INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Smart Cars on Smart Roads: Problems of Control

by

Pravin Varaiya

Department of Electrical Engineering and Computer Sciences

December 1991

PATH TECHNICAL MEMORANDUM 91-5

Smart cars on smart roads: Problems of control

Pravin Varaiya
Department of Electrical Engineering and Computer Sciences
University of California: Berkeley CA 94720

[1]Work supported by the **PATH** program, Institute of Transportation Studies, University of California, Berkeley, and NSF Grant ECS-911907.

Abstract

Proponents of Intelligent Vehicle/Highway System or IVHS see it as a new technology which will make a major change in highway transportation. Control, communication and computing technologies will be combined into an IVHS system that can significantly increase safety and highway capacity without new roads having to be built. This paper outlines key features of one highly automated IVHS system, shows how core driver decisions are improved, proposes a basic IVHS control system architecture, and offers a design of some control subsystems. It also summarizes some experimental work. We hope that the paper will stimulate interest in IVHS among control engineers.

1 Introduction

Highway congestion is imposing an intolerable burden on many urban residents. Because congestion occurs when the demand for travel exceeds highway capacity, a sound approach to reducing congestion will involve a mix of policies affecting demand and capacity depending on local circumstances and priorities. These policies include building more highways, reducing demand by raising tolls or other taxes, promoting mass transit or greater vehicle occupancy (car-pooling), and developing a high speed communication network which, for many purposes, can replace the need for travel. This paper discusses the potential of another policy option called Intelligent Vehicle/Highway System (IVHS).

IVHS proponents claim that a proper combination of control, communication and computing technologies, placed on the highway and on the vehicle, can assist driver decisions in ways that will increase highway capacity and safety without building more roads. However, there is a diversity of opinion about the appropriate form of this 'intelligence'. This diversity stems from differences in judgement about:

- Function – the range of driving functions that should be automated, and the degree of automation:
- Architecture – the decomposition of these functions into control tasks, and the assignment of those tasks to IVHS subsystems;
- Design – the division of intelligence between vehicle and highway, and how the enabling technologies are to be combined to realize this architecture;
- Evolution – the timing of system development and deployment, and the extent to which the architecture should accommodate new functions not included in earlier designs:
- Evaluation – the effectiveness, costs and benefits of different IVHS proposals.

This paper is concerned with the first three aspects alone.

§2 presents a framework for describing IVHS functions and their relation to key driver decisions. Different IVHS proposals can be compared according to the degree of influence they exert on those decisions. These proposals may range from systems which are relatively simple from a control viewpoint, since they merely provide information or advice to drivers, to systems that are fully automated and leave few decisions to the driver. We argue that only full automation can achieve significant capacity increases.

93 outlines a fully automated IVHS system which promises a threefold increase in capacity. Designing such a system poses a challenging problem of control.

§4 proposes a four-layer hierarchical control architecture which decomposes this problem into more manageable units. Starting at the bottom, the layers are called the regulation, planning, link, and network layers. The regulation layer's task is to execute a set of feedback laws for throttle, braking, and steering. The planning layer's task is

to coordinate the movement of neighboring vehicles. These tasks can be implemented by protocols executed by a finite state machine supervisor. The link layer is responsible for the control of aggregate traffic flow. Lastly, the network layer assigns routes to each vehicle and controls admission into the automated highway network.

§5 lists significant elements of the hardware needed to support the control architecture described in §4.

§6 summarizes experimental work being done at Berkeley.

§7 provides some remarks on IVHS design.

2 IVHS functions

Table 1 lists a sequence of six decisions made by an automobile driver in the course of a trip.[1] The decisions are divided into pre-trip, in-trip and post-trip phases. Also listed are IVHS goals appropriate to each decision and more tangible tasks which, if properly carried out,. promote these goals. The last column indicates that the tasks may be accomplished using various intervention strategies: providing information, offering advice, taking direct control of the decision. or changing incentives by pricing, eg. tolls.

Phase	Driver decision	IVHS goal	IVHS task	Strategy
Pre-trip	Trip generation, modal choice, etc	More efficient resource utilization	Demand shift	I, P
In-trip	Route choice	Reduce travel time	Route guidance and flow control	I,A,C
	Path planning	Smooth traffic	Congestion control	I,A,C
	Maneuver	Increase safety, flow	Vehicle coordination	I,C
	Regulation	Increase safety, flow	Proper spacing, steering, etc.	I,C
Post-trip	Parking, etc	Add value	Efficient use	I,A,C,P

A = Advice, C = Control, I = Information, P = Pricing

Table 1: Driver decisions and IVHS functions

In our view, this chain of six decisions forms the ***core*** of a larger set of decisions that a driver makes. When we include all the choices that eventually culminate in the decision to undertake a trip, this set becomes very large indeed. Even the set of in-trip decisions includes other items such as stopping to pay **a** toll, stopping for gas, listening to the radio, using the telephone, talking to passengers. Drivers may attach great value to a system that makes such non-core decisions more efficient. For example, an automatic toll collection system may eliminate the need to stop at a toll booth, and a system that allows the driver

[1] This table applies to a typical commute trip. Additional decisions are involved in trips by drivers of commercial vehicles.

to reserve a parking place *en route* may save much time. But these decisions do not seem to belong to the core of IVHS.

Returning to Table 1, note that as the strategy adopted to influence driver decisions shifts from giving information to offering advice to exerting preemptive control, the decisions become more automatic and predictable, and the burden on system 'intelligence' increases. In the remainder of this section we focus on the in-trip decisions. We shall argue that partial automation can not achieve significant increases in capacity.

Table 1 indicates how these decisions can be improved to lead to higher capacity and safety. First, a better choice of route can reduce travel time. This choice can be improved by 'off line' information in the form of maps useful to drivers unfamiliar with the highway network): and by 'on line' information about changes in traffic delays (caused by incidents or recurrent congestion), enabling the route to be adapted to changing traffic conditions. But simulation and analytical studies and data from demonstration experiments suggest little or no improvement from better choice of route under recurrent congestion and some improvement under incident-induced congestion.[2] Consideration of the second decision shows that additional improvements are possible if instantaneous speeds are altered in ways that reduce or eliminate congestion.[3] The evidence suggests an upper bound of 15 % on the capacity increase that can be achieved by improving route choice and path planning decisions.[4]

We now turn to the third in-trip decision – the way drivers maneuver their vehicles. hfaneuvers refer primarily to the way drivers change lanes, including entry and exit from a highway. Maneuvers require the coordination of movement of neighboring vehicles. Improper coordination can result in congestion and accidents. In today's unautomated traffic system, coordination is achieved by signaling (turn signals and brake lights) and social convention (eg. providing room to a driver in the adjacent lane who indicates an intention to change lanes): buttressed by a legal code. A partially automated system can improve maneuvers by providing a collision warning signal or a collision avoidance system which temporarily preempts driver action. There does not seem to be any analytical or experimental work that estimates the benefits of such a system in terms of accident avoidance. It seems unlikely, however, that such a system will contribute to a significant capacity increase;

[2] A simulation study based on the CACS project suggests that travel time in Tokyo could be reduced by 6 % [1]; U.K. researchers estimate an average benefit of 10 % from dynamic route guidance [2]; preliminary results from the Berlin route guidance experiment show no savings in average travel time under normal conditions [3]; simulations of the Santa Monica freeway (SMART) corridor suggest insignificant savings under recurrent congestion and savings on the order of 10 minutes for a 40 minute trip under incident induced conditions [4, 5]; theoretical considerations also suggest little or no benefits from route guidance under recurrent congestion [6]; lastly. experiments using the CONTRAM simulation model show that single 'best route' guidance can even lead to *negative* benefits [7]. I am indebted to H. Al-Deek for many of these references.

[3] A theoretical model shows considerable room for improvement *provided* speeds can be controlled [8]; however, a careful examination of driver response to *advisory* speeds posted by the Dutch Motorway Control and Signalling System showed no increase in capacity [9].

[4] A study commissioned by the European Prometheus program projects a decrease in travel time of 10 % if all vehicles are equipped with a route guidance system. This estimate was announced at the IVHS System Architecture Workshop. University of Michigan, Ann Arbor, October 24-25, 1991.

therefore: we may still accept our upper bound of 15 %.[5]

The fourth in-trip decision concerns regulation. This primarily refers to the way a driver adjusts the vehicle speed while keeping to one lane. **A** fundamental empirical fact about driver behavior is that, in steady state conditions, this speed $v(d)$ is an increasing function of the distance d from the vehicle in front. The flow $\phi(d)$ in vehicles/lane/minute is the product of speed in meters/minute and the number of vehicles per meter

$$\phi(d) = v(d) \times \frac{1}{d+s}$$

where s is vehicle length. While the driver 'response function' $v(d)$ depends on the physical condition of the highway (lane width, illumination, surface conditions, etc.), empirical studies suggest that the flow has a maximum value of about 40 vehicles/lane/minute.

