Title
Design Of An Extended Architecture For Degraded Modes Of Operation Of AHS

Authors
Lygeros, John
Godbole, Datta N.
Broucke, Mireille E.

Publication Date
1995

CALIFORNIA PATH PROGRAM
INSTITUTE OF TRANSPORTATION STUDIES
UNIVERSITY OF CALIFORNIA, BERKELEY

Design of an Extended Architecture for Degraded Modes of Operation of AHS

John Lygeros, Datta N. Godbole, Mireille E. Broucke

California PATH Working Paper
UCB-ITS-PWP-95-3

This work was performed as part of the California PATH Program of the University of California, in cooperation with the State of California Business, Transportation, and Housing Agency, Department of Transportation; and the United States Department Transportation, Federal Highway Administration.

Report for MOU 135

April 1995

ISSN 1055-1417

Design of an Extended Architecture for Degraded Modes of Operation of AHS *

John Lygeros, Datta N. Godbole
Intelligent Machines and Robotics Laboratory
Department of Electrical Engineering and Computer Sciences
University of California,Berkeley, CA 94720

Mireille E. Broucke
California PATH, University of California,Berkeley
Richmond Field Station, Richmond, CA 94804

lygeros, godbole, mire@robotics.eecs.berkeley.edu

Abstract

We propose a hierarchical control architecture for dealing with faults and adverse environmental conditions on an Automated Highway System (AHS). Our design builds on a previously designed control architecture that works under normal conditions of operation. The faults that are considered in our design are classified according to capabilities remaining on the vehicle or roadside after the fault has occurred. Information about these capabilities is used by supervisors in each of the layers to select appropriate control strategies. We outline the extended control strategies that are needed by these supervisors of each layer of the hierarchy and, in certain cases, give examples of their detailed operation.

Keywords: Automated Highway Systems, malfunctions, adverse environment, fault tolerant design, safety

1 Introduction

Intelligent Vehicle Highway Systems (IVHS) has been an active research area within the California PATH[1] project for the past several years. The objective is to come up with an Automated Highway System (AHS) design that will significantly increase safety and highway capacity by adding intelligence to both the vehicle and the roadside, without having to

*Research supported by the PATH program, Institute of Transportation Studies, University of California, Berkeley under MOU 135

[1]PATH stands for Partners for Advanced Transit and Highways

build new roads. Several approaches to this problem have been proposed within the PATH project, ranging from Autonomous Intelligent Cruise Control (where the driver is in control of vehicle steering) to full automation. An underlying assumption in most of these designs has been that operation will take place under normal conditions. The definition of "normal" may vary from case to case, but, in general, it means benign environmental conditions and faultless operation of all the hardware, both on the vehicles and on the roadside. Some studies to deal with "abnormal" conditions have been made (for example [2, 3, 4]), but they have been in the form of specific faults rather than a general framework. Our goal is to propose an AHS design that will perform well under almost any condition. The only abnormal conditions of operation that we will ignore are faults in the design (e.g. a deadlock in the protocols) and in the implementation of the software, since we assume the design will be verified before implementation. Even with this restriction it is clear that the task is rather large. In this report we will only give an overview of what is involved and establish a framework for tackling the problem. The framework will partition the task into more manageable parts and formalize the requirements that each of them will need to satisfy.

Our framework will be modeled on the control architecture proposed in [1, 5] for normal modes of operation. Before presenting the framework we will give a brief overview of this design to fix the terminology and notation. The main idea is that the control problem posed by the AHS is too large to be dealt with by means of a single controller. Therefore a hierarchical control structure is introduced.

The design of the control hierarchy outlined in [1] centers around the notion of "platooning". It is assumed that traffic on the highway is organized in groups of tightly spaced vehicles, called platoons. Intuition suggests that doing this should lead to an increase in the capacity and throughput of the highway; indeed theoretical studies indicate that the capacity increase if such a scheme is implemented successfully will be substantial (as high as four times the current capacity). Moreover, this will be achieved without a negative impact on passenger safety. By having the vehicles within a platoon follow each other with a small intra-platoon separation (about 1 meter), we guarantee that if there is a failure and an impact is unavoidable, the relative speed of the vehicles involved in the collision will be small, hence the damage to the vehicles and the injuries to the passengers will be minimized. The inter-platoon separation, on the other hand, is large (of the order of 30 meters) to isolate the platoons from each other. The idea behind this is that, if needed, the platoons will have enough time to come to a stop before they collide. In addition a large separation guarantees that transient decelerations will be attenuated as they propagate down the freeway. Clearly implementation of such a scheme will require automatic control of vehicles, as human drivers are not fast and reliable enough to produce the kinds of inputs necessary for forming platoons. In the architecture outlined in [1] the system is organized in five layers. The block diagram of figure 1 shows different layers of the control hierarchy.

The top layer, called the **network layer**, is responsible for the flow of traffic on the entire highway system[2]. Its task is to prevent congestion and maximize throughput by dynamic routing of traffic. The second layer, called the **link layer**, coordinates the operation of

[2]The highway system might consist of interconnection of several highways around an urban metropolis.

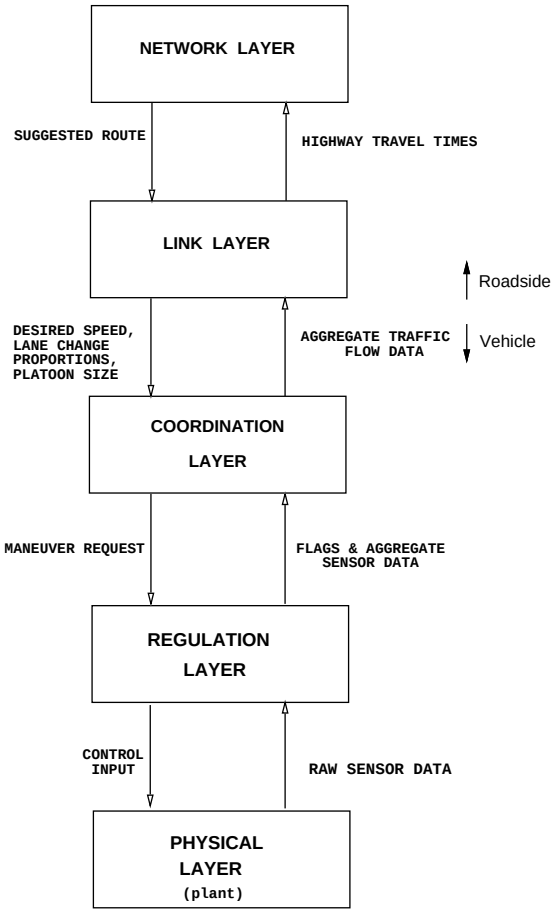

Figure 1: IVHS Architecture

whole sections (links) of the highway and operates at the roadside. Its primary concern is to maximize throughput while maintaining safe conditions of operation. With these criteria in mind, it calculates an optimum platoon size and an optimum velocity for each highway section. It also decides which lanes the vehicles should follow to get to their destination as fast as possible. Finally, it monitors incidents on the highway and diverts traffic in order to minimize the impact of the incident on traffic flow and safety. Because the link layer bases its control actions on large numbers of vehicles, it treats the vehicles in a section in an aggregate manner rather than considering the state of individual vehicles or platoons. The commands it issues are not addressed to individual vehicles but to all the vehicles in a section; a typical command would be "30% of the vehicles who wish to get off the highway at the next exit should change lane now" or "all platoons in this section should try to be 10 vehicles long". A possible design for the link layer is described in [2]. It is based on a "fluid flow" traffic model similar to the ones developed for manual traffic.

The next level of hierarchy below the link layer is the **coordination layer**. It resides in each vehicle and its task is to coordinate the operation of platoons with their neighbors. It receives the link layer commands and translates them to specific maneuvers that the

platoons need to carry out. For example, it will ask two platoons to join to form a single platoon whose size is closer to the optimum or, given a command like "30% of the vehicles going to the next exit change lane now", it will decide which vehicles will comprise this 30% and split the platoons accordingly in order to let them out. The current design [6] uses protocols, in the form of finite state machines, to organize the maneuvers in a systematic way. They receive the commands of the link layer and aggregated sensor information from the individual vehicles (of the form "there is a vehicle in the adjacent lane"). They then use this information to decide on a control policy and issue commands to the regulation layer. The commands are typically of the form "accelerate to join the preceding platoon" or "decelerate so that another vehicle may move into your lane ahead of you".

Below the coordination layer in the control hierarchy lies the **regulation layer**. Its task is to receive the coordination layer commands and translate them to throttle, steering and braking input for the actuators on the vehicle. For this purpose it utilizes a number of continuous time feedback control laws ([7, 8, 9, 10]) that use the readings provided by the sensors to calculate the actuator inputs required for a particular maneuver. The regulation layer communicates with the coordination layer to inform it of the outcome of the maneuver.

The bottom layer is not part of the control hierarchy. It is called the **physical layer** and it contains the actual plant (in this case the vehicles with their sensors, actuators and communication equipment and the highway topology).

The references given above describe models and control strategies that fulfill the requirements set for all the layers in this architecture. All of these laws have been verified theoretically and tested in simulation and, in some cases, in experiments. They have proven to perform well, mostly under the assumptions that the conditions of operation are "normal" (in the sense discussed above). Some laws also exist for operation under degraded conditions. For example [2] contains a link layer control law to divert traffic when a lane is closed (because of an accident for example). [3] describes a regulation layer control law for steering in case of a tire burst. Our goal here is to encompass all these laws (for normal and degraded conditions of operation) into a general framework. To complete the framework it will be necessary to design a number of new control laws to supplement the existing ones.

