

UC Berkeley

Research Reports

Title

Optimized Vehicle Control/Communication Interaction in an Automated Highway System

Permalink

<https://escholarship.org/uc/item/8p23v0q4>

Authors

Hedrick, J. K.

Chen, Y.

Mahal, S.

Publication Date

2001-10-01

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

Optimized Vehicle Control/Communication Interaction in an Automated Highway System

J.K. Hedrick, Y. Chen, S. Mahal
University of California, Berkeley

**California PATH Research Report
UCB-ITS-PRR-2001-29**

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 of Transportation, Federal Highway Administration.

The contents of this report reflect the views of the authors who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California. This report does not constitute a standard, specification, or regulation.

Final Report for MOU 320

October 2001

ISSN 1055-1425

Optimized Vehicle Control/Communication Interaction in an Automated Highway System

Final Report

Submitted to: PATH (MOU 320)

J.K. Hedrick

Y. Chen

S. Mahal

Mechanical Engineering Department
University of California at Berkeley
Berkeley, CA 94720-1740

Contents

List of Figures	iv
List of Tables	vi
1 Introduction	1
1.1 Motivation Behind Automated Highways	1
1.2 Elementary Maneuvers in AHS	2
1.3 Control of Automated Vehicles	2
1.4 Robust Communication Protocols	3
2 String Stability	5
3 Vehicle Dynamics and Control	9
3.1 Simplified Vehicle Model	9
3.2 Spacing Control Strategies	11
3.2.1 <i>Semi-Autonomous Control</i>	12
3.2.2 <i>Using additional lead velocity and acceleration profiles</i>	13
3.2.3 <i>Including lead vehicle position information</i>	14
3.3 String Stability Analysis	15
4 Description of Communication System	17
4.1 Available Technologies	19
4.1.1 <i>Infrared Based Systems</i>	19
4.1.2 <i>Laser Based Systems</i>	19
4.1.3 <i>Radio Based Systems</i>	22
4.1.4 <i>Summary</i>	23

4.2 Existing Communication Architectures	23
4.2.1 <i>Infra-Red Communication System</i>	23
4.2.2 <i>Utilicom/Hughes System</i>	24
4.2.3 <i>WaveLAN System</i>	25
4.2.4 <i>MPI System</i>	26
4.2.5 <i>Communication System Fusion</i>	27
4.3 Types of Communication Delays	27
5 Interaction Between Control and Communication Software	29
5.1 Software Structure	29
5.2 Addressing Vehicles/Nodes	31
6 Robustness of Follower Control Towards Communication Delays	33
6.1 Model of Communication Architectural Delays	34
6.2 Effects of Delays on String Stability	35
6.2.1 <i>Delay in all Control Messages</i>	35
6.2.2 <i>Lead Vehicle Information Delay</i>	39
6.2.3 <i>Preceding Vehicle Information Delay</i>	43
7 Simulation Results	48
8 Nomenclature and Assumptions for Protocol Design	55
8.1 Nomenclature for Distributed Systems	55
8.2 Assumptions for the Communication System	56
9 Platoon Split Protocol	57
9.1 Initiating a Platoon Split	57
9.2 Servicing a Platoon Split Request	57
9.3 Protocol Design	59
9.3.1 <i>Lossy Channel Model</i>	60
9.3.2 <i>Abort Handler for the Platoon Split Maneuver</i>	69
10 Platoon Join Protocol	71
10.1 Initiating a Platoon Join	71
10.2 Servicing a Platoon Join Request	72
10.3 Protocol Design	72
10.3.1 <i>Case 1: Respondent Platoon in Front of Initiator Platoon</i>	72
11 Split and Join Maneuvers Redesigned for Dual Channel Radios	79
11.1 Redesigned Split Protocol	80
11.2 Combined Finite State Machine for the Platoon Split Protocol	87
11.3 Redesigned Join Protocol	88
11.3.1 <i>Case 1: Respondent Platoon Leading Initiator Platoon</i>	90
11.3.2 <i>Case 2: Initiator Platoon Leading Respondent Platoon</i>	97
11.4 Combined Finite State Machine for the Platoon Join Protocol	102

12 Field Tests	104
12.1 Implementation on Test Vehicles	104
<i>12.1.1 Controller</i>	<i>104</i>
<i>12.1.2 Communication System</i>	<i>104</i>
<i>12.1.3 Maneuver Protocols</i>	<i>105</i>
12.2 Maneuver Protocol Tests	112
12.3 Controller Tests	112
<i>12.3.1 Lead Vehicle Controller Tests</i>	<i>113</i>
<i>12.3.2 Split Transition Controller Test</i>	<i>115</i>
<i>12.3.3 Join Transition Controller Test</i>	<i>116</i>
13 Summary of Conclusions, and Future Work	118
13.1 Communication Delays	118
13.2 Maneuver Protocols	119
14 Bibliography	121

List of Figures

1.1	Platoon	3
3.1	Vehicle Model	10
3.2	Bode Plots for the Various Platooning Strategies	16
4.1	AHS Control Structure	18
4.2	Boomerang Transmission System	20
4.3	Double Boomerang Transmission System	21
4.4	Single Laser Variant of Two-Way Line of Sight System	21
4.5	Transmission Range	22
4.6	Token Bus Communication Architecture	25
4.7	MPI Communication Structure	26
4.8	MPI Wide Area Network	27
4.9	Communication Architectural Delay	28
5.1	Automated Highway Structure	30
5.2	Detailed View of Hybrid Control and Configuration Manager	31
6.1	Time Delay Example	35
6.2	Lead and Preceding Data Time Delay	36
6.3	Lead Vehicle Information Delay Bode Plots	42
6.4	Preceding Vehicle Information Delay Bode Plots	45
7.1	Comb_Sim Control Loop	49
7.2	Desired Trajectory	50
7.3	Spacing Errors Using a Perfect Communication System	51
7.4	Preceding Vehicle Triggers Control Loop with Swaroop's Gains	51
7.5	Preceding Vehicle Triggers Control Loop with Rizzuti's Gains	52
7.6	Modified Trajectory	52
7.7	Preceding Vehicle Triggers Control Loop w/ Modified Trajectory	53
7.8	Lead Vehicle Triggers Control Loop	54

(Platoon Split Protocol Finite State Machines)	
FSM 1A	62
FSM 1B	62
FSM 1C	63
FSM 1D	63
FSM 1A.b	67
FSM 1C.b	67
(Platoon Join Protocol Finite State Machines)	
FSM 2A	73
FSM 2B	73
FSM 2C	74
FSM 2D	74
(Dual-Channel Platoon Split Finite State Machines)	
FSM 3A	82
FSM 3B	82
FSM 3C	83
FSM 3D	83
11.1 Platoon Split Protocol	87
(Dual-Channel Join Protocol Finite State Machines)	
FSM 4A	91
FSM 4B	91
FSM 4C	92
FSM 4D	92
FSM 5A	98
FSM 5B	98
FSM 5C	99
FSM 5D	99
11.2 Platoon Join Protocol	103
12.1A Implemented Split Protocol	106
12.1B Implemented Split Protocol (continued)	107
12.2A Implemented Join Protocol	108
12.2B Implemented Join Protocol (continued)	109
12.2C Implemented Join Protocol (continued)	110
12.2D Implemented Join Protocol (continued)	111
12.3 Lead Car Control Test Data ($v_{prev} < v_{ideal}$)	113
12.4 Lead Car Control Test Data ($v_{prev} = v_{ideal}$)	114
12.5 Lead Car Control Test Data ($v_{prev} > v_{ideal}$)	114
12.6 Split Transition Data	116
12.7 Join Transition Data	117

List of Tables

4.1	Utilicom/Hughes Time Slot Division	25
11.1	Messages for the Split Maneuver	81
11.2	Messages for the Join Maneuver	89
	Messages for the Join Maneuver (continued)	90

Abstract

One of the main goals of an Automated Highway System environment is to increase the throughput of vehicles traveling on the highway. By moving vehicles in a platoon (a group of tightly spaced vehicles), the traffic flow capacity can be greatly increased. The control law developed for vehicles to safely travel in a platoon is dependent on the lead and preceding vehicle's velocity and acceleration profiles. This information guarantees string stability (i.e. spacing errors between vehicles do not increase down the chain of vehicles). These profiles are transmitted to the vehicle via wireless communication. Unfortunately, a perfect wireless communication does not exist. In this paper, the effects of various communication delays on string stability will be analyzed.

The concept of platooning in an Automated Highway System (AHS) allows a group of vehicles to share information across a wireless local area network (LAN). This sharing of information allows vehicles belonging to the same platoon to maintain a smaller inter-vehicular spacing that would otherwise be possible. Of course, once these platoon/LANs exist on the AHS, a method must exist to add vehicles to a platoon and also to remove vehicles from a platoon. This report also develops handshaking protocols that allow the LANs associated with each platoon to reconfigure themselves in response to any physical changes to the composition of the platoon. Since the LANs operate over a wireless communication system, these protocols are designed to be robust towards packet losses, as well as satisfying certain safety and liveness conditions.

Chapter 1

Introduction

This report concerns itself with the longitudinal maneuvers for platoons in an automated highway system. The challenges in designing the framework for these maneuvers is twofold. First is the control aspect. A controller should be able to not only to handle the physical requirements of these longitudinal maneuvers, but also to maintain some form of string stability to the traffic flow. That is, any disturbance in the spacing errors between vehicles should not be amplified from vehicle to vehicle as it propagates “backwards” (i.e. in the opposite direction of the traffic flow’s velocity) along the highway. The effect of communication delays on the string stability of such a controller is investigated. The second aspect is one of network reconfiguration. A platoon can be looked at as a collection of nodes/vehicles connected by a wireless local area network (LAN). Each time the composition of the platoon changes, with the addition of a vehicle for example, these changes must be reflected in the addresses used by the vehicle/nodes to communicate within the LAN. In order to accomplish this, a protocol must be designed to generate the necessary handshaking, and which also must be robust towards the possibility of message losses on the wireless LAN.

1.1 Motivation Behind Automated Highways

As the density of vehicles using California’s highways increases, we must look into ways of increasing the capacity of these highways. One method of doing this is to increase the density of the vehicle flow by decreasing the spacing between vehicles. Of course, the response time of human drivers is insufficient to safely navigate the highways under such conditions, so intelligent vehicle highway systems (IVHS), in which vehicles are computer controlled, are the next step.

Partners for Advanced Transit and Highways (PATH) work on longitudinal control of vehicles has focused on the concept of vehicle following. In order to maintain the close following

distance desired, it is necessary for vehicles to exchange information about their dynamic behavior. This situation lends itself to the creation of vehicle platoons, which are groups of vehicles that are engaged in following each other longitudinally, and are linked via a wireless local area network. Studies have shown (Varaiya, 1993) that by forming platoons, the throughput of a highway can be increased from 2000 vehicles per lane per hour to more than 6000 vehicles per lane per hour on a given highway¹.

1.2 Elementary Maneuvers in AHS

Once the concept of platoons is established, the question of how to add vehicles to platoons (once they have entered the highway), as well as how to combine and break up platoons must be considered. From this point on, the term platoon will be used to describe two or more vehicles linked longitudinally in a local area network, and the term free agent will be used to describe a single vehicle which does not belong to a platoon. Elementary maneuvers as suggested by Varaiya and Shladover (1991) are: a platoon join, in which either two platoons or one platoon and a free agent join to become a single platoon; a platoon split, in which a single platoon breaks up to become either two platoons or one platoon and a free agent; and a lane change, in which a free agent changes lanes. This report will focus on the platoon join and split maneuvers, and the control and networking challenges associated with them.

1.3 Control of Automated Vehicles

The main goal of the longitudinal controller is to safely follow its predecessor at some given distance. By using a radar to determine the spacing between two vehicles and communications for exchange of necessary data combined with the additional sensors and actuators on board, this can be achieved. The controller must not only guarantee stability of each individual vehicle but also the stability of all the vehicles traveling together in the platoon. The stability of groups of interconnected systems is known as string stability.

The string stability must be guaranteed, that is the spacing errors between the vehicles must attenuate down the platoon as illustrated in figure 1.1. Swaroop et. al. established the criteria for string stability for a platoon of vehicles. It was determined that for string stability the control laws must depend on both the relative position, velocity and acceleration of both the lead and preceding vehicles. This data except for the relative position of the preceding vehicle is relayed via wireless radio communications.

¹ Based on an average platoon size of 15 vehicles, with intra-platoon distance of 2 meters, inter-platoon distance of 60 meters, vehicle length of 5 meters, and speed of 72 km/hr.

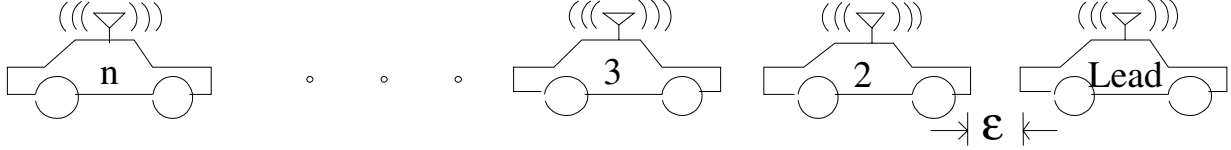


Figure 1.1: Platoon

where

$$\epsilon_i = x_i - x_{i-1} + L_i \quad (1.1)$$

and for string stability

$$\|\epsilon_2\|_\infty > \|\epsilon_3\|_\infty > \|\epsilon_4\|_\infty > \dots > \|\epsilon_{n-1}\|_\infty > \|\epsilon_n\|_\infty \quad (1.2)$$

For a platoon to move safely, it relies strongly on a good communication system. Since communication systems are not perfect and packet losses can occur, this paper analyzes the effects of this disturbance on string stability. The effect of intermittent loss of this data has been studied in simulation by Yu-Han Chen (1993). No analytical conclusions have been made. Chapter six presents an analysis of the effects of communication delays on string stability.

1.4 Robust Communication Protocols

A platoon join or split maneuver requires that the local area network (LAN) associated with a platoon be reconfigured. On a low level, this means reassigning addresses to the different nodes (i.e. vehicles) in that network. In any wireless LAN, however, there exists the possibility that one or more information packets may be lost. When this happens, the platoon must realize this and take appropriate action, otherwise the LAN may not be correctly configured at the end of the maneuver. Any node that does not possess a valid address, and is unable to divine the other addresses in its LAN, will be unable to send or receive further packets which are essential to the stable control of the platoon. The operation of such a protocol should be independent of the hardware over which it is implemented. However, understanding the capabilities and limits of that hardware is also important. Chapter four provides a review of available communication technologies.

Communication protocols for the platoon merge and split maneuvers that are robust towards packet losses were developed by Viswanath and Lindsey, “Design, Verification, and Simulation of Wireless Communication Protocols for the AHS”. Papers prior to this, such as Hsu et. al. (1991), have developed communication protocols for the merge and split maneuvers, but did not take into account the possibility of packet losses. Their protocols dealt with packet losses by including detection and packet retransmission states into their protocols. The logical correctness of these protocols were then verified using COSPAN, an automatic verification tool. Though the protocols developed by Viswanath and Lindsey recognized that there needed to be some form of interaction between the coordination layer (which contains the merge and split protocols) and the regulation layer (which contains the vehicle control software), they did not incorporate this

interaction into their protocols. In chapter five, the framework for the interaction between the vehicle coordination layer software and the control software is laid out.

Also, since changing the inter-vehicular distance during a platoon merge or a platoon split increases the potential for collision, the physical act of merging or splitting a platoon of vehicles should be performed when the most control information is available to all the vehicles involved. This detail has not been previously considered in the development of the protocols governing the platoon merge and platoon split maneuvers. The communication protocols that are developed in chapters nine through eleven take this factor into account by performing the physical acts of decreasing or increasing inter-vehicular spacing when all the vehicle/nodes involved in the maneuver belong to the same platoon/LAN.

Chapter 2

String Stability

String stability guarantees that the peak spacing errors will not amplify down the platoon. For example, if the platoon is string stable then the spacing error $\| \varepsilon_i \|_\infty$ seen by car i will always be less than the spacing error $\| \varepsilon_{i-1} \|_\infty$ seen by the preceding car $i-1$. To determine string stability, the relationship between the input and the output spacing errors must be determined.

To establish this relationship, let $G(s)$ be the spacing error transfer function relating the input-output spacing error ¹.

$$G(s) = \frac{E_i(s)}{E_{i-1}(s)} \quad (2.1)$$

The function $g(t)$ is its inverse Laplace transform, \mathcal{L}^{-1} , thus

$$\varepsilon_i(t) = g(t) * \varepsilon_{i-1}(t) \quad (2.2)$$

where $*$ represents the convolution operator.

From linear control theory, it is known that

¹The output spacing error is defined as the error seen by the current vehicle whereas the input spacing error is the error seen by the preceding vehicle.

$$\| g * \varepsilon_{i-1} \|_{\infty} \leq \| g \|_1 \| \varepsilon_{i-1} \|_{\infty} \quad (2.3)$$

where the norms in equation 2.3 are defined as

$$\| g \|_1 = \int_0^{\infty} | g(t) | dt \quad (2.4)$$

and

$$\| g \|_{\infty} = \sup_{t \geq 0} | g(t) | \quad (2.5)$$

From the relationship presented in equation 2.3, it can be seen that a sufficient condition for spacing error attenuation is

$$\| g \|_1 < 1 \quad (2.6)$$

Since time domain analysis can be quite difficult, it is desirable to switch to the frequency domain if possible. The next set of equations will establish the string stability criteria in the frequency domain.

Let's start with

$$\max_w \left| \int_0^{\infty} g(t) e^{-j\omega t} dt \right| \leq \max_w \int_0^{\infty} | g(t) | | e^{-j\omega t} | dt \quad (2.7)$$

Using the following property

$$| e^{-j\omega t} | = | \cos(\omega t) - j \sin(\omega t) | = 1 \quad (2.8)$$

the right hand side of equation 2.7 becomes

$$\int_0^{\infty} | g(t) | dt = \| g \|_1 \quad (2.9)$$

Using the definitions in equations 2.4 and 2.5 along with the definition of Laplace transform shown below in equation 2.10,

$$G(s) = \int_0^{\infty} g(t)e^{-st} dt \quad (2.10)$$

The left hand side of equation 2.7 is reduced to $\| G \|_{\infty}$. By substituting this and the result shown in equation 2.9 into equation 2.7, the following relationship is established.

$$\| G \|_{\infty} \leq \| g \|_1 \quad (2.11)$$

Combining this with

$$| G(0) | \leq \max_w | G(jw) | \quad (2.12)$$

it can be seen that

$$| G(0) | \leq \| G \|_{\infty} \leq \| g \|_1 \quad (2.13)$$

and

$$| G(0) | = \left| \int_0^{\infty} g(t) dt \right| \quad (2.14)$$

If the impulse response does not change signs, the inequality in the above equation becomes an equality as shown

$$\left| \int_0^{\infty} g(t) dt \right| = \int_0^{\infty} | g(t) | dt \quad (2.15)$$

concluding that

$$\| G \|_{\infty} = \| g \|_1 \quad (2.16)$$

From the prior result in the time domain shown in equation 2.6, it can be concluded that in order to apply frequency domain conditions, it is necessary that the impulse response $g(t)$ must not change signs, and $\| G \|_{\infty}$ must be less than one for string

stability. If $\|G\|_\infty$ is less than one, but the impulse response of $g(t)$ changes signs than no conclusions can be made in the frequency domain. Although if $\|G\|_\infty$ is greater than one, the system will be string unstable whether or not the impulse response changed signs. This result is not dependent on the impulse response sign change since from equation 2.13 it can be seen that if $\|G\|_\infty$ is greater than one than $\|g\|_1$ will be greater then one.

Chapter 3

Vehicle Dynamics and Control

A knowledge of the vehicle plant dynamics is required for the control design technique chosen, therefore a model of the vehicle's longitudinal dynamics is necessary. In this chapter, a simplified model is derived. This will lead into the string stability analysis of various control laws concluding that lead and preceding vehicle acceleration, velocity and relative position information is required.

3.1 Simplified Vehicle Model

Newton's first law says

$$m_i \ddot{x}_i = \sum F_v \tag{3.1}$$

where m_i is the mass of the vehicle, \ddot{x}_i is its acceleration and $\sum F_v$ is the sum of the forces acting on the vehicle. The forces acting on the body are shown on the freebody diagram in figure 3.1. These include road force on the front and rear wheels (F_{rf} , F_{rr}) and drag forces due to wind and grade (F_d). Additional variables shown on the figure are the front and rear brake torques (T_{bf} , T_{br}).

With the following assumptions

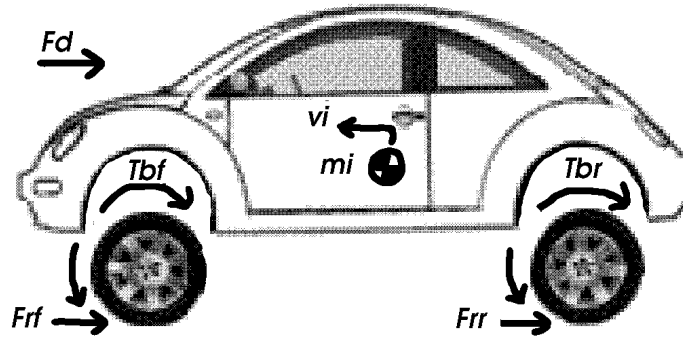


Figure 3.1: Vehicle Model

- The intake manifold dynamics are very fast compared to the vehicle dynamics.
- The torque converter is locked.
- There is no tire slip.
- The drive axle is rigid.

A simple vehicle dynamic model (Hedrick, 1991) for a vehicle's longitudinal motion can be derived:

$$\dot{v}_i = [k_1 T_{net}(\alpha_i, v_i) - k_2 T_L(v_i)] \quad (3.2)$$

where T_L is the load torque which comprises the external forces. The terms k_1 and k_2 are lumped together to reflect all terms to the vehicle's center of mass. These terms include gear ratios, mass of the vehicle and moments of inertias.

The first assumption allows the net engine torque, T_{net} , to be expressed directly as a function of throttle angle α_i and engine speed ω_i .

$$T_{net}(\alpha_i, \omega_e) \quad (3.3)$$

The second and third assumption state that there is pure rolling of the tires therefore the engine speed can be directly related to the vehicle's velocity by the gear ratio, r_i^* .

$$v_i = r_i^* \omega_e \quad (3.4)$$

If these assumptions hold, a T_{net} can be produced to exactly offset the load torques, and so any desired \dot{v}_i can be produced. In other words, set

$$u_i = [r_1 T_{net}(\alpha_i, v_i) - r_2 T_L v_i] \quad (3.5)$$

and you have

$$\dot{v}_i = u_i \quad (3.6)$$

3.2 Spacing Control Strategies

There are two main methods of control that have been studied in the AHS environment; constant spacing and constant headway. In constant spacing control, the desired spacing between vehicles is tracked whereas in constant headway control, a desired headway¹ is maintained.

Studies have shown (Swaroop, 1994) that with a headway time of .2 seconds, spacing control has at least 30% more traffic capacity than headway control. Reducing the headway down to .1 seconds, spacing control still has at least 25% more traffic capacity than headway control. In conclusion, the advantages of using constant spacing over constant headway control is to increase the throughput of vehicles on the highway, although constant headway control is more favorable since no external data is required.

For constant spacing control, external data is required for string stability. A wireless communication system is utilized to transfer this data. In this section, the control law to maintain constant spacing for various platooning strategies will be derived. For each of these strategies, the string stability will be determined.

¹Headway is the time it takes a vehicle to cover the distance between itself and the preceding vehicle.

3.2.1 Semi-Autonomous Control

In this control strategy, the preceding vehicle's acceleration and velocity information is available. The desired control law is

$$u_{id} = ka\ddot{x}_{i-1} - kv\dot{\varepsilon}_i - kp\varepsilon_i \quad (3.7)$$

where

$$\varepsilon_i = x_i - x_{i-1} \quad (3.8)$$

by modeling the actuator lag and signal processing delay as a first order filter

$$\tau\dot{u}_i + u_i = u_{id} \quad (3.9)$$

and substituting the relationship shown in equation 3.8, it can be seen that

$$\tau\left(\frac{d^3\varepsilon_i}{dt} + \dot{a}_{i-1}\right) + (\ddot{\varepsilon}_i + a_{i-1}) = u_{id} \quad (3.10)$$

by rearranging the variables, the above equation can be simplified to

$$\tau\frac{d^3\varepsilon_i}{dt} + \ddot{\varepsilon}_i = u_{id} - u_{i-1d} \quad (3.11)$$

by plugging equation 3.7 into the above equation 3.11, the transfer function $G(s)$ becomes

$$\frac{E_i(s)}{a_o(s)} = \frac{-\tau s}{\tau s^3 + s^2 + k_v s + k_p} \quad (3.12)$$

$$\frac{E_i(s)}{E_{i-1}(s)} = \frac{k_a s^2 + k_v s + k_p}{\tau s^3 + s^2 + k_v s + k_p} \quad (3.13)$$

It can be seen in the above equation that for all τ greater than zero, the maximum gain of the transfer function will be greater than one making the system string unstable. Although

one should look at the impulse response before looking at the bode plot, it is much easier to determine if the system is string unstable in the frequency domain. This is the reason why the bode plots were generated before the impulse response plots. If the maximum magnitude of the bode plot was less than one over the whole frequency range, than the impulse response was generated to check if any conclusions can be made in the frequency domain.

3.2.2 Using additional lead velocity and acceleration profiles

In this scenario, the lead vehicle's velocity and acceleration profiles along with the preceding vehicle's acceleration, velocity and position information is available. The desired sliding surface s_i incorporates this information and is defined as

$$s_i = \dot{\varepsilon}_i + q_1 \varepsilon_i + q_3 (v_i - v_o) \quad (3.14)$$

in order to make $\dot{s}_i = -\lambda s_i$, a synthetic control law is defined to be

$$u_i = \frac{1}{1 + q_3} [\ddot{x}_{i-1} + q_3 \ddot{x}_o - (q_1 + \lambda) \dot{\varepsilon}_i - q_1 \lambda \varepsilon_i - \lambda q_3 (v_i - v_o)] \quad (3.15)$$

by modeling the actuator lag including signal processing delays as a first order filter

$$\tau \dot{u}_i + u_i = u_{id} \quad (3.16)$$

and using the same approach in the previous section, where the above equation can be simplified to equation 3.11 and by plugging in the above desired control law shown in equation 3.14, the transfer function $G(s)$ becomes

$$\frac{E_i(s)}{a_o(s)} = \frac{-\tau s}{\tau s^3 + s^2 + \frac{(\lambda(1+q_3)+q_1)s}{1+q_3} + \frac{\lambda q_1}{1+q_3}} \quad (3.17)$$

$$\frac{E_i(s)}{E_{i-1}(s)} = \frac{\frac{1}{1+q_3}(s^2 + (\lambda + q_1)s + \lambda q_1)}{\tau s^3 + s^2 + \frac{(\lambda(1+q_3)+q_1)s}{1+q_3} + \frac{\lambda q_1}{1+q_3}} \quad (3.18)$$

Here it can be seen at zero frequency, all terms will be eliminated except for the last variables making the transfer function is equal to one. This will the system string stable in the weak sense.

3.2.3 Including lead vehicle position information

By including an additional variable, the lead vehicle position information, the sliding surface shown in equation 3.14 can be modified as shown in equation 3.19. This extra variable will make the system fully string stable instead of only string stable in the weak sense as shown in the analysis below.

$$s_i = \dot{\varepsilon}_i + q_1 \varepsilon_i + q_3 (v_i - v_o) + q_4 (x_i - x_o + \sum_{j=1}^i L_j) \quad (3.19)$$

in order to make $\dot{s}_i = -\lambda s_i$, we define a synthetic control law to be

$$u_i = \frac{1}{1 + q_3} [\ddot{x}_{i-1} + q_3 \ddot{x}_o - (q_1 + \lambda) \dot{\varepsilon}_i - q_1 \lambda \varepsilon_i - (q_4 + \lambda q_3) ((v_i - v_o) - \lambda q_4 (x_i - x_o + \sum_{j=1}^i L_j))] \quad (3.20)$$

by modeling the actuator lag as a first order filter, as shown in equation 3.9, the transfer function $G(s)$ becomes

$$\frac{E_i(s)}{a_o(s)} = \frac{-\tau s}{\tau s^3 + s^2 + \frac{\lambda(1+q_3)+q_1+q_4}{1+q_3}s + \frac{\lambda(q_1+q_4)}{1+q_3}} \quad (3.21)$$

$$\frac{E_i(s)}{E_{i-1}(s)} = \frac{\frac{1}{1+q_3}(s^2 + (\lambda + q_1)s + \lambda q_1)}{\tau s^3 + s^2 + \frac{\lambda(1+q_3)+q_1+q_4}{1+q_3}s + \frac{\lambda(q_1+q_4)}{1+q_3}} \quad (3.22)$$

The impulse response did not change signs therefore by looking at the bode plot, one can determine if this control law will be string stable. From equation 3.22, it can be seen that it is necessary that q_4 is greater than zero since $\frac{q_1}{q_1+q_4}$ must be less than one for string stability. The other condition that the gain must meet is $q_1 q_3 \geq q_4$ which is established by

Swaroop (Swaroop, 1994). Rizzuti (Rizzuti, 1994) analyzed discrete time string stability and determined an additional criteria that is necessary which is shown in equation 3.23.

$$\frac{q_1}{q_1 + q_4} + \frac{q_1 q_3 - q_4}{\lambda(1 + q_3)(q_1 + q_4)} < 1 \quad (3.23)$$

The bode plot generated in the next section used gains which satisfied each of the conditions discussed above.

3.3 String Stability Analysis

Figure 3.2 shows the bode plots and the peak responses of all three scenarios discussed in the previous section. If the lead vehicle information is not available, one can see that the system is clearly string unstable. The maximum magnitude is above 0 dB. Using only the lead vehicle's velocity and acceleration data, at low frequencies the gain is one making the system string stable in the weak sense. Using additional lead vehicle's position information, one can see that the gain is always less than one making the system string stable. In conclusion, it is necessary to have access to the lead vehicle's position, velocity and acceleration information for string stability.

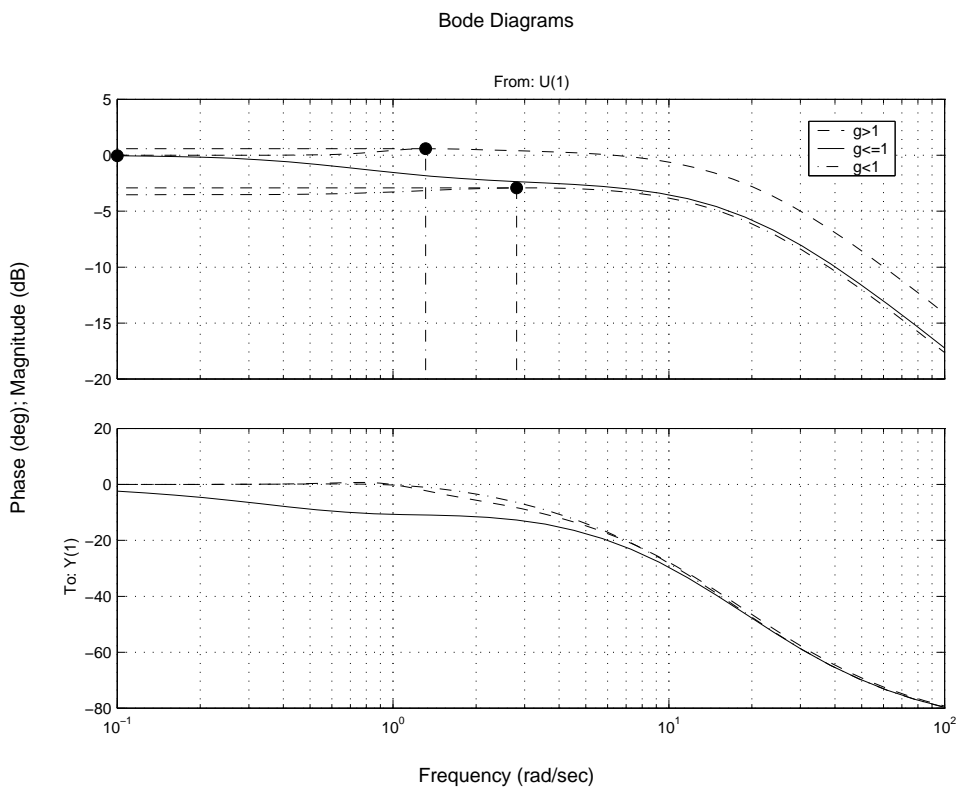


Figure 3.2: Bode Plots for the Various Platooning Strategies

Chapter 4

Description of Communication System

The exchange of information between various sources has become crucial to the development of a viable Intelligent Vehicle Highway System. Communicating data between vehicles, whether it be with an adjacent vehicle or with the leader of a platoon of vehicles, is becoming a large factor in the development of an automatic control for individual vehicles on the highway. Different aspects of control require slightly different methods of communication. As pointed out by Foreman (1995), communication of control information (the data needed to maintain the close spacing of vehicles) is very deterministic, and the quantity of information being exchanged is large. A person designing a control law for a given vehicle would like to be able to get necessary information at regular intervals, or at least to be given a maximum time limit when new information will be communicated. However, communications for maneuvers, whether they be for a single vehicle or for a group of vehicles, occurs in more or less random bursts. The channel activity in this case is also lower than for the control case.

In order to maintain string stability, the lead and preceding vehicle information must be communicated using wireless radios in the regulation layer (figure 4.1). Data exchange is not only required in the regulation layer, but it is also required in higher layers of the AHS architecture. For instance in the coordination layer, data communications is required to coordinate platoon maneuvers. Another examples where radio communications can be utilized is to inform to vehicles of driving conditions and emergencies.

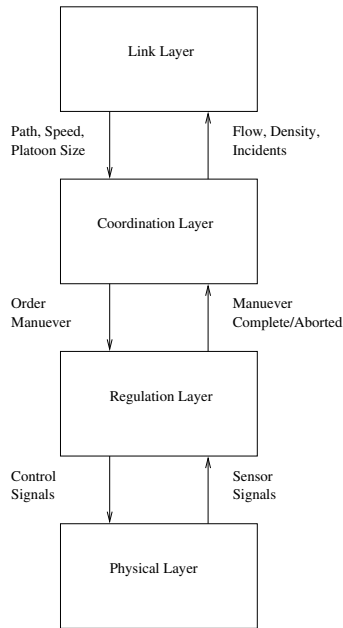


Figure 4.1: AHS Control Structure

Both vehicle-to-vehicle as well as vehicle-to-infrastructure communications are necessary in the AHS environment. There are two types of messages that need to be exchanged; control and command messages. The control messages contain information required to maintain a stable inter-vehicle spacing and are time critical, whereas the command messages contain information regarding a particular maneuver or an emergency.

An ideal AHS communication system will be broken into two networks; a local area network (LAN) and a wide area network (WAN). All intra-platoon (within a platoon) communications will operate within the LAN, and inter-platoon messages (i.e. messages sent between platoons or vehicle-to-infrastructure) are sent via the WAN. Control messages are always sent via the LAN whereas command messages can be sent either via LAN or WAN. The method of sending command messages vary from architecture to architecture whereas all existing PATH communication architectures send control messages within the platoon only. In this project, the main focus is on control message exchange.

There are many hardware options for routing information from vehicle to vehicle, and a general overview of the currently available “off-the-shelf” technologies can be found in Sachs and Varaiya (1993). More specific methods for using these various technologies are described in the remainder of this chapter. Most communication systems between vehicles fall into the categories of line-of-sight systems and broadcast systems. The line-of-sight systems consist primarily of optical devices such as infrared or lasers, while broadcast systems are mostly radio based. The broadcast systems are more complex in the sense that some type of protocol must be established so that two vehicles do not transmit simultaneously, thereby garbling both transmissions. Both line of sight and broadcast communication systems are currently being developed for used in the

AHS, and the first sections of this chapter attempts to provide an overview of what is potentially available for use in an automated vehicle.

The capability of the physical radio itself was not explored in our research since each year radio technology is evolving. For example, the data rates of wireless LAN radios in fall of 1997 are ten times that of radios in the spring of 1995[ref]. The primary interest is in the architectures that are available at PATH and the delays they introduce. All communication systems introduce some kind of transmission delay. An additional delay can also be present which is dependent on the architecture chosen for the AHS environment. These delays along with the overall communication architecture of the existing platforms at PATH will be discussed in the second part of this chapter.