Taking all these factors into account one can estimate that any partially automated IVHS system which leaves the driver in control of the vehicle can achieve a capacity of at most 15 % above 40 vehicles/lane/min. Driver behavior will continue to be the capacity 'bottleneck' in such a system.[6]

3 A fully automated system

We first argue that, organizing the traffic in platoons can dramatically increase capacity. We then sketch a scenario of an IVHS system in which the four in-trip decisions are fully automated. Finally. we formulate the control system design problem.

3.1 The platooning concept

One key to increasing capacity is to organize traffic in tightly spaced platoons as illustrated in Figure 1. Taking intra-platoon spacing as d, inter-platoon spacing as D, vehicle length as s. and a steady state speed of v m/min, gives a capacity of

$$C = v \times \frac{12}{ns + (n-1)d + D} \text{ vehicles/lane/min} \qquad (1)$$

if traffic is organized in n car platoons. Some values are listed in Table 2 for $s = 5$ m and $v = 1{,}200$ m/min or 72 km/h.

[5] We are *not* arguing that partial automation is useless. Indeed many aspects of driving conditions can be improved by better incident detection, traffic management, real time information, collision warning or avoidance systems, etc. These improvements may be greatly valued **by** some drivers. We **ore** arguing that the resulting capacity increases will be small.

[6] We are concentrating on capacity in this paper. Data suggest that incorrect driver decisions account for 90 % of accidents, so automation may also reduce accident rates.

n	d	D	C
1	–	**30**	35
5	2	60	64
15	2	60	105
20	**1**	**60**	**133**
∞	1	–	200

Table **2:** Capacity with platoons

The case $n = 1$ which gives $C = 35$ represents today's capacity limits. The case $n = 20$ appears feasible as is argued later. It quadruples capacity. Equation **(1)** suggests other design variables which affect capacity including, especially, vehicle size and speed. Two issues need to be addressed. The first concerns safety. The second has to do with how platoons would be formed and how vehicles would maneuver under a platooning system.

We give a cursory examination of safety, evaluating only the possibility and effect of a vehicle colliding with a suddenly decelerating vehicle in front of it. We assume that the inter-platoon distance of 60 m allows enough time for the lead vehicle in the rear platoon to slow down and avoid an accident.. It remains only to consider collisions within a platoon. If a vehicle decelerates at a_1 m/s/s and the following vehicle immediately decelerates at a_2 m/s/s, and if $\delta a = a_1 - a_2 > 0$ there will be a collision and the relative speed upon impact is

$$\delta v = \sqrt{2\delta a \delta x}$$

where δx = d is the initial intra-platoon distance. Take an extreme case: $a_1 = 10$ m/s/s (a deceleration of 1g), $a_2 = 7$ m/s/s, and $d = 1$ m. Then $\delta v \approx 7$ km/hour, which we consider to be a 'safe' impact. An important conclusion follows from this exercise. Since a human driver has a reaction delay of between **0.25** and **1.2** seconds, that driver cannot guarantee adequate safety. The vehicle must be under automatic control so that it can decelerate immediately. Moreover, the vehicle controller will need to sense the speed and acceleration of the vehicle in front, of it. This controller is discussed further in §4.2 and §6. We only note here that experimental evidence indicates that such controllers can be built.[7]

3.2 An IVHS scenario

Let, us imagine an automated network of interconnected highways. Appropriately equipped vehicles may enter and exit this network at various 'gates' and travel through this network under control authority distributed between the highway and vehicle. The network is embedded in a larger transportation system containing several inhomogeneous networks.[8] For

[7] **This superficial discussion of safety will raise more questions than it answers: Why is relative speed on impact the correct measure of severity of impact? What happens if, upon impact, a vehicle moves into an adjacent lane? What about failures? Such questions are not yet adequately answered.**

[8] **The automated network mag consist of segregated lanes within each highway similar to today's** HOV **lanes. Procedures have to be in place to verify that vehicles entering the automated network are properly equipped.**

esample, a vehicle leaving the automated network may enter an unautomated network of urban arterials.

Upon entering a gate, the vehicle announces its destination. The (IVHS) system responds by assigning to it a *route* through the network. **A** route R is a sequence like

$$R = (H_1, s_1, f_1),\ (H_2, s_2, f_2)\ \ldots$$

The interpretation is that the route consists of a sequence of segments. The first segment of the route is on the highway named H_1 starting at gate s_1 and finishing at gate f_1; the second segment is on highway H_2 from gate s_2 to f_2; and so on. It is understood that an interchange lane connects f_1 on H_1 to s_2 on H_2, etc., as illustrated in Figure **2.**

We assume that a vehicle continuously senses the section on which it is traveling. **A** 'section' is a triple $(H.l,d)$ where H is the highway name, l is the lane number, and d is the lane section number. **A** section is about. 100 m long. Suppose the vehicle enters a highway H through some gate s at section (H,l_1,d_1) and announces (based on its route) that it will esit at gate f. In response the system assigns it a ***path.*** (l_2, d_2, l_3). The interpretation is that the vehicle must change to lane l_2, travel along it until section d_2, and then change to lane l_3 from which it exits via gate f. See Figure **3.**

Its path having been assigned: the vehicle must evolve in real time a *plan* which is close to this path and esecute a *trajectory* that conforms to the plan.

With this scenario as background, we can formulate the problem as one of actually designing the control system which carries out the tasks of route and path assignment, plan evolution and trajectory esecution – all in **a** way that organizes the traffic in platoons.

4 Control system architecture

Figure 4 gives a block diagram of a four-layer architecture. Starting at the top, the layers are called network: link: planning, and regulation. Their functions correspond to the four in-trip tasks of Table 1.

There is one network layer controller for the whole automated network. Its task is to assign a route to each vehicle entering the system. Thus, from an abstract viewpoint, this controller must determine a routing table with entries of the form

$$\text{(origin: destination)} \rightarrow \text{Vehicle route}$$

This table would be updated (perhaps every **15** minutes) in a way that optimizes traffic flow. The updates would be based on information about the aggregate state of the traffic. Because there is a large literature in traffic engineering and operations research that deals with the problem of calculating the routing table, we will not discuss this problem any further. We note! however, a variety of design options: route calculation done centrally vs by the vehicle's on-board computer; static routes vs dynamic routes which are updated as the vehicle's trip progresses; a 'socially optimum' solution which minimizes total travel

time vs an 'individually optimum' solution which minimizes each vehicle's travel time. The network layer is also responsible for controlling admission into the automated system. Such control could be exercised at the 'gates' to the network in a way similar to ramp metering. The purpose of admission control would be to limit entry at those gates where unrestricted entry would lead to congestion. We do not discuss admission control further.

There is one link layer controller for a long segment of each highway. Its tasks are to assign a path to each vehicle in ways that balance traffic across all lanes and to assign a target speed for each section in order to smooth flow, avoid congestion, and adapt to incidents. The assignment is recalculated in response to changes in aggregate density and speed on each section, perhaps on a minute by minute basis. (In our design the link layer also assigns the target platoon size.) We do not discuss this layer any further.

There is a planning layer controller and a regulation layer controller for each vehicle. Their respective tasks are to evolve a plan which is close to the assigned path and to execute the vehicle's trajectory which conforms to the plan. We'will discuss these controllers in greater detail. Before doing that, however, we make two observations about the architecture. First,. the four-layer hierarchy is natural from the viewpoint of control system design since it satisfactorily resolves three dimensions of difference involved in the four tasks: time scale, information span: and spatial impact of the decisions. The three dimensions are summarized in Table **3.** We see that as we go up the hierarchy, the frequency of decisions decreases, more aggregated information is used, and the impact of the decision affects vehicles over larger distances.

Layer	Time scale	Information span	Spatial impact
Network	Once every 15 min	Network-wide aggregate flow data	Route assignment affects entire network
Link	Once every min	Aggregate density, flow for each highway section	Path, section speed assignment affects several kms of hwy
Planning	Plan evolves every minute	Coordination with neighboring vehicles	Plan affects neighboring vehicles
Regulation	Fraction of second – vehicle time constant	Vehicle state information	Individual vehicle or platoon

Table **3:** Differences among layers

The second observation is that the layers of Figure **4** are logical entities. Many different physical designs can implement the same logical entities. In particular, different choices of partition of control authority between highway and vehicle are compatible with this architecture. These choices should be judged by criteria of cost of implementation, robustness, performance, private vs public ownership of equipment, etc. We noted above some of these choices for the network layer. Another illustration of this point is provided in the following discussion of the planning and regulation layer design.

A variety of controller designs have been proposed for the planning and regulation layer tasks. At one extreme lie designs in which a centralized controller determines the

position of every vehicle, similar to the way modern urban trains are controlled. Such designs were studied in the early 1970s by groups at TRW, GM, Rohr Industries and elsewhere. Those studies and others are carefully reviewed in [10]. At the other extreme lie proposals made since the late 1980s and inspired by robotics and AI based approaches to the control of an autonomous vehicle navigating in an unstructured environment [11, 12, 13, 14, 15]. Such approaches emphasize recognition, learning, and trajectory planning in the face of diverse 'threats' and 'obstacles'.