2 Architecture for degraded modes of operation

As discussed in the introduction, the goal is to design an extended hierarchical control system that will guide the AHS in some optimal manner under normal and fault conditions. We will define what is meant by "optimal" and "normal and fault conditions". Also, it is useful to emphasize the magnitude of the design problem and why one is led to chosing a hierarchical control structure. To control all vehicles on a length of highway of say, 50km, where there may be several simultaneous faults, requires masses of information to be processed and commands to be issued. As in the normal mode, the solution requires a hierarchical (multi-agent) control structure, treating the vehicles as semi-autonomous agents. The levels of the extended hierarchy will be the same as the levels of the hierarchy described in the introduction. In particular, more abstract plant descriptions will be used

as we move up the hierarchy.

2.1 Information Flow

In order to come up with meaningful control laws in a given situation our controller will need information about what the situation is. In designing the normal mode of operation it was assumed that the capabilities of all the vehicles and the freeway are fixed and known a priori. Therefore, the only information the normal mode controller needs to know, in order to come up with a control law, is the current state of the system. Because of the hierarchical structure of the normal mode, the flow of information about the state of the system was also arranged in a hierarchy; that is the higher levels of the architecture receive more abstract information than the lower levels.

In extending the hierarchy to deal with degraded modes of operation we need to consider the additional complications that arise from the fact that the system capability is not fixed. We partition the factors that affect the capability into two classes. The first class contains all the faults that may develop on the vehicle or the roadside. The performance degradation that arises because of these factors is in a sense discrete, as we will assume that these faults take place instantaneously and therefore change the system capabilities in steps. To simplify the design of the architecture, we will also assume that faults are irreversible, i.e. once a fault occurs it can not be fixed by itself. The second class contains factors that lead to gradual degradation of performance (for example adverse weather conditions such as rain or fog, brake wear etc). To put it in other words, the factors in the first class determine what the system can do while those in the second determine how well it can do it.

Summarizing the above discussion, an extended architecture that will be capable of dealing with both normal and degraded conditions of operation will need information about three aspects of the system behavior:

- The current state
- quantitative capabilities (what the system can do)
- qualitative capabilities (how well it can do it)

The flow of information for all three of these things will be arranged in a hierarchy, parallel to the control hierarchy. Therefore the overall system architecture will look like the block diagram in figure 2. There will be three hierarchical structures monitoring the behavior of the plant. The **Sensor Structure** carries the information about the current state of the system, the **Quantitative Capability Structure** carries the information about what the plant is capable of doing and the **Quantitative Capability Structure** carries the information about how well it can do it. The loop is closed by the **Control Structure** that will use all this information to produce control inputs to the plant.

In order to produce a complete design one needs to specify in detail the three branches of the information structure and the Control Structure. We will not describe the sensor structure[3]

[3]Details of the sensor structure can be found in [11].

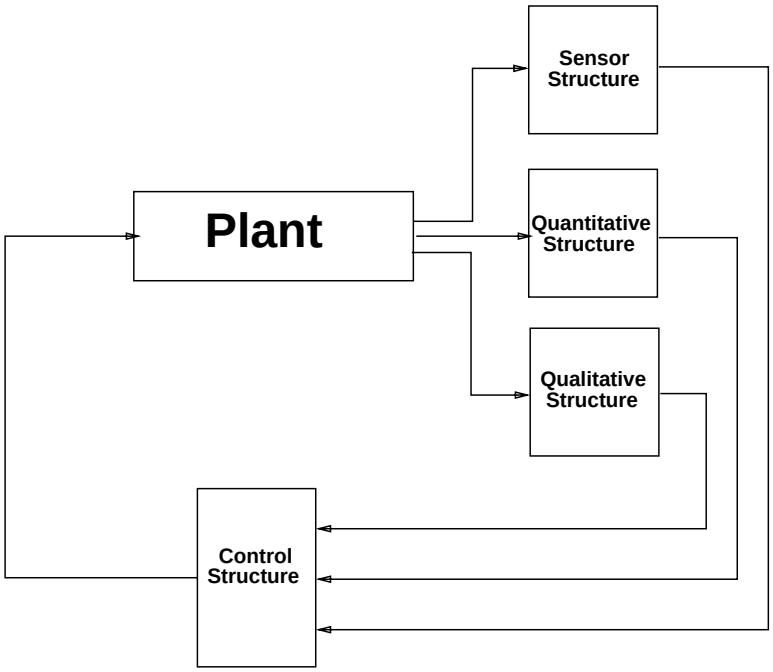

Figure 2: Overview of the Supervision Problem

but will give a high level description of the other two structures. In section 3.2, we will propose a possible design for the Quantitative Capability Structure and give an outline of a design and a reference to a more thorough treatment for the Qualitative Structure. In the next section we will outline the main features of the Control Structure.

2.2 Control Flow

As already mentioned the design of the controller should be optimal in some sense. The controller should receive the information provided by the Sensor, Quantitative and Qualitative Capability Structures and use it to produce inputs for the vehicle actuators (throttle, brake and steering angle) so that a certain global cost criterion will be minimized. The cost criterion should reflect safety, capacity and environmental impact. As explained above, due to the size of the problem, a hierarchical supervisor will be introduced. Even though this is essential to keep the design tractable, it makes the task of global optimization a lot harder. Indeed it is highly unlikely that the hierarchical design will be optimal for any meaningful global cost function.

To keep the idea of an optimal solution alive we need to introduce some notion of layered optimality. At each level of the hierarchy a different cost function will be used to decide on the optimal actions. All these cost functions should reflect capacity and safety maximization in the descriptive language of the layer in question. The structure of the i^{th} level of the controller in this context is shown in figure 3. It should be noted here that by design the higher levels of the hierarchy have access to information about a larger part of the system.

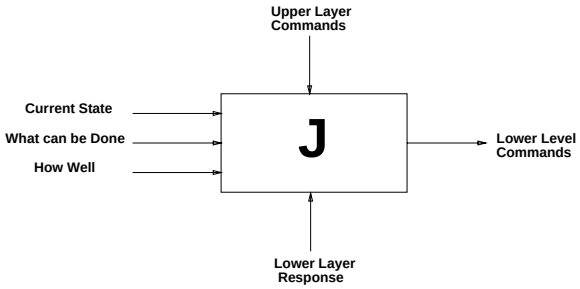

Figure 3: Inputs and Outputs to each level of the Controller

Therefore they are better suited to assess the overall capacity. Lower levels on the other hand have access to more detailed information, therefore are better suited to assess safety. As we will assume that the prime concern of the design should be safety, we can think of introducing the higher level commands simply as a bias in the lower level cost function. Then, if a lower level detects that a command is unsafe, it will not implement it and notify the higher layer of this fact.

The task of designing a controller that is optimal at each layer is difficult. Formulating acceptable cost criteria that will capture safety and capacity in the various descriptive languages will be needed. In the first pass of the design we will forgo optimization by restricting ourselves to a fixed set of control strategies that have been optimized by simple performance constraints such as acceleration limits and minimum time maneuvers. Later the cost functions will be fully specified and the design will be optimized by layers, though, as we have observed, global optimization for criteria like safety and capacity are currently beyond the scope of our design methodology.

Thus, according to this structure, our controller at each level of the hierarchy will consist of a number of control laws and a *supervisor* to pick the strategy that minimizes the cost function for that level.

3 Fault handling

In the control scheme of [1], sensors, actuators and communication equipment will be added to the vehicles and infrastructure for automatic control of vehicles. The additional instrumentation along with the basic mechanical (moving) parts of the vehicle are prone to failure. A comprehensive list of faults, considered in this study, is given in the Appendix. A fault in either vehicle or infrastructure will reduce the capability of the system to operate in "normal mode". This will directly influence the control laws that can be designed for faulted condition. Thus monitoring the capabilities of the system is one of the basic tools for fault handling. In the next section, we present a way of modeling the capabilities of the controller. This information will then be used in section 3.2 for fault classification.

3.1 Quantitative Capability Structure

The control scheme for normal operating conditions presented in [1] relies on a number of sensors, actuators and communication devices, both on the vehicles and on the roadside. All this additional hardware as well as the standard mechanical parts of the vehicles are prone to failure. Such a failure, in either the vehicle or the infrastructure will directly influence the capabilities of the system as a whole and therefore restrict the controls that the supervisor can implement.

To monitor the capability of the system we propose a design based on a hierarchy of predicates. Each predicate will monitor one capability and will return a 1 (True) if the system possesses the capability in question or a 0 (False) otherwise. The predicates will be arranged in a hierarchy similar to that of the normal mode supervisor. The values returned by the higher level predicates will depend on the values of the lower level predicates. This scheme can be used to systematically go through combinations of faults and design specialized control laws that utilize the remaining capabilities so that the impact of the faults on the system is minimized in each case. We will start describing this hierarchy at the bottom and work our way up.

3.1.1 Physical Layer Predicates

The supervisor structure assumes that the vehicles and the roadside have access to certain resources, namely sensors, actuators and communication devices. We model each one of these resources as a predicate, that returns 1 if the resource is available and functioning and 0 otherwise. Assuming that the supervisor requires n_a actuators, n_s sensors and n_c communication devices, the quantitative capability of the physical layer can be expressed as a vector of zeros and ones of dimension $n_s + n_a + n_c$:

$$\{0,1\}^{n_s+n_a+n_c}$$

This vector reflects which resources are functioning and which are not. It should be noted that for simplicity the actuator predicates are interpreted as reflecting the capability of the vehicle to accelerate, decelerate and turn. Therefore they incorporate information about basic vehicle functionality, like engine and tires being in proper working order, enough fuel, etc. Predicates for these basic functionalities can explicitly be added at the cost of a small increase in the complexity of the design.

3.1.2 Regulation Layer Predicates

The regulation layer contains a number of control laws, both longitudinal and lateral. Each one of these laws makes use of a number of physical layer resources, primarily sensors and actuators. For a regulation layer controller to be functional all of these resources need to be available. Therefore the applicability of a regulation layer controller can be modeled by a predicate whose value depends on the values of the predicates for the physical layer.