4.1 Available Technologies

4.1.1 Infrared Based Systems

Theoretical work has been done by Foreman, et. al., which models an infrared LED transmitter and receiver. Since the geometry of existing highways limits the radius of curvature of the road to be greater than 1,150 feet, the maximum angle that a line-of-sight system needs to cover is about five degrees left or right. The optical beam width of the LED transmitter was then chosen to be a total solid angle of ten degrees. The work in this paper primarily models the power requirement, signal to noise ratio, and bit error rate of a transmitter/receiver pair. An adaptive baud rate is proposed to compensate for (and take advantage of) the variation of the bit error rate as a function of the baud rate and distance between transmitter and receiver. This system has the potential of providing two way communication between two vehicles by mounting a transmitter and receiver pair on the front and rear of each equipped vehicle.

4.1.2 Laser Based Systems

The Boomerang Communications Method (and its variants) is a line-of-sight communications system which uses a laser to transmit information between adjacent vehicles. The use of a spreading code in the transmission of data allows the laser to provide range information in addition to transmitting data. The “original” Boomerang Transmission System was a one-way communication system from vehicle A to vehicle B presented in a paper by Uchida, et. al. (1994),

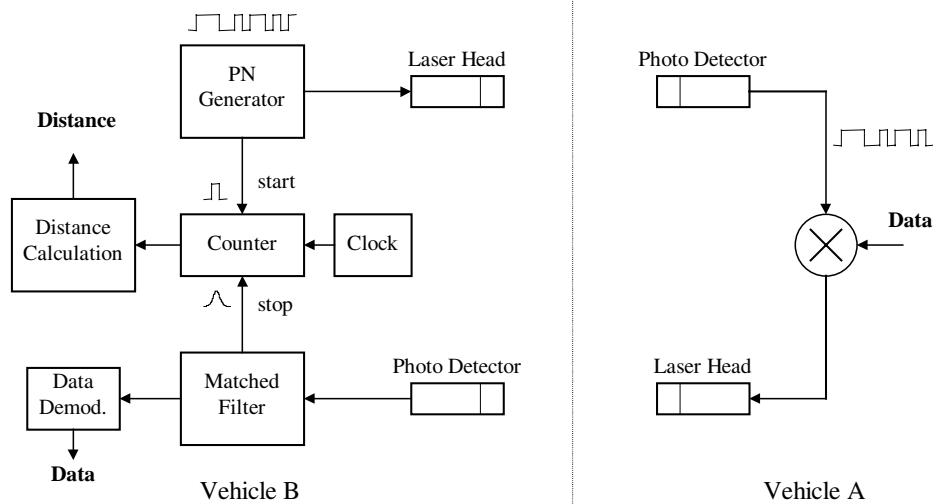


Figure 4.2: Boomerang Transmission System

Vehicle B transmits, via laser beam, a PN code to vehicle A. This triggers a counter in vehicle B that starts measuring the propagation delay time between the two vehicles. After vehicle A receives the transmission, it multiplies a data packet with the received PN code and transmits it back to vehicle B via a second laser beam. The returned signal is processed by a PN matched filter which computes the correlation between the generated PN signal with the returned signal. A peak signal in the filter output indicates the start of the returned PN code phase. The time of flight of the laser beam can be computed from the time interval between the generation of the PN code and the peak signal of the matched filter output. The distance, R , between the two vehicles is computed by the equation:

$$R = \frac{c \cdot N_c}{2 \cdot f_c}$$

where f_c is the counter pulse frequency, N_c is the value of the counter, and c is the velocity of light. The advantages of this type of system are:

- 1) robustness against interference and multipath fading,
- 2) generates very little interference to existing systems,
- 3) vehicle A does not need to know the PN code sequence of vehicle B,
- 4) it can still be used as a range finder even when vehicle A is not equipped with a transponder.

A two-way communication and ranging system, called the Double Boomerang Transmission System has been also been proposed by Mizui, et. al. (1994). This provides both participating vehicles with the range information of the other vehicle, and a two way communications link, using two optical transmitter/receiver pairs and two PN codes. A concept figure is drawn below,

4.1.3 Radio Based Systems

Unlike the previously mentioned line-of-sight optical technologies, the use of radio requires some type of networking protocol to prevent simultaneous transmissions from interfering with each other. Radio frequencies in the Gigahertz range are chosen to support inter-vehicle communications due to the high data rate that is required.

Some work has been done using Code Division Multiple Access (CDMA) to distinguish between platoons (closely spaced groups) of vehicles, and Time Division Multiple Access techniques for transmitting data between vehicles within a platoon. It has been proposed by Porche, et. al. (1992) to use a single frequency per platoon, and to implement a token-passing protocol to avoid a collision of data transmissions. In this configuration, each vehicle broadcasts its ID with its data. The ID numbers are based on the position that the vehicle occupies in the platoon. This paper also assumes a fixed number of vehicles in a platoon. To start the token passing protocol, the lead vehicle broadcasts its information on the channel, and starts a timer. Each following vehicle broadcasts its information only after receiving the broadcast of the vehicle immediately in front of it. When the last vehicle in the platoon has broadcast, the lead vehicle restarts the cycle. The lead vehicle also restarts the token cycle if the last vehicle does not broadcast before its timer reaches 50 milliseconds. With the preset cycle time of 50 ms, this configuration will only be able to support a limited number of vehicles in a platoon.

Hatakeyama and Takaba (1994) proposed two reservation based TDMA methods. It considers an infinite straight line of vehicles in which a time-frame is assigned to every vehicle. Within a frame, every vehicle can transmit its information to other vehicles in its transmission range without collisions. In each time frame, at least

$$2 \cdot \frac{l_r}{l_{\min}} + 1$$

time-slots are needed to accommodate all the vehicles within transmission range.

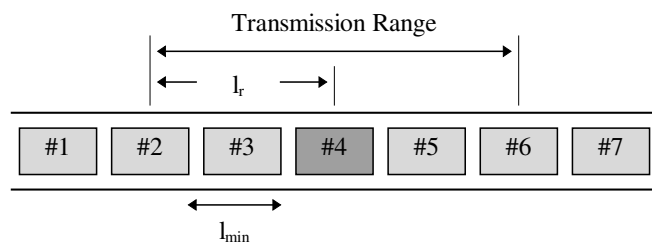


Figure 4.5: Transmission Range

For methods #1, the smaller numbered time-slot corresponds to the more forward vehicle. Each vehicle broadcasts its position information as part of its data during its time-slot. A vehicle decides its time slot number based on the most forward vehicle whose information could be received correctly in the previous frame. Knowing both the position of the most forward vehicle within transmission range, and its own position, a vehicle would chose

$$\left(\text{tn}_f + \frac{|x_s - x_f|}{l_{\min}} \right) + 1$$

as its time slot number in the next frame. In method #2, each vehicle broadcasts the time-slot numbers that it could receive correctly in the previous frame. As a result, every vehicle can deduce the time-slot numbers used within its transmitting range, and thus it chooses the unused time-slot number in that range in which to broadcast its data.

Kaltwasser and Kassubek (1994) explored a random access carrier-detecting method, called the COCAIN protocol. It operates under the assumption that the actual signal strength of a channel can be measured. In this configuration, each vehicle “probes” the channel when it wishes to transmit a data packet. If the channel is busy, the vehicle waits one “access delay” time before attempting to transmit again. If the channel is not busy, then the vehicle transmits its data packet, and must wait for one “re-access delay” time interval before it can attempt to transmit another packet. These two delays can be chosen to optimize the communication system with regards to:

- 1) Number of received packets during a certain time interval,
- 2) Number of received packet headers during a certain time interval,
- 3) Amount of time the channel is busy during a certain time interval.

4.1.4 Summary

In general, the line-of-sight and broadcast communication systems are each best suited for slightly different tasks within the scope of vehicle to vehicle communications. Broadcast systems are better than line-of-sight systems for getting information quickly to a large number of vehicles, but becomes less effective when many vehicles are vying for a single channel. Line-of-sight systems are ideal for transmitting information between adjacent vehicles, since there is no contention for that particular “channel”.

4.2 Existing Communication Architectures

There are many existing communication platforms at PATH. Each platform has its advantages and disadvantages. The different communication systems studied were the infrared radios, the WaveLAN radios, the MPI system and the Hughes/Utilicom system. In this section, a brief description of each platform is presented.

4.2.1 Infra-Red Communication System

The infra-red radios can only be used for point to point communications. The infra-red transceivers are placed on the front and the rear of a vehicle and are capable of short distance,

line of sight communications. The advantage of using this system is that each channel is unique for a pair of vehicles which helps to reduce the possibility of interference.

Interference is a major factor in choosing a communication system. Packet drops and loss of communications are largely be due to interference. Since the infrared radios use direct communications, this reduces interference making the radios robust. The problem of using this type of system in the platooning environment is the process of sending information from a vehicle to another vehicle that is not directly behind or in front of it. The information must be relayed back causing delays.

Since in the AHS environment the platoon leader’s information must be transmitted to all the platoon members, it must be passed down through all the preceding vehicles. For example, if vehicle 1 is sending a message to vehicle 2, vehicle 3 can be sending a message to vehicle 4 at the same time, but each vehicle must relay the lead vehicle information to it’s follower. Not only lead vehicle information, but also command messages must be relayed through all the vehicles in between the transmitting vehicle and the lead vehicle.

Besides the disadvantage of sending command and lead vehicle messages, one of the main advantages of using the infrared system is the capability of sending messages simultaneously. Another advantage is that each channel is allocated the full band allowing for higher data rates. The maximum communication speed of this system is 1.23 Megabits per second. The baud rate changes dynamically in fixed steps based on the channel performance with a minimum speed of 19.2 KBits per second.

4.2.2 Utilicom/Hughes System

The Utilicom/Hughes System was successfully implemented in the 1997 AHS demonstration in San Diego. The demonstration consisted of an eight vehicle platoon traveling at 65 miles per hour with inter-vehicle spacing of 6 meters¹. By using a combination of both time slots and channels, this system provides ten vehicles the opportunity to transmit data every 20 milliseconds.

In table 4.1, the communication architecture of the Utilicom/Hughes radios is shown. By using frequency or code division multiplexing, two or more radios can simultaneously transfer data within a single time slot. In the first time slot, the lead vehicle broadcasts it’s information while all other vehicles listen to the broadcast. The next time slot is divided over three channels. A radio can either transmit or receive, but it can not do both at the same time.

	Time Slot 0	Time Slot 1	Time Slot 2	Time Slot 3
Vehicle ID #1	Tx on channel 1	Rx on chan 1,2,3	Rx on chan 1,2,3	Rx on chan 1,2,3
Vehicle ID #2	Rx on channel 1	Tx on channel 1	idle	idle
Vehicle ID #3	Rx on channel 1	Rx on channel 1	Tx on channel 1	idle
Vehicle ID #4	Rx on channel 1	idle	Rx on channel 1	Tx on channel 1

¹ During the testing phase, an eight vehicle platoon with inter-vehicle spacing of 4 meters traveling 65 miles per hour was also successfully executed.

Vehicle ID #5	Rx on channel 1	Tx on channel 2	idle	Rx on channel 1
Vehicle ID #6	Rx on channel 1	Rx on channel 2	Tx on channel 2	idle
Vehicle ID #7	Rx on channel 1	idle	Rx on channel 2	Tx on channel 2
Vehicle ID #8	Rx on channel 1	Tx on channel 3	idle	Rx on channel 2
Vehicle ID #9	Rx on channel 1	Rx on channel 3	Tx on channel 3	idle
Vehicle ID #10	Rx on channel 1	idle	Rx on channel 3	Tx on channel 3

Table 4.1: Utilicom/Hughes Time Slot Division

After the lead vehicle sends data in the first time slot, during the next three slots it receives data from either channel 1, 2 or 3. It alternatives between these channels every 20 milliseconds allowing each vehicle to send command messages every third cycle. In this case, the command messages are sent every 60 milliseconds.

This setup is very efficient for platoon sizes up to ten vehicles. Unfortunately the platoon size is pre-programmed, and the platoon size can not be changed dynamically. This is a big disadvantage since in the AHS environment, cars will be joining onto and/or splitting from platoons resulting in changes to the platoon size.

4.2.3 WaveLAN System

The WaveLAN system consist of AT&T's WaveLAN 900 MHz CDMA radios and uses spread-spectrum technology. The WaveLAN radio is capable of data transfer rates up to 2 Megabits per second, and tests have shown that the radios can reliably exchange data up to 100 meters.

The WaveLAN system is setup as a token bus architecture as shown in Figure 4.6. This scheme guarantees each car in the platoon an opportunity to transmit once every token cycle which is set to 20 milliseconds. The size of the time slots to transmit data is dependent on the platoon size. The time slots are evenly sliced among all the vehicles within a platoon giving each vehicle an opportunity to transmit data every control period. For a five vehicle platoon, the size of the time slot is 4 milliseconds.

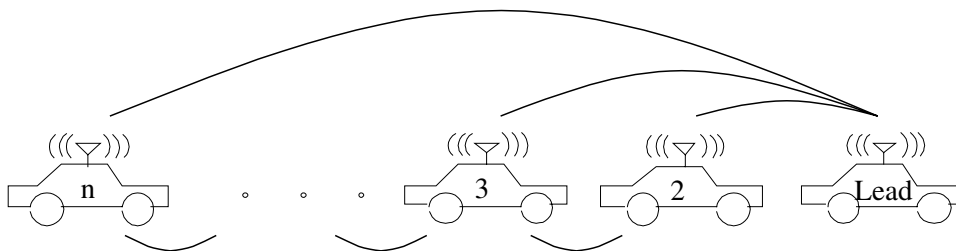


Figure 4.6: Token Bus Communication Architecture

In the token bus architecture, the first time slot is allocated to the lead vehicle. The lead vehicle broadcasts it's position, velocity and acceleration plus some additional data to all the other vehicles in the platoon within it's time slot. The next time slot allows the second vehicle to send control messages to its follower and command messages to the lead vehicle and so on.

Depending on the token cycle time and the size of the platoon, it is very easy to compute the size of the time slots. This allows us to easily determine the time between the vehicle's data transmissions. Using this information, we can solve the delays that will be encountered in this scheme. These delays will be covered in the next section.

4.2.4 MPI System

The MPI system that was delivered to PATH in 1997. This system supports the AHS requirements of both vehicle-to-vehicle and vehicle-to-infrastructure communications. It consists of the two networks discussed earlier; LAN and WAN. The overall system architecture is shown in figure 4.7.

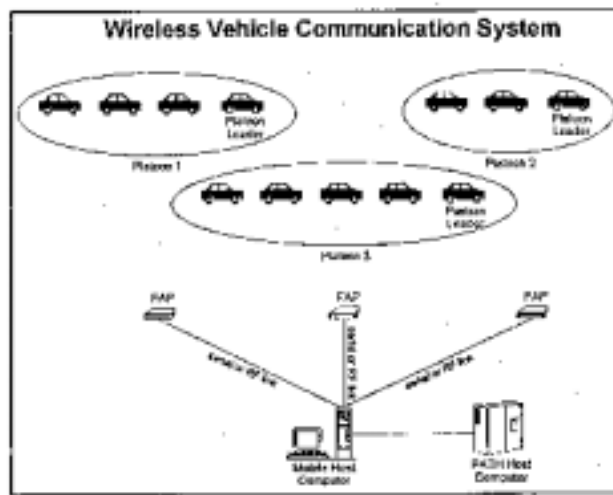


Figure 4.7: MPI Communication Structure

Each platoon makes up a single LAN supporting up to ten vehicles. The system is configured to allow only control messages to be transmitted over the LAN. These messages are updated every 20 milliseconds. The messages are sent similar to the WaveLAN system where each platoon member can receive data from both the platoon leader as well as the car immediately in front of it.

The WAN handles all vehicle to vehicle and vehicle to infrastructure communications. By using Fixed Access Points (FAP), each vehicle can communicate with a mobile host computer which in turn can communication with the PATH host computer as illustrated in figure 4.8. This allows messages about the activity on the highway to be sent to a central location. It also handles all the communication media necessary for longitudinal maneuvers (i.e. joins, splits and lane changes) and also platoon to platoon communications.

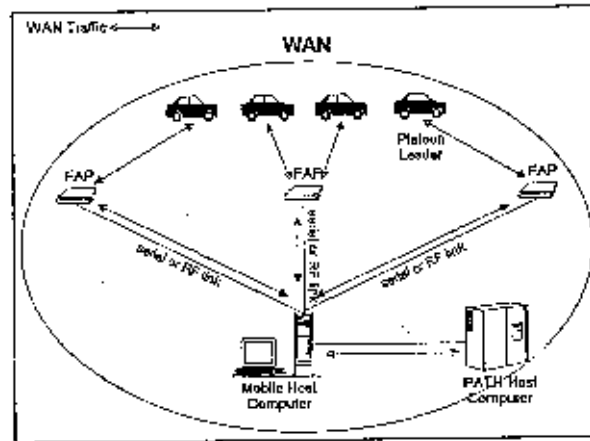


Figure 4.8: MPI Wide Area Network

A downside to this system is that messages can not be sent directly from a platoon member to the platoon leader. The message must be past via the WAN to the mobile host computer which will relay it to the platoon leader.

4.2.5 Communication System Fusion

Each system has it's pros and cons. None of the systems described above except for MPI can fully support the AHS environment. As data traffic increases, even the MPI system might not be a good choice since the command messages are sent over the WAN. By fusing two systems together, an ideal system can be constructed. For instance, by using the infrared radios along with the WaveLan, one can minimize the delay of the information sent between vehicles for intra-platoon communications.

4.3 Types of Communication Delays

There will always be some kind of delay in information required by the control law for string stability. This delay can be due to packet losses, transmission time and/or the type of architecture chosen.

Delays due to packet losses are intermittent, and methods to study the effects of these delays on the stability of system are being explored. If the communicated message is corrupted, it will be dropped and treated like a packet loss. The control algorithm will use the latest data received. This data will not be updated unless valid data is received.

Transmission time is refer to the time associated with sending bits over the network. This delay can be determined by the radio's bit rate. It is the time it takes for a bit to go from it's source to it's destination. This delay does not take into consideration packet losses or data collisions.

Delays associated with the communication architecture chosen will refer to as “Communication Architectural Delays”. These delays consist of the amount of time the transmitted data has to wait in a buffer before being used in by the control loop as illustrated in figure 4.9. These delays vary from architecture to architecture.

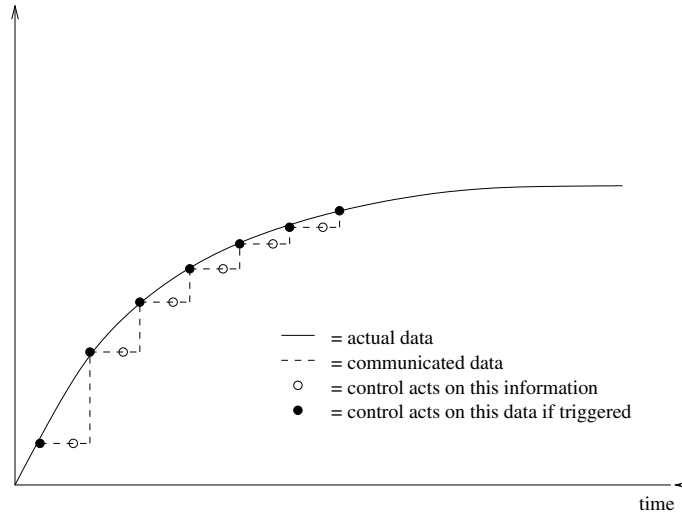


Figure 4.9: Communication Architectural Delay

In our analysis, the delays associated with packet losses are not covered. The delays due to transmission time and computational processing time are assumed to be much smaller than the delays due to the architecture chosen therefore they have been neglected. The analysis only focuses on the effects of “Communication Architecture Delays” on string stability.

Chapter 5

Interaction Between Control and Communication Software

5.1 Software Structure

The structure of the components of the automated highway is shown in figure 5.1. The leftmost column consists of the components belonging to the roadside. The middle column consists of the components in the wireless communication network in each vehicle. The rightmost column describes the components that control vehicle movement and maintain their relative positions on the highway. A higher position in this diagram indicates that the component operates on a higher level of abstraction in the AHS. The two sets of arrows leading from the communication network layer to the vehicle coordination layer and vehicle regulation layer indicates that control messages can pass directly to the vehicle controllers without going through the vehicle coordination layer. This dissertation is concerned primarily with the vehicle coordination layer and the vehicle regulation layer. The finite state machine (FSM) that coordinates the platoon split and platoon join maneuvers operates in the coordination layer, while the hybrid vehicle controller (see Chen, 2000) operates in the vehicle regulation layer.

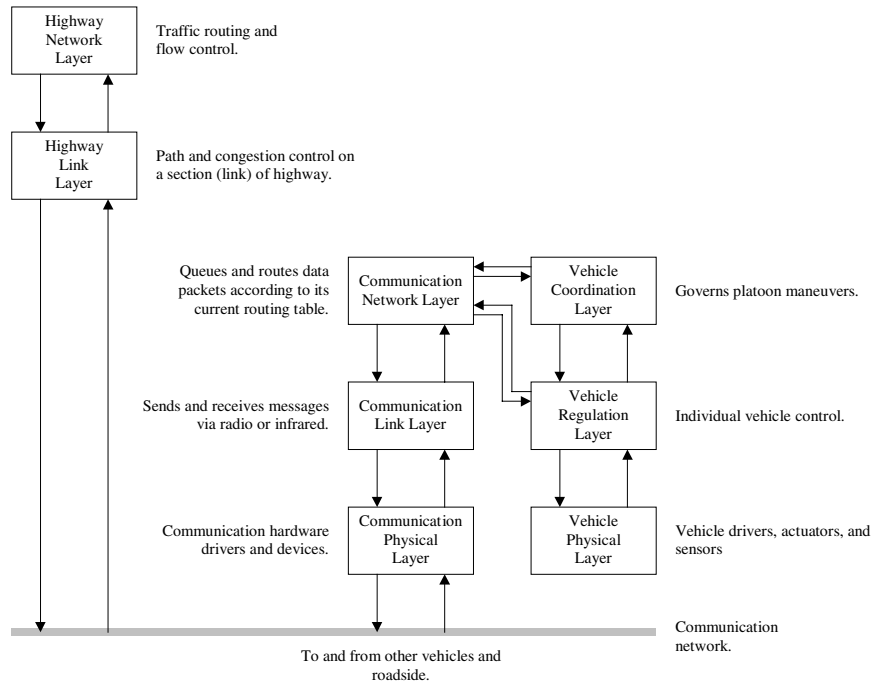


Figure 5.1: Automated Highway Structure

In an effort to decouple the design of the hybrid controller and the finite state machine that governs the behavior during join or split maneuvers, the two subsystems interact with each other using shared memory. The FSM's governing the maneuvers act as an "upper layer", which directs which control mode should be used by issuing control commands (see figure 5.2) which are then read by the hybrid controller. Since this FSM handles the reconfiguration of the platoons during a maneuver, it will be referred to as a configuration manager. The hybrid control system responds to these commands from this upper layer, and if appropriate indicates whether or not a particular "control task" is achieved in a particular mode by changing its control mode flags which are then read by the coordination layer. For lead and follower modes, there is no control task that can be "completed". However, for the transition control modes, whether it be moving vehicles apart or moving them closer together, it is necessary to know when these transitions are complete, and hence the use of control mode flags. In order to avoid contention for shared memory, two sets of memory locations are used. One is written to by the maneuver FSM and read by the hybrid controller, the other is written to by the hybrid controller, and read by the configuration manager. As a result of this decoupling, the configuration manager can ignore the dynamics of the controller, and the controller can be designed independently of the maneuver protocols.

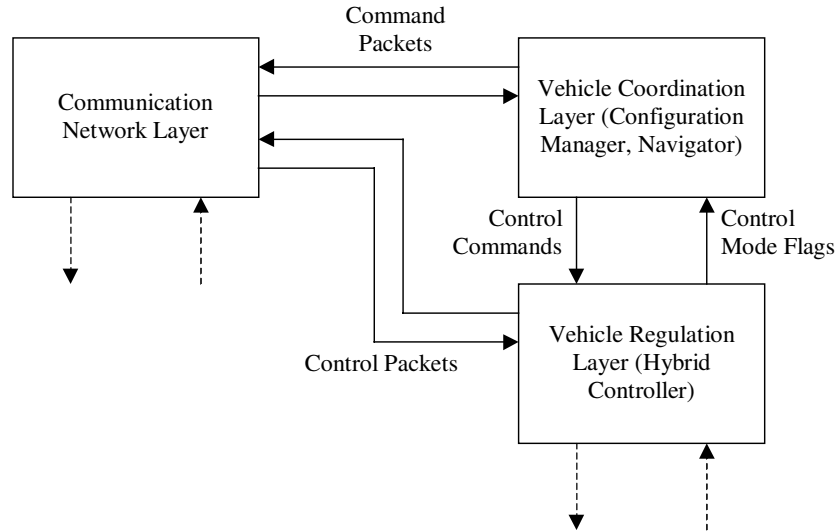


Figure 5.2: Detailed View of Hybrid Control and Configuration Manager

The configuration manager receives input from another process, called a “navigator” in figure 5.2, that would combine information about the desired destination of the vehicle with information from the roadside regarding traffic flow and recommended routes. This navigator process would then decide what maneuvers are needed (in the short term) for the vehicle to reach its destination. If the navigator process decided a lane change was needed, for example, it would then determine what maneuvers were necessary to make the vehicle in question a free agent, then order the lane change from whatever process governed that maneuver. The navigator acts as an overseer to the configuration manager. In this dissertation, we will be concerned with the function of this navigator process only in the sense that it must eventually exist in order for a vehicle to autonomously plot its own course along the highway, and that an interface must be designed into the configuration manager process to address this eventuality.

The communication network layer takes care of actually sending the data that the configuration manager and hybrid controller generate. In the test vehicles, where a fully developed network layer has not yet been implemented, this task is accomplished by a process called SendChild, which has a single slot sending queue. When either the configuration manager or controller needs to send data, it places its data into this queue. When the queue is empty, then the components is able to send new data. This SendChild process also waits for and de-packetizes any packets coming in from the wireless communication system, and sends the incoming data to the configuration manager or the controller, as appropriate.

5.2 Addressing Vehicles/Nodes

In order to properly exchange messages between vehicles/nodes in a network, each of them must be assigned a unique address. One method that seems to work well is to address each vehicle/node first by its platoon/LAN ID, and then by its node ID. Since the LAN ID for a platoon should be unique, the license plate number of the lead vehicle in the platoon is a good candidate for use as a LAN ID. The node ID for a vehicle within a platoon, however need not be

unique. Therefore, the position of the vehicle in the platoon is a good candidate for use as a node ID.

In addition to this address information, it is useful for the vehicle to know the size of the platoon it belongs to. The combination of platoon ID, node ID, and platoon size allows any vehicle to infer its relation to the vehicles around it. A vehicle is now able to determine whether it is a platoon leader, or a free agent. It is also able to determine, for example, how many split maneuvers are necessary for it to become a free agent.

Chapter 6

Robustness of Follower Control Towards Communication Delays

Each communication architecture will present some kind of delay between the actual measurement and when the data is used. In this project, the motivation was to see how large these delays can become before the system is string unstable.

Logically, it would seem if the platoon is moving at a constant velocity then any loss of data would not effect the performance of the platoon. Also if the change in acceleration is really slow, a few packet drops would be acceptable. Our main concern in the initial stages of this project was the amount of packet drops or overall acceptable delays during platoon maneuvers. Since all the vehicles behind the maneuvering vehicle must accelerate and decelerate during the course of the maneuver, the loss of communications can cause major effects. The focus was to determine limits on the range of acceptable delay times.

The initial results showed that no delays were acceptable in the communication system. As the research continued, this result was verified and validated through simulations. This twist changed the focus of the research. Instead of worrying about what happens during longitudinal maneuvers, the overall performance of the platoon in any condition became our main concern.

This chapter is broken into two sections. The first section explains how the communication delay is modeled. The next section analyzes the effects of these delays on string stability for three cases. One of the cases reviews the scenario where only the lead vehicle information is delayed while the other case covers the effects of preceding vehicle's velocity and acceleration

information delay. Finally, the third case reviews how the system works in reality. This case shows the effects of the combination of both the lead and preceding vehicle data delay. For each case, the necessary equations are derived for analysis and conclusions are based on the string stability criteria established in chapter 2.

6.1 Model of Communication Architectural Delays

As mentioned in section 4.2, there will be some “Communication Architectural Delay” in the system. Each vehicle in the platoon has the opportunity to send data once during the control loop period¹. If the control loop was synchronized with the communication data exchange, the only delays encountered will be due to transmission and computational processing time. Since the control loop depends on data from both the lead and preceding vehicle, the system can not be fully synchronized. There will be some delay which is dependent on the architecture since both vehicles can not transmit data to a particular vehicle at the same time².

There are three main cases the control law can be executed. The first case is fully independent of the radio transmissions. The other two cases are dependent on the time it receives a data packet. Since the period of both lead and preceding vehicle transmission is equal the control loop cycle, the control loop can be triggered by an interrupt generated when either the lead or the preceding vehicle data packet is received (figure 6.1).

Depending on which data triggers the control loop, the other data will have some kind of delay. For the token bus architecture these delays are known. For example for platoon size of five vehicles, it can be seen that if the control loop of vehicle 4 was triggered when it received information from vehicle 3, the lead information will have a delay of 8 milliseconds. In other words, the lead information was sent 8 milliseconds prior to the preceding vehicles information, but it has been sitting in some buffer waiting to be used.

This same logic can be used to determine the delays that would be encountered in the preceding vehicle information if the control loop is triggered when the lead vehicle data packet is received.

¹The control loop period can be set to any size. Throughout our analysis, the standard control loop period of 20 millisecond was used.

²All radios used in the existing PATH communication systems discussed in the previous section allow each vehicle to only listen to one channel. If two sources send data on the same channel, there will be a data collision causing both packets to be dropped.

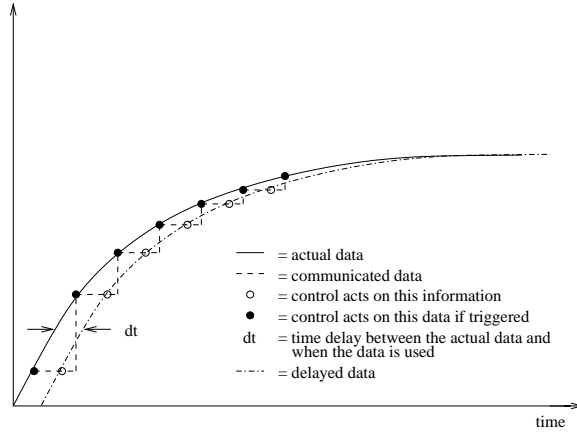


Figure 6.1: Time Delay Example

To model the delays associated in the three different scenarios, a pure time delay was used. The Laplace transformation of a pure time delay shown in equation 6.1 was used in the following sections.

$$\mathcal{L}(a_o(t - t_d)) = e^{-st_d} A(s) \quad (6.1)$$

Note in equation 6.1, t_d is equal to dt on figure 6.1. For no time delays, t_d will equal zero therefore the Laplace transformation will be reduce to equation 6.2.

$$\mathcal{L}(a_o(t)) = A(s) \quad (6.2)$$

6.2 Effects of Delays on String Stability

6.2.1 Delay in all Control Messages

In this section, the spacing error transfer function between vehicle i and vehicle $i-1$ is established when delays in both preceding vehicle and lead vehicle information are encountered as shown in figure 6.2. Note this is the case of how the vehicles are running today. The control law cycle is independent of when the control messages are received. There might be cases

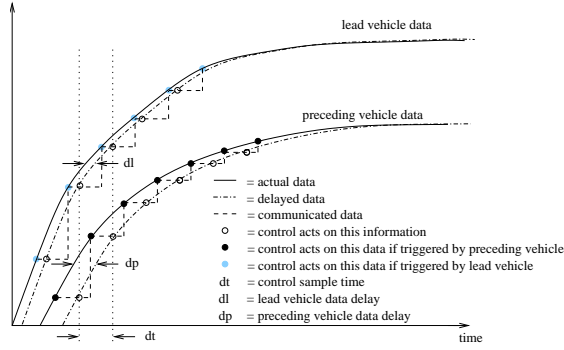


Figure 6.2: Lead and Preceding Data Time Delay

where the control law cycle is in synchronize with the reception of certain control messages. The next sections will cover these cases.

The time delays are defined as the following

- τ_{d1} is the timing delay of the preceding vehicle information seen by vehicle i
- τ_{d2} is the timing delay of the lead vehicle information seen by vehicle i
- τ_{d3} is the timing delay of the preceding vehicle information seen by vehicle $i-1$
- τ_{d4} is the timing delay of the lead vehicle information seen by vehicle $i-1$

Note dp and dl shown in figure 6.2 are equivalent to τ_{d1} & τ_{d3} and τ_{d2} & τ_{d4} respectively.

The next set of equations will establish the spacing transfer function with communication delays.

Using the property established in equation 3.11

$$\tau \frac{d^3 \varepsilon}{dt} + \ddot{\varepsilon} = u_{i_d} - u_{i-1_d} \quad (6.3)$$

where

$$u_{i_d} = \frac{1}{1 + q_3} [\ddot{x}_{i-1}(t - \tau_{d1}) + q_3 \ddot{x}_o(t - \tau_{d2}) - (q_1 + \lambda)(v_i - v_{i-1}(t - \tau_{d1})) - q_1 \lambda \varepsilon_i]$$

$$-(q_4 + \lambda q_3)((v_i - v_o(t - \tau_{d2})) - \lambda q_4(x_i - x_o(t - \tau_{d2}) + \sum_{j=1}^i L_j)] \quad (6.4)$$

and

$$u_{i-1d} = \frac{1}{1 + q_3} [\ddot{x}_{i-1}(t - \tau_{d3}) + q_3 \ddot{x}_o(t - \tau_{d4}) - (q_1 + \lambda)(v_i - v_{i-1}(t - \tau_{d3})) - q_1 \lambda \varepsilon_i - (q_4 + \lambda q_3)((v_i - v_o(t - \tau_{d4})) - \lambda q_4(x_i - x_o(t - \tau_{d4}) + \sum_{j=1}^i L_j))] \quad (6.5)$$

After rearranging the terms, it can be seen that

$$\tau \frac{d^3 \varepsilon_i}{dt} + \ddot{\varepsilon}_i + \left(\lambda + \frac{q_1 + q_4}{1 + q_3}\right) \dot{\varepsilon}_i + \left(\frac{\lambda(q_1 + q_4)}{1 + q_3}\right) \varepsilon_i = \frac{1}{1 + q_3} [\lambda q_1 \varepsilon_{i-1} + \zeta] \quad (6.6)$$

where ζ contains all the time delay terms shown in equation 6.7.

$$\begin{aligned} \zeta = & q_3(a_o(\tau + \tau_{d2}) - (a_o(\tau + \tau_{d4})) + (q_4 + \lambda q_3)(v_o(\tau + \tau_{d2}) - (v_o(\tau + \tau_{d4}))) + \lambda q_4(x_o(\tau + \tau_{d2}) - (x_o(\tau + \tau_{d4}))) \\ & + a_{i-1}(\tau + \tau_{d1}) + (\lambda + q_1)(v_{i-1}(\tau + \tau_{d1})) \\ & - a_{i-2}(\tau + \tau_{d3}) - (\lambda + q_1)(v_{i-2}(\tau + \tau_{d3})) \end{aligned} \quad (6.7)$$

Let's break this equation even further to simplify the discussion in the following sections since portions of equation 6.7 will be used to establish the transfer functions later.

$$\zeta = \beta + \gamma + \theta \quad (6.8)$$

where β are the terms associated with the lead vehicle delay seen by both vehicles i and $i-1$, γ are the terms associated with the preceding vehicle delay seen by vehicle i , and finally θ are the terms associated with the preceding vehicle delay seen by vehicle $i-1$.

$$\beta = q_3(a_o(\tau + \tau_{d2}) - (a_o(\tau + \tau_{d4})) + (q_4 + \lambda q_3)(v_o(\tau + \tau_{d2}) - (v_o(\tau + \tau_{d4}))) + \lambda q_4(x_o(\tau + \tau_{d2}) - (x_o(\tau + \tau_{d4}))) \quad (6.9)$$

$$\gamma = a_{i-1}(\tau + \tau_{d1}) + (\lambda + q_1)(v_{i-1}(\tau + \tau_{d1})) \quad (6.10)$$

$$\theta = -a_{i-2}(\tau + \tau_{d3}) - (\lambda + q_1)(v_{i-2}(\tau + \tau_{d3})) \quad (6.11)$$

Now let's take the Laplace transform,

$$H_{11}E_i(s) = \frac{1}{1 + q_3}[G_1E_{i-1}(s) + G_2A_o(s) + G_3A_{i-1}(s) - G_4A_{i-2}(s)] \quad (6.12)$$

where

$$H_{11} = \tau s^3 + s^2 + (\lambda + \frac{q_1 + q_4}{1 + q_3})s + (\frac{\lambda(q_1 + q_4)}{1 + q_3}) \quad (6.13)$$

$$G_1 = \lambda q_1 \quad (6.14)$$

$$G_2 = \frac{1}{s^2}(e^{-\tau_{d2}s} - e^{-\tau_{d4}s})(q_3s^2 + (q_4 + \lambda q_3)s + \lambda q_4) \quad (6.15)$$

$$G_3 = \frac{e^{-\tau_{d1}s}}{s}(s + (\lambda + q_1)) \quad (6.16)$$

$$G_4 = \frac{e^{-\tau_{d3}s}}{s}(s + (\lambda + q_1)) \quad (6.17)$$

The above equation 6.12 is the general solution in the frequency domain and equation 6.6 is the general solution in the time domain.