In our opinion both extremes must be avoided. The centralized train-like design achieves total control and predictability of traffic but at an enormous cost in computation and communication. The approach will also require major design effort to protect against catastrophic failures. The autonomous vehicle approach, on the other hand, is highly decentralized! but a great deal of 'intelligence' is required on the vehicle so that it can maneuver safely in the face of uncertainties in the movement of neighboring vehicles. The approach relies on technology that is likely to be expensive and is not yet proven.[9]

The design presented below strikes a balance between these extremes in a way that avoids their weakest points. Unlike the centralized approach, it distributes most of the information and control authority among individual vehicles. Unlike the autonomous vehicle approach, it reduces uncertainties by coordinating the movement of neighboring vehicles. Neighbors' movements are then more predictable and the intelligence needed in each vehicle is drastically reduced. We believe that most of the technology needed to implement the design is already available.

We now describe the design. The key idea is to distribute the control and to restrict and coordinate the movement of vehicles so that the control task is simplified. There are two sets of restrictions. The first set restricts a plan to consist of (1) an initial sequence of lane change maneuvers during which the vehicle moves from lane l_1 to l_2, a single lane change at a time; (2) a lane keeping mode during which the vehicle stays in lane l_2 until section d_2; and (3) a terminal sequence of lane change maneuvers to move from lane l_2 to l_3. See Figure 3. A further restriction is that only free agents may perform a lane change. (A free agent is a 1-car platoon; the lead vehicle of a platoon is called a leader, the rest are followers. See Figure 1.)

Thus platoon formation and breakup occur only during the lane keeping mode. We need two maneuvers for this: in a merge maneuver, two platoons join to form one platoon; in a split maneuver, one platoon splits into two. If a vehicle in a platoon wishes to make a lane change?it must first become a free agent. This will require one split maneuver (if it is the first or last vehicle in a platoon) or two split maneuvers (otherwise). This set of restrictions allows us formally to define a plan as a finite sequence of lane change, merge and split maneuvers with the further restriction that only a free agent can make a lane change. These maneuvers are indicated in Figure 5.[10]

[9] A careful study is needed to determine whether our criticism of these types of design is valid.

[10] In the merge maneuver, the platoon led by B joins the one led by A. There are two variants of the split maneuver. depending on whether it is the leader (A) or a follower (B) that wants to split. A lane change maneuver is initiated by the free agent A and will involve a vehicle (B or B_i) in the adjacent lane and the platoon leader in the far lane.

The link layer controller assigns to each section of the highway a target platoon size N and target speed v. Leaders attempt to meet these targets as well as possible, consistent with safety. The second set of restrictions is concerned with safety. We require that (1) only leaders (and free agents) can initiate maneuvers and followers maintain platoon formation at all times; **(2)** a leader carries out at most one maneuver at a time; **(3)** a leader coordinates each maneuver with leaders of neighboring platoons to ensure safety; **(4)** only after agreement is reached, does the leader execute the maneuver.

Some comments on these safety-related restrictions may be worthwhile. Because platoons of size at least two can only engage in a merge or split maneuver, these decisions should be taken by a leader.[11] The followers merely follow. If a follower wishes to become a free agent (in order to change lane), it must request its leader to initiate the necessary split maneuver. These reasons suggest restriction **(1)**. Restriction **(2)** greatly simplifies the design: if a leader is permitted to plan two consecutive maneuvers simultaneously, coordination becomes more complex; moreover, it then seems necessary to allow for the possibility that the second maneuver might be cancelled after the first is executed. Restriction **(3)** is at the heart of safety: maneuvers of adjacent vehicles must be coordinated to ensure their compatibility. This concern for safety is clearest in the case of lane change in Figure **5.** The free agent in the bottom lane must make sure that no vehicle from the two upper lanes will move into the empty space in the middle lane.

With these restrictions imposed in the interest of simplicity and safety, we focus on the design of the planning and regulation layers. We assume that the planning layer knows at all times the link layer-assigned target speed, platoon size, and path, and the vehicle's current position (highway section,). It also knows whether it is a leader, follower, or free agent. Based on this information it decides which maneuver (merge. split, or lane change): if any. it should attempt to evolve a plan close to the assigned path. It then exchanges messages with the relevant neighboring platoon leaders to coordinate their movement so that the vehicle can safely carry out the maneuver. Upon reaching agreement with these neighbors. it requests its regulation layer to implement the feedback law corresponding to that maneuver.

As esplained in **\$4.2** five types of feedback laws are needed. The regulation layer implements one of these laws following the request from its planning layer. When the maneuver is complete: the regulation layer informs the planning layer which may then plan the nest maneuver. The feedback law computes the values for the throttle, braking and steering actuators, based on sensor observations of the vehicle's state. The physical layer is a differential equation model of the vehicle state, with actuator inputs and sensor outputs.

The vehicle's planning and regulation layer controllers together form a hybrid controller in the sense of [16]. (They also form an 'intelligent?control system [17].) The planning layer is a discrete event system which supervises the regulation layer which directly controls a continuous-variable system. The next two subsections describes these layers in more detail.

[11] It may be desirable, however, to add two maneuvers in which vehicles enter or leave the automated highway in platoons (rather than **as** free agents) at entrance and exit ramps. These maneuvers could be modifications of the lane change maneuver.

4.1 Planning layer

The planning layer has three tasks:

- to decide which maneuver to attempt in order to realize its assigned path,
- to coordinate that maneuver with the planning layers of neighboring vehicles to ensure safety, and
- to supervise its regulation layer in the execution of a trajectory corresponding to the maneuver.

In order to carry out these tasks the planning layer tracks some information which we call the *planning_state*

$$planning\_state = (N, v, l, d, veh\_ID, pltn\_ID, pltn\_size, pltn\_pos, busy)$$

Here N, v are the target platoon size and speed assigned by the link layer, l and d are the highway lane and section where the vehicle is currently located (determined by some sensor). veh_ID and $pltn\_ID$ are identifiers (the latter may be the ID of the lead vehicle), $pltn\_size$ is the current size of the platoon, $pltn\_pos$ is the vehicle's current position in the platoon ($pltn\_pos = 1$ means leader), and $busy$ is a flag which is set if the vehicle is currently engaged in a maneuver and unset otherwise.

The three planning tasks involve discrete events, therefore this layer is best realized as a discrete event dynamical system. Several formalisms are possible [18, 19]. Finite state machines are selected because of the software available for formal specification and verification [20, 21].

Because the three tasks are hierarchical in nature, this controller is realized as a set of coupled state machines arranged in the hierarchy shown in the left panel of Figure **6.** The top layer contains a single machine PM (path monitor). It compares the current location with the assigned path, selects a maneuver, and then orders machine SUPR. (supervisor) to carry out, that maneuver. If the maneuver is lane change and if the vehicle is not a free agent, SUPR. orders machine BFRE (become free agent) to conduct one or two split maneuvers at the end of which the vehicle becomes a free agent. The coordination of neighboring vehicles' planning layers needed for each of the three maneuvers is carried out (after being so ordered by SUPR or BFRE) by one of the ten machines in the lowest layer. These machines are named BPCmerge, APCsplit, **...** APRmerge, APRsplit, **...** . We now consider what these machines do.

They achieve coordination by exchanging messages, which we call ***protocols,*** because their function is analogous to communication protocols [22]. Consider first the merge maneuver at the top of Figure ***5.*** This maneuver requires a protocol between two vehicles – the rear vehicle B commands the maneuver, and the forward vehicle A responds to it. B's merge protocol machine is called BPCmerge **(P** for protocol, C for command). A's machine is called APRmerge (R. for response). Of course, each vehicle's controller must have both machines because it may be required to command or to respond. From Figure ***5*** again, in

the case of a split, a vehicle may find itself in one of four roles: as leader it commands a split or responds to one (handled by APCsplit, APRsplit); as follower it commands a split or responds to one (BPCsplit, BPRsplit). Lastly, again from Figure **5,** in a lane change, a vehicle may take on one of four roles: as vehicle A it commands a lane change maneuver (APCchg), or it responds to a command as a leader in the next lane (BPRchg), as a follower in the next lane (BLPRchg), or as a leader in the far lane (CPRchg). Thus this coordination schemes gives rise to ten protocol machines in all.

We now describe in more detail how the two merge protocol machines are specified and verified for correct behavior. The full design is described in [23]. The complete planning layer finite state machine has about 500,000 states and **3,000,000** transitions.

Figure 7 shows a pair of state machines. The top machine is in vehicle B which commands the merge and the bottom is in vehicle **A** which responds to the request. Transitions in the two machines which correspond to sending and receiving a message are coupled together. Thus, for example, the transition labeled *SEND request_merge* in **B's** machine occurs simultaneously with the transition *REC request-merge* in **A's** machine. Similarly, *SEND ack_request_merge* in A and *REC ack_request_merge* in B occur together. We now explain how the machines operate.