Consider, for example, the Autonomous Intelligent Cruise Controller proposed in [10] as the default law for the leader of a platoon. This longitudinal control law uses sensor readings

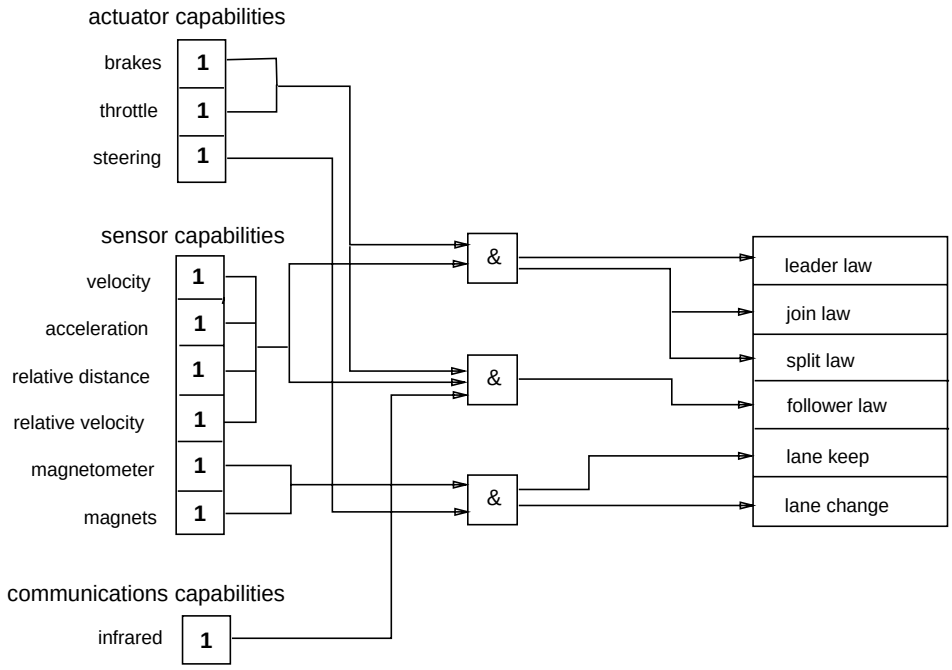

Figure 4: Connection between Physical and Regulation layer capabilities

of velocity and acceleration of the vehicle, as well as readings of the spacing and relative velocity of the preceding vehicle to calculate values for the throttle and and brake inputs. Without getting into the details of the control law, we can see that the lead controller predicate can be viewed as an *AND* predicate on the values returned by the predicates for the velocity, acceleration, spacing and relative velocity sensors and the brake and throttle actuators. The law proposed in [7] for the followers in a platoon, on the other hand, makes use of additional information about the state of the leader of the platoon. It is assumed that this information will be transmitted to all the followers using an infrared communication link. Therefore, the predicate for the longitudinal follower law should also depend on the predicate for the infrared communication link.

In this formalism the quantitative capabilities of the regulation layer can be encoded by a vector of zeros and ones, of dimension equal to the number of control laws available to the layer. If there are n_{long} longitudinal laws and n_{lat} lateral laws this vector will have the form:

$$\{0,1\}^{n_{long}+n_{lat}}$$

As shown in the example, the design of the control laws implies a mapping from the vector coding the capabilities of the physical layer to the vector coding the capabilities of the regulation layer:

$$F_R : \{0,1\}^{n_s+n_a+n_c} \longrightarrow \{0,1\}^{n_{long}+n_{lat}}$$

figure 4 shows this mapping for the control designs in [8, 9, 10, 12]

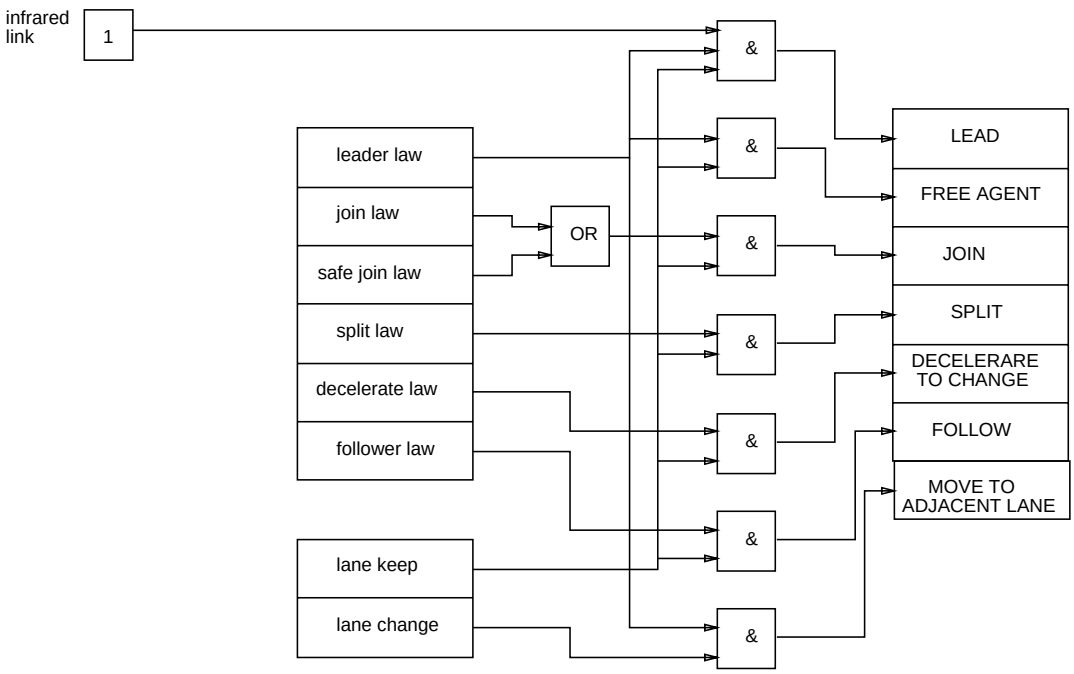

Figure 5: Regulation layer supervisor capabilities

3.1.3 Regulation Layer Supervisor Capability Predicates

From the point of view of the coordination layer, the regulation layer control laws represent resources that can be used to carry out maneuvers (such as merging and splitting platoons and changing lanes). Typically each maneuver will need to make use of two control laws, one longitudinal and one lateral. Therefore, in order for the coordination layer to be able to invoke certain maneuvers, the relevant control laws should be operational. For example, for the coordination layer to command a platoon leader to join, at least one (of possibly many) longitudinal join law and a lateral lane keeping law should be operational.

These capabilities of the regulation layer, when seen from the point of view of the coordination layer, can be modeled by predicates on the regulation layer capability vector. Those predicates can be put together to form a regulation layer supervisor capability vector. Let n_{man} denote the number of maneuvers that may be requested by the coordination layer. Then the regulation layer supervisor capability vector will be a vector of zeros and ones of dimension n_{man}. The design of the supervisor induces a mapping between the capability vectors of the regulation layer and its supervisor.

$$F_I : \{0,1\}^{n_{long}+n_{lat}} \longrightarrow \{0,1\}^{n_{man}}$$

For the normal maneuvers presented in [6] and the control laws of [8, 9, 10, 12, 13], the map F_I can be seen in figure 5.

3.1.4 Coordination Layer Supervisor Predicates

In order to operate normally, the coordination layer of [6] requires the vehicle to be able to perform certain maneuvers. More specifically a normal vehicle should be able to lead a platoon, be a follower, join a platoon (provided it is a platoon leader), split from a platoon (provided it is a follower), decelerate to facilitate a lane change and move from one lane to another. As discussed in the previous section, the capability to carry out these maneuvers will be coded by the regulation layer supervisor capability vector. In addition, to execute the protocols that organize the maneuvers, the coordination layer needs access to certain communication capabilities. Therefore, whether the coordination layer can operate in its normal mode or not can be expressed as a predicate on the values of the capability vectors for the regulation layer supervisor and the communications. A fault in the physical layer that will damage any one of these capabilities will clearly render the normal coordination layer design inoperable. For this reason alternative coordination layer protocols that are still operational under reduced capabilities will have to be designed.

An example of such a set of protocols is the *Take Immediate Exit* strategy described in section 4.2. The objective of this strategy is to take a vehicle that has developed a fault (and therefore can not function in the normal mode) out of the highway as soon as possible. The design makes use of additional maneuvers, which can easily be added to the predicate structure for the regulation layer and its supervisor. As we shall see in section 4.2, some of these maneuvers will require close cooperation with the neighboring vehicles. Therefore the applicability of the *Take Immediate Exit* strategy for the coordination layer can be expressed as a predicate on the values of the regulation layer supervisor capability vector and the communication capability vector of the faulty vehicle and the regulation layer supervisor capability vectors of the neighboring vehicles. It is assumed that knowledge about the capabilities of the neighbor will be obtained once communication has been established.

Once many such strategies have been designed (see section 4.2) the coordination layer capability can be expressed as a vector of zeros and ones. The dimension of this vector will be equal to the number of these strategies, which will be denoted here by n_{coord}. The design of the strategies induces a mapping:

$$F_C : \{0,1\}^{n_{man}} \times \{0,1\}^{n_c} \times \{0,1\}^{N n_{man}} \longrightarrow \{0,1\}^{n_{coord}}$$

Here N stands for the maximum number of neighboring platoons that may need to cooperate in an emergency maneuver. The part of F_C dealing with normal operation is shown in figure 6. Similar maps have to be constructed for any strategy introduced to deal with degraded conditions of operation.

