

# A Microscopic Traffic Simulation Model for IVHS Applications

by  
Qi Yang

Bachelor of Science in Geography  
Peking University, Beijing, China  
(1983)

Master of Science in Geography  
Peking University, Beijing, China  
(1985)

Submitted to the Department of Civil and Environmental Engineering  
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE  
in Transportation  
at the  
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 1993

© Massachusetts Institute of Technology 1993

All right reserved

Author \_\_\_\_\_  
Department of Civil and Environmental Engineering  
August 10, 1993

Certified by \_\_\_\_\_  
Harris N. Koutsopoulos  
Professor of Civil and Environmental Engineering  
Thesis Supervisor

Accepted by \_\_\_\_\_  
Eduardo Kausel  
Chairman, Departmental Committee on Graduate Studies

ARCHIVES

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

OCT 14 1993

LIBRARIES

# **A Microscopic Traffic Simulation Model for IVHS Applications**

Qi Yang

Submitted to the Department of Civil Engineering on August 12, 1993,  
in partial fulfillment of the requirements  
for the degree of Master of Science in Transportation

## **Abstract**

Intelligent Vehicle Highway Systems (IVHS) are currently being developed as a potential solution to the congestion problem. These systems are expected to improve significantly the utilization of existing road networks using advanced traffic control and communication technologies. However, interactions among different components of the systems and their impact on traffic characteristics are very complex. The purpose of this thesis is to develop a Microscopic Traffic SIMulator (MITSIM) that can be used to model traffic operations in the presence of advanced traffic control and surveillance systems. The results from the model will help transportation decision makers evaluate competing alternatives and designs based on safety, mobility, congestion relief, environmental protection, and fuel conservation considerations.

MITSIM uses car-following and lane changing algorithms to simulate vehicle movements in integrated traffic networks with a detailed representation of road networks and traffic control and surveillance devices. The model also simulates incidents and toll booth operations.

Time-dependent travel demand and responsive traffic controls are inputs to MITSIM. The output from the model includes: (1) the measures of effectiveness (MOE's) that are required in the system evaluation, such as speed, travel times, queues, delay, event response time, diverted traffic, etc.; (2) various data collected by surveillance devices installed in the simulated network.

The MITSIM model has been implemented in C++ using object-oriented design. It uses graphics to dynamically display aggregate traffic characteristics such as speed and density, and to animate individual vehicles' movements for selected parts of the road network.

***Thesis Supervisor:*** Haris N. Koutsopoulos

***Title:*** Assistant Professor of Civil and Environmental Engineering

# Acknowledgments

I acknowledge with deep gratitude the valuable guidance received for this work from my advisor, Professor Haris N. Koutsopoulos. His patience, encouragement and help were invaluable and will always be appreciated. I am also grateful to Professor Moshe Ben-Akiva and Professor David Bernstein for their guidance and encouragement. Their support and criticism contributed in a large measure to the success of this research.

Special thanks go to Emily Parknay, she proofread an early draft of this thesis and provided valuable comments. I would also like to thank Haiping Xu, Ibrahim El Sanhoury, Anil Mukundan, Andras Farkas, Yuting Kuo, Jay Lee, Charles Y. Huang, and J.V. Nageswar Rao for their help.

Thanks also go to my fellow students and staff of the Transportation Systems Division and the Center for Transportation Studies. Their help and friendship made my past two years at MIT enjoyable and memorable.

Instead of starting from scratch, I have used many ideas, directly or indirectly, from earlier work in the area of traffic simulation. I would like to thank many individuals whose work has been used in this research.

The research was supported by the Draper Laboratory, the John A. Volpe National Transportation Systems Center, and MIT. Their financial and institutional support is gratefully acknowledged.

Finally I would like to thank my wife, Xiangming Li, for her love, understanding, encouragement and unconditional support.

# Contents

<b>Abstract</b> .....	2
<b>Acknowledgments</b> .....	3
<b>Chapter 1 Introduction</b> .....	12
1.1 Intelligent Vehicle Highway Systems (IVHS) .....	13
1.2 Evaluation of IVHS Applications .....	14
1.2.1 Basic Components of IVHS Applications .....	15
1.2.2 An Off-line Traffic Simulation Framework .....	15
1.3 Thesis Outline .....	17
<b>Chapter 2 Review of Existing Models</b> .....	18
2.1 Existing Traffic Simulation Models.....	18
2.2 Requirements for Evaluation of IVHS Applications and Limitations of Existing Models.....	21
2.3 Conclusion.....	23
<b>Chapter 3 Overview of the Simulation Framework and Structure of the Model</b> .....	25
3.1 The Overall Framework.....	25
3.2 Traffic Simulation Logic.....	27
3.3 Implementation Considerations .....	29
<b>Chapter 4 Elements of the Traffic Network</b> .....	30
4.1 Road Network.....	30
4.1.1 Nodes .....	30
4.1.2 Links and Segments .....	31
4.1.3 Lanes .....	35
4.1.4 Toll Facilities.....	37
4.2 Traffic Control Devices .....	40
4.2.1 Traffic Signals and Ramp Meters.....	41
4.2.2 Lane Use Signs (LUS) .....	43
4.2.3 Variable Speed Limit Sign (VSLS).....	45
4.3 Simulation of the Surveillance System.....	46
4.3.1 Point Data Sensors .....	48
Flow .....	49

	Speed .....	49
	Occupancy .....	50
	Headway .....	52
	Over-height Attribute .....	52
4.3.2	Point to Point Data Sensors .....	53
4.3.3	Area Data Sensors .....	53
4.3.4	Cellular Phone Call-in and Service Patrol .....	54
4.3.5	Other Sensors .....	54
<b>Chapter 5</b>	<b>Vehicle Movements .....</b>	<b>55</b>
5.1	Car-following Logic .....	55
5.1.1	Notations and Definitions .....	56
5.1.2	Review of Representative Car-following Models .....	58
	General Car-following Model .....	58
	CARSIM's Car-following Model .....	60
5.1.3	Car-following Model in MITSIM .....	62
	Three Regimes of Car-following Relationships .....	63
	Free Flowing Acceleration Rate .....	64
	Emergency Deceleration Rate .....	67
	Car-following Acceleration Rate .....	68
5.1.4	Response to Traffic Signals .....	68
5.1.5	Merging .....	69
	Merging Range .....	70
	Merging Acceleration Rate .....	71
5.1.6	Courtesy Yielding .....	72
5.1.7	Select a Proper Acceleration Rate .....	74
5.1.8	Start-up Delay .....	74
5.1.9	Update Speed and Position .....	74
5.2	Lane-changing Logic .....	75
5.2.1	Lane Change Status Variable .....	76
5.2.2	Determination of Mandatory Lane Changing .....	77
5.2.3	Determination of Discretionary Lane Changing .....	79
5.2.4	Target Lane .....	80
5.2.5	Acceptable Gaps in Lane Changing .....	80
5.3	Toll Booth Operations .....	81
5.3.1	Lane Selection .....	82
5.3.2	Delay .....	82
5.4	Incidents .....	83
<b>Chapter 6</b>	<b>Representation of Travel Demand .....</b>	<b>86</b>

6.1	Origin-Destination Matrices .....	86
6.2	Vehicle Type.....	87
6.3	Departure Time.....	88
6.4	Route Choice .....	89
6.4.1	Path Enumeration .....	89
6.4.2	Shortest Path .....	90
6.4.3	Stochastic and Adaptive Routing .....	91
<b>Chapter 7</b>	<b>Output and User Interface.....</b>	<b>93</b>
7.1	Collecting Data in Simulation.....	93
7.1.1	Standard Output .....	93
7.1.2	Customized Output .....	94
7.2	Monitoring the Simulation Process .....	94
7.2.1	PC Version .....	95
7.2.2	Workstation Version .....	97
<b>Chapter 8</b>	<b>A Case Study .....</b>	<b>100</b>
8.1	The Network .....	100
8.1.1	Data Files for Road Network .....	101
8.1.2	Data File for the Surveillance System.....	104
8.2	A Debugging Scenario .....	106
8.3	A Comparison Scenario.....	108
<b>Chapter 9</b>	<b>Conclusion.....</b>	<b>112</b>
9.1	Next Steps.....	112
9.1.1	Implementation of Measures of Effectiveness.....	112
9.1.2	Communications with the Control and Surveillance System Models.....	112
9.1.3	Incorporation of Urban Streets.....	113
9.1.4	Emergency Vehicle Operations.....	113
9.1.5	Reversible Lanes.....	114
9.1.6	Calibration and Validation.....	114
9.2	Future Research .....	115
9.2.1	Control Strategy Model .....	115
9.2.2	Computational Considerations .....	115
9.3	Conclusion.....	117
<b>Appendix</b>	<b>.....</b>	<b>118</b>
	NETSIM.....	118
	INTRAS/FRESIM.....	121