Suppose that initially the two machines are in their *idle* states. Suppose B's machine makes the transition *MERGE PLATOON* (this would be coupled to a transition from B's SUPR machine). It then enters a new state in which it *checks if in range* meaning that B determines if there is a vehicle in the range of its front distance sensor with which it can merge. It also checks if its *busy* flag is unset (in which case the vehicle is not engaged in a maneuver). If the answer to either test is *no,* the machine returns to the idle state. If the answer to both tests is *yes,* B sets the busy flag and then makes the transition *SEND request_merge,* i.e. it sends a merge request to the leader of the platoon, vehicle A, that it located with its distance sensor. B's machine then moves into the *wait* state. If it now receives a negative acknowledgement from **A** *(RECnuchequest-merge),* B unsets the busy flag and returns to *idle.* If it receives a positive acknowledgement *(RECack_request_merge),* it. moves to a state in which it orders its regulation layer to accelerate and merge with A's platoon. When its regulation layer completes execution of the merge, B sends a confirm merge message to **A** *(SEND confirm-merge),* unsets its busy flag, and returns to *idle.*

When **A** receives *request_merge* from B, it checks if it is busy and whether the combined platoon size is below the target size N, i.e. $pltn\_size(A) + pltn\_size(B) \leq N$. If the answer to either test is *no,* it sends a negative acknowledgement to B. If the answer to both tests is yes, **A** sets its own busy flag (as it is now engaged in a maneuver), sends a positive acknowledgement to **B** and moves to the *wait for confirm* state. When it receives *confirm-merge* from B, **A** updates the *planning_state* (of the combined platoon), unsets its busy state! and returns to idle. The merge maneuver is complete when **A** and **B** both return to *idle.*

We want to propose a formal specification of the merge protocol state machines, BPCmerge and APRmerge, and prove or verify that this specification is correct. However, the machines of Figure *7* and our interpretation of how they operate are inadequate for the

purposes of formal specification and verification. Two reasons account for this inadequacy. First, we need a formal language to specify both the state machines and the criteria that define correct behavior. We have selected COSPAN [20, 21], to be introduced shortly. Second, and more importantly, the states and transitions of the two machines in Figure 7 refer not only to the protocol messages *per* **se** but also to other variables and events that are part of the 'environment' in which the merge protocol operates. **A** protocol design must make a sharp separation between the protocol itself and the environment within which it works. For the merge protocol, this environment consists of five distinct elements:

- B's distance sensor which locates **A** within a certain range,
- the status of B's busy flag,
- the response of B's regulation layer to the command to merge with A's platoon?
- the status of **A's** busy flag,
- the result of the test $pltn\_size(A) + pltn\_size(B) \leq N$ performed by **A.**

Because the machines of Figure 7 confound the operations of the protocol machines and their environment, the first step towards formalization is to model each separately. The result, is the seven state machines of Figure 8. The two machines, BPCmerge and APRmerge. above the horizontal line are the protocol machines themselves. The remaining five. BR. BQ, BV, AQ and **ASUM,** respectively model the five environment elements listed above.[12] The lines interconnecting the machines indicate how they are coupled.

The seven machines of Figure 8 are specified in COSPAN, a formal language for describing coupled non-deterministic state machines. We will not describe the syntax of COSPAN here. Instead we give an equivalent description in terms of the seven state machine diagrams of Figure 9. These diagrams are interpreted as follows. The circles denote states, the double circles denote initial states. In each state, a machine produces one output selected non-deterministically from the set enclosed in { ... }. The arcs connecting states are transitions. **A** transition is ***enabled*** at a particular time if the output of all seven machines satisfies the condition associated with that arc. For example, in state ***CHECK RANGE,*** BPCmerge produces output *check_range*, and the transition from ***CHECK RANGE*** to ***SET BUSY*** is enabled only if machine BR produces output ***car-ahead*** and BQ produces output *not_busy*.[13]

A ***behavior*** of the system of machines in Figure 9 is defined to be any infinite sequence of pairs of state-output vectors $(x(t), y(t)), t = 0, 1, \ldots$ $x(t)$ and $y(t)$ have seven components,

[12] Referring back to Figure 6. the machines listed in the right panel model the environment for the protocol machines listed in the left panel. Thus. for example, the environment of PM (path monitor) consists of RDSNR (road sensor) and LN# (lane number) which respectively model the result of checking the assigned path with the vehicle's section and lane against the path assigned by the link layer.

[13] Observe that the protocol machines BPCmerge and APRmerge are deterministic while the environment machines are non-deterministic. Uncertainty in the environment (eg. the range sensor BR may or may not find a vehicle with which to merge) is typically modeled by non-determinism, while the control is typically **a** deterministic function of the observations.

one component for each machine) such that: $x(0)$ is the vector of initial states, the vector $y(t)$ is produced in state $x(t)$, and the transition $x(t) \rightarrow x(t+1)$ is enabled by the output $y(t)$. The system of Figure 9 thus defines a set of behaviors. The ***language*** generated by this system is denoted $\mathcal{L}$. It consists of all infinite sequences $y(t), t = 0, 1, \ldots$, such that $(x(t), y(t)), t = 0, 1, \ldots$, is a behavior for some state sequence $x(t)$, $t = 0, 1, \ldots$. In other words, $\mathcal{L}$ is the set of **all** output sequences that can be generated by the protocol machines and their environment.

The correctness of the behaviors of the merge protocol machines is verified in two steps: we first specify the language of ***acceptable*** output sequences, denoted M; we then check that, every generated output sequence is acceptable, i.e. $\mathcal{L} \subset \mathcal{M}$.[14] M is specified by the machine of Figure 10 called the ***monitor,*** together with ***acceptance*** conditions. The monitor is just like the machines in Figure 9, except that it has no outputs (so it cannot affect the behavior of the protocol machines); and its transitions are entirely determined by the outputs of the protocol machines, hence the name monitor. The acceptance conditions are **cyset** $\{0\}$ and **recur** $1 \rightarrow 2$, $3 \rightarrow 0$. Let $z(t)$ be the state of the monitor at time t. Then, by definition of cyset and **recur,** M consists of all infinite sequences $y(t)$ of the machines of Figure 9 such that (1) after a finite T, either $z(t) \equiv 0$ for all $t > T$; or (2) one of the transitions

$$z(t) = 1 \rightarrow z(t+1) = 2, z(t) = 3 \rightarrow z(t+1) = 0$$

occurs infinitely often.[15] Having defined the machines in Figures 8 and 9, COSPAN does an exhaustive search to determine whether $\mathcal{L} \subset M$. The answer turns out to be yes for the design presented here.

We now argue why sequences in M should be considered acceptable so that passing the test. $\mathcal{L} \subset \mathcal{M}$ indeed verifies that the merge protocol machines behave correctly. The monitor starts in state **0.** The transition into state 1 is enabled only if BPCmerge outputs *req_merge*, indicating the start of the merge maneuver. However: this transition will not occur unless B's range sensor machine, BR, outputs ***car-ahead*** and B is not busy, i.e. BQ outputs *not_busy*. It is possible that BR or BQ will never select these outputs since they are non-deterministic. Therefore, there is an acceptable behavior which will cause the monitor never to leave state **0** – the **cyset** $\{0\}$ condition. Once the merge request is issued, APRmerge can either grant or deny the request, causing the monitor to go through state 2 or 3, respectively. In either case the monitor will return to **0,** and the merge maneuver will be repeated. Thus the transitions $1 \rightarrow 2, 3 \rightarrow 0$ can occur infinitely often – hence the **recur** condition.

At this stage we can conclude that the merge protocol is correctly implemented by the pair BPCmerge and APRmerge since $\mathcal{L} \subset M$. The astute reader might, however, raise two objections. The first objection will be that while $\mathcal{L} \subset M$ is necessary for correctness, it is not sufficient. For example, 'correctness' would imply that the monitor should make

[14] Thus $\mathcal{L}$ and $\mathcal{M}$ are ω-regular languages and verification or correctness is formally posed as a question of language containment.

[15] The **cyset** condition means that the monitor state remains in the cyset after some time; the **recur** condition means that those transitions occur infinitely often. These are often called 'fairness' conditions, see eg. [24].

the transition $1 \rightarrow 2$ infinitely often if BR always outputs ***car-ahead,*** **BQ** and **AQ** always output *not-busy* and **ASUM** always outputs *sum_ok.* (Such a test guarantees 'liveness': the merge maneuver is successfully undertaken if the environment does not prevent it.) The response to this valid objection is the following: the merge maneuver is so simple that the designer can understand it fully and can therefore translate this understanding into a full set of tests,

$$\mathcal{L} \cap \mathcal{E}_i \subset M \cap \mathcal{M}_i, \quad i = 1,\ldots,N$$

where $\mathcal{E}_i(\mathcal{M}_i)$ is a restriction on the behavior (monitor) machines, expressed by different **cyset** and **recur** conditions. The proposed design did, in fact, pass such a comprehensive list, of tests.

The second objection is more compelling. It accepts that the designer can arrive at a comprehensive list for the simple merge maneuver, but argues that the complete planning layer is so complex (involving *500,000* states) that it is practically impossible to specify a comprehensive list of tests. This objection is akin to the argument that it is practically impossible to test, for correctness of a large and complex computer program, especially if – as in the protocol case – the program is interactive. The response to this argument is that software designers should control program complexity by adopting the discipline of 'structured programming', i.e. by making the program modular and hierarchical and using abstractions to limit, interaction among modules.