3.1.5 Link Layer Supervisor Predicates

The link layer design of [2] makes use of information about the density and average velocity of traffic in a link to come up with control inputs (in the form of suggested paths, platoon sizes etc), that will maximize the throughput of the highway. In order to improve the resolution of the information and the commands, the link is partitioned into smaller sections. Each section consists of a single lane and its length is (typically) smaller than that of the

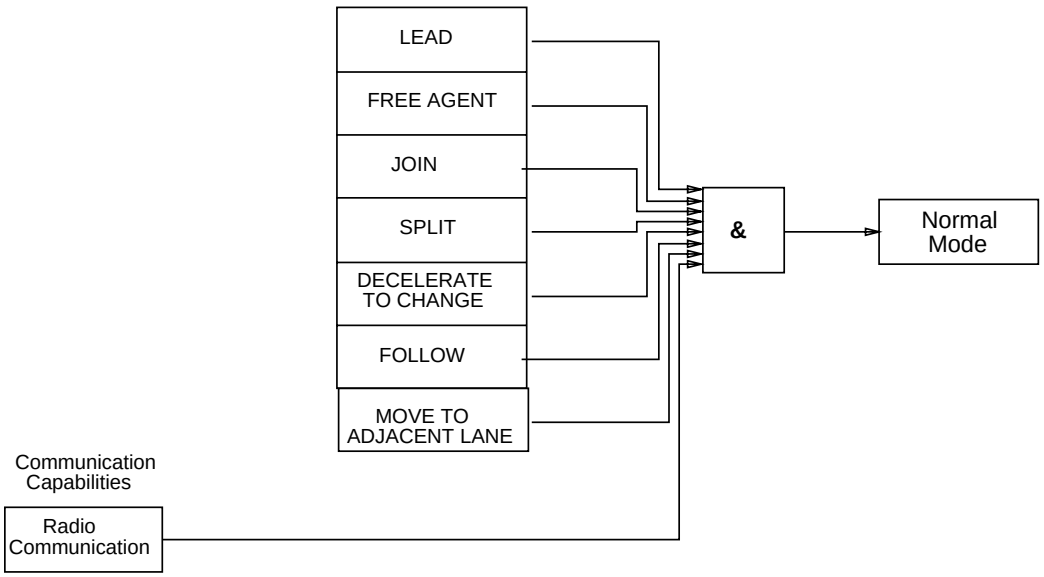

Figure 6: Coordination layer supervisor capabilities

link. Even though the controllers we propose for the link layer operation under faults are quite a bit different (refer to section 4.1) we will still keep this partitioning of a link into sections in our quantitative capability structure.

In addition to density and average velocity information (which will be provided by the sensor hierarchy), a link layer design for degraded conditions of operation needs information about discrete events (such as a lane being blocked) that limit the capabilities of its sections. The example presented in section 4.1 indicates four such events: section is blocked, section contains no vehicles, section contains vehicles queued behind an accident and section contains emergency vehicles. Similar events are also relevant for any entrances and exits that may be contained in the link. These properties can be modeled as a set of predicates for each section, entrance or exit that return one if the link possesses the property (e.g. is blocked) and zero otherwise. The value returned by these predicates should depend on the capability vectors of all vehicles in the section. For example if a section contains a broken down vehicle then the predicate for "section is blocked" should return one. In addition the values of the predicates should also reflect certain infrastructure faults. For example the fault "uncontrolled object in the lane" (which may refer to debris from an accident in one lane spilling over to an adjacent lane) should also cause the predicate "lane closed" to return one. Let n_I denote the number of the relevant infrastructure faults, N_i the number of platoons in section i and n_{sec} the number of predicates for each section ($n_{sec} = 4$ in the above discussion). Then for each section, as well as the entrances and exits contained in the link we can define maps:

$$\begin{aligned} F_{s_i} &: \{0,1\}^{N_i n_{coord}} \times \{0,1\}^{n_I} \longrightarrow \{0,1\}^{n}_{sec} \\ F_{en_j} &: \{0,1\}^{N_j n_{coord}} \times \{0,1\}^{n_I} \longrightarrow \{0,1\}^{n}_{sec} \\ F_{ex_k} &: \{0,1\}^{N_k n_{coord}} \times \{0,1\}^{n_I} \longrightarrow \{0,1\}^{n}_{sec} \end{aligned}$$

where i, j, k range over the number of sections, entrances and exits contained in the given link. As will be seen in section 4.1, the values of the output predicates for each section will be used by the link layer controllers as events that will trigger transitions from one desired velocity and density profile to another.

3.2 Qualitative Capability Structure

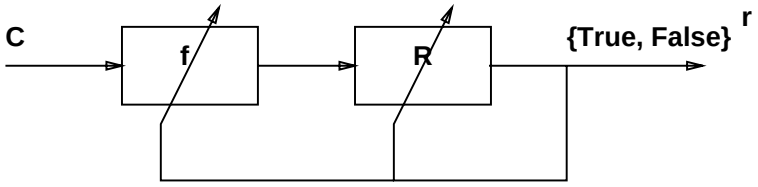

Figure 7: On-line controller tuning

Assessing the qualitative capability of the system is intimately related to determining the kind of disturbances against which the system is robust. There are three factors involved in this process. The first is the causes of gradual performance degradation, which the supervisor will have to guard against. These causes include adverse weather conditions (such as rain, fog or snow) and gradual hardware degradation (such as brake wear). The second factor is the qualitative capability parameters that can be used to monitor the capability of the system. These capability parameters depend on the layer of the architecture and include, for example, the maximum and minimum deceleration available to the vehicle (for the physical layer) and the maximum tracking error of the various continuous time controllers (for the regulation layer). The final factor is the qualitative performance requirements. They can be thought of as bounds on the capability parameters.

Robustness analysis involves finding functional relationships between causes of gradual performance degradation and the qualitative capability parameters. If we denote the set of causes of performance degradation by

$$\mathcal{C} = \{C_i / i = 1, \ldots c\}$$

and the set of qualitative capability parameters by $\mathcal{P}$, then the task of robustness analysis involves determining a map:

$$f : \mathcal{C} \longrightarrow \mathcal{P}$$

The performance requirements can then be thought of as predicates on the values of the capability parameters:

$$R_i : \mathcal{P} \longrightarrow \{True, False\} \quad i = 1, \ldots r$$

Then the range of conditions $\hat{\mathcal{C}}$ under which the performance of the system is acceptable is given by the relation:

$$\hat{\mathcal{C}} = \bigcap_{i=1}^{r} f^{-1}(R_i^{-1}(\text{True})) \subset \mathcal{C}$$

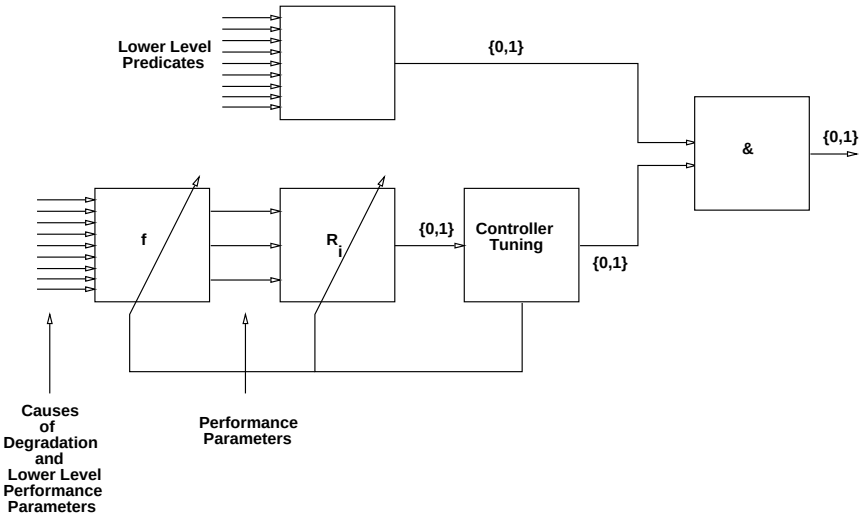

Figure 8: Introduction of robustness predicates

Enhancing the robustness of the system involves enlarging $\hat{\mathcal{C}}$. One way of doing this is by on-line tuning of the controllers. If the requirements of the control laws are not met by the capability parameters at any time (e.g. a join controller may need the vehicle to decelerate faster than the capability of the vehicle at the current time), the control laws are tuned until the new requirements are met by the parameters. This process can be represented by figure 7.

Even after the domain $\hat{\mathcal{C}}$ has been maximized there will probably still be some conditions in $\mathcal{C}$ which are not covered. These conditions for which performance is unacceptably degraded will have to be treated by the supervisor in a way similar to the treatment of loss of capability due to faults. In this sense, the effect of gradual degradation and limits of robustness can be modeled as an extra term on the predicates (figure 8). For certain combinations of environmental parameters, a given sensor, actuator, communication device or controller (be it regulation, coordination or link) can still produce acceptable performance by appropriately tuning some of its parameters. After some point the degradation is so severe that the resource is effectively useless.

Modeling qualitative capability in this way and carrying out the robustness analysis of the current architecture is in itself a separate research topic. It is presented in detail in [14].

3.3 Classification of faults by capability

A comprehensive list of faults pertaining to vehicle as well as infrastructure failures is attached in the appendix. To simplify the task of designing degraded modes, the faults were grouped in six classes (section 3.3.1 through 3.3.6), according to the capabilities remaining to the vehicle/system after the fault has occurred. The reason for this aggregation of faults is to allow us to design degraded modes for each class, thereby resolving many faults at once.

Every fault mentioned in the appendix can be assigned to a unique class. In fact the list in the appendix is already classified according to the classes we are going to define in this section. The first four classes (section 3.3.1 through 3.3.4) consider faults in the vehicle. The classes are arranged in descending order of severity (from the point of view of safety as well as spatial extent of degradation of performance). The faults in section 3.3.5 will involve large areas of the highway whereas those in section 3.3.6 are restricted to the entry and exit lanes.

The fault classification presented below is supposed to be general, i.e. it does not depend on the list of faults in the Appendix. Thus given a fault not on the list, it can still be uniquely classified in one of the classes.

