Design of an Extended Architecture for Degraded Modes of Operation of IVHS [1]

John Lygeros, Datta N. Godbole, Mireille E. Broucke
Department of Electrical Engineering and Computer Sciences
University of California,Berkeley, CA 94720
{lygeros, godbole, mire}@robotics.eecs.berkeley.edu

Abstract

This paper presents a hierarchical control architecture for dealing with faults and adverse environmental conditions on an Automated Highway System (AHS). Our design builds on a previously developed control architecture [1] for normal operating conditions. The faults considered in the extended architecture are classified by capabilities remaining on the vehicle or roadside after the fault has occurred. The set of available capabilities is used by supervisors in each of the layers of the hierarchy to select appropriate control strategies. We outline the control strategies needed by the supervisors and give examples of their detailed operation.

1. Introduction

One of the goals in California and the nation's IVHS effort is the design of an Automated Highway System (AHS) that can significantly increase both safety and highway capacity by adding intelligence to the vehicle and the roadside and without building new roads. Several approaches have been proposed, 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 takes 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 concerned with specific faults rather than a general framework. Our goal is to propose an AHS design that will perform safely under almost any condition with the exception of faults in the design (e.g. a deadlock in the protocols) and faults in the implementation of the software. Even with this restriction the task is large. The magnitude of the design problem leads to a hierarchical control structure which facilitates complexity management.

The control hierarchy for normal operation outlined in [1] is based on the idea of "platooning". A platoon is a group of tightly spaced vehicles with separations of about 1 meter and with an inter-platoon distance of the order of 30 meters. It has been shown that platooning results in a substantial increase in capacity and safety. However, platooning requires automatic control of vehicles, as human drivers are not fast or reliable enough to produce the kinds of inputs necessary for maintaining a platoon. In the architecture outlined in [1] the controller is organized in four layers. Starting from the top, The **network layer**, is responsible for flow of traffic on a highway network. Its objective is to prevent congestion and maximize throughput by dynamic routing of traffic. The **link layer** is responsible for maximizing flow on a section (link) ensuring that vehicles make their exits. It also manages incidents (reducing congestion) by commanding maneuvers, such as lane changes, to groups of vehicles [2]. The **coordination layer**, which resides in the vehicles, is responsible for coordinating the movement of platoons with their neighbors. The design of [5] uses protocols, in the form of finite state machines, that systematically execute maneuvers such as merging two platoons, splitting a platoon, and lane change. Finally, the **regulation layer** receives the coordination layer commands and translates them to throttle, steering and braking inputs for the actuators on the vehicle using a number of continuous time feedback control laws.

2. Architecture for degraded modes of operation

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. The only information the normal mode controller requires is the current state of the system. Because of the hierarchical structure the flow of the system state information is also arranged in a hierarchy; the higher levels

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

of the architecture receive more abstract information that 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 capabilities are not fixed. We partition the factors that affect the capability into two classes. The first class contains all the faults that occur on the vehicle or the roadside. We assume these faults are instantaneous and irreversible, therefore changing system capabilities in a discrete event. 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). We will say that faults affect *what* functions the system can perform (quantitative capabilities), while gradual degradation factors affect *how well* the system can perform these functions (qualitative capabilities). Overall, an extended architecture will need the following information: (1) current state, (2) quantitative capabilities, and (3) qualitative capabilities. Thus, three hierarchical structures are needed to monitor the behavior of the plant (Figure 1). 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 **Qualitative Capability Structure** carries the information about how well it can perform. The loop is closed by the **Control Structure** that will use all this information to produce control inputs to the plant.

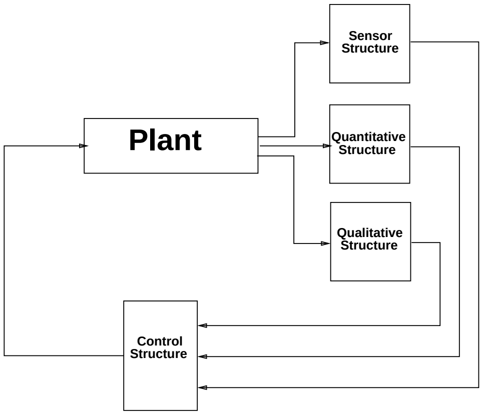

Figure 1: Overview of the Supervision Problem

At each level of the hierarchy performance criteria are defined to decide on optimal actions. These performance criteria reflect capacity and safety maximization in the descriptive language of the layer in question. By design, the higher levels of the hierarchy have access to information about a larger part of the system. Therefore they are better suited to control capacity. Lower levels have access to more detailed information and are better suited to control safety.

2.1. 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 vehicle 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 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.