An approach similar to structured programming was followed in the design of the planning layer state machines. **As** shown in Figure 6, these machines are arranged in a hierarchy. The ten protocol machines in the lowest layer are specified and tested one maneuver at a time. In designing the upper layer machines these protocol machines are replaced by very simple abstractions.[16] In this way each simple sublayer of Figure **5** can be verified separately and, it is hoped, comprehensively, so that one may have confidence in the final design.

4.2 Regulation Layer

The design of the platoon layer presented above presupposes the capability on the part of the regulation layer to implement five types of feedback control laws to accomplish certain tasks. We describe these tasks briefly.

Follower spacing control. When a vehicle is a follower, it must be controlled by a feedback law that maintains the required tight spacing with the vehicle in front of it in its platoon. This control action is typically decomposed into longitudinal control which determines acceleration and braking, and lateral control which determines the steering action needed to maintain the vehicle in its lane. Earlier work on longitudinal control assumed that a follower had access to the relative distance and speed between itself and the vehicle in front. When a platoon of vehicles is controlled in this way, there is a 'slinky' effect and vehicles in the rear of the platoon tend to exhibit an oscillatory movement [26]. However?if every platoon follower in addition has access to the speed and acceleration of

[16] COSPAN has **facilities for checking that the simplifying abstractions or reductions are valid** [21, 25].

the leader: the performance is immensely improved. Simulation studies indicate that a platoon of size 15 can be adequately controlled to maintain a 1 m spacing [27, 28, 29, 30]. Eventually: it will be necessary to take into account the coupling between longitudinal and lateral movements of the vehicle [31].

Leader tracking target speed. In the lane keeping mode, the leader should try and track the target speed announced by the link layer, while maintaining a safe headway ($\approx$ 60m) from the vehicle in front. This task should be similar to that of current cruise control appropriately modified to account for the headway requirement. For a summary of European effort in "intelligent cruise control" see [32].

Accelerate to merge. This feedback law is used by a leader to accelerate and merge with the platoon in front. This law could be implemented by first calculating a nominal trajectory given the distance and speed of the vehicle in front (the last vehicle in the preceding platoon), and then embedding the corresponding nominal open loop control in a longitudinal control feedback loop.

Decelerate to split. A follower who has just assumed the role of leader is required to slow down to achieve a safe headway (60m) from the vehicle in front. This law should be similar to: and simpler than, the previous feedback law.

Free agent change lane. This feedback law enables a free agent to move to a vacant space in the adjacent lane. Again the approach of embedding a nominal open loop control in a feedback loop seems appropriate. This maneuver will require accurate position sensing systems. This task seems to involve the most demanding sensor requirements.

It should be noted that these are types of control laws; each type represents a class of laws indexed by several parameters. For instance, the spacing control law would be parametrized by the required spacing distance; similarly: the change lane law would be parametrized by the location of the vacant space, and the speed of vehicles in the adjacent lane. There may be other parameters, as well, provided by 'preview' information about the geometry of the road, road conditions, etc, [33].

The proposed design specifies in a very simple fashion the interface between the platoon and regulation layers: the platoon layer issues a command, and the regulation layer eventually returns a response indicating successful completion. This interface needs to be enriched: the platoon layer may pass several parameters to the regulation layer, and the latter may return 'success' or various kinds of 'errors' and 'exceptions'. The theory of control of such systems remains to be developed.

In conclusion, we note the absence of published research that considers the last three types of control laws. There is, also, little research dealing with how the regulation layer should switch from one control law to another. For example, what is a graceful way of switching from the "accelerate to merge" law to "follower spacing control" law?

5 Hardware requirements

Implementation of the control design outlined in the previous section will require significant hardware on the roadside and in the vehicle to collect or communicate information.

Roadside monitors. They measure traffic conditions such as flow and speed. Based on these data the link layer calculates a path for each vehicle and the target platoon size and speed, and communicates them to the vehicle. Traffic measurements may be made by loop detectors, ultrasonic sensors (used in Japan), or vision systems. Information may be communicated to vehicles by infra-red beacons, by variable message signs placed on the road. or may be broadcast by radio.

Sensors. Vehicles must be equipped with a longitudinal sensor that measures the relative distance and speed between itself and the vehicle in front of it. Such sensors may be based on radar, ultrasonics (as in a Polaroid camera), or vision. In order to change lanes the vehicle must be equipped with sensors that locate vehicles on the side within a range of about 30 m. The regulation layer also needs the vehicle's position and ground speed (which could be obtained in different ways) and several measurements of the state of the vehicle for which several sensors are available.

Inter-vehicle communication. The planning layer requires the ability to communicate with neighboring vehicles within a range of about 60 m. Such communication links need to be reliable, and incur very small delay (about 20 ms). Again, various solutions may be proposed including broadcast or cellular radio, infrared beacons, communication through a roadside station. There is significant progress in the development of systems for communicating routing and navigation data, but these systems do not meet the delay constraints needed for real time control.

6 Experimental results at Berkeley

A very simple two vehicle platoon control system has been implemented. Details are available in [34, 35]. The control system components include:
(1) A radar system made by VORAD [36] to measure relative distance and speed;
(2) Sensors on the vehicle to measure speed, acceleration, throttle angle, brake pressure, engine speed: intake manifold mass flow rate and intake manifold temperature;
(3) Throttle and brake actuation system;
(4) A 80386-based PC equipped with a data acquisition board to interface with the sensors and actuators and a communication board to interface with the radio;
(5) Digital radio transceiver made by PROXIM operating at 122 kbps.

The vehicles are manually steered since they lack a steering actuator. Only the "follower spacing control" law (see §4.2) has been implemented and tested, and the results are as expected and encouraging. In particular, the follower's position can be kept to within a few cms. This indicates that maintaining an intra,-platoon spacing of 1 m is a realistic objective. The next step will involve working with four vehicles. This step will

allow. in addition, the design and testing of the "accelerate to merge" and the "decelerate to split" laws. A simplified planning layer with these protocols will also be implemented. Experiments with lateral control use a vehicle with a steering actuator (but no throttle or braking actuators) [37]. Longitudinal and lateral control will be integrated when vehicles with throttle, braking and steering actuators become available.

We note that control design is only one element in this experimental effort. A lot of work and diverse skills are needed for hardware, software and system integration.

7 Summary

In this section we comment on the state of IVHS research.

Prometheus and Drive are the two major West European programs devoted to IVHS. (In Europe IVHS is called RTI for Road Transport Informatics.) Funded at several hundred million dollars, Prometheus is run by the major European auto and electronics companies. Unfortunately, its reports are not public. Academic and research institutions are more active participants in Drive. A useful summary of work in Drive is available in a recent conference proceedings [38]. Individual reports cited in that proceedings are not readily available.[17]

It may be no surprise to learn that for the past twenty years Japan has been engaged in a long-term effort in IVHS, undertaken by auto makers and sometimes coordinated by MITI. Demonstrations of traffic management systems and driver navigation aids have led to improvements in highway infrastructure and automobile products. Noteworthy are projects aimed at determining the feasibility of 'full automation' [39]. These projects integrate a variety of communication, computing and control technologies into a working system. The long-term, focused effort is impressive.

By contrast, IVHS research in the U.S. has been sporadic, scattered, subject to the stop-and-go of federal funding.[18] The situation seems to be changing. There is widespread expectation of more and, one hopes, sustained funding. The recent founding of the IVHS Society of America, with congressional blessing, offers the promise of creating a national forum for the exchange of information and opinion and for some degree of coordination of effort. Because its members represent a broad cross-section of auto and electronics companies? transportation agencies. highway users, and academic institutions, the Society has the potential to influence significantly IVHS research in the U.S.

It is customary to divide work in IVHS into four groups: ATMS (Advanced Traffic

''Additional references may be useful. The Proceedings of the annual Vehicle Navigation and Information Systems (VNIS) Conference and International Symposium on Automotive Technology and Automation (ISATA) give a good sense of mainstream **work.** A useful introduction to IVHS can be obtained from the February 1991 issue of the IEEE Transactions on Vehicular Technology. Finally, recent annual ACC meetings have sessions on IVHS.

[18]The lack of consistent attention to IVHS may not be entirely irrational. Because land has been plentiful until recently. funds have been devoted to increasing highways rather than making them more efficient. In Japan. land has been scarce for a long time and IVHS research began earlier.

Management System), ADIS (Advanced Driver Information System), AVCS (Advanced Vehicle Control Systemj, and CVO (Commercial Vehicle Operationsj.[19] This classification is meaningful when we consider non-core IVHS functions. For instance, "automatic toll collection" is an ATMS function because transportation authorities are usually in charge of tolls; similarly, "scheduling of delivery trucks" is naturally regarded as a CVO function.

However, when we consider the core IVHS functions, this classification can be misleading as these functions will appear in each group. For example, "route guidance" is often classified an ATMS function, "travel time information" to assist route selection is an ADIS function, and "route assignment" is an AVCS function. From this example we see that the distinction between ATMS, ADIS and AVCS cannot be one based on differences in their core functions. Rather, the distinction concerns the 'variables' that are used to influence driver decisions: ATMS relies on 'advice', ADIS on 'information' and AVCS on 'control', see Table 1. From a control system-theoretic perspective it is misleading to regard ATMS, ADIS and AVCS as subsystems of an overall IVHS system, as is often suggested. The significant factor for control system design is the degree of influence that can be exercised over driver decisions: the greater this degree, the greater is the burden on system 'intelligence' and the more predictable is the resulting traffic behavior.

