MICRO-MACROSCOPIC MODELING AND SIMULATION OF AN AUTOMATED HIGHWAY SYSTEM

by

Ramakrishnan Nagarajan

Thesis Submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the Degree of Master of Science in Civil Engineering

Approved:

Dr. P. Kachroo, Co-Chairman

Dr. R. Sivanandan, Co-Chairman

Dr. A. A. Trani

September, 1996 Blacksburg, Virginia

Keywords: Automated Highway Systems, Simulation, Longitudinal Control
LD
5655
V855
1996
N343
0.2
MICRO-MACROSCOPIC MODELING AND SIMULATION OF AN AUTOMATED HIGHWAY SYSTEM

by:

Ramakrishnan Nagarajan

Co-Chairman: Dr. P. Kachroo
Co-Chairman: Dr. R. Sivanandan

Department: Civil Engineering

Abstract

Intelligent Transportation Systems (ITS), which uses modern electronics and communications technology to guide or control the operation of vehicles holds great promise for increasing the capacity of existing roads, reducing congestion and accident losses, and contributing to the ease and convenience of travel. The most sophisticated of all the ITS technologies that may ultimately yield the largest benefits is the Automated Highway Systems (AHS). The AHS approach to enhance the performance of our highways is to apply automation techniques to vehicles and roadways to increase the capacity and efficiency of existing facilities, while retaining the advantages of individual mobility. The idea is to have a system with instrumented highways and vehicles which allows the automation of the driving function.

The overall objective of this research study involves the modeling and analysis of an AHS system, using a simulation tool specifically developed for this purpose. A multi-layer control system architecture that conforms to the one developed at the University of California, Berkeley, provides a framework for the micro and macroscopic modeling of the system. The focus of the system modeling is towards the lower layers of this control system architecture, involving a comprehensive modeling of the regulation and physical layers and a simple, yet realistic modeling of the functionalities of the link layer. The regulation and physical layer design incorporates a complete powertrain modeling of the vehicle that includes one-wheel rotational dynamics, linear vehicle dynamics, engine dynamics and actuator dynamics.
The longitudinal control of vehicles is an important component of highway automation research. Parametric studies of the dynamics of the vehicle model have revealed that the vehicle transients cannot be predicted through simple linear models. Therefore, the longitudinal controller design is based on the non-linear sliding mode control theory. For analysis and comparison purposes, two other headway control models are designed and calibrated.

Dynamic Visual Micro-Macroscopic Traffic Simulation (DYNAVIMTS) is a simulation package for evaluating different aspects of AHS including architectures, configurations, and designs of different controllers, engine models, sensors and communications. The package consists of three separate modules: communicator, simulator, and animator. The communicator and the simulator run on a Sun Sparc, while the animator runs on a Windows NT platform. The animation program produces a three-dimensional color animation of AHS traffic. The modeling of the lower layers in the software is a micro-simulation, where the system elements and the control policies are each individually modeled and parametrically specified so that users can easily interact with the system.

The behavior of a single platoon of automated vehicles of different platoon sizes is tested under different models and varying lead vehicle profiles to provide a comprehensive microscopic analysis of the system. A mesoscopic analysis of platoons of vehicles is also conducted to study the performance of the sliding mode controller, in comparison to the other existing models. Various scenarios have been simulated and it is seen that steady state flows of about 8000 veh/hr can be sustained. The sliding mode controller with complete vehicle modeling is found to perform better than the other headway control models and sustains upto 2m intra-platoon spacing. These simulations also clearly demonstrate the ability of the link layer model to support the concept of interplatooning as the traffic demand increases and related disturbances are produced in the system.
Acknowledgements

My greatest thanks are due to my thesis advisor and Co-Chairman, Dr. Pushkin Kachroo. It was Pushkin’s invaluable advice, enthusiastic inspiration, and never ending guidance, both as an advisor and as a friend, that made it possible for me to complete my research work at the Center for Transportation Research.

I also extend my thanks to Dr. R. Sivanandan, Co-Chairman, and Dr. A. A. Trani, for their insight, for serving as my committee members, and for reviewing thesis manuscripts and providing critical suggestions.

I am also very fortunate to have had the opportunity to work with my very good friend and colleague, Srikanth Jonnalagadda. This thesis could never have been a reality without his unending encouragement and very insightful suggestions, and also a wonderful friendship through the last few years. Another good friend and fellow graduate student, Chris Nutakor, also deserves special mention because of his unique character and strong influence on my work.

I also express my thanks to my classmates from IIT M, my seniors and juniors at Virginia Tech, my friends in Raleigh, and my colleagues at the Center.
# TABLE OF CONTENTS

1. INTRODUCTION .......................................................................................................................... 1
   1.1 THE PROBLEM ....................................................................................................................... 1
   1.2 AUTOMATED HIGHWAY SYSTEMS - A SOLUTION .............................................................. 2
   1.3 OVERVIEW OF AHS .............................................................................................................. 2
      1.3.1 THE VISION ................................................................................................................... 2
      1.3.2 BENEFITS .................................................................................................................... 4
      1.3.3 MAJOR DEPLOYMENT AND OPERATIONAL GOALS .................................................. 5
   1.4 SIMULATION ............................................................................................................................ 5
      1.4.1 THE NEED .................................................................................................................... 6
      1.4.2 DYNAMIC VISUALIZATION ......................................................................................... 7
   1.5 RESEARCH OBJECTIVES ...................................................................................................... 7

2. LITERATURE REVIEW .................................................................................................................. 10
   2.1 AHS PROGRAMS .................................................................................................................... 10
      2.1.1 AHS PROGRAM IN THE UNITED STATES ................................................................. 10
      2.1.2 AHS PROGRAM IN EUROPE ....................................................................................... 13
      2.1.3 AHS PROGRAM IN JAPAN .......................................................................................... 15
   2.2 AHS EVOLUTION .................................................................................................................... 16
   2.3 AHS CONCEPTS ...................................................................................................................... 18
      2.3.1 AHS COMPONENTS ..................................................................................................... 18
      2.3.2 FUNCTIONAL ASPECTS OF AHS ................................................................................. 19
   2.4 LONGITUDINAL CONTROL ...................................................................................................... 21
      2.4.1 VEHICLE MODELING .................................................................................................. 21
      2.4.2 LONGITUDINAL CONTROL OF A PLATOON OF VEHICLES ....................................... 22
   2.5 LATERAL CONTROL ................................................................................................................ 23
      2.5.1 VEHICLE LATERAL DYNAMICS MODELS ................................................................ 23
      2.5.2 MAGNETIC REFERENCE AND SENSOR SYSTEM .................................................... 24
      2.5.3 LATERAL CONTROL LAW DEVELOPMENTS ............................................................... 24
   2.6 EXIT AND ENTRY OF VEHICLES ............................................................................................ 25
   2.7 SIMULATION TOOLS FOR AHS SYSTEMS .......................................................................... 25

3. RESEARCH METHODOLOGY ...................................................................................................... 29
   3.1 CONTROL SYSTEM ARCHITECTURE .................................................................................... 29
   3.2 MACRO-MESOSCOPIC MODELING ....................................................................................... 34
      3.2.1 LINK LAYER FUNCTIONALITIES ................................................................................... 34
      3.2.2 SYSTEM DISTURBANCES ............................................................................................. 35
      3.2.3 INTERPLATOON DYNAMICS ....................................................................................... 36
   3.3 MICROSCOPIC MODELING ................................................................................................... 39
      3.3.1 WHEEL DYNAMICS ..................................................................................................... 40
      3.3.2 VEHICLE DYNAMICS ................................................................................................ 43
      3.3.3 ENGINE DYNAMICS .................................................................................................. 44
      3.3.4 ACTUATOR DYNAMICS ............................................................................................... 47
      3.3.5 COMBINED SYSTEM DYNAMICS .............................................................................. 47
   3.3.6 LONGITUDINAL CONTROL ............................................................................................. 49
      3.3.6.1 Jerk Control System ................................................................................................. 50
      3.3.6.2 Acceleration Control System .................................................................................... 51
      3.3.6.3 Sliding Mode Control ............................................................................................... 52
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Control System Architecture</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Interplatooning Process Controlled at the Link Layer Level</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>A Typical Mu-Lambda Curve</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Mu-Lambda Curves for Different Road Conditions</td>
<td>43</td>
</tr>
<tr>
<td>3.5</td>
<td>Wheel Dynamics</td>
<td>43</td>
</tr>
<tr>
<td>3.6</td>
<td>Vehicle Dynamics</td>
<td>43</td>
</tr>
<tr>
<td>3.7</td>
<td>Four Strokes of an Internal Combustion Engine</td>
<td>45</td>
</tr>
<tr>
<td>3.8</td>
<td>Basic Fuel Supply System</td>
<td>46</td>
</tr>
<tr>
<td>3.9</td>
<td>Vehicle/Brake/Road Dynamics: One-Wheel Model</td>
<td>49</td>
</tr>
<tr>
<td>3.10</td>
<td>Platoon of Vehicles</td>
<td>49</td>
</tr>
<tr>
<td>3.11</td>
<td>Jerk Control System</td>
<td>50</td>
</tr>
<tr>
<td>3.12</td>
<td>Acceleration Control System</td>
<td>51</td>
</tr>
<tr>
<td>4.1</td>
<td>AHS Evaluation Framework Proposed by Virginia Tech</td>
<td>63</td>
</tr>
<tr>
<td>4.2</td>
<td>Software System Architecture</td>
<td>64</td>
</tr>
<tr>
<td>4.3</td>
<td>Opening Screen of Dynavimts</td>
<td>66</td>
</tr>
<tr>
<td>4.4</td>
<td>A Typical ArcEdit Session</td>
<td>68</td>
</tr>
<tr>
<td>4.5</td>
<td>Regulation Layer Interface in Dynavimts</td>
<td>70</td>
</tr>
<tr>
<td>4.6</td>
<td>Controller Design Interface in Dynavimts</td>
<td>71</td>
</tr>
<tr>
<td>4.7</td>
<td>Input Interface for Platoon Parameters in Dynavimts</td>
<td>72</td>
</tr>
<tr>
<td>4.8</td>
<td>Input Interface for Simulation Parameters in Dynavimts</td>
<td>73</td>
</tr>
<tr>
<td>4.9</td>
<td>Structure of Communicator Source Files</td>
<td>74</td>
</tr>
<tr>
<td>4.10</td>
<td>Simulation Kernel Logic (without CSIM Libraries)</td>
<td>75</td>
</tr>
<tr>
<td>4.11</td>
<td>Program Processing Structure</td>
<td>81</td>
</tr>
<tr>
<td>4.12</td>
<td>Input File Structure</td>
<td>82</td>
</tr>
<tr>
<td>4.13</td>
<td>Output File Structure</td>
<td>83</td>
</tr>
<tr>
<td>4.14</td>
<td>Simulator Source File Structure</td>
<td>84</td>
</tr>
<tr>
<td>5.1</td>
<td>Vehicle velocity vsTime (Lead Vehicle Profile Analysis)</td>
<td>90</td>
</tr>
<tr>
<td>5.2</td>
<td>Acceleration vsTime (Lead Vehicle Profile Analysis)</td>
<td>91</td>
</tr>
<tr>
<td>5.3</td>
<td>Vehicle velocity error vsTime (Lead Vehicle Profile Analysis)</td>
<td>91</td>
</tr>
<tr>
<td>5.4</td>
<td>Mu (Friction coefficient) vs Time (Lead Vehicle Profile Analysis)</td>
<td>92</td>
</tr>
<tr>
<td>5.5</td>
<td>Lambda (Wheel slip) vs Time (Lead Vehicle Profile Analysis)</td>
<td>92</td>
</tr>
<tr>
<td>5.6</td>
<td>Distance vs Time (Model I)</td>
<td>99</td>
</tr>
<tr>
<td>5.7</td>
<td>Velocity vs Time (Model I)</td>
<td>99</td>
</tr>
<tr>
<td>5.8</td>
<td>Acceleration vs Time (Model I)</td>
<td>100</td>
</tr>
<tr>
<td>5.9</td>
<td>Velocity error vs Time (Model I)</td>
<td>100</td>
</tr>
<tr>
<td>5.10</td>
<td>Headway vs Time (Model I)</td>
<td>101</td>
</tr>
<tr>
<td>5.11</td>
<td>Distance vs Time (Model II)</td>
<td>101</td>
</tr>
<tr>
<td>5.12</td>
<td>Velocity vs Time (Model II)</td>
<td>102</td>
</tr>
<tr>
<td>5.13</td>
<td>Acceleration vs Time (Model II)</td>
<td>102</td>
</tr>
<tr>
<td>5.14</td>
<td>Velocity error vs Time (Model II)</td>
<td>103</td>
</tr>
<tr>
<td>5.15</td>
<td>Headway vs Time (Model II)</td>
<td>103</td>
</tr>
<tr>
<td>5.16</td>
<td>Distance vs Time (Model III)</td>
<td>104</td>
</tr>
<tr>
<td>5.17</td>
<td>Velocity vs Time (Model III)</td>
<td>104</td>
</tr>
<tr>
<td>5.18</td>
<td>Acceleration vs Time (Model III)</td>
<td>105</td>
</tr>
<tr>
<td>5.19</td>
<td>Velocity error vs Time (Model III)</td>
<td>105</td>
</tr>
<tr>
<td>5.20</td>
<td>Headway vs Time (Model III)</td>
<td>106</td>
</tr>
<tr>
<td>5.21</td>
<td>Sliding surface 1 vs Time (Model III)</td>
<td>106</td>
</tr>
<tr>
<td>5.22</td>
<td>Sliding surface 2 vs Time (Model III)</td>
<td>107</td>
</tr>
</tbody>
</table>
LIST OF TABLES

TABLE 3.1 WHEEL PARAMETERS ................................................................. 40
TABLE 3.2 AVERAGE PEAK VALUES FOR FRICTION COEFFICIENT .................. 42
TABLE 5.1 SLIDING MODE CONTROLLER PARAMETERS FOR LEAD VEHICLE .... 96
TABLE 5.2 SLIDING MODE CONTROLLER PARAMETERS FOR FOLLOWING VEHICLE . 96
TABLE 5.3 PLATOON SIZE AND CORRESPONDING DESIRED INTERPLATOON SPACINGS 97
TABLE 5.4 CALCULATION OF AHS CAPACITY (STEADY STATE VELOCITY = 25M/S) 110
TABLE 5.5 CALCULATION OF AHS CAPACITY (STEADY STATE VELOCITY = 30M/S) 111
TABLE 5.6 CALCULATION OF AHS CAPACITY (STEADY STATE VELOCITY = 35M/S) 112
1. INTRODUCTION

1.1 THE PROBLEM

The present system of roads and highways with manually operated cars and trucks has provided an excellent means of transportation, serving both individual and group needs. However, its ability to meet increasing demands and high volumes of traffic has been limited. As traffic has increased, so has congestion. In addition, more than forty thousand people are killed and another five million injured each year in road accidents. The construction of more highways to increase capacity is not only a costly and impractical option, but also environmentally undesirable.

What is needed is a ‘smarter’ surface transportation system, one that actually functions as an integrated intermodal system, offering improved safety, more efficient use of the existing infrastructure, and enhanced user choices.

The Intelligent Transportation Systems (ITS) approach, which uses electronics and communications technology to guide or control the operation of vehicles holds great promise for increasing the capacity of existing roads, reducing congestion and accident losses, and contributing to the ease and convenience of travel [1]. These applications may provide current, individualized information to travelers that enables them to make better decisions about routes, travel times and means of travel. Such applications may also allow authorities to manage transportation facilities and control traffic more efficiently. In the future, they may assist drivers through automation of vehicle control. Momentum has been building in the United States for a large scale initiative that would demonstrate the value of these technologies in highways and transit and that would
produce advances upon which more sophisticated and effective systems could be constructed.

1.2 AUTOMATED HIGHWAY SYSTEMS - A SOLUTION

The most sophisticated of all the ITS technologies, that may ultimately yield the largest benefits by increasing road capacity and improving safety, is the Automated Highway Systems (AHS). The AHS approach, to enhance the performance of our highways is to apply automation techniques to vehicles and roadways to increase the capacity and efficiency of existing facilities while retaining the advantages of individual mobility. The idea is to have a system with instrumented highways and instrumented vehicles which allows the automation of the driving function. By monitoring and controlling the positions of vehicles with respect to the roadway and each other, these systems may prevent collisions and allow vehicles to operate at closer spacings and higher speeds [1]. Realization of automatic vehicle control will require a substantial long-term research effort, though some applications will be available soon.

1.3 OVERVIEW OF AHS

1.3.1 THE VISION

An Automated Highway System envisages the use of modern electronics, sensors, and communications on highways and vehicles to provide ‘fully automated’ vehicle operation. Under this concept, a system of vehicles could operate on both conventional roads under manual control and on specially instrumented guideways under automatic control. This means that as an AHS vehicle moves on to an automated section of the highway, control of the vehicle’s steering (lateral control) and its braking and acceleration
(longitudinal control) is taken over by the AHS and is assigned back to the driver as he exits the automated section.

The main advantage of the AHS option over other ITS alternatives is that it complements and augments the individual automobile. The automobile is a popular form of mechanized transportation because it is comfortable, flexible, immediately available to the owner, and provides good service for many travel needs. AHS maintains these desirable features of the automobile and improves upon many if not all of its negative aspects. The acceptance of the AHS concept depends heavily on the ability of the AHS to provide benefits in the areas of capacity, safety, energy, level of service and environmental and community impacts. The potential of AHS to provide such benefits is discussed in the following section.

The studies being conducted on Automated Highway Systems and related concepts have focused on addressing issues impacting the introduction and deployment of this technology [2]. The technical approach to conducting these studies could be divided into the following phases:

- Development and selection of system concepts and implementation strategies
- Automated Highway System studies and Evaluation
- Development of system concepts and implementation recommendations

The purpose of the first phase would be to establish candidate AHS concepts to be investigated, the scenarios within which they could be evaluated, the evolutionary implementation strategies that could be analyzed and a set of parameters and measures by which the practicality of the system deployments could be judged. The second phase would involve the trade studies that would be conducted through various means to establish preferred system states for the AHS concepts and implementation plans. Simulation could play a very important role in this phase in that it could be used as a very effective and efficient test-bed for conducting these studies. The need for such a

INTRODUCTION
simulation tool has been discussed in the next section. The development of recommendations would be the final phase, in which the result of the preceding analyses would be reviewed and critiqued and plan to move the AHS concept forward to some level of hardware deployment would be developed.

1.3.2 BENEFITS

Simulations and analysis of conceptual AHS systems has shown that AHS has the potential to double or triple the capacity of our highways and dramatically increase highway safety. It also offer the additional advantage of reducing the negative environmental impact of highway vehicles. Specific examples of the benefits are:

*Increase in Capacity*: Capacity increases of the order of 300% can be realized by moving significantly more vehicles over the same land space. Basically the automation of vehicle operation will result in improvements in traffic throughput and decreases in congestion [2].

*Increase in Safety*: The safety aspect would be a result of the virtual elimination of the human factor in vehicle control, and its replacement by faster reacting, more consistent, and highly reliable automatic control systems. It is interesting to note that today over 90% of all crashes involve human error as at least a contributing factor. The system malfunctions in an AHS scenario can lead to accidents, these could be reduced to a minimum. The potential savings in terms of the increase in safety can make a major contribution to the cost-effectiveness of the AHS [2].