Physical layer predicates. The resources of the physical layer include sensors, actuators and communication devices. If 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 $n_s + n_a + n_c$. This vector reflects which resources are functioning and which are not.

Regulation layer predicates. 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 be of dimension $n_{long} + n_{lat}$. Each law utilizes a set of physical layer resources. In order for the regulation layer controller to be functional all of its resources must be available. This 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 2 shows a possible mapping.

Regulation layer supervisor predicates. The regulation layer control laws represent resources used by the coordination layer to execute maneuvers (such as merging and splitting platoons and changing lanes). In order for the coordination layer to be able to invoke certain maneuvers, the relevant control laws should be operational. Formally, let n_{man} denote the number of maneuvers that may be requested by the coordination layer. Then the capability vector is a vector of zeros and ones of dimension n_{man}. The

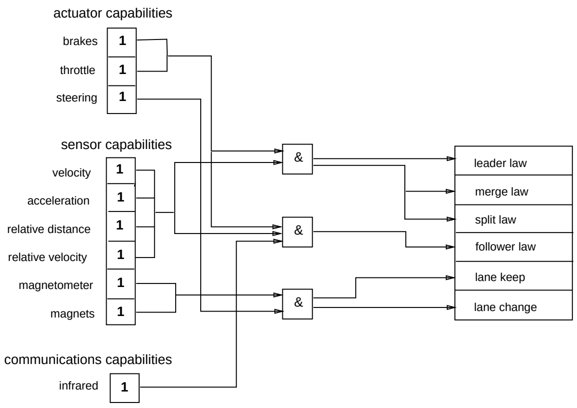

Figure 2: Physical and Regulation layer capabilities

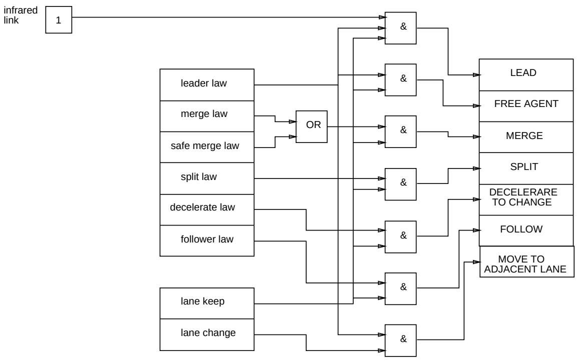

Figure 3: Regulation layer supervisor capabilities

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 [5], the map F_I can be seen in Figure 3.

Coordination layer supervisor predicates. The coordination layer of [5] requires a vehicle to be able to perform certain maneuvers and this capability is encoded in the regulation layer supervisor capability vector. In addition to execute the protocols that organize the maneuvers, the coordination layer needs access to communication capabilities. Formally, if the number of coordination strategies is n_{coord}, the capability vector for the coordination layer induces a mapping:

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

Here N stands for the maximum number of neighboring vehicles that need to cooperate in a maneuver.

Link layer supervisor predicates A highway link is partitioned into sections one lane wide and typically $2km$ long, entrances and exits. Within a section the link requires information about four possible events: section not blocked, section contains vehicles, section contains no vehicles queued behind an accident and section contains no emergency vehicles. These can be modeled as a vector of capabilities of dimension n_{sec} for each section. Let n_I denote the number of the relevant infrastructure faults and N_i the number of platoons in section i. Then for each section, each entrance and each exit 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.

2.2. Qualitative Capability Structure

The qualitative capability of the system is related to the system robustness. Gradual performance degradation can be caused by factors including adverse weather conditions such as rain, fog or snow and gradual hardware degradation such as brake wear. Qualitative capability parameters define the range of normal operation for each layer; for example, the maximum and minimum deceleration (physical layer) or the maximum tracking error of a controller (regulation layer). The qualitative performance requirements define the acceptable bounds on the capability parameters. Formally, 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:

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

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

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

Enhancing the robustness of the system, so that requirements of the control laws are met by the capability parameters, involves enlarging $\hat{\mathcal{C}}$ which may be achieved by on-line tuning of the controllers. If the desired $\mathcal{P}$ cannot be achieved by tuning, the corresponding predicate is set to zero and the supervisor selects another control law (see Figure 4).