Note if both vehicles saw the same delay in both lead and preceding vehicle information then $\tau_{d2} = \tau_{d4}$ and $\tau_{d1} = \tau_{d3}$. This will eliminate G_2 from equation 6.12 plus it will reduce equations 6.16 and 6.17 to equation 6.18.

$$G_{34} = \frac{e^{-\tau_{d3}s}}{s}(s + (\lambda + q_1))(A_{i-1}(s) - (A_{i-2}(s))) \quad (6.18)$$

where

$$(A_{i-1}(s) - (A_{i-2}(s))) = \ddot{E}_{i-1} = s^2 E_{i-1} \quad (6.19)$$

resulting in

$$\frac{E_i(s)}{E_{i-1}(s)} = \frac{\frac{1}{1+q_3}(e^{-\tau_{d1}s}s^2 + e^{-\tau_{d1}s}(\lambda + q_1)s + \lambda q_1)}{\tau s^3 + s^2 + \frac{\lambda(1+q_3)+q_1+q_4}{1+q_3}s + \frac{\lambda(q_1+q_4)}{1+q_3}} \quad (6.20)$$

If $\tau_{d1} = \tau_{d3} = 0$ then equation 6.20 will simplify to equation 3.22 which was established in chapter 3.

The general solution can be manipulated to give us a transfer function in terms of E_i and E_{i-1} . Unfortunately, the solution is not in a tight form as desired. Therefore, this result is broken into two distinct cases; lead vehicle information delay and preceding vehicle information delay. The analysis for these two cases are discussed in the next two sections.

6.2.2 Lead Vehicle Information Delay

In this section, a variation of the general solution developed in the previous section will be analyzed. In this scenario, the control law will be triggered when the preceding vehicle information is received. It will assume that the transmission delay of the preceding vehicle information is negligible making τ_{d1} and τ_{d3} equal zero.

By substituting zero for τ_{d1} and τ_{d3} in equation 6.6, the equation will reduce to

$$\tau \frac{d^3 \varepsilon_i}{dt} + \ddot{\varepsilon}_i + \left(\lambda + \frac{q_1 + q_4}{1 + q_3}\right) \dot{\varepsilon}_i + \left(\frac{\lambda(q_1 + q_4)}{1 + q_3}\right) \varepsilon_i = \frac{1}{1 + q_3} [\ddot{\varepsilon}_{i-1} + (\lambda + q_1) \dot{\varepsilon}_{i-1} + \lambda q_1 \varepsilon_{i-1} + \zeta_1] \quad (6.21)$$

where $\zeta_1 = \beta$ shown in equation 6.9.

After taking the Laplace transform, the following transfer function is generated:

$$E_i = H_2 E_{i-1} + H_{3i} A_o \quad (6.22)$$

where

$$H_2 = \frac{\frac{1}{1+q_3}(s^2 + (\lambda + q_1)s + \lambda q_1)}{\tau s^3 + s^2 + \frac{(\lambda+q_1+q_4+\lambda q_3)}{1+q_3}s + \frac{\lambda(q_1+q_4)}{1+q_2}} \quad (6.23)$$

and

$$H_{3i} = \frac{(e^{-\tau d_2 s} - e^{-\tau d_4 s})(q_3 s^2 + (q_4 + \lambda q_3)s + \lambda q_4)}{\tau s^5 + s^4 + \frac{(\lambda(1+q_3)+q_1+q_4)}{1+q_3}s^3 + \frac{\lambda(q_1+q_4)}{1+q_3}s^2} \quad (6.24)$$

Since no delays will ever be encountered between the phantom vehicle and the lead vehicle, the transfer function in equation 6.25 will always hold.

$$H_1 = \frac{E_1}{A_o} \quad (6.25)$$

where H_1 is derived in chapter 3 equation 3.21

By solving for A_o and substituting it into equation 6.22, the transfer function for vehicle 2 can be established (equation 6.27).

$$A_o = \frac{E_1}{H_1} \quad (6.26)$$

therefore

$$\frac{E_2}{E_1} = H_2 + \frac{H_{32}}{H_1} \quad (6.27)$$

A similar approach can be taken to solve for the next vehicle's transfer function. In this case, $i = 3$ in equation 6.22 resulting in

$$E_3 = H_2 E_2 + H_{33} A_o \quad (6.28)$$

Substituting equation 6.26 into equation 6.28, results in

$$E_3 = H_2 E_2 + \frac{H_{33}}{H_1} E_1 \quad (6.29)$$

and by solving for E_1 in equation 6.27 and substituting it into the above equation 6.29, results in the following transfer function for vehicle 3.

$$\frac{E_3}{E_2} = H_2 + \frac{H_{33}}{H_1} \left(\frac{1}{H_2 + \frac{H_{32}}{H_1}} \right) \quad (6.30)$$

After rearranging the terms in equation 6.30, the transfer function can be simplified (equation 6.31).

$$\frac{E_3}{E_2} = H_2 + \frac{H_{33}}{H_1 H_2 + H_{32}} \quad (6.31)$$

By recursively using the transfer functions from the vehicles ahead, a general solution can be derived. This solution is shown below in equation 6.32. To simplify the equation, H_1 is set to H_{31} .

$$\frac{E_i}{E_{i-1}} = H_2 + \frac{H_{3i}}{H_{3(i-1)} + H_{3(i-2)}H_2 + H_{3(i-3)}H_2^2 + \dots + H_{31}H_2^{i-2}} \quad (6.32)$$

H_{3i} is dependent on τ_{d2} and τ_{d4} except for H_{31} .

The bode plots³ were generated for vehicle 2, 3, 4 and 5 (figure 6.3) for a scenario which consisted of a five vehicle platoon with token ring time of 20 milliseconds. This allows each vehicle to have the token for 4 milliseconds. Since the control law is triggered once the preceding vehicle information is received, vehicle 2 will see no delay in lead vehicle information, whereas vehicle 3, vehicle 4 and vehicle 5 will see a 4, 8, 12 milliseconds delay in the lead vehicle information respectively.

Vehicle 2 does not encounter any delays in this scenario, therefore it is string stable. The rest of the vehicles in this scenario are string unstable. This is true for any lead information delay seen by the vehicles including infinitesimal delays. The crossover frequency of the spacing errors seen by vehicle 3 using the gains that were used by Swaroop ($\lambda = 1.0, q1 = 0.8, q3 = 0.5, q4 = 0.4$) is approximately .125 Hz and using Rizzuti's gains ($\lambda = 0.5, q1 = 0.72, q3 = 0.43, q4 = 0.25$) is approximately .100 Hz.

³The bode plots were generated using Matlab, and the pure time delay was approximated using pade's approximation.

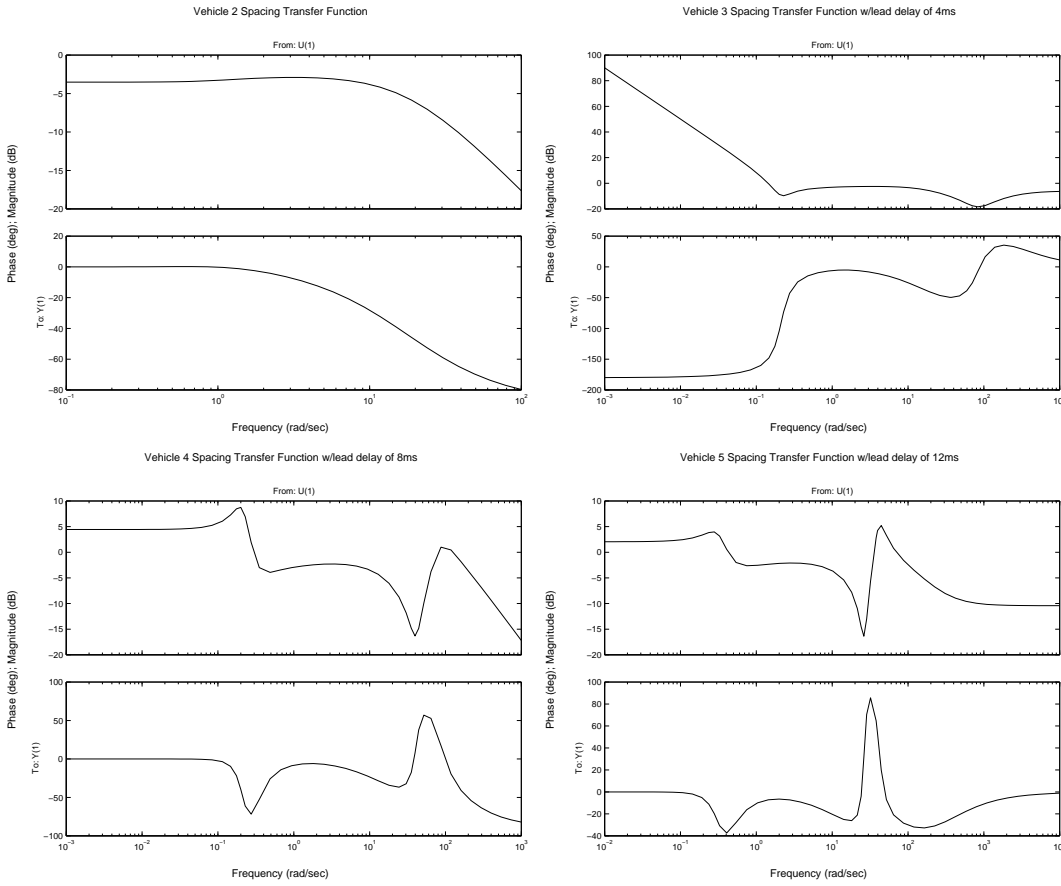


Figure 6.3: Lead Vehicle Information Delay Bode Plots

6.2.3 Preceding Vehicle Information Delay

The scenario studied in this section is similar to the previous section. In this scenario, the control loop will be triggered when the lead vehicle information is sent whereas in the previous section the control loop was triggered when the preceding vehicle information was received. The general solution derived in section 5.2.1. will be reduced to reflect no delays in the lead vehicle information. Since τ_{d2} and τ_{d4} will equal zero, this will eliminate β in equation 6.8. In this case, equation 6.12 will become

$$H_{11}E_i(s) = \frac{1}{1 + q_3}[G_1E_{i-1}(s) + G_3A_{i-1}(s) - G_4A_{i-2}(s)] \quad (6.33)$$

where H_{11} , G_1 , G_3 and G_4 are shown in equations 6.13, 6.14, 6.16 and 6.17 respectively.

It can be seen that the technique used in the previous section where the transfer function can be easily manipulated by solving for A_o does not work in this case. The equation 6.33 must be further manipulated.

The spacing error definition will be used to eliminate unknown quantities

$$\varepsilon = x_i - x_{i-1} \quad (6.34)$$

For the following analysis, it is easier to manipulate the equations in the time domain. By starting with equation 6.6 and setting τ_{d2} and τ_{d4} to zero, we will get

$$\tau \frac{d^3\varepsilon_i}{dt} + \ddot{\varepsilon}_i + \left(\lambda + \frac{q_1 + q_4}{1 + q_3}\right)\dot{\varepsilon}_i + \left(\frac{\lambda(q_1 + q_4)}{1 + q_3}\right)\varepsilon_i = \frac{1}{1 + q_3}[\lambda q_1\varepsilon_{i-1} + \gamma + \theta] \quad (6.35)$$

and by replacing $a_{i-1}(\tau + \tau_{d1})$ by

$$a_{i-2}(\tau + \tau_{d1}) + \ddot{\varepsilon}_{i-1}(\tau + \tau_{d1}) \quad (6.36)$$

and $v_{i-1}(\tau + \tau_{d1})$ by

$$v_{i-2}(\tau + \tau_{d1}) + \dot{\varepsilon}_{i-1}(\tau + \tau_{d1}) \quad (6.37)$$

in equation 6.38, γ will become

$$\gamma = a_{i-2}(\tau + \tau_{d1}) + \ddot{\varepsilon}_{i-1}(\tau + \tau_{d1}) + (\lambda + q_1)(v_{i-2}(\tau + \tau_{d1}) + \dot{\varepsilon}_{i-1}(\tau + \tau_{d1})) \quad (6.38)$$

and θ will remain the same (equation 6.11).

After rearranging the terms, it can be seen that

$$\tau \frac{d^3 \varepsilon_i}{dt} + \ddot{\varepsilon}_i + \left(\lambda + \frac{q_1 + q_4}{1 + q_3}\right) \dot{\varepsilon}_i + \left(\frac{\lambda(q_1 + q_4)}{1 + q_3}\right) \varepsilon_i = \frac{1}{1 + q_3} [\lambda q_1 \varepsilon_{i-1} + \gamma + \theta] \quad (6.39)$$

$$\tau \frac{d^3 \varepsilon_i}{dt} + \ddot{\varepsilon}_i + \left(\lambda + \frac{q_1 + q_4}{1 + q_3}\right) \dot{\varepsilon}_i + \left(\frac{\lambda(q_1 + q_4)}{1 + q_3}\right) \varepsilon_i = \frac{1}{1 + q_3} [\eta + \theta_2] \quad (6.40)$$

where

$$\eta = \lambda q_1 \varepsilon_{i-1} + \ddot{\varepsilon}_{i-1}(\tau + \tau_{d1}) + (\lambda + q_1)(\dot{\varepsilon}_{i-1}(\tau + \tau_{d1})) \quad (6.41)$$

$$\theta_2 = \theta + a_{i-2}(\tau + \tau_{d1}) + (\lambda + q_1)(v_{i-2}(\tau + \tau_{d1})) \quad (6.42)$$

where θ is shown in equation 6.11. After taking the Laplace transform, equation 6.40 will be a function of E_i , E_{i-1} and A_{i-2} . For vehicle 2, A_{i-2} will equal A_o and the same approach used in the previous section can be used to derive it's spacing error transfer function. For vehicles following vehicle 2 (i.e. for $i > 2$), A_{i-2} will not equal A_o therefore θ_2 must be manipulated further.

By using

$$a_{i-2} = a_{i-3} + \ddot{\varepsilon}_{i-2} \quad (6.43)$$

and substituting it into equation 6.42 and taking the Laplace transform, equation 6.40 will be a function of E_i , E_{i-1} , E_{i-2} and A_{i-3} . For vehicle 3, A_{i-3} will equal A_o and E_{i-2} will equal E_1 . By using A_o from equation 6.26, and by solving for E_1 from vehicle 1's spacing error transfer function, a transfer function of vehicle 3 can be derived.

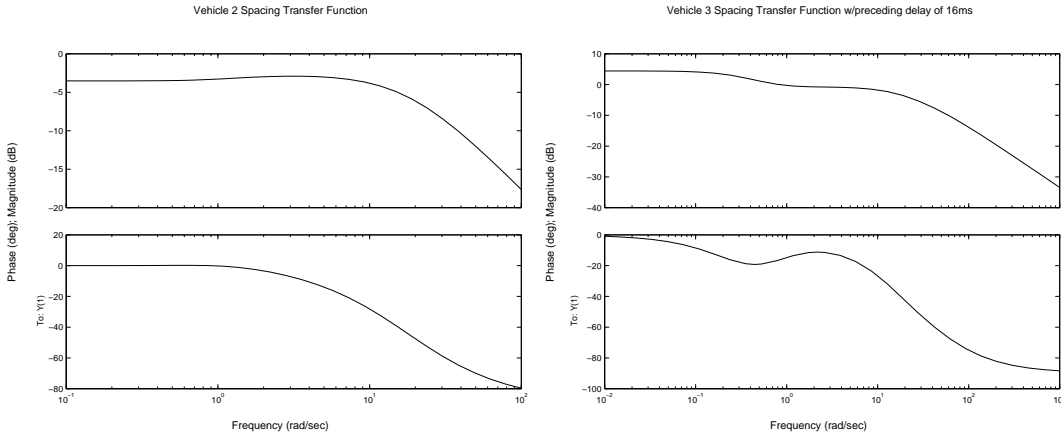


Figure 6.4: Preceding Vehicle Information Delay Bode Plots

To clarify the technique to determine the spacing transfer function for vehicle 3, additional steps that are required are shown below. This process has been described above but has been repeated for clarification.

Similar to the previous section, vehicle 2 will not see any delays therefore this vehicle is string stable whereas vehicle 3 is string unstable as shown on the bode plots in figure 6.4.

$$E_i = H_2 E_{i-1} + H_3 A_{i-2} \quad (6.44)$$

note for $i=2$ this becomes

$$E_2 = H_2 E_1 + H_3 A_0 \quad (6.45)$$

using equation 6.26, we get

$$\frac{E_2}{E_1} = H_2 + \frac{H_3}{H_1} \quad (6.46)$$

For $i>2$, this simple substitution is not possible. In equation 6.48, the A_{i-2} will not be A_0 , therefore other manipulations to the equation must be made. The inverse Laplace transformation of $H_3 A_{i-2}$ looks like the following

$$L^{-1}(H_3A_{i-2}) = h_3a_{i-2} = h_3(a_{i-3} + \ddot{\epsilon}_{i_2}) \quad (6.47)$$

This will give us

$$E_i = H_2E_{i-1} + H_3A_{i-3} + s^2H_3E_{i-2} \quad (6.48)$$

For vehicle 3, $A_{i-3} = A_0$ and $E_{i-2} = E_1$ which can be manipulated by substituting A_0 in terms of E_1 and then again by substituting E_1 in terms of E_2 from the previous transfer functions for vehicles 1 and 2. This will result in a transfer function which can be analyzed. The next set of equations establish this relationship.

$$E_3 = H_2E_2 + H_3A_0 + s^2H_3E_1 \quad (6.49)$$

after substituting in A_0 in terms of E_1 , we get

$$E_3 = H_2E_2 + \left(\frac{H_3}{H_1} + s^2H_3\right)E_1 \quad (6.50)$$

after substituting in E_1 in terms of E_2 from equation 6.46, we get

$$\frac{E_3}{E_2} = H_2 + \frac{\frac{H_3}{H_1} + s^2H_3}{H_2 + \frac{H_3}{H_1}} \quad (6.51)$$

arranging the variables, equation 6.51 can be simplified to

$$\frac{E_3}{E_2} = H_2 + \frac{H_3 + s^2H_1H_3}{H_1H_2 + H_3} \quad (6.52)$$

The transfer functions for the additional preceding vehicles can be determined using this process. As one can see, the transfer functions will become quite messy.

The bode plots of the transfer function for vehicles 2 and 3 are shown in figure 6.4. Similar to the previous section, there is no communication delays seen by either vehicle 1 or 2 therefore it is a string stable system. As for the spacing transfer function relating vehicle

3 to vehicle 2, this system is string unstable. The crossover frequency using the Swaroop's gains is approximately 1.25 Hz and using Rizzuti's gains is approximately 2 Hz.

Chapter 7

Simulation Results

Each scenario discussed in the previous section was simulated initially via Matlab and then using Comb_Sim. Comb_Sim is a simulation package developed in the Vehicle Dynamics Laboratory. It simulates a realistic vehicle model with both longitudinal and lateral controllers. The Matlab program simulates only a longitudinal controller with the plant dynamics modeled as a double integrator.

The communication time delays were modeled as pure time delays in Simulink (Matlab). This was easily implemented by using a time delay block which was triggered after a specified amount of time allowing the platoon to initialize properly. The simulink block diagram used is included in the appendix.

As for Comb_Sim, the main control loop had to be modified to simulate communication delays. Before the modifications were made, the controllers were executed for each vehicle during every control cycle assuming a perfect radio communication system. The program was modified so that the vehicles in the platoon were decoupled allowing the controllers for each vehicle to run at different cycles. Each vehicle still has the same control cycle time, but each vehicle executes its controllers independent of each other.

The control loop is executed every 20 milliseconds. The vehicle dynamics are updated every time step which is set to 1 millisecond. The sequence of events in the control loop is illustrated in figure 7.1. Each vehicle executes all the routines in the control loop within one time step. The time step is incremented at the end of the while loop.

Once the program is initialized, a program executes a while loop until the user defined

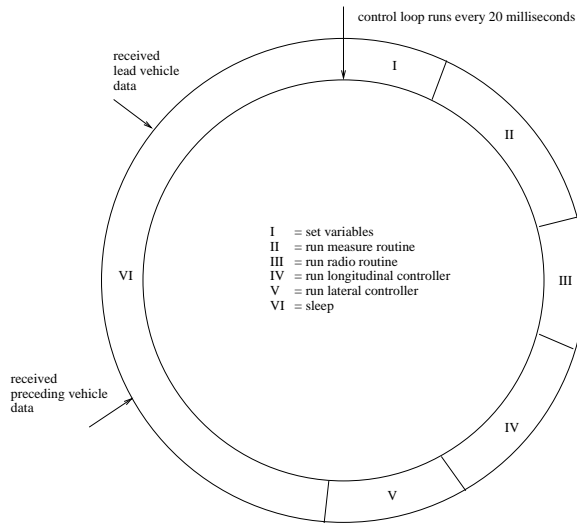


Figure 7.1: Comb_Sim Control Loop

simulation time has expired. This simulation time can be changed easily in the simulation_param.cfg configuration file. The three different loops that are executed within the main loop consist of the plant loop, the control loop and the data recording loop. The loop is executed every cycle whereas the update frequency of the other two loops are hard coded in the initialization stage. The timing of the whole program is controlled by the time step counter.

To run the scenario with no delays, the control loop for each vehicle was executed within one time step. This assures that all vehicles receive the necessary information from the lead and preceding vehicle without any delays.

The delay scenarios were based on the TDMA scheme as described in previous chapters. The delays are based on a 20 millisecond control period. The simulation created a platoon size of five vehicles. Since vehicle 2 receives information directly from the lead vehicle which is also the preceding vehicle, it is assumed to not observe any delays in this information. If the receipt of the preceding vehicle information triggers the control algorithm, then a delay of 4, 8, 12 milliseconds in the lead vehicle information will be seen by vehicles 3, 4, 5 respectively. This is true for the two cases where the control loop is triggered when data is received via wireless communication.

A summary of the delays that might be encountered are as follows. If the receipt of the lead vehicle information triggers the control algorithm, then a delay of 16, 12, 8 milliseconds in

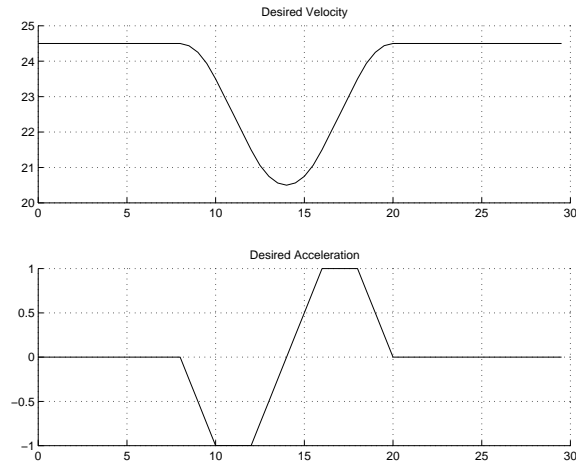


Figure 7.2: Desired Trajectory

the preceding vehicle information will be seen by vehicles 3, 4, 5 respectively. The Comb_Sim simulation program was modified to determine these numbers based on the control period and size of the platoon.

Figure 7.2 shows the desired trajectory used for the simulations. The first set of plots 7.3 shows the spacing errors observed with a perfect communication system. The gains are varied in this scenario. Three sets of gains are used; gains used by Swaroop, gains used in the San Diego demo and the gains specified by Rizzuti. Rizzuti determined that an additional criteria on the controller gains was necessary since the system is not continuous but discrete. The last set of gains satisfy this criteria.

The next set of plots show the effects of lead vehicle delay. Figure 7.4 shows the spacing errors observed using the gains established by Swaroop. It can be seen that the system is not string stable. Although when the same program was executed using Rizzuti's gains, it can be seen the system is string stable (figure 7.5). When the desired trajectory was modified to the trajectory shown in figure 7.6, the spacing errors between vehicle 4 and vehicle 5 grew making the system unstable. Plots of the spacing errors were generated using all three set of gains for the modified desired trajectory. All three cases show that the system is string unstable (figure 7.7).

The next set of plots show the effects of preceding vehicle delay. The Comb_Sim program is not capable of simulating preceding vehicle delay therefore Simulink was used instead. Figure 7.8 shows the spacing errors observed using the gains established by Swaroop with

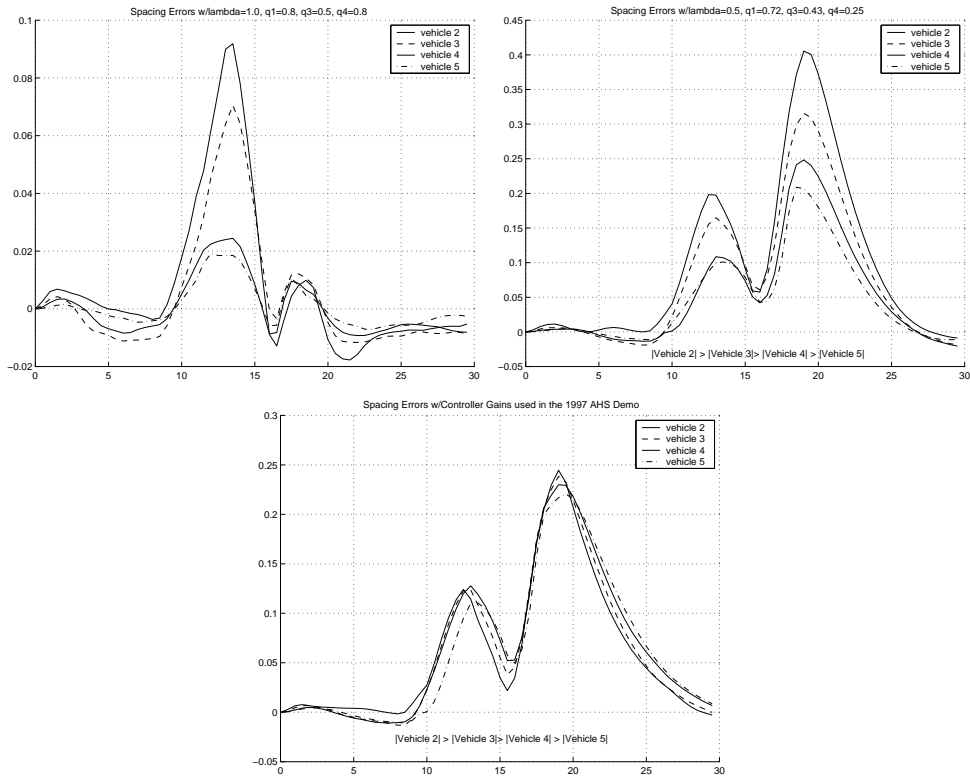


Figure 7.3: Spacing Errors Using a Perfect Communication System

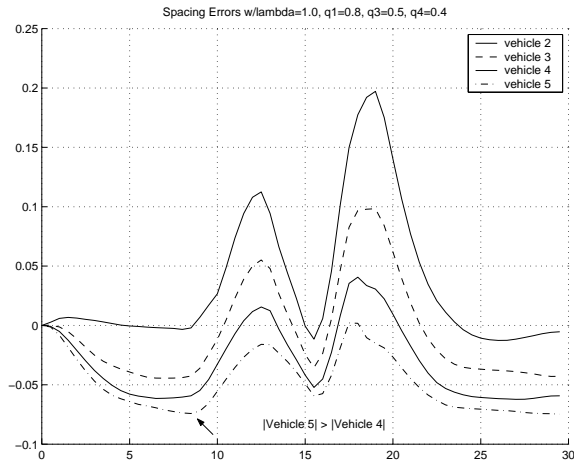


Figure 7.4: Preceding Vehicle Triggers Control Loop with Swaroop's Gains

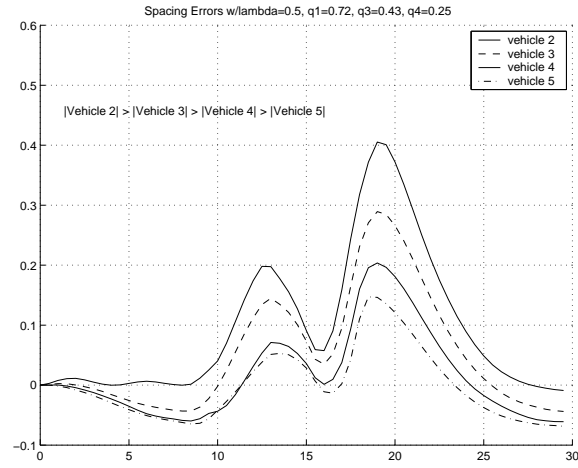


Figure 7.5: Preceding Vehicle Triggers Control Loop with Rizzuti's Gains

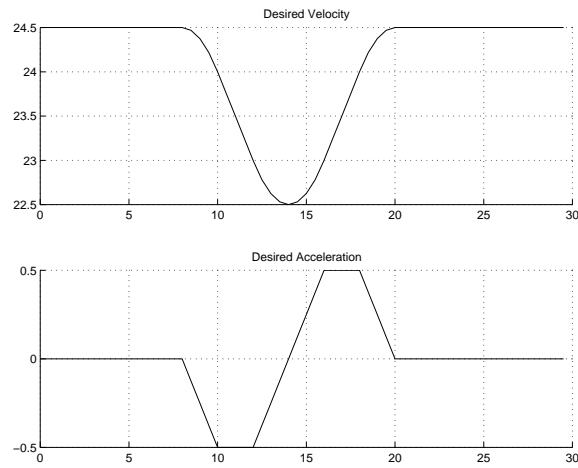


Figure 7.6: Modified Trajectory

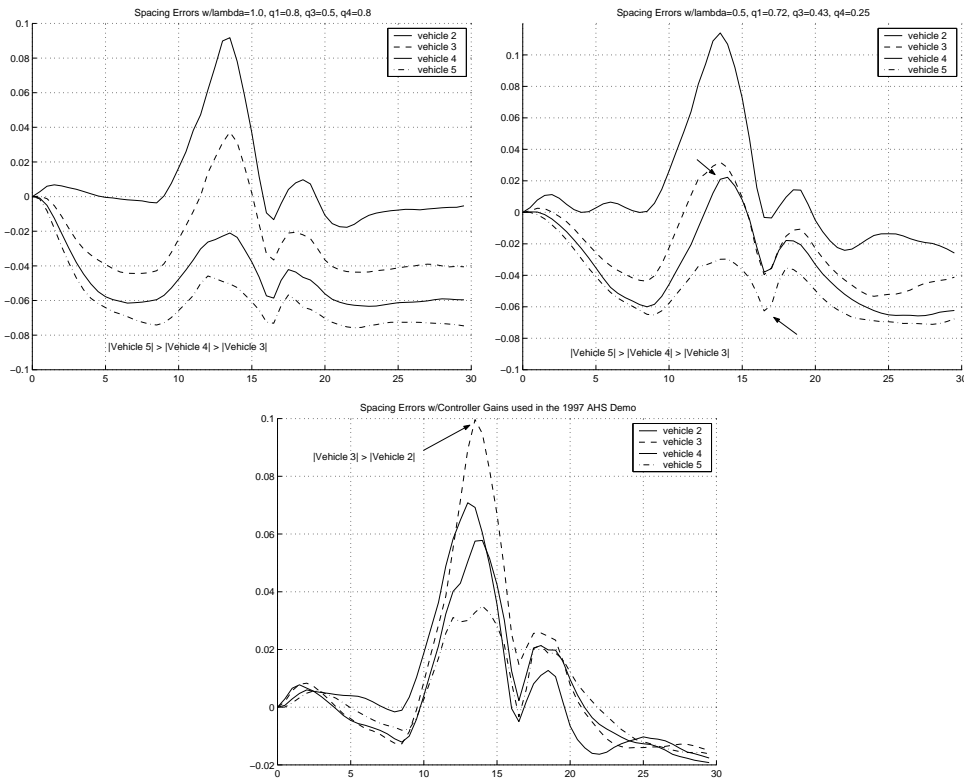


Figure 7.7: Preceding Vehicle Triggers Control Loop w/ Modified Trajectory

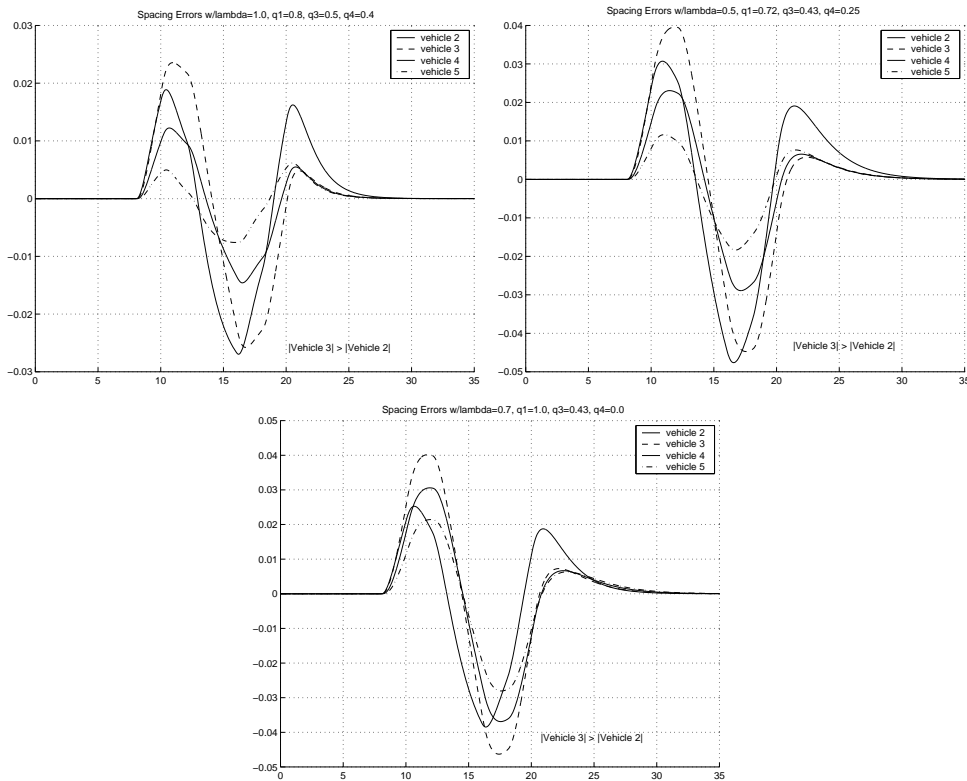


Figure 7.8: Lead Vehicle Triggers Control Loop

the lead vehicle triggers the control loop. It can be seen that the system is not string stable. Similar results were obtained using both Rizzuti's gains and the gains used in the 1997 AHS demonstration. The original trajectory shown in figure 7.2 was used to obtain these results.

It was concluded that any delay in the communicated information required by the existing control algorithms will not be string stable over all conditions. There will always be delays in communication systems.

Chapter 8

Nomenclature and Assumptions for Protocol Design

8.1 Nomenclature for Distributed Systems

For the systems described in this thesis, we will assume a collection of processes that form a loosely coupled message-passing system without any shared memory between processes, and the absence of a global clock. A *distributed program* consists of N processes $\{P_1, \dots, P_N\}$, which communicate solely through asynchronous messages. In a single *run* of a distributed program, each process P_i generates an *execution trace* consisting of a finite sequence of local states and events in process P_i . The values of all variables in the process P_i are associated with the *state* of that process. If we assume that there is no non-determinism within the process itself, this means that the event that occurs next in process P_i is completely determined from the state and any messages received. A run, r , can be mathematically represented by a vector of execution traces where $r[i]$ represents the execution trace of process P_i .

Let us now describe a few relations that describe the relationship between states in a run. The “*locally precedes*” relation (\prec_{im}) is defined as follows. The state s in trace $r[i]$ locally precedes state t in trace $r[i]$ (denoted by $s \prec_{im} t$) if and only if s immediately precedes the t in the execution trace of process P_i .

The “*remotely precedes*” relation (\rightsquigarrow) is defined as follows. Event e in trace $r[i]$ remotely precedes event f in trace $r[j]$ (denoted by $e \rightsquigarrow f$) if and only if e is the send event of a message and f is the receive event of the same message. State s in trace $r[i]$ remotely precedes state t in trace $r[j]$ (denoted by $s \rightsquigarrow t$) if and only if a message is sent by process P_i after state s , which is received by process P_j , resulting in state t .