*Decrease in Environmental Impacts*: The reduction in the environmental impacts is expected from the fact that vehicles would be constrained to move at fixed speeds, thereby avoiding both unnecessary fuel consumption and excessive noxious exhaust products.

*Increase in Productivity*: Increased capacity and more efficient traffic management will mean less congestion and fewer traffic jams, which in turn will mean fewer lost hours of productivity and less frustration for all drivers.
1.3.3 MAJOR DEPLOYMENT AND OPERATIONAL GOALS

The major goals for the deployment and operations of a functional AHS are as follows:

*Improve Operating Effectiveness*-Increase throughput of people, goods and vehicles and improve operation in adverse weather.

*Improve Transportation Service*-Provide a full range of services, reduce travel time ad improve travel reliability

*Improve User Desirability*-Improve safety, enhance personal mobility, increase comfort of highway travel, provide user-friendly service, reduce insurance costs and ensure affordable cost.

*Improve Community Desirability*-Reduce land use, reduce property impact, reduce need for emergency support, and reduce construction disruption.

*Improve State Transportation Agency Desirability*-Provide a basis for long-term upgrade to major highways, enable smooth transition, enable smooth installation, enable practical operation, provide better cost/benefit ratio, and integrate with and support transit operations.

*Provide Societal Benefits*-Strengthen the nation’s economy, nurture the U.S. AHS industry, support national emergencies, reduce fossil fuel consumption and reduce pollutants from vehicles.

The near term objective of the AHS program however, is to test the feasibility of the AHS concepts through simulations and identify concepts that can be used for operational tests and evaluation.

1.4 SIMULATION

A simulation is the operation of a model that represents the dynamic behavior of a system. The model is amenable to manipulation that would be impossible, too expensive,
or impractical to perform on the system it portrays. The operation of the model can be studied and from it, properties concerning the behavior of the actual system can be inferred [3].

1.4.1 THE NEED

Simulation is an effective way of pretesting systems, plans, or policies before developing expensive prototypes and actual implementation of the concept. Using computer based simulation, it is possible to trace out in detail the consequences and implications of a proposed course of action. Such a feature enables us to study the causal relationships in problems that cannot be modeled analytically. In addition the ability to introduce stochastic elements allows one to estimate the performance of a real-world system under some projected set of operating conditions.

There is a need to exploit these advantages of computer based simulation and develop an inexpensive and safe platform for testing AHS concepts. This approach is further justified by the following facts:

- We are presently at an evolutionary stage in the development of Automated Highway Systems. At this juncture, there is a necessity to study and evaluate all possible AHS configurations and ideas. Simulation provides a common test-bed for all these feasibility studies.
- A complex, real-world scenario of an automated highway system with stochastic elements cannot be accurately described by mathematical models. Simulation would be one of the best means of investigation in such situations.
- The costs involved in evaluating AHS through operational tests are enormous. Investing such huge amounts of money should be preceded by a considerable understanding of the problem. Simulation provides a cost effective means of achieving this purpose.
1.4.2 DYNAMIC VISUALIZATION

A dynamic visualization environment provides an interactive computer graphics environment for displaying the dynamic motions of a system as well as important system quantities such as vehicle acceleration and velocity [4]. Iannou et. al. (1992), in their paper "Autonomous Intelligent Cruise Control", discuss the importance of a dynamic visualization tool to assist in the understanding and evaluation of performance factors like slinky effects and oscillations. According to them, the rationale for using a dynamic visualization environment is:

- People have a fundamental ability to understand motion.
- The ability to understand the highly coupled motion of a dynamic system is not substantially decreased as the number of degrees of freedom in the system is increased.

The use of graphical plots along with the digital readouts allows one to easily capture the behavior of the system. Since the automated modes allow only small deviations, appropriate speed regulation and scaling features help in making these deviations more apparent. Interactive facilities, that enable the user to perform different viewing transformations, change various program parameters and manipulate output displays, greatly increase the flexibility and robustness of the simulation tool.

1.5 RESEARCH OBJECTIVES

The objectives in this research study can be categorized into three major areas:

SYSTEM MODELING

- Define a control system architecture under which all the AHS elements, concepts and scenarios can be designed. The framework should also involve the definition of all the major system control tasks.
- Develop a model of the vehicle, incorporating wheel-roadway interactions and engine dynamics.
- Develop/identify headway control models based on existing ones, for comparison purposes.
- Design a longitudinal controller based on the theory of sliding mode control.
- Modify the sliding mode controller to work for the following vehicle as well as for the lead vehicle to follow desired profiles.
- Identify disturbances and possible inter-platoon dynamics or interactions that can occur at the link layer level of the control system architecture.
- Develop a mesoscopic link layer model to portray inter-platoon behavior that can happen in an AHS system.

SOFTWARE MODELING AND DEVELOPMENT
- Design and develop a simulation tool that can be used to model the micro-macroscopic aspects of traffic flow over a highway system. The software would have three components:
  1. A front-end graphical user interface, developed in C using the OSF/Motif libraries and the Xt Intrinsics libraries on a Sun Sparc.
  2. A simulator module, developed in the C++/CSIM programming language on a Sun Sparc.
  3. An animation module, developed in C using OpenGL libraries on a Windows NT workstation.

SIMULATION AND ANALYSIS

Lead Vehicle Profile Analysis
- Perform simulations for different desired profiles to test the lead vehicle controller.
- Change road-surface conditions to test the robustness of the controller.
**Microscopic Analysis**

- Define a set of scenarios and measures of effectiveness over which the controllers can be evaluated.
- Simulate and compare the performance of the controllers under these operating scenarios for a single platoon of vehicles.
- Estimate sustainable flows under ideal conditions for vehicles using the different controllers over the defined scenarios.

**Macro-mesoscopic Analysis**

- Simulate the interactions between platoons, starting from the entrance ramp to steady state conditions, with vehicles using the automated controllers.
- Analyze the responses in macroscopic variables over the length of the test-bed and compare the flows calculated under ideal conditions with the ones observed for the link layer model.
2. LITERATURE REVIEW

This chapter deals with the on-going research in AHS. In addition to a general overview of the research efforts in the country and abroad, some relevant areas of research were reviewed in detail. Though it was not possible to model all these issues, it was necessary to study the concepts to gain a better understanding of the working of an Automated Highway System. The findings of the review of the literature on the concepts associated with an AHS are reported in the sections that follow.

2.1 AHS PROGRAMS

2.1.1 AHS PROGRAM IN THE UNITED STATES

The AHS program in the U.S. is a part of a broader plan involving the development of Automated Vehicle Control Systems (AVCS). As a part of this plan, an intensive and systematic multi-year program of research, development, demonstration and evaluation is on to reach the goal of improving highway safety and reducing congestion through the application of AVCS concepts. It is envisioned that a large number of individual AVCS technologies will undergo system development in the coming years. Although many different initiatives are proceeding somewhat independently, an overall program plan is as follows [5]:

AVCS-I: (Individual Vehicle Control, Autonomous Vehicle Control Systems) Focuses on perceptual enhancements, warning devices and control mechanisms.

AVCS-II: (Cooperative Driver-Vehicle-Highway Systems) Requires both vehicle and highway based equipment and focuses on platooning.
AVCS-III: (Automated Vehicle -Highway Systems) Implements the complete automation of the driving function of vehicles operating on specially equipped freeway facilities.

The California PATH Program is the forerunner of research in the U.S. on AVCS. One of the major focus areas in AVCS research in this program is AHS. Recent developments are discussed in the following section.

The California Program on Advanced Technology for the Highway (PATH) has emphasized work on AVCS technology to a greater extent than any of the other current ITS programs. The PATH activities in AVCS are found in areas such as:

- defining useful AVCS concepts that appear to be worthy of further study or refinement
- performing initial feasibility evaluations of promising concepts or designs, via analyses or simulations
- evaluating trade-off among different AVCS alternatives and between AVCS and other transportation system changes
- identifying the gaps in available technology that need to be filled in order to provide a desired AVCS service
- evaluating the applicability to AVCS of technologies that were developed elsewhere, for other purposes (aerospace, defense, process control etc.)

Some of the basic assumptions underlying the PATH vehicle control development are as follows:

- The final goal is to completely automate the driving function on certain suitably equipped road facilities. Partial automation is not a desirable alternative.
- Assume that the most appropriate solutions will involve cooperation between the vehicle and the roadway and the roadway will be equipped with suitable reference markers and communication infrastructure, and that vehicles may be equipped with cooperative devices.
- Concept of fully autonomous vehicle that could travel under automatic control on any road or street would not be adopted.
- Strict adherence to the "Brick Wall" safety criterion would not be adopted.
- Approach is not to design a system to have no rear-end collision, but to design one in which the collisions that do inevitably occur have only minor effects.

The researchers at the Texas A&M university have developed a vision based autonomous vehicle named Binocular Autonomous Research Team (BART), a test-bed through which they have successfully tested their research concepts [6]. Vision based vehicle following includes automatic steering and speed control of an autonomous vehicle visually following or tracking the motion of a lead vehicle. The following vehicle is required to travel smoothly while maintaining a safety distance from the lead vehicle. The developed system consists of the following modules: Image processing module, recursive filtering module, and the driving command generator. The image processing module obtains the range and the heading angle of the lead vehicle by visually identifying a unique tracking feature on the lead vehicle. The recursive filtering module determines the appropriate response to the stimulus and the driving command generator implements the command.

Computer simulations were conducted using the map of a road with two 15-ft wide lanes. The roadway consisted of two turns of turning radii 155 and 509 ft, and three straight path segments having a total length of 1051 ft. A sampling interval of 1.5s and a delay time of 500ms were considered. A cruising speed of 22 ft/s was assumed. Simulations runs were performed to study lateral deviations, variations in steering wheel angle, speed, observation noise and model noise.

Experimental verification was done through live tests at a runway facility at the Riverside campus of Texas A&M university. In the test runs, the paths included three types of segments: straight, lane changing and curved segments. Sampling rates were varied from .5 to 1.5 seconds. BART showed successful results for speeds upto 20mph. Variations in
steering wheel angle, heading angle and range were compiled for the various test runs. At speeds higher than 20 mph, BART deviated too much to the right or to the left, especially at sharp turns. Currently steps are being taken to overcome this drawback.

2.1.2 AHS PROGRAM IN EUROPE

In Europe, ITS is known as Road Transport Informatics. One European IVHS effort, Program for European Traffic with Highest Efficiency and Unprecedented Safety (PROMETHEUS), has worked out a research agenda that is noteworthy for its emphasis on vehicle on-board functions [7]. PROMETHEUS combines the efforts of researchers in the automobile and electronic industries, as well as in university and national research laboratories, in a program to develop IVHS technologies and marketable systems as early as 1993.

The PROMETHEUS concept of IVHS consists of an electronic copilot serving as the link between the human driver and the four functional systems. The electronic copilot is the on-board information and assistance system; it receives information on the driving environment from the on-board sensors, from other vehicles, and from the roadside infrastructure. The copilot processes data and communicates with driver by way of a human-machine interface incorporating visual, acoustical and tactile feedback. Thus, although the driver maintains final control of the vehicle, the electronic copilot operates in parallel to monitor and evaluate the vehicle status and assist and inform the driver.

The three functional systems serving this electronic copilot are:

Safety Information System: The safety information system allows the electronic pilot to monitor the status of the vehicle and the driving environment and to inform the drivers of any hazards. It serves as the eyes and ears of the vehicle, sensing and informing the driver about such hazards as blind spot or slippery road surfaces. This system includes
functions for obstacle detection, vision enhancement, and monitoring of the road, the driver and the vehicle.

Active Support System: The active support system allows the electronic copilot to intervene actively in emergency situations where the driver risks losing control of the vehicle. Anti lock brake systems are a current example of this technology. The active support system would perform safety margin determination, critical course determination, dynamic vehicle control, and supportive driver information.

Cooperative Driving System: The cooperative driving system is based on vehicle to vehicle communication that allows vehicles to notify others of their actions or to emit warnings.

Intelligent Cruise Control is one of the PROMETHEUS systems for which short term implementation has been suggested. Intensive work on the development and evaluation of ICC has been done in the past few years. The basic objective of ICC has been defined as “harmonizing relative speeds and distances between vehicles in a single lane traffic flow”.

ICC-Information/ Warning informs the driver about the distance to the preceding vehicle and the difference in speed. When approaching the preceding vehicle, the current deceleration is continuously displayed in relation to the set point of deceleration, which is determined based on the relative safety distance.

ICC-Automatic Braking will brake automatically if the driver does not react to a situation in which a deceleration equal to or more than a predefined value is necessary to slow down the vehicle constantly until the speed of the preceding vehicle at the safety distance reached.

For the traffic simulation with ICC, the microscopic, time-based simulator AS (Autobahn Simulator) was used. The stochastic model in AS was basically divided into three parts: generation of vehicles, movement of vehicles, and the infrastructure with its influence on traffic. In regard to the movement of vehicles, four driving states were identified and the
parameters delimiting the states were set. The states were: uninfluenced, approaching, following, and emergency braking.

For vehicles equipped with ICC, the parameters were modified. The traffic simulations were carried out for the following road and traffic conditions:

- Road type: motorway sections with 2 or 3 lanes
- Section length: 10 km
- Equipment rates: 0%, 30%, 50%, and 100%
- Proportion of heavy traffic: 14.4%(2-lane section) and 16.7%(3-lane section)
- Four different flow rates

The simulation results revealed no significant influence of ICC on density, mean local and travel speed. But, variations in headways were smaller and harmonized. Safety was improved since time gaps less than 1 second were considerably reduced.

2.1.3 AHS PROGRAM IN JAPAN

The Japanese unlike the Americans and Europeans, have approached largely on a project basis and have not emphasized a conceptual definition of the technology. The two major Japanese IVHS research programs have been Road/Automobile Communication System(RACS) and Advanced Mobile Traffic Information and Communication Systems(AMTICS) [1,8]. Both RACS and AMTICS combine the efforts of government and industry and both have their goal the development of traffic management and traveler information systems.

Research on vehicle automation is also being conducted in Japan. In the Personal Vehicle Systems Project, Nissan and Fujitsu are developing an automatic vehicle capable of driving 30km/hr on a special test track. Research on obstacle recognition and avoidance is also part of the project. A separate project at Mazda combines computer vision, artificial intelligence, and automation technologies in an effort to develop an autonomous
highway vehicle. A project with a more short term goal is developing infrastructure oriented technologies for collision avoidance, adaptive cruise control, and lane guidance for installation on a new Tokyo-Kobe expressway.

The Japanese have recently announced a program to develop vehicle control technology that is potentially highly significant because of its emphasis on basic research for long term objectives. The Super Smart Vehicle System Project is the preliminary phase of a planned national research and development project targeting technologies that would find general applications 20-30 years from today. The project would develop vehicle-vehicle and road-vehicle communications for accident avoidance and other vehicle control functions. Participants are private electronics and automobile firms, universities, and the government.

2.2 AHS EVOLUTION

The evolution in the technology has necessitated the need for a corresponding evolutionary growth in the efforts to achieve a fully Automated Highway System. The development of a fully functional Automated Highway System from the current system of manually driven cars is envisioned to take place in three major evolutionary phases.

The first phase would be constituted of both manually operated cars as well as cars equipped with Intelligent Cruise Control (ICC). ICC Information/Warning informs the driver about the distance to the preceding vehicle and the difference in speed. An acoustic warning system could also be installed which would warn the driver if the current acceleration exceeds a certain precalculated value for that speed differential and relative distance. An Automatic Braking System will brake automatically if the driver does not react to a situation in which a deceleration equal to or more than a predefined value is necessary to slow down the vehicle until the safety distance is reached. The assessment
results in [7] showed considerable safety benefits whereas the decrease in transportation efficiency was negligible.

The second phase involves an architecture in which fully automated vehicles act autonomously so that mixtures of automated and manually operated vehicles are permitted. Agre and Clare describe the studies conducted on this phase in their paper, “Spontaneous Platooning: A self organizing approach to improve traffic flow capacity” [9]. The concept was tested and validated using a microscopic simulation model of a freeway segment with on and off-ramps and associated vehicle-following, merging, exiting, and lane-changing rules. Numerous scenarios were tested and flow capacities upto 7200 veh/hr were achieved.

The final phase involves the complete segregation of automated vehicles from manual driven vehicles. Extensive studies on this phase have been done under the PATH Program at the University of California at Berkeley [10]. The flow benefits of Autonomous Intelligent Cruise Control was studied and evaluated in scenarios consisting of separate lanes for automated and manual vehicles. The manual lane, which would also act as a transition lane would reduce flow capacities of the highway to a large extent depending on the proportion of automated vehicles. Furthermore, the research at PATH also argues that it is highly unlikely that a partially automated system allowing both manual and automated vehicles will contribute to a significant capacity increase. Thus, an architecture consisting of a fully automated system was defined and tested for feasibility at PATH, using an Automated Highway Simulator called SmartPath. The layered system architecture promises to yield vast benefits in terms of both safety and capacity.
2.3 AHS CONCEPTS

An AHS concept is a system configuration that is defined by a set of characteristics, and is fundamentally different from other conceptual-level AHS system configuration. It is important to note that the AHS goals will not vary from one AHS concept or design approach to another. What varies are the design characteristics of the various system concepts or design approaches postulated to meet the goals. Stevens in his paper on AHS system concepts defines an initial set of AHS components using the system characteristics [11].

2.3.1 AHS COMPONENTS

AHS goals can be met by a variety of conceptual-level system configurations or concepts. Both functional and physical characteristics may distinguish among AHS concepts. The functional and physical characteristics of an AHS can be described in the context of these major components.

- Vehicle
- Roadway infrastructure
- Command and control
- Entry and exit infrastructure
- Communications
- Operations and maintenance

The first three AHS components - vehicle, roadway infrastructure, and command and control have all these major characteristics. The variations in the communications component and operations and maintenance component does not distinguish one overall AHS concept from another. The entry and exit infrastructure component has characteristics and variables that are not fundamental differences, i.e., changes in approaches will not change the way in which either the vehicle or roadway is designed,
and the entry and exit strategies could be changed without changing the rest of the system. However, the strategies adopted for entry and exit will influence the behavior of the system and will determine the effectiveness of the concept employed.

2.3.2 FUNCTIONAL ASPECTS OF AHS

An Automated Highway System is a complex system in which maneuvers of vehicles must be properly coordinated so that efficient and safe performance can be achieved. It is necessary to explicitly define the AHS functions before performing an operational or functional evaluation of the system. The functional definition proposed by, Zhang, Shladover and Hall of the California PATH program, provides a useful foundation for further work on the operational aspects of an AHS [10]. The proposed definition is briefly discussed below:

AHS operations are specified using two sets of functions, namely operational functions and elemental functions. The operational functions define the type of operation a vehicle can perform while the elemental functions state how the system accomplishes these operations. This two level definition describes both the operations of the AHS vehicles and the functions of the hardware and software components that implement the operations.