2.3. Classification of faults by capability

A comprehensive list of faults pertaining to vehicle as well as infrastructure failures can be found in [6]. To simplify the task of designing degraded modes, the faults were grouped in six classes according to the

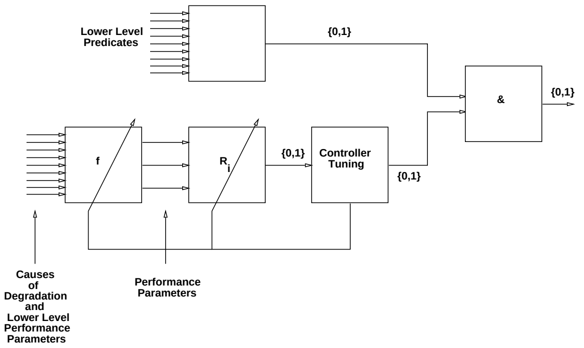

Figure 4: Introduction of robustness predicates

capabilities remaining to the vehicle/system after the fault has occurred.

Vehicle stopped/must stop: 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.

Vehicle needs assistance to get out: 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.

Vehicle needs no assistance to get out: 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).

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 he needs to alter the travel plan.

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.

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.

3. Control strategies

Based on the available capabilities of the system, the supervisor selects control strategies in order to respond to the fault. We discuss new control strategies needed to deal with the possible values of capability vectors within each fault class and for each layer.

3.1. Link layer design

The link layer controller for the extended architecture consists of two layers, a supervisor and a regulator. The supervisor takes as input the capabilities identified with each section. When a capability predicate of a section changes the supervisor issues 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 (1) incident avoidance, (2) emergency vehicle access, (3) congestion dissipation, and (4) create gap. Combinations of control objectives can be present at the same time within a link (due to multiple faults for example). They will be combined into a single command for the link layer regulator using a desired density/velocity profile generator of the link supervisor. The profile generator produces 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 regulator uses the profiles to generate commands for the individual platoons in the link. These commands (which include desired velocity and lane changes) are such that the traffic in the link converges to the velocity/density profiles.

3.2. 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.

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. If the faulty vehicle has lost its braking capability, then it uses *Aided Stop* strategy in which the vehicle in front of the faulty car applies gentle braking to bring both the vehicles to stop. If the faulty vehicle is a leader, then it executes a *Front Dock* maneuver to become a follower. For other subclasses, the faulty vehicle employs either a *Gentle Stop*, or a *Crash Stop* strategy. The names suggest severity of braking employed to bring the vehicle to a 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. We have also designed maneuvers for the vehicles stopped in the queue to *backup* and then *catch up* with the adjacent lane traffic so as to move out.

For faults in the class "vehicle needs assistance to get out" a strategy called *Take Immediate Exit* is executed by the coordination layer. 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 takes the next exit. This strategy is used by all subclasses except in cases where the vehicle capabilities limit its use. In particular, if the vehicle can not sense distant objects (needed for leader operation), *Take Immediate Exit - Escorted* is used. In this case, the faulty vehicle 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 platoon (called the *escorting vehicle*) now executes a TIE strategy to drop off the faulty vehicle at the nearest exit. Note that the link layer need not be involved for faults in this class. Finally, 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.

To implement above control strategies the coordination layer supervisor makes use of the normal mode maneuvers along with the following new maneuvers; *Forced Split*, *Emergency Lane Change* and *Front Dock*. In *Front Dock*, the last vehicle of the preceding platoon decelerates to join the faulty vehicle platoon as a leader. Thus it can be considered as a reciprocal of the normal mode *merge* maneuver. The maneuvers *Forced Split* and *Emergency Lane Change* are variations of the normal mode maneuvers *split* and *lane change*. Due to space limitations, we do not describe these maneuvers and strategies in detail. (c.f. [7])

3.3. Regulation layer control laws

Most of the coordination layer maneuvers described above can be performed by tuning the regulation layer feedback control laws designed for normal mode maneuvers. A few maneuvers such as front dock and platoon lane change (needed for TIE-E and queue management) need special control laws to be designed. We also need backward looking longitudinal distance and rate sensors on all vehicles.

4. 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 in the presence of faults and adverse environmental 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 a priori assumes fixed capabilities of the system, an assumption which is violated in degraded modes. We outlined an extended architecture designed to 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 addition maneuvers and control laws. Future work will entail further development and optimization of the design.

Acknowledgment: The authors would like to thank Akash Deshpande, 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 blowout 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." PATH Technical Report UCB-ITS-PRR-92-18, Institute of Transportation Studies, University of California, Berkeley, 1994.

[5] 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.

[6] J. Lygeros, D. N. Godbole, and M. E. Broucke, "Design of an extended architecture for degraded modes of operation of AHS." PATH Research Report, Institute of Transportation Studies, University of California, Berkeley, 1995.

[7] D. N. Godbole, J. Lygeros, E. Singh, A. Deshpande, and A. Lindsey, "Design and verification of coordination layer protocols for degraded modes of operation of AHS." (preprint) PATH Technical Report, Institute of Transportation Studies, University of California, Berkeley, 1995.