The “causally precedes” relation (\rightarrow) is defined as the transitive closure of the union of \prec_{im} and \rightsquigarrow . In other words, $s \rightarrow t$ if and only if either:

1. $(s \prec_{im} t)$ OR $(s \rightsquigarrow t)$, or
2. there exists a state u such that $(s \rightarrow u)$ AND $(u \rightarrow t)$.

Two states, s and t , are defined to be “concurrent” (denoted by $s \parallel t$) if s does not causally precede t and t does not causally precede s .

8.2 Assumptions for the Communication System

We will assume a wireless communication system that is capable of sending and receiving on a single channel. This assumption is made because all the radio systems to date in the test vehicles have been single channel, half duplex models. The routing table, which allows a vehicle/node to exchange messages within its LAN holds the following information:

```
Platoon ID:    0ABC123
Platoon Node:  1
Platoon Size:  5
```

The platoon ID is a unique identifier for a platoon/LAN. A suggestion is to use the license plate number of the lead node of the platoon to ensure the uniqueness of this number. The platoon node provides a unique identity for the vehicle/node within the platoon/LAN as well as indicating the position within the platoon. A “1” indicates the lead node in a platoon, a “0” indicates a free agent, and any other positive number indicate followers in a platoon. The platoon size gives a vehicle/node knowledge of what other valid addresses exist within its platoon/LAN. In order to properly address a message, the address of both the sender and the destination should be supplied. An example of this is as follows:

```
Source.PlatoonID:    0ABC123
Source.PlatoonNode:  0
Source.PlatoonSize:  1
Destination.PlatoonID:  9ZYX876
Destination.PlatoonNode: 1
Destination.PlatoonSize: 5
```

This example shows a free agent sending a message to the leader of a five car platoon. For communication systems with broadcast capability, a special code may be defined for the platoon node identifier that allows the destination platoon node to be “all nodes with the given destination platoon ID”. In field tests, this is indicated by a destination platoon node value of -1.

Chapter 9

Platoon Split Protocol

9.1 Initiating a Platoon Split

In this model of the automated highway system, only vehicles which are free agents are allowed to make lane changes and entry/exit maneuvers. Any given vehicle would go about becoming an free agent through two stages: first by becoming the lead vehicle or last vehicle in the platoon; then by splitting off from the end of a platoon and becoming an free agent. Thus, a vehicle would initiate a platoon split for the purpose of becoming an free agent.

There should also be an algorithm for determining when a vehicle should begin requesting a split maneuver. The process of becoming an free agent takes a finite amount of time, and currently can take up to two separate split maneuvers to accomplish. Also, there are presumably other vehicles in the platoon that also wish to become free agents and are making requests of their own. Exactly what this algorithm will be should become clearer as the protocol for the split maneuver becomes more defined, and we can put some time bounds on in.

9.2 Servicing a Platoon Split Request

In a platoon of more than two vehicles, there can be more than one split request generated at once. Since the maximum number of vehicles (which can also be thought of as nodes in a distributed system) allowed in a platoon is fairly small (<50 vehicles), a centralized arbitration scheme is indicated. A natural choice for this arbiter is the lead vehicle in a platoon, for the very simple reason that every platoon must, by definition, possess a leader. The lead vehicle in a platoon already serves as something of an information hub for control data, so many

communication systems designed for use in a platoon are already set up for this type of information flow.

A simplistic, but inefficient arbitration algorithm is to always service the request generated by the front-most vehicle. If node numbers within the platoon are assigned in ascending order from the lead vehicle, this means that the request generated from the lowest numbered node receives priority. If we state that split maneuvers receive priority over join maneuvers, and given that a platoon has a finite number of vehicles, this means that all split requests will eventually be serviced. However, this means that the higher numbered nodes in the platoon will always wait longer before their requests are serviced, and that a platoon can only service one request at a time. We can bound the time that it takes for any given request to be serviced under this algorithm by taking the worst case scenario of every single vehicle generating a split request. In a platoon of N vehicles, assuming that a split maneuver required M seconds to complete, this would yield an upper bound of $M \cdot N$ seconds before the request is serviced (meaning that the requested split maneuver is complete). Granted, this occurs only for the last vehicle in the platoon, and only if all vehicles generate a split request at the same time, but it is still not as efficient as it could be.

A more efficient arbitration algorithm can be achieved by realizing that the two platoons that result from a split request now have two platoon leaders and can potentially service two split requests. In order to take advantage of this, the arbitration algorithm should choose to service the request such that the resulting platoons have a similar number of remaining split requests to service. Let us assume that the lead node (i.e. vehicle) in a platoon keeps track of split requests through the use of a Boolean array `s_request []`, which has one element for every node in the platoon. The i -th element in the array is `TRUE` when the i -th node has generated a request for a split maneuver, otherwise it is `FALSE`. The idea is to service the request that would divide the single platoon into two platoons with a relatively equal number of outstanding split requests.

```
sum = 0.0;
count = 0;
for i=1; i<=PLATOON_SIZE; i++
{
    if( s_request[i] == TRUE )
    {
        sum = sum + i;
        count = count + 1;
    }
}
mean = sum / count;
temp = quotient(mean);          /*Take the integer part of 'mean'*/
if( remainder(mean) > 0.0 )    /*If 'mean' has a nonzero remainder,*/
    temp = temp + 1;           /*take the next highest integer.*/
while( s_request[temp] == FALSE )
    temp = temp + 1;
service(temp);                 /*Service the node 'temp'*/
```

This algorithm takes the mean of the node numbers that generated split requests and services the request from the lowest numbered node that is greater or equal to this mean. To clarify, the node who's request is serviced becomes a lead node as a result of the split. An upper bound on the time it would take to service any given request in a platoon of N vehicles can be obtained by

taking the worst case scenario of every node generating a request at the same time. Assuming that each split maneuver takes M seconds to complete, then the upper bound is given by:

$$M \cdot \text{int_lub}(\log_2 N) \text{ (seconds)}$$

where ‘int_lub(.)’ is a function that yields the smallest integer that is greater than or equal (i.e. an integer least upper bound) to its argument. This is much smaller than the upper bound given by the simple arbitration algorithm presented previously and the difference between the two algorithms increases as the size of the platoon increases, since $\log_2 N$ grows much more slowly than N .

More efficient algorithms may be obtained with additional information, such as which split requests are generated by vehicles which actually wish to become free agents and which are generated because the platoon size has become too large. However, the arbitration algorithm presented here uses the minimum available information (i.e. the split requests themselves), and is fairly simple.

9.3 Protocol Design

Before we attempt to prove any properties associated with these finite state machines, a bit of nomenclature must first be established. Each finite state machine represents a process from a distributed systems standpoint. Each process will be referred to by its designation (1A, 1B, 1C, or 1D) as in the figures FSM 1A through FSM 1D. A state in a given process will be referred to by its number as labeled in the previous figures, which is the number in the right hand corner of the box enclosing the state. These state machines are drawn as Moore machines, where actions taken by the state machine itself are a property of a state, while the reaction to external events are a property of the transitions. If no label is applied to a transition, that indicates that the transition is always taken. This “always taken” type of transition is used when it is convenient to have the state machine perform two tasks sequentially. To make the proofs easier to understand, the notation $xx.yy$, or equivalently xx_{yy} will be used to refer to state ‘xx’ in process ‘yy’.

Note that there may be more than one vehicle executing process 1B and likewise more than one vehicle executing process 1C. We can define a system state vector as the representation of the states of the FSM’s in each vehicle (i.e. node) at a given time, with the i -th entry in the vector representing the state of the FSM in the i -th vehicle in the platoon. For example, the system state vector of a 10-car platoon before a split maneuver is requested, and with the sixth vehicle in the platoon wishing to request a split maneuver would be written as follows:

$$V = [1_{1A} \quad 0_{1B} \quad 0_{1B} \quad 0_{1B} \quad 0_{1B} \quad 0_{1C} \quad 0_{1D} \quad 0_{1D} \quad 0_{1D} \quad 0_{1D}]$$

There are actually two ways a split maneuver can be initiated. The first method is to have a follower vehicle (denoted in the finite state machines as LC’ because it will become a lead vehicle once the maneuver is complete) request the maneuver by sending a SPLIT_REQ message

to the leader node in the platoon. In this case, depending on the actual position of the follower now designated as LC', the processes 1B and 1D may or may not exist. However the desired properties of the distributed system will remain true in spite of this. The second method of initiating a split maneuver is to have the lead vehicle itself request the maneuver. In this case, no external request message is needed and process 1C would simply start from the O_{1C} state from the top.

9.3.1 Lossy Channel Model

Once a split maneuver has been requested, we must now conceive of a procedure by which we service this request. Let us assume a lossy FIFO channel between nodes. In this channel model, messages sent to a destination may be lost. Let us also assume that the channel handles message resends and acknowledgments transparently. That is, when a message is successfully sent by process i to process j , and the acknowledgement to that message was successfully sent by process j and received by process i , the channel will indicate to process i that the message was successfully sent to process j . Thus a successful send/broadcast of a message implies that the message was sent and received, and that the necessary acknowledgements were also sent and received. If a message from process i to process j is lost, or if one or more of the acknowledgements to that message is lost, then the channel will try to resend the message some preset number of times before indicating to process i that the message has "timed-out" if none of the resends were successful. If the channel returns a success, this indicates that the message and its associated acknowledgements were sent and received successfully. However, if a message times-out, this could either indicate that the message was never reached its destination or that the message was received but the acknowledgement was lost.

Since that the possibility of lost messages exists, we must start to think about ways to recover from these lost messages. In some cases, a lost message only means that the run of the distributed system makes no forward progress. In other cases, especially if some vehicles in a platoon have undergone reconfiguration while others have not, it becomes necessary to recover a previous state of the platoon. For this purpose, it is proposed that each lead vehicle/node in a platoon keep track of the number of split maneuvers that have been performed under its current platoon ID. In essence, creating an identifying number for the maneuver itself. Also, each vehicle would keep a short queue of the most recent maneuvers that it has undergone, and the platoon ID, platoon position, and channel that was associated with it before and after each maneuver. That way, the last state of the platoon can always be recovered.

A procedure for recovering from lost messages must now be thought out. For now, let us leave the detail of such a recovery mechanism vague, and instead set the rules by which it will operate (to an external observer), and the information it will require upon entering the procedure and what information it will supply at the exit points of the procedure. Since such a recovery procedure will handle mostly abort mechanisms, it will be referred to as an "abort handler" for the remainder of this document.

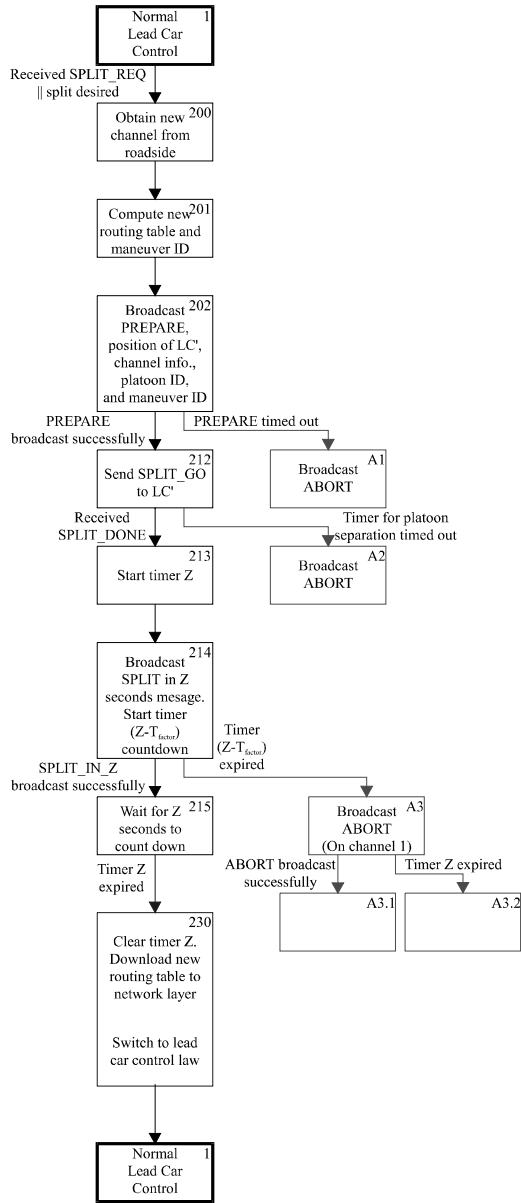
Currently, each vehicle stores its addressing information in a routing table. This routing table identifies the platoon in which the vehicle is currently operating, the size of that platoon, and the position it occupies within that platoon. In network terms, the routing table stores the ID of the local area network (LAN), the number of nodes in that LAN, and the node ID within the LAN. In order to recover a previous state from an aborted maneuver, a set of “old” parameters from the routing table should be stored. Note that in the state machines from the simple channel model, the “new” routing tables that would take effect are computed early in the split protocol. In order to make recovery possible, a three entry routing table is proposed: one entry for the “current” routing table; one entry for the “old” routing table; and one entry for the “new” routing table. The “current” entry describes the information used by the local node (i.e. vehicle) to identify itself at the moment. The “old” entry describes the information last used. The “new” entry will only contain valid information when a maneuver is in progress, and before the reconfiguration state of that maneuver is reached. Another set of information must be created in each node to identify the current maneuver, and whether the reconfiguration state is reached. A single entry will suffice for this information, since the safety condition that is required of all maneuvers guarantees that no maneuver can begin without the previous one being completed. This single entry will need to contain a unique identifier for the maneuver in progress (a suggested method is to combine the platoon ID and the number of maneuvers that this platoon has undergone to create a unique identifier), as well as a progress flag that can either be set as `in_progress` or `reconfigured`. Once the reconfiguration state is reached in a maneuver, the “current” routing table is copied into the “old” routing table, the “new” routing table is copied into the “current” current routing table, and the maneuver progress flag is changed from `in_progress` to `reconfigured`.

Using the lossy channel model, and the abovementioned information, the task of the abort handler once a message time-out has occurred is to:

1. Determine the state of the platoon (the details of this will vary depending on what type of node, follower or leader, it is).
2. Recover from the time-out if possible.
3. Abort the maneuver if recovery is not possible.

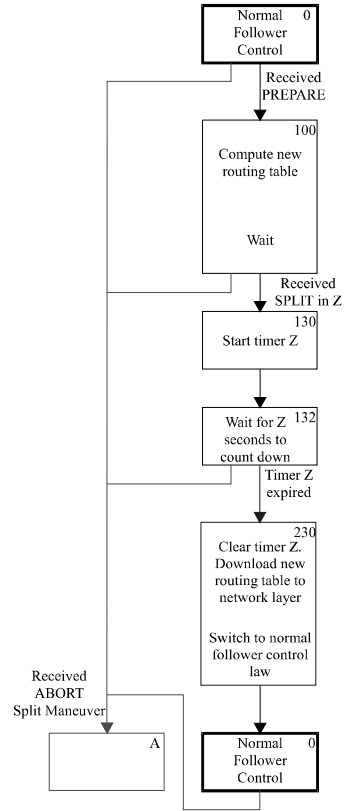
Having defined our assumptions for the lossy FIFO channel model, and what an abort handler is expected to do, the FSM's for the four process types under this channel model are as follows:

Leader



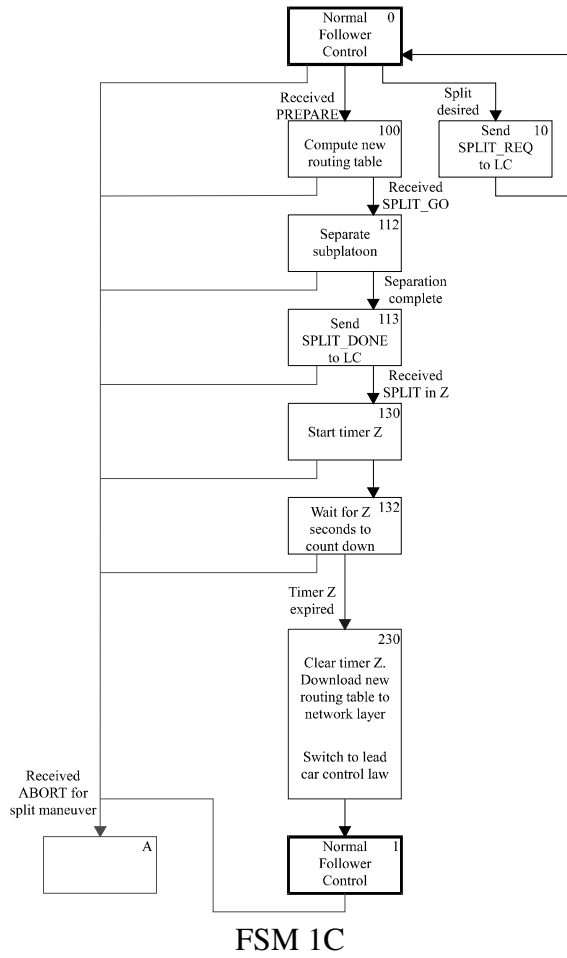
FSM 1A

Followers with node numbers less than LC'

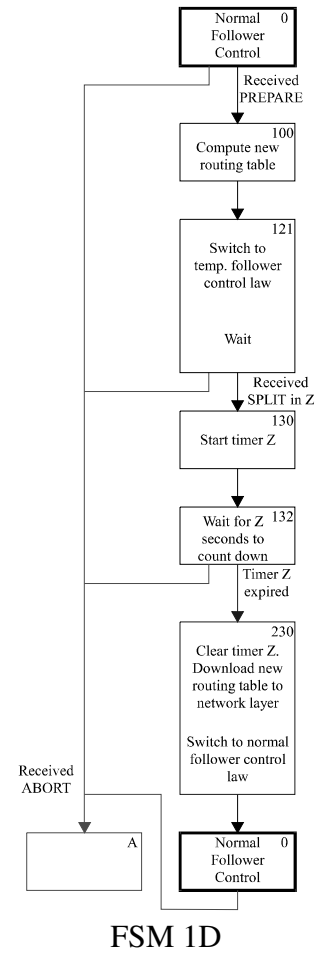


FSM 1B

LC' (The follower that becomes a lead car once the split is complete)



Followers with node numbers greater than LC'



The desired properties of the distributed system are as follows:

1. Another split maneuver cannot be started before the previous one is successfully complete (safety condition). By "successfully complete", it is meant that all nodes alter their routing tables correctly and reenter normal leader/follower modes.
2. Reconfiguration states occur concurrently in all vehicles within the platoon.
3. Once a split maneuver is started, it will either be successfully completed or all nodes will enter the abort handler (liveness condition).

For the purpose of this proof, it will be taken for granted that a split maneuver has been initiated, either by a follower (the LC' node) or the leader node. It makes no difference which mechanism activates the maneuver as far as proving the desired properties of the distributed system is concerned.

1. Let us define an initial system state vector

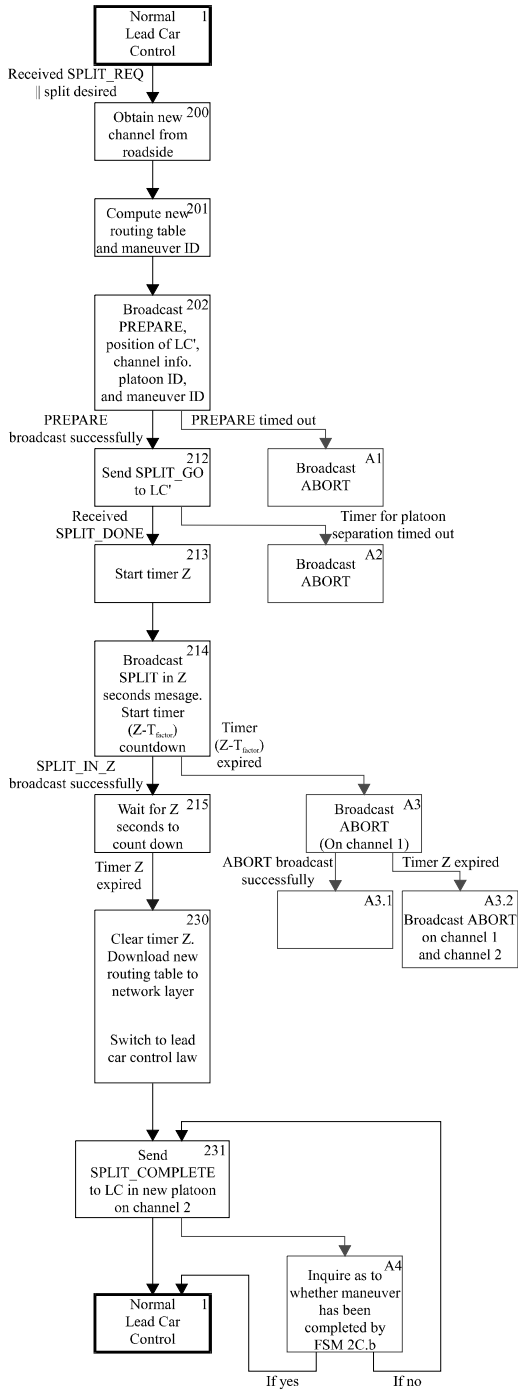
$$V_0 \equiv \{ \quad 1_{1A} \quad | \quad 0_{1B} \quad \cdots \quad | \quad 0_{1C} \quad | \quad 0_{1D} \quad \cdots \quad \}.$$
2. $202_{1A} \rightsquigarrow 100_{1B}$, $202_{1A} \rightsquigarrow 100_{1C}$, $202_{1A} \rightsquigarrow 100_{1D}$, because the transitions between states $0_{1B,1C,1D}$ and state $100_{1B,1C,1D}$ occur as a result of the `prepare` message broadcast in state 202_{1A} .
3. Since we have assumed that acknowledgements to the `prepare` message broadcast in state 212_{1A} are handled transparently by the channel, it can be concluded that $100_{1B} \rightsquigarrow 212_{1A}$, $100_{1C} \rightsquigarrow 212_{1A}$, $100_{1D} \rightsquigarrow 212_{1A}$.
4. From the figures FSM 1B, FSM 1C, and FSM 1D, we can see that $100_{1B} \prec_{im} 130_{1B}$, $100_{1C} \prec_{im} 112_{1C}$ and $100_{1D} \prec_{im} 121_{1D} \prec_{im} 130_{1D}$.
5. $212_{1A} \rightsquigarrow 112_{1C}$ since the transition between state 100_{1C} and 112_{1C} occurs as a result of the `split_go` message sent by state 212_{1A} .
6. From statements (3), (4), and (5), the occurrence of the transition between state 202_{1A} to state 212_{1A} implies that FSM 1C is in state 100_{1C} .
7. $214_{1A} \rightsquigarrow 130_{1B}$ and $214_{1A} \rightsquigarrow 130_{1D}$ since the transitions from 100_{1B} to 130_{1B} and from 121_{1D} to 130_{1D} occur as a result of the `split_in_z` message sent by state 214_{1A} , and also because the transition between 100_{1D} to 121_{1D} occurs automatically (i.e. without any external triggering event).
8. It can be seen from figure FSM 1A, that $212_{1A} \prec_{im} 213_{1A} \prec_{im} 214_{1A}$.
9. From statements (3), (4), (7), and (8), the transition from the occurrence of the transition from state 202_{1A} to 212_{1A} implies that all FSM 1B's are in the state 100_{1B} and all FSM 1D's are in the state 121_{1D} .
10. From statements (6) and (9), we can conclude that the occurrence of the transition from state 202_{1A} to 212_{1A} implies the system state vector

$$V_1 \equiv \{ \quad 212_{1A} \quad | \quad 100_{1B} \quad \cdots \quad | \quad 100_{1C} \quad | \quad 121_{1D} \quad \cdots \quad \}$$
11. Assume that paths A1, A2, and A3 in FSM 1A cause an `abort_maneuver` message to be broadcast to all vehicles. Let us not differentiate between these different abort messages for now.
12. From statements (1), (10), and (11), the occurrence of the transition from state 202_{1A} to state A1 implies a system state vector $V_{1,\delta}$ such that $V_0 < V_{1,\delta} < V_1$. In other words, the system state vector $V_{1,\delta}$ will never be greater than V_1 .

13. From statement (12), it can be concluded that the distributed system can make no forward progress once the `prepare` message has timed out (i.e. state A1 is reached). If we then allow state A1 to rebroadcast the `abort_maneuver` message until it is successfully received and acknowledged by all nodes, then all processes in the distributed system will be guaranteed to enter the abort handler once FSM 1A enters state A1.
14. From figure FSM 1C, we can see that $112_{1C} \prec_{im} 113_{1C}$ and that the transition between the two states is dependent only on the completion of the split transition mode (i.e. the following distance for LC' is increased in preparation of becoming a platoon leader).
15. $113_{1C} \rightsquigarrow 213_{1A}$ since the transition from 212_{1A} to 213_{1A} occurs as a result of the `split_done` message sent by state 113_{1C} .
16. From figure FSM 1A, we can see that $213_{1A} \prec_{im} 214_{1A}$, and that the transition between those two states occurs automatically.
17. From statements (5), (7), (10), (14), (15), and (16), we can conclude that the occurrence of the transition from 212_{1A} to 213_{1A} implies the system state vector $V_2 \equiv \{ 213_{1A} \mid 100_{1B} \ \dots \mid 113_{1C} \mid 121_{1D} \ \dots \}$.
18. From statements (10) and (17), and since $212_{1A} \prec_{im} 213_{1A}$, we can conclude that the occurrence of the transition from state 212_{1A} to state A2 implies the system state vector $V_{2,\delta}$ such that $V_1 < V_{2,\delta} < V_2$. In other words, $V_{2,\delta}$ will never be greater than V_2 .
19. From statement (18), it can be seen that the distributed system will make no forward progress once the transition from 212_{1A} to A2 takes place (i.e. the timer for platoon separation expires). If we then allow state A2 to rebroadcast an `abort_maneuver` message until it is successfully received and acknowledged by all nodes, then all processes in the distributed system are guaranteed to enter the abort handler once FSM 2A enters state A2.
20. From figure FSM 1C, we can see that $113_{1C} \prec_{im} 130_{1C}$, and that the transition between those two states is triggered by the `split_in_z` message sent by state 214_{1A} .
21. We can conclude that $214_{1A} \rightsquigarrow 130_{1C}$ from statements (16), (17), and (20).
22. Since we have assumed that acknowledgements to the `split_in_z` message are handled transparently by the channel, we can conclude that $130_{1B} \rightsquigarrow 215_{1A}$, $130_{1C} \rightsquigarrow 215_{1A}$, $130_{1D} \rightsquigarrow 215_{1A}$. In addition, the transition from 214_{1A} to 215_{1A} will not occur until all FSM 1B's have entered state 130_{1B} and all FSM 1D's have entered state 130_{1D} .
23. From figures FSM 1B, FSM 1C, FSM 1D, we can see that $130_{1B} \prec_{im} 132_{1B}$, $130_{1C} \prec_{im} 132_{1C}$, and $130_{1D} \prec_{im} 132_{1D}$. We can also see from the figures that the transitions between those pairs of states occur automatically.
24. From figure FSM 2A, we can see that the transition from state 214_{1A} to state A3 occurs before timer Z expires, by definition.
25. From statements (7), (16), (21), (22), (23), and (24), we can conclude that the occurrence of the transition from state 214_{1A} to state 215_{1A} implies the system state vector $V_3 \equiv \{ 215_{1A} \mid 132_{1B} \ \dots \mid 132_{1C} \mid 132_{1D} \ \dots \}$.

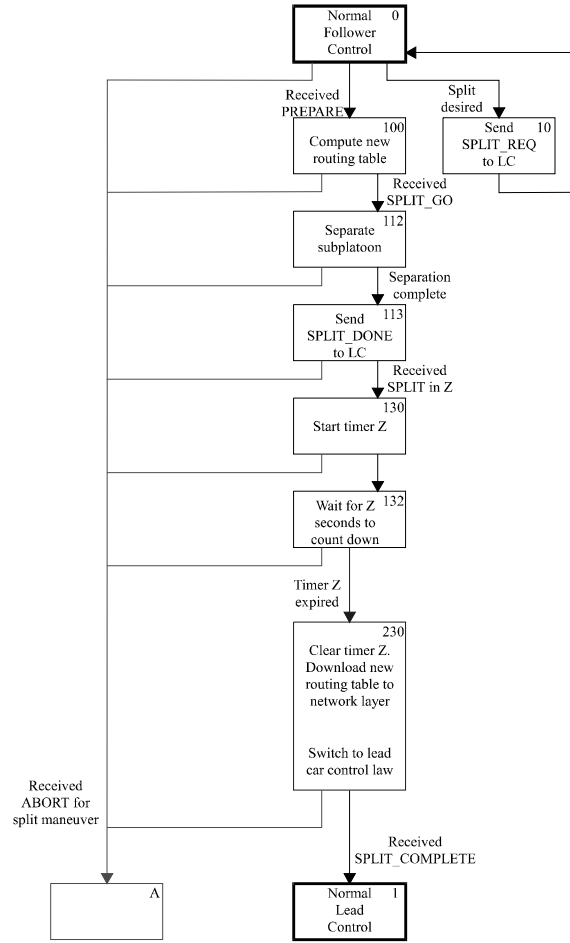
26. From figures FSM 1A, FSM 1B, FSM 1C, and FSM 1D, it can be seen that $214_{1A} \prec_{im} 230_{1A}$, $132_{1B} \prec_{im} 230_{1B}$, $132_{1C} \prec_{im} 230_{1C}$, and $132_{1D} \prec_{im} 230_{1D}$. It can also be seen that the transitions between these state pairs is triggered by the expiration of timer Z.
27. From statements (25) and (26), we can conclude that the states in the system state vector $V_C \equiv \{ 230_{1A} \mid 230_{1B} \ \dots \mid 230_{1C} \mid 230_{1D} \ \dots \}$ are concurrent. Since V_C consists of the states in which reconfiguration occurs, then the reconfiguration states are concurrent, and the second desired property of the distributed system has been proven.
28. In addition to the concurrency property shown in statement (27), we can also show some temporal properties of the transitions between V_3 and V_C . If the timer data that is sent in the message `split_in_z` is updated each time the message is broadcast, then from statement (26) we can conclude that the transitions in each process from the states in vector V_3 to the states in vector V_C occur within $2\rho+\tau$ seconds of each other. Here, ρ is the software delay associated with reading and/or starting a software timer as well as message encoding/decoding, and τ is the transmission delay of the wireless channel.
29. From statements (17), (24), and (25), we can conclude that the occurrence of the transition from state 214_{1A} to state A3 implies a system state vector $V_{3,\delta}$ such that $V_2 < V_{3,\delta} < V_3$.
30. From figure FSM 2A, we can see that the transition from state A3 to state A3.1 will occur before timer Z expires by definition. The occurrence of this transition implies that all nodes have entered the abort handler.
31. The tricky part occurs after timer Z expires and the transition from state A3 to A3.2 occurs. At this point, some follower nodes might have switched over to the new routing tables and the new channel. Therefore, the `abort_maneuver` messages will need to be broadcast on both channel 1 and channel 2 in order to reach all the nodes.
32. Even more serious than the problem in statement (30) is that if the LC' node has changed to the new configuration, then it now functions as a lead node, and can conceivably start a new maneuver while the old one is still in the abort handler. This is a BIG problem that must be eliminated.

Modified Leader



FSM 1A.b

Modified LC'



FSM 1C.b

33. To solve the problem encountered in statement (32), we add a state 231_{1A} as shown in figure FSM 1A.b such that $230_{1A} \prec_{im} 231_{1A}$, and the transition between 230_{1A} and 231_{1A} occurs automatically. We also add a condition on the transition from state 230_{1C} and state 1_{1C} as shown in figure FSM 1C.b.
34. $231_{1A} \rightsquigarrow 1_{1C}$, since the transition from state 230_{1C} to state 1_{1C} occurs as a result of the `split_complete` message sent by state 231_{1A} .
35. From figures FSM 2B and FSM 2D, we can see that $230_{1B} \prec_{im} 0_{1B}$, $230_{1D} \prec_{im} 0_{1D}$, and that the transition between the state pairs is triggered by a timer that is internal to each process.
36. From statements (31), (34), and (35), we can conclude that the transition from state A3 to A3.2 for the modified FSM's implies that some of the follower nodes are now in state 0_{1B} and/or state 0_{1D} , but none have transitioned to state 1. Therefore, within this group of nodes/vehicles, there is still only one lead node capable of controlling maneuvers, and the state of that leader is in the abort handler.
37. From statement (36), if we allow the lead node to broadcast `abort_maneuver` messages from state A3.2, until it is successfully received and acknowledged by all nodes, then we can guarantee that all nodes in the distributed system will enter the abort handler.
38. From figure FSM 1A.b we can see that the $231_{1A} \prec_{im} 1_{1A}$, and that the transition from 231_{1A} to 1_{1A} is occurs when the `split_complete` message has been successfully received and acknowledged by FSM 1C.
39. From statements (27), (28), (33), (34), (35), and (38), we can conclude that the occurrence of the transition from state 231_{2A} to state 1_{2A} implies the system state vector $V_4 \equiv \{ 1_{1A} \mid 0_{1B} \ \cdots \mid 1_{1C} \mid 0_{1D} \ \cdots \}$, which is a successful completion state vector for the split maneuver.
40. From statements (25), (27), and (35), the transition from state 231_{1A} to state A4 implies the system state vector $V_{4,\delta} \equiv \{ A4_{1A} \mid 0_{1B} \ \cdots \mid ?_{1C} \mid 0_{1D} \ \cdots \}$, where the '?' could be one of two cases:
- (a) FSM 1C.b is still in state 230_{1C} , meaning that the `split_complete` message sent by FSM 1A.b in state 231_{1A} was lost, or
 - (b) FSM 1C.b is currently in state 1_{1C} , or has passed through state 1_{1C} and has already initiated another maneuver. This means that the acknowledgement sent by FSM 1C.b in response to the `split_complete` message was lost.
- As stated in case (b), the possibility exists that another maneuver could have been initiated or even completed by the new lead car that was running process FSM 1C.b. As a result, the address "LC of platoon 2" may no longer even exist. This slight problem can be solved by having process FSM 1A.b note the unique vehicle address of the node running process FSM 1C.b when the split maneuver is initiated, and using that unique vehicle address for any inquiries from state A4.
- (41) From statement (40), we can see that the system state vector $V_{4,\delta}$ implied by the transition of process FSM 1A.b into state A4 does not necessitate an abort procedure. From a control standpoint, all the nodes represented in the state vector $V_{4,\delta}$ are in the correct control modes. The only thing that remains is to reassure the lead node (FSM 1A.b) that the node previously labeled as LC' (which was running FSM 1C.b) has made the

transition from 230_{1C} to 1_{1C} , and to trigger that transition if necessary, before the lead node can authorize another maneuver. If we assume that the lead node will continue to inquire as to the state of FSM 1C.b, until a response is obtained, then the maneuver will eventually be completed.

- (42) From FSM 1A.b, FSM 1C.b, and statements (35), the transition from state 230_{1A} to state 231_{1A} implies the system state vector
- $$V_5 \equiv \left\{ \begin{array}{c|c|c|c} 231_{1A} & 0_{1B} & \cdots & 230_{1C} & 0_{1D} & \cdots \end{array} \right\}.$$
- (43) From statements (42) and (34), the transition from state 230_{1C} to state 1_{1C} implies that the process FSM 1C.b has knowledge that the system state vector is now greater than the state vector V_5 . Which in turn implies that the process FSM 1C.b has knowledge that all nodes involved in the maneuver have successfully been reconfigured.

Now, taking all the statements that have been shown above, we can prove that the distributed system defined by the split maneuver state machines has the three desired properties that were specified. From statements (27) and (28) it has already been shown that the reconfiguration states across the distributed system are concurrent, and also that they occur within a bounded time interval of each other. The liveness condition can be shown by collectively looking at statements (13), (19), (30), (36), (37), (39), (40), and (41), we can see that once a split maneuver is initiated, all nodes will either successfully complete or all nodes will enter the abort handler. The safety condition, cannot be guaranteed as it was originally stated. However, a relaxed safety condition where a lead node cannot initiate a maneuver before it knows that all other nodes in its platoon have been successfully reconfigured, can be guaranteed. Taking statements (40) and (42), we can see that the lead nodes cannot initiate another maneuver before it has knowledge that all other nodes involved in the current maneuver have been successfully reconfigured.

9.3.2 Abort Handler for the Platoon Split Maneuver

Now the specifics of the abort handler must be determined. By examining the FSM's in the case of the lossy FIFO channel model, it can be seen that the lead node is responsible for directing the rest of the nodes into entering the abort handler. It can also be seen that there are three stages of entry into the abort handler, denoted by the states $A1_{1A.b}$, $A2_{1A.b}$, and $A3_{1A.b}$ (as stated previously, the state $A4_{1A.b}$ does not really represent an abort state so much as a final verification state for the split maneuver). To simplify things, the three stages of entry into the abort handler for the split maneuver will be referred to as entry stages A1, A2, and A3. At this point, our options are open as to what should be done at each stage of entry. At each stage, we could either try to restore the distributed system (and thus the platoon) to its state before the maneuver began, or we could try to recover from whatever lost message caused the entry into the abort handler and complete the maneuver. In this decision, consideration should be given to the physical state of the vehicles in the platoon as well as the FSM states in which the nodes (i.e. vehicles) find themselves.