TRAFLO .....	123
NETFLO .....	125
FREFLO .....	126
CORQ.....	129
KRONOS .....	130
CONTRAM .....	131
DYNASMART .....	135
THOREAU .....	137
INTEGRATION .....	139
DYNAMIC TRAFFIC MODEL.....	142
<b>Bibliography.....</b>	<b>146</b>

## List of Figures

Figure 1.1 The Simulation Framework .....	16
Figure 3.1 A Simulation Framework for IVHS Applications.....	26
Figure 3.2 Flow Chart of the Traffic Simulation Model .....	28
Figure 4.1 Definition of Nodes .....	31
Figure 4.2 A Hypothetical Freeway Network .....	32
Figure 4.3 Representation of the Road Network.....	32
Figure 4.4 Coordinate System for Geometric Data .....	35
Figure 4.5 Lane Alignment and Pointers .....	35
Figure 4.6 Toll Plaza .....	38
Figure 4.7 Time Dependent Toll Fare .....	40
Figure 4.8 Linked Lists of Control Devices .....	41
Figure 4.9 A 2-phase Traffic Signal and Corresponding Signals Status .....	43
Figure 4.10 Implementation of Surveillance System .....	47
Figure 4.11 Variables in Point Data Calculation .....	48
Figure 4.12 Calculation of Vehicle's Presence Time in Detection Zone .....	51
Figure 4.13 Headway Determination.....	52
Figure 5.1 Notations and Definitions in Car-Following Models .....	56
Figure 5.2 Buffer Space between Following and Lead Vehicles.....	67
Figure 5.3 Merging.....	70
Figure 5.4 Acceptable Merging Time Headway .....	72
Figure 5.5 Courtesy Yielding .....	73
Figure 5.6 Mandatory and Discretionary Lane Changes .....	76
Figure 5.7 Probability of Starting Mandatory Lane Changing .....	78
Figure 5.8 Lead and Lag Gaps in Lane Changing .....	81
Figure 5.9 Lane Selection at a Toll Booth.....	82
Figure 5.10 Position of an Incident.....	84
Figure 6.1 Alternative Paths in a Small Network .....	89
Figure 6.2 Route Choice.....	91



Figure 7.1 Layout of Network and Animation Windows....	98
Figure 8.1 Schematic of San Diego Freeway in Los Angeles .....	100
Figure 8.2 Node and Link Codes of the Network.....	101
Figure 8.3 Distribution of Measured Speed.....	110
Figure 8.4 Comparison with Field Data.....	111

## List of Tables

Table 2.1 Representative Traffic Simulation Models .....	19
Table 4.1 Link Information.....	33
Table 4.2 Segment Information.....	33
Table 4.3 Geometric Data of a Segment .....	35
Table 4.4 Lane Alignment Information.....	36
Table 4.5 Data for the Toll Booth 1 .....	38
Table 4.6 Toll Fares at Toll Booth 1 .....	39
Table 4.7 Time Table for Toll Fares .....	39
Table 4.8 Distribution of Desired Speed Ratio .....	45
Table 5.1 General Car-following Model and Steady-state Macroscopic Flow Equations .....	60
Table 5.2 Car-following Lower and Upper Bounds .....	64
Table 5.3 Maximum Acceleration Rates for High Performance Passenger Cars .....	65
Table 5.4 Maximum Acceleration Rates for Low Performance Passenger Cars .....	65
Table 5.5 Maximum Acceleration Rates for Buses and Heavy Single-Unit Trucks .....	66
Table 5.6 Maximum Acceleration Rates for Trailer Trucks.....	66
Table 5.7 Normal Deceleration Rates for Passenger Cars .....	67
Table 5.8 Service Rates at Toll Booth.....	83
Table 6.1 O-D Table .....	87
Table 6.2 Paths Associated with the O-D Pair from Node 1 to 4.....	90
Table 8.1 Link Data .....	101
Table 8.2 Segment Data .....	102
Table 8.3 Lane Data.....	102
Table 8.4 Geometric Data .....	104
Table 8.5 Detector Data .....	105
Table 8.6 Five Generated Vehicles.....	106

Table 8.7 Travel Time and Space Mean Speed .....	106
Table 8.8 Results from Point Data Sensors .....	107
Table 8.9 Results from VRC Readers and Over-Height Detectors .....	108
Table 8.10 Travel Demand .....	108
Table 8.11 Comparison with Field Data and INTRAS Generated Data.....	109

# Chapter 1

## Introduction

Traffic congestion has become a major problem in most urban areas around the world, with negative impacts on people and freight mobility. The large increase in traffic volumes is expected to have a detrimental effect on safety and pollution levels and to cause economic losses due to delays and accidents. Moreover, it is not expected that the construction of new road facilities, if any, will be able to alleviate the anticipated congestion and other problems. Therefore, there is a need for innovative approaches for more effective management of existing resources. During the coming decades, efficient operation of existing road networks is expected to be achieved through dynamic traffic management schemes that make use of advanced technologies.

Within this context, Intelligent Vehicle Highway Systems (IVHS) are currently being developed as a potential solution to the congestion problem. These systems, based on the linking of road infrastructure, vehicles, and drivers using advanced communication technologies, computers, information display equipment, and traffic control systems, are expected to significantly improve the utilization of existing road networks.

However, a large number of diverse and complex interactions between the different components and traffic characteristics must be understood before the system can be implemented. In general, in evaluating alternative transportation systems and strategies all relevant impacts should be considered. In this evaluation, either for an existing system or a system under development, it is important to understand how an integrated system performs in terms of alleviation of congestion and pollution, reduction of travel time and monetary costs, and who will benefit from a proposed system. The purpose of this thesis is to:

- Develop a Microscopic Traffic Simulator (MITSIM) which can be used to model traffic operations in an integrated control system;
- Demonstrate the use of computer simulation to produce information required for the evaluation of IVHS applications.

## 1.1 Intelligent Vehicle Highway Systems (IVHS)

IVHS is an umbrella term for a wide range of technologies including electronics, computer hardware and software, control and communications. These technologies find application in five functional areas <sup>[41]</sup>:

- Advanced Traffic Management Systems (ATMS)
- Advanced Traveler Information Systems (ATIS)
- Advanced Vehicle Control Systems (AVCS)
- Commercial Vehicle Operations (CVO)
- Advanced Public Transportation Systems (APTS)

**Advanced Traffic Management Systems (ATMS):** ATMS systems collect, utilize and disseminate real-time traffic data on conditions of the road network. Dynamic traffic control systems respond to changing traffic conditions across different jurisdictions and type of roads. In more sophisticated implementations, ATMS predict traffic congestion and coordinate traffic control and routing instructions to vehicles over wide areas in order to maximize the efficiency of the network, or maintain priorities for high-occupancy vehicles. These predictions are based on special dynamic traffic models and simulations.

**Advanced Traveler Information Systems (ATIS):** ATIS systems provide a variety of information that assist travelers to their destination following the best alternative. This information may include: optimal routes, turning commands, parking conditions, locations of accidents, weather and road conditions, etc. This information will be received with in-vehicle systems or road-side signs, or by cellular phones or computers.

**Advanced Vehicle Control Systems (AVCS):** AVCS systems enhance the driver's control of the vehicle and make travel safer. They include a broad range of concepts that will become operational at different time scales. Developments include intelligent cruise control systems, collision warning systems, automatic

steer-away systems, radar-based platoon systems, etc. These technologies will not only enhance safety but also improve capacity by maintaining smaller headways.

**Commercial Vehicle Operations (CVO):** Operators of trucks, buses, vans, taxis and emergency vehicles will likely be early adopters of IVHS technologies. These vehicles can be equipped with Automatic Vehicle Location (AVL) and Automatic Vehicle Identification (AVI) systems, and benefit from automatic toll collection systems.

**Advanced Public Transportation Systems (APTS):** APTS systems use the technologies that comprise ATIS, ATMS, and CVO to improve operations of high occupancy vehicles (HOV's), including transit buses and van- and car-pools. Through ATIS, APTS systems will inform travelers about alternative schedules, and anticipate travel times and costs on different modes of transportation.

The above discussion indicates that IVHS provides a wide range of options to transportation planners and operators. Flexible tools for the evaluation of various strategies, and models for the operations of the advanced systems will become increasingly important in coming years. Transportation decision makers will be confronted with difficult choices among competing alternatives and designs which should be evaluated based on safety, mobility, congestion relief, clean air, and energy conservation considerations. Field testing, although it plays an essential role in demonstrating system concepts, is costly and has many limitations in accurately evaluating system benefits. Consequently, systems, policies and strategies must be evaluated prior to implementation. This evaluation should be supported by appropriate models.