We conclude with some remarks on IVHS design.

The process of IVHS design can be schematically described as involving the following steps.

1. *Functional specification* is a list of functions that IVHS will support. Central to any list is a core set of driver decisions targeted for improvement. However, numerous other functions dealing with highway operations or driver needs may be included. Typical examples are toll collection: emergency vehicles operations, incident detection: parking reservations. Any functional specification must emerge from some vision of how an IVHS system would look as a whole and how it will interface with the rest of the transportation system. §2 gave a very basic functional specification, while §3 sketched one IVHS scenario.

2. *Control system architecture* specification requires (1) the translation of functions into tangible control tasks, (2) the assignment of these tasks to individual subsystems together with the information available to carry out those tasks, and (3) the specification of system interfaces and subsystem interconnections. The architecture should be extendable to allow for new functions. An architecture (dealing with a few core functions) was outlined in §4.

3. *Control* system *design* involves (1) the assignment of subsystems to controllers on the vehicle and on the roadside, and (2) designing each subsystem to carry out the assigned tasks. We have proposed that the network and link layers in Figure 4 should be assigned to the roadside while the planning and regulation layers should be assigned to the vehicle. The design of each system will depend on the task to be performed. We proposed a discrete event controller for the planning layer (§4.1) and a set of feedback laws for the regulation layer (§4.2).

[19] For example. the Special Issue on IVHS of the Transactions on Vehicular Technology is divided under these headings.

4. *Physical design* involves specification of the hardware and software of the control system. §6 gave a very brief summary of such a design for a simple case.

5. *Communication, system design* is an essential component of the IVHS system since the control is distributed between roadside and the vehicles. The design process will require specification of a logical communication architecture, a communication system design and a physical implementation. **A** hint of the requirements was given in §5.

The transportation system has a major impact on society. A qualitative change in that system as envisaged by IVHS advocates will, in the long run, significantly affect our daily lives, the spatial organization of work, and the environment. The challenge of IVHS design is formidable. Transportation engineering as traditionally defined cannot by itself meet this challenge. Neither can control theory by itself. A socially responsive approach will require the integration of several disciplines.

8 Acknowledgments

This paper is partly based on work conducted by several people over the last three years as part of the PATH program of the University of California, Berkeley, in cooperation with the State of California Department of Transportation and the United States Federal Highway Administration. In particular? I have made liberal use of the results reported in [40, 23, 28. 41. 29. **42,** 33, *37.* 34]. It is a pleasure, also, to acknowledge very fruitful discussions with Steve Shladover, Kwang-Soo Chang, Bob Kurshan, Ann Hsu, Farokh Eskafi, and members of the PATH seminar on IVHS. The opinions expressed here are my own and do not necessarily reflect their views or the official view or policies of the State of California.

References

[1] F. Kobayashi, "Feasibility study of route guidance," *Transportation Research Record,* vol. 737, pp. 107–112, 1979.

[2] D. J. Jeffrey, "Route guidance and in-vehicle information systems," in *Information Technology Applications in Transport* (P. Bonsall and M. Bell, eds.), **VNU** Science Press, Utrecht, Netherlands, 1987.

[3] J. Sparmann, "LISB route guidance and information system: First results of the field trial," in *Proceedings* of *the Vehicle Navigation and Information Systems Conference,* (Toronto, Canada). September 11-13 1989.

[4] H. Al-Deek, M. Martello, A. May, and W. Sanders, "Potential benefits of in-vehicle information systems in a real freeway corridor under recurring and incident induced congestion," tech. rep., UCB-ITS-PRR-88-2, Institute of Transportation Studies, University of California, Berkeley, CA 94720, 1988.

[5] H. Al-Deek and A. May, "Potential benefits of in-vehicle information systems (IVIS): Demand and incident sensitivity analysis," tech. rep., UCB-ITS-PRR-89-1, Institute of Transportation Studies, University of California, Berkeley, CA 94720, 1989.

[6] H. Al-Deek and A. Kanafani, "Some theoretical aspects of the benefits of en-route vehicle guidance," tech. rep., UCB-ITS-PRR-89-2, Institute of Transportation Studies, University of California, Berkeley, CA 94720, 1989.

[7] N. Hounsell, M. McDonald, and L. Breheret, "The modelling of dynamic route guidance." in *Advanced Telematics in Road Guidance. Proceedings of the DRIVE Conference,* pp. 89–98, Elsevier, February 4-9 1991.

[8] U. Karaaslan, P. Varaiya, and J. Walrand, "Two proposals to improve freeway traffic flow,'? in *Proceedings of the 1991 American Control Conference,* (Boston, MA), pp. 2539–2544, June 26-28 1991.

[9] S. A. Smulders, "Control of freeway traffic flow," tech. rep., Center for Mathematics and Computer Science, Report OS-R.8806, Amsterdam, 1988.

[10] J. Bender, "An overview of systems studies of automated highway systems," *IEEE Transnctions on Vehicular Technology,* vol. 40, pp. 82–99, February 1991.

[11] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 10, pp. 362–373, May 1988.

[12] A. M. Waxman, J. LeMoigne, L. Davis, B. Srinivasan, T. Kushner, E. Liung, and T. Siddalingaiah. "A visual navigation system for autonomous land vehicles," *IEEE Journal of Robotics and Automation,* vol. 3, pp. 124–140, April 1987.

[13] M. Markarinec, *An accident avoidance system for an autonomous highway vehicle.* PhD thesis, Northwestern University, 1989.

[14] E. Freund. Ch. Buhler, U. Judaschke, B. Lammen, and R. Mayr, "A hierarchically structured system for automated vehicle guidance,'' in *22nd International Symposium on Automotive Technology and Automation, Vol 1,* (Automotive Automation Ltd., Croyden, England), pp. 351–359,1990.

[15] A. Niehaus and R.. Stengel, "Probability-based decsion making for automated highway," in *Proceedings of the Vehicle Navigation and Information Systems Conference,* (Dearborn, MI), pp. 1125–1136, October 20-23 1991.

[16] A. Gollu and P. Varaiya, "Hybrid dynamical systems,'' in *Proceedings of the 28th Conference on Decision and Control,* (Tampa, FL), pp. 2708–2712, December 1989.

[17] *IEEE Control Systems Magazine: Special Issue on Intelligent Control,* vol. 11. June 1991.

[18] P. Varaiya and A. Kurzhanski, eds., *Discrete Event Systems: Models and Applications,* vol. IIASA **103,** Lecture Notes in Control and Information Sciences. Springer. 1988.

[19] *Proceedings of the IEEE: Special Issue on Dynamics of Discrete Event Systems*, vol. 77. January 1989.

[20] Z. Har'El and R. P. Kurshan, *COSPAN User's Guide*. AT&T Bell Laboratories, Murray Hill, NJ, 1987.

[21] Z. Har'El and R. P. Kurshan, "Software for analytical development of communications protocols," *AT&T Technical Journal*, pp. 45–59, January/February 1990.

[22] G. Holzmann, *Design and Validation of Computer Protocols*. Prentice Hall, 1991.

[23] A. Hsu, S. Sachs, F. Eskafi, and P. Varaiya, "The design of platoon maneuvers protocols for IVHS," tech. rep., UCB-ITS-PRR-91-6, Institute of Transporation Studies, University of California, Berkeley, CA 94720, April 1991.

[24] E. Clarke, E. Emerson, and A. Sistla, "Automatic verification of finite-state concurrent systems," *ACM Transactions on Programming Languages and Systems*, vol. 8, pp. 244–263, April 1986.

[25] R. Kurshan, "Reducibility in analysis of coordination," *Lecture Notes in Control and Information Sciences*, vol. 103, pp. 19–39, 1989.

[26] S. Tsugawa and S. Murata, "Velocity control for vehicle following through vehicle/vehicle communication," in *22nd International Symposium on Automotive Technology and Automation, Vol 1*, (Automotive Automation Ltd., Croyden, England), pp. 343–350, 1990.

[27] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," tech. rep., UCB-ITS-PRR-89-3,6, Institute of Transportation Studies, University of California. Berkeley. CA 94720, 1989.

[28] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *Proceedings of the 1990 American Control Conference, Volume 1*, (San Diego, CA), pp. 291–296, June 1990.

[29] D. McMahon, J. Hedrick, and S. Shladover, "Vehicle modeling and control for automated highway systems," in *Proceedings of 1990 American Control Conference*, (San Diego, CA), pp. 297–303, 1990.

[30] H. Peng and M. Tomizuka, "Lateral control of front-wheel-steering rubber-tire vehicles," tech. rep., UCB-ITS-PRR-90-5, Institute of Transportation Studies, University of California, Berkeley, CA 94720, 1990.