3.3.1 Vehicle stopped/must stop

This class contains the most serious faults. The vehicle can not continue moving on the AHS safely and has either already come to a stop or it should be commanded to do so and wait to be towed away. Because of the severity of the situation, all the layers of the control architecture will undergo some degradation in performance and assist in resolving the fault condition.

Faults in this class will typically lead to a false "Capable of being a free agent" predicate in the Interface. Depending on the type of fault we identify three subcategories which are differentiated by the technique that is used to stop the faulty vehicle. The subclasses and the faults contained in each one of them are listed in the appendix. In section 4.1.2 possible control strategies for dealing with each one of these subcategories and the situation that arises once the faulty vehicle has stopped are outlined.

3.3.2 Vehicle needs assistance to get out

The faults in this class are slightly less serious. The vehicle may continue but has lost some essential capability and it must therefore exit the AHS as soon as possible. Moreover, it needs the assistance of its neighbors and/or the infrastructure. Typically faults in this category can be handled locally and need not involve the higher levels of the architecture (link and network). As before the faults are divided into subclasses, according to the affected capability. Control strategies for dealing with each one of these subcategories will be outlined in section 4.2.

3.3.3 Vehicle needs no assistance to get out

The faults in this class are even less serious. Typically the vehicle is fully functional but should leave the system soon to avoid further problems and hazards (in case a second fault occurs for example). Typically faults in this class are handled by special controllers in the regulation layer and neither the neighboring vehicles nor the roadside need to be alerted. In a sense the faults in this class can be contained by using just regulation layer action.

3.3.4 Vehicle need not get out

This class contains minor faults that require no special action but should nonetheless be recorded and the driver should be notified in case the travel plan needs to be altered and so that the fault can be fixed at the destination.

3.3.5 Infrastructure Failures

This class includes all faults that induce a reduction in the capability of the infrastructure. They usually lead to severe degradation in performance. Some of them can be handled by the normal mode controllers of the link and network layers, but some may need drastic changes in the operation of the system. The faults reflected in the infrastructure predicates discussed in section 3.1.5 are contained in this class.

3.3.6 Driver/Computer Interaction Down

Problems in this class mainly occur during the entry and exit to the system. We assume that once on the automated freeway, the driver may not interfere with the system operation and therefore can not induce any special faults.

4 Control strategies for degraded modes

In the previous section, we presented a classification of faults that can occur on an AHS. Recall that the classes were; vehicle stopped (or must stop), vehicle needs assistance to leave, vehicle needs no assistance to get out, vehicle need not get out, infrastructure failures, and driver/computer interaction failures. The first criterion of classification, thus, was some rough notion of severity capturing the degree to which the fault violates safety criteria. The safety criteria, at this stage, have not be explicitly specified, but will be needed for the optimal design of the supervisor. We showed that there was a natural subclassification of faults in terms of capability of the roadside or vehicles to execute control strategies.

Based on the available capabilities of the roadside or vehicle, the supervisor selects control strategies in order to respond to the fault. The supervisor uses one of two schemes to select the control strategy. If the fault is vehicle borne, the outcome of the capabilities maps directly to a control strategy that is implemented by the coordination layer. If the fault is in the infrastructure, or the fault requires assistance from the link layer, it also translates into reduced capabilities of the link layer. However, this set of capabilities does not directly allow a map from infrastructure capabilities to control strategies because the model for the link layer is not a discrete event system. As the current design of the link layer uses a fluid flow model, a translation is made from capabilities to control strategies for the link layer using a density/velocity profile generator, which will be described below.

As already discussed the lower layers have access to more detailed information and are therefore better suited to assess the safety of a given situation. Moreover their commands are more localized, that is they affect a smaller subset of the system. Therefore, as a general

principle, as many decisions or control actions as possible will be delegated to the lower layers of the control hierarchy in order to make the system more robust to error. Thus, control actions are localized and failures that affect them are also localized. Localization of failures is our first criterion for optimality of the extended architecture. What is presented in this section is an outline of control strategies that are proposed for dealing with the fault classes presented in the previous section. These control strategies have not been explicitly optimized along other dimensions of system performance, most importantly capacity and safety. It is an ongoing effort to prove that the extended architecture is in fact safe and optimal in terms of capacity maximization.

4.1 Link layer design

The goal of the link layer controller in the normal mode is to achieve smooth flow of traffic over a length of highway on the order of a few kilometers. The link layer controller achieves smooth flow in spite of disturbances from vehicles entering, vehicles exiting, and maneuvers. In case of reduced capabilities of the system, the link layer takes an extended role to help clear faults on the highway and to divert traffic away from incidents resulting from faults. The link layer is not considered a safety critical subsystem, in the sense that vehicles on an AHS are equipped with sufficient intelligence to avoid being involved in a catastrophic collision. In vehicle borne faults, the link layer assists in clearing the fault, and thus is still not safety critical. For infrastructure faults, such as an uncontrolled object on the AHS, the link layer adopts a strategy meant to best circumvent a catastrophe. Other control objectives are for vehicles to make their exits and for capacity to be maximized. We will describe the control design framework by which these collective (possibly competing) goals of the link layer will be achieved.

4.1.1 Link layer controller

The link layer is modeled by a fluid flow model whose state variables are density $k = k(x, y, t)$ and aggregate velocity field $\mathbf{v}(x, y, t)$, where x is the length along the highway, and y is lateral distance, equivalent to a continuous form of the lane number. The model consists of a law of conservation of vehicles or continuity equation and an aggregate velocity equation modeling vehicle follower behavior. The velocity law includes a relaxation term to capture dynamics of tracking the desired aggregate velocity and an anticipation term, to model headway keeping of vehicles. For implementation purposes the link is spatially discretized into 1km long sections. Details of this strategy can be found in [2].

Given a fluid flow model as the basis for control of the link layer, in the normal mode, the link layer control strategy is to balance density and regulate velocity, while also ensuring that all vehicles make their exits. The control inputs for the link layer are proportions of vehicles changing lanes within sections of the link (regulating lateral flow) and aggregate velocity (regulating longitudinal flow). In a faulted condition, the link layer may use an extended set of control inputs, including, in addition to velocity and lane change control, average platoon size control, maneuver control, ramp metering, and exit control. These control inputs need to be integrated in a coherent manner in order to achieve the control

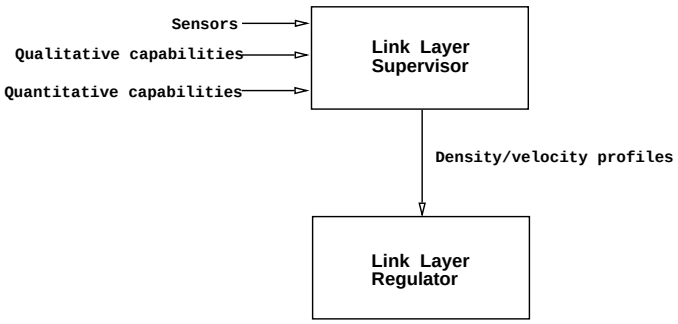

Figure 9: Link layer controller overview

objectives.

Referring to figure 9, the link layer controller for the extended architecture will consist of a supervisor that takes as input, capabilities identified with each section. When a capability predicate of a section changes, either because of a fault or a robustness spill over such as congestion, the supervisor will issue a sequence of control commands in the form of desired density and velocity profiles. The control objectives for the link layer during degraded modes include:

- incident avoidance
- emergency vehicle access
- congestion dissipation
- create gap

Combinations of control objectives can be present at the same time within a link (due to multiple faults in a link for example). They will be combined into a single command for the link layer regulator using the desired density/velocity profile generator of the link supervisor. The desired density/velocity profile generator will produce a profile of aggregate velocity and density which achieves the control objectives of the extended architecture, while also maximizing capacity and ensuring that all vehicles make their exits. The profile generator will create stationary density and velocity profiles that satisfy the continuity equation. It is then necessary to design a lower level controller that will produce the necessary control inputs to track the desired density/velocity profiles. This tracking control law forms the the second layer of the link layer controller which is called the link layer regulator.

We now give an example of a sequence of scenarios which correspond to switching of values of capability predicates of link sections. Corresponding to changes of capability predicates are desired density/velocity profiles issued by the link supervisor and tracked by the link regulator.