At entry stage A1, the two sub platoons have not been physically separated, and it would be equally simple to return the platoon to its starting state (i.e. system state vector V_0) or to try and

recover from the loss of the PREPARE message or from the loss of one or more of its acknowledgements. It was chosen in this case to return the platoon to its starting state.

At entry state A2, the two sub platoons may or may not be physically separated, and this should be determined by querying the node LC' as to what FSM state it is in. At this stage, it might be better to try and recover from the message loss or the separation time-out, since decreasing the spacing between two sub platoons is inherently more dangerous than increasing the spacing between the two sub platoons.

At entry stage A3, the two sub platoons are physically separated (although they are not yet separate in a networking sense), and it is definitely better to try and recover from the message loss that caused the entry into the abort handler.

Chapter 10

Platoon Join Protocol

10.1 Initiating a Platoon Join

In order to maximize the capacity of the automated highway, vehicles should be spaced as closely as possible. Since the intervehicular spacing between platoons is much larger than the intervehicular spacing between vehicles in the same platoon, this means that vehicles should be grouped into fewer, larger platoons in order to increase the capacity of the highway.

Notice that for the platoon split maneuver, the “atomic unit” was a vehicle. In other words, any vehicle in a given platoon was allowed to initiate the maneuver. For the platoon join maneuver, however, the “atomic unit” is now a platoon. Only the platoon leaders are allowed to initiate a join with another platoon leader. It should be noted that a vehicle which is a free agent is considered to be the platoon leader of a single car platoon. Since there are two platoons involved in any join maneuver, we can refer to them as the *initiator* platoon (i.e. the platoon that sends the initial request message), and the *respondent* platoon (i.e. the platoon that responds to the initial request).

As was stated before, a platoon split request should be given higher priority than a platoon join maneuver. The simple reason being that the freedom to leave the highway should be more important (at least from a human driver’s standpoint) than the desire to maximize vehicle capacity on the highway. As a result, a platoon should only participate in a join maneuver when there are no platoon split requests from within the platoon.

10.2 Servicing a Platoon Join Request

Servicing a platoon join request is more complex than servicing a platoon split request. For the split maneuver, the set of participating “atomic units” was a platoon; which had a well defined leader and therefore an obvious choice for an arbiter. For the join maneuver, the set of participating “atomic units” is a group of platoons on a given section of highway. From a networking standpoint (these platoons are connected to each other and to the roadside via a wireless communication system), any pair of these platoons on a given link of the network (i.e. on a given section of highway) are possible candidates for a platoon join. From a physical standpoint, however, only those platoons which are physically adjacent to each other, and are in the same lane can participate in a mutual join maneuver. The fact that the joining platoons need to be in the same lane can be easily remedied from the network standpoint by including the lane number in the address of the platoon leader on the highway/network. The condition that the joining platoons be adjacent is not so easily remedied, and the optimal solution depends heavily on the hardware and position information that is available on the highway. Since the final design of the highway itself is uncertain, let us simply state that it must be determined that the initiator and respondent platoons for the join maneuver are adjacent to each other. Notice that this condition still leaves the question of which of the two platoons involved in the join maneuver is in front of the other. This problem can be solved within the join protocol itself.

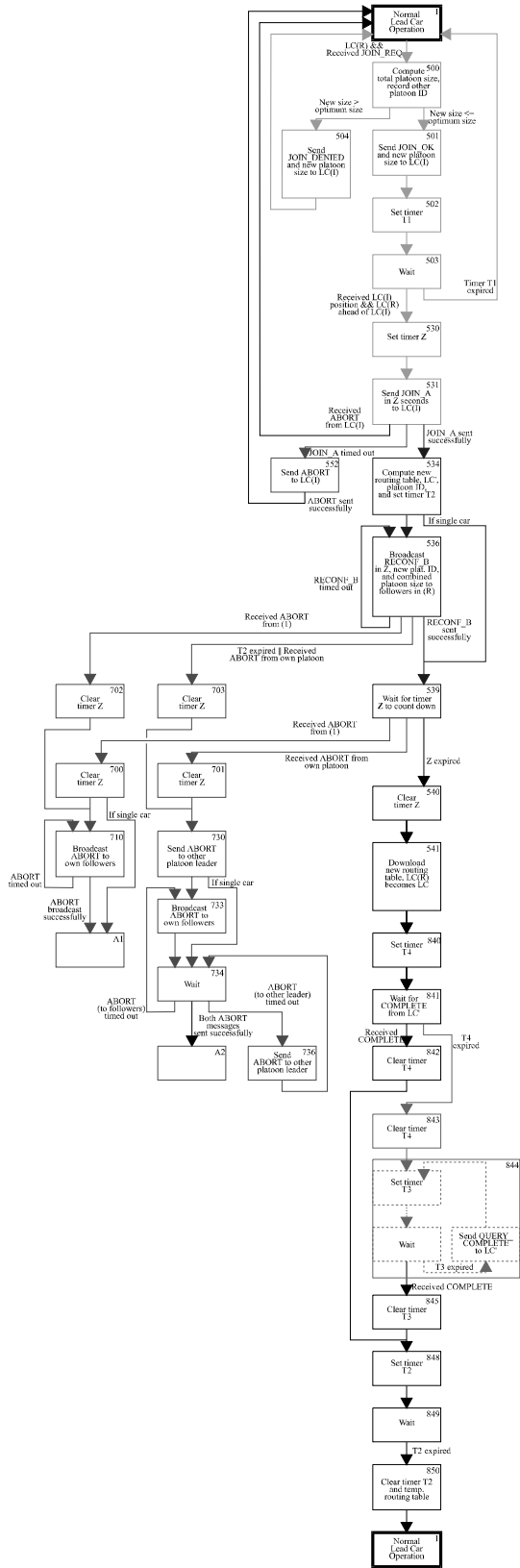
Now that the conditions under which a pair of platoons can participate in a join maneuver is settled, the question arises: If more than two platoons wish to participate in a join maneuver, what entity arbitrates any resource conflicts that might arise. This arbitration could be centralized on a roadside base station, or it could be decentralized in the individual platoon leaders. As there was no clear choice at this point, it was chosen to decentralize this arbitration into each platoon leader. The leader of the respondent platoon would, after considering the final platoon size after a join, either accept or decline the “invitation” to join with the initiator platoon. It should be noted that in order to allow vehicles to exit the AHS when desired, the priority for servicing a split request should be set higher than the priority for servicing a join request. This will result in a slightly decreased vehicle density but should more easily facilitate the exit of a vehicle from the AHS.

10.3 Protocol Design

10.3.1 Case 1: Respondent Platoon in Front of Initiator Platoon

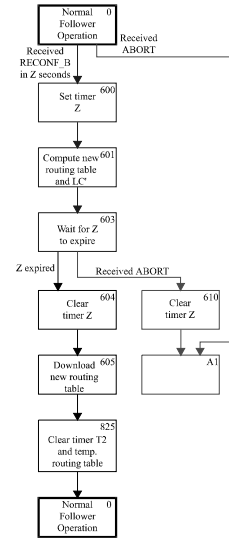
For the join maneuver, there are four classes of vehicles involved: the leaders of the initiator and respondent platoons, and the followers of the initiator and respondent platoons. As with the split maneuver, we start by assuming a lossy FIFO channel between nodes. We will first consider the case in which the respondent platoon is in front of the initiator platoon (case 1). Later we will consider the case in which the initiator platoon is in front of the respondent platoon. The proofs for these two cases are similar, but are different enough that it was prudent to consider them separately. The finite state machines for each of the four classifications of vehicles for case 1 are shown in the figures labeled FSM 2A through FSM 2D.

Leader of Forward Respondent Platoon



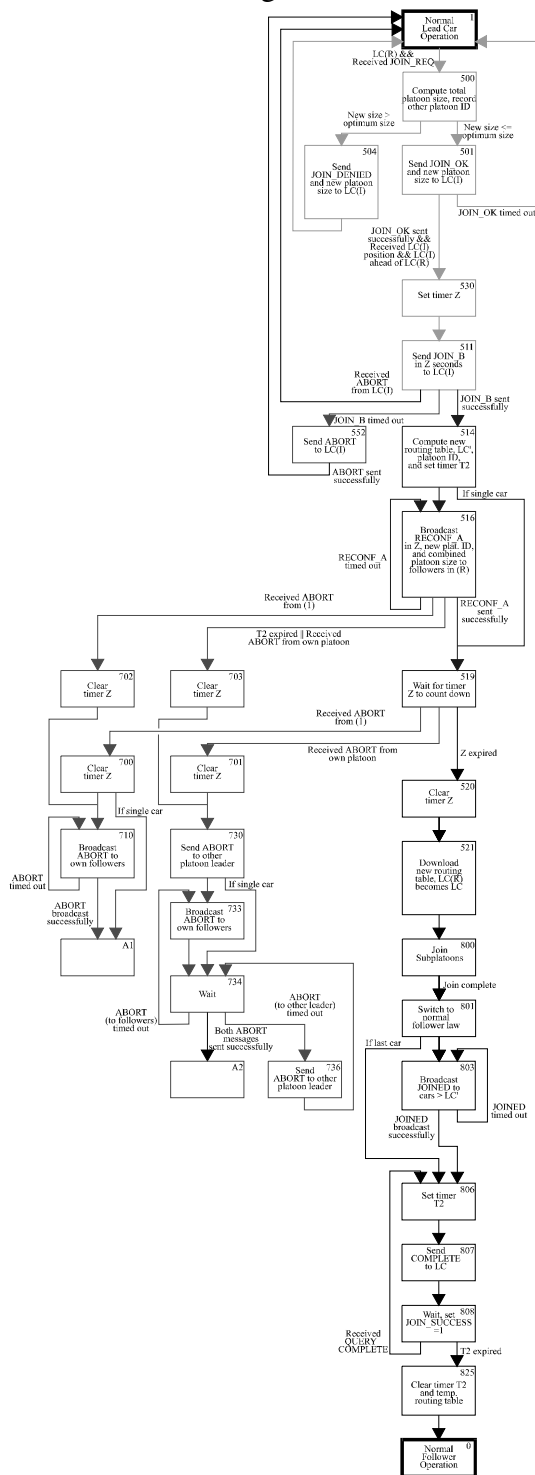
FSM 2A

Follower in Forward Respondent Platoon



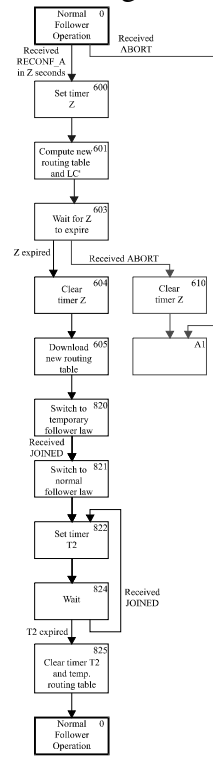
FSM 2B

Leader of Trailing Initiator Platoon



FSM 2C

Follower in Trailing Initiator Platoon



FSM 2D

The desired properties for the platoon join maneuver are the same as for the platoon split maneuver, namely:

1. Another join maneuver cannot be started before the previous one is complete (safety condition).
2. Reconfiguration states occur concurrently in all vehicles within the platoon.
3. Once a join maneuver is started, it will be completed (liveness condition).

Notice that in FSM's 2A and 2C, states 400_{2C} - 406_{2C} and 500_{2A} - 530_{2A} for the two lead nodes do not respond to ACK's to messages sent but to other messages. These states represent the exchange of data that needs to occur before any type of reconfiguration, either physical or network-wise, can take place. In these states, timer T1 acts as an explicit "message timed out" indicator, reflecting the time it would take for an initial message to be sent, acknowledged (which happens transparently in our lossy FIFO channel model), and replied to.

1. Assume an initial state vector $V_0 \equiv \{ 1_{2A} \mid 0_{2B} \cdots \mid 1_{2C} \mid 0_{2D} \cdots \}$.
2. $1_{2A} \prec_{im} 500_{2A}$ and the transition occurs when the JOIN_REQ message sent in 402_{2A} is received by node A.
3. $0_{2B} \prec_{im} 600_{2B}$ and the transition occurs when the RECONF_B message broadcast in 536_{2A} is received by node(s) B.
4. $0_{2B} \prec_{im} A1_{2B}$ and the transition occurs when the ABORT message sent in 710_{2A} is received by node(s) B.
5. $1_{2C} \prec_{im} A1_{2C}$ and the transition occurs when the ABORT message sent in 552_{2A} or in 730_{2A} is received by node C.
6. $1_{2C} \prec_{im} 400_{2C}$ and the transition occurs when node C desires a join maneuver to occur.
7. $0_{2D} \prec_{im} 600_{2D}$ and the transition occurs when the RECONF_A message broadcast in 434_{2C} is received by node(s) D.
8. $0_{2D} \prec_{im} A1_{2D}$ and the transition occurs when the ABORT message broadcast in 710_{2C} or in 733_{2C} is received by node(s) D.
9. From statements (1) through (8), the initial state vector V_0 makes no forward progress until node C wants to initiate a join maneuver, at which point the state vector will transition to $V_1 \equiv \{ 1_{2A} \mid 0_{2B} \cdots \mid 400_{2C} \mid 0_{2D} \cdots \}$.
10. State 400_{2C} transitions to state 1_{2C} if the roadside cannot find a suitable respondent platoon (i.e. no other platoon is within range and in the same lane as the initiator platoon), thus returning the system state vector to V_0 .
11. State 400_{2C} transitions to state 401_{2C} if a suitable respondent platoon is found by the roadside.
12. $401_{2C} \prec_{im} 402_{2C}$ and the transition occurs unconditionally.
13. $402_{2C} \prec_{im} 403_{2C}$ and the transition occurs unconditionally.
14. Assume that node C desires a join maneuver, and that a suitable respondent platoon (i.e. nodes A and B) exist.
15. Under the conditions in statements (1), (9), (12), (13), (14), the state vector will transition from V_0 to V_1 to $V_1 \equiv \{ 1_{2A} \mid 0_{2B} \cdots \mid 403_{2C} \mid 0_{2D} \cdots \}$.
16. $403_{2C} \prec_{im} 404_{2C}$ and the transition occurs when the JOIN_OK message sent in 501_{2A} sent in state 501_{2A} is received by node C.

17. Assume a state vector V_2 . Then from statements (2), (3), (4), (7), (8), and (13), if node C transitions from state 402_{2C} to state 404_{2C} , this implies a state vector $V_3 \equiv \{ 501_{2A} \mid 0_{2B} \cdots \mid 404_{2C} \mid 0_{2D} \cdots \}$.
18. $403_{2C} \prec_{im} 407_{2C}$ and the transition occurs either:
- Upon receiving a JOIN_DENIED message sent from state 504_{2A} , which implies a state vector $V_{4.1} \equiv \{ 504_{2A} \mid 0_{2B} \cdots \mid 407_{2C} \mid 0_{2D} \cdots \}$ or,
 - When timer $T1_{2C}$ (which was set in 502_{2A}) times out, meaning that either the JOIN_REQ message never reached node A, implying a state vector $V_{4.2} \equiv \{ 1_{2A} \mid 0_{2B} \cdots \mid 407_{2C} \mid 0_{2D} \cdots \}$; or that the JOIN_REQ message was received by node A but the reply (either the JOIN_OK or JOIN_DENIED message) was lost, implying state vectors $V_{4.3} \equiv \{ 501_{2A} \mid 0_{2B} \cdots \mid 407_{2C} \mid 0_{2D} \cdots \}$ or $V_{4.4} \equiv \{ 504_{2A} \mid 0_{2B} \cdots \mid 407_{2C} \mid 0_{2D} \cdots \}$ respectively.
19. The system state vector V_3 represents a successful first contact between nodes A and C, while $V_{4.1}$, $V_{4.2}$, $V_{4.3}$, and $V_{4.4}$ represent all possible abort modes up to this point.
20. Let us look at how these abort modes evolve:
- Assume the state vector $V_{4.1}$. $504_{2A} \prec_{im} 1_{2A}$ and $407_{2C} \prec_{im} 1_{2C}$. Both transitions are unconditional. Therefore, $V_{4.1}$ unconditionally transitions to V_0 .
 - Assume the state vector $V_{4.2}$. $407_{2C} \prec_{im} 1_{2C}$ and the transition is unconditional. Therefore, $V_{4.2}$ unconditionally transitions to V_0 .
 - Assume the state vector $V_{4.3}$. $407_{2C} \prec_{im} 1_{2C}$ and the transition is unconditional. $501_{2A} \prec_{im} 502_{2A} \prec_{im} 503_{2A}$ and these two transitions are also unconditional. Therefore $V_{4.3}$ unconditionally transitions to $V_{4.5} \equiv \{ 503_{2A} \mid 0_{2B} \cdots \mid 1_{2C} \mid 0_{2D} \cdots \}$.
 - $503_{2A} \prec_{im} 530_{2A}$ and the transition occurs when the POSITION message sent in state 405_{2C} is received by node A.
 - Even if node C tries to reinitiate a join maneuver, it can progress no further than state 403_{2C} since node A is in state 503_{2A} .
 - From statements (20.c.i) and (20.c.ii), node C is unable to transition to state 530_{2A} from state vector $V_{4.5}$.
 - From statement (20.c.iii), state $V_{4.5}$ must then transition to V_0 when timer $T1_{2A}$ times out.
 - Assume the state vector $V_{4.4}$. $V_{4.4}$ is equivalent to $V_{4.1}$ and follows the same evolution to V_0 .
21. Assume the state vector $V_3 = \{ 501_{2A} \mid 0_{2B} \cdots \mid 404_{2C} \mid 0_{2D} \cdots \}$. $404_{2C} \prec_{im} 405_{2C} \prec_{im} 406_{2C}$ and both transitions are unconditional. In addition, $501_{2A} \prec_{im} 502_{2A} \prec_{im} 503_{2A}$ and both the transitions are unconditional. Therefore, the state vector V_3 will unconditionally transition to $V_5 \equiv \{ 503_{2A} \mid 0_{2B} \cdots \mid 406_{2C} \mid 0_{2D} \cdots \}$.
22. $503_{2A} \prec_{im} 530_{2A}$ and the transition occurs when the POSITION message sent in state 405_{2C} is received by node A.

23. Assume the state vector V_5 . If state 503_{2A} transitions to state 530_{2A} , this implies the state vector $V_{6.1} \equiv \{ 530_{2A} \mid 0_{2B} \cdots \mid 405_{2C} \mid 0_{2D} \cdots \}$ which by statement (21), will unconditionally transition to the state vector $V_{6.2} \equiv \{ 530_{2A} \mid 0_{2B} \cdots \mid 406_{2C} \mid 0_{2D} \cdots \}$.
24. $503_{2A} \prec_{im} 0_{2A}$ and the transition occurs when the timer $T1_{2A}$ which is set in state 502_{2A} expires, implying that the POSITION message sent in state 405_{2C} is lost.
25. $406_{2C} \prec_{im} 430_{2C}$ and the transition occurs when the JOIN_A message sent in state 531_{2A} is received by node C.
26. $406_{2C} \prec_{im} 407_{2C}$ and the transition occurs when timer $T1_{2C}$ which is set in state 405_{2C} expires.
27. $503_{2A} < 531_{2A}$.
28. Assume the state vector V_5 and that the POSITION message sent in state 405_{2C} is lost as in statement (24). By statements (25) and (27), state 406_{2C} cannot transition into state 407_{2C} because state 503_{2A} cannot reach state 531_{2A} under the conditions in statement (24). This implies that state vector V_5 makes no forward progress until the timers $T1_{2A}$ and $T1_{2C}$ both expire, resulting in the state vector $V_7 \equiv \{ 1_{2A} \mid 0_{2B} \cdots \mid 407_{2C} \mid 0_{2D} \cdots \}$, which then unconditionally transitions to V_0 .
29. $530_{2A} \prec_{im} 531_{2A}$ and the transition is unconditional.
30. By statement (29) the state vector $V_{6.2}$ will transition unconditionally to $V_8 \equiv \{ 531_{2A} \mid 0_{2B} \cdots \mid 406_{2C} \mid 0_{2D} \cdots \}$.
31. Up to this point, it has been shown that a successful path towards completing the join maneuver will end with a state vector of V_8 , while all abort modes will return to the initial state vector V_0 . Also, up to this point, only the lead vehicles in the initiator and respondent platoons have been involved, and the point of this exchange of messages is to pass on the data necessary for both platoons to compute the platoon addresses that will result from the join maneuver.
32. $406_{2C} \prec_{im} 430_{2C}$ and the transition occurs when the JOIN_A message sent in state 531_{2A} is received by node C.
33. $406_{2C} \prec_{im} 407_{2C}$ and the transition occurs when timer $T1_{2C}$ which is set in state 405_{2C} expires, implying that the JOIN_A message sent in 531_{2A} was lost.
34. $531_{2A} \prec_{im} 534_{2A}$ and the transition occurs when the JOIN_A message has been sent to node C successfully.
35. By statements (32) and (34), if state 531_{2A} transitions to state 534_{2A} , this implies a state vector $V_9 \equiv \{ 534_{2A} \mid 0_{2B} \cdots \mid 430_{2C} \mid 0_{2D} \cdots \}$.
36. State 531_{2A} transitions to state 1_{2A} if an ABORT message is received by node A from node C, which occurs only in state 730_{2C} .
37. $430_{2C} \prec_{im} 431_{2C}$ and the transition occurs unconditionally.
38. If state 531_{2A} transitions to 532_{2A} , the possibility exists that node C received the JOIN_A message sent by node A, but the corresponding acknowledgement never returns to node A. This means that nodes C and D could enter their reconfiguration states while nodes A and B make no forward progress.

The possibility of statement (38) is very bad in terms of safety. The best way around this problem is to have a single node (i.e. node A) act as the coordinator for the join maneuver instead of having nodes A and C coordinate their own platoon follower vehicle/nodes. This would require a radio or other device that would enable all nodes to transmit and receive messages on two channels at once. This would also allow a single node the possibility of using two addresses at once, one on each channel. The split maneuver could work around the one-channel-at-a-time restriction safely, but because the join maneuver starts out with two channels, this makes it difficult for a single coordinator to handle both platoons with only one single-channel radio. Being able to transmit and receive on two channels simultaneously greatly simplifies the protocol design for both split and join maneuvers. It also eliminates the need for reconfiguration to be concurrent, since one of the two channels will always be functional during the maneuver. The development of protocols that use a dual-channel communication system in each node is the subject of chapter 11.

Chapter 11

Split and Join Maneuver Redesigned for Dual Channel Radios

With the addition of dual-channel capability, the routing table for a vehicle/node needs to be modified slightly. In addition to simply adding a second set of variables to hold the secondary routing table, it must also keep track of which address corresponds to each channel. The following is an example of an independent agent in the process of joining up from behind with a four-car platoon to become a five-car platoon.

```
PrimaryChannel.PlatoonID:      0ABC123
PrimaryChannel.PlatoonNode:    0
PrimaryChannel.PlatoonSize:    1
PrimaryChannel.Channel:       (channel 1)
SecondaryChannel.PlatoonID:    9ZYX876
SecondaryChannel.PlatoonNode:  5
SecondaryChannel.PlatoonSize:  5
SecondaryChannel.Channel:      (channel 2)
```

When not involved in any maneuvers, a vehicle/node uses its primary channel and routing table. Only when a vehicle/node is in the process of a split or join maneuver does it make use of its secondary channel and routing table. After a maneuver is successfully complete, its secondary channel and routing table becomes its primary channel and routing table, and its primary channel and routing table becomes its secondary channel and routing table. The reason we perform this “swapping” instead of simply copying the secondary information into the primary slot is that some vehicle/nodes may still be using the (old) primary channel for control information for the few seconds it takes for all vehicle/nodes to reconfigure their routing tables. The routing table swapping ensures that no control information is lost. Eventually, the channel and routing table information occupying the secondary slot will become unused, and even if some other platoon

starts using that channel, the uniqueness of the platoon ID will ensure that no stray messages go where they don't belong. Having both routing tables around virtually eliminates the need for an abort handler. The only reason an abort condition can occur now is if a vehicle encounters an emergency and becomes unable to continue with the maneuver. However, this type of error, as well as the proper response of a platoon to this type of error, is beyond the scope of this dissertation, and we will not take it into consideration when designing the platoon maneuver protocols. How such an abort condition is detected and the proper responses need to be better defined before they can be incorporated into protocol design. Having dual active routing tables also eliminates the need for the reconfiguration states to be concurrent, since we still have the "old" channel and routing table to fall back on (at least until the next maneuver is initiated).

The actual information needed to specify a channel is dependent on how the communication network layer is written. Since this specification is beyond the scope of this dissertation, the method of specifying a channel is purposely left vague. When the coordination layer wishes to send a message through the communication network layer, it must now indicate which channel (either channel 1 or channel 2 in this dissertation) it wishes the message to be sent on. So the source and destination addressing for each message now becomes,

```
Source.PlatoonID:      0ABC123
Source.PlatoonNode:    0
Source.PlatoonSize:    1
Destination.PlatoonID: 9ZYX876
Destination.PlatoonNode: 1
Destination.PlatoonSize: 5
Channel:                (channel 1)
```

It is implied that the source and destination of a message must be able to access the same channel. Some intelligence must also be added to the software that decodes the control information. The software must be able to tell which primary and secondary source addresses are equivalent, and appropriately sort out the lead vehicle and previous vehicle control information

11.1 Redesigned Split Protocol

The redesigned split maneuver is a great deal simpler than the original version which relies on a single one-channel radio. For the split maneuver, we start out with a single platoon operating over a single channel. After the maneuver is complete, we end up with two platoons operating over two separate channels. It will be assumed that a platoon which desires a split will be able to query the roadside for an unused channel. It is also assumed that the license plate number of the lead vehicle in the platoon will be used as the platoon ID. Therefore, when a follower vehicle wants to split off (and become a lead car itself) it must provide the current lead vehicle with its license plate number to be used as the "new platoon's" ID.

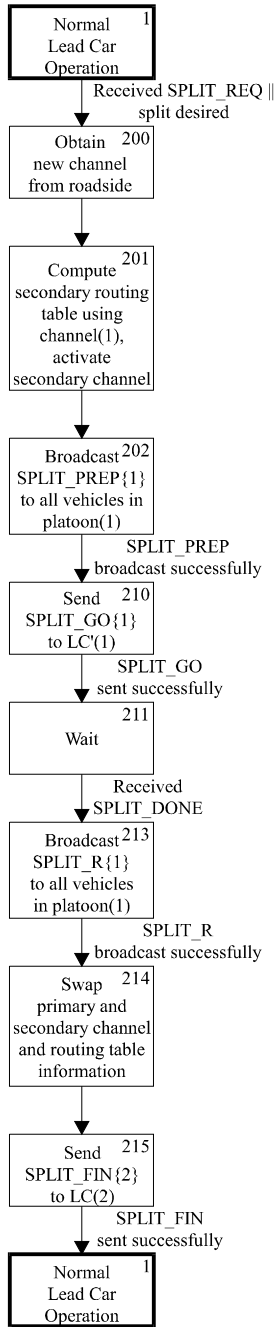
For the purposes of designing the protocol for the split maneuver, the original channel that the platoon is operating over will be referred to as channel 1 (indicated by appending a {1} to the

end of a message name in the finite state machines) and the “new” channel will be referred to as channel 2. Table 11.1 describes the messages used in this version of the split maneuver. Asterisks indicate that the value of the variable is dependent on the situation. Figures FSM 3A through FSM 3D show the finite state machines for the four possible vehicle/node types that can be involved in a split maneuver.

Table 11.1: Messages for the Split Maneuver

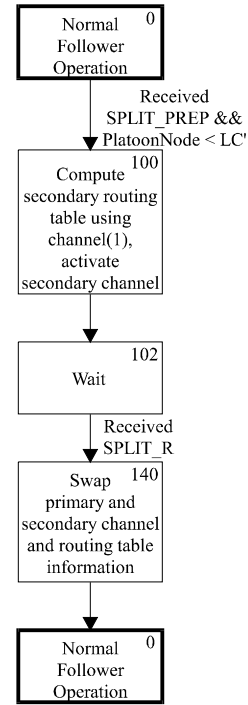
Message	SPLIT_REQ	Message	SPLIT_PREP
Source.PlatoonID	PrimaryID	Source.PlatoonID	PrimaryID
Source.PlatoonNode	*	Source.PlatoonNode	1
Source.PlatoonSize	*	Source.PlatoonSize	*
Destination.PlatoonID	PrimaryID	Destination.PlatoonID	PrimaryID
Destination.PlatoonNode	1	Destination.PlatoonNode	All
Destination.PlatoonSize	*	Destination.PlatoonSize	*
Channel	1	Channel	1
Data	license plate # of LC'	Data 1	node number of LC'
		Data 2	license plate # of LC'
		Data 3	new channel
Message	SPLIT_GO	Message	SPLIT_DONE
Source.PlatoonID	PrimaryID	Source.PlatoonID	PrimaryID
Source.PlatoonNode	1	Source.PlatoonNode	LC'
Source.PlatoonSize	*	Source.PlatoonSize	*
Destination.PlatoonID	PrimaryID	Destination.PlatoonID	PrimaryID
Destination.PlatoonNode	LC'	Destination.PlatoonNode	1
Destination.PlatoonSize	*	Destination.PlatoonSize	*
Channel	1	Channel	1
Data	(none)	Data	(none)
Message	SPLIT_R	Message	SPLIT_FIN
Source.PlatoonID	PrimaryID	Source.PlatoonID	PrimaryID
Source.PlatoonNode	1	Source.PlatoonNode	1 or 0
Source.PlatoonSize	*	Source.PlatoonSize	*
Destination.PlatoonID	PrimaryID	Destination.PlatoonID	SecondaryID
Destination.PlatoonNode	All	Destination.PlatoonNode	1 or 0
Destination.PlatoonSize	*	Destination.PlatoonSize	*
Channel	1	Channel	2
Data	(none)	Data	(none)

Lead vehicle in the platoon



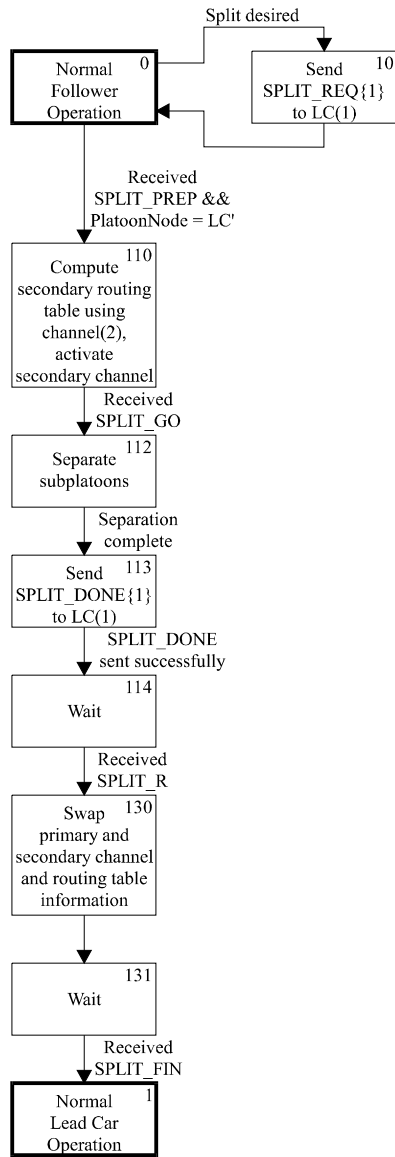
FSM 3A

Followers with node numbers < LC'



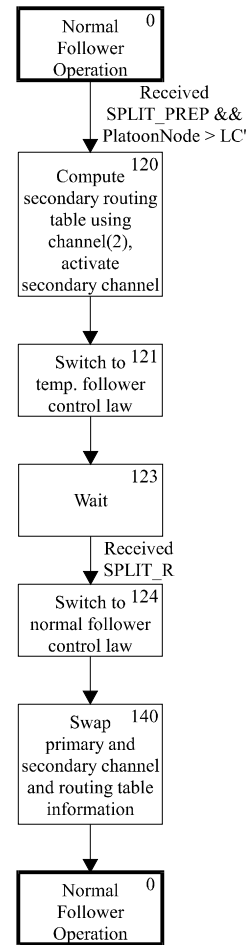
FSM 3B

LC' (the vehicle that becomes a lead car after the split is complete)



FSM 3C

Followers with node numbers > LC'



FSM 3D

The desired properties of the distributed system are similar to the desired properties of the original system undergoing a split maneuver. Notice that the concurrency condition on the reconfiguration states is no longer necessary.

1. Another split maneuver cannot be started before the previous one is successfully completed (safety condition). By "successfully complete", it is meant that all nodes alter their routing tables correctly and reenter normal leader/follower modes.
2. Once a split maneuver is started, it will be successfully completed (liveness condition).

The proof of correctness for the split maneuver protocol shown in FSM 3A through FSM 3D is as follows:

1. Assume an initial state vector $V_0 \equiv \{ 1_{3A} \mid 0_{3B} \cdots \mid 0_{3C} \mid 0_{3D} \cdots \}$.
2. $1_{3A} \prec_{im} 200_{3A}$ and the transition occurs when:
 - a. The SPLIT_REQ message sent in state 10_{3C} is received by node C, indicating that $10_{3C} \rightsquigarrow 200_{3A}$, or
 - b. Node A wants to split up the platoon. In this case, the node number of node C can be arbitrarily chosen by node A to be any node number in the platoon greater than one.
3. $0_{3B} \prec_{im} 100_{3B}$ and the transition occurs when the SPLIT_PREP message sent in state 202_{3A} is received by node B, indicating that $202_{3A} \rightsquigarrow 100_{3B}$.
4. $0_{3D} \prec_{im} 120_{3D}$ and the transition occurs when the SPLIT_PREP message sent in state 202_{3A} is received by node D, indicating that $202_{3A} \rightsquigarrow 120_{3D}$.
5. $0_{3C} \prec_{im} 10_{3C}$ and the transition occurs when node C wants to initiate a split maneuver.
6. $10_{3C} \prec_{im} 0_{3C}$ and the transition occurs unconditionally.
7. $0_{3C} \prec_{im} 110_{3C}$ and the transition occurs when the SPLIT_PREP message sent in state 202_{3A} is received by node C, indicating that $202_{3A} \rightsquigarrow 110_{3C}$.
8. $1_{3A} < 202_{3A}$.
9. By statements (2) through (8), the initial state vector V_0 makes no forward progress if no vehicle/node wants to initiate a split maneuver. In the case that a vehicle does want to initiate a platoon split, there are two possibilities:
 - a. The lead vehicle wants to initiate a platoon split (whether to facilitate an exit from the highway or just to reduce the size of the platoon), in which case by statement (2b) the state vector will transition from V_0 to $V_1 \equiv \{ 200_{3A} \mid 0_{3B} \cdots \mid 0_{3C} \mid 0_{3D} \cdots \}$.
 - b. A follower wants to initiate a platoon split, in which case by statement (5), the state vector will transition from V_0 to $V_{2.1} \equiv \{ 1_{3A} \mid 0_{3B} \cdots \mid 10_{3C} \mid 0_{3D} \cdots \}$, and then by statement (6), $V_{2.1}$ will unconditionally transition to $V_{2.2} \equiv \{ 1_{3A} \mid 0_{3B} \cdots \mid 0_{3C} \mid 0_{3D} \cdots \} = V_0$.
10. From statements (9b) and (2a), if the SPLIT_REQ message sent in state 10_{3C} is lost, the state vector $V_{2.1}$ will revert back to V_0 , and while the maneuver does not start, no harm is done to the system safety-wise. If, however, the SPLIT_REQ message is received by node A, the state vector $V_{2.1}$ will transition to V_1 .
11. By statement (10), if any vehicle/node wishes to initiate a split, the system state vector will either end up in V_0 (as a result of a packet loss) or V_1 . If the nodes which desire a platoon split are persistent, then eventually the SPLIT_REQ message will be received by node A, and the system state vector will always end up at V_1 . The only variable in this procedure is the platoon node number of the vehicle/node designated as LC'.
12. $200_{3A} \prec_{im} 201_{3A} \prec_{im} 202_{3A}$ and both the transitions occur unconditionally.
13. Assume a state vector V_1 . Then by statement (12), V_1 will unconditionally transition to $V_3 \equiv \{ 202_{3A} \mid 0_{3B} \cdots \mid ?_{3C} \mid 0_{3D} \cdots \}$ where the "?" could either be state 0_{3C} or state 10_{3C} .