[31] S. Sheikholeslam and C. A. Desoer, "Combined longitudinal and lateral control of a platoon of vehicles: A system-level study," tech. rep., PATH Technical Memo 91-3, Institute of Transportation Studies, University of California, Berkeley, CA 94720. September 16, 1991.

[32] F. Broqua, G. Lerner, V. Mauro, and E. Morello, "Cooperative driving: Basic concepts and a first assessment of "intelligent cruise control" strategies," in *Advanced Telematics in Rood Guidance. Proceedings* of *the DRIVE Conference*, pp. 908–929, Elsevier, February 4-9 1991.

[33] H. Peng and M. Tomizuka, "Preview control for vehicle lateral guidance in highway automation," in *Proceedings of the 1991 American Control Conference,* [Boston, **MA),** June 26-28 1991.

[34] K. Chang, W. Li, **A.** Shaikhbahai, and P. Varaiya, "A preliminary implementation for vehicle platoon control system," in *Proceedings* of *the 1991 American Control Conference,* (Boston, **MA),** pp. 3078–3083, June 26-28 1991.

[35] K. Chang, W. Li, P. Devlin, **A.** Shaikhbahai, P. Varaiya, J. Hedrick, D. MacMahon, V. Narendran, and D. Swaroop, "Experimenation with a vehicle platoon control system." in *Proceedings* of *the Vehicle Navigation and Information, Systems Conference!* (Dearborn. MI), pp. 1117–1124, October 20-23 1991.

[36] J. Davis, "Adapting radar to the automotive environment." Technical Report, Radar Control Systems Corp. (VORAD), May 1987.

[37] T. Hessburg. H. Peng, M. Tomizuka, W. Zhang, and E. Kamei, "An experimental study on lateral control of a vehicle." in *Proceedings* of *the 1991 American Control Conference*, (Boston, **MA),** June 26-28 1991.

[38] *Advanced Telematics in Road Guidance. Proceedings* of *the DRIVE Conference.* Elsevier. 1991.

[39] S. Tsugawa, H. Watanabe, and H. Fujii, "Super Smart Vehicle System – Its concept and preliminary works," in *Proceedings* of *the Vehicle Navigation and Information Systems Conference*, (Dearborn, MI), pp. 269–277, October 20-23 1991.

[40] P. Varaiya and S. Shladover, "Sketch of an IVHS systems architecture," tech. rep., UCB-ITS-PRR-91-3, Institute of Transportation Studies, University of California, Berkeley, CA 94720, February 1991.

[41] S. Sheikholeslam and C. **A.** Desoer, "Longitudinal control of a platoon of vehicles with no communication of lead vehicle information," in *Proceedings* of *the 1991 American Control Conference, Volume 3,* (Boston, **MA),** pp. 3102–3107, June 1991.

[42] S. Shladover, C. Desoer, J. Hedrick, M. Tomizuka, J. Walrand, W. Zhang, D. McMahon, **H.** Peng, S. Sheikholeslam, and **N.** McKeown, "Automated vehicle control developments in the PATH program," *IEEE Transactions on Vehicular Technology,* vol. 40, pp. 114–130, February 1991.