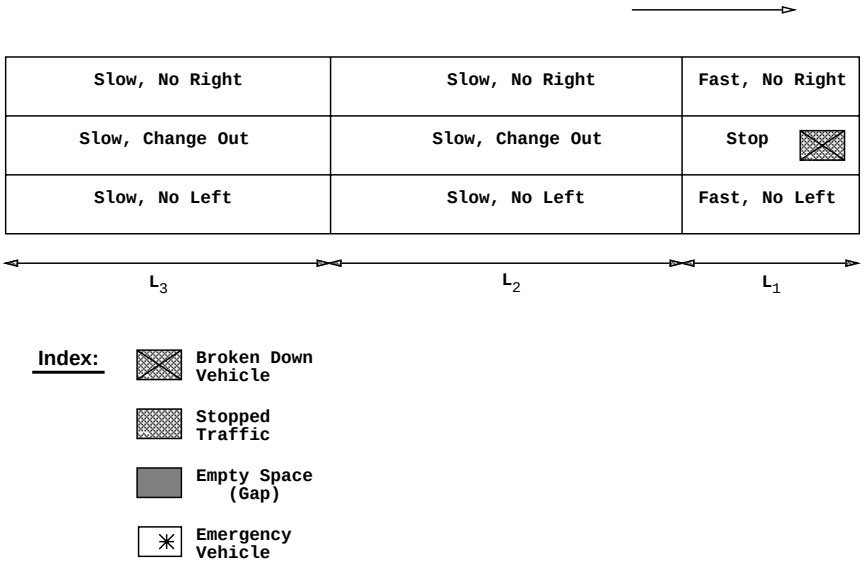

Figure 10: Stage 1: Vehicle stopped on highway

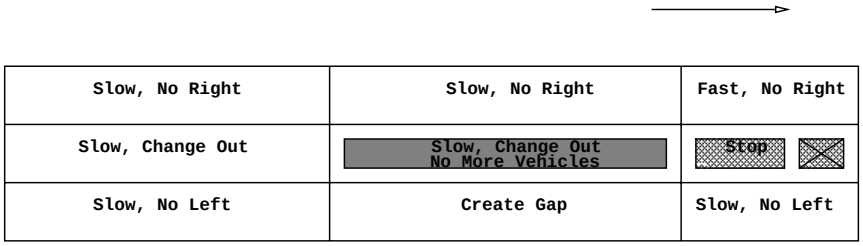

Figure 11: Stage 2: Vehicle stopped on highway

4.1.2 Example: link control strategy for a stopped vehicle

Suppose a faulty vehicle has stopped in the middle lane of a three lane highway. The vehicles immediately behind the faulty vehicle will stop and queue behind the stopped vehicle. The safety critical task of stopping vehicles upstream before they hit the stopped vehicle is carried out by the regulation layer control laws. The link layer controller will invoke an incident avoidance control strategy. The following figures present snapshots of capabilities within the link at different time intervals. We show temporal and spatial extent of the degraded performance. In Stage 1 (figure 10), the section labeled stop has a change of capability of the form "section is blocked". The commands slow, no right, change out, etc. represent abstractions of the desired density and velocity profiles. We include them here to give a sense of a typical behavior that may result from a link supervisor command. Adjacent lanes are slowed down to facilitate the vehicles from the stopped lane to change out. Some vehicles will queue behind the incident.

In Stage 2 (figure 11), there are no vehicles in the stopped lane in section L_2. There is a gap created in an adjacent lane which travels towards the stopped vehicles at the speed of that

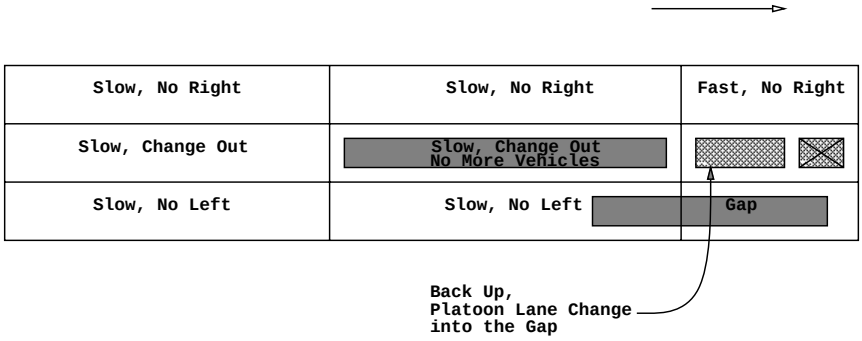

Figure 12: Stage 3: Vehicle stopped on highway

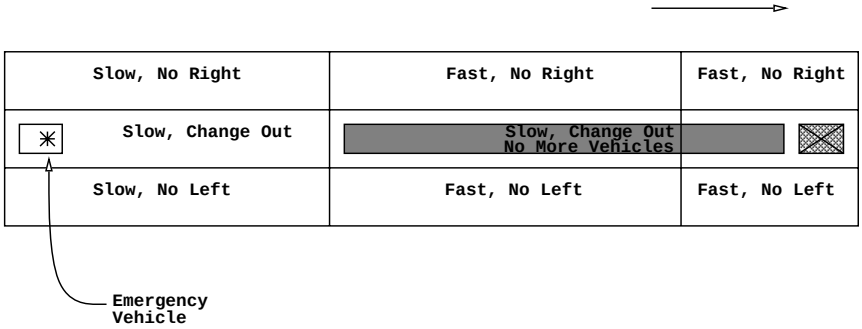

Figure 13: Stage 4: Vehicle stopped on highway

lane. This strategy has been triggered by the predicate "section contains queued vehicles" becoming true for section L_1, as well as the predicate "section contains no vehicles" for section L_2 becoming true.

As the gap approaches (figure 12), the queued up vehicles *Back Up* in the empty space in L_2, speed up to adjacent lane speed and change lane into the gap. The gap creation and vehicle removal will go on until all the vehicles which are queued up behind the faulty vehicle are cleared.

In the meantime (figure 13), the emergency vehicles move towards the incident using the blocked lane. As the lane is empty from L_2 onwards and the vehicles in this lane in L_3 are moving out, the emergency vehicle will probably be moving faster than the vehicles in adjacent lanes. Alternatively, if the emergency vehicle shows up earlier, then we can stop lane changes from blocked lane to one of the adjacent lanes and let the adjacent lanes carry the emergency vehicle faster. This control strategy is triggered to the link supervisor by the entry of the emergency vehicle to the AHS and is a change of capability termed "section contains emergency vehicles".

At the end of stage 5 (figure 14), the emergency vehicle has reached the stopped vehicle and is moving ahead of it (and any remaining queued up vehicles) using the algorithm of Stages 2 and 3 (gap creation in adjacent lane).

Finally, in the recovery mode (figure 15) some restrictions on speed and lane changing

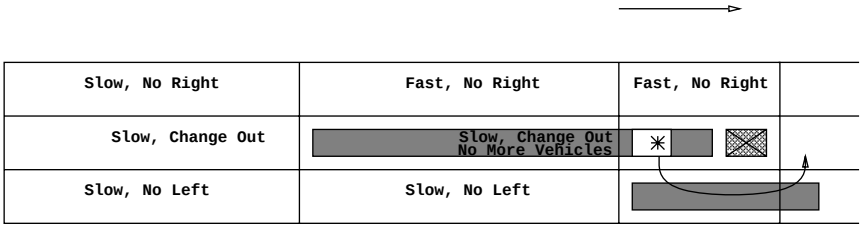

Figure 14: Stage 5: Vehicle stopped on highway

activity are imposed to avoid further crashes due to large velocity differentials across lanes.

By this example, we demonstrate that the link layer, though it is modeled as a continuous fluid flow model, has an event driven component which is handled by the link layer supervisor based on changes of capability predicates.

4.2 Coordination layer design

Analogous to the link layer, the coordination layer consists of a two level control structure. The coordination supervisor is the strategic planning level. It determines sequence of maneuvers that a vehicle carries out. The lower level contains protocols for coordination of individual maneuvers with the neighbors. We call this level the coordination layer maneuver level. The normal mode coordination layer is structured in a similar way. New strategies are added both to the coordination supervisor and to the coordination maneuver level in order to extend the coordination layer control design for faulted conditions.

Referring to the list of faults in the appendix, we assign new coordination strategies to each type of fault. We will list the names of these strategies first, and later provide short descriptions of how the strategies operate. Part of the design of the coordination layer is to build and verify protocols that execute the control strategies. The coordinating protocols and supervisor strategies are modeled using finite state machines. We attach figures showing an example strategy, called *Take Immediate Exit* (figure 16) and an example protocol for *Forced Split* maneuver (figure 17). Currently we are in the process of verifying the design using automatic verification software such as COSPAN. This will be a mathematical proof

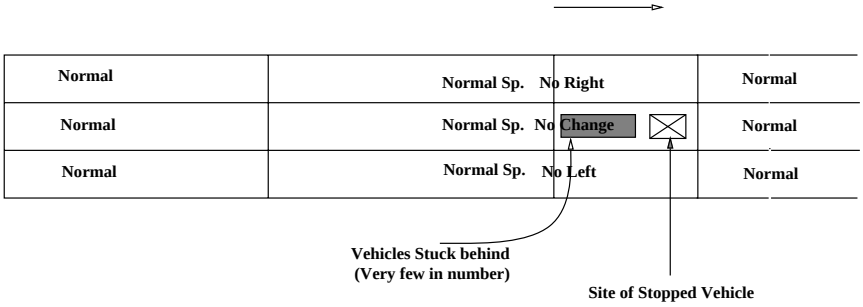

Figure 15: Recovery after the stopped vehicle is towed away

that the coordination layer logic works properly.

For faults in the class "vehicle stopped/must stop" a two step strategy is employed. In the first step a strategy for stopping the vehicle is chosen while the second step determines what needs to be done once the vehicle is stopped. If the vehicle is stopped before the fault is detected only the second step is relevant. The strategy employed for the first step depends on which subclass the fault belongs to. For subclass 1 the control strategy is *Gentle Stop*, subclass 2 *Crash Stop*, and subclass 3 *Aided Stop*. Once the vehicle comes to rest, the link layer employs strategies to ease congestion, divert traffic away from the incident, assist emergency vehicles, and get the queued vehicles out. For faults in the class "vehicle needs assistance to get out" a strategy called *Take Immediate Exit* is executed by the coordination layer (See figure 16). This strategy is used by all subclasses except in cases where the vehicle capabilities limit its use. In particular, for faults of subclass 2, *Take Immediate Exit - Escorted* is used. Note that the link layer need not be involved for faults in this class. For faults in the class "Vehicle needs no assistance to get out" a control strategy called *Take Immediate Exit - Normal* is chosen by the coordination layer supervisor. We now describe these coordination layer control strategies in greater detail.

4.2.1 Coordination Layer Supervisor

The task of the coordination layer supervisor is to pick the best possible strategy for the given circumstance. Many of the strategies presented below need coordination from the neighboring vehicles. The supervisor starts with the assumption that the neighbors are capable of carrying out all the necessary maneuvers. If the strategy is unsuccessful, because one of the neighbors did not abort its current maneuver, then the supervisor updates the capabilities of the neighbors and chooses the next best strategy that is feasible with the updated capabilities. The working of the supervisor will not only be based on a discrete capability monitor but also on the continuous state of the system. New strategies for coordination supervisor include:

Gentle Stop and Crash Stop Strategies: These control strategy consists of bringing the faulty vehicle to a complete stop on the highway. The names of the strategies indicate the severity of braking used to execute the maneuver.