Operational Functions

The operational functions are the functions that implement operational events under both normal and abnormal conditions. The abnormal conditions may include accident conditions in which a vehicle is required to perform emergency maneuvers to avoid an accident, incident conditions, in which a vehicle is instructed to change lanes or routes to avoid congestion and failure conditions in which hazard reduction control is needed to prevent a vehicle from encountering hazards.
The operational functions are defined in such a way that more than one function can occur simultaneously. For e.g., two operations named lane tracking and velocity regulation can accomplish an operational event "lane flow" which involves controlling a vehicle to follow a traffic lane and adjusting the velocity of the vehicle to match a target speed. Seventeen operational functions have been defined in this manner to account for operations such as entering and exiting the system transition from human to automatic control, vehicle check-in etc.

**Elemental Functions**

The complete AHS system can be defined using a set of functions which implement the operational functions. These functions, called the elemental functions, are grouped into a five layered IVHS architecture [10]. These layers and the type of elemental functions associated with each are discussed below:

*Network layer:* The network layer is responsible for route and flow control within a network. Based on the nature of esquire the network layer can provide either information reflecting the traffic conditions on a specific route or route recommendations designed to achieve a desired traffic flow. For e.g. route recommendation, monitoring traffic conditions and vehicle ID assignment are elemental functions that belong to this layer.

*Link layer:* The link layer is responsible for path and congestion control within an individual link on the assigned route. The link layer may select a lane for each vehicle, set target speeds for each vehicle or platoons, for each section of the route and manipulate platoon size depending on the flow. It may also prioritize the vehicle’s operation during cooperative maneuvers and manage incident responses.

*Coordination Layer:* The coordination layer is responsible for microscopic management of a sub-section within a link. The elemental functions in the coordination layer would inspect and monitor vehicle and traffic flows, issue permission/rejection, and coordinate maneuvers. The coordination layer also provides information on the road surface
conditions and the weather. In a system with platoons, the coordination is also responsible for splitting and joining platoons.

*Regulation Layer:* Regulation layer carries out the directions of the coordination layer. It tracks targets speeds, maintains separation between vehicles and between platoons and provides commands to perform steering and speed control for maintaining the lateral position of the vehicle and the longitudinal separation between vehicles. It is also responsible for monitoring vehicle conditions and for on-board failure detection.

*Physical Layer:* The physical layer includes the actuation and sensing devices that actually carry out the control commands of the regulation layer and feed information back to it. The physical layer is also responsible for human-machine interaction.

### 2.4 LONGITUDINAL CONTROL

The PATH work on longitudinal control emphasizes the concept of vehicle-follower control over point-follower control. This is based on the relative advantages and disadvantages of these concepts [10,12,13,14,15].

#### 2.4.1 VEHICLE MODELING

A mathematical representation for the power train including internal combustion engine dynamics and tire/road frictional interface is a necessity for the evaluation of potential longitudinal control models. As part of the PATH program, a non-linear vehicle power train program called LONSIM has been developed that incorporates features to indicate that a vehicle is planning to leave the platoon or that it is facing some emergency situation. The package comprehensively incorporates the nonlinear features of the internal combustion engine as well as the tire/road frictional characteristics. The inputs to
the program are the throttle and the braking commands and the outputs are the vehicle's acceleration, velocity and position.

The analysis and simulation of strategies for controlling the spacings between vehicles in a platoon show that significant improvements are obtained when the control system in each vehicle has timely access to the following information:

- Speed and acceleration of vehicle
- Distance to the preceding vehicle
- Acceleration and speed of the preceding vehicle
- Acceleration and speed of the first vehicle in the platoon

The time constant is obtained by fitting a first order model to measured speed and acceleration. The speed and acceleration of each vehicle can be measured by sensors inside the vehicle. The distance to the preceding vehicle can be measured by a microwave radar or by an optical radar or triangulation device. The speed and acceleration of the preceding can be derived from this. Telecommunications can also transmit the speed and acceleration of the first vehicle in the platoon.

The choice of the communication technology is based on communication characteristics like transmission delay, cross talk, effect of motion, and effects of environmental conditions, on the reliability and on the cost anticipated for high volume productions.

2.4.2 LONGITUDINAL CONTROL OF A PLATOON OF VEHICLES

The subject of design and analysis of vehicle longitudinal control laws have been studied extensively. The current control laws take full advantage of the recent advances in communication and measurement. A recently proposed one differed from the ones existing in literature in that it uses the acceleration of the first vehicle in the platoon. Communication speed and data processing should be very fast since it has to exceed the time constant as measured by the vehicle dynamics. Simulation runs are conducted to
check that the deviations in positions of all the vehicles from their respective pre-assigned positions are less than a certain limit and that the deviations in successive vehicle spacings from the front to the back of a platoon of identical vehicles do not exceed a maximum limit. Models have also been modified to incorporate the effect of non-identical vehicles.

The modeling, design and analysis of the vehicle-follower system was verified in a program of full scale testing. The experimental program is being conducted on a eight mile stretch of high occupancy vehicle lanes in the median of Interstate 15 to the north of San Diego. The experimental program is built around the use of a Doppler Radar System developed by the Radar Systems Control Corp (RCS) as an automotive collision warning device. The ultimate outcome of the testing is expected to be an assessment of the feasibility of the close formation platoon operations and the development of performance specifications for the next generation of system that will be needed.

2.5 LATERAL CONTROL

The PATH lateral control work is focused on the concept of cooperation between the vehicle and the roadway, with an intelligent vehicle receiving much of the information it needs from special elements installed in the roadway [10]. This approach is thought to be much simpler and less expensive than the fully autonomous vehicle concept which relies on a vision system to see the existing lane edge striping.

2.5.1 VEHICLE LATERAL DYNAMICS MODELS

Two dynamic models have been developed for the design and analysis of the vehicle lateral controller. A six degree of freedom nonlinear model is for representing the vehicle dynamics as realistically as possible, and a linear model retains only the lateral and yaw dynamics. The former is utilized in the simulation study for evaluating the controller, and
the latter in the design of the controller. The vehicle dynamics depend on the cornering stiffness, load and longitudinal velocity.

2.5.2 MAGNETIC REFERENCE AND SENSOR SYSTEM

An assessment of vehicle lateral control technologies has revealed that the sensing / reference system for automated vehicle lateral control should possess the anticipatory capability for acquiring information on both the vehicle state and preview information on upcoming road geometry in order to enable the lateral control system to smoothly and accurately guide the vehicle along curved roads, particularly at high speeds. In light of this an intelligent sensing/reference system should be applied. PATH studies indicate that approaches might be used to acquire this information include

- a smart vehicle borne sensing system that would objectively perceive the road geometry and the vehicle state information-a direct video sensing approach
- a system which would utilize an intelligent reference system capable of conveying information to the vehicle-a coding and sensing approach
- a combination of the two.

2.5.3 LATERAL CONTROL LAW DEVELOPMENTS

The objective of the lateral controller is to keep the tracking error small while maintaining good ride quality. Ride quality can be quantified in terms of the vehicle lateral acceleration. Therefore, a primary design consideration is the tradeoff between the tracking error and lateral acceleration. The requirements for tracking error and acceleration are not uniform over the entire frequency range. Based on these observations, a promising design tool for the lateral controller is the frequency shaped linear quadratic optimal control approach.
As in the case with longitudinal control development work, PATH has advanced to the stage where they need experimental verification of the analyses and designs that have been developed thus far. PATH also needs to verify that the vehicle lateral dynamics are adequately represented by the two models they are using for design and simulation of control responses.

2.6 EXIT AND ENTRY OF VEHICLES

The infrastructure used for the exit and entry of vehicles may not require the redesign of an entire AHS concept, but the methods employed will affect, to a large extent, the operations on the AHS [1,10]. Some of the exit and entry variables, which will influence the behavior of the system and determine the effectiveness of the concept employed are:

*Time of Check-in:* On the fly in buffer lanes versus slow/stop at check-in stations; All concepts will have on the fly check-in as a goal, if technically possible.

*Queue for Check-in:* Intermixed vehicles versus separate lanes for certain types of vehicles; this variant would be decided by assessment of which is more efficient at a particular entry plaza, and in fact a given system could employ both options; either approach could be used by most concepts.

*Queue for Merging:* All vehicles enter sequentially intermixed into the same AHS lane versus vehicles separated by type and entering by platoon, possibly into specialized lanes; this variant would also be decided by assessment of which is more efficient at a particular entry plaza, and in fact a given system could employ all of the options.

2.7 SIMULATION TOOLS FOR AHS SYSTEMS

**SmartPath**

SmartPath is a highway system simulator developed at the PATH program at the University of California, Berkeley [16]. It is designed to provide a framework for
simulation and evaluation of Intelligent Vehicle Highway System alternatives. SmartPath can simulate automated, manual or mixed mode traffic; it also accommodates different control, communication and computing architectures. SmartPath is micro-simulator, i.e. the functional elements and the behavior of each vehicle and highway with respect to normal and degraded mode of operations are each individually modeled.

SmartPath consists of two separate modules: simulation and animation. The SmartPath animator is a tool to view and examine the simulated data of the AHS in the most natural way. The effect is akin to what might be seen from a traffic helicopter moving over the AHS. The user of the helicopter can be restricted to the highway or forced to follow a specific car. The simulation data provides information about the speed, position, and maneuvers of each vehicle in the AHS at every unit of simulation time. With the animation interface, the user can select a vehicle and view the interaction between the vehicle and its neighboring vehicles.

In SmartPath, a vehicle can have different behaviors depending on the type of lane it is traveling on, for example, when a vehicle is in a manual lane, it has manual behavior, and as soon as it enters a transition lane or an automated lane, it switches from the manual behavior to the transition or automated behavior, respectively.

CSIM, a C-based, general purpose, process-oriented simulation environment, which allows different modules inside the car to operate independently, with the ability to communicate with each other, was used build SmartPath. In SmartPath, a car is composed of five independent and communicating modules. The five modules are sensors, communications, regulation, maneuvers and supervisor. The sensors and communication module provide information about the vehicle's surrounding environment needed to safely perform a maneuver. The regulation module represents the regulation layer and the supervisor and maneuver modules jointly model the coordination layer.
The physical layer is integrated with the regulation layer assuming that the car dynamics accurately respond to the commands of the regulation layer.

Highway is the second entity that needs to be modeled. In SmartPath, a highway is defined by its length, maximum number of lanes, number of automated lanes and number of exits and entrances and their locations. The physical topology of the highway is not part of the specification and is irrelevant to the simulation. However, the physical topology must be specified thoroughly for the animation. A highway is divided into smaller structures called sections. The length of a section in a highway is a user defined parameter which applies to all sections of the highway. A section can have no more than one exit or entrance ramp and the ramp lengths are equal to the section lengths.

The first process is ‘sim’, which initializes the various simulation parameters from an input file. ‘Sim’ constructs the network, makes the routing table and creates the link layer process. The ‘entrance-car-maker’ and the ‘automated-car-maker’ processes, scheduled at appropriate times, create the cars from entrances and automated lanes, respectively. The cars are generated by the creation of their respective supervisor process. The supervisor process of each car then creates processes for the sensors, communications and regulation modules. The maneuver module is a collection of three processes and according to the maneuver that needs to be performed, the appropriate process is activated. ‘Sim’ also creates the update process that updates the position of all vehicles after the regulation layer calculates the distance it has to move over that update period. The sensor process is then activated to detect the presence of other cars.

TWOODS
A spontaneous platooning concept proposed by Agre et al in [9] was verified by the authors using microscopic simulation testbed developed at the Rockwell International Science Center. The IVHS simulation tool is implemented in the TIME WARP OBJECT ORIENTED SIMULATION (TWOODS) system. This system is particularly well suited
for the microscopic simulation needs of IVHS research. Although the current version executes on a single Sun Workstation, the TWOODS environment supports parallel and distributed simulation so that the additional computational power can be brought to bear as the model complexity and scope increases.

TWOODS is written in C++, which is becoming the industry standard object oriented language. The IVHS simulation test-bed has an X-Windows based graphical user interface. This permits straightforward construction of mouse driven elements for user-interaction, such as pop-up windows, scroll bars and buttons. X-Windows is also an industry standard allowing portability and easier modification by later program developers.

The basic entities in the current IVHS simulation model are (1) vehicles and (2) lanes. The vehicles travel on a single segment of a one-way highway which consists of multiple lanes. The outer lane may have any number of on and off ramps at selectable locations. Speed limits can be placed anywhere on any lane; on and off ramp speed limits also may be specified. The essential simulation objects, however, are the vehicles, or more specifically, the rules that determine vehicle actions. The discrete event simulation is operated in a time-stepped mode with a basic update rate, typically selected at 100ms. At each update, the position and speed of every vehicle is modified and a new acceleration is computed for the next time interval according to the vehicle behavior rules.
This chapter deals with the development of macroscopic and microscopic models for simulating an automated highway system defined under a control system architecture that conforms to the one developed in the PATH program at University of California, Berkeley [10]. The first section of this chapter presents the details of this control system architecture. This is followed by a link level mesoscopic formulation involving the modeling of the interplatoon dynamics and potential disturbances to traffic flow in an AHS system. The latter half of this chapter deals with microscopic formulation of the engine and controller dynamics dictating the behavior of each individual vehicle according to a prescribed set of rules. Previous research undertaken on this implemented simple control laws that would describe the behavior of the vehicles in the system. These control laws did not incorporate detailed low-level systems dynamics modeling of the behavior of vehicles to an applied input torque. This study also incorporates realistic vehicle dynamic models in the simulation of the system and continuing research would involve verification and validation of the models through experimental work.

3.1 CONTROL SYSTEM ARCHITECTURE

Automatic control of a vehicle is carried out by a four-layered hierarchical control architecture. The layers of the architecture, starting at the top, are network layer, link layer, planning layer and regulation layer, as shown in Figure 3.1.:

*Network Layer:* The whole automated highway network is controlled by one network layer controller. Every vehicle that enters the automated network would be assigned a route by the network controller. The route would be determined based on the origin and
Figure 3.1 Control System Architecture
destination of the vehicle. Thus the controller determines a routing table with origin-destinations and vehicle routes. The table is also updated by the controller with an approximate update rate of around 15 minutes. The aggregate state of the traffic, which is used to update the routing table, determines the decisions taken by the controller. The network layer also controls admission into the automated system through entrance ramps.

**Link Layer:** Every long segment of each highway in the network is controlled by a link layer controller. The tasks of the link layer controller would include assigning paths to each vehicle so that there is a balance between the flows in the various lanes, and assigning target speeds in each section so as to smooth out the flow and avoid congestion. The target speeds would also be reassigned by the link layer controller during incidents. The path assignments are updated approximately every minute by the controller, in response to changes in aggregate traffic flow characteristics in each section. The link layer controller also assigns the target platoon size for the section.

**Planning Layer:** There is a planning layer controller for each vehicle. The controller performs three tasks:

- Determines which of the three maneuvers, merge, split, and change-lane should be executed in order to realize its assigned path and to maintain the vehicle’s trajectory and platoon size close to the link layer’s recommendations.
- Coordinates that maneuver with the planning layers of neighboring vehicles to ensure safety.
- Supervises its regulation layer in the execution of a trajectory corresponding to the maneuver.

In the merge maneuver, two platoons join to form a single platoon; in split, one platoon separates to form two platoons and in change-lane, a free agent changes lane. Merge and split are initiated by the planning layers of platoon leaders, while change-lane is initiated by the planning layer of a free-agent. Though, followers are allowed to request the leaders to initiate a split, they are not allowed to initiate a maneuver. The safe execution
of a maneuver also involves the initiation of a structured exchange of messages with the leaders of platoons. These messages are also termed as protocols, because their function is analogous to communication protocols. The exchange results in the granting of permission by the other platoon leaders, so that the maneuver could be performed safely. This step is then followed by instruction to the regulation layer to execute the maneuver. These tasks of the planning layer necessitate the tracking of information about the state of the vehicle. This state, also known as the planning_state of the vehicle consists of the following information.

\(N\) - Target platoon size assigned by the link layer

\(v\) - Target speed also assigned by the link layer

\(l\) - Highway lane, in which the vehicle is currently located

\(d\) - Highway section, in which the vehicle is currently located

\(veh\_ID\) - Identifier variable of the vehicle

\(pltn\_ID\) - Identifier variable of the platoon in which the vehicle is currently located

\(pltn\_size\) - Current size of the platoon

\(pltn\_pos\) - Vehicle’s current position in the platoon

\(busy\) - Flag which shows whether the vehicle is currently engaged in a maneuver or not.

This layer has been described as a discrete event dynamical system consisting of finite state machines, since all the tasks involved are achieved by discrete events. The possible scenarios that could arise for the planning layer controller are as follows:

Platoon leader- Initiate merge, split, and acknowledge requests for split, merge and lane_change. Appropriate messages are then passed on to the regulation layer.

Follower- Initiate a request for split, receive acknowledgments and pass appropriate messages to the regulation layer.

Free-agent- All scenarios of a platoon leader and lane_change.

**Regulation layer:** The regulation layer implements the maneuvers requests of the planning layer through five types of feedback control laws and in the absence of a request, the tracking law for the platoon leader and the follower law for the follower is initiated.
**Follower Spacing Control**- When the vehicle is a follower in a platoon, it maintains the close spacing with the vehicle ahead through a feedback control law. The control is decomposed into longitudinal and lateral control. Longitudinal control determines the acceleration or braking of the vehicle and the lateral control determines the steering action that the vehicle needs to do to perform certain maneuvers or to stay in its lane. The control law works on the assumption that the information on relative distance and relative velocity between itself and the vehicle in front and the speed and acceleration of the platoon leader is available.

**Leader Tracking Target Speed**- The leaders of a platoons try to track the target speed specified by the link layer, while maintaining a safe distance from the vehicle in front. This task is similar to that of the current cruise control appropriately modified to account for the headway requirements.

**Accelerate to Merge**- This feedback law is used by the platoon leaders when they have to accelerate and merge with the platoon in front. Given the speed and location of the vehicle in front, a nominal trajectory is chalked out and the corresponding open loop control is embedded into the longitudinal control feedback loop.

**Decelerate to Split**- When a split maneuver is initiated, the follower who assumes the role of a leader has to slow down to a safe distance from the vehicle ahead. The law is similar to and a subset of the previous law and can be initiated only by a follower.

**Free Agent Change Lane**- A one-vehicle platoon is called a free-agent. A free-agent moves safely to an adjacent lane by executing this law. Extremely accurate sensing-systems are required for the execution of this maneuver. The implementation of this law is similar to the previous two laws.

The dynamics of the vehicles has been modeled as a part of the physical layer. It can be considered as a differential equation model of the vehicle's states that receives steering, throttle and brake actuator commands from the regulation layer and returns information such as the vehicle's speed, acceleration, position, engine state etc, which are needed to implement feedback control laws.
3.2 MACRO-MESOSCOPIC MODELING

This section describes a link layer model that could be used to simulate the operations and the interactions of platoons of automated vehicles in an AHS system. The key concepts in defining link layer strategies would depend largely on the profiles generated for the lead vehicle, possible lane changing maneuvers, and exit and entry of vehicles into the system. A simulation software, specifically developed for evaluating different aspects of AHS including architectures, configurations and strategies is described in Chapter 4. This software, DYNAVIMTS, is designed to address microscopic as well as macroscopic traffic behavior in the multi-layer AHS architecture described in the previous section. Though the software developed contains the framework to build all the components of this architecture, the study by itself focuses on a detailed modeling of the lower layers of this architecture. The second half of this chapter presents the details of a systematic comprehensive modeling of the regulation and physical layers. The vehicle-level aspects of this architecture are tested on a simplified, yet realistic model of the inter-platoon dynamics involved in the link layer. This would lay the foundation to future research directed towards a macroscopic network level modeling of the entire architecture. The scope of this study has thus been restricted towards a mesoscopic modeling of vehicles over segments of a link.