14. $202_{3A} \prec_{im} 210_{3A}$ and the transition occurs when node A has received the acknowledgements to the SPLIT_PREP message sent in state 202_{3A} from all nodes in platoon(1).
15. $202_{3A} < 210_{3A} < 213_{3A}$.
16. $100_{3B} \prec_{im} 102_{3B}$ and the transition is unconditional.
17. $102_{3B} \prec_{im} 140_{3B}$ and the transition occurs when the SPLIT_R message sent in state 213_{3A} is received by node B, indicating that $213_{3A} \rightsquigarrow 140_{3B}$.
18. $110_{3C} \prec_{im} 112_{3C}$ and the transition occurs when the SPLIT_GO message sent in state 210_{3A} is received by node C, indicating that $210_{3A} \rightsquigarrow 112_{3C}$.
19. $120_{3D} \prec_{im} 121_{3D} \prec_{im} 123_{3D}$ and both of the transitions are unconditional.
20. $123_{3D} \prec_{im} 124_{3D}$ and the transition occurs when the SPLIT_R message sent in state 213_{3A} is received by node D, indicating that $213_{3A} \rightsquigarrow 124_{3D}$.
21. Assume a state vector V_3 . By statements (3), (4), (7), and (14) through (20), if state 202_{3A} transitions to state 210_{3A} , this indicates that V_3 has transitioned into $V_4 \equiv \{ 210_{3A} \mid 102_{3B} \cdots \mid 110_{3C} \mid 123_{3D} \cdots \}$.
22. With the state vector V_4 , all vehicle/nodes have computed and “activated” both primary and secondary channels and routing tables.
23. Assume a state vector V_4 . By statements (15), (17), (18), and (20), if state 210_{3A} transitions to state 211_{3A} , that implies a state vector $V_5 \equiv \{ 211_{3A} \mid 102_{3B} \cdots \mid 112_{3C} \mid 123_{3D} \cdots \}$.
24. $112_{3C} \prec_{im} 113_{3C}$ and the transition occurs when the subplatoons have been separated.
25. $211_{3A} \prec_{im} 213_{3A}$ and the transition occurs when the SPLIT_DONE message sent in state 113_{3C} is received by node A, indicating that $113_{3C} \rightsquigarrow 213_{3A}$.
26. Assume that platoon separation always completes. By statements (24) and (25), this means that V_5 will transition to $V_6 \equiv \{ 211_{3A} \mid 102_{3B} \cdots \mid 113_{3C} \mid 123_{3D} \cdots \}$.
27. $213_{3A} \prec_{im} 214_{3A}$ and the transition occurs when node A receives acknowledgements corresponding to the SPLIT_R message sent in 213_{3A} from all nodes in platoon(1).
28. Assume the state vector V_6 . By statements (17), (20), (25), and (27), if state 113_{3C} transitions to state 114_{3C} this indicates that V_6 has transitioned into $V_7 \equiv \{ 213_{3A} \mid 102_{3B} \cdots \mid 114_{3C} \mid 123_{3D} \cdots \}$.
29. $114_{3C} \prec_{im} 130_{3C}$ and the transition occurs when the SPLIT_R message sent in 213_{3A} is received by node C.
30. By statements (15), (17), (20), and (29), state vector V_7 can make no forward progress until node A broadcasts the SPLIT_R message.
31. $140_{3B} \prec_{im} 0_{3B}$ and the transition is unconditional.
32. $124_{3D} \prec_{im} 140_{3D} \prec_{im} 0_{3D}$ and both the transitions are unconditional.
33. $130_{3C} \prec_{im} 131_{3C}$ and the transition is unconditional.
34. $131_{3C} \prec_{im} 1_{3C}$ and the transition occurs when the SPLIT_FIN message sent in state 215_{3A} is received by node C.
35. $214_{3A} \prec_{im} 215_{3A}$ and the transition is unconditional.

36. Assume the state vector V_7 . By statements (17), (20), (29), and (31) through (35), if state 213_{3A} transitions to state 214_{3A} , this indicates a state vector $V_8 \equiv \{ 214_{3A} \mid 0_{3B} \cdots \mid 131_{3C} \mid 0_{3D} \cdots \}$. At this point, all vehicles are operating under the “new” routing tables (i.e. the secondary and primary channel and routing table information have swapped places). However, notice that nodes A and C cannot initiate another maneuver yet. This is as intended.
37. By statement (35), the state vector V_8 unconditionally transitions to $V_9 \equiv \{ 215_{3A} \mid 0_{3B} \cdots \mid 131_{3C} \mid 0_{3D} \cdots \}$.
38. Assume the state vector V_9 . By statement (34), if state 215_{3A} transitions to state 1_{3A} , this indicates a state vector $V_{10.1} \equiv \{ 1_{3A} \mid 0_{3B} \cdots \mid 1_{3C} \mid 0_{3D} \cdots \}$. This indicates a successful completion of a platoon split maneuver.
39. Assume the state vector V_9 . If node C receives the SPLIT_FIN message, but node A does not receive the corresponding acknowledgement, then this implies the state vector $V_{10.2} \equiv \{ 215_{3A} \mid 0_{3B} \cdots \mid 1_{3C} \mid 0_{3D} \cdots \}$.

Notice that under the conditions of statement (39), it is possible for node C to initiate another maneuver, but that node A is still able to resend the SPLIT_FIN message to node C since it is addressed to the lead vehicle of platoon(2). While node A is unable to initiate another maneuver, there is nothing to stop node C from doing so under the conditions of statement (39). This technically violates the safety condition that no other maneuver should be able to start before this split maneuver is complete. This is a problem that cannot be solved with a distributed system using a lossy FIFO channel, since while we can guarantee that the system cannot make forward progress past some state vector, once we allow forward progress the nature of that progress cannot be controlled.

Notice that even though the safety condition is technically violated, no real harm will come to the vehicle nodes from a control standpoint, since they have essentially been operating under the “new” channel and routing table since achieving the state vector V_8 . Thus, while we cannot satisfy the originally stated safety condition, we can satisfy a modified safety condition:

1. No node will be left operating with an invalid address during or after the maneuver.

Now we only need to worry about the liveness condition. Notice that node A can always contact node C (since the unique license plate number of node C was used for its PlatoonID), even if it must use the roadside WAN to forward the SPLIT_FIN message. Thus node A will eventually transition into its normal lead control state (1_{3A}) and complete the final step in the split maneuver without the finite state machine of node C taking any further action. All that is necessary is for the communication software (which is implicit in our channel model) on node C to successfully acknowledge the receipt of the SPLIT_FIN message. With our lossy FIFO channel model, this happens transparently from the standpoint of the maneuver protocols. As the end state given in statement (38) satisfies both safety and liveness conditions, the split protocol has now been proven to satisfy the desired liveness condition, and a relaxed safety condition.

11.2 Combined Finite State Machine for the Platoon Split Protocol

Figure 11.1 shows the finite state machine that results when FSM 3A through FSM 3D are combined. Implementing the finite state machine in figure 11.1 in each vehicle/node allows each vehicle/node the ability to act as any one of the four node types during a platoon split maneuver. Notice that in figure 11.1 there are two boxes labeled “140”. These are, in fact, the same state.

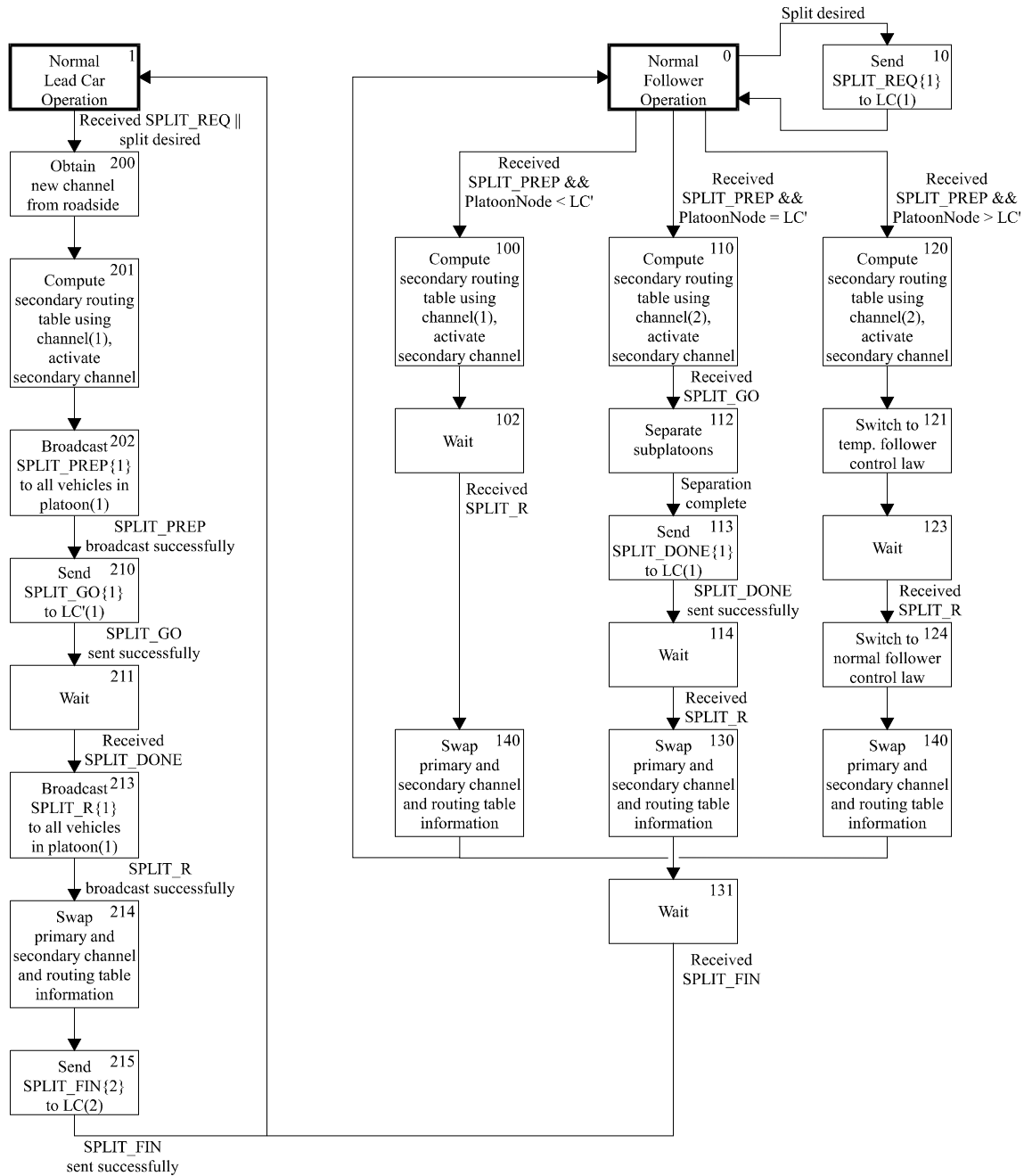


Figure 11.1: Platoon Split Protocol

11.3 Redesigned Join Protocol

This redesigned join maneuver places a bit more responsibility in setting up the first contact situation between initiator and respondent platoons on the roadside stations. The initiator will indicate its desire to take part in a join maneuver by sending a JOIN_QUERY message to the roadside (along with its platoon ID, node number, and platoon size as part of the source address of the message). The roadside will have the responsibility of determining whether there are any suitable respondent platoons by using the following criterion:

1. The joining of two platoons may not result in a platoon that exceeds the maximum platoon size.
2. The two platoons must occupy the same lane of traffic and be adjacent to one another.

If the roadside determines that a suitable platoon exists, it will reply to the lead vehicle/node of the initiator platoon with a JOIN_PARTN message, indicating the address and channel of the suitable join partner. If, however, the roadside determines that there are no suitable join partners available, it will reply to the lead vehicle/node of the initiator platoon with a NO_JOIN message.

The issue of first contact between the two platoons is more complex than for the split maneuver since both initiator and respondent are themselves lead vehicle/nodes and are capable of initiating other maneuvers. In the redesigned join maneuver protocol, the states dealing with first contact between two platoons are similar as in the original protocol. Now, however, only the JOIN_QUERY, JOIN_PARTN, and NO_JOIN messages (see table 11.2 and FSM 4A through 4D) are sent over the roadside wide area network (WAN). All other messages are sent directly from one platoon to the other using their primary and secondary radios. Taking this change into account, the duration of the T1 timer is set so that it does not expire before the initial message and the message intended as a reply can be sent and returned (perhaps even allowing for any transparent resends that the channel might need to perform). Using this timer results in a “cleaner” protocol than if we had designed the finite state machine to react to message acknowledgements in the first contact stage.

In the finite state machines, we must also indicate which channels the messages are to be sent over. We will have two channels when we begin a join maneuver, the one used by the initiator platoon and the one used by the respondent platoon. The two channels will be indicated by a {I} for initiator, {R} for respondent, {1} for the primary channel, and {2} for the secondary channel. For example, if a JOIN_OK message is being sent from the respondent platoon to the initiator platoon using the channel of the initiator platoon, this will be indicated in the finite state machines by JOIN_OK{I}. The leader and follower vehicle/nodes of the two platoons are differentiated, as before, by a (I) or (R).

Table 11.2 lists the messages used in the platoon join maneuver, and their contents. In table 11.2, an asterisk indicates that the value is situation dependent.

Table 11.2: Messages for the Join Maneuver

Message	JOIN_QUERY	Message	NO_JOIN
Source.PlatoonID	InitiatorID	Source.PlatoonID	Roadside
Source.PlatoonNode	InitiatorNode (0 or 1)	Source.PlatoonNode	N/A
Source.PlatoonSize	*	Source.PlatoonSize	N/A
Destination.PlatoonID	Roadside	Destination.PlatoonID	InitiatorID
Destination.PlatoonNode	N/A	Destination.PlatoonNode	0 or 1
Destination.PlatoonSize	N/A	Destination.PlatoonSize	*
Channel	RoadsideChannel	Channel	RoadsideChannel
Data 1	Lane #	Data	(none)
Data 2	Absolute Position of LC(I)		
Message	JOIN_PARTN	Message	JOIN_REQ
Source.PlatoonID	Roadside	Source.PlatoonID	InitiatorID
Source.PlatoonNode	N/A	Source.PlatoonNode	InitiatorNode
Source.PlatoonSize	N/A	Source.PlatoonSize	InitiatorSize
Destination.PlatoonID	InitiatorID	Destination.PlatoonID	RespondentID
Destination.PlatoonNode	0 or 1	Destination.PlatoonNode	RespondentNode
Destination.PlatoonSize	*	Destination.PlatoonSize	RespondentSize
Channel	RoadsideChannel	Channel	RespondentChannel
Data 1	RespondentID, RespondentNode, RespondentSize	Data	Absolute Position of LC(I)
Data 2	RespondentChannel		
Message	JOIN_ACCTP	Message	JOIN_PROCD
Source.PlatoonID	RespondentID	Source.PlatoonID	InitiatorID
Source.PlatoonNode	RespondentNode (0 or 1)	Source.PlatoonNode	InitiatorNode (0 or 1)
Source.PlatoonSize	RespondentSize	Source.PlatoonSize	InitiatorSize
Destination.PlatoonID	InitiatorID	Destination.PlatoonID	RespondentID
Destination.PlatoonNode	InitiatorNode	Destination.PlatoonNode	RespondentNode
Destination.PlatoonSize	InitiatorSize	Destination.PlatoonSize	RespondentSize
Channel	InitiatorChannel	Channel	RespondentChannel
Data	(none)	Data	(none)
Message	JN_ABORT	Message	JOIN_OK
Source.PlatoonID	RespondentID	Source.PlatoonID	RespondentID
Source.PlatoonNode	RespondentNode (0 or 1)	Source.PlatoonNode	0 or 1
Source.PlatoonSize	RespondentSize	Source.PlatoonSize	RespondentSize
Destination.PlatoonID	InitiatorID	Destination.PlatoonID	InitiatorID
Destination.PlatoonNode	InitiatorNode	Destination.PlatoonNode	0 or 1
Destination.PlatoonSize	InitiatorSize	Destination.PlatoonSize	InitiatorSize
Channel	InitiatorChannel	Channel	InitiatorChannel
Data	(none)	Data 1	RespondentChannel
		Data 2	Position of (R) relative to (I)

Table 11.2 (continued)

Message	RECONF_A	Message	RECONF_B
Source.PlatoonID	*	Source.PlatoonID	*
Source.PlatoonNode	0 or 1	Source.PlatoonNode	0 or 1
Source.PlatoonSize	*	Source.PlatoonSize	*
Destination.PlatoonID	*	Destination.PlatoonID	*
Destination.PlatoonNode	All	Destination.PlatoonNode	All
Destination.PlatoonSize	*	Destination.PlatoonSize	*
Channel	PrimaryChannel of sender	Channel	SecondaryChannel of sender
Data 1	Combined platoon size	Data 1	Combined platoon size
Data 2	PrimaryChannel of sender	Data 2	PrimaryChannel of sender
Message	JOIN_PLAT	Message	JOINED
Source.PlatoonID	PrimaryID	Source.PlatoonID	PrimaryID
Source.PlatoonNode	1	Source.PlatoonNode	LC'
Source.PlatoonSize	*	Source.PlatoonSize	*
Destination.PlatoonID	PrimaryID	Destination.PlatoonID	PrimaryID
Destination.PlatoonNode	LC'	Destination.PlatoonNode	1
Destination.PlatoonSize	*	Destination.PlatoonSize	*
Channel	PrimaryChannel	Channel	PrimaryChannel
Data	(none)	Data	(none)
Message	JOIN_FIN		
Source.PlatoonID	PrimaryID		
Source.PlatoonNode	1		
Source.PlatoonSize	*		
Destination.PlatoonID	PrimaryID		
Destination.PlatoonNode	All		
Destination.PlatoonSize	*		
Channel	PrimaryChannel		
Data	(none)		

There are two types of platoon join: one in which the initiator platoon is ahead (and in the same lane as) the respondent platoon; and one in which the initiator platoon is behind the respondent platoon. Protocols will be designed for both cases, and the proof of correctness for each case will be done separately. The two desired conditions for the distributed system formed by the vehicle/nodes are as follows.

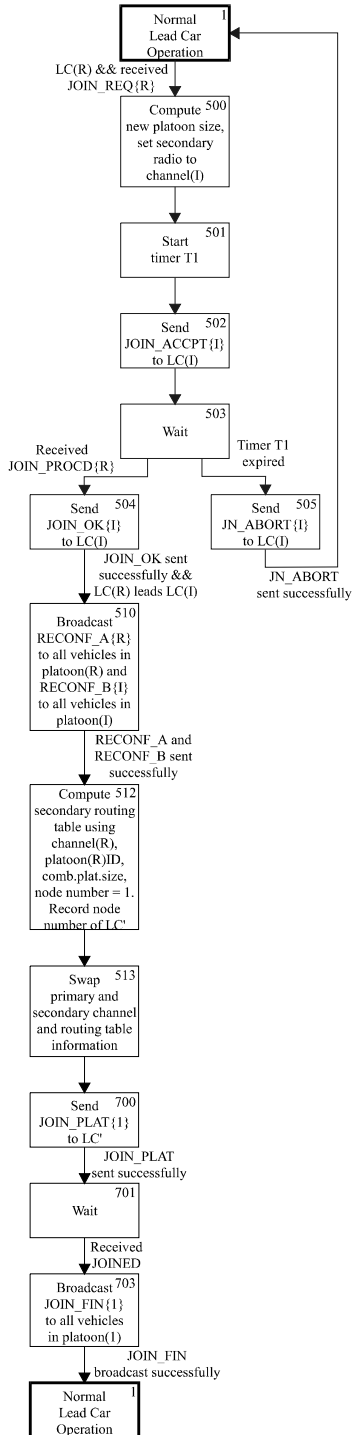
1. Another join maneuver cannot be started before the previous one is successfully completed (safety condition). By "successfully complete", it is meant that all nodes alter their routing tables correctly and reenter normal leader/follower modes.
2. Once a join maneuver is started, it will be successfully completed (liveness condition).

11.3.1 Case 1: Respondent Platoon Leading Initiator Platoon

FSM 4A through FSM 4D describe the protocols for each of the four possible types of vehicle/nodes in a join maneuver. Notice that the FSM's are designed such that the JOIN_REQ and JOIN_ACCPT messages are presented to the channel only once in each join maneuver (although the channel might try to resend a message multiple times, the resends are transparent to the maneuver protocol). Note that the duration of timer T_{14C} is such that it will not expire before the JOIN_ACCPT message is received by node C, assuming that both JOIN_REQ and JOIN_ACCPT messages are received by their destination nodes. Similarly, the duration of timer

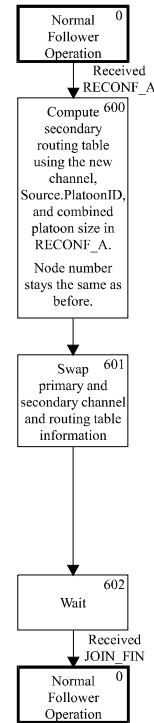
$T1_{4A}$ is such that it will not expire before JOIN_PROCD message is received by node A, assuming that both JOIN_ACCPT and JOIN_PROCD messages are received by their destination nodes.

Lead Vehicle of Respondent Platoon



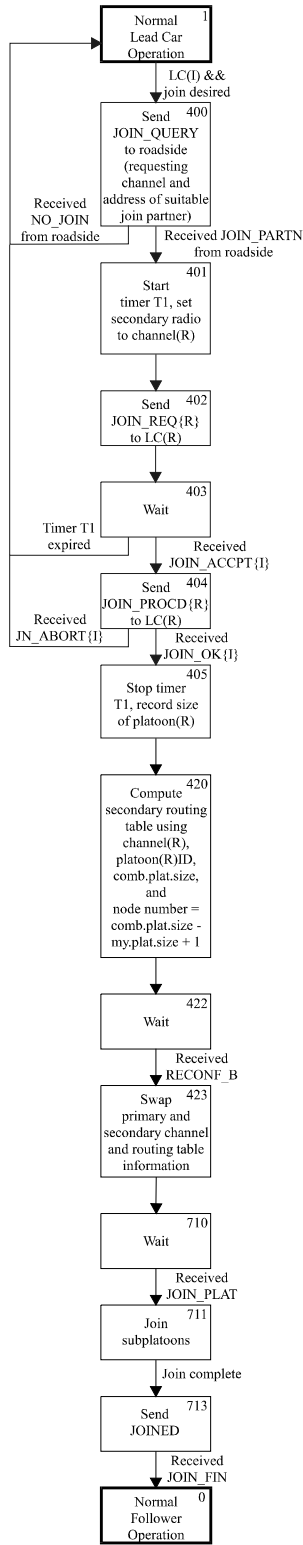
FSM 4A

Follower in Respondent Platoon



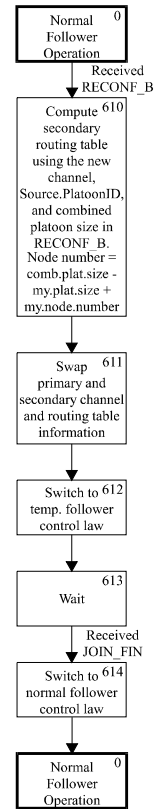
FSM 4B

Lead Vehicle of Initiator Platoon



FSM 4C

Follower in Initiator Platoon



FSM 4D

1. Assume an initial system state vector $V_0 \equiv \{ 1_{4A} \mid 0_{4B} \cdots \mid 1_{4C} \mid 0_{4D} \cdots \}$. In this case, node A is the lead vehicle of the respondent platoon and node C is the lead vehicle of the initiator platoon.
2. $1_{4A} \prec_{im} 500_{4A}$ and the transition occurs when the JOIN_REQ message sent in state 402_{4C} is received by node A, which indicates that $402_{4C} \rightsquigarrow 500_{4A}$.
3. $0_{4B} \prec_{im} 600_{4B}$ and the transition occurs when the RECONF_A message sent in state 510_{4A} is received by node B, which indicates that $510_{4A} \rightsquigarrow 600_{4B}$.
4. $1_{4C} \prec_{im} 400_{4C}$ and the transition occurs when node C wants to initiate a join maneuver.
5. $0_{4D} \prec_{im} 610_{4D}$ and the transition occurs when the RECONF_B message sent in 510_{4A} is received by node D, which indicates that $510_{4A} \rightsquigarrow 610_{4B}$.
6. $1_{4A} < 500_{4A} < 502_{4A} < 503_{4A} < 504_{4A} < 510_{4A} < 512_{4A}$.
7. $1_{4C} < 402_{4C}$.
8. By statements (2) through (7), the initial system state vector V_0 can make no forward progress unless node C wants to initiate a join maneuver.
9. Assume that node C wishes to initiate a join maneuver.
10. $400_{4C} \prec_{im} 401_{4C}$ and the transition occurs when the JOIN_PARTN message sent by the roadside is received by node C.
11. $400_{4C} \prec_{im} 1_{4C}$ and the transition occurs when the NO_JOIN message sent by the roadside is received by node C.
12. By statements (2) through (7) and assumption (9), the initial system state vector V_0 will transition to $V_1 \equiv \{ 1_{4A} \mid 0_{4B} \cdots \mid 400_{4C} \mid 0_{4D} \cdots \}$.
13. Assume the system state vector V_1 . If no suitable join partner platoon exists, the roadside will send the NO_JOIN message to node C, and the system state vector will transition to the initial state vector V_0 by statement (11).
14. Assume the system state vector V_1 . If a suitable join partner exists, the roadside will send the JOIN_PARTN message to node C, and the system state vector will transition to $V_2 \equiv \{ 1_{4A} \mid 0_{4B} \cdots \mid 401_{4C} \mid 0_{4D} \cdots \}$.
15. $401_{4C} \prec_{im} 402_{4C} \prec_{im} 403_{4C}$ and the transitions are both unconditional. The JOIN_REQ message will be sent in state 402_{4C} .
16. $500_{4A} \prec_{im} 501_{4A} \prec_{im} 502_{4A} \prec_{im} 503_{4A}$ and all of these transitions are unconditional. The JOIN_ACCPT message will be sent in state 502_{4A} .
17. $403_{4C} \prec_{im} 1_{4C}$ and the transition occurs when the timer $T1_{4C}$ (which is set in state 401_{4C}) expires.
18. $403_{4C} \prec_{im} 404_{4C}$ and the transition occurs when the JOIN_ACCPT message sent in state 502_{4A} is received by node C, which indicates that $502_{4A} \rightsquigarrow 404_{4C}$.
19. Assume the system state vector V_2 . By statements (2), (3), (5), (6), (15), and (17), if the JOIN_REQ message sent in state 402_{4C} is not received by node A, then the system state vector will transition from V_2 to $V_3 \equiv \{ 1_{4A} \mid 0_{4B} \cdots \mid 403_{4C} \mid 0_{4D} \cdots \}$. The system state will stay in V_3 until timer $T1_{4C}$ (which was set in state 401_{4C}) expires, at which time the system state will transition from V_3 to V_0 . Timer $T1_{4C}$ exists in the join protocol so that node C has a way of exiting V_3 under the aforementioned conditions.

20. Assume the system state vector V_2 . By statements (2), (3), (5), (6), (15), and (16), if the JOIN_REQ message sent in state 402_{4C} is received by node A, then the system state will transition from V_2 to $V_4 \equiv \{ 503_{4A} \mid 0_{4B} \cdots \mid 403_{4C} \mid 0_{4D} \cdots \}$.
21. $503_{4A} \prec_{im} 504_{4A}$ and the transition occurs when JOIN_PROCD message sent in state 404_{4C} is received by node A, which indicates that $404_{4C} \rightsquigarrow 504_{4A}$.
22. $503_{4A} \prec_{im} 505_{4A}$ and the transition occurs when the timer $T1_{4A}$ (which was set in state 501_{4A}) expires.
23. $504_{4A} \prec_{im} 510_{4A}$ and the transition occurs when the JOIN_OK message sent in state 504_{4A} is successfully sent to node C (i.e. the acknowledgement for this message is received by node A). Since FSM 4A remains in state 504_{4A} , the JOIN_OK message will be resent until the transition condition is satisfied.
24. $505_{4A} \prec_{im} 1_{4A}$ and the transition occurs when the JN_ABORT message sent in state 505_{4A} is successfully sent to node C (i.e. the acknowledgement for this message is received by node A). Since FSM 4A remains in state 505_{4A} , the JN_ABORT message will be resent until the transition condition is satisfied.
25. $404_{4C} \prec_{im} 405_{4C}$ and the transition occurs when the JOIN_OK message sent in state 504_{4A} is received by node C, indicating that $505_{4A} \rightsquigarrow 1_{4C}$.
26. $404_{4C} \prec_{im} 1_{4C}$ and the transition occurs when the JOIN_ABORT message sent in state 505_{4A} is received by node C, indicating that $505_{4A} \rightsquigarrow 1_{4C}$.
27. State 404_{4C} will continue to send the JOIN_PROCD message until the conditions in either (25) or (26) occur.
28. Assume the system state vector V_4 . By statements (3), (5), (6), (16), (17), (21), and (22), if the JOIN_ACCEPT message sent in state 502_{4A} is never received by node C, the system state will transition from V_4 to $V_5 \equiv \{ 503_{4A} \mid 0_{4B} \cdots \mid 1_{4C} \mid 0_{4D} \cdots \}$ when timer $T1_{4C}$ (which was set in state 401_{4C}) expires. Then, by statements (21), (22), and the fact that state 404_{4C} was not reached, the system state will transition from V_5 to $V_6 \equiv \{ 505_{4A} \mid 0_{4B} \cdots \mid 1_{4C} \mid 0_{4D} \cdots \}$ when timer $T1_{4A}$ (which was set in state 501_{4A}) expires. Then by statement (24), the system state will transition from V_6 to V_0 . Notice that FSM 4C (now in state 1_{4C}) does not need to react to the JN_ABORT message sent in 505_{4A} for the transition from V_6 to V_0 to occur. The channel only needs to acknowledge that node C has received the JN_ABORT message and to return the corresponding acknowledgement to node A. Timer $T1_{4A}$ exists in the join protocol so that node A has a way of exiting V_5 under the aforementioned conditions.
29. Assume the system state vector V_4 . By statements (3), (5), (6), (16), (17), (21), and (22), if the JOIN_ACCEPT message sent in state 502_{4A} is received by node C, the system state will transition from V_4 to $V_7 \equiv \{ 503_{4A} \mid 0_{4B} \cdots \mid 404_{4C} \mid 0_{4D} \cdots \}$. At this point (system state V_7), the initiator platoon has confirmation that the respondent platoon exists and is receptive to a join maneuver (i.e. the respondent platoon will not begin any other maneuvers on its own for the time being).

30. Assume the system state vector V_7 . By statements (3), (5), (6), (21), (22), and (27), if timer $T1_{4A}$ expires before the JOIN_PROCD message can be received by node A, the system state will transition from V_7 to $V_8 \equiv \{ 505_{4A} \mid 0_{4B} \cdots \mid 404_{4C} \mid 0_{4D} \cdots \}$ and will remain in V_8 until the JN_ABORT message is received by node C (the system state will transition from V_8 to V_6 by statement (26)), and its acknowledgement is returned to node A (at which time the system state will transition from V_6 to V_0 by statement (24)).
31. Assume the system state vector V_7 . By statements (3), (5), (6), (21), (22), (27), if the JOIN_PROCD message is received by node A before the timer $T1_{4A}$ expires, then the system state transitions from V_7 to $V_9 \equiv \{ 504_{4A} \mid 0_{4B} \cdots \mid 404_{4C} \mid 0_{4D} \cdots \}$. At this point (system state V_9), the respondent platoon has confirmed that:
- The initiator platoon knows that the respondent platoon exists, and
 - The initiator platoon is committed to the join maneuver (i.e. it cannot initiate any other maneuver without further input from the respondent platoon).
32. Statements (1) through (32) have proven that a successful first contact situation will result in a system state vector V_9 , and that all other situations result in the system state eventually returning to V_0 . For example, even if node C leaves state 1_{4C} before node A can return to state 1_{4A} in these abortive cases, node C can only proceed as far as state 403_{4C} since the proper responses from node A will not be forthcoming, and will return to state 1_{4C} when timer $T1_{4C}$ expires once again.
33. $405_{4C} \prec_{im} 420_{4C} \prec_{im} 422_{4C}$ and both the transitions are unconditional.
34. $422_{4C} \prec_{im} 423_{4C}$ and the transition occurs when the RECONF_B message sent in state 510_{4A} is received by node C.
35. Assume the system state vector V_9 . By statements (3), (5), (6), (23), (25), (33), and (34), if state 504_{4A} transitions to state 510_{4A} , this implies a system state vector $V_{10} \equiv \{ 510_{4A} \mid 0_{4B} \cdots \mid 422_{4C} \mid 0_{4D} \cdots \}$, and the system state can make no forward progress until the RECONF_A and RECONF_B messages are sent out by state 510_{4A} .
36. $510_{4A} \prec_{im} 512_{4A}$ and the transition occurs when the RECONF_A and RECONF_B messages sent in state 510_{4A} are sent successfully (i.e. all the acknowledgements are received by node A). FSM 4A is designed so that the messages will be resent until the transition condition is satisfied.
37. $512_{4A} \prec_{im} 513_{4A} \prec_{im} 700_{4A}$ and both the transitions are unconditional.
38. $600_{4B} \prec_{im} 601_{4B} \prec_{im} 602_{4B}$ and both the transitions are unconditional.
39. $602_{4B} \prec_{im} 0_{4B}$ and the transition occurs when the JOIN_FIN message sent in state 703_{4A} is received by node B, indicating that $703_{4A} \rightsquigarrow 0_{4B}$.
40. $422_{4C} \prec_{im} 423_{4C}$ and the transition occurs when the RECONF_B message sent in state 510_{4A} is received by node C, indicating that $510_{4A} \rightsquigarrow 423_{4C}$.
41. $423_{4C} \prec_{im} 710_{4C}$ and the transition is unconditional.
42. $710_{4C} \prec_{im} 711_{4C}$ and the transition occurs when the JOIN_PLAT message sent in state 700_{4A} is received by node C, indicating that $700_{4A} \rightsquigarrow 711_{4C}$.
43. $610_{4D} \prec_{im} 611_{4D} \prec_{im} 612_{4D} \prec_{im} 613_{4D}$ and all three transitions are unconditional.

44. $613_{4D} \prec_{im} 614_{4D}$ and the transition occurs when the JOIN_FIN message sent in state 703_{4A} is received by node D, indicating that $703_{4A} \rightsquigarrow 614_{4D}$.
45. $614_{4D} \prec_{im} 0_{4D}$ and the transition is unconditional.
46. $510_{4A} < 512_{4A} < 700_{4A} < 703_{4A}$.
47. Assume the system state vector V_{10} . By statements (3), (5), (36), (38), (39), (40), (41), (42), (43), (44), and (46), if state 510_{4A} transitions to state 512_{4A} , this indicates that the system state has transitioned from V_{10} to $V_{11.1} \equiv \{ 512_{4A} \mid 602_{4B} \cdots \mid 710_{4C} \mid 613_{4D} \cdots \}$, which by statement (37) will then unconditionally transition to $V_{11.2} \equiv \{ 700_{4A} \mid 602_{4B} \cdots \mid 710_{4C} \mid 613_{4D} \cdots \}$. The system state will not be able to make any further forward progress until the JOIN_PLAT message is sent out by state 700_{4A} .
48. At this point (system state $V_{11.2}$), the two independent platoon/LANs have been reconfigured into a single LAN, with addresses described by the now primary routing tables in each vehicle/node. However, the vehicles in this single LAN are still physically separated into two subplatoons.
49. $700_{4A} \prec_{im} 701_{4A}$ and the transition occurs when the JOIN_PLAT message sent in state 700_{4A} is sent successfully to node C (i.e. the acknowledgement for the JOIN_PLAT is received by node A). The JOIN_PLAT message will be resent by state 700_{4A} until the transition condition is satisfied.
50. $701_{4A} \prec_{im} 703_{4A}$ and the transition occurs when the JOINED message sent in state 713_{4C} is received by node A.
51. $711_{4C} \prec_{im} 713_{4C}$ and the transition occurs when the two subplatoons are physically joined.
52. Assume that the controller can always join the two subplatoons.
53. Assume the system state vector $V_{11.2}$. By statements (39), (42), (44), (46), and (49), if state 700_{4A} transitions to state 701_{4A} , this indicates that the system state has transitioned from $V_{11.2}$ to $V_{12} \equiv \{ 701_{4A} \mid 602_{4B} \cdots \mid 711_{4C} \mid 613_{4D} \cdots \}$ which, by statements (51) and (52), will unconditionally transition to $V_{13} \equiv \{ 701_{4A} \mid 602_{4B} \cdots \mid 713_{4C} \mid 613_{4D} \cdots \}$.
54. $713_{4C} \prec_{im} 0_{4C}$ and the transition occurs when the JOIN_FIN message sent in state 703_{4A} is received by node C, indicating that $703_{4A} \rightsquigarrow 0_{4C}$. The JOINED message is sent in state 713_{4C} , and will be resent until the transition condition is satisfied.
55. Assume the system state vector V_{13} . By statements (39), (44), (50), and (54), the system state vector will eventually transition to $V_{14} \equiv \{ 703_{4A} \mid 602_{4B} \cdots \mid 713_{4C} \mid 613_{4D} \cdots \}$. The system can make no further progress until the JOIN_FIN message is broadcast by state 703_{4A} . Notice that even though the join maneuver is essentially complete at this point, no further maneuvers can be authorized until node A enters state 1_{4A} . This satisfies the desired safety condition for the join protocol.
56. $703_{4A} \prec_{im} 1_{4A}$ and the transition occurs when node A successfully sends the JOIN_FIN message (which is sent in 703_{4A}) to nodes B, C, and D (i.e. all the acknowledgements are received by node A). The JOIN_FIN message is resent by state 703_{4A} until the transition condition is satisfied.