Aided Stop Strategy: The faulty vehicle (which has a brake failure) is aided by the vehicle immediately ahead of it in the same platoon to come to a stop. If the faulty vehicle is a leader, it uses the *front dock* maneuver to become a follower before executing *aided stop*.

Take Immediate Exit: The *Take Immediate Exit* strategy is used by a faulty vehicle to get out of the AHS as soon as possible. The strategy consists of up to two *forced split* maneuvers to become a free agent. The free agent then executes a number of *emergency lane change* maneuvers until it reaches the rightmost automated lane from where it will take the next exit.

Take Immediate Exit - Escorted: This strategy is used by a faulty vehicle that has lost the capability to be a leader but can still be a follower. In this case, the faulty vehicle

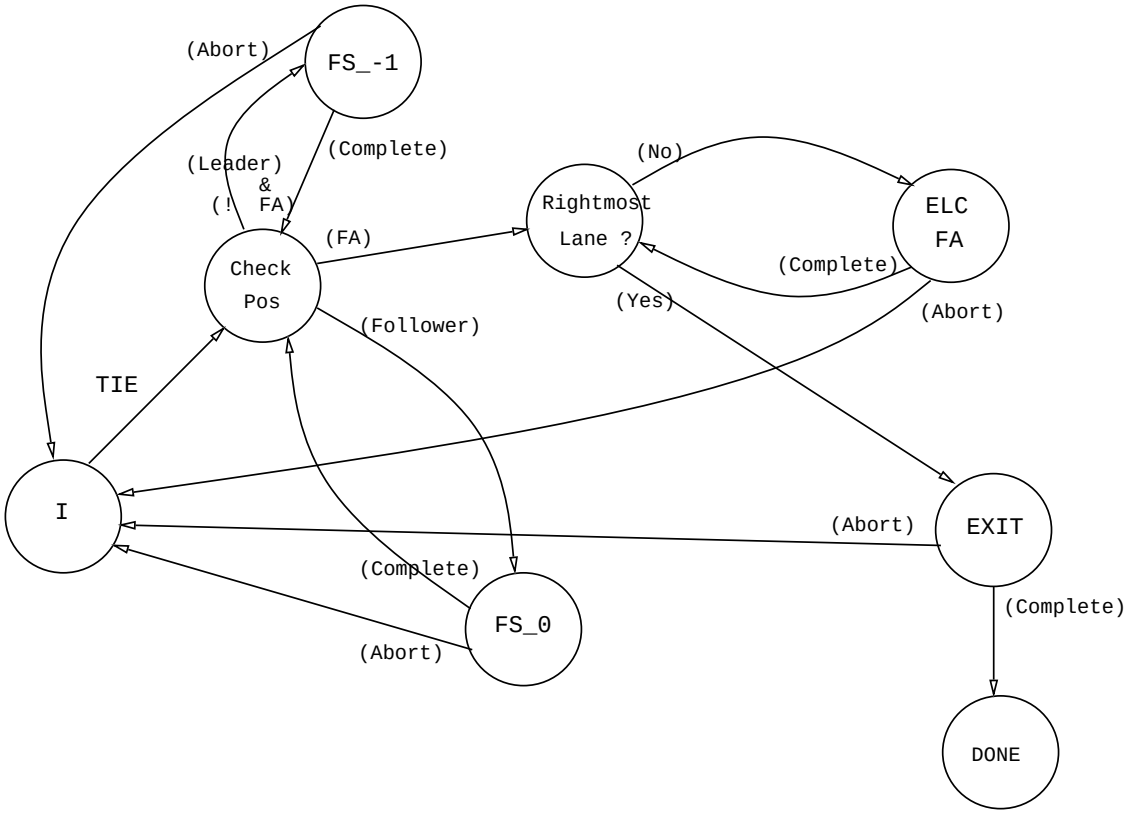

Figure 16: State Machine for Take Immediate Exit

leaves the system as part of a two vehicle platoon in which the faulty vehicle is the follower. This requires a *front dock* maneuver if the faulty vehicle is a leader of a platoon to start with. The leader of this new platoon (called the *escorting vehicle*) will now escort the faulty vehicle out of AHS by executing number of *emergency lane change* maneuvers of the two vehicle platoon to take immediate exit. Once out of the AHS, the escorting vehicle drops off the faulty vehicle in a special turnout called "dormitory" and re-enters the AHS at the next entrance.

Take Immediate Exit - Normal: This strategy is similar to the previous strategy except the faulty vehicle uses a normal lane change protocol of [6] and control laws of [10] instead of *emergency lane change*.

4.2.2 Coordination layer maneuvers

To implement above control strategies the coordination layer supervisor makes use of the following emergency maneuvers (in addition to some of the normal mode maneuvers).

Forced Split: This maneuver is similar to the split maneuver of [6]. Being an emergency maneuver, it can abort some of the normal mode maneuvers. Forced split is typically

used by a faulty vehicle to become a free agent. A follower can request the leader of the platoon to initiate a *forced split*. The leader can break the platoon at any location using *forced split*. (See figure 17).

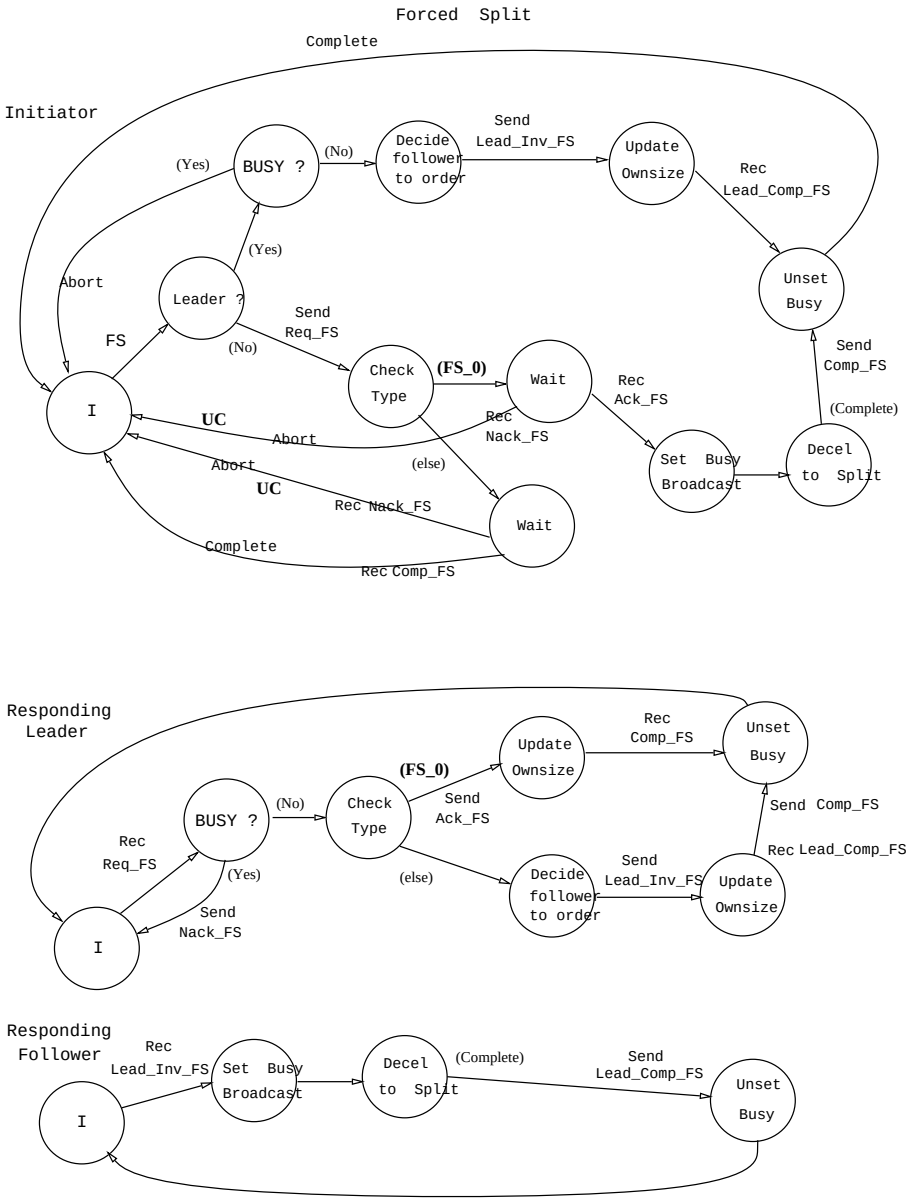

Figure 17: Forced Split maneuver protocol

Emergency Lane Change: This maneuver typically will be used by a free agent with reduced capabilities. The faulty vehicle will request two platoon leaders in the next lane to create and maintain a gap so that the faulty vehicle can change lane into it. We will also use a special case of *emergency lane change* for a platoon of vehicles to change lane into a gap. For this case, we will need an additional platoon lane change

lateral controller in the regulation layer.

Front Dock: This maneuver is initiated by a platoon leader that wants to join the vehicle in front but, because of a fault, it can not execute the join maneuver of [6] (which requires accelerating to close the gap and then decelerating to match the speed). The initiating vehicle requests the leader of the preceding platoon for a *front dock*. The leader of the preceding platoon will order the last vehicle in its platoon (itself in the case of a free agent) to front dock with the initiator. This vehicle will then decelerate to close the gap between itself and the initiator. In the end, the initiator becomes the first follower of the new platoon. Thus the leader of this new platoon has joined the trailing platoon by decelerating.

Aided Stop: This maneuver is initiated by a follower that has a brake failure because of which it can not get any deceleration using its brakes. The faulty vehicle will use its engine to decelerate and ask the leader to assist in bringing it to a stop on the highway. The responding vehicle (the vehicle immediately ahead of the faulty vehicle) will apply gentle braking and let the faulty vehicle collide with it from behind. The responding vehicle will then use its brakes to bring the combined mass of both the vehicles to a stop.