3.2.1 LINK LAYER FUNCTIONALITIES

The section of highway between two nodes in a network is called a link and each of these links is controlled by a link layer controller. The tasks of the link layer controller, as described earlier, would include

1. Assigning paths to each vehicle so that there is a balance between the flows in the various lanes.
2. Assigning target speeds in each section so as to smooth out the flow and avoid congestion and to assign target platoon size for the section.
3.2.2 SYSTEM DISTURBANCES

A link layer model of an actual multi lane AHS system with entrances and exits, is expected to experience disturbances in the following ways:

Lane Changing: This maneuver is initiated either by a platoon leader or by a free agent and is performed when the free-agent is trying to exit or when the platoon tries to change lanes to optimize lane usage.

Entrances: Vehicles could either enter as free-agents or as a platoon, depending on how the entry system is designed. This causes the platoons already on the highway to catch up with the vehicles that have entered, or slow down to accommodate them and still maintain the necessary safety spacing.

Since the system under study has been simplified to a single lane AHS without any entrances or exits, there is a need to introduce disturbances that would emulate the ones that occur in an actual system. This has been simulated through an interplatooning link layer model that tries to bring platoons to a pre-specified steady state inter-platoon spacing. The link layer controller essentially specifies a profile for the lead vehicle of a platoon, based on the current interplatoon distance and the operating modes of the lead vehicle and the vehicle in front. The objectives of this strategy can be summarized as follows.

- Profiles are generated for the lead vehicles of platoons to simulate the operation of merging up to a free-agent that has just entered the highway.
- The spacing between platoons should be minimized so that throughput is maximized in the system.
- The model tries to reduce the number of gaps into which vehicles cannot merge, so that vehicles entering the system can do so with least perturbations.
- The controller also tries to make sure that the generated profile keeps the oscillations in the system within a specified limit.
• The propagation of disturbances, also known as slinky effects, is also kept to a minimum by restricting the number of platoons involved in the inter-platoon dynamics.

3.2.3 INTERPLATOON DYNAMICS

The inter-platooning process described in this model is controlled by the link layer controller. This involves obtaining information about state variables and operating modes of vehicles in the system, appropriate switching of these modes, assigning velocity profiles to free-agents and determining the behavior of vehicles in each of the vehicle’s operational states.

Behavioral States of Vehicles

The operating mode of the vehicles in the system can be in one of four possible states. It has been assumed that this operating mode is one of several state information that a vehicle can obtain from the vehicle in front. In the case of lead vehicles of platoons, this information is assumed to be transmitted through road-side infrastructure, since the vehicle in front could be out of sensor range. This information is then used to determine any possible mode changes for that vehicle. In an implementation of the actual system, the planning layer would initiate and control these operating mode changes, but for the purpose of this study, the link layer is assumed to perform all these functions. A brief description of all the modes and the vehicle behavioral rules in each mode is given below.

1. Platooning mode: All vehicles entering the system are initialized to this mode. The lead vehicle of each platoon follows the desired velocity profile, irrespective of the state of the vehicle in front. The following vehicles use the headway controller to maintain the desired velocity and spacing.

2. Platooned mode: As soon as the lead vehicle of a platoon reaches its steady state velocity, it changes its operating mode from “platooning” to “platooned”. This forces a
similar immediate change in all the other vehicles in this platoon, even though they might not have reached the steady state velocity or spacing. This approximation has been introduced into the model for the following reasons:

- Changes in vehicle operating modes affect only the following platoons, which are usually at a distance of about 70m, when this switch takes place.
- Previous simulations of a single platoon of vehicles showed that the average error in velocity and spacing was negligible when compared to the 70m headway that would be sensed by the lead car of the following platoon.
- If all the vehicles in a platoon are allowed to reach steady state before a change in operating mode, then the headway between platoons becomes too large to be reduced to the desired value within a reasonable amount of time, given the limits on vehicle dynamics.

3. *Inter-platooning mode*: If a vehicle is in the platooned mode and if the vehicle in front is within a range, as determined by the link layer controller, then it can change to the inter-platooning mode. The link layer controller, then specifies a profile for the lead vehicle of the platoon based on its current state variables and that of the last vehicle in the predecessor platoon (platoon in front). A description of this interplatooning model is provided in the next section.

4. *Inter-platooned mode*: When the lead vehicle of a platoon reaches the desired interplatooning spacing, it changes its mode from “inter-platooning” to “inter-platooned”. This continues until the next disturbance in the system, which could of the form of new vehicles entering or exiting of vehicles in this platoon.

*Inter-platooning Model*

The model uses a constant acceleration and a constant deceleration profile for the successor platoon to close down the gap with the predecessor. Figure 3.2 shows a schematic description of the vehicle behavior in this model.
Figure 3.2 Interplatooning process controlled at the link layer level

Given the structure of the acceleration profile, the velocity that the vehicle needs to attain and the time taken for the whole process are calculated based on the following equations:

\[ s_1 = ut \]  \hspace{1cm} (3.1)

\[ s_2 = \frac{v^2 - u^2}{acc} \]  \hspace{1cm} (3.2)

\[ s_2 - s_1 = dips - ips \]  \hspace{1cm} (3.3)

Replacing for \( s_1 \) and \( s_2 \) in Equation (3.3), we get

\[ \frac{v^2 - u^2}{acc} - ut = dips - ips \]  \hspace{1cm} (3.4)

\[ t = \frac{2(v - u)}{acc} \]  \hspace{1cm} (3.5)

Replacing for \( t \) in Equation (3.5), we get

\[ \frac{v^2 - u^2}{acc} - \frac{2u(v - u)}{acc} = dips - ips \]  \hspace{1cm} (3.6)

Solving for \( v \) in Equation (3.6), we get
\[ v = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \] (3.7)

where

\( a = 1 \)
\( b = -2u \)
\( c = u^2 + acc(dips - ips) \)

\( acc \) is the acceleration (decln) of the vehicle
\( v \) is the peak velocity of the successor platoon
\( u \) is the steady state velocity
\( t \) is the time taken for the interplatooning process
\( dips \) is the desired inter-platoon spacing
\( ips \) is the initial spacing between the platoons

### 3.3 MICROSCOPIC MODELING

In this section, a realistic powertrain model of a vehicle developed for simulation purposes is described. The components of the powertrain are described under different sub-sections. The latter half of the chapter deals with the non-linear control strategy adopted for this research effort and also provides a description of the design of the longitudinal controller. The vehicle model identifies five state variables and two input variables. Three of the state variables in this model are associated with one-wheel rotational dynamics and linear vehicle dynamics and the other two are associated with the engine and actuator dynamics. The wheel dynamics and vehicle dynamics are derived by applying Newton's law. The engine dynamics are a simplified model of a continuous four stroke spark ignition engine and the actuator dynamics are derived from the braking system and the throttle characteristics of the vehicle.
3.3.1 WHEEL DYNAMICS

The dynamic equation for the angular motion of the wheel is

\[ \dot{\omega}_w = \frac{[T_e - T_b - R_w F_t - R_w F_w]}{J_v} \]  (3.8)

where \( J_v \) is the effective moment of inertia of the vehicle reflected onto the wheel, \( \dot{\omega}_w \) is the angular velocity of the wheel, the over dot indicates differentiation with respect to time, and the other quantities are as defined in Table 3.1. The total torque acting on the wheel divided by the moment of inertia of the wheel equals the wheel angular acceleration. The total torque consists of shaft torque from the engine which is opposed by the brake torque and the torque components due to the tire tractive force and the wheel viscous friction force. The wheel viscous friction force is the friction force, which is a function of the wheel angular velocity developed on the tire-road contact surface. The tractive force developed on the tire-road contact surface is dependent on the wheel slip which is the difference between the vehicle speed and the wheel speed, normalized by the vehicle speed for braking and the wheel speed for acceleration. The engine torque and the effective moment of inertia of the driving wheel depend on the transmission gear shifts.

| \( R_w \) | Radius of the wheel |
|\( N_v \) | Normal reaction force from the ground |
|\( T_e \) | Shaft torque from the engine |
|\( T_b \) | Brake torque |
|\( F_t \) | Tractive force |
|\( F_w \) | Wheel viscous friction |

Table 3.1 Wheel Parameters
Figure 3.3 A typical Mu-Lambda Curve

Applying a driving torque or a braking torque to a pneumatic tire produces tractive force at the tire-ground contact patch [17, 18]. The driving torque produces compression at the tire tread in front of and within the contact patch. Consequently, the tire travels less distance than it would if it were free rolling. In the same way, when a braking torque is applied, it produces tension at the tire tread within the contact patch and at the front. Because of this tension, the tire travels more distance than it would if it were free rolling. This phenomenon is referred to as the deformation slip or wheel slip [17, 18, 19,]. The adhesion coefficient is a function of wheel slip \( \lambda \). Figure 3.3 shows a typical \( \mu-\lambda \) curve. References [17,18] are the sources for the typical curve and a mathematical description of the tire model. Mathematically, wheel slip is defined as

\[
\lambda = \frac{\omega_w - \omega_v}{\omega} \quad \omega \neq 0
\]  

(3.9)

where, \( \omega_v \) is vehicle angular velocity defined as

\[
\omega_v = \frac{V}{R_w}
\]

(3.10)

which is equal to the linear vehicle velocity, \( V \), divided by the radius of the wheel. The variable \( \omega \) is defined as

\[
\omega = \max(\omega_w, \omega_v) = \begin{cases} 
\omega_w & \text{for } \omega_w \geq \omega_v \\
\omega_v & \text{for } \omega_w < \omega_v
\end{cases}
\]  

(3.11)
which is the maximum of vehicle angular velocity and wheel angular velocity.

The tire tractive force is given by

$$F_v = \mu (\lambda)N_v$$  \hspace{1cm} (3.12)

where the normal tire force, $N_v$, depends on vehicle parameters such as the mass of the vehicle, location of the center of gravity of the vehicle, and the steering and suspension dynamics. The adhesion coefficient, which is the ratio between the tractive force and the normal load, depends on the road-tire conditions and the value of the wheel slip. For various road conditions, the curves have different peak values and slopes, as shown in Figure 3.4. The adhesion coefficient-slip characteristics are influenced by operational parameters like speed and vertical load. The average peak values for various road surface conditions are shown in Table 3.2 [18].

<table>
<thead>
<tr>
<th>Surface</th>
<th>Average Peak Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asphalt and concrete</td>
<td>0.8-0.9</td>
</tr>
<tr>
<td>Asphalt (wet)</td>
<td>0.5-0.6</td>
</tr>
<tr>
<td>Concrete (wet)</td>
<td>0.8</td>
</tr>
<tr>
<td>Earth road (dry)</td>
<td>0.68</td>
</tr>
<tr>
<td>Earth road (wet)</td>
<td>0.55</td>
</tr>
<tr>
<td>Gravel</td>
<td>0.6</td>
</tr>
<tr>
<td>Ice</td>
<td>0.1</td>
</tr>
<tr>
<td>Snow (hard packed)</td>
<td>0.2</td>
</tr>
</tbody>
</table>

The model for wheel dynamics is shown in Figure 3.5. The figure shows the acceleration case for which the tractive force and wheel viscous friction force are directed toward the motion. The wheel is rotating in the clockwise motion and slipping against the ground, i.e. $\omega_w > \omega_v$. The slipping produces the tractive force towards right causing the vehicle
to accelerate towards right. In the case of deceleration, the wheel still rotates in the clockwise motion but skids against the ground, i.e. $\omega_w < \omega_r$. The skidding produces the tractive force towards left causing the vehicle to decelerate.

![Figure 3.4 Mu-Lambda Curves for Different Road Conditions](image)

![Figure 3.5 Wheel Dynamics](image)

### 3.3.2 VEHICLE DYNAMICS

![Figure 3.6 Vehicle Dynamics](image)

RESEARCH METHODOLOGY
The vehicle model considered for the system dynamics is shown in Figure 3.6. The parameters in the figure are defined as:

\( N_v \): Wind drag force (function of vehicle velocity)
\( M_v \): Vehicle mass
\( N_w \): Number of driving wheels (during acceleration) or the total number of wheels (during braking).

The linear acceleration of the vehicle is governed by the tractive forces from the wheels and the aerodynamic friction force. The tractive force, \( F_t \), is the average friction force of the driving wheels for acceleration and the average friction force of all wheels for deceleration. The dynamic equation for the vehicle motion is

\[
V = \frac{N_w F_t - F_v}{M_v}
\] (3.13)

The linear acceleration of the vehicle is equal to the difference between the total tractive force available at the tire-road contact and the aerodynamic drag on the vehicle, divided by the mass of the vehicle. The total tractive force is equal to the product of the average friction force, \( F_t \), and the number of relevant wheels, \( N_w \). The aerodynamic drag is a nonlinear function of the vehicle velocity and is highly dependent on weather conditions. It is usually proportional to the square of the vehicle velocity. The distance traveled by the vehicle, \( s \), is computed as an integral of the velocity.

\[
\dot{s} = v
\] (3.14)

### 3.3.3 ENGINE DYNAMICS

The engine is a self-contained power unit which converts the heat energy of the fuel into mechanical energy for moving the vehicle. In the IC (Internal Combustion) engine, an air-fuel mixture is introduced into a closed cylinder where it is compressed and then ignited. The combustion of the fuel causes a rapid rise in cylinder pressure which is converted to useful mechanical energy by the piston and crank-shaft. The four strokes of the IC engine are shown in Figure 3.7. The four-stroke sequence is repeated continually,
with power delivered to the crank-shaft on only one of the four strokes—the combustion stroke.

![Diagram of engine components](image)

**Figure 3.7 Four strokes of an Internal Combustion Engine**

The fuel stored in the fuel tank is fed into the carburetor through a pump. The carburetor mixes the fuel and filtered air, and the vaporized mixture is sent into the inlet manifold. The throttle valve controls the amount of fuel-air mixture entering the inlet manifold, which then directs it into the cylinders. Typically, the air-fuel ratio, by mass, varies in the range 12:1 to 17:1. The throttle valve angle is controlled by the accelerator pedal, which, when at a vertical position, directs the full volume of the air-fuel mixture to produce maximum engine power. Thus the amount of air-fuel mixture that the inlet manifold can hold puts a limit on the torque that the engine would be capable of exerting and in turn the tractive force on the wheel. Figure 3.8 shows a schematic diagram of the basic fuel supply system.
The mass continuity equation for the inlet manifold is

\[ \dot{m}_u = \dot{m}_{u_i} - \dot{m}_{u_o} \quad (3.15) \]

The mass flow rate of air from the carburetor into the inlet manifold through the throttle valve is

\[ \dot{m}_{u_i} = k_i TC(\alpha) f_1(P_M) \quad (3.16) \]

The mass flow rate of air from the inlet manifold into the cylinder is

\[ \dot{m}_{u_o} = f_2(m_u, \omega_e) \quad (3.17) \]

where \( \dot{m}_{u_i} \) and \( \dot{m}_{u_o} \) are the mass flow rate into and out of the inlet manifold respectively, \( k_i \) is a constant related to the geometry of the manifold, \( TC(\alpha) \) is an invertible throttle characteristic, \( f_1(P_M) \) is a non-linear function of the ratio of the pressure in the inlet manifold to the atmospheric pressure and \( f_2(m_u, \omega_e) \) is a function of the mass of air and the rotational speed of the engine.
The two state model of the inlet manifold dynamics has been converted to a simplified one state model for simulation purposes. The dynamics could be described as follows

\[ \dot{m}_w = c_1 TC(\alpha) - c_2 \omega \cdot m_u \]  

(3.18)

where \( c_1 \) and \( c_2 \) are constants used to approximate the model.

### 3.3.4 ACTUATOR DYNAMICS

The dynamics involved in the throttle actuators are very fast. Therefore, it has been assumed that there is no lag between the change in angle of the accelerator pedal and the corresponding angle change in the throttle valve mechanism. But, the modeling of the brake torque has not been done as simplistically as the throttle actuator. Instead, it has been modeled as a first-order lag model described by

\[ \dot{i}_b = t_1 t_b + t_2 t_{hc} \]  

(3.19)

where \( t_b \) is the brake torque, \( i_b \) is the rate of change of the brake torque, \( t_1 \) and \( t_2 \) are constants that depend on the maximum brake torque time constant, and \( t_{hc} \) is the commanded brake torque.

### 3.3.5 COMBINED SYSTEM DYNAMICS

The dynamic equations for the whole system can be written in state variable form by defining convenient state variables. By defining the state variables as

\[ x_1 = V / R_w \]  

(3.20)

\[ x_2 = \omega_w \]  

(3.21)

\[ x_3 = s \]  

(3.22)

\[ x_4 = m_u \]  

(3.23)

\[ x_5 = t_b \]  

(3.24)
the dynamics of the combined system can be represented by

\[
\begin{align*}
\dot{x}_1 &= -f_1(x_1) + b_1 \mu (\lambda) \\
\dot{x}_2 &= -f_2(x_2) - b_2 \mu (\lambda) + b_3 T \\
\dot{x}_3 &= x_1 \\
\dot{x}_4 &= -c_2 x_2 x_4 + c_1 TC(\alpha) \\
\dot{x}_5 &= t_i x_5 + t_z t_{hc}
\end{align*}
\]  

(3.25)  

(3.26)  

(3.27)  

(3.28)  

(3.29)

where

\[
T = T_e - T_b
\]  

(3.30)

\[
\lambda = (x_2 - x_1) / x
\]  

(3.31)

\[
f_1(x_1) = [F_v(R_w x_1)] / (M_v R_w)
\]  

(3.32)

\[
b_1 = N_x N_w / (M_v R_w)
\]  

(3.33)

\[
f_2(x_2) = F_w(x_2) / J_w
\]  

(3.34)

\[
b_2 = R_w N_v / J_w
\]  

(3.35)

\[
b_3 = 1 / J_w
\]  

(3.36)

\[
x = \max(x_1, x_2)
\]  

(3.37)

The state variables in the system are:

- \(x_1\) - vehicle velocity
- \(x_2\) - the wheel velocity
- \(x_3\) - distance
- \(x_4\) - mass of air-fuel mixture in the intake manifold
- \(x_5\) - brake torque

The combined dynamic system can be represented as shown in the Figure 3.9. The control inputs to the model are the throttle input (\(TC(\alpha)\)) and the commanded brake torque (\(t_{hc}\)). During acceleration, the throttle input which is converted to the engine torque is the primary input whereas during deceleration, the commanded brake torque is the primary input.
Figure 3.9 Vehicle/Brake/Road Dynamics: One-Wheel Model

3.3.6 LONGITUDINAL CONTROL

Figure 3.10 Platoon of vehicles

Longitudinal control strategies are necessary to regulate the spacing and velocity of each vehicle in an automated highway system. The longitudinal control algorithm must maintain the spacing policy under normal maneuvers such as acceleration, deceleration, turning, and merging. The controller must also insure good performance over a variety of operating points and external conditions without sacrificing safety or reliability. Parametric studies of the dynamics of the vehicle model have revealed that the vehicle transients cannot be predicted through simple linear models. Therefore, owing to the
highly non-linear nature of the system, the use of a non-linear control strategies would be appropriate for this problem.