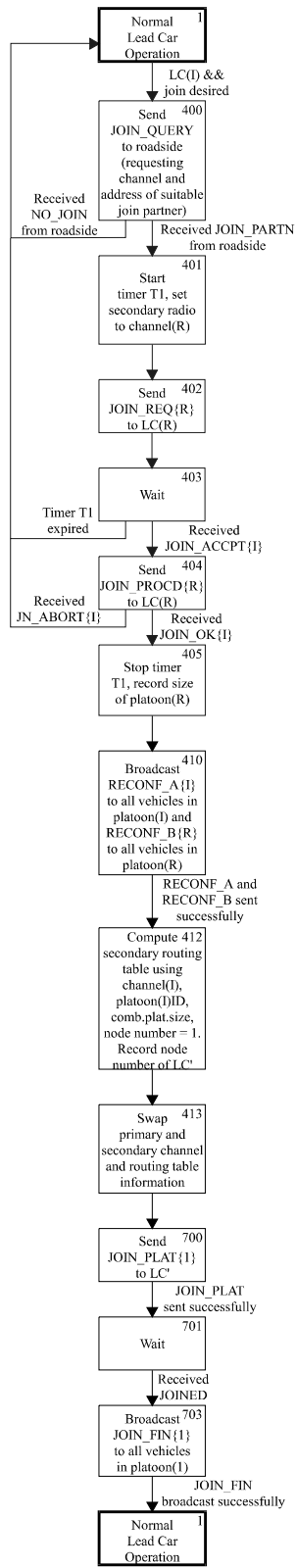
57. $614_{4D} \prec_{im} 0_{4D}$ and the transition is unconditional.
58. Assume the system state vector V_{14} . By statements (39), (44), (54), and (57), if state 703_{4A} transitions to state 1_{4A} , it indicates that the system state has transitioned from V_{14} to $V_{15} \equiv \{ 1_{4A} \mid 0_{4B} \ \dots \mid 0_{4C} \mid 0_{4D} \ \dots \}$.
59. Statements (1) through (58) have proven that once begun, the join maneuver will either terminate with a system state vector V_0 (in the case of an abortive first contact situation) or a system state vector V_{15} (in the case of a successful join). This satisfies the desired liveness condition for the join protocol.

For the join protocol, we were able to satisfy both the safety and liveness conditions. This was possible since the join maneuver ends up with a single lead vehicle/node, whereas the split maneuver ends up with two lead vehicle/nodes. It is much simpler for one node to control the actions of a group of vehicle/nodes than for two nodes to coordinate the actions of such a group.

11.3.2 Case 2: Initiator Platoon Leading Respondent Platoon

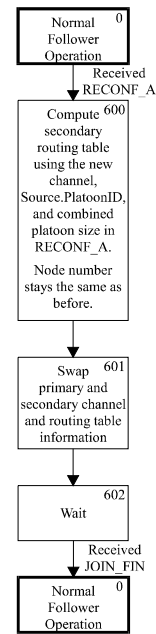
FSM 5A through FSM 5D describe the protocols for the four possible node types involved in the join maneuver. In this case, node A is the lead vehicle/node of the initiator platoon, and node C is the lead vehicle/node of the respondent platoon. The messages are as described in table 11.2.

Lead Vehicle of Respondent Platoon



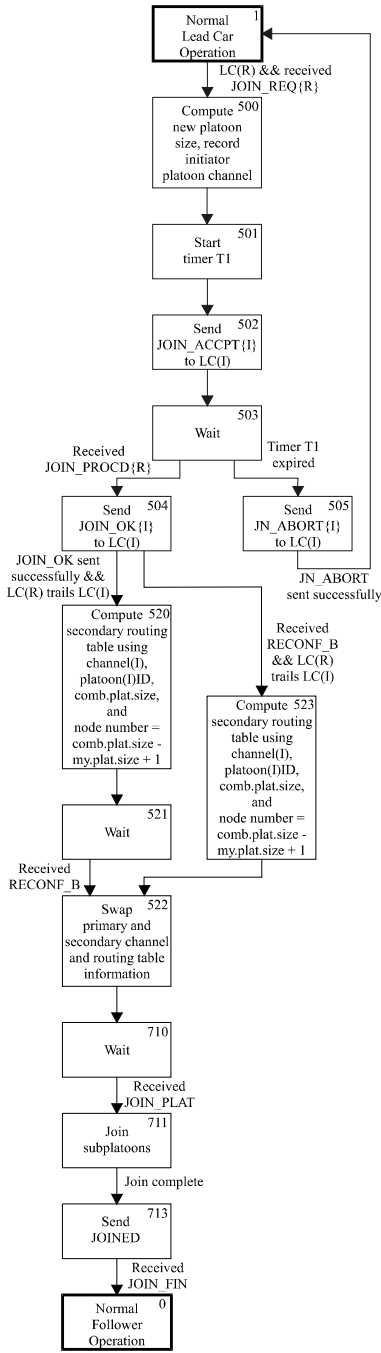
FSM 5A

Follower in Respondent Platoon



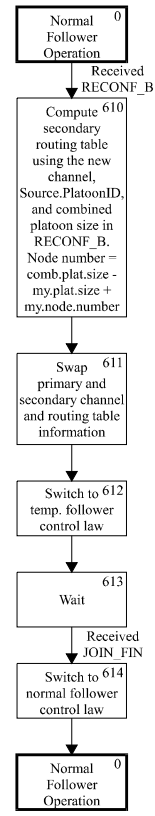
FSM 5B

Lead Vehicle of Initiator Platoon



FSM 5C

Follower in Initiator Platoon



FSM 5D

The proof given in statements (1) through (32) for case 1 of the join maneuver can also be applied to case 2. Except for the fact that node A has been swapped with node C, up to that point the states of the join protocol in case 2 are exactly the same as in case 1. We can therefore use the result that a successful first contact situation will result in the system state evolving from the initial state vector $V_0 \equiv \{ 1_{5A} \mid 0_{5B} \ \cdots \mid 1_{5C} \mid 0_{5D} \ \cdots \}$ to the system state vector $V_9 \equiv \{ 404_{5A} \mid 0_{5B} \ \cdots \mid 504_{5C} \mid 0_{5D} \ \cdots \}$, while all of the abortive first contact situations result in the system state returning to V_0 .

33. $404_{5A} \prec_{im} 1_{5A}$ and the transition occurs when the JN_ABORT message sent in state 505_{5C} is received by node A, indicating that $505_{5C} \rightsquigarrow 1_{5A}$.
34. $404_{5A} \prec_{im} 405_{5A}$ and the transition occurs when the JOIN_OK message sent in state 504_{5C} is received by node A, indicating that $504_{5C} \rightsquigarrow 405_{5A}$.
35. $405_{5A} \prec_{im} 410_{5A}$ and the transition is unconditional.
36. $0_{5B} \prec_{im} 600_{5B}$ and the transition occurs when the RECONF_A message sent in state 410_{5A} is received by node B.
37. $600_{5B} \prec_{im} 601_{5B} \prec_{im} 602_{5B}$ and both of the transitions are unconditional.
38. $602_{5B} \prec_{im} 0_{5B}$ and the transition occurs when the JOIN_FIN message sent in state 703_{5A} is received by node B, indicating that $703_{5A} \rightsquigarrow 0_{5B}$.
39. $504_{5C} \prec_{im} 520_{5C}$ and the transition occurs when the JOIN_OK message sent in state 504_{5C} is sent successfully (i.e. the corresponding acknowledgement from node A is received by node C).
40. $504_{5C} \prec_{im} 523_{5C}$ and the transition occurs when the RECONF_B message sent in state 410_{5A} is received by node C, indicating that $410_{5A} \rightsquigarrow 523_{5C}$.
41. $520_{5C} \prec_{im} 521_{5C}$ and the transition is unconditional.
42. $523_{5C} \prec_{im} 522_{5C}$ and the transition is unconditional.
43. $521_{5C} \prec_{im} 522_{5C}$ and the transition occurs when the RECONF_B message sent in state 410_{5A} is received by node C, indicating that $410_{5A} \rightsquigarrow 522_{5C}$.
44. $522_{5C} \prec_{im} 710_{5C}$ and the transition is unconditional.
45. $710_{5C} \prec_{im} 711_{5C}$ and the transition occurs when the JOIN_PLAT message sent in state 700_{5A} is received by node C, indicating that $700_{5A} \rightsquigarrow 711_{5C}$.
46. $0_{5D} \prec_{im} 610_{5D}$ and the transition occurs when the RECONF_B message sent in state 410_{5A} is received by node D, indicating that $410_{5A} \rightsquigarrow 610_{5D}$.
47. $610_{5D} \prec_{im} 611_{5D} \prec_{im} 612_{5D} \prec_{im} 613_{5D}$ and all three of the transitions are unconditional.
48. $613_{5D} \prec_{im} 614_{5D}$ and the transition occurs when the JOIN_FIN message sent in state 703_{5A} is received by node D, indicating that $703_{5A} \rightsquigarrow 614_{5D}$.
49. $404_{5A} < 410_{5A} < 700_{5A} < 703_{5A}$.

50. Assume the system state vector V_9 . By statements (36), (39), (40), (46), and (49) state 504_{5C} will continue to resend the JOIN_OK message until one of two events occur:
- The JOIN_OK message is received by node A, but the corresponding acknowledgement never reaches node C, in which case the system state will transition from V_9 to $V_{10} \equiv \{ 410_{5A} \mid 0_{5B} \cdots \mid 504_{5C} \mid 0_{5D} \cdots \}$ by statements (34) and (35), or
 - The JOIN_OK message is received by node A and the corresponding acknowledgement is received by node C, in which case the system state will transition from V_9 to $V_{11} \equiv \{ 410_{5A} \mid 0_{5B} \cdots \mid 521_{5C} \mid 0_{5D} \cdots \}$ by statements (34), (35), and (39).
51. $410_{5A} \prec_{im} 412_{5A}$ and the transition occurs when the RECONF_A and RECONF_B messages sent in state 410_{5A} are sent successfully (i.e. node A receives all the acknowledgement messages to the RECONF_A and RECONF_B messages). Node A will remain in 410_{5A} (rebroadcasting the RECONF_A and RECONF_B messages) until the transition condition is satisfied.
52. $412_{5A} \prec_{im} 413_{5A} \prec_{im} 700_{5A}$ and both of the transitions are unconditional.
53. Assume the system state vector V_{10} . By statements (36), (37), (38), (40), (42), (44), (46), (47), (48), (49), and (51), if state 410_{5A} transitions to state 412_{5A} , it indicates that the system state has transitioned from V_{10} to $V_{12} \equiv \{ 412_{5A} \mid 602_{5B} \cdots \mid 710_{5C} \mid 613_{5D} \cdots \}$, which by statement (52) will then unconditionally transition to $V_{13} \equiv \{ 700_{5A} \mid 602_{5B} \cdots \mid 710_{5C} \mid 613_{5D} \cdots \}$.
54. Assume the system state vector V_{11} . By statements (36), (37), (38), (43) through (49), and (51), if state 410_{5A} transitions to state 412_{5A} , it indicates that the system state has transitioned from V_{11} to V_{12} , which by statement (52) will then unconditionally transition to V_{13} .
55. Notice that both possible outcomes given in statement (50) will evolve into the system state vector V_{13} .
56. By statements (38), (45), (48), and (49), the system state vector V_{13} can make to forward progress until the JOIN_PLAT message is sent out by state 700_{5A} . Note that at this point, the two platoon/LANs have been reconfigured to be a single platoon/LAN with addresses as described by the now primary routing tables. However, the vehicles in the platoon are still grouped into two subplatoons.
57. $700_{5A} \prec_{im} 701_{5A}$ and the transition occurs when the JOIN_PLAT message sent in state 700_{5A} is sent successfully to node C (i.e. node A receives the corresponding acknowledgement from node C). Node A will remain in state 700_{5A} , and resending the JOIN_PLAT message until the transition condition is satisfied.
58. $711_{5C} \prec_{im} 713_{5C}$ and the transition occurs when the two subplatoons are physically joined.
59. Assume that the controller is always able to join the subplatoons.
60. $713_{5C} \prec_{im} 0_{5C}$ and the transition occurs when the JOIN_FIN message sent in state 703_{5A} is received by node C. The JOINED message is sent in state 713_{5C} , and will continue to be resent by node C until the transition condition is satisfied.

61. Assume the system state vector V_{13} . By statements (38), (45), (48), (49), (57), and (58), if state 700_{5A} transitions to state 701_{5A} , it indicates that the system state has transitioned from V_{13} to $V_{14} \equiv \{ 701_{5A} \mid 602_{5B} \cdots \mid 711_{5C} \mid 613_{5D} \cdots \}$, which by assumption (59) will then transition into the state vector $V_{15} \equiv \{ 701_{5A} \mid 602_{5B} \cdots \mid 713_{5C} \mid 613_{5D} \cdots \}$.
62. $701_{5A} \prec_{im} 703_{5A}$ and the transition occurs when the JOINED message sent in state 713_{5C} is received by node A, indicating that $713_{5C} \rightsquigarrow 703_{5A}$.
63. $703_{5A} \prec_{im} 1_{5A}$ and the transition occurs when the JOIN_FIN message sent in state 703_{5A} is successfully broadcast to all vehicles in the combined platoon (i.e. node A receives the corresponding acknowledgements from all vehicles in the combined platoon).
64. Assume the system state vector V_{15} . By statement (60), V_{15} will eventually transition to $V_{16} \equiv \{ 703_{5A} \mid 602_{5B} \cdots \mid 713_{5C} \mid 613_{5D} \cdots \}$. Notice that V_{16} can make no further progress until the JOIN_FIN message is sent by state 703_{5A} . At this point (system state V_{16}) the maneuver is essentially complete, but no other maneuver can be authorized until node A returns to its normal leader state 1_{5A} . This satisfies the desired safety condition for the join protocol.
65. $614_{5D} \prec_{im} 0_{5D}$ and the transition is unconditional.
66. Assume the system state vector V_{16} . By statements (38), (48), (60), and (65), if state 703_{5A} transitions to state 1_{5A} , it indicates that the system state has transitioned from V_{16} to $V_{17} \equiv \{ 1_{5A} \mid 0_{5B} \cdots \mid 0_{5C} \mid 0_{5D} \cdots \}$.
67. Statements (33) through (66) have proven that once begun, the join maneuver will either terminate with a system state vector V_0 (in the case of an abortive first contact situation) or a system state vector V_{17} (in the case of a successful join). This satisfies the desired liveness condition for the join protocol.

11.4 Combined Finite State Machine for the Platoon Join Protocol

Figure 11.2 shows the finite state machine that results when FSM 4A through FSM 4D and FSM 5A through FSM 5D are combined. Implementing the finite state machine in figure 11.2 in each vehicle/node allows each vehicle/node the ability to act as any one of the four node types involved in the join maneuver. It also enables the lead vehicle/nodes to handle each of the two cases of the join maneuver.

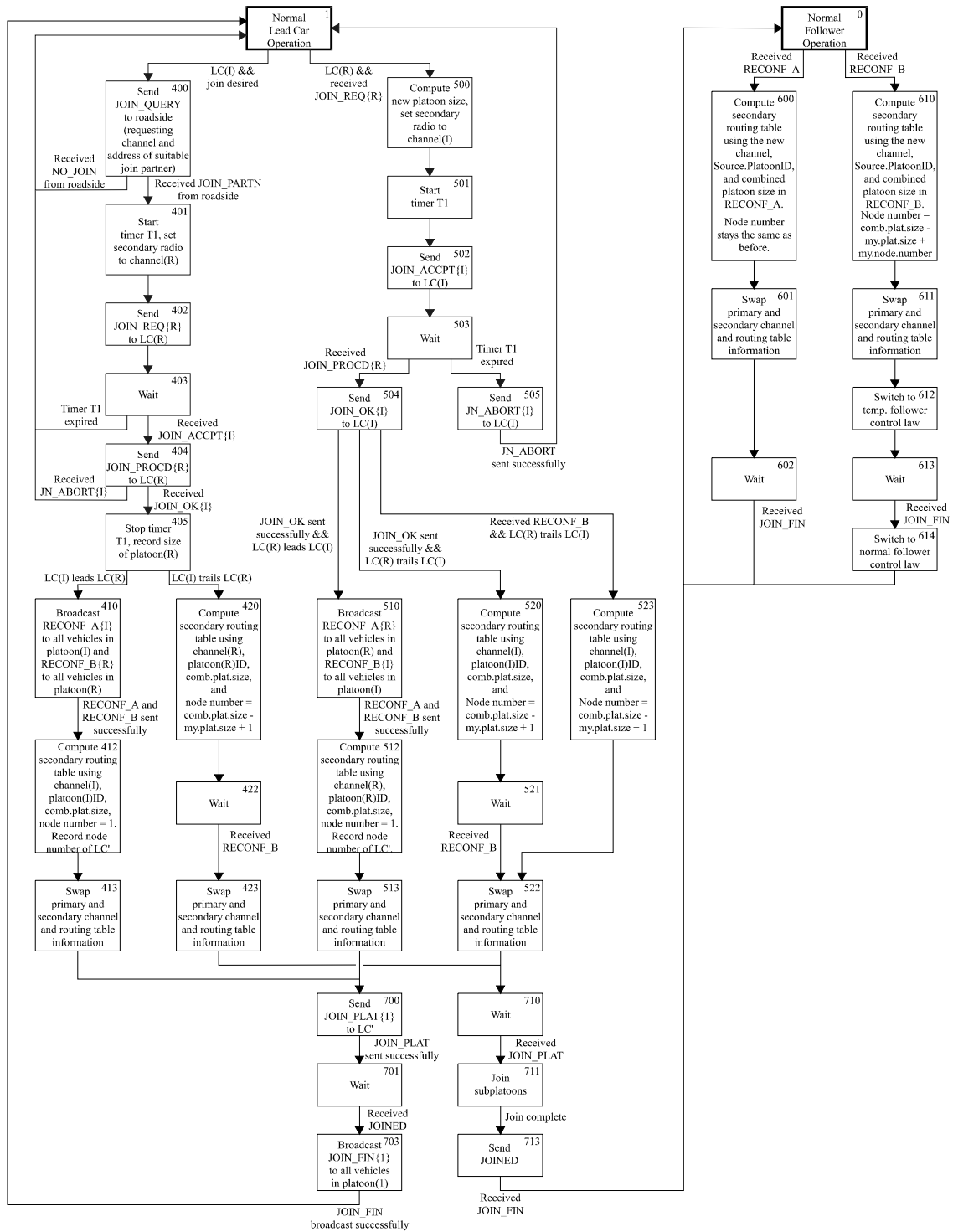


Figure 11.2: Platoon Join Protocol

Chapter 12

Field Tests

12.1 Implementation on Test Vehicles

The test vehicles consist of Lincoln Towncars equipped with radar, throttle and brake actuators, engine and wheelspeed sensors, a wireless radio, and a personal computer.

12.1.1 Controller

The hybrid controller for a lead/follower vehicle developed by Chen (2000) is used to control vehicle spacing in the test vehicles.

12.1.2 Communication System

The radios on the test vehicles are WaveLAN radios manufactured by Lucent Technologies. These radios are hardwired to transmit and receive on a single channel, so two independently operating platoons cannot be fully implemented. However, the ability of the join and split protocols to complete their tasks as well as the performance of the control laws can still be evaluated. To accomplish this, we simply limit one of the “platoons” involved in a join or split maneuver to be a free agent (i.e. a platoon of one vehicle). Since a single vehicle does not need to transmit control information to itself, a single channel radio system is sufficient for these tests.

Each vehicle/node is further restricted to one radio per vehicle, while the protocols call for two radios per vehicle. As a result, the finite state machines in figures 11.1 and 11.2 cannot be implemented. However, the software interface can be written to accept any finite state machine, and then as a proof of functionality, the finite state machines in FSM 1A through FSM 1D for the

split maneuver and the finite state machines in FSM 2A through FSM 2D for the join maneuver are actually coded implemented on the test vehicles. Even though the join protocol given in FSM 2A through FSM 2D did not satisfy the safety and liveness conditions given in chapter 10, it functions well enough when implemented in the test vehicles to verify the functionality of the rest of the software, and to act as a “stand-in” finite state machine when testing the join control laws.

12.1.3 Maneuver Protocols

The communication network and link layer software has not yet been developed and implemented on the Lincoln Towncars. As a result, the protocols for the platoon split and join maneuvers have been modified slightly to explicitly send the acknowledgements in reply to messages received. These modified finite state machines are shown in figures 12.1A, 12.1B, 12.2A, 12.2B, 12.2C and 12.2D. Notice that these are not the dual-channel radio versions of the join and split protocols, but these finite state machines are implemented to prove that the entire system can function as a whole (given that we are limited to a single radio per vehicle/node). The dual-channel protocols were not used because each test vehicle is equipped with only a single channel radio, and there was no way to test the functionality of a dual-channel protocol.

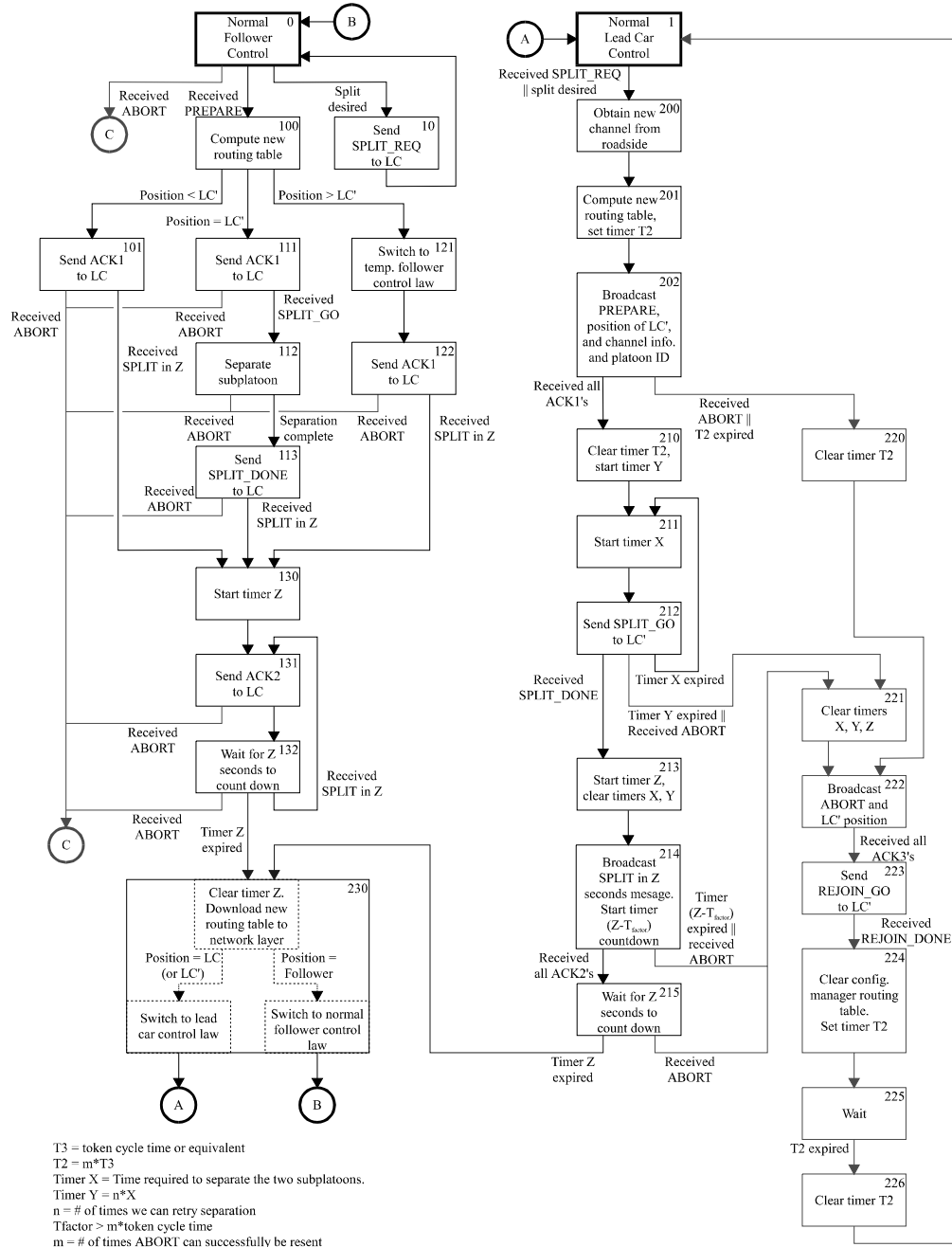


Figure 12.1A: Implemented Split Protocol

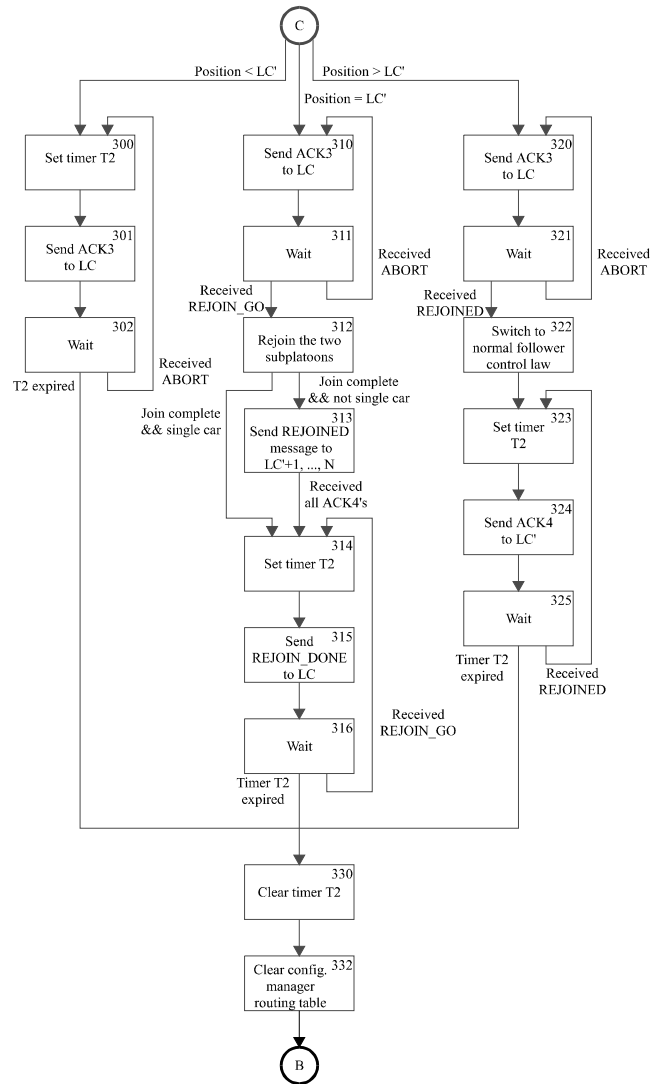


Figure 12.1B: Implemented Split Protocol (continued)

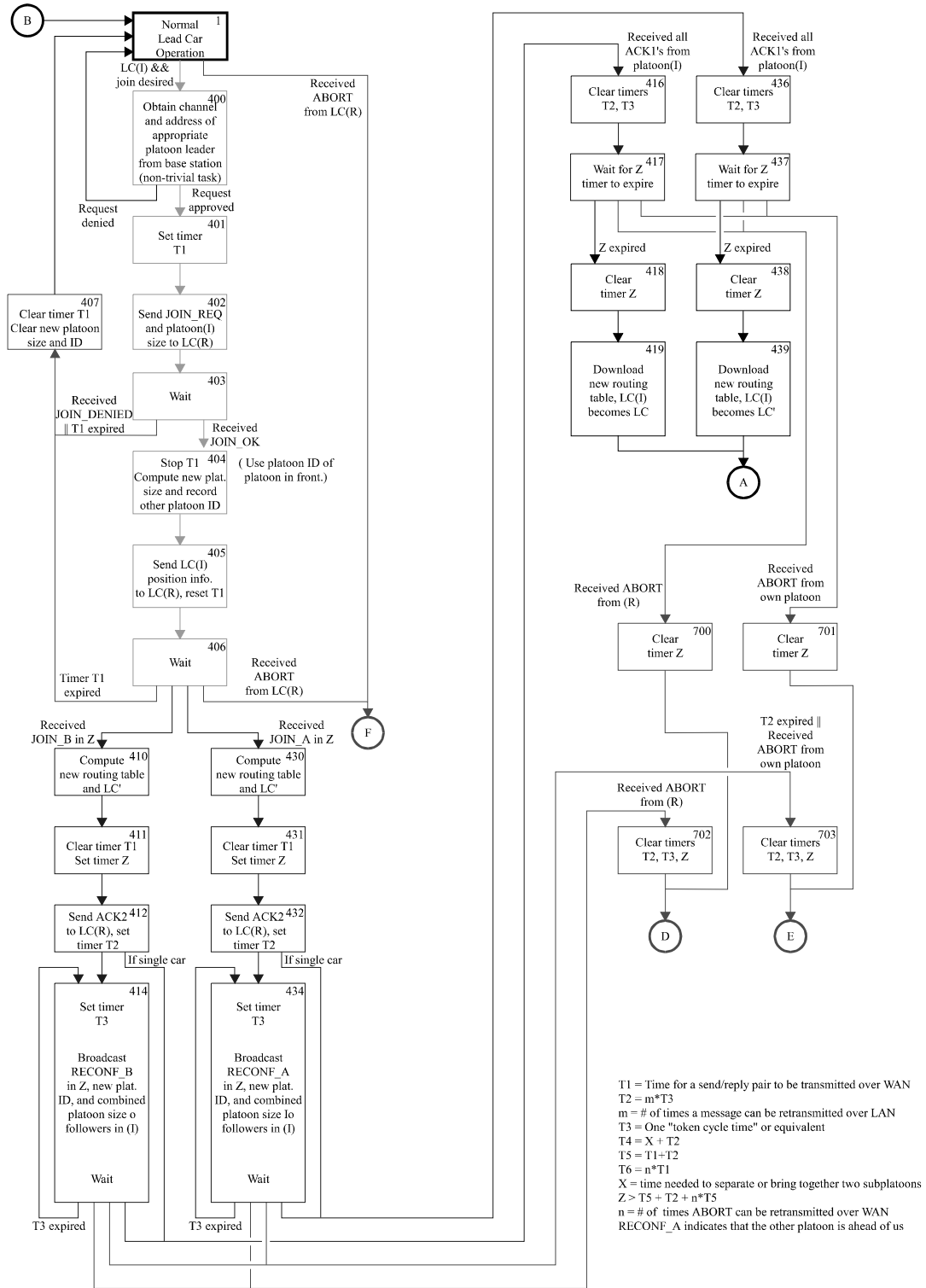


Figure 12.2A: Implemented Join Protocol

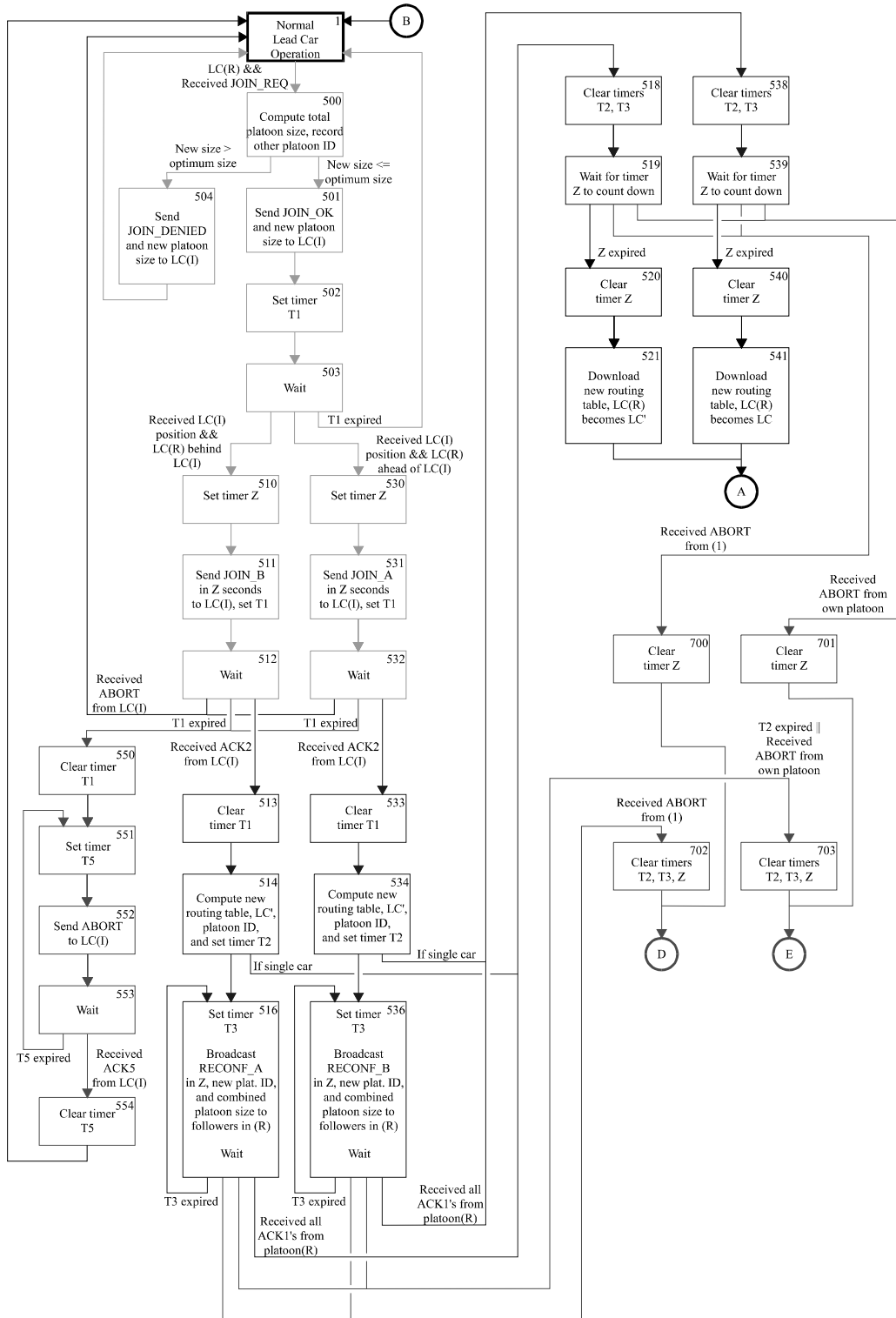


Figure 12.2B: Implemented Join Protocol (continued)

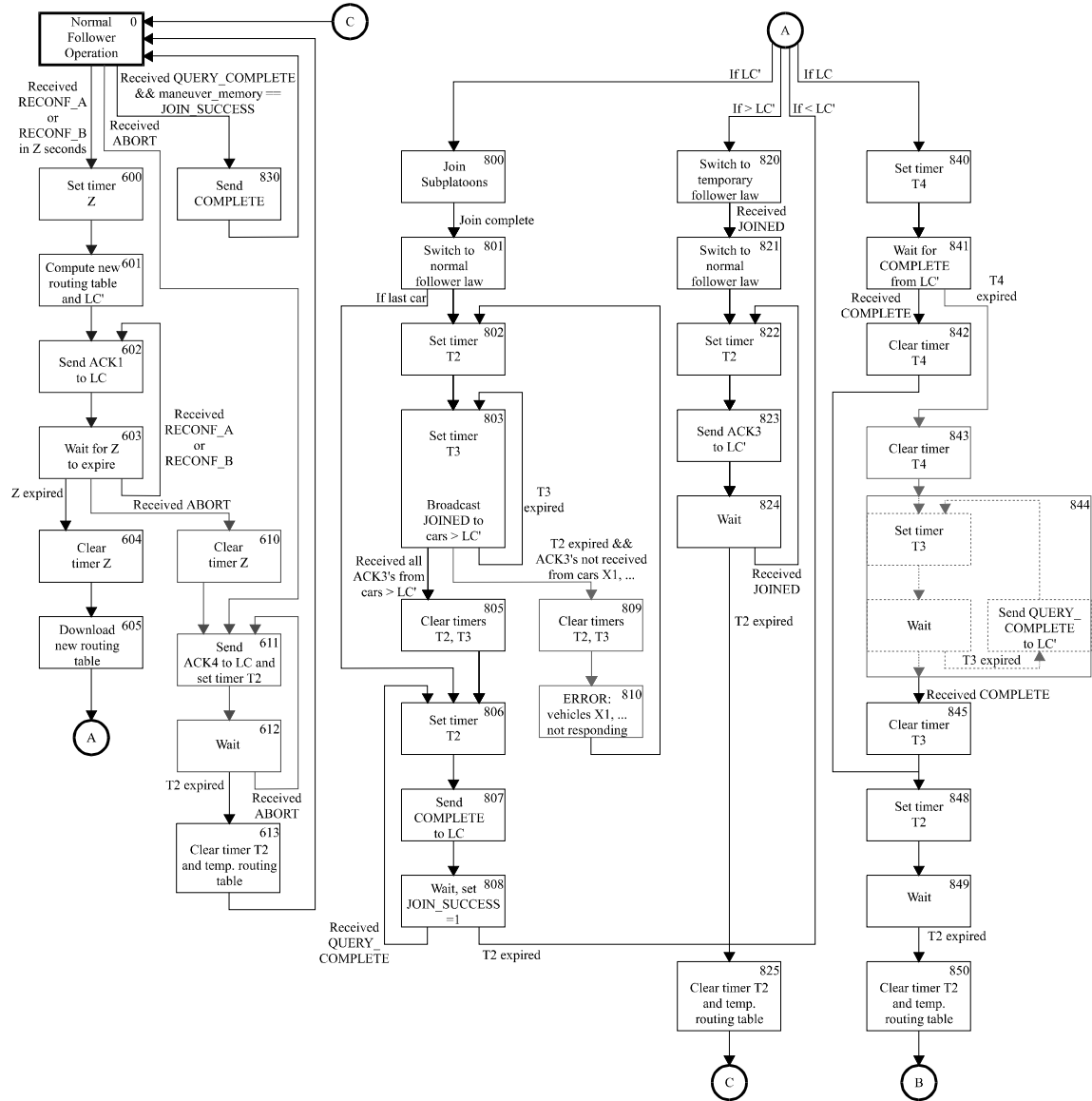


Figure 12.2C: Implemented Join Protocol (continued)

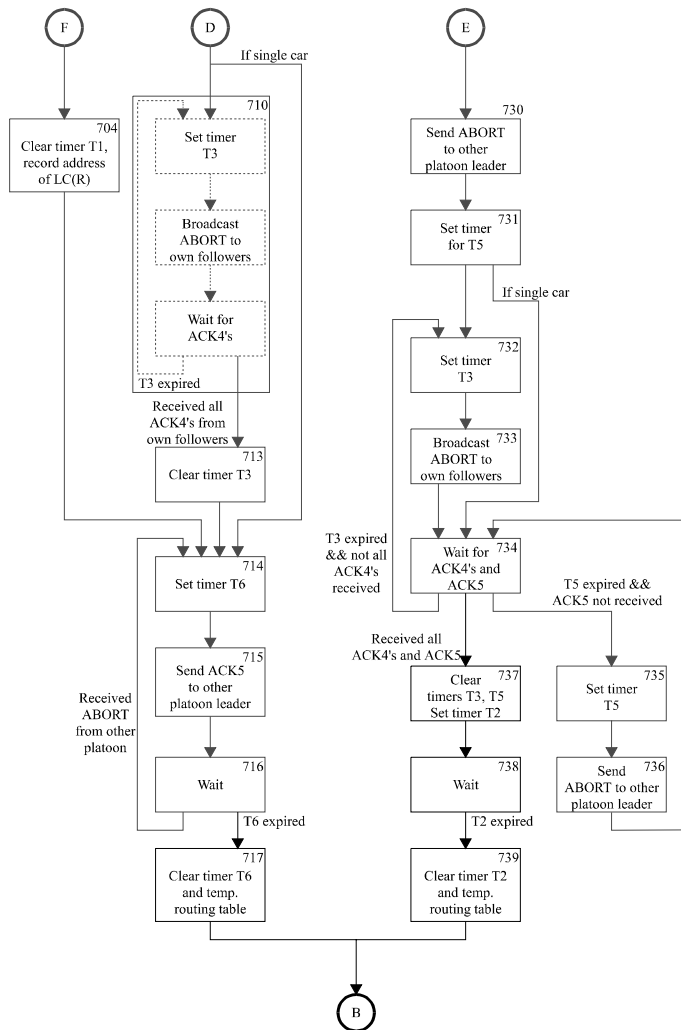


Figure 12.2D: Implemented Join Protocol (continued)

12.2 Maneuver Protocol Tests

The maneuver protocol tests will consist of three maneuvers: a platoon split; a platoon join with the initiator platoon being in front; and a platoon join with the initiator platoon being in the rear. These tests will verify that the finite state machines presented in figures 12.1A, 12.1B, 12.2A, 12.2B, 12.2C, and 12.2D are correctly coded onto the vehicle. Because of the lack of fully functional test vehicles, these tests will be performed on stationary vehicles. Since the only portion of the split and join protocols that depend on vehicle motion are the states in which the platoons use their transition control modes, the completion of the transition control mode is “artificially” provided in the controller software module. A log file is created which traces the state evolution and control mode used on each vehicle. This log file is then used to debug any behavior in the test vehicles that deviates from the finite state machines in figures 12.1A, 12.1B, 12.2A, 12.2B, 12.2C, and 12.2D. Using this technique, state traces were generated for:

- A two-car platoon splitting into two free agents.
- Two free agents joining into a two-car platoon.
- A three-car platoon with the lead vehicle splitting off to become a free agent.
- A three-car platoon with the last vehicle splitting off to become a free agent.
- A (leading, initiator) two-car platoon and a (trailing, respondent) free agent joining to become a three-car platoon.
- A (leading, respondent) two-car platoon and a (trailing, initiator) free agent joining to become a three-car platoon.
- A (leading, initiator) free agent and a (trailing, respondent) two-car platoon joining to become a three-car platoon.
- A (leading, respondent) free agent and a (trailing, initiator) two-car platoon joining to become a three-car platoon.

These state traces were checked against the finite state machines illustrated in figures 12.1 and 12.2, and were found to be a faithful implementation of these finite state machines. The control modes that were commanded by these finite state machines were also checked and found to be correct.

12.3 Controller Tests

Once the split and join protocols are correctly implemented in the vehicles, and the interface between the control software and the finite state machines was checked, the controllers themselves were tested. A “phantom” vehicle was coded into the software to provide something for the test vehicle to follow. This phantom vehicle is an entity that exists only in software, and simulates a vehicle proceeding at a constant 20 miles per hour. A simulated radar signal is computed based on the test vehicle’s current and past velocity, and inserted in the place of actual radar data. The single test vehicle could be made to behave as a follower in this phantom vehicle’s platoon, or it could be made to join with this phantom vehicle. Since this phantom vehicle does not really exist, and therefore cannot respond to the handshaking messages generated by the split and join protocols, the protocols on the test vehicle were modified to “fool” them into thinking that the required response messages had been received.

12.3.1 Lead Vehicle Controller Tests

Figures 12.3 through 12.5 are plots of the desired and measured velocity, control state, spacing, and control surface data taken from field tests of the hybrid lead-car controller. The control state plot shows the time evolution of the states in the hybrid control law, each state being assigned an integer value. The spacing plot shows the distance between the preceding “phantom” vehicle and the test vehicle. The control surface tracking plot indicates how well the controller in the “active” state in the hybrid control law is tracking its particular control surface. In each of these test runs, the velocity of the preceding vehicle is 8 m/s. The parameter that varies between figures 12.3 through 12.5 is the value of the ideal velocity. The control parameters used are: $c_0=1.0$, $c_1=1.5$, $c_2=0.5$, and $\eta_{lead}=2.0$. For the control state plot, the value of the control state indicates the discrete state of the hybrid control law (see Chen, 2000 -- figure 5.13):

<u>Control State Value</u>	<u>Lead Controller State</u>	<u>Description</u>
7	L0	Out of Range
8	L1	In Range
9	L3	Too close and closing
10	L2	Too close and separating

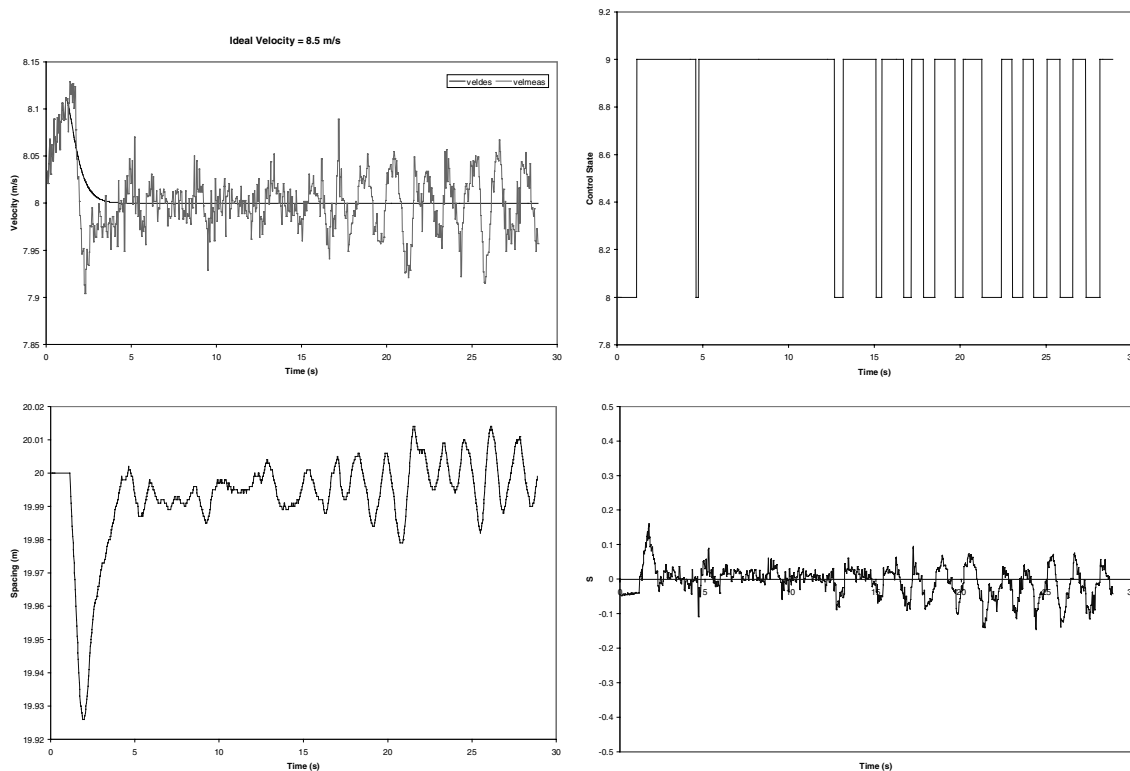


Figure 12.3: Lead Car Control Test Data ($v_{prev} < v_{ideal}$)

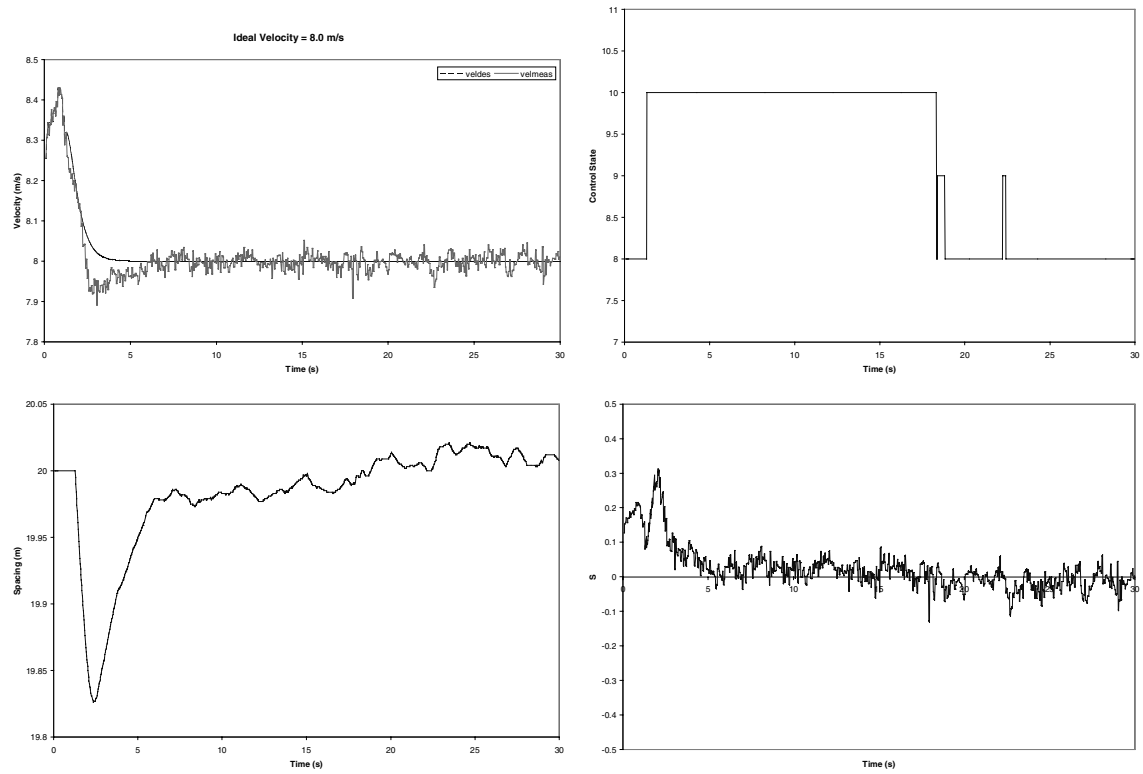


Figure 12.4: Lead Car Control Test Data ($v_{prev} = v_{ideal}$)

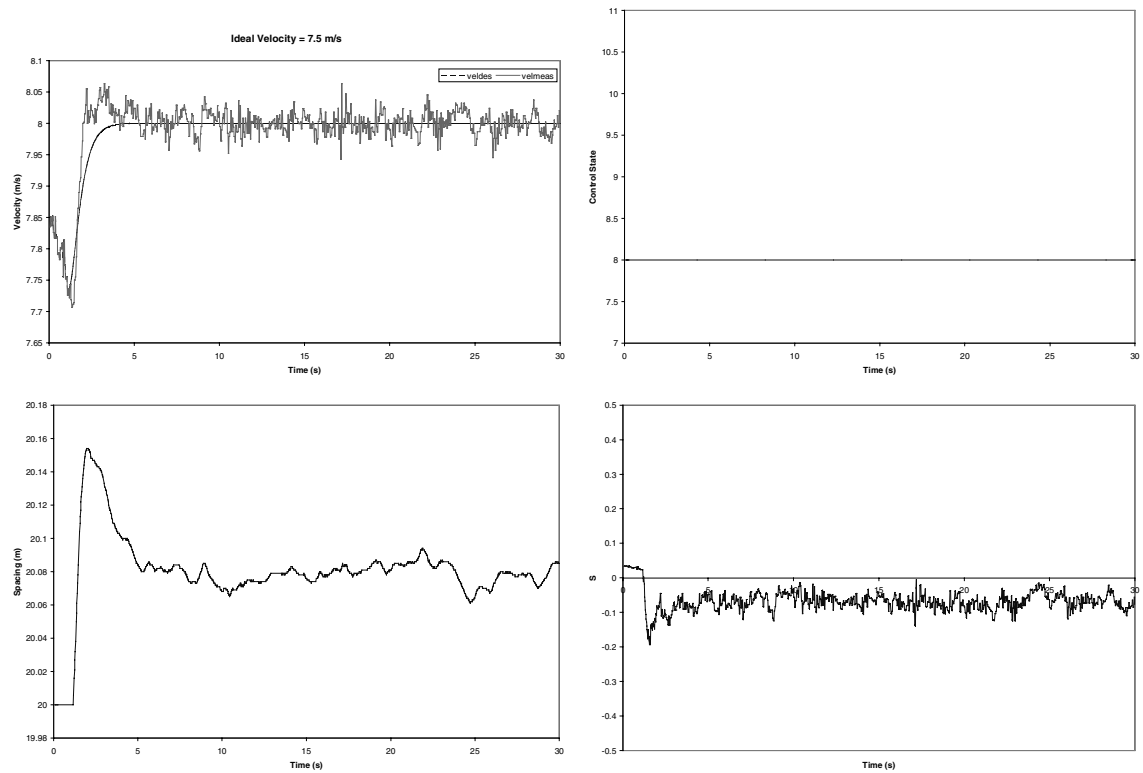


Figure 12.5: Lead Car Control Test Data ($v_{prev} > v_{ideal}$)

In all of these tests, the lead vehicle tracks the desired spacing and desired velocity relatively well. Figure 12.3 shows some oscillatory behavior that results when the ideal velocity and the previous vehicle's velocity are very close together and $v_{prev} < v_{ideal}$. In attempting to track both v_{prev} and v_{ideal} , the lead car controller enters state L_3 , where the two vehicles are too close and moving closer together. This results in a decrease in control effort in an attempt to slow the car down, thus moving the lead car controller back into state L_1 , where the process repeats itself. When the nonzero value of the constant c_2 was increased to 0.5, this oscillation became much more noticeable to the passengers. When the nonzero value of c_2 was set to 0.1, this oscillation was barely discernable. While this is a cyclic behavior, the plots in figure 12.3 show that it is not an unstable cyclic behavior, as the velocity, spacing, and control surface stay close to their desired values. This cyclic behavior does not show up in simulation, probably due to the unmodeled dynamics or uncertainties present in the actual test vehicle.

12.3.2 Split Transition Controller Test

The split transition test was performed by initially setting up the test vehicle as the second car in a two car platoon (i.e. a follower). The values in the control state plot are interpreted as follows.

<u>Control State</u>	<u>State as given in Chen, 2000--Figure 5.13</u>
1	J_0
2	J_1
3	J_2
4	F_0
5	S_0
6	S_1
7	L_0
8	L_1
9	L_3
10	L_2

Figure 12.6 shows the desired and measured velocity, control state, spacing, and control surface data for the platoon split test. The initial inter-platoon spacing was set to 20 meters and the intra-platoon spacing was set to 8 meters. Both the velocity of the phantom lead vehicle and the ideal velocity were set at 8 m/s. As can be seen in figure 12.6, the controllers generate the correct desired velocity trajectories in order to move the two vehicles from an intra-platoon spacing to an inter-platoon spacing. Looking at figure 12.6, it can be seen that there is a "spike" where the measured velocity deviates from the desired velocity trajectory, and likewise where the control surface deviates from zero. The field tests performed by Connolly (1996) did not show such a large deviation. However the tests performed by Connolly were done at velocities around 20 m/s, while the test shown in figure 12.6 were done at velocities around 8 m/s. The control law used in the J_1 state (see Chen, 2000) of the hybrid controller is the same as the one used by Connolly. The deviation in velocity tracking in figure 12.6 can be seen to start when the vehicle has slowed to around 6 m/s, this undesired behavior can be attributed to the limitations of the wheel speed sensor, which, being a discrete encoder, is known not to work very well at low vehicle speeds.

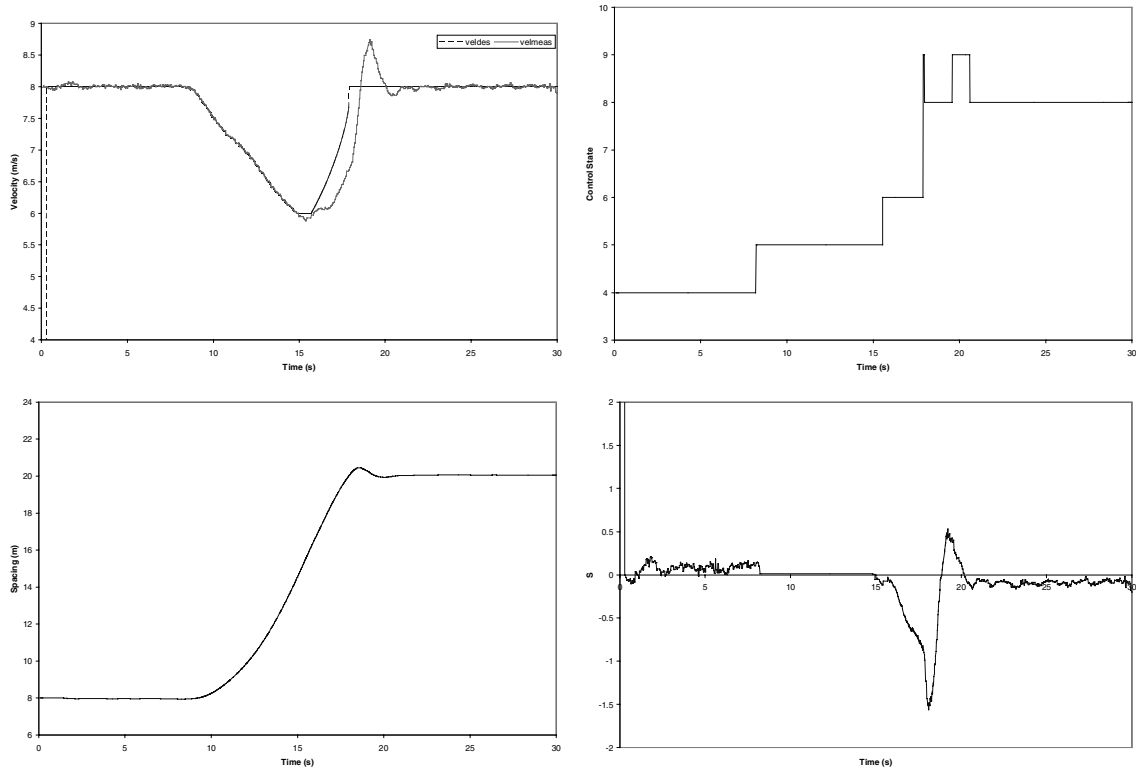


Figure 12.6: Split Transition Data

12.3.3 Join Transition Controller Test

The join transition test was performed by initially setting up the test vehicle as an independent agent and having it join with a phantom vehicle which is also an independent agent. The control state values are interpreted the same way as for the split transition test. The initial inter-platoon spacing was set at 20 meters, and the intra-platoon spacing was set to 8 meters. Both the velocity of the phantom vehicle and the ideal velocity were set to 8 m/s. Figure 12.7 shows the desired and measured velocity, control state, spacing, and control surface data that was collected from this test. As can be seen from figure 12.7, the join transition controllers generate the correct velocity trajectories needed to bring the vehicle from an inter-platoon spacing to an intra-platoon spacing. The controller also manages to track the desired velocity fairly well. The spike in the control surface tracking plot in figure 12.7 is due to the discrete switching between desired control surfaces and is actually very small in magnitude.

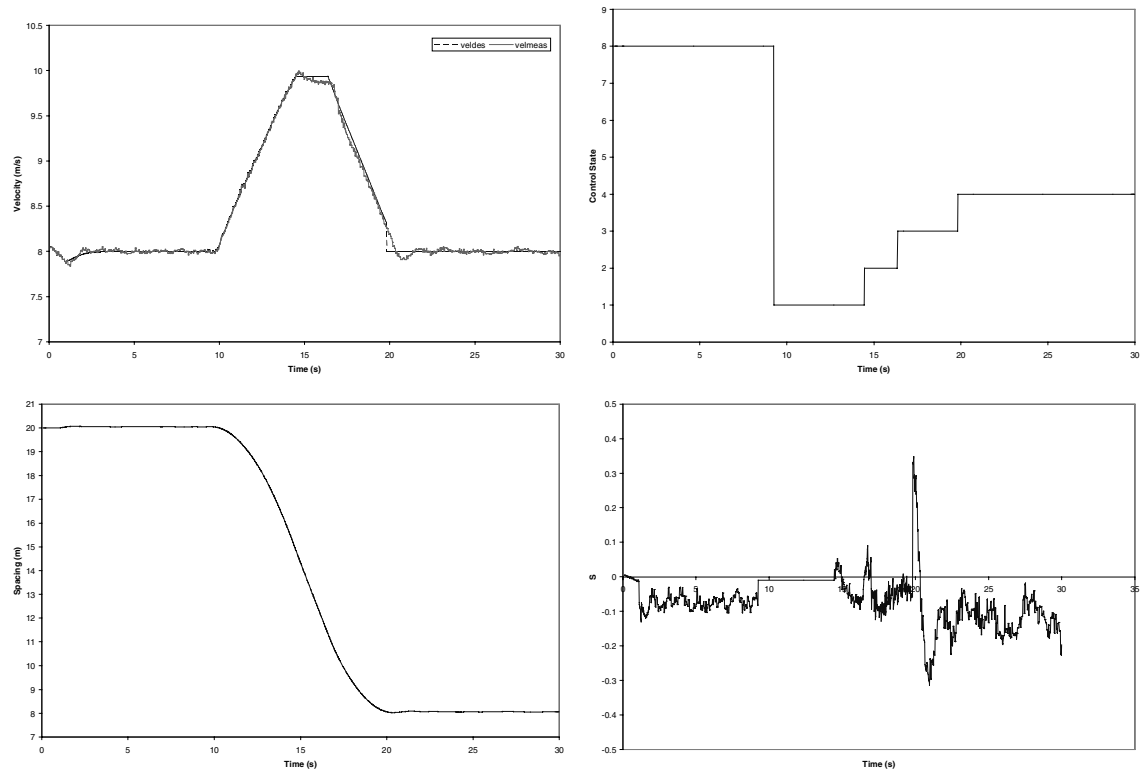


Figure 12.7: Joint Transition Data

Chapter 13

Summary of Conclusions, and Future Work

13.1 Communication Delays

Chapters 6 and 7 in this report explored the effects of communication delays on string stability. It was concluded that any delay in the communicated information required by the existing control algorithms will not be string stable over all conditions. In fact, certain maneuvers will result in string instability.

There will always be delays in communication systems. This report only reviewed the types of communication delays that can occur, and developed a method to check whether or not the system is string stable for the existing control algorithm and gains. The range of maneuvers that cause the string to be unstable have not been fully investigated in this report.

One of the important issues that needs to be addressed is the conditions under which string instability occurs, and determine whether these scenarios will be encountered in the AHS environment. Under real operating conditions, the spacing errors may only grow by a few centimeters, which may be acceptable. In the case that these errors are not acceptable, additional research would need to be done to address this problem.

One avenue of attack would be the development of control algorithms that are robust to communication delays. These control algorithms might be switched with different control algorithms during certain maneuvers. It may be possible that gain scheduling alone might be sufficient. The next logical steps are to address these issues.

Another area to study is the robustness of the control algorithms towards random packet losses, and the conditions under which the control logic should declare an abort condition. Currently the system counts the number of packets lost and declares a communication fault when X number of packets have been lost consecutively and proceeds to an abort condition. It seems that this number should vary depending on the vehicle's acceleration. For example, if a platoon is moving at constant speed, packet losses would not be a problem since the same acceleration and velocity data would be sent. By studying the effects of packet losses on string stability, premature declaration of communication failures can be eliminated.

13.2 Maneuver Protocols

In order to facilitate communications between platoons, a method of addressing a vehicle/node in a platoon/LAN was developed in chapters 5.2 and 8.2 which gave a vehicle enough information to infer its relation to surrounding vehicles. Preliminary arbitration algorithms for split and join maneuvers requests were discussed in chapters 9.2 and 10.2, respectively.

The purpose of the maneuver protocols developed in chapters 9, 10, and 11 is to provide the handshaking necessary to ensure that each vehicle/node correctly changes its address to reflect changes in its platoon configuration. The maneuver protocols also signals the combined hybrid controller to take certain state transitions. These maneuver protocols needed to satisfy a safety condition (another maneuver could not be started before the current one was complete), and a liveness condition (once started, a maneuver will move towards completion). The protocols were proven to satisfy these two conditions through the use of "cause-and-effect" logic for distributed systems. The general procedure involves finding system state vectors (or "slices") at which point the system cannot make any further forward progress unless a certain condition is met. Designing a protocol with such "stages" in mind allows us to structure the progress of a maneuver (whether it be a platoon split or a platoon join).

A major result of this dissertation came about when developing the protocols to be robust towards packet losses. It was discovered that in order to satisfy the safety conditions, each vehicle needed to be able to transmit and receive messages on two channels simultaneously. With this "dual channel" capability, the design of the platoon join and split protocols was greatly simplified over the case where each vehicle had only "single channel" capability. Although this is not the first dissertation to develop maneuver protocols that are robust towards packet losses, it is the first to develop such a protocol that interacts with the vehicle regulation layer (i.e. the vehicle control laws).

Because the test vehicles were limited to single-channel capability, the dual-channel maneuver protocols could not be implemented for testing. However, the software framework needed to implement any finite state machine based protocol was implemented and tested using the single-channel versions of the platoon join and platoon split protocols given in figures 12.1A, 12.1B, 12.2A, 12.2B, 12.2C, and 12.2D. The interface between the maneuver protocols and the combined hybrid controller was also implemented and tested using the single-channel versions of the protocols. This software has demonstrated its functionality in both the static tests that were performed for the maneuver protocols, and also in the dynamic tests that were done for the

vehicle control. In both static and dynamic tests, the correct state evolution was verified for the implemented finite state machines through the use of log files and the address of the vehicle/node was correctly recorded both before and after the maneuvers.

This report goes a long way in putting together the necessary pieces to form a fully operational automated vehicle. However, there is much more to be done. Future work would include implementing and testing the dual-channel versions of the platoon join and platoon split protocols developed in chapter 11. The navigator process that was mentioned in chapter 5.1 was largely ignored in this dissertation, but is a necessary piece of a completed automated vehicle that needs to be developed. There are also limitations to the controller and maneuver protocols developed in this dissertation. For example, the controller does not involve any sort of emergency behaviors, and the maneuver protocols assume that the wireless communication system does not suffer from any malfunctions. Dealing with these emergency situations is another factor that must be considered before a “customer-ready” automated vehicle can be fully realized.

Bibliography

- [1] Yu-Han Chen; “Effect of Communication Delays on the Performance of Vehicle Platoons”; M.S. Thesis; University of California, Berkeley; May 1993.
- [2] Yu-Han Chen; “Control and Networking Aspects in the Design of Longitudinal Platoon Maneuvers which are Robust Towards Packet Losses”; Ph.D. Dissertation; University of California, Berkeley; December 2000.
- [3] Thomas R. Connolly; “Longitudinal Transition Maneuvers in an Automated Highway System”; Masters Thesis; Department of Mechanical Engineering, University of California at Berkeley; May 18, 1996
- [4] Bret Foreman; “A Survey of Wireless Communications Technologies for Automated Vehicle Control”; University of California at Berkeley; Systems and issues in ITS. Warrendale, PA; Society of Automotive Engineers, 1995; p. 73-79
- [5] Bret Foreman, Brett Schein, Susan Streisand; “An Infrared Inter-Vehicle Communication System”; Internal Cory Hall PATH Lab Report; University of California at Berkeley
- [6] Jonathan Frankel, et. al.; “Safety Oriented Maneuvers for IVHS”; Department of Mechanical Engineering, University of California at Berkeley
- [7] J. C. Gerdes, J. K. Hedrick; “Vehicle Speed and Spacing Control Via Coordinated Throttle and Brake Actuation”; Proceedings 1996 IFAC Conference, San Francisco, CA
- [8] Datta Godbole, John Lygeros; “Longitudinal Control of the Lead Car of a Platoon”; November 1993
- [9] Takeshi Hatakeyama, Sadao Takaba; “A Network Architecture of the Inter-Vehicle Packet Communication System”; Proceedings of the 1994 Vehicle Navigation & Information Systems Conference; A5-4; August 1994

- [10] J. K. Hedrick, D. H. McMahon, D. Swaroop; "Vehicle Modeling and Control for Automated Highway Systems"; ITS Publication, UCB-ITS-PRR-93-24; 1993
- [11] J. K. Hedrick, D. Swaroop; "Dynamic Coupling in Vehicles Under Automatic Control"; 13th IAUSD Symposium; August 1993.
- [12] J. K. Hedrick, et. al.; "Longitudinal Vehicle Controller Design for IVHS Systems"; Proceedings of the American Control Conference, Boston; June 1991.
- [13] J. K. Hedrick, M. Tomizuka, and P. Varaiya; "Control Issues in Automated Highway Systems"; IEEE Control Systems; December 1994.
- [14] A. Hitchcock; "Intelligent Vehicle/Highway System Safety: Multiple Collisions in Automated Highway Systems"; PATH/ITS, University of California at Berkeley
- [15] Ann Hsu, Farokh Eskafi, Sonia Sachs, Pravin Varaiya; "The Design of Platoon Maneuver Protocols for IVHS"; PATH Research Report, Institute of Transportation Studies, University of California, Berkeley; April 20, 1991
- [16] Ivy Pei-Shan Hsu, Jean Walrand; "Communication Requirements and Network Design for IVHS"; Institute of Transportation Studies, University of California at Berkeley; November 1993
- [17] J. Kaltwasser, J. Kassubek; "A New Cooperative Optimized Channel Access for Inter-Vehicle Communication"; Proceedings of the 1994 Vehicle Navigation & Information Systems Conference; A5-1; August 1994
- [18] Julian Leong, Jogn Tseng, Bret Foreman; "Network Layer Documentation--PATH Wireless Mobile Communications"; Institute of Transportation Studies, University of California at Berkeley; 1995
- [19] D. H. McMahon, J. K. Hedrick, S. E. Shladover; "Vehicle Modeling and Control for Automated Highway Systems"; Proceedings 1990 ACC, San Diego, CA; pp. 297-303
- [20] Kiyoshi Mizui, Masatoshi Uchida, Masao Nakagawa; "Vehicle-to-Vehicle 2-Way Communication & Ranging System Using Spread Spectrum Technique"; Proceedings of the 1994 Vehicle Navigation & Information Systems Conference; A5-3; August 1994
- [21] Isaac R. Porche, Kwang Soo Chang, William Li, Pravin Varaiya; "Real Time Task Manager for Communications and Control in Multicar Platoons"; Proceedings of the Intelligent Vehicles '92 Symposium; p. 409; June29 - July 1, 1992
- [22] C. D. Rizzuti; "Discrete Time String Stability"; M.S. Thesis; University of California, Berkeley.

- [23] Sonia R. Sachs, Pravin Varaiya; “A Communication System for the Control of Automated Vehicles”; PATH Technical Memorandum 93-5; Department of Electrical Engineering and Computer Sciences, University of California at Berkeley; September 1993
- [24] Issei Sasaki, Toshio Hirayama, Takeshi Hatsuda; “Vehicle Information Networking Based on Inter-Vehicle Communication by Laser Beam Injection and Retro-Reflection Techniques”; Proceedings of the 1994 Vehicle Navigation & Information Systems Conference; A5-5; August 1994
- [25] D. V. A. H. G. Swaroop; “String Stability of Interconnected Systems: An Application to Platooning in Automated Highway Systems”; Doctoral Thesis; Department of Mechanical Engineering, University of California at Berkeley; 1994
- [26] D. V. A. H. G. Swaroop, et. al.; “A Comparixon of Spacing and Headway Control Law for Automatically Controlled Vehicles”; M.S. Thesis; University of California, Berkeley.
- [27] Masatoshi Uchida, Yasuo Kagawa, Akihiro Okuno; “A Vehicle-to-Vehicle Communication and Ranging System Based on Spread Spectrum Technique -- SS Communication Radar”; Proceedings of the 1994 Vehicle Navigation & Information Systems Conference; A5-6; August 1994
- [28] Pravin Varaiya, Steven E. Shladover; “Sketch of an IVHS Systems Architecture”; PATH Research Report; Institute of Transportation Studies, University of California at Berkeley; February 1991
- [29] P. Varaiya; “Smart Cars on Smart Roads: Problems of Control”; IEEE Trans. on AC; February 1993.
- [30] Priya B. Viswanath, A. E. Lindsey; “Design, Verification and Simulation of Wireless Communication Protocols for the AHS”; Department of Electrical Engineering and Computer Science, University of California, Berkeley
- [31] “AVCS Vehicle-Follower Longitudinal Control Technologies”; PATH Internal Final Report on 65V313-001; January 1998.
- [32] “MPI - Wireless Vehicle Communication System”; Technical Manual provided by MPI; October 1996.
- [33] “UtiliCom - Automated Highway System Vehicle Radio Communication”; Software Requirements and Design Document; July 1996.