## 1.2 Evaluation of IVHS Applications

MITSIM is designed as a tool to simulate traffic operations under various IVHS strategies. Before proceeding to describe the model in detail, we identify the basic components that need to be simulated, and the requirements the model should satisfy in order to be useful. We then examine the most appropriate approach that meets these requirements.

### 1.2.1 Basic Components of IVHS Applications

The development of MITSIM is motivated by the evaluation and refinement of Integrated Project Control System (IPCS) of the Central Artery/Tunnel Project in Boston. In this project, ATIS, ATMS and some other IVHS technologies will be implemented. The network elements that need to be simulated and evaluated include:

**Road Network:** The simulated road network consists of multiple lane highways and ramps, toll booths, tunnels, and eventually arteries and urban streets.

**Vehicles:** Individual vehicles may have different characteristics such as speed, acceleration, emission, traffic information availability, etc.

**Surveillance System:** Surveillance systems implemented in the IPCS consist of (1) a wide range of sensors and detectors including loop detectors, vehicle to roadside communications (VRC) devices, closed circuit television (CCTV) and operators, overheight detectors, vehicle type (length classes) detectors, carbon monoxide (CO) sensors, etc.; (2) external agency information such as police voice reports, cellular phone call-ins; and (3) historical information stored in the control center.

**Control System:** The control system may include static message signs such as direction signs, exit signs; regulatory devices such as traffic signals, speed limit signs, ramp metering, lane use signs (LUS), mainline metering, blank-out signs (BOS), etc.; and advisory devices such as variable message signs (VMS), highway advisory radio, in-vehicle systems, etc.

### 1.2.2 An Off-line Traffic Simulation Framework

In order to incorporate all the above elements, a model system has been developed. This system consists of three main elements: (1) travel demand model, (2) control strategy model, and (3) traffic simulation model (*see* Figure 1.1).

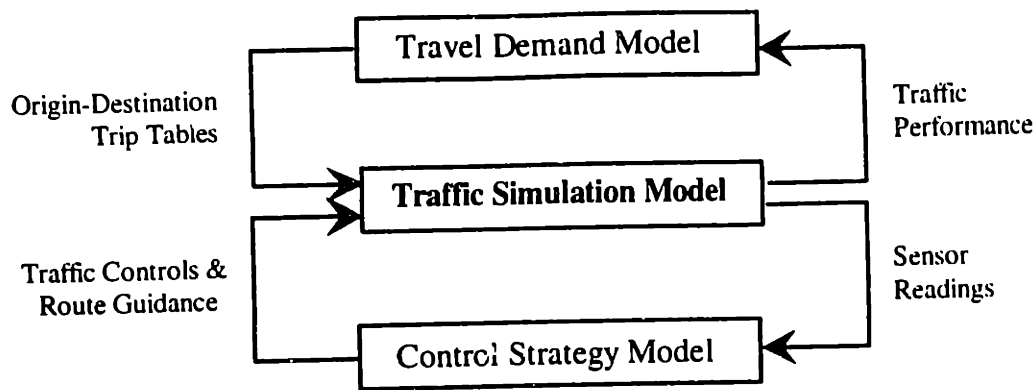


Figure 1.1 The Simulation Framework

**Travel Demand Model:** The travel demand model takes as input the historical information of origin/destination trip matrix, travel time/cost, and "real time" link traffic counts to estimate time dependent travel demand between each pair of origins and destinations and predict drivers' pre-trip route choices.

**Control Strategy Model:** The control strategy model is designed to simulate both the information collection and the control mechanism in traffic networks. It consists of three modules:

- **Surveillance System Module:** The surveillance system module simulates traffic sensors (detectors in the pavement, video cameras, beacons and equipped vehicles). Information is collected continuously, rapidly processed and made available to other elements of the system.
- **Traffic Prediction Module:** The traffic prediction module has the responsibility of providing the information that is needed to generate control strategies.
- **Control and Routing Module:** The control and routing module generates traffic control and route guidance strategies in response to information provided by the surveillance system and the traffic prediction modules.

**Traffic Simulation Model:** The traffic simulation model takes travel demand (represented by time dependent origin/destination trip tables), traffic controls, and route guidance as input, and simulates vehicle movements in the network. The outputs of the traffic simulation model include the measures of effectiveness (MOE's) that are required in the system evaluation, such as density, speed, travel times, queues, delay, event response time, diverted traffic, etc. These measures



and their evolution over time can be provided at the single station, lane, segment, link, O-D pair and entire network level.

These models are coordinated with each other and construct the basic architecture of the a comprehensive traffic simulation system under development at MIT. The focus of this thesis is on the last element of the traffic simulation model.

### **1.3 Thesis Outline**

Chapter 2 provides a literature review of several existing traffic simulation models. Detailed considerations and the implementation of the MITSIM model are described in remaining chapters. Chapter 3 presents a framework and overview of the simulation model. Chapter 4 describes the elements of the traffic network (road network, traffic control and surveillance systems) and their representation in the traffic simulation model. Chapter 5 presents the algorithms that are used to move vehicles in the simulated network. Chapter 6 describes how time-dependent travel demand is represented and simulated vehicles are generated. Chapter 7 discusses the user interfaces. Chapter 8 presents a numeric example for a small network, and a comparison of the results with field data and INTRAS-generated results. Finally, Chapter 9 suggests further work for model development and validation.

# Chapter 2

## Review of Existing Models

There exists a large number of traffic simulation models. These models have been developed for different applications since the 1950s. A review of several representative simulation models are presented in the appendix of this thesis. In this chapter we summarize these models and identify the requirements of the traffic simulation models for evaluation of IVHS applications. For more extensive review of existing traffic simulation models, the reader is referred to a report by Koutsopoulos and Yang <sup>[47]</sup>, and also the reports and papers by <sup>[87, 45, 60, 68, 70, 76]</sup>.

### 2.1 Existing Traffic Simulation Models

A review of several representative traffic simulation models are presented in the appendix of this thesis. This review includes the TRAF/NETSIM family which consists of NETSIM, INTRAS/FRESIM, TRAFLO, NETFLO, and FREFLO; CORQ and CONTRAM which are traffic simulation and dynamic assignment models developed during the 1970's and the 1980's; and THOREAU, INTEGRATION, DYNASMART, and Dynamic Traffic Model which are simulation models developed (or under development) for IVHS applications in recent years. Table 2.1 summarizes the characteristics of these models.

Table 2.1 Representative Traffic Simulation Models

Name	Version	Objective	Modeling Approach
CORQ	1983	Analysis of queuing in freeway/arterial corridors	Macroscopic simulation and assignment
INTRAS/ FRESIM	1988/ 1993	Evaluation of freeway operations and control strategies	Microscopic simulation
KRONOS	1987	Evaluation of traffic management strategies for freeways	Macroscopic simulation
NETSIM	1985	Evaluation of urban networks including sophisticated signal control systems	Microscopic simulation
CONTRAM	1988- present	Planning, policy analysis for urban networks, currently modified for IVHS applications	Dynamic, assignment and macroscopic simulation
INTEGRATION	1984- present	Evaluation of traffic control and management for integrated freeway/arterial networks	Mesoscopic simulation
DYNASMART	1991- present	Real time IVHS operations	Dynamic assignment and mesoscopic simulation
DYNAMIC TRAFFIC MODEL	1992- present	Decision support system for real time ATIS/ATMS operations	Dynamic assignment and macroscopic traffic simulation
THOREAU	1992- present	Evaluation of ATIS/ATMS for integrated networks	Microscopic and macroscopic simulation

In general, these models can be classified into three categories based on their resolution in representing traffic operations and controls.

**Macroscopic Simulation Models:** In macroscopic simulation models, traffic flow is usually represented by traffic cells or packets. Network performance is measured using flow-concentration relationships or link performance functions. Traffic signals and other controls are approximated using capacities and are not modeled explicitly. Macroscopic models are usually implemented in a time-based simulation framework. The states of the network, traffic flows and speeds on various links, for example, are updated at discrete time steps. Traffic delays and queuing phenomena are modeled in an aggregate fashion using relationships between demand and capacity. FREFLO [72] and CONTRAM [15, 49, 50, 51] are examples of macroscopic traffic simulation models.

Macroscopic traffic simulation models have the modest input data requirements and the highest efficiency in terms of run-time and size of the network they can handle. These models can be used for planning purposes or as decision support systems at control centers. However, the information they provide is aggregate and not sufficient for detailed IVHS system evaluation.