### 3.3.6.1 Jerk Control System

This is a third order classical control model adopted in many of the studies conducted in the PATH program[10, 12-15]. The information to the controller is based on the deviation of each vehicle from its assigned position, $\Delta_i$,

$$\Delta_i = x_{i-1} - x_i - L$$  \hspace{1cm} (3.38)

where $x_i$ is the position of ith vehicle and L is the length of the vehicle. For the whole platoon, the control system functions as the diagram shown below.

![Figure 3.11 Jerk Control System](image)

where $w$ is the deviation of the lead vehicle’s velocity at time $t$ from its steady state velocity. $h(s)$ and $g(s)$ are the transfer functions between the leading vehicle to the following vehicle and the ith to the (i+1)th vehicle respectively.

$$h(s) = \frac{s^2 - k_\rho s - k_v}{s^3 + c_\rho s^2 + c_\nu s + c_p}$$

$$g(s) = \frac{(c_a + k_a)s^2 - (c_v + k_v)s + c_p}{s^3 + c_\rho s^2 + c_\nu s + c_p}$$

Transferring this model from $s$ domain to time domain, we obtain the following longitudinal control equation for the jerk that needs to be exerted in each vehicle.
\[ J_i(t) = [(c_a + k_a)a_{i-1}(t) - c_a a_i(t)] + [c_v + k_v)v_{i-1}(t) - c_v v_i(t)] + c_p[x_{i-1}(t) - x_i(t) - L] \]

Since this third order negative feedback control system takes relative acceleration between the vehicles as a part of the control information, it responds very quickly to deviations in relative velocity and distance. Attaining steady state with such a system becomes difficult due to its highly sensitive nature. Therefore, this control system was not included as a part of this study.

3.3.6.2 Acceleration Control System

The controller of vehicles are designed to regulate the acceleration in this type of control system. Each vehicle adjusts its acceleration according to the relative distance and relative velocity with the vehicle directly ahead of it. There is no communication with any other vehicle in the system. The leading vehicle of the platoon is not influenced by the following vehicles and can use any suitable acceleration profile. The controller design of each vehicle has two negative feedback loops, one from its velocity and the other from its position. The information flow in the system is shown in the Figure 3.12.

![Figure 3.12 Acceleration Control System]
3.3.6.3 Sliding Mode Control

The technique of sliding mode control, which is based on Lyapunov stability, has been chosen as the non-linear control strategy for this study. Sliding mode control is a robust feedback control approach which can be used to tackle the parametric and modeling uncertainties of a class of non-linear systems[19]. In this approach, a sliding surface, S, is defined, which upon differentiation yields the control variable. A possible drawback in using the sliding mode technique might be the high-frequency chattering encountered due to discontinuity across the switching surface.

**Background:**

Consider a non-linear system defined as

\[ x^{(n)} = f(x, t) + b(x, t)u(t) \]  \hspace{1cm} (3.39)

where \( x(t) = [x(t) \ x'(t) \ldots x^{(n-1)}(t)] \) is the state vector, \( u \) is the control input and \( x \) is the output state. The superscript \( n \) on \( x(t) \) signifies the order of differentiation.

A time varying surface \( S(t) \) can be defined by equating \( s(t) \) to zero, where

\[ s(t) = (\frac{d}{dt} + \gamma)^{n-1} \tilde{x}(t) \]  \hspace{1cm} (3.40)

\( \gamma \) is a constant and \( \tilde{x}(t) = x(t) - x_d(t) \) is the error in the output state where \( x_d(t) \) is the desired output state. Condition

\[ \frac{1}{2} \frac{d}{dt} (s(t)^2) \leq -\eta |s(t)|, \ \eta > 0 \]  \hspace{1cm} (3.41)

makes the surface \( S(t) \) an invariant set. All trajectories outside \( S(t) \) point towards the surface and the trajectories on the surface remain there.

Consider a second order system

\[ \ddot{x}(t) = f(x, t) + u(t) \]  \hspace{1cm} (3.42)
where \( f(x,t) \) is generally non-linear and is estimated as \( \hat{f}(x,t) \), \( u(t) \) is the control input and \( x(t) \) is the output, desired to follow trajectory \( x_d(t) \). The estimation error on \( f(x,t) \) is assumed to be bounded by some known function \( F = F(x,t) \), so that

\[
\left| \hat{f}(x,t) - f(x,t) \right| \leq F(x,t)
\]  

(3.43)

A sliding variable can be defined according to

\[
s(t) = \left( \frac{d}{dt} + \gamma \right) \tilde{x}(t) + \gamma \tilde{x}(t)
\]  

(3.44)

Define a range for the error of estimation of the gain \( b(x,t) \)

\[
0 \leq b_{\text{min}}(x,t) \leq b(x,t) \leq b_{\text{max}}(x,t)
\]  

(3.45)

The gain and its bounds can be time-varying or state dependent, and is estimated by

\[
\hat{b}(x,t) = \sqrt{b_{\text{min}}(x,t) b_{\text{max}}(x,t)}
\]  

(3.46)

Define

\[
\alpha(x,t) = \sqrt{b_{\text{max}}(x,t) / b_{\text{min}}(x,t)}
\]  

(3.47)

The following control law ensures the sliding condition

\[
u(t) = \hat{b}(x,t)^{-1} \left[ \hat{u}(t) - k(x,t) \text{sgn}(s(t)) \right]
\]  

(3.48)

where

\[
k(x,t) \geq \alpha(x,t)(F(x,t) + \eta ) + (\alpha(x,t) - 1) \left| \hat{u}(t) \right|
\]  

(3.49)
Chattering Reduction:

Define a thin boundary of thickness around the switching surface as

\[ B(t) = \{ s(t) \mid s(t) < \phi \} \]  (3.50)

It is guaranteed that all the trajectories outside the boundary layer are attracted towards the boundary by imposing the following condition

\[ |s(t)| \geq \phi(t) \Rightarrow \frac{d}{2} s(t)^2 \leq (\phi(t) - \eta) |s(t)| \]  (3.51)

To satisfy the above condition, the control law used is

\[ u(t) = \hat{u}(t) = -\tilde{k}(x, t)sat(s(t), \phi(t)) \text{ with } \tilde{k}(x, t) = k(x, t) - \dot{\phi}(t) \]  (3.52)

Inside the boundary, the system trajectories can be expressed in terms of the variable \( s \) as

\[ \dot{s}(t) = -\tilde{k}(x, t)s(t)/\phi(t) - \Delta f(x, t) \]  (3.53)

where \( \Delta f(x, t) = \hat{f}(x, t) - f(x, t) \). Since \( \Delta f(x, t) \) and \( \tilde{k}(x, t) \) are continuous, the above equation becomes

\[ \dot{s}(t) = -\tilde{k}(x_d, t)s(t)/\phi(t) + (-\Delta f(x_d, t) + o(\xi)) \]  (3.54)

where \( o(\xi) \) represents the error terms introduced by replacing \( x(t) \) by \( x_d(t) \) in the first two terms in equation. The variable \( s(t) \) can be viewed as the output of a first order low pass filter with bandwidth \( \gamma \), if we let \( \tilde{k}(x_d, t)/\phi(t) = \gamma \). This filter removes the high frequency chattering to give a smooth \( s(t) \). The bandwidth should be small as compared to high frequency unmodeled dynamics. Using \( \gamma \), the variation of \( \phi(t) \) with time is obtained as follows

\[ \dot{\phi}(t) = -\gamma \phi(t) \tilde{k}(x_d, t) \]  (3.55)

The expression for sliding gain is obtained by using the above equation.

\[ \tilde{k}(x, t) = \gamma \phi(t) + k(x, t) - k(x_d, t) \]  (3.56)
The same filter can also be used for a constant boundary width. In this case, \( \bar{k}(x,t) = k(x,t) \) and the control law is

\[
 u(t) = \hat{u}(t) - k(x,t)msat(a(x,t),s(t),\phi) \tag{3.57}
\]

where

\[
 msat(a(x,t),s(t),\phi) = a(x,t)s(t) / \phi \text{ for } |s(t)| < \phi \tag{3.58}
\]

The dynamics inside the boundary layer becomes

\[
 \dot{s}(t) = -\bar{k}(x_d,t)as(t) / \phi + (-\Delta f(x_d,t) + o(\tilde{e})) \tag{3.59}
\]

Appropriate values can be assigned to \( a(x,t) \) to achieve a desirable bandwidth

\[
 a(x,t) = \frac{\gamma \phi}{k(x_d,t)} \tag{3.60}
\]

**Controller Design:**

The control to maintain a vehicle behind another vehicle at a specified spacing in a longitudinal platoon is obtained by using four sliding surfaces, the first one for tracking the vehicle velocity and spacing, the second for tracking the wheel velocity, the third for tracking the mass of air in the inlet manifold so as to obtain the throttle input and the fourth surface to track the braking torque and in turn obtain the input brake torque. The technique is discussed below.

Let \( s_i \) be the first sliding surface defined as

\[
 s_i = \dot{\epsilon} + c_1 \epsilon \tag{3.61}
\]

where \( \epsilon \) is the spacing error between the vehicle being controlled and the vehicle in front. The spacing error is defined as the difference between the actual distance between the two vehicles and the desired distance between them. When the desired distance
between the two vehicles is a constant, \( \dot{e} \) becomes the velocity error between the vehicle to be controlled and the vehicle in front. Differentiation of Equation (3.61) yields

\[ \dot{s}_1 = \ddot{e} + c_i \dot{e} \]  \hspace{1cm} (3.62)

where \( \dot{e}_1 = x_1 - x_{1_{\text{des}}} \), and \( x_{1_{\text{des}}} \) is the desired \( x_1 \) which is equal to the velocity of the front vehicle for a constant desired spacing. For the first sliding surface, \( \mu(\lambda) \) is the control input. For the second sliding surface, the torque \( T \) is the control input, which effectively controls the input for the first equation also. Using the sliding mode design procedure and introducing \( -k \text{sgn}(s_1) \) term for robustness, we obtain

\[ \dot{s}_1 = \dot{x}_1 - \dot{x}_{1_{\text{des}}} + c_i \dot{e} = -k \text{sgn}(s_1) \]  \hspace{1cm} (3.63)

Substituting \( \dot{x}_1 \) by the estimated quantities in the right hand side of Equation (3.63) yields

\[ -f_1(\dot{x}_1) + \hat{b}_1 \hat{\mu}(\lambda) = \dot{x}_{1_{\text{des}}} - c_i \dot{e} - k \text{sgn}(s_1) \]  \hspace{1cm} (3.64)

From this equation, we obtain the desired adhesion value, \( \hat{\mu}(\lambda) \).

\[ \hat{\mu}(\lambda) = \frac{1}{\hat{b}_1} \left[ \dot{x}_{1_{\text{des}}} + f_1(\dot{x}_1) - c_i \dot{e} - k \text{sgn}(s_1) \right] \]  \hspace{1cm} (3.65)

From the desired adhesion value, the desired slip value \( \lambda_{\text{de}} \) can be calculated using the \( \mu-\lambda \) curve. Since the actual curve is not known, we use a nominal \( \mu-\lambda \) curve. The error in the estimated and actual \( \mu \) value is denoted by \( \mu_e(\lambda) \). The sliding gain of the first surface can overcome this error by utilizing the bounds on \( b_1 \mu_e(\lambda) \). The estimated value of \( f_1(\dot{x}_1) \) in the Equations (3.64) and (3.65) is shown by \( \hat{f}_1(\dot{x}_1) \). Since there is prior knowledge of the acceleration of the lead vehicle, the same is used for \( \dot{x}_{1_{\text{des}}} \). For chattering reduction, the function \( \text{sgn}(.) \) is replaced by \( \text{int}(.) \). Here also, the control law can be made adaptive to reduce the discontinuity across the switching surface. To obtain the desired slip, the value of desired angular wheel velocity \( x_{2_{\text{de}}} \) is calculated, and then a second sliding surface is defined as follows.

\[ s_2 = x_2 - x_{2_{\text{de}}} \]  \hspace{1cm} (3.66)
The control law for the second surface is designed as the first one to obtain the net torque desired at the wheel.

\[ T_{\text{des}} = \frac{1}{b_3} \left[ \dot{x}_{2,\text{des}} + b_2 \mu (\lambda) - k \cdot \text{sgn}(s_2) \right] \quad (3.67) \]

The desired torque should be obtained as a resultant of the torque from the engine onto the wheel and the brake torque.

\[ T_{\text{des}} = T_e - T_b \quad (3.68) \]

The engine torque is derived from the mass of air inside the inlet manifold through the following relationship.

\[ T_e = R \cdot c_3 \cdot m_u \quad (3.69) \]

where \( R \) is the overall gear ratio and \( c_3 \) is a constant.

Since the throttle input and the brake torque input cannot be given at the same time, the following algorithm has been developed to determine the desired mass of air and the desired braking torque.

**Algorithm**

(i) Assume desired braking torque, \( t_{\text{mdes}} = 0 \). Determine the mass desired using

\[ x_4_{des} = m_{ades} = \frac{(T_{\text{des}} - T_b)}{(Rc_3)} \quad (3.70) \]

(ii) Compute the throttle input using a third sliding surface defined by

\[ s_3 = x_4 - x_{4,\text{des}} \quad (3.71) \]

and the control law

\[ TC(\alpha) = \frac{1}{c_1} \left[ \dot{x}_{4,\text{des}} + c_2 x_2 x_4 - k \cdot \text{sgn}(s_4) \right] \quad (3.72) \]

(iii) If \( TC(\alpha) < 0 \), then

\[ x_{4,\text{des}} = m_{ades} = 0. \]

\[ TC(\alpha) = 0. \]
Compute $t_{bdes}$ using the relationship

$$x_{5des} = t_{bdes} = T_e - T_{des}$$ (3.73)

Determine the brake torque input $t_{bc}$ by defining a fourth sliding surface

$$s_4 = x_5 - x_{5des}$$ (3.74)

and using the control law

$$t_{bc} = \frac{1}{t_2} [\dot{x}_{5des} + t_1 t_h - k \cdot \text{sgn}(s_4)]$$ (3.75)

The design of the control law for the second, third and fourth sliding surfaces are the same as that for the first sliding surface.

3.3.7 HEADWAY CONTROL MODELS

Automatic headway control has been an active research topic in the context of Intelligent Highway System studies these years. It seems to be an effective way to solve the congestion and safety problems in the United States transportation systems. The most common headway control systems used are based on platoon control laws or Autonomous Intelligent Cruise Control (AICC). In the platoon control approach, the control applied to a vehicle, in a line or platoon of vehicles, is determined by its state variables with respect to the other vehicles' states in the platoon. The platoons are usually closely spaced, containing two or more vehicles. In AICC, autonomous controls are used by each vehicle to keep a desired spacing behind its predecessor, which is usually a constant for all vehicles at any speed. The following sections describe the three platoon control models that were designed, calibrated and analyzed as a part of the microscopic modeling of the AHS system.
Model I - Traditional Car Following Model

The first model that was used in the analysis is a general car following model that would portray the manual driving process[5]. This model, which is an acceleration control system one, is of the form:

\[ A_i = \frac{K_1(V_{i-1} - V_i)}{(X_{i-1} - X_i)^{K_2}} \]

The acceleration of the following vehicle, \( A_i \), is proportional to the relative velocity with the leading vehicle, and the relative distance is taken as the sensitivity to calibrate acceleration. \( V_i \) and \( X_i \) are the velocity and position of the \( i \)th vehicle, respectively. \( K_1 \) and \( K_2 \) are constant gains.

The behavior of the model can be described as follows: Assume that all the cars in the platoon travel at the same speed initially. There are two possible maneuvers that can disturb the equilibrium of the system.

1. If the disturbance given to the steady state system is in the form of a deceleration to the lead vehicle, then the following vehicle’s controller senses a negative relative velocity. This causes the controller to impart a deceleration to the vehicle, causing it to reduce speed. The two vehicles would still get closer due to the time lag in the system. As the relative distance becomes smaller, the deceleration of the following vehicle increases and produces the desired response to the control system. Thus, this model theoretically works effectively under such a disturbance.

2. On the other hand, when an acceleration is given to the lead vehicle, the controller senses a positive relative velocity and responds with an acceleration to the following vehicle. Since the velocity of the lead vehicle is greater than that of the following vehicle, the relative distance keeps increasing. This decreases the acceleration of the following vehicle forcing the disturbance and the response into an undesirable vicious loop. The system comes back to steady state once the lead vehicle’s acceleration becomes zero or negative. The final result is that the relative distance increases.
proportionately to an increase in velocity, which is typical of a manual driving scenario.

**Model II - Acceleration and Deceleration Control Model**

Based on the analysis of the function and drawbacks of the first model, a safety following distance, $ds$, was introduced into a similar acceleration control system. The equation for the controller of the resulting model is:

$$A_i = K_1 \frac{(V_{i-1} - V_i)}{(X_{i-1} - X_i)} + K_2 \frac{(X_{i-1} - X_i - ds)}{(X_{i-1} - X_i)}$$

In this model, the relative velocity and the relative distance work together to calibrate the acceleration of the following vehicle. The two possible disturbances can be analyzed as follows:

1. When the leading vehicle decelerates, relative velocity and the relative distance decrease. Therefore both the terms in the equation contribute towards countering the disturbance. The smaller the relative distance, the greater the deceleration and the vehicle brakes quickly to adjust to the lead vehicle profile.

2. When the lead vehicle accelerates, both terms are positive and therefore the following vehicle accelerates to catch up with the lead vehicle. The greater the relative distance, the greater the value of the second term, resulting in initiating a very quick response to the control system. The model nevertheless, requires careful calibration to enable appropriate responses to the acceleration and deceleration scenarios.

**Model III - Combination of Model II and Sliding Mode Controller**

The study of the performance of model II (acceleration and deceleration control model) showed that the controller becomes highly unstable if the desired spacing becomes less than 6m for the proposed initial conditions in the microscopic modeling of the system. Analysis and simulation of the sliding mode controller revealed that if the system was
subject to significant initial-error condition, then the controller starts showing highly undesirable oscillations of the different state variables. Given that the two models would perform well under their domains, a combination of the two models covering the whole state variable domain of the system is proposed.

The combined model can be represented as follows:

\[
\text{Following Vehicle Controller} = \begin{cases} 
\text{Model II Controller if } (X_{t,1}-X_{t}) > 6m \\
\text{Sliding Mode Controller if } (X_{t,1}-X_{t}) \leq 6m 
\end{cases}
\]

This combined model, theoretically could produce the following benefits:

- Increase in capacity due to the ability of the sliding mode controller to operate under small spacings.
- Realistic vehicle behavior due to the inclusion of vehicle and engine dynamics in the modeling of the sliding mode controller.
- Robustness due to the capability of the acceleration and deceleration control model to work under severe deviations from steady state values.
4. SOFTWARE MODELING AND DEVELOPMENT

4.1 INTRODUCTION

DYNAVIMTS (DYnamic VIstual MIncro/macroscopic Traffic Simulation Tool) is a simulation package that models and simulates the micro and macroscopic behavior of vehicles over a network. The software is envisioned to integrate the different research efforts at the CTR, Virginia Tech, into one comprehensive ITS software package. In the development of the simulation tool, particular emphasis has been laid on modeling Automated Highway Systems based on the definition of a control architecture that would conform to the framework developed under the PATH program at University of California, Berkeley [10]. The details of various layers can be changed in this software to test what effect various designs of the layers would have on the system.