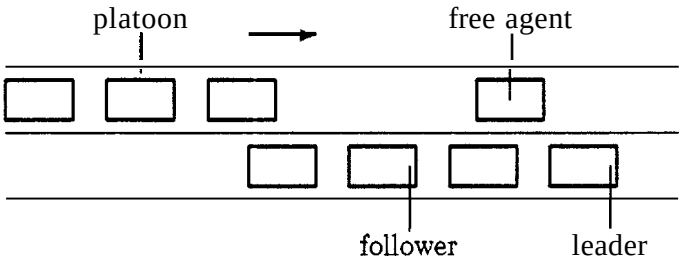

Figure 1: The platoon concept

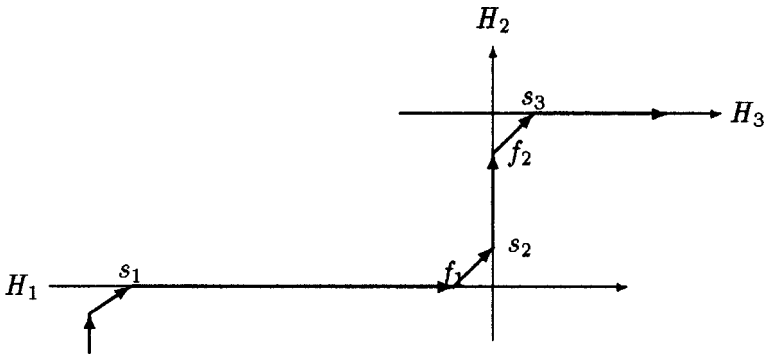

Figure 2: A route is a sequence of segments

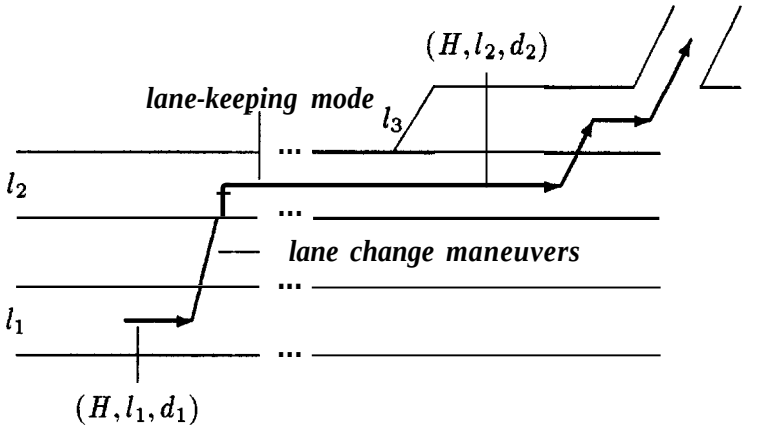

Figure 3: A vehicle's plan

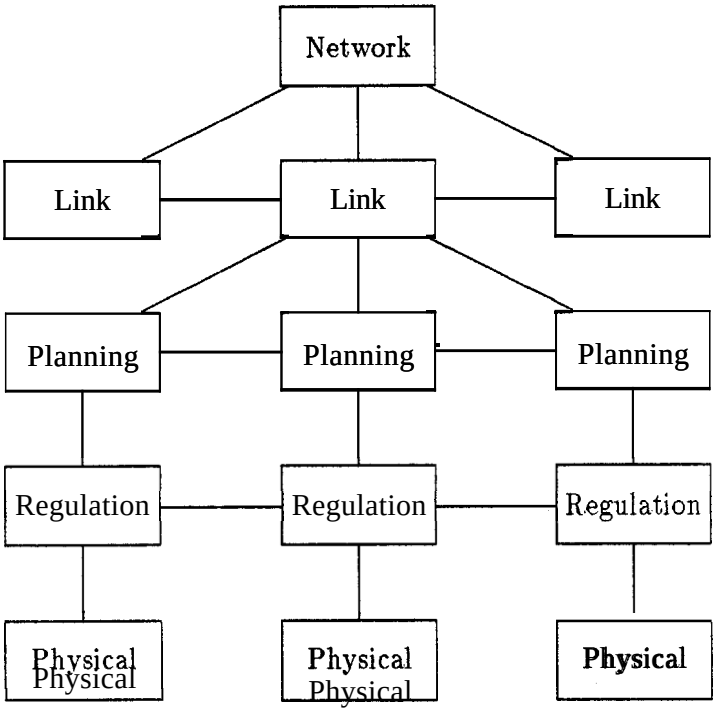

Figure **4:** Control system architecture

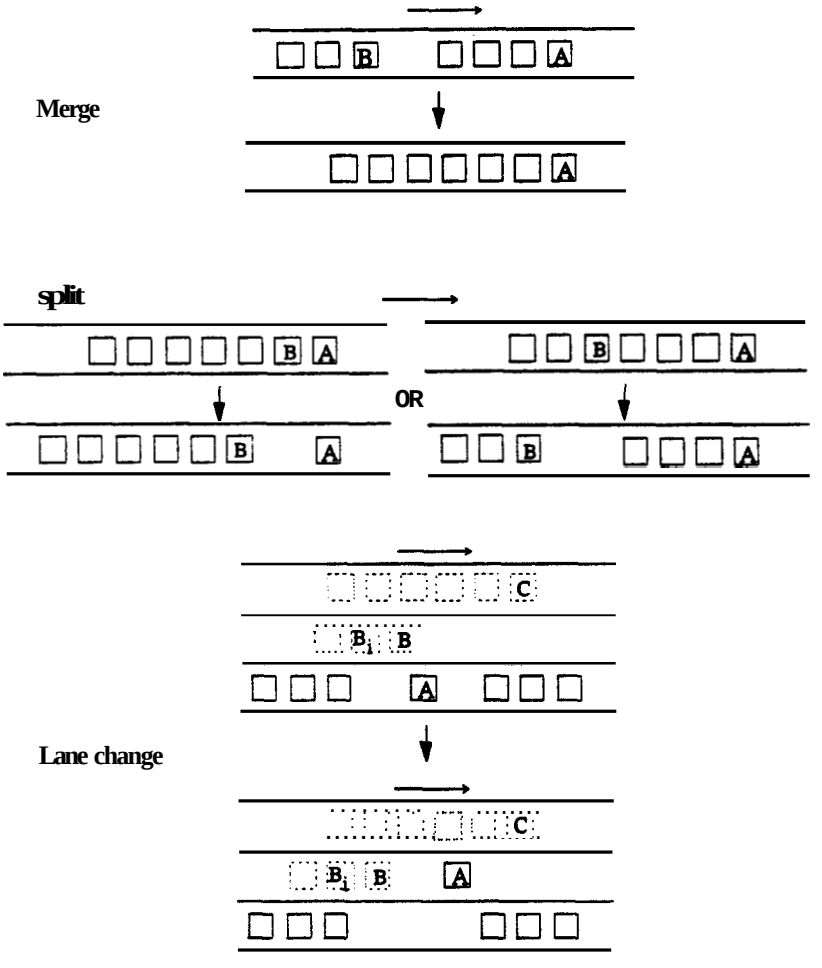

Figure 5: Three maneuvers

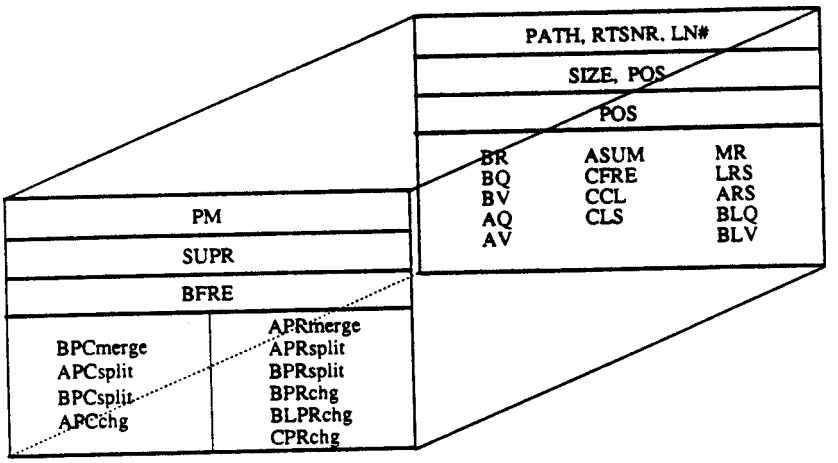

Figure *6:* Planning layer state machines

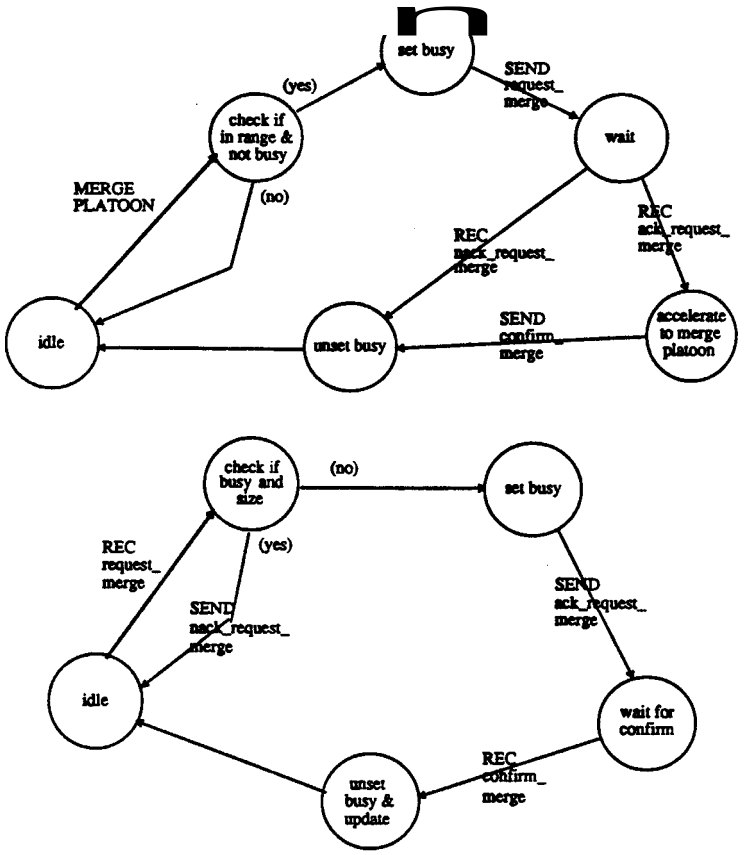

Figure 7: State machines for **merge**

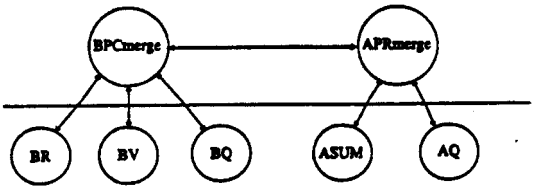

Figure 8: **Machines** for **merge** protocol **and** its **environment**

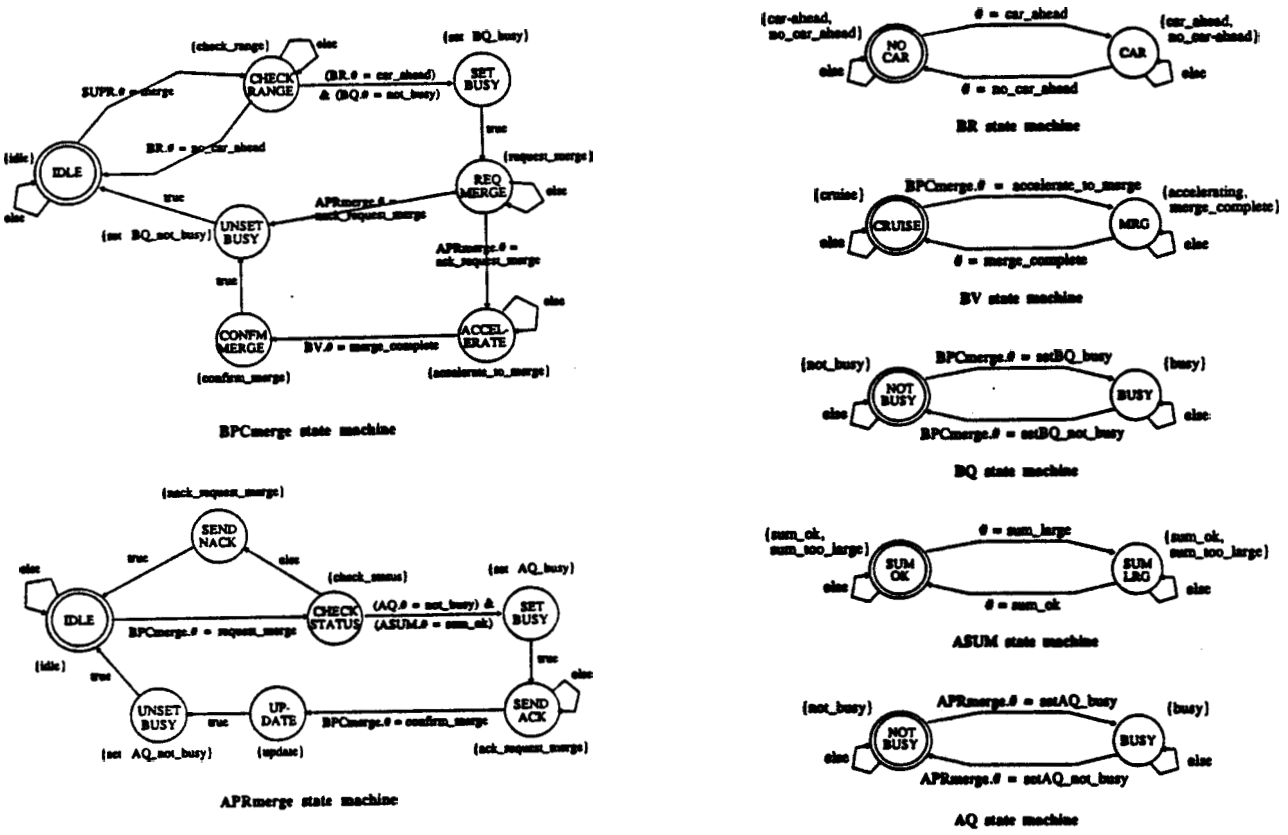

Figure 9: COSPAN specification of the merge protocol

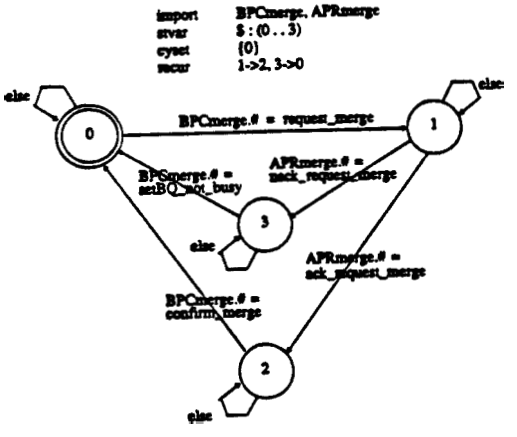

Figure 10: Merge monitor