**Mesoscopic Simulation Models:** Mesoscopic traffic simulation models have the same characteristics as above but with detailed intersection representation instead of aggregate throughput capacities. They simulate traffic signals and limited traffic operations such as merging and metering. Queue delays at intersections are explicitly modeled (compared to macroscopic models). The flow representation in a mesoscopic simulation model can be traffic packets or individual vehicles. Their performance measures are based on the use of speed-concentration relationships and, hence, the representation of vehicle movements is aggregate. Therefore, several important aspects, such as operations of the surveillance system, incident detection, and emergency response, cannot be captured accurately. Examples of such models include DYNASMART and INTEGRATION.

**Microscopic Simulation Models:** In microscopic traffic simulation models, individual vehicles are identified and traffic controls are usually simulated explicitly. The simulation uses car-following and lane-changing models to represent traffic operations. Traffic signals, control strategies and surveillance systems are also represented in detail. Hence microscopic models have great flexibility in representing a wide variety of operating scenarios. Since they have the highest resolution and the most realistic representation of traffic operations, microscopic models are expected to be more accurate and suitable for detailed design and evaluation purposes. A disadvantage of microscopic models is the high computing (in terms of running time and memory) and data requirements. Examples of microscopic models include INTRAS/ FRESIM [92, 93, 71], NETSIM, and THOREAU.

## 2.2 Requirements for Evaluation of IVHS Applications and Limitations of Existing Models

A traffic simulation model should have the following capabilities in order to be useful in the context of evaluation of IVHS applications:

- Support a wide range of traffic control strategies
  - Ramp metering;
  - Mainline control (e.g., mainline metering, lane use signs, variable speed limits signs, reversible lanes, etc.).
  - Coordination between traffic control and motorist information systems;
  - Coordination between freeway, arterial, and urban street control;
- Support different levels of surveillance system
  - Different sensor technologies such as point data sensors (e.g., inductive loop detectors, radar detectors, etc.), point to point data sensors (e.g., vehicle to roadside communications), area data sensors (e.g., closed circuit television), external agency reports (police patrols, cellular phone call-ins);
  - Different reliability and accuracy of sensor devices.
- Support multiple vehicle classes
  - Cars/buses/trucks/others;
  - Private;
  - Commercial;
  - Guided/unguided;
  - High occupancy;
  - Emergency.
- Support different types of motorist information systems
  - Variable message signs;
  - Highway advisory radio;
  - In-vehicle route guidance systems.
- Support different drivers types and trip characteristics.

- Have flexible design and implementation that facilitates testing and evaluation of a wide range of IVHS systems
  - New sensor technologies;
  - Advanced control logic.
- Provide various measures of system performance
  - Average speeds;
  - Average travel times and delays;
  - Travel time reliability;
  - Fuel consumption;
  - Pollution;
  - Rates of queue development and dissipation;
  - Queue lengths at various instances in time;
  - Spill-backs and their extent;
  - Incident detection times;
  - Incident and emergency response time.

Since integrated and dynamic traffic control and surveillance systems are normally used in the implementation of ATIS and ATMS systems, a detailed representation of traffic operations is necessary for accurate evaluation of IVHS applications. Incident detection and response, emergency vehicle operations, mainline control are examples of IVHS control actions that have detailed traffic input requirements. Microscopic simulation models are the only ones that provide this level of details.

The older generation of microscopic models (e.g. NETSIM, INTRAS/FRESIM) have several drawbacks that limit their use in the context of IVHS evaluation. As Table 2.1 indicates, these models were developed exclusively for specific network types (e.g. freeways or urban networks) but not for integrated networks. Furthermore, even the most recent versions of freeway models (e.g., FRESIM) cannot be used for evaluation of certain control strategies (e.g. mainline control). The type of sensors in the surveillance system that can be modeled is limited. Another disadvantage of these models is the fact that the demand component is exogenous. As a result, reaction of motorists to the various controls cannot be easily incorporated. NETSIM, for example, requires as input

turning percentages at intersections. For the IVHS evaluation, however, it is important to have models that are sensitive to the various controls implemented and actions taken (e.g. response to variable message signs). Finally, due to the fact that the older generation models were developed using rigid software development methods, they are not flexible in their structure, and, as a result, very difficult to modify and enhance.

Among the new generation traffic simulation models, only THOREAU models traffic operations at the required level of detail. However, the model is still under development and initial reports indicate that it is very slow (it may take, for example, several hours to simulate a few minutes of traffic). The reason for this inefficiency is that the model was developed using an existing simulation language that adds a large overhead in computer operation. Furthermore, currently the model does not support any control logic. This element has to be programmed and added by the user.

Based on these considerations, we believe that development of an microscopic simulation model, using modern software development technologies, provides the flexibility and functionality required for system evaluation of IVHS applications.

## 2.3 Conclusion

In the previous section we concluded that (1) a microscopic simulation is required for the evaluation of IVHS applications; (2) some flexibility and functionality required for the IVHS evaluation is missing from existing models. Therefore, it is necessary to develop a microscopic traffic simulation model that builds upon existing traffic simulation models and incorporates the following considerations:

**Accuracy and Flexibility:** This traffic simulation model is designed as a tool to answer various "what if" questions in terms of traffic controls, incident response, etc., that are raised in the system evaluation and operations design. A microscopic model can support detailed simulation of various surveillance systems and be used in testing of the alternative algorithms for optimal signal setting, ramp-metering, freeway mainline control and vehicle routing. Although

running time is not an important consideration, the simulation model is designed to run with reasonable speed.

**Stochasticity of Traffic Flows:** Traffic flow characteristics fluctuate over time and space. The simulation of stochastic properties of traffic flows in a microscopic model is an important factor for its use in detailed system evaluation. By simulating individual vehicles using the car-following logic, the model is able to incorporate the diversity of traffic flow and variation in drivers' behavior. The speed-concentration relationships used by macroscopic models assume that a given freeway segment has a known *capacity* (or *jam density*). However, these capacities may not be readily available, especially for a new system that will be equipped with new traffic management technologies. Using the car-following logic in a microscopic model, vehicle movements are dependent on interactions with leading and following vehicles in the same lane and adjacent lanes. Instead of making assumptions of homogenous speed and density for the vehicles moving on a section of the link, and treating capacity as an exogenous parameter, individual vehicles can have different attributes such as reaction time, desired speed, acceleration, emission parameters, acceptable gaps for merging, etc. As a result, capacity becomes an output, instead of an input, of the model.

**Vehicle Route Choice Behavior:** The traffic simulation model is designed with the ability to model drivers' route choice behavior. Such behavior may be affected by variable message signs, advisory radio, on-board route guidance systems and vehicle to roadside communication facilities. By incorporating driver's routing behavior in the simulation, different types of traffic information systems and advisory logic can be tested and evaluated.

**Advanced Traffic Controls:** ATMS may include variable speed signs and lane use signs, metering and closure signals and dense network of detectors. In order to model such systems, the simulation model will represent the surveillance system in detail, generate the data items obtained from various detectors, and mimic the control logic used. The microscopic modeling approach provides the flexibility required by simulating sensor data and incorporating new approaches for advanced traffic controls, and future developments in sensor technologies.



# Chapter 3

## Overview of the Simulation Framework and Structure of the Model

MITSIM is a microscopic traffic simulator designed to model the performance of transportation networks under alternative configurations and IVHS strategies. It simulates vehicle movements given the travel demand and traffic management scheme. In order to support the traffic operations in an ATIS/ATMS environment, travel demand between each O-D pair and control policy (traffic signals and message signs) are time-dependent and can be updated dynamically during the course of simulation. The output of MITSIM includes various measures of effectiveness (MOE's) required in the system evaluation of an application. For example, they may include density, speed, travel times, queues, delay, fuel consumption, incident response time, diverted traffic, etc. These measures and their evaluation over time can be provided at the single station, lane, segment, link, O-D pair or at the network level. The specific elements of MITSIM and the simulation structure are described in this chapter.

### 3.1 The Overall Framework

Figure 3.1 shows the overall framework under development for the detailed evaluation of IVHS strategies. In order to incorporate all the above elements, the framework consists of three major components: (1) travel demand model, (2) control strategy model, and (3) traffic simulation model.

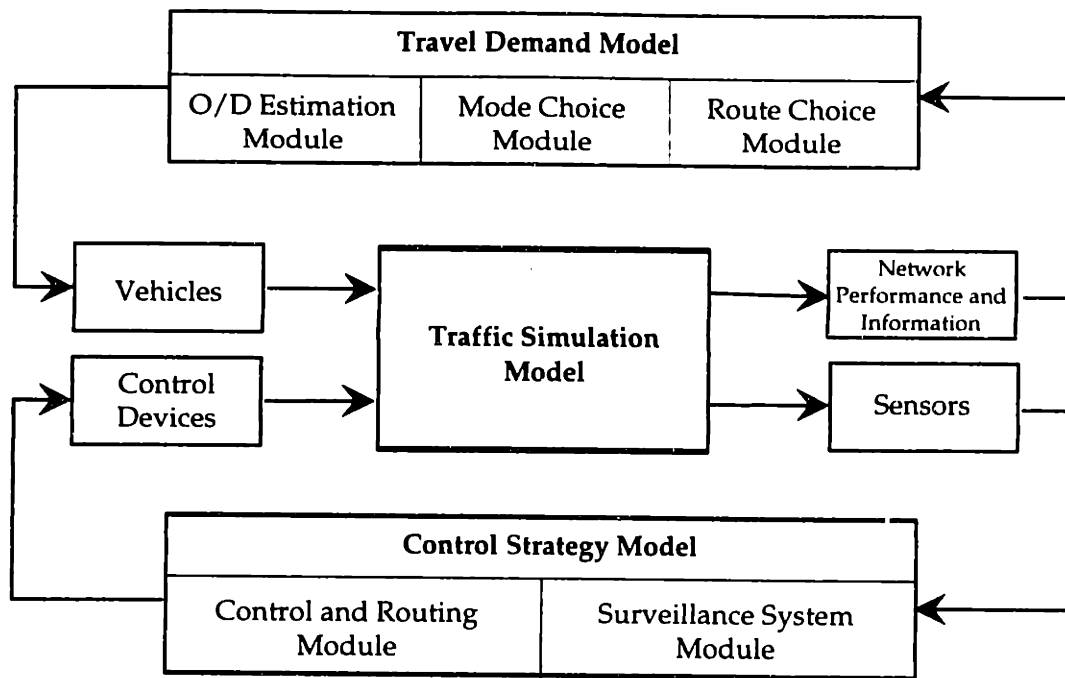


Figure 3.1 A Simulation Framework for IVHS Applications

**Travel Demand Model:** The travel demand model takes as input the historical information of origin-destination trip matrix, travel time/cost, and "real time" link traffic counts to estimate time-dependent travel demand between each pair of origins and destinations and predict drivers' pre-trip mode and route choices.

**Control Strategy Model:** The control strategy model is designed to simulate both the information collection and the control mechanism in traffic networks. It consists of three modules:

- **Surveillance System Module:** The surveillance system module simulates traffic sensors (detectors in the pavement, video cameras, beacons and transponder equipped vehicles). Information is collected continuously, rapidly processed and made available to other elements of the system.
- **Traffic Prediction Module:** The traffic prediction module has the responsibility of providing the information that is needed to generate control strategies.
- **Control and Routing Module:** The control and routing module generates traffic control and route guidance strategies in response to the information provided by the surveillance system and the traffic prediction modules.

**Traffic Simulation Model:** This is the key element of the system. It takes travel demand (represented by time dependent origin/destination trip tables), traffic controls, and route guidance as input, and simulates vehicle movements in the network.

These models are coordinated with each other and construct the basic architecture of MITSIM. A time-based simulation approach is used in the implementation of this system. Each component in the system is executed at a frequency specified by the user, or whenever an event that needs to call the corresponding procedure occurs. The *traffic simulation model* admits new vehicles into the network at their scheduled departure times, and removes vehicles from the network when they arrive at their destinations. Traffic signal settings and message signs are updated by the *control strategy model*. At the beginning of each iteration, acceleration or deceleration as well as lane changes are calculated for all the vehicles currently in the network. Within each iteration, vehicles' speeds and positions are updated based on their current acceleration (deceleration) rate at a higher frequency (the simulation clock is advanced, for example, every 1/10 seconds); if a vehicle crossed or occupies a sensor reader (detector, VRC receiver, etc.), data items to be collected from that sensor are sent to the *surveillance system model*.

The focus of this thesis is on the last element of MITSIM, the traffic simulation model.

## 3.2 Traffic Simulation Logic

The traffic simulation model consists of a loop of sequential modules that are invoked at specified frequencies (time-based) or when certain events occur (event-based). The loop is processed for each simulation interval (for example, every 0.1 second). The simulation logic is summarized in the flow chart in Figure 3.2.

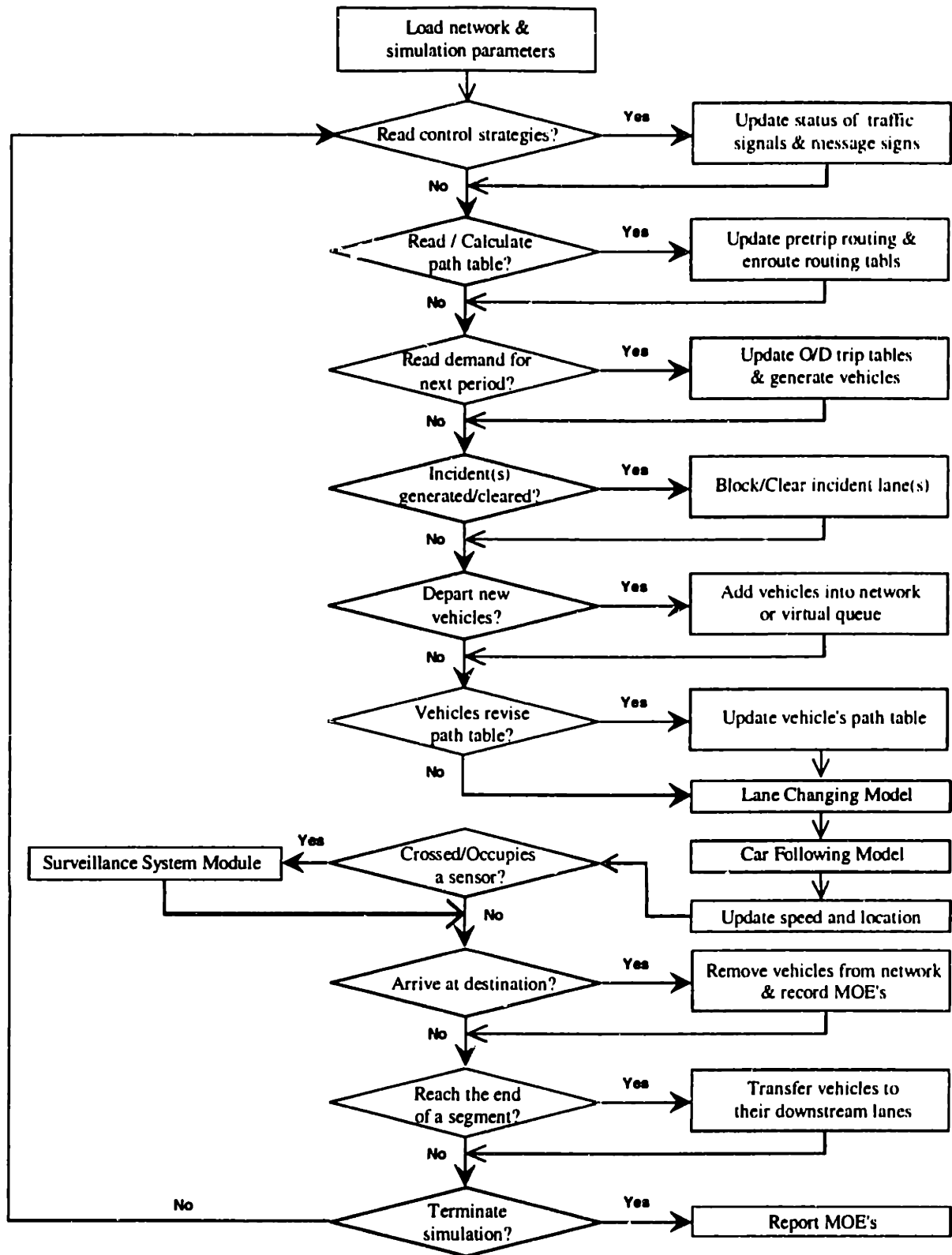


Figure 3.2 Flow Chart of the Traffic Simulation Model

The main elements of the simulation model are:

- Traffic network
  - Road network
  - Operation of traffic control system
  - Operation of the surveillance system
- Vehicle movement logic

These elements are discussed in detail in Chapter 4 and 5.

### 3.3 Implementation Considerations

The model is implemented in C++ using the object oriented programming paradigm. The program is designed to run on either UNIX workstations that support the X window system, or PCs.

To implement the traffic simulation presented in Section 3.2 efficiently, certain memory saving considerations have to take place. A modular structure is employed which ensures that the program can be compiled to support dynamic loading of modules and overlays.

Vehicles that will depart during the next time interval (e.g., 15 minutes) are generated based on time dependent O/D matrices. The generated vehicles are sorted by their departure time and stored in a file. Only the vehicles currently traveling in the network are kept in the memory. Once a vehicle arrives at its destination, the memory used for that vehicle is returned to the system for reuse.

The shortest path and routing information between each pair of nodes are also stored in files, and retrieved when they are needed.

The program dynamically allocates all the required memory for the data structures and objects used in the model. The size of the problem that can be handled is only restricted by the memory available.

# Chapter 4

## Elements of the Traffic Network

The elements of the simulated network in MITSIM include the road network, traffic control devices and surveillance systems. The representation of these elements in the simulation model is described in this chapter.

### 4.1 Road Network

MITSIM is designed to support detailed representation of integrated road networks (freeways and urban streets). The elements of the road network are organized in a hierarchical fashion and include nodes, links, segments and lanes.

#### 4.1.1 Nodes

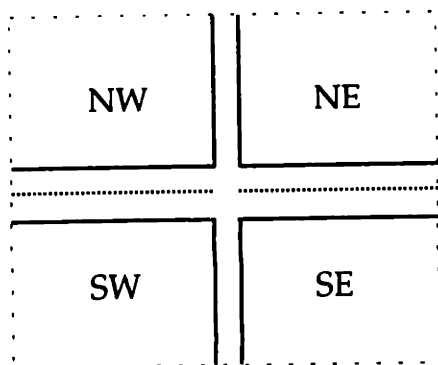
Nodes include signalized and unsignalized intersections, ramp intersections and external sources/sinks that receive and/or discharge vehicles. Each node is represented by a data structure that contains the following information:

- Type:
  0. Intersections
  1. External nodes
  2. Centroids of traffic zones
- Adjacent links (if *Node.type* = 0 or 1)
  - Upstream links
  - Downstream links
- Distributors (if *Node.type* = 2)
- Coordinates

*Node.type* indicates whether a node is a normal intersection, an external node of the simulated network, or a centroid of a traffic zone. For an intersection node or an external node, the adjacent upstream and downstream links are stored. For a node that represents the centroid of a traffic zone, a list of distributors that disperse vehicles to and from that node is also stored.

Figure 4.1(a) shows an example of an area which consists of 4 traffic zones, separated by an east-west two-way street and a south-north one-way street. Figure 4.1(b) illustrates the node and link representation of this road network. In this network, node 1 is a normal intersection, nodes 2-5 are external nodes, and nodes 6-9 are centroids of traffic zones. For node 1 the adjacent links include upstream links 1, 4 and 5 and the downstream links 2, 3, and 6. The centroid node 6, corresponding to the area NE, may have links 1, 2 or 6 as its distributors.

(a) Road Map



(b) Network Representation

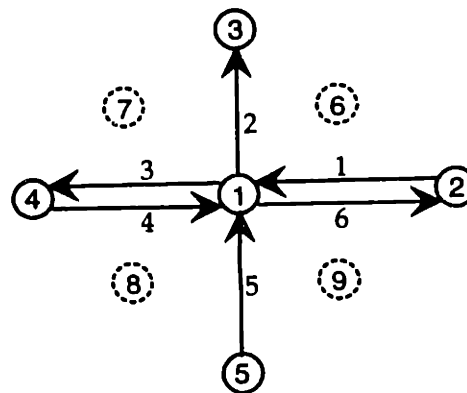


Figure 4.1 Definition of Nodes

### 4.1.2 Links and Segments

Links are the directional roadways that connect nodes. A link may be further decomposed into segments. For demonstration purposes we use the freeway network shown in Figure 4.2.

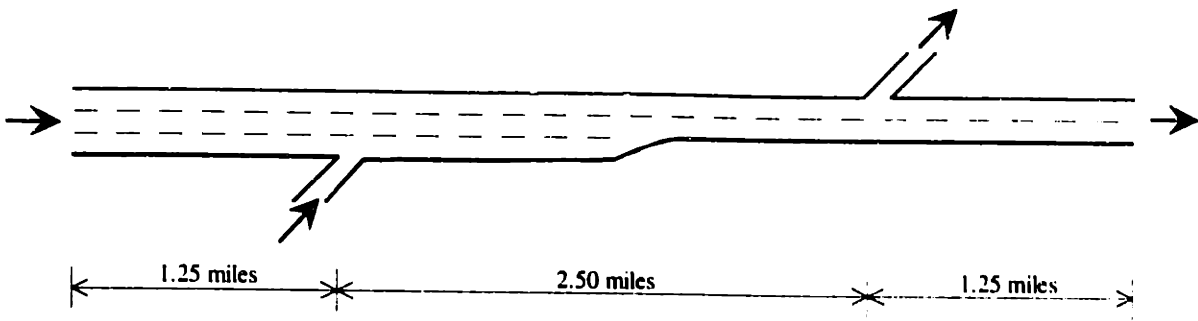


Figure 4.2 A Hypothetical Freeway Network

The coding of nodes and links of this hypothetical network is shown in Figure 4.3(a). Every node and link are assigned unique identification (ID) numbers. The order of these ID numbers can be arbitrary but they must be continuous integers.

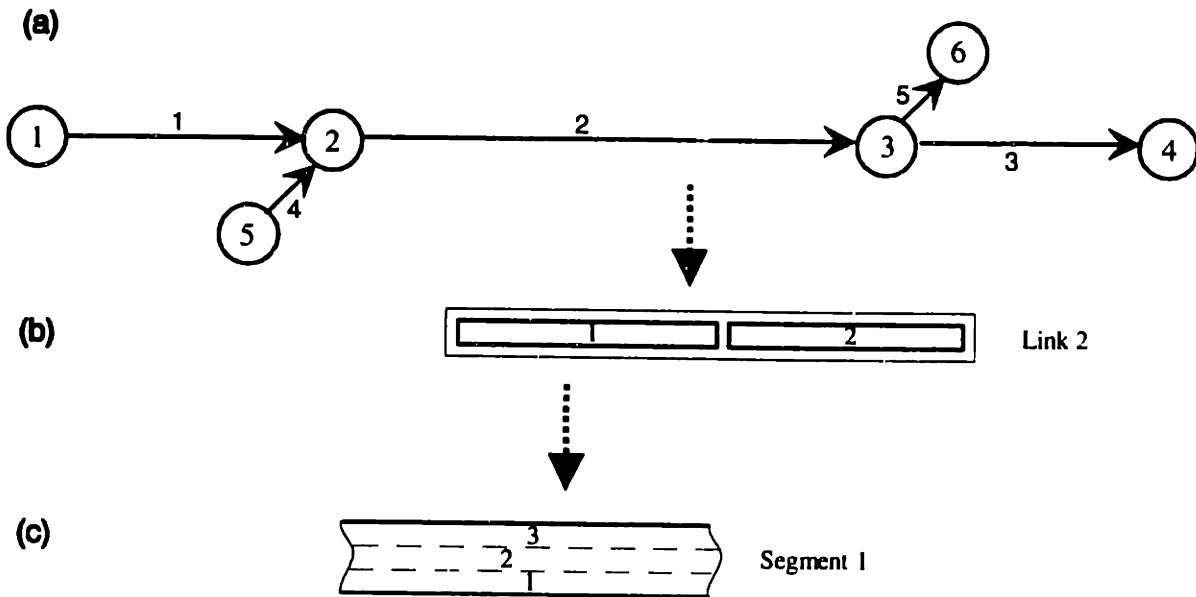


Figure 4.3 Representation of the Road Network

A link can be either a freeway, a ramp, or an urban street. Two-way roadways are represented by two links, one for each direction. The input required for each link includes its start and end nodes, the ID number of its paired link if it belongs to a two-way roadway, average travel times and variances, and fixed cost (see Table 4.1). Each link may have several segments or only one segment. Dividing long links into short segments may improve the running



speed<sup>1</sup> of the program (especially when the traffic is congested) but increases its memory requirements.

Table 4.1 Link Information

Link ID	Start Node	End Node	# of Segs.	Paired Link	Avg. Travel Time		Travel Time Var.		Fixed Cost
					Peak	Off-Peak	Peak	Off-Peak	
1	1	2	1	0	2.0	1.5	0.15	0	0
2	2	3	2	0	4.0	3.0	0.30	0	0
3	3	4	1	0	2.0	1.5	0.15	0	0
4	5	2	1	0	2.0	1.0	0.20	0	0
5	3	6	1	0	2.0	1.0	0.20	0	0

Figure 4.3(b) shows that link 2 is composed of 2 segments. Segments are defined as homogenous sections of a roadway. If different parts of a link vary in characteristics such as the number of lanes, curvature, grade, and pavement quality, then the link is divided into segments so that the characteristics of the link along each segment remain constant. The segments that belong to a link are numbered from 1 to  $n$  starting from upstream to downstream sections. For each segment, the input data includes its length, number of lanes, grade, posted speed limits, free flow speed, and geometric characteristics (see Table 4.3). Segment information for the hypothetical network in Figure 4.2 is given in Table 4.2.

Table 4.2 Segment Information

Segment ID		Length (miles)	# of Lanes	Grade (%)	Speed Limit (mph)	Free Flow Speed (mph)
Link	Segment					
1	1	1.25	3	0	50	60
2	1	1.25	3	0	50	60
2	2	1.25	2	0	50	60
3	1	1.25	2	0	50	60
4	1	0.50	1	0	30	30
5	1	0.50	1	0	30	30

---

<sup>1</sup> Vehicles in each lane of the simulated network are kept in a double-linked list. Each vehicle has pointers to the lead and following vehicles, but no direct information about left and right neighbors since they may change dynamically. Searching operations are performed on the linked lists of vehicles whenever a driver needs to know his/her neighbors (e.g., in lane-changing algorithm). Dividing long link into short segments may reduce the searching operations (at the expense of increasing the number of segments).

Each link segment is assumed to have the same number of lanes and other traffic operating parameters. A change in grade, in horizontal curvature or in pavement condition is sufficient reason to divide a link into segments. Grade is used as one of the parameters in the calculation of acceleration rates for vehicles that travel on the segment with a certain speed. Curvature and pavement conditions limit vehicle performance which in turn, affect vehicles' free flow speed. An upper bound for free flow speed is calculated based on the radius of curvature, super elevation and pavement condition [58]:

$$V = \sqrt{15 R (e + f)} \quad (4.0)$$

where:

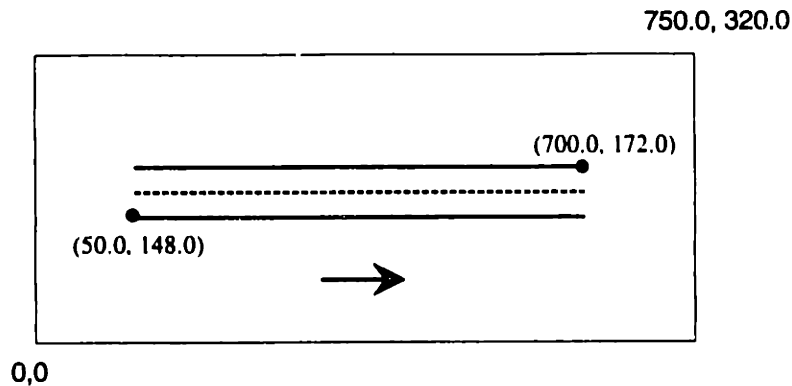
- $V$  = upper bound on free flow speed, miles per hour;
- $e$  = rate of roadway superelevation, foot per foot;
- $f$  = friction coefficient for given pavement condition;
- $R$  = radius of curve in feet.

The simulation model applies the free flow speed, a road condition dictated by the upper bound, in the calculation of an individual vehicle's desired speed (see Equation 4.1).

The data that describes the geometry of the segments are also provided as input to the simulator when graphical animation is required. To describe the geometry of a link segment, the two curb lines are digitized. Each curb line may consist of one or multiple line segments, and be represented by a list of vertices. The geometric information has the following format:

- Link ID and Segment ID
- Geometry of right curb line
  - # of vertices;
  - a list of  $x, y$  coordinates (the number depends on its curvature).
- Geometry of left curb line
  - # of vertices;
  - a list of  $x, y$  coordinates (the number depends on its curvature).

Figure 4.4 Coordinate System for Geometric Data



For example, the geometry data for the segment shown in Figure 4.4 contains the data items outlined in Table 4.3. The data for the right and left curb line of this segment are tabulated sequentially, and each includes all the vertices (two vertices in this example) listed from upstream to downstream.

Table 4.3 Geometric Data of a Segment

Link ID	Segment ID	# of Vertices	x Coordinate	y Coordinate
1	1	2	50.0	148.0
			700.0	148.0
		2	50.0	172.0
			700.0	172.0

### 4.1.3 Lanes

Link segments are further decomposed into lanes as shown in Figure 4.3(c). Each segment may have any number of lanes. All lanes contained in a link segment, numbered from right to left, are represented as an array of structures. Each lane has also two arrays of pointers to the upstream and downstream lanes that this lane is aligned with (*see* Figure 4.5).

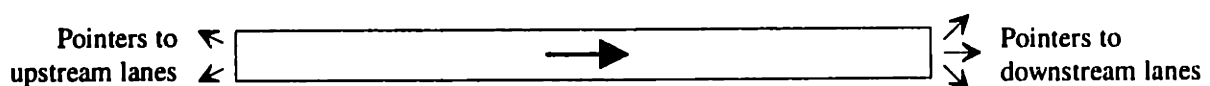


Figure 4.5 Lane Alignment and Pointers

Input data that describes lane alignment includes the number of upstream lanes, and link and lane IDs of the lanes aligned. Based on this information, pointers to the aligned downstream lanes are generated automatically. The information for all lanes of our explanatory example in Figure 4.2 is shown in Table 4.4. In this table, the first 3 rows indicate that there is no upstream lane for the 3 lanes of the only segment of *link 1*. The fourth row indicates that 2 lanes, namely *lane 1* of the last segment of *link 1* and *lane 1* of the last segment of *link 4*, merge into *lane 1* of *segment 1* of *link 2*. Segment ID is not necessary for identifying the aligned upstream lane (see the 5<sup>th</sup> column in Table 4.4) because it must be the last segment of an upstream link.

Table 4.4 Lane Alignment Information

Lane Id			Lane Alignment		Regulation
Link	Seg.	Lane	# of Lanes	Upstream Lanes (link, lane)	
1	1	1	0		2
1	1	2	0		3
1	1	3	0		1
2	1	1	2	1, 1; 4, 1	2
2	1	2	1	1, 2	3
2	1	3	1	1, 3	1
2	2	1	2	2, 1; 2, 2	2
2	2	2	1	2, 3	1
3	1	1	1	2, 1	2
3	1	2	1	2, 2	1
4	1	1	0		0
5	1	1	1	2, 2	0

The last column in Table 4.4 indicates the regulations for lane changes and lane use (*rules*). The first two bits (bits 0 and 1) of its value indicate if lane changing to its right and left neighbor lanes are allowed. The 4 possible values are:

0. No lane changing is allowed (00);
1. Change to the right is allowed but not to the left (01);
2. Change to the left is allowed but not to the right (10);
3. Change either to the left or the right is allowed (11).

The values in parentheses are the binary representation. For the shoulder lane on the right, the value can not be 1. For the shoulder lane on the left, the value can not be 2. If there is only one lane in the segment, the value must be 0.

Bits 7-9 of the variable *rules* specify the regulation for lane use, and the values used are as follows:

- 64. Electronic toll collection (ETC) vehicles use only;
- 128. Transit and car-pool vehicles use only;
- 256. Truck exclusive use.

Other bits of the variable *rules* are reserved.

Vehicles that are currently located in a lane are represented in a double linked list. Each lane has two pointers that keep track of the first and last vehicles in this list. Pointers to lane specific detectors and control devices are also maintained in the lane object (*see* Section 4.2 and 4.3 for details).

#### 4.1.4 Toll Facilities

Toll facilities simulated in the MITSIM model may consist of conventional toll booths and electronic toll booths. Every toll facility is assumed to be located at the downstream end of a segment. In other words, a toll plaza naturally separates a roadway into two or more links or segments. A segment that has a toll plaza at its downstream end is assigned a pointer to the corresponding toll booth data structure (the segments without toll booths are assigned NULLs for these pointers). Figure 4.6 shows a hypothetical network with toll facilities. The only toll booth is located between *segment 2* and *3* of *link 4*. In this example, vehicles in *segment 2* of *link 4* must stop or yield when they arrive at the toll booth in order to get an entry ticket or pay the toll.

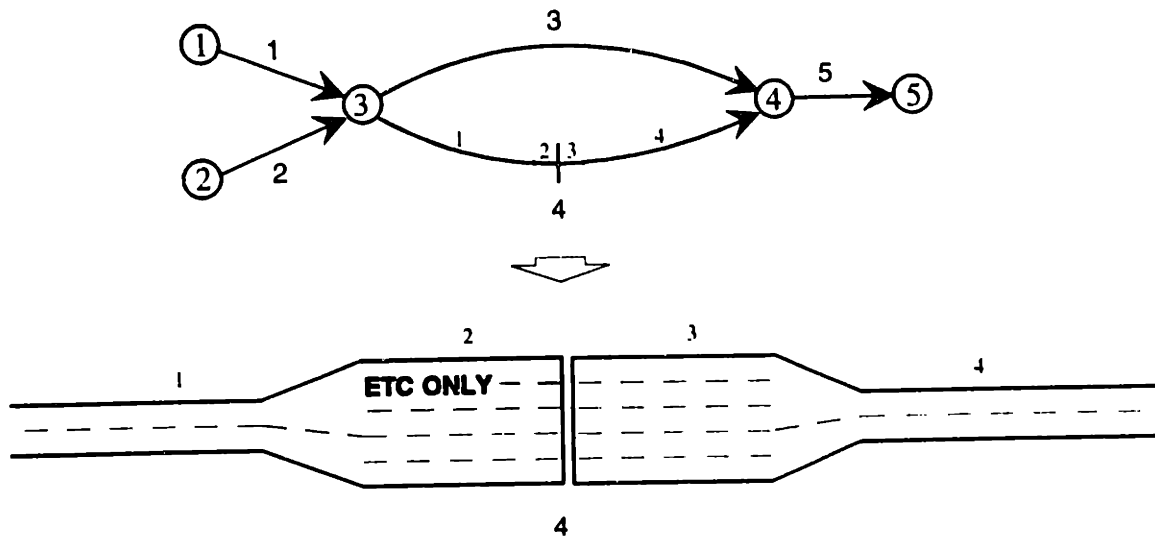


Figure 4.6 Toll Plaza

Vehicles equipped with electronic toll collection (ETC) devices may select the ETC lanes where they do not need to stop, while other vehicles have to pass conventional toll lanes and possibly incur queuing delays. The input data required in the simulation model for a toll booth includes its location (link and segment ID's), toll fare tables, maximum speed and average service rate for each toll lane. Table 4.5 shows an example of a lane specific information that describes the toll booth in Figure 4.6.

Table 4.5 Data for the Toll Booth 1 (at Segment 2 of Link 2)

Lane ID	Lane Use Regulation	Speed (mph)	Mean Service Rate (vehicles/hour)
1	0	0	800
2	0	0	800
3	0	0	900
4	64	45	1,800
5	64	45	1,800

In this table, each row corresponds to one lane. Toll lanes are numbered from right to left. The second column shows that the first three lanes can be used by any vehicle, and the last two lanes are reserved for ETC vehicles. The third column indicates the maximum speed for passing the toll booth, and the last column shows the average service rate for paying toll fare.

How much one pays at a toll plaza may affect his/her route choice behavior. Furthermore, congestion pricing may be used for demand management and traffic analysts may be interested in the study of different pricing policies. In the traffic simulation model, the toll fare a vehicle pays is recorded and reported when the vehicle arrives its destination. Price information at a toll booth is described by a fare table. This table is three dimensional (including time, vehicle type, and upstream entry), and indicates how much the driver should pay when he/she arrives at the toll booth in a given time period. Table 4.6 shows an example toll fare table. Each row in the table applies to vehicles that come from the same origin (e.g., ramp or upstream toll booth). To facilitate studies of congestion pricing, drivers who use the system during different time periods may be charged differently. The toll fare can be based on either arrival time at the toll booth or entry time into the system.

Table 4.6 Toll Fares at Toll Booth 1 (in cents)

	Peak Period (1)				Off-Peak Period (2)			
	Car	Pickup	Bus	Truck	Car	Pickup	Bus	Truck
Link 1	100	100	200	300	50	50	100	150
Link 2	50	50	100	150	25	25	50	75
Other	150	150	300	450	75	75	150	225

User can specify the number of periods for applying different charges to motorist. A time table is used to indicate the times corresponding to each period. For example, the peak and off-peak periods illustrated by the fare structure of Figure 4.7 are represented in Table 4.7.

Table 4.7 Time Table for Toll Fares

Time Period	00:00 - 07:00	07:00 - 09:00	09:00 - 16:00	16:00 - 19:00	19:00 - 24:00
Toll Fare Type	off-peak	peak	off-peak	peak	off-peak
	2	1	2	1	2

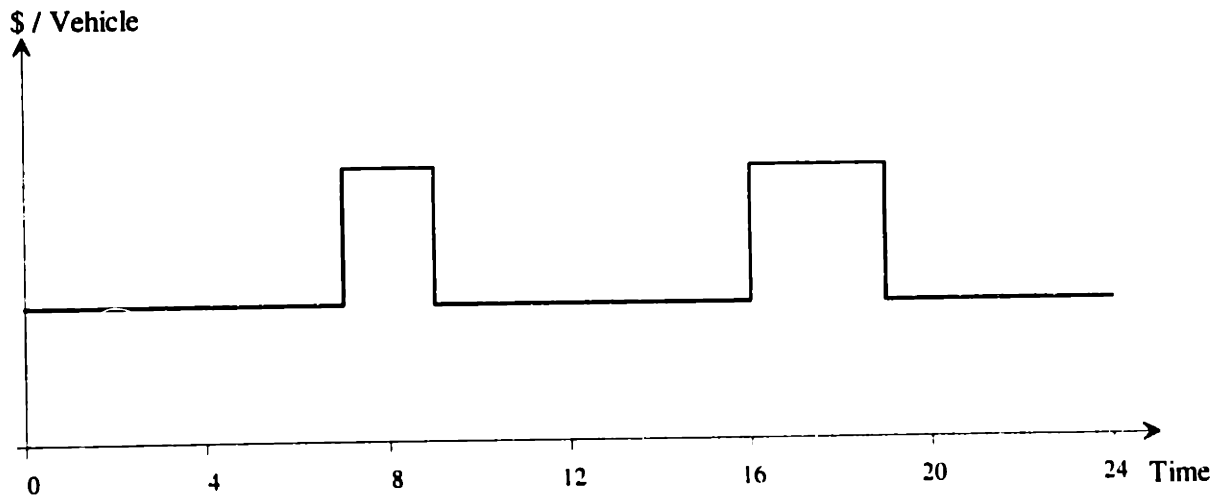


Figure 4.7 Time Dependent Toll Fare

If all types of vehicles are charged with the same fare no matter where they come and when they arrive, toll fare table reduces to a single scalar (e.g., \$1/vehicle). However, a general representation of toll fares is provided in this simulation for modeling advanced traffic management schemes such as congestion pricing.

## 4.2 Traffic Control Devices

The devices of the simulated traffic control system include:

- Traffic signals and ramp meters;
- Lane use signs;
- Variable speed limit signs;
- Direction signs;
- Variable message signs;
- Vehicle to Roadside Communication devices (VRC);
- Highway Advisory Radio (HAR).

These control devices can be area specific (e.g., HAR), link specific (e.g., variable speed limit signs, variable message signs, etc.), or lane specific (e.g., traffic lights, lane use signs, etc.). Linked lists are used to represent link or lane specific control devices. The objects in this list, each corresponding to a specific control



device, are sorted by their longitudinal locations from the downstream end of the segment and each have pointers to the next item in the same list (see Figure 4.8). The segment or lane has a pointer to the first item (upstream) in the list.

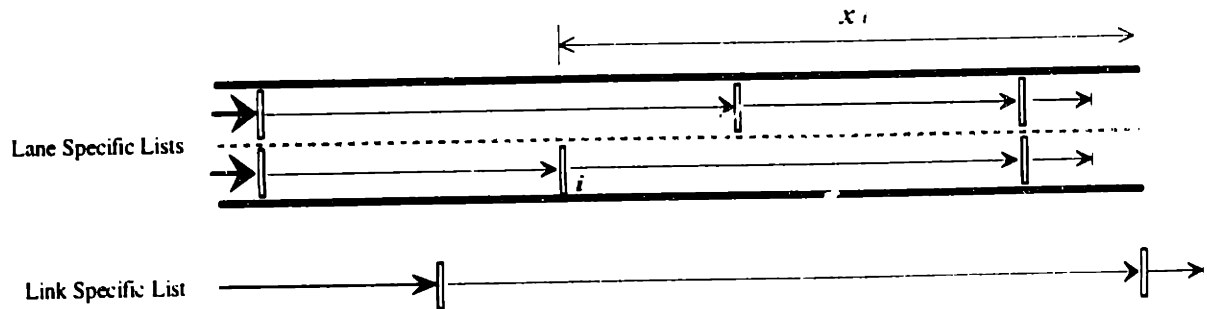


Figure 4.8 Linked Lists of Control Devices

Each vehicle moving