Swaroop, "Longitudinal vehicle controller design for IVHS system," in *Proc. American Control Conf.*, 1991, pp. 3107–3112.

[40] H. Peng and M. Tomizuka, "Vehicle lateral control for highway automation," in *Proc. American Control Conf.*, 1990, pp. 788–794.

[41] W. Chee and M. Tomizuka, "Lane change maneuver of automobiles for the intelligent vehicle and highway systems (IVHS)," in *Proc. American Control Conf.*, 1994, pp. 3586–3587.

[42] J. B. Michael, D. N. Godbole, R. Sengupta, and J. Lygeros, "Capacity analysis of traffic flow over a single lane AHS," *ITS J.*, vol. 4, 1998.

[43] A. Hitchcock, "Casualties in accidents occurring during split and merge maneuvers," Inst. Transportation Studies, Univ. California, Berkeley, Tech. Rep., PATH Tech. Memo 93-9, 1993.
[44] J. A. Haddon, "Evaluation of AHS throughput using Smart Cap," in *Proc. American Control Conf.*, 1997.
[45] S. Shladover, "Operation of automated guideway transit vehicles in dynamically reconfigured trains and platoons," U.S. Dept. Transportation, Tech. Rep. UMTA-MA-0085-79-3, 1979.
[46] P. Varaiya, "Smart cars on smart roads: Problems of control," *IEEE Trans. Automat. Contr.*, vol. 38, pp. 195–207, Feb. 1993.
[47] B. S. Y. Rao and P. Varaiya, "Roadside intelligence for flow control in an IVHS," *Transportation Res. Part C*, vol. 2, no. 1, pp. 49–72, 1994.
[48] P. Li, R. Horowitz, L. Alvarez, J. Frankel, and A. Robertson, "Traffic flow stabilization," in *Proc. American Control Conf.*, 1995, pp. 144–149.
[49] R. W. Hall, "Longitudinal and lateral throughput on an idealized highway," *Transportation Sci.*, vol. 29, no. 2, pp. 118–127, 1995.
[50] M. E. Broucke and P. Varaiya, "A theory of traffic flow in automated highway systems," *Transportation Res. Part C*, vol. 4C, no. 4, pp. 181–210, 1996.
[51] A. Hsu, F. Eskafi, S. Sachs, and P. Varaiya, "Protocol design for an automated highway system," *Discrete Event Dynamic Systems*, vol. 2, no. 1, pp. 183–206, 1994.
[52] R. P. Kurshan, *Computer-Aided Verification of Coordinating Processes: The Automata-Theoretic Approach*. Princeton, NJ: Princeton Univ. Press, 1994.
[53] Y. Yang and B. Tongue, "Intra-platoon collision behavior during emergency operations," *Vehicle System Dynamics*, vol. 23, no. 4, pp. 279–292, 1994.
[54] J. Lygeros, "To brake or not to brake? Is there a question?," in *Proc. IEEE Conf. Decision and Control*, 1996, pp. 3723–3728.
[55] J. Lygeros and D. Godbole, "An interface between continuous and discrete event controllers for vehicle automation," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 229–241, 1997.
[56] D. N. Godbole, J. Lygeros, E. Singh, A. Deshpande, and A. Lindsey, "Design and verification of coordination layer protocols for degraded modes of operation of AHS," in *Proc. IEEE Conf. Decision and Control*, 1995, pp. 427–432.
[57] A. E. Bryson and Y.-C. Ho, *Applied Optimal Control*. Hemisphere, 1975.

John Lygeros (S'89–M'97) received the B.Eng. degree in electrical and electronic engineering in 1990 and the M.Sc. degree in control and systems in 1991, both from Imperial College of Science Technology & Medicine, London, U.K. In 1996 he received the Ph.D. degree in the electrical engineering from the University of California, Berkeley.

From June to October 1996 he was a Visiting Postdoctoral Researcher with the California PATH project, Institute of Transportation Studies, University of California, Berkeley. Between November 1996 and September 1997 he was a Postdoctoral Research Associate at the Laboratory for Computer Science of the Massachusetts Institute of Technology. He is currently a Postdoctoral Researcher at the Electrical Engineering and Computer Sciences Department of University of California, Berkeley. His research interests include hierarchical and hybrid systems, nonlinear control theory and their applications to Automated Highway Systems, and Air Traffic Management.

Dr. Lygeros is the corecipient (with Dr. D. N. Godbole) of the 1997 Eliahu Jury award for "Excellence in Systems Research" awarded by the Electrical Engineering and Computer Sciences Department of the University of California, Berkeley.

Datta N. Godbole (M'95) received the B.E. degree in electrical engineering from the University of Pune, India, in 1987, the M. Tech. degree in systems and control engineering from the Indian Institute of Technology, Bombay, India, in 1989, and the Ph.D. degree in the electrical engineering and computer science department from the University of California, Berkeley, in 1994.

Since 1995, he has been working as an Assistant Research Engineer with the California PATH program at the University of California, Berkeley. His research interests include hybrid control systems, hierarchical control of complex multi-agent systems, nonlinear control, and applications to intelligent transportation systems and vehicle dynamics.

Dr. Godbole is a recipient of the 1987 University Gold Medal in engineering from the University of Pune, India, and a co-recipient (with Dr. J. Lygeros) of the 1997 Eliahu Jury award for "Excellence in Systems Research" from the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley.

Shankar Sastry (S'79–M'80–SM'90–F'95), for photograph and biography, see this issue, p. 521.