Gentle Stop: This maneuver is used by a faulty vehicle that is ordered to stop and can do so by using its brakes. The fault is not severe enough to require the vehicle to use maximum emergency braking. The vehicle will use gentle braking in order to minimize the discomfort to the passengers and the disturbance to vehicles behind.

Crash Stop: This maneuver is similar to *Gentle Stop* except the severity of the fault requires the faulty vehicle to apply maximum emergency braking.

Queue Buildup and Queue Management Whenever a faulty vehicle is stopped on the highway, vehicles in the same lane immediately behind the faulty vehicle will form a queue of stopped vehicles. The link layer controller helps stop the queue buildup by diverting traffic upstream of the stopped vehicle from the stopped lane to the other lanes. If the emergency vehicle (e.g. tow truck) has not appeared until the queue buildup has stopped, we need a strategy to get the queued vehicles moving again. In our strategy, the queue is dissipated in Last In First Out (LIFO) fashion. We use the *queue buildup* maneuver to keep track of the current leader of the queue. With a LIFO strategy, the queue leader is the last car of the queue. When the queue buildup stops, there is a large gap behind the last vehicle of the queue. This gap is created by the link layer strategies of diverting the traffic upstream of the stopped vehicle. This is the proper stage for *queue dissipation*. Queue dissipation will be carried out in some fixed platoon sizes. The link layer will order creation of gap in the adjacent lane upstream of the accident. This gap will travel towards the queued up vehicles. At this time, a platoon of appropriate size will break up from the queue and back up into the gap behind the queue. This platoon will stop its backward motion when it creates a sufficient gap in between its front car and the last vehicle of the remaining queue. The platoon is now ready to accelerate up to the speed of the next lane and change lane (using *emergency lane change for the platoon*) into the gap that is approaching it.

The backup distance depends on the speed of the approaching gap and the constraints on acceleration and jerk.

4.3 Regulation layer control laws

Finally, we propose extensions to the regulation layer of the vehicles in order to achieve the control strategies described in the previous section for the coordination layer. The following control laws are based on a continuous time dynamic model of the vehicle.

4.3.1 Longitudinal Control Laws

Gentle Stop Control law to bring vehicle to a stop with gentle deceleration (e.g. deceleration not more than $0.1m/s^2$).

Crash Stop Control law to apply maximum braking to bring vehicle to a stop as quickly as possible.

Aided Stop *Gentle stop* control law with "tight" lateral control to maintain lane position even after the rear vehicle has collided with it.

Front Dock Feedback control law to join the vehicle behind by decelerating to close up the gap.

Back Up Control law to back up in an empty space. The initial and final velocities of the vehicle will be zero with a specified gap created in front of the vehicle at the end of the maneuver.

Create & Maintain gap Control laws to create a gap of specified size behind the preceding platoon and to maintain it with the specified velocity.

4.3.2 Lateral Control Laws

Tire blow lateral control Feedback controller to maintain lateral position within a lane in case of tire blowout. Such a control has already been developed (see [3]) and is currently being tested experimentally.

Platoon Lane Change Lateral control law for the followers of the platoon to change lane.

Recapture Magnetic Marker If a vehicle wanders out of its lane (because of damaged lane markers for example) we need to bring it back by using information other than the one provided by the magnetic markers. A control law can be designed using both the left and right lateral sensors to keep the vehicle in between its neighbors, until the lane markers are located again by the magnetometer sensor.

5 Conclusions and Future Work

In this paper, we presented a framework for designing control laws for an AHS system that will be capable of operating under any conditions. We illustrated that the control structure used under normal operating conditions is insufficient for degraded modes of operation. The reason is that the normal mode assumes fixed capabilities of the system a priori. This assumption is violated in a degraded mode. We gave an outline of an extended architecture that will help resolve this problem. We presented an explicit design of the part of the architecture that monitors the system capabilities in the presence of faults. The capabilities framework formed the inputs to extended link, coordination, and regulation layer supervisors that select control strategies for operation under adverse conditions. For the link layer we described a density/velocity profile generator and link layer regulator. For the coordination and regulation layers we have listed the necessary additional maneuvers and control laws, and how they should operate.

Work is in progress on formal verification of the extended coordination layer protocols. In particular we will prove that the system is deadlock free and at every time instant the coordination layer is commanding the regulation layer to apply some control to the actuators. This will be a mathematical proof of the fact that the logical design of the extended coordination layer works properly. The overall goal is to prove that the extended control architecture will have a control law to deal with any situation arising from any choice of three or less faults in a neighborhood of the highway. To prove that the overall system is "safe" in case of up to three faults in a neighborhood will require development of hybrid system tools to analyze the system formed by combined coordination and regulation layer system. That is one of the long term goals of our project. Work also needs to be done in the area of link layer and regulation layer design.

It should be noted that the solution proposed in section 4 is by no means unique. Alternative control strategies can definitely be developed to deal with the same problem. Our long term goal is to come up with explicit optimality criteria that will allow us to compare the various designs. These criteria should be based on some formal description of capacity and safety of the AHS. They will allow us to conclude which one among possibly many control policies is the best in a given situation.

Acknowledgment: The authors would like to thank Roberto Horowitz, Antonia Lindsey, Shankar Sastry, Ekta Singh, and Pravin Varaiya for helpful discussions providing insight into this problem.

References

[1] P. Varaiya, "Smart cars on smart roads: problems of control," *IEEE Transactions on Automatic Control*, vol. AC-38, no. 2, pp. 195–207, 1993.

[2] B. S. Y. Rao and P. Varaiya, "Roadside intelligence for flow control in an IVHS," *Transportation Research - C*, vol. 2, no. 1, pp. 49–72, 1994.

[3] M. Tomizuka, S. Patwardhan, W. B. Zhang, and P. Devlin, "Theory and experiments of tire blow-out effects and hazard reduction control for automated vehicle lateral control system," in *American Control Conference*, pp. 1207–1209, 1994.

[4] A. Hitchcock, "A specification of an automated freeway with vehicle-borne intelligence," Tech. Rep. UCB-ITS-PRR-92-18, PATH Technical Memo, Institute of Transportation Studies, University of California, Berkeley, 1994.

[5] P. Varaiya and S. E. Shladover, "Sketch of an IVHS systems architecture," Tech. Rep. UCB-ITS-PRR-91-3, Institute of Transportation Studies, University of California, Berkeley, 1991.

[6] 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.

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

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

[9] S. Sheikholeslam and C. A. Desoer, "Longitudinal control of a platoon of vehicles," in *American Control Conference*, pp. 291–297, 1990.

[10] D. N. Godbole and J. Lygeros, "Longitudinal control of the lead car of a platoon," in *American Control Conference*, pp. 398–402, 1994.

[11] S. Alag, K. Goebel, and A. Agogino, "A framework for intelligent sensor validation, fusion and supervisory control for automated vehicles in ivhs," Tech. Rep. 94-0901-0, BEST lab, University of California, Berkeley, 1994.

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

[13] J. Frankel, L. Alvarez, R. Horowitz, and P. Li, "Safe merge maneuver." (preprint).

[14] J. Lygeros and D. N. Godbole, "Robustness analysis of PATH control architecture." (preprint).

A Faults

A.1 Vehicle stopped/must stop

1. no throttle control, no engine power, out of gas, no power transfer, vehicle to vehicle communication down (Radio - Needed for coordination)

2. no steering control, uncontrolled object ahead, no control computer, magnetometer failure, no sensing of distance and velocity of car ahead (long and short range)

3. no brake control

A.2 Vehicle needs assistance to get out

1. no control of transmission / selection of gear

2. no long-range (longitudinal) sensing of vehicles

3. no short-range (longitudinal) sensing of vehicles

4. no lateral sensing of vehicles

5. flat tire - reduced steering capability

6. Vehicle - Vehicle communication down (Infra-Red: Needed for Follower operation)

A.3 Vehicle needs no assistance to get out

1. Non-crucial Sensor fault: engine sensor (e.g. intake manifold pressure sensor), accelerometer, wheel speed, etc.

2. low on gas

3. Single fault in a redundant sensor set

4. Vehicle-roadside communication down because of on-board equipment failure

A.4 Vehicle need not get out

1. Lights won't go on

2. In vehicle displays not working

3. Out of range of magnets, magnetometers working, not changing lanes

A.5 Infrastructure Failures

1. Network layer down or communication between link and network down

2. Link layer down or communication between link and vehicle down in the entire link due to roadside equipment failure

3. unable to communicate with object on AHS

4. Lane(s) Blocked

5. Exit(s) Closed

6. Entry(s) Closed

7. Robustness Spill Over and environment
   In this category we group all problems caused by unfavorable conditions that the normal mode controller is not robust enough to handle. These problems may not be the result of faults, but may arise due to gradual performance degradation. Since they result to certain normal mode predicates calculating as false, however, degraded modes will have to be designed for them. If the gradual performance degradation is limited to a single vehicle then it will be classified into one of the classes 3.3.1 through 3.3.4. Here we consider the effect on the infrastructure. This is mainly caused due to environmental degradation such as rain or snow. They will be grouped in two subclasses:

   - loss or reduced traction with road (lateral & longitudinal)
   - reduction in sensor range or accuracy (caused by rain, dust, sunshine, etc.)

A.6 Driver/Computer Interaction Down

Problems in this class mainly occur during the entry and exit to the system. We assume that once on the freeway the driver may not interfere with the system operation and therefore can not induce any special faults.

1. Improper Exit: Driver unable to take control and/or system unable to transfer control at exit

2. Improper Entry: Wrong destination/route entered by driver or system unable to start automatic control at entrance or manual driver tries to enter automated TL

A.7 Faults not considered

1. Software implementation errors

2. Design errors such as protocol design errors, control design errors

3. Communications errors including: wrong message, message to wrong car, etc.