DYNAVIMTS is used as an integral part of the AHS evaluation framework developed at the Virginia Tech Center for Transportation Research [6]. The progression in AHS development from conceptualization to implementation has four steps according to our approach (Figure 4.1). The first block is the DYNAVIMTS software, which is preceded by mathematical modeling. The second block consists of conducting small scale experiments in the FLASH laboratory [20,21]. Then, hardware tests comprising the third block, are performed on a test site with actual vehicles. The “Smart Highway” being built near Virginia Tech will be a suitable test bed for conducting such experiments using actual vehicles and controlled traffic conditions. The fourth block is the deployment of AHS on conventional highways.
Figure 4.1 AHS Evaluation Framework Proposed by Virginia Tech

These four blocks can be considered as the building blocks of a comprehensive testing and evaluation methodology for AHS. The input can be a hypothesis, a model, or technologies. The evaluation and testing procedure defined by this methodology is not seen as a single feedthrough four-block process but, as having some feedback and feedforward loops depending on the results obtained at each block.

4.2 SOFTWARE SYSTEM ARCHITECTURE

DYNAVIMTS is a multi-purpose AHS simulation software developed using different software / tools running on different computing platforms. Due to the use of these diverse tools running on different platforms, it has a distributed system architecture. One of the advantages of having a distributed architecture is the capability of distributing several time consuming functions of DYNAVIMTS to different computers. The main simulation runs on a Sun Sparc1000 workstation. The output of this simulation is transferred to a Pentium machine which has the animation software developed using Open GL. DYNAVIMTS consists of three basic modules: the Communicator (user interface), the Simulator, and the Animator (output visualization module). The structure of the system architecture is shown in Figure 4.2.
Depending upon the problem being studied, macroscopic and microscopic simulation models will be used. For example, to study the effects of vehicle dynamics to the stability of platoons or the effects of different brake technologies to the safety, microscopic vehicle simulation models can be used. DYNAVIMTS has a full range of vehicle dynamics and control models that can be used for microscopic simulation studies. On the other hand, the effects of highway automation to the overall traffic flow on the automated highway or on the traffic network can be studied using macroscopic traffic flow models that are derived using fluid flow analogy. Moreover, it is possible to simplify the microscopic models used by DYNAVIMTS by ignoring some of the vehicle dynamics that might not be relevant to the specific problem. This way, a series of mesoscopic simulation models can easily be created.

The simulation software is developed using C++, which is an object oriented language. The design of DYNAVIMTS is highly modular and thus flexible. Due to its modularity and object-oriented design, it is extremely easy to add new models in order to expand capabilities of DYNAVIMTS. This feature allows us to test and integrate new models without any major changes to the main body of the program.
4.3 COMPONENTS OF DYNAVIMTS

DYNAVIMTS has three major components: the Communicator, the Simulator, and the Animator. The User-Interface is used to input the simulation parameters and control the simulation. The Simulator is the processing module for performing various algorithms using numerical solvers. The Animator is the visualization module. Following are more details about each of the components:

4.3.1 THE COMMUNICATOR MODULE

The front-end graphical user interface for the simulation package is developed in C using the OSF/Motif libraries and the Xt Intrinsics libraries on a Sun Sparc. The structure of the interface and the underlying routines have been modeled on the basis of the five-layered control architecture developed at PATH, University of California, Berkeley. The sub-module for the network layer design calls ArcInfo’s ArcEdit utilities to construct the network, while all other features invoke C/C++ routines to accomplish their tasks. The GUI also consists of routines to invoke Incident Management software, traffic assignment algorithms and other related programs to form the central part of an integrated ITS simulation software package.

The opening screen of the graphical user interface has push buttons providing access to the following sub modules: Network Layer design, Link Layer Design, Planning Layer Design, Regulation layer Design, and Physical Layer Design (Figure 4.3). The user also has options to compile the simulator source code, run the simulator, quit the application and invoke on-line help facilities.
Network Layer Design

The network layer design module of the GUI uses a Geographical Information System tool, ArcInfo, to model the highway network. Geographical Information Systems (GISs) are basically computer-based systems for storing and manipulating geographic information. A broader definition would refer to GISs as complex combinations of hardware, software, data, organizations and institutional arrangements for collecting, storing, analyzing, and disseminating information about areas of the earth. The main advantages in using a GIS for spatially referenced information are in the volumes of data it can handle, the visual interface provided and in the speed with which the data can be accessed, manipulated, analyzed and displayed.

ArcInfo is a GIS with tools for automation, analysis, display, and management of geographic information. ArcInfo has a layered architecture; the foundation is the data
engine used to access and manage the geographic database. At the next level, it contains a series of major programs that group key functionality into useful operating environments. This application uses the features of the following programs:

- *Arc* - Used for workspace management, data conversion and maintenance and spatial analysis.

- *Arcplot* - Used for map display and query as well as for performing many sophisticated spatial operations such as pathfinding, dynamic segmentation, surface analysis and spatial selection.

- *Arcedit* - ArcInfo's editing system.

- *Info and Tables* - Two programs used to operate on tabular data files and feature attribute tables.

- At a higher level, it contains powerful and flexible command language, providing access to sophisticated geoprocessing tools which operate on the various data sources supported by ArcInfo.

*Starting ArcInfo* - Clicking on the network layer design push button makes a system call to invoke an ArcInfo process, which in turn opens up an ArcEdit session with the appropriate menus for editing and constructing highway networks. Arcedit automatically runs “startup.aml” to open up the menu. The menu items are created from “main.menu”. The Arcedit feature of ArcInfo has been used to graphically construct new networks or edit existing networks. Its sophisticated graphic and editing capabilities provide the tools for necessary for accurate data entry and manipulation. The ArcInfo Macro Language (AML) programs, which are run by clicking on menu items in the Arcedit screen, are used to modify the display on the screen and manipulate link attributes by calling appropriate Arcedit functions. Figure 4.4 shows a typical Arcedit session. The initial database is created automatically by the Info program. The AMLs which call Arcedit functions for modifying link attributes, in effect modify these Info tables.
Figure 4.4 A Typical ArcEdit Session.

Network construction: The “Network” menu item provides the options shown in Figure 4.4. The AMLs invoked by these options are in brackets. This directory also contains the “info” and “network” directories, which hold the info and arc data files of the network respectively.

- New (newnet.aml): This opens up a window, asking the user for the name of the coverage that is to be created. Tics which allow coverage coordinates to be registered to a common coordinate system, are then created for the network.
- Open (opennet.aml): This opens up an existing network.
- Create Attributes (create_att.aml): This creates the attributes or item names for all the links in the new network.
- Build (buildnet.aml): Builds the network and creates the Info tables.
• Save (Arcedit command-save all yes): Saves all changes and writes them into appropriate files.

• SaveAs (saveas.aml): Saves all changes and writes them into a file specified by the user.

• Download (download.aml): This option downloads Info data into an ASCII file for simulation purposes.

Network editing: The “Edit” menu item provides options to

• Add links (add_links.aml): Add links to an open network. The type of links to be added can also be specified from the list of options.

• Delete links (delete_links.aml): Select and Delete links from a network.

• Undo (Arcedit command-oops): Changes made to a network are undone.

Network display functions: The “Pan/Zoom” item provides display utilities to

• Zoom in (aezi.aml): Click on a point on the network to zoom in to a section centered around it.

• Zoom out (aezo.aml): Click on a point on the network to zoom out.

• Pan (aepan.aml): This could be used to pan left and pan right of a selected point.

• Full view (aefullvi.aml): Displays the full view of the network.

Attributes editing: The “Attributes” menu item has options to

• Initialize (init.aml): Initializing link attributes of a new network.

• Query (query.aml): Querying attributes of selected links.

• Modify (modify.aml): This opens up a form (modify_form.menu), presenting an interface for modifying attributes of selected links.

Interface with other software: Hooks are provided for the callbacks in menu items under this option to invoke other software modules.

• Incidents: The “Incidents” menu item has options to call AIMSS (Area-wide Incident Management Support System) to analyze
  1. Different types of accidents
  2. Bottlenecks
3. Failures, etc.

*Analysis*: The “Analysis” menu item provides options to call C routines to implement

- Static traffic assignment algorithms
- Dynamic traffic assignment algorithms

*Quit Network Layer design*: The “Quit” menu item gives options to

- Discard edits and Quit (dis_quit.aml).
- Save Changes and Quit (sav_quit.aml).

*Regulation Layer Design*

Clicking on the “Regulation Layer Design” pushbutton calls the routine “regulation” from the “regulat.c” source file. This routine opens up a form with pushbuttons having the following menu options (Figure 4.5).

![Figure 4.5 Regulation Layer Interface in DYNAVIMTS](image)
Controller Design: The type of controller to be modified can be chosen from a list of options in a cascade menu. The user can choose the type of controller parameters that is to be modified in the controller. At the current stage, a fully functional sliding mode feedback controller has been incorporated into the simulator. Control parameters like gains, boundary layer thickness, etc. could be modified by selecting the appropriate menu option. Figure 4.6 shows the interface for the design of sliding mode controller.

![Controller Design Interface](image)

Figure 4.6 Controller Design Interface in DYNAVIMTS

Platoon Parameters: Clicking on the “platoon parameters” push button calls the routine “PlatoonPar” from the source file “platoon.c”. The user can modify platoon parameters like platoon control strategy based on what information is obtained from which vehicle, and maximum vehicles in a single platoon (Figure 4.7).
Figure 4.7 Input Interface for Platoon Parameters in DYNAVIMTS

Physical Layer Design
The routines in the source file “physical.c” are invoked by clicking on this button. The various parameters related to the components of the vehicle could be modified from this menu option. The components include the
- Engine
- Traction system
- External features like mass, length, etc.
- The other elements in this layer are modified directly in the source code.

Simulation Parameters
Clicking on this button invokes routines from the file “simulat.c”. The user can specify time related aspects of that particular simulation run like
- Total simulation time
- Update interval
• Sensor update interval
• Warm-up period and
• Inter-arrival distribution and mean

Figure 4.8 Input Interface for Simulation Parameters in DYNAVIMTS

Communicator File Structure:
All files associated with the interface are in the “DYNAVIMTS_HOME” directory. The source files for the interface development are in the directory “src”, the include files are “include”, the executable in “bin”, and the ArcInfo aml files are “aml”. Figure 4.9 shows the file structure.
4.3.2 THE SIMULATOR MODULE

The simulator is developed in the C++ programming language using CSIM libraries. The simulator creates objects of the three basic classes - vehicle, link and sensor, based on the input data specified for that particular run. These objects interact among themselves and with the CSIM objects, through their member functions. The simulator also comprises of C routines for implementing control specific algorithms like Rk4 solution for a system of differential equations and also several friction-slip functions describing the road-surface conditions over the network. The input data for the simulator is specified in ASCII files and contains details on the network configuration, controller parameters, platooning strategies, vehicle design, and time parameters. The output of the simulator is also in several ASCII files for animation and display of different types of graphical plots. The
following sections describe the CSIM library, the simulation program architecture and the input and output file structures of the simulator.

Figure 4.10 Simulation Kernel Logic (without CSIM libraries)
The code base for the simulation kernel is also rewritten without the use of CSIM libraries. Though, this reduced the speed of execution and increased the volume of code, the software could now be ported to any platform. All possible events that could occur in the system are stored in an event-array. This array is sorted and most imminent event is scheduled for execution. Figure 4.10 is a representation of the logic used in the implementation of the system kernel.

4.3.2.1 CSIM: A process-oriented simulation package

CSIM is a process oriented discrete event simulation package for use with C or C++ programs. All the necessary operations involved in a simulation have been implemented as a library of routines. CSIM has been used to model a variety of systems, including computer systems, network and protocols, peripheral subsystems, and database systems, to name a few.

*Process-oriented Simulation:* A process-oriented simulation program models a system as a collection of processes interacting with each other and/or competing for resources[3]. The active objects of the system are modeled as processes and the passive objects as the resources. The main program first declares the resources, if there are any and then individual program segments mimic the behavior of the processes in the system. There can be several active processes or active instances of the same process. Each of these processes appear to be executing concurrently in simulated time, even though they are actually executing sequentially on a single processor. The CSIM runtime package ensures that the each instance of every process has its owntime run-time environment and this environment includes local variables and input arguments.

*Object-oriented Structure of CSIM:* The C++/CSIM version, which has been used in the development of DYNAVIMTS, allows C++ programs to use the CSIM library. In the C++/CSIM library, all structures of CSIM are defined as classes. These classes are associated with a set of member variables and member functions, also known as methods.
This structure allows the programmer to build new classes which are derived from the existing C++/CSIM base classes. In addition, there are also a number of procedures and functions, not associated with any of the classes, that have been used to perform the various functions of the simulation program. The structures provided in CSIM that have been used in the simulation are as follows:

- **Processes**-These are the active entities that request service at facilities, wait for events, etc. They interact with all the other CSIM objects and can be in one of four states:
  - Actively computing
  - Ready to begin computing
  - Holding (allowing simulated time to pass)
  - Waiting for an event to happen
- **Events**-These are passive entities that are used to synchronize the activities of the processes in the software. A CSIM event has two states: occurred (OCC) and not occurred (NOT_OCC).
- **Streams**-Streams of random numbers.

The methods of CSIM classes that have been used in the software are:

- **Setting events**: When an event.set() command is executed, all waiting processes are activated and the event is set to the not_occurred state.
- **Waiting for events**: When an event.wait() command is executed, the process is suspended while waiting for the event to occur. If the event is already in the occurred state, the process continues to execute and the event is set to the not_occurred state.

### 4.3.2.2 Model Structure

The basic entities that have been modeled in the system are Vehicle, Link and Sensors (roadside). Each vehicle is an object with a set of attributes defining its state in the system. Vehicles would be the active entities of the system and would therefore be
modeled as instances of the same process. Links have been modeled as the basic units, with which the highway network is constructed. The links on the roadway would consist of sensors which pass on macroscopic information to the roadside controller. The various operations that the entities can execute in the system have been programmed as methods or member functions of the classes.

**Vehicle Class:** The class "Vehicle" is the superclass of all active objects of the system. different types of vehicles like cars, trucks, etc. are derived classes of the superclass. Presently, only cars have been modeled and they inherit all the attributes of the class "Vehicle". The member functions that would be common to all types of vehicles, have been defined in the superclass and those specific to cars have been defined in the derived class.

Attributes have been classified into state variables, link and network layer variables, vehicle dynamic variables, controller variables, and sensor variables. State variables depend on the modeling detail and the method used. For lumped parameter modeling, the number of state variables is finite. At present, there are state variables for wheel angular velocities, vehicle linear velocities, location of the vehicle, braking dynamics and engine dynamics.

Link and network layer variables carry information such as _platoon_no, which is a platoon identification number, no_in_platoon which indicates the vehicle position in the platoon, car_number which holds the vehicle identification number, curr_link_no which is the number of the link in which the vehicle is currently in, the number of the lane in which the vehicle is currently in, the type of surface over which the vehicle is currently moving, and entrance_status representing where the vehicle enters the system (such as a ramp, along the corridor, etc.). Currently, all vehicles are assumed to be entering the system along the corridor. There are external parameters/variables which carry information such as moments of inertia of the wheels, gear ratios, radii of the wheels, etc.
Link Class: Link is the second AHS entity that has been modeled in the simulator. The link has been defined as the section of road between successive ramps on the highway. The class “Link” is the superclass from which subclasses “St_links” (straight links) and “Cur_links” (curved links) have been derived. Currently, the system is assumed to be consisting of only straight links.

Roadside Sensors: The last entity that has been modeled is the road side sensor. These have been modeled to procure macroscopic information during the simulation runs. A link has been assumed to consist of a certain number of roadside sensors that have been equally spaced over the length of the link.

4.3.2.3 Program Process Architecture
The simulator program is started up by clicking on the “simulate” button in the opening screen of the interface. Description of all the processes and their relation with the member functions and other C routines is given below.

Sim: The first process that is started in any CSIM model is the “sim” process. Since, it is called from the CSIM library (which consists of C routines), the sim procedure has to be declared as “extern C”. This process acts as both the initializer and the vehicle generator. The sim process declares all the other processes and initializes the simulator by calling the functions init_links(), init_oper_data(), and init_vehicle_data(). These functions initialize some global variables and member variables and also call member initialization functions to initialize some link data., vehicle data and simulation time data. The “sim” process then starts up the Increment(), SensorIncrement() and several Sensor() processes depending on the initialization data. The VehicleProcesses() are then invoked with a time interval based on the type of distribution and the distribution mean. Since, only one platoon of vehicles is being studied currently, the “sim” process creates one platoon and then waits till the rest of the processes finish execution. Finally, it prints a report of the
simulation and deletes all the instances created in the program by calling the Delete() function.

VehicleProcess: Every VehicleProcess() is associated with a vehicle in the system. It creates an instance of the class "Vehicle" and simulates the behavior of the object for every time step by calling the member functions of the class. It then waits for a time increment before repeating the same process. All vehicles in the system repeat the steps in every simulation time step in the order of their creation.

SensorProcess: This process just executes one function to pass the count of vehicles between the previous update and the current one (called curr_count), to the controller (emulated by storing the reading in an array). The sensors themselves have been created by the link initialization functions. The SensorProcesses like VehicleProcesses then wait for the event done_sensor_update to be set to the OCCURED_STATE by the SensorIncrement() process after every sensor update interval.

Increment and Sensor Increment Processes: These processes are responsible for simulation time to elapse according to the specified update intervals. They also set their corresponding update events to the OCCURED_STATE after all the vehicles or the sensors have finished execution for that time step. Figure 4.11 below shows the flow of control and the program structure of the routines in the simulator.

Events: The events done_update, done_sens_update and done_sensor_incr synchronize the execution of different processes in the system. When scheduling of processes overlaps, the events prioritize them and schedule them according to the order in which they should execute.
4.3.2.4 Input File Structure

The input data for the simulator is stored in a tree of sub directories, inside the parent directory “input_data” (Figure 4.12). The two main sub directories of input_data are “data” and “defaults”. The “data” directory consists of files that the simulator would use for that particular run. The “defaults” directory consists of a default set of input files. The file structure and file formats in both the directories are the same.

The network configuration, which is initially in the Info file format has to be downloaded into the “network” sub directory in an ASCII file format. This data is represented in two
files—"link.dat" and "node.dat". The file "link.dat" contains adjacency data for the network and "node.dat" consists of node-related data. The physical layer data is stored in the sub directory "physical". This directory contains sub directories "engine", "traction", and "external" with data on vehicle's engine parameters, traction parameters and external parameters (mass, length, etc.) respectively. This sub directory holds data on the timing constraints involved in the simulation runs. The file "oper.dat" has data on the time parameters, while "distri.dat" has data on the distribution type for the interarrival times and the distribution mean.

![Figure 4.12 Input File Structure](image)

4.3.2.5 Output File Structure

All output data is written to sub directories inside the "output" directory. The files used to create the graphical plots are written into the "matplot" subdirectory, while the animation data is written into the "animation" sub directory. In the "matplot" sub
directory, one file is created for every car in the system. The files are named “out1” for car1, “out2” for car2 and so on, and are dependent on the type of controller used by the vehicles in the system. The file “graf.dat” in the “animation” sub directory, contains the simulated data required for animation. Figure 4.13 shows the output file structure.

![Output File Structure Diagram]

**Figure 4.13 Output File Structure**

### 4.3.2.6 Simulator File Structure

All files associated with the interface are in the directory `/home/ramnag/dynavimts/simulator`. The source files for the simulator are in the directory “src”, the include files in “include”, the executable in “bin”, and the archived object files are in “lib”.

---

*SOFTWARE MODELING AND DEVELOPMENT*  
83
Figure 4.14 Simulator Source File Structure

4.3.3 THE ANIMATOR MODULE

The output visualization module of the software is started up after a run is executed by the simulator. This feature of the software consists of two components: Graphical plots and Animation. The following sections briefly describe the features offered by these components.

4.3.3.1 MATLAB Graphical Plots

The MATLAB engine is invoked by clicking on any of the options offered by the View Plots push button in the output window of the interface. The engine then runs the appropriate "m" to display the plots. These calls are made using the external interface libraries of MATLAB. Currently, the options offered by this feature include
• Graphical representation of all the state variables associated with each vehicle in the system. These variables would be specific to the type of controller adopted by the vehicle.

• Comparative plots of the velocities, wheel velocities, distances traveled, headways, mass of air in the intake manifold, engine torques, and brake torques of all vehicles in a single platoon.

• Comparative plots of error related variables like sliding variables, position errors, and velocity errors of vehicles in a single platoon.

4.3.3.2 Animation

The animation module of the software is developed in C using OpenGL libraries on a NT workstation. The simulator transfers the required data once this option is invoked and starts up the animator. This section presents some of the features of the animation. These include the different types of display capabilities depicting the visualization of the movement of vehicles in the network and related user interactions. Currently, the user interactions are restricted to key bindings which have been specified along with the description of the features.

Bird’s Eye View

A bird’s eye view, representing a plan view of the whole network, is displayed as the opening screen. This view helps in getting an understanding of the position and motion of all vehicles in the network and the network structure as a whole. Statistical data such as volume and density and their variation along the network could also be represented in this view.

Zooming and Panning

This feature of the interface enables the user to zoom in from the bird’s eye view into any segment of the network. A facility to alter the scale to which the zoom is done, is provided to the user as a part of the interactive features. Since the automated modes
might allow only small spacings and deviations from the desired positions in some cases, the zooming in feature would make these deviations more apparent and greatly enhance the understanding of the behavior of the various longitudinal and lateral spacing control models. The user can also zoom out to the original screen at any point during the animation. The user is also provided with features to pan around the network. This feature can be imagined as viewing the roadway from different positions on a semi-circular dome covering the network. Panning would be analogous to traversing the surface of the dome. Details of the scale of zoom are to be added.

- key “i” for zooming in.
- key “o” for zooming out.
- keys ↑ and ↓ for panning in direction perpendicular to the axis of the road.
- keys → and ← for panning in direction along the axis of the road.
- keys “z” and “Z” for rotation about the z axis (perpendicular to the surface of the road).

Driver's View
The driver's view would be presented as a sub-module within the bird's eye view. This feature would enable the user to zoom into any chosen vehicle in the network. This differs from the zoom feature in that the screen moves with the selected vehicle in this case. The display would be from the driver's perspective and provide a sense of what the driver is experiencing. The driver's view assists in comprehending the aspects of human factors involved in traveling under automated conditions.

- key “c” for driver's view.
- key “b” to get back to bird’s eye view.

Pause Capability
The pause feature of the interface makes it possible for the user to intervene the running of the simulation and temporarily halt its functioning.
• click on left mouse button for pause.
• click again to resume.

*Fast Forward and Rewind Capability*

Sometimes, it might be necessary to study a certain part of the simulation for a second time to get a better grasp of the performance of the models. The rewind option could be used in this case to halt the simulation and rewind the program to any user specified time. The fast forward option would function in the opposite manner to the rewind feature. This could directly get the simulation to a future user specified time and continue the normal run from that point. A requirement for the rewind and fast forward option to function might be that the simulation should have been run at least once earlier, for the necessary data to be already stored in memory.

*Stop Simulation*

This option, which could be invoked at any point during the animation, just stops the process and brings the user back to the opening screen.

• Click on right mouse button for stopping the animation.
5. AHS EVALUATION AND ANALYSIS

The first section of the analysis focuses on how the lead vehicle of a platoon is controlled to follow a given profile under varying road conditions. This is followed by simulation tests on a single platoon of automated vehicles of different platoon sizes, tested under different models and varying lead vehicle profiles. The results of these are used to compute the capacity of the AHS roadway under ideal platooning conditions for pre-defined inter-platoon spacings. Finally, a mesoscopic analysis of platoons of vehicles is conducted to simulate the inter-platoon behavior for a simplified link layer model of the AHS system. The actual flows and average velocities across the section of the link are obtained from the sensor-generated data. A comparison between these observed flows and computed flows is also done as a part of this analysis.

The test bed under consideration is a 5km long free-way segment with sensors embedded at 10m intervals along the roadway. All vehicles in the system are assumed to be automated. For simplicity, only a single lane is modeled in the analysis; the perturbations caused by interplatooning and the lead vehicles of each platoon following a profile, adequately stresses the traffic flow.

5.1 LEAD VEHICLE PROFILE ANALYSIS

The perturbations to the system in both micro and macroscopic analysis is based on the profiles given to the lead vehicle of each platoon on the roadway. In reality, the lead vehicle would only try to follow the profile that is specified for it. This means that the lead vehicle itself is going to have deviations from the actual desired profile, which are in turn going to be amplified and passed on to the rest of the platoon due to the slinky
effect.[12]. Therefore, the lead vehicle's observed profile should be followed by the other vehicles to model a more realistic control system.

The controller that is used for the lead vehicle is a modified version of the one used for the following vehicles, the design of which is described in Chapter 3. The only difference is in the equation of the first sliding surface in the controller design. The modified equation is as follows

\[ s_1 = \dot{\varepsilon} \]

where \( s_1 \) is the first sliding surface and \( \dot{\varepsilon} \) is the difference between the actual velocity and the desired velocity generated by the profile.

In this analysis, an effort has also been made to capture the vehicle-roadway interaction and also to study the vehicle performance on different surface types. Chapter 3 describes in detail, the theory behind the formulation of the differential equations to model this interaction and the associated vehicle behavior.

5.1.1 SCENARIO DESCRIPTION

*Desired Profile:* The desired profile for the lead vehicle to follow is an acceleration-constant velocity-deceleration profile. The vehicle accelerates from an initial velocity of 20m/s to a maximum velocity of 30m/s at 0.5m/s², continues at a constant velocity for 40 seconds, and then decelerates at 0.25m/s². Since, the gains in the controller are not changed for different surfaces, low values of acceleration and deceleration are chosen so that the controller can be designed to work over a wider range of \( \mu-\lambda \) variations.

*Surface Conditions:* The effect of road-surface on the vehicle dynamics is modeled through differences in the \( \mu-\lambda \) relationship. The lead vehicle controller is tested for the
given profile under three different road surface conditions, namely concrete, gravel and snow.

5.1.2 SIMULATION RESULTS

The following graphical plots illustrate the behavior of the lead vehicle operating under the specified profile and surface conditions. The simulations were done for periods of 90 seconds for different road surface conditions.

![Graph](image)

**Figure 5.1 Vehicle velocity vs Time (Lead Vehicle Profile Analysis)**
Figure 5.2 Acceleration vs Time (Lead Vehicle Profile Analysis)

Figure 5.3 Vehicle velocity error vs Time (Lead Vehicle Profile Analysis)
Figure 5.4 Mu (Friction coefficient) vs Time (Lead Vehicle Profile Analysis)

Figure 5.5 Lambda (Wheel slip) vs Time (Lead Vehicle Profile Analysis)
5.1.3 ANALYSIS OF RESULTS

This discussion highlights some of the results that were obtained from the simulation tests conducted to analyze lead vehicle behavior under different road surface conditions.

5.1.3.1 Controller Performance

Since the same set of gains are used by the controller under different road conditions, there is a need to choose an optimal set that would function reasonably well for all three surfaces. Under favorable surface conditions (concrete and gravel), the controller is found to follow the desired profile with an average velocity error (Figure 5.3) of 0.005 m/s and a maximum error of 0.3 m/s. The kinks in the state variable plots (Figures 5.1 - 5.5) represent a change in the desired profile. The maximum time taken to reach steady state after a disturbance in the profile is 4.8 secs, which is when the profile changes from constant velocity to deceleration for gravel.

5.1.3.2 Effect of Road Surfaces

The lead vehicle is found to operate best on a concrete surface. The specified profile could be followed on gravel as well, though the oscillations are found to be more than that on concrete. Under snowy conditions, the initial acceleration profile could be traced by the lead vehicle, but is eventually found to fail when the desired profile changes to constant velocity. This is because, the wheels start slipping due to a sudden increase in braking torque. This is illustrated in Figure 5.5.

5.1.3.3 Deviations from Desired Profile

Though the average error in the velocity is only 0.005 m/s, considerable deviations from the desired profile could be seen when there is an abrupt change in the estimated desired values. The error in those sections are found to be as high as 0.3 m/s for gravel and 0.26 m/s for concrete. The deviations also last for over a range of 2 seconds for concrete (acceleration to constant velocity) to 4.8 seconds for gravel (constant velocity to...
deceleration). These deviations need to be incorporated in the controller of the following vehicle to simulate the actual system. The acceleration of the lead vehicle shows considerable oscillations, especially under snow, as shown in Figure 5.2. These oscillations could also make a significant difference to a following vehicle controller, that uses the lead vehicle’s acceleration as one of its model inputs.

5.2 MICROSCOPIC ANALYSIS

This section presents the results and analysis of experiments carried out on a vehicle platoon control system operating under the different controllers discussed in Chapter 3. The models are first calibrated based on a set of MOEs over a predefined range of operating conditions. These calibrated models are then simulated and analyzed for several scenarios. The capacity of the AHS roadway and other related data are then obtained and tabulated. A summary of the results of the simulation runs is presented at the end.

5.2.1 MODEL CALIBRATION

Numerous test runs over a wide range of scenarios were conducted to come up with an optimal set of gains for the three models. This optimal response is decided on a set of criteria that include:

- System response time or the time to attain steady state
- Oscillations in the response curves
- Mean and maximum values of certain critical state variables like jerk, acceleration, throttle input, braking torque, etc.

The models with their calibrated coefficients are shown below:

Model 1

\[ A_t = \frac{20(V_{t-1} - V_t)}{(X_{t-1} - X_t)^{0.75}} \]
Model II

\[ A_i = 40 \cdot \frac{(V_{i-1} - V_i)}{(X_{i-1} - X_i)} + 20 \cdot \frac{(X_{i-1} - X_i - ds)}{(X_{i-1} - X_i)} \]

The calibrated constants were chosen on a trial and error basis and a prescribed set of rules that kept the response curves within acceptable limits. A limiting steady state time of 25sec and velocity error of ±10m/s were placed to narrow down the range of gains. The velocity error range did not include the initial error condition. The number of oscillations before the response started to behave asymptotically was also limited to three. The limit imposed on the acceleration and deceleration values was 10m/s². This tended to be on the optimistic side, considering the advances in technology that could be possible before the system is actually implemented. Moreover, this limit would be present in the evaluation of all three models and would not affect any relative evaluations of performance, capacities, etc..

**Sliding Mode Controller**

The equation below shows the terms involved, the constants and the state variables for a single sliding surface, the description of which are in Chapter 3. Sliding surface-1 has been chosen as an example.

\[ - \hat{f}_1(x_i) + \hat{b}_1 \mu (\lambda) = \hat{x}_{ide} - c_1\hat{\phi} - k * \text{sgn}(s_1) \]

Tables 5.1 and 5.2 show the values of the different parameters used in the calibration of the sliding mode controller for the lead and the following vehicle. The modeling errors which represent the error in estimation of the different variables were assumed to be about 10%. The boundary layer thicknesses were adjusted so that the sliding variable moves into the layer as soon as possible. The component of the sliding gain, \( \eta_1 \), showed
almost similar behavior with negligible differences over a range of values and did not reach steady state if the values went out of this working range. Therefore, a random value was chosen within this range for the corresponding sliding surfaces. Maximum limiting values of 0.028 kg for the mass of air-fuel mixture and 60Nm for the brake torque were placed based on the limitations in vehicle performance and design.

Table 5.1 Sliding Mode Controller Parameters for Lead Vehicle

<table>
<thead>
<tr>
<th>Sliding mode controller related parameters</th>
<th>Sliding surface 1</th>
<th>Sliding surface 2</th>
<th>Sliding surface 3</th>
<th>Sliding surface 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary layer thickness ($\phi$)</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Modeling error on $f(x,t)$</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Modeling error on $b(x,t) = \alpha$</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Component of sliding gain $k$ ($\eta$)</td>
<td>0.25</td>
<td>10.0</td>
<td>0.2</td>
<td>2.0</td>
</tr>
</tbody>
</table>

Table 5.2 Sliding Mode Controller Parameters for Following Vehicle

<table>
<thead>
<tr>
<th>Sliding mode controller related parameters</th>
<th>Sliding surface 1</th>
<th>Sliding surface 2</th>
<th>Sliding surface 3</th>
<th>Sliding surface 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundary layer thickness ($\phi$)</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Modeling error on $f(x,t)$</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
<td>10%</td>
</tr>
<tr>
<td>Modeling error on $b(x,t) = \alpha$</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Component of sliding gain $k$ ($\eta$)</td>
<td>2.1</td>
<td>20.0</td>
<td>0.2</td>
<td>2.0</td>
</tr>
</tbody>
</table>
5.2.2 SCENARIO DESCRIPTION

The following parameters are varied to present a wide range of scenarios for a comprehensive microscopic analysis of the system.

*Platoon Size:* The number of vehicles in the platoon is varied from a minimum of two to a maximum of four. The limit of four cars represents a threshold after which the control system starts showing oscillations beyond acceptable values.

*Intra-platoon Spacing:* This parameter depends entirely on the behavior of each model in the spontaneous platooning process. The traditional car following model adjusts its spacings depending on the velocity, while the other models reach a steady state spacing that can be specified as an input parameter to the model. The acceleration control model’s desired spacing is fixed at 6m, below which it is found to cause crashes. The sliding mode controller’s desired spacing is 2m.

*Inter-platoon Spacing:* The spacing between platoons is selected according to the size of the platoon. This essentially gives enough distance for a platoon to decelerate and reach steady state in case of failures in the platoon in front. The chosen inter-platoon spacings are shown below in Table 5.3.

<table>
<thead>
<tr>
<th>Platoon Size</th>
<th>Inter-platoon spacing</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>25m</td>
</tr>
<tr>
<td>3</td>
<td>30m</td>
</tr>
<tr>
<td>4</td>
<td>35m</td>
</tr>
</tbody>
</table>

*Velocity Profiles:* Since the behavior of the platoon is tested for entrance, as well as steady state conditions, an acceleration-constant velocity profile is chosen. A nominal
value of acceleration at 0.5m/s² is applied to smooth out the oscillations in the response. The lead vehicle velocity profile is as follows:
Lead vehicle accelerates from an entrance ramp velocity of 20m/s to a maximum of 30m/s at 0.5m/s², and continues at that velocity until it reaches steady state.

5.2.3 SIMULATION RESULTS

Several simulation test runs were conducted to analyze the different concepts in platooning for the scenarios described above. Vehicles were let into the system at a uniform inter-arrival time of 1.5 seconds and allowed to platoon spontaneously. The lead vehicle of the platoon tracked the desired profile through a controller, designed according to the principles described in the section on lead vehicle profile analysis. The roadway surface is assumed to be concrete throughout the test-bed. The controller for the following vehicles was designed based on the three different models, mentioned in the earlier sections.

The runs showed that platoon sizes greater than four produced oscillations in acceleration, that were beyond acceptable values. Therefore, a platoon size of four is used in all further mesoscopic analysis. The other key aspects that were studied from the runs were time taken to reach steady state, minimum attainable spacing, and possible capacity of the AHS based on a single platoon behavior. All these issues are illustrated through the graphical plots and tables presented in the latter half of this section.

The plots of the key state variables, distance, velocity, and acceleration, for a four-car platoon are shown for the three models. The variations in velocity-error and headway, which present an idea of the controller performance are also shown. The plots for the sliding surfaces for the sliding mode controller model give further insight into the model behavior.
5.2.3.1 Graphical plots for Model I

Figure 5.6 Distance Vs Time (Model I)

Figure 5.7 Velocity Vs Time (Model I)
Figure 5.8 Acceleration Vs Time (Model I)

Figure 5.9 Velocity error Vs Time (Model I)
5.2.3.2 *Graphical plots for Model II*

Figure 5.10 *Headway Vs Time (Model I)*

Figure 5.11 *Distance Vs Time (Model II)*
Figure 5.12 Velocity Vs Time (Model II)

Figure 5.13 Acceleration Vs Time (Model II)
Figure 5.14 Velocity error Vs Time (Model II)

Figure 5.15 Headway Vs Time (Model II)
5.2.3.3 Graphical plots for Model III

Figure 5.16 Distance Vs Time (Model III)

Figure 5.17 Velocity Vs Time (Model III)
Figure 5.18 Acceleration Vs Time (Model III)

Figure 5.19 Velocity error Vs Time (Model III)
Figure 5.20 Headway Vs Time (Model III)

Figure 5.21 Sliding surface 1 Vs Time (Model III)
Figure 5.22 Sliding surface 2 Vs Time (Model III)

Figure 5.23 Sliding surface 3 Vs Time (Model III)
5.2.4 ESTIMATION OF SUSTAINABLE FLOW

The simulations runs showed that Model I (a traditional car-following model) does not support the concept of platooning, as described in the context of AHS systems. The headways increase from their initial values as the cars accelerate to higher speeds. Hence, the flows that are computed from using this model, are more representative of a manual highway system and are used for comparison purposes. Model II and model III, on the other hand, exhibited spontaneous platooning, and were used to compute the capacity of an AHS under ideal steady state conditions. The flows at an any section on the AHS is calculated using the formula [22],

Figure 5.24 Sliding surface 4 Vs Time (Model III)
\[ Q = \frac{3600 \cdot v \cdot N}{N(l + d) - d + D} \]

where,

- \( Q \) - Flow on the section
- \( v \) - Velocity on the section
- \( N \) - Average platoon size
- \( l \) - Length of the car
- \( d \) - Intra-platoon spacing
- \( D \) - Inter-platoon spacing

This equation is derived using the traditional relationship between the flow, density and the speed of the vehicles in the system.

\[ Q = k \cdot v \]

The basic assumption is that platoons of vehicles with obtained intra-platoon spacings pack the roadway completely with the specified inter-platoon spacings. This scenario, though not realistic, can be a good measure for comparative evaluation purposes.

The length of a car is assumed to be 4.5m and the inter-platoon spacing is fixed according to Table 5.3. The intra-platoon spacings for Model II and Model III are exact measured values, while Model I uses the average of the headways between different vehicles in the platoon. The time taken to reach steady state is the time taken till both the velocity and the headway for all the vehicles in the platoon reached a constant value with an allowable error of 0.000001. This very low value of allowable error is chosen to obtain conservative estimates, and also to maintain a degree of consistency over this comparative evaluation. All headways and velocities are measured at steady state. The results of the evaluation are tabulated below.
Table 5.4 Calculation of AHS capacity (steady state velocity = 25 m/s)

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Platoon size</th>
<th>Time taken to reach steady state (secs)</th>
<th>Inter-platoon dist (m)</th>
<th>Headway (m)</th>
<th>Velocity (m/s)</th>
<th>Flow (veh/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model I</td>
<td>2</td>
<td>19.90</td>
<td>25</td>
<td>20.71</td>
<td>25</td>
<td>3290.07</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>19.98</td>
<td>30</td>
<td>21.12</td>
<td>25</td>
<td>3149.05</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>20.07</td>
<td>35</td>
<td>21.55</td>
<td>25</td>
<td>3059.90</td>
</tr>
<tr>
<td>Model II</td>
<td>2</td>
<td>29.65</td>
<td>25</td>
<td>6.00</td>
<td>25</td>
<td>4500.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>30.11</td>
<td>30</td>
<td>6.00</td>
<td>25</td>
<td>4864.86</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>30.22</td>
<td>35</td>
<td>6.00</td>
<td>25</td>
<td>5070.42</td>
</tr>
<tr>
<td>Model III</td>
<td>2</td>
<td>37.14</td>
<td>25</td>
<td>2.00</td>
<td>25</td>
<td>5000.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>39.91</td>
<td>30</td>
<td>2.00</td>
<td>25</td>
<td>5684.21</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>41.22</td>
<td>35</td>
<td>2.00</td>
<td>25</td>
<td>6101.69</td>
</tr>
<tr>
<td>Model Type</td>
<td>Platoon size</td>
<td>Time taken to reach steady state (secs)</td>
<td>Inter-platoon dist (m)</td>
<td>Headway (m)</td>
<td>Velocity (m/s)</td>
<td>Flow (veh/hr)</td>
</tr>
<tr>
<td>------------</td>
<td>--------------</td>
<td>----------------------------------------</td>
<td>-----------------------</td>
<td>-------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Model I</td>
<td>2</td>
<td>28.82</td>
<td>25</td>
<td>21.14</td>
<td>30</td>
<td>3917.30</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>28.90</td>
<td>30</td>
<td>21.58</td>
<td>30</td>
<td>3738.74</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>29.01</td>
<td>35</td>
<td>22.02</td>
<td>30</td>
<td>3628.42</td>
</tr>
<tr>
<td>Model II</td>
<td>2</td>
<td>33.59</td>
<td>25</td>
<td>6.00</td>
<td>30</td>
<td>5400.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>34.36</td>
<td>30</td>
<td>6.00</td>
<td>30</td>
<td>5837.83</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>37.38</td>
<td>35</td>
<td>6.00</td>
<td>30</td>
<td>6084.50</td>
</tr>
<tr>
<td>Model III</td>
<td>2</td>
<td>37.23</td>
<td>25</td>
<td>2.00</td>
<td>30</td>
<td>6000.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>39.67</td>
<td>30</td>
<td>2.00</td>
<td>30</td>
<td>6821.05</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>40.68</td>
<td>35</td>
<td>2.00</td>
<td>30</td>
<td>7322.03</td>
</tr>
</tbody>
</table>
Table 5.6 Calculation of AHS capacity (steady state velocity = 35m/s)

<table>
<thead>
<tr>
<th>Model Type</th>
<th>Platoon size</th>
<th>Time taken to reach steady state (secs)</th>
<th>Inter-platoon dist (m)</th>
<th>Headway (m)</th>
<th>Velocity (m/s)</th>
<th>Flow (veh/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model I</td>
<td>2</td>
<td>37.76</td>
<td>25</td>
<td>21.58</td>
<td>35</td>
<td>4534.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>37.87</td>
<td>30</td>
<td>22.05</td>
<td>35</td>
<td>4315.06</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>37.99</td>
<td>35</td>
<td>22.49</td>
<td>35</td>
<td>4183.61</td>
</tr>
<tr>
<td>Model II</td>
<td>2</td>
<td>43.60</td>
<td>25</td>
<td>6.00</td>
<td>35</td>
<td>6300.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>46.61</td>
<td>30</td>
<td>6.00</td>
<td>35</td>
<td>6810.81</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>47.52</td>
<td>35</td>
<td>6.00</td>
<td>35</td>
<td>7098.59</td>
</tr>
<tr>
<td>Model III</td>
<td>2</td>
<td>39.34</td>
<td>25</td>
<td>2.00</td>
<td>35</td>
<td>7000.00</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>41.10</td>
<td>30</td>
<td>2.00</td>
<td>35</td>
<td>7957.89</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>41.65</td>
<td>35</td>
<td>2.00</td>
<td>35</td>
<td>8542.37</td>
</tr>
</tbody>
</table>
5.2.5 ANALYSIS OF RESULTS

The simulation runs with a single platoon of vehicles using the car-following models that were developed, showed certain trends that have been highlighted in this discussion. The comparisons and analysis is done based on certain aspects of AHS that have been evaluated through this micro-simulation.

5.2.5.1 Controller Performance

The performance of the following car’s controller has been evaluated based on the following criteria.

*AHS Capacity:* This is the flow that was calculated on the assumption that the roadway is packed with platoons of vehicles at their corresponding inter-platoon distance. Model I, which tries to simulate a manual system, resulted in the least capacity. Model III provides the highest capacity with an intra-platoon spacing of 2m. A representative scenario with a platoon size of 4 and a steady state velocity of 30m/s is shown in Figure 5.25 below.

![Figure 5.25 Flows Vs Model Type](image-url)
Oscillations in Response: This would be an important consideration in terms of passenger comfort and other human factor aspects of AHS. As the plots for acceleration and sliding variables indicate, Model II results in oscillations of high amplitude and low frequency, while the sliding mode controller acts vice-versa. For the purpose of this study, it is assumed that they are within bearable limits. Another factor in these oscillations is the underlying assumption of a 10% modeling error for the sliding mode controller, without which it could have exacted a much better response curve.

Time Taken to Reach Steady State: The time taken to achieve stability largely depended on the desired intra-platoon spacing. The manual control model was the fastest for all the profiles and Model II with a desired spacing of 6m was second fastest for the first two profiles. The difference in response time between model II and III was much lesser in comparison with model I (Figure 5.26). An interesting observation for the 20-35m/s profile was that, Model III with a desired spacing of 2m reached stability faster than Model II. This could be attributed to the calibration procedure that was adopted, i.e., a different set of gains could have given a better performance with this profile for Model II. But, the criterion for calibration was to ensure that the models performed reasonably well for range of profiles with a single set of gains.

![Figure 5.26 Time for Stability Vs Model Type](image-url)
5.2.5.2 Effect of Platoon Size

Increasing the platoon size is necessary for achieving high flow on an AHS. However, it is equally important to have controllers that can handle the slinky effect due to this increase. Essentially, this would be a trade-off between the time taken for stability, the oscillations and the capacity gained. Model I showed a decrease in flow because of a very high steady state headway which increased with increase in platoon size. Model II and Model III showed an increase in flows, with Model III having a higher marginal gain (Tables 5.4 - 5.6).

5.2.5.3 Manual Vs AHS System

Model I, a traditional car following model, was introduced into the analysis for the purpose of a comparison between flows that can be achieved on a manual system and an AHS system. The flows for Model I have been calculated based on a very high entry rate of vehicles, while the same entry rate could be achieved for an AHS with appropriate ramp metering. Therefore, the estimated flows for the traditional car following model are an over-estimate of realistic values. A more realistic value for current peak flow conditions would be about 2400 vehicles/hr.

5.2.5.4 Entrance Ramp Design

The length of the acceleration lane would depend on the time taken to reach stability at the maximum velocity and also the velocity at which vehicles could merge in the system. A maximum velocity of 35m/s takes only about a second longer to reach steady state than a maximum of 30m/s for model III. This results in a corresponding significant increase in flow in the range of 1000 vehicles/hr. The same comparison for model II showed an increase in flow of about the same 1000 veh/hr for an increase in time of 10 seconds. This would mean a huge increase in the length of the acceleration lane, which might be infeasible for the corresponding capacity gains.
5.2.5.5 *Model Authenticity*

The authenticity of the controller would play an important role in the translation of these simulations to actual field tests. The sliding mode controller used in model III incorporates realistic vehicle dynamics, including vehicle-road surface interactions to model a genuine representation of the actual system. Moreover, modeling errors up to 10% have been incorporated to account for possible misinterpretation of the actual scenario. Model I and II do not include such in-depth vehicle modeling.

5.3 MACROSCOPIC ANALYSIS

In the section on microscopic analysis, several controllers were analyzed and tested for a variety of scenarios. The results showed that the controller designed based on sliding mode and acceleration control model yielded maximum benefits in terms of flows that can be achieved, time for stability and several other MOEs. These tests are extended here to a mesoscopic simulation of platoons of vehicles over a segment of a link. This would simulate the interplatoon behavior in an actual system and would also determine the actual flow in the link in comparison to the flows calculated in the previous section.

5.3.1 SCENARIO DESCRIPTION

Vehicles are generated with an inter-arrival time of 1 second at an entrance ramp to the AHS. They enter at a speed of 15m/s and accelerate at 0.5m/s² to a steady state speed 30m/s. Optimal values of steady state speed of 30m/s and maximum platoon size of 4 are based on the results obtained from the microscopic simulation of a single platoon of vehicles. The following vehicles operate based on the two AHS controllers discussed under the previous section. The maximum possible speed, that can be attained by a vehicle at any time is fixed at 35m/s, based on the constraints to vehicle dynamic variables. The lead vehicles of platoons adjust their profiles to reach a steady state inter-
platoon spacing of 30m. In this scenario, four platoons of four vehicles each try to attain a steady state at 30m/s with an inter-platoon spacing of 30m and intra-platoon spacing based on the controller of the following vehicles. The number of platoons has been restricted to four to avoid propagation of errors due to slinky effect.

5.3.2 SIMULATION RESULTS

Simulation runs were conducted for four platoons of four vehicles each, under the scenarios described above. The platoons were able to sustain the disturbances reasonably well with velocities averaging 32.4m/s at the section of maximum disturbance. The final inter-platoon spacings ranged from 28m to 32m. Figures 5.27 and 5.28 show the flow and velocity variations

---

Figure 5.27 Flow variations along the test-bed
Figure 5.28 Velocity variations along the test-bed

When the following vehicles operated Model III, a steady state flow of 7700veh/hr and peak flows of 8200veh/hr was observed. For the scenario with following vehicles operating under model II, the steady state flow and the peak flow were the same at 6650veh/hr.

5.3.3 ANALYSIS OF RESULTS

5.3.3.1 Velocity Variations

The major fluctuations in the average velocities of the vehicles can be seen to happen over three different phases, as shown in Figure 5.28.
• The initial disparity between the desired and the actual values of state variables causes velocities to shoot up and come down in phase 1. Since both controllers are operating under model II, the average velocity profile is the same for both controllers.

• In phase 2, model III starts using sliding mode control and has a higher average velocity profile to close up intra-platoon spacings to 2m. Inter-platooning also starts happening simultaneously with average velocities going up to 32.4m/s, inorder to catch up with platoons in front.

• The vehicles reach a steady state velocity of 30m/s during phase 3.

5.3.3.2 Flow Variations

The flow variations can be described as occurring in four different phases (Figure 5.27).

• Phase 1: Due to large increases in velocity initially, the flows show a considerable increase for both models.

• Phase 2: The rate of increase in the flow is lesser over this section, since the vehicles have not yet started inter-platooning. The increase is due to further reduction in intra-platoon spacing and is different for different controllers.

• Phase 3: The flows increase rapidly due to inter-platooning and reach a peak when the velocities are at their maximum.

• Phase 4: The systems reaches a steady state at the desired inter and intra-platoon spacings and AHS velocities.

5.3.3.3 Observed Vs Computed Flow

The observed flows in this mesoscopic simulation are found to be slightly more than the flows that were calculated for a similar scenario in the microscopic analysis of a single platoon (Figure 5.29). The differences can be explained by the fact that the final inter-platoon spacings that were achieved under this mesoscopic model are little less than the ones assumed for the microscopic simulation. Though the desired value was 30m, the
observed inter-platoon spacings were close to 28m, resulting in higher values for the flows observed.

![Bar chart showing flows in veh/hr for Models II and III]

Figure 5.29 Comparison between observed and calculated flows

5.3.3.4 Length of Transition Section

The length of the transition section near entrance ramps would depend on the extent of disturbances due to vehicles entering the section. For the purpose of this study, it has been assumed that no more than four platoons would need to inter-platoon and replicas of these four platoons would account for the steady stream of vehicles entering the system. With this assumption, it is observed that a transition section of about 1km would have to be provided at the entrance ramps. This is a very conservative estimate, since the accelerations and the deceleration limits are at a very nominal value of 0.5m/s².
6. CONCLUSIONS AND RECOMMENDATIONS

6.1 CONCLUSIONS

A multi-layer control system architecture has been defined based on the architecture developed at the University of California, Berkeley, to describe the micro and macroscopic behavior of vehicles in a fully automated highway system. The tasks involved in the implementation of this architecture have also been defined to structure the simulation software that has been developed for the analysis and evaluation of the system. The focus of the system modeling has been towards the lower layers of this control system architecture, involving a comprehensive modeling of the regulation and physical layers and a simple, yet realistic modeling of the functionalities of the link layer. The disturbances and possible inter-platoon dynamics or interactions that can occur at the link layer level of the control system architecture have been identified and a link layer model intended to simulate a subset of these disturbances has been developed.

A powertrain model of the vehicle has been developed as a part of the modeling of the regulation and physical layers. The vehicle model identifies five state variables and two input variables. Three of the state variables in this model are associated with one-wheel rotational dynamics and linear vehicle dynamics, and the other two are associated with the engine and actuator dynamics. Parametric studies of the dynamics of the vehicle model have revealed that the vehicle transients cannot be predicted through simple linear models. Therefore, the longitudinal controller design has been based on the non-linear sliding mode control theory. The design involved the use of four different sliding surfaces: the first one for tracking the vehicle velocity and spacing, the second for tracking the wheel velocity, the third for tracking the mass of air in the inlet manifold so as to obtain the throttle input, and the fourth surface to track the braking torque and in
turn obtain the input brake torque. For analysis and comparison purposes, three different headway control models were designed and calibrated. The first was a traditional car following model to simulate the manual driving process, the second was an acceleration control without any vehicle dynamics and, the third was a combination of the acceleration control model and the sliding mode controller. The introduction of vehicle dynamics into the modeling of vehicle behavior lent authenticity to the sliding mode controller design and performance.

A simulation software has been specifically developed for evaluating different aspects of AHS including architectures, configurations and controller strategies. This software, DYNAMICS, is designed to address microscopic as well as macroscopic traffic behavior in the multi-layer AHS architecture. The software models the network layer using Arc/Info, a geographic information system tool, lending it enormous versatility in terms of incorporating incident management systems, fault tolerance and other network level applications. The micro-simulation kernel, modeling the lower layers of the architecture, uses the C/C++ programming language. The modularity of the software in modeling the behavior of the vehicle and its basic elements individually, makes it easy for the user to modify, discard, or replace a module. The animator module that has been developed in OpenGL, provides a 3-D dynamic visualization environment with interactive computer graphics.

The analysis and evaluation portion of this research study involved a comprehensive microscopic analysis of the behavior of a single platoon of automated vehicles of different platoon sizes, tested under different models and varying lead vehicle profiles. A mesoscopic analysis of platoons of vehicles was also conducted to simulate the inter-platoon behavior for a simplified link layer model of the AHS system. The analysis of the generation of lead vehicle profiles showed a definite need for incorporating a controller for the lead vehicle, rather than just using the desired profile for the following vehicle’s controller. The study of the behavior of vehicles under different road conditions
showed enormous differences in model behavior necessitating the need for a study of the effect of abrupt changes in road surface conditions.

The micro-simulation of single platoon of vehicles showed that intra-platoon spacings of upto 2m could be obtained using the sliding mode controller, sustaining traffic volumes of 8000 veh/hr on the roadway. The volumes were calculated under ideal conditions in which the system is packed with platoons of the specified size and inter-platoon spacings. The link layer model that was developed for the mesoscopic simulation was intended to simulate a subset of the disturbances that would actually occur in a full-fledged AHS system and also result in a packed AHS system as assumed in the micro-simulation. The controllers were found to withstand the disturbances introduced by the link layer controller. The volumes observed were a little more than the calculated values, due to the approximations involved in the switching of operating modes by the link layer controller.

Each headway control model is unique to a set of operating conditions. This implies that each operation might warrant an alteration in the control model itself. The sliding mode controller or a third order jerk control model would work very well under short intra-platoon spacings, but do not suit scenarios where vehicles are at large headways. Therefore, model III was proposed as a combination of two different controllers. The excellent results shown by this model opens up the possibility of employing different control laws for different operating regimes and maneuvers. This will greatly improve the ability of the system to handle disturbances generated due to any operation.

6.2 FUTURE RESEARCH

The link layer model that has been developed, accommodates only inter-platooning. This model needs to be modified to incorporate deplatooning to study the exit and lane-change of vehicles in the system. The model should also be improved from its current open-loop
control to a closed-loop feedback control system. The link layer model also currently attempts to pack the AHS system to evaluate maximum capacity. An interesting variation would be to optimally space the vehicles depending on the current volume in the system. The single-lane highway should also be converted to a multi-lane system with a number of exits and entrances to provide a more complete modeling of the AHS.

Further work needs to be done on the sliding mode controller to make it function under a larger range of error conditions. A complete vehicle model would require the modeling of lateral movements and the introduction of appropriate state variables into the dynamics of the system. Lateral control laws also need to be incorporated into the vehicle controller. The parameters and gains in a headway control model influence the behavior of the model only over a certain range of operating conditions. The position, velocity, acceleration of the vehicle in the platoon, and vehicle maneuver define the operating conditions under which the model behaves. It has not been possible to use a model with fixed coefficients for all values of the above variables and still get optimal responses. Hence, there might be a need to generate a matrix of model parameters to obtain ideal responses from the vehicle. A set of n-dimensional matrices, corresponding to the different operations or maneuvers would describe the vehicle behavior in the AHS. The state variables relevant to the functioning of the model would correspond to the axes.

The callbacks in the communicator code currently invokes separate processes. This tends to slow down the software due of the underlying system calls. Instead, the whole application needs to be run as one process with a multi-threaded architecture. Introduction of distributed computing would greatly reduce the load on the processor and thus enable the user to simulate a large number of vehicles over a shorter period of real time. The animation is currently done on a Windows NT platform after the simulator generates all the data. This code needs to be ported to the SUN Sparc and integrated into the same process so that the animation could be done in real-time.
REFERENCES


Ramakrishnan Nagarajan was born on the 24th of March, 1972 in Madras, India. He completed his schooling at D.A.V. Higher Secondary School in Madras and then received his B.Tech. degree in Civil Engineering from the Indian Institute of Technology, Madras, in 1993. He graduated from the Virginia Polytechnic Institute and State University with a Master of Science in Civil Engineering, in 1996. He is currently working as a software consultant at IBM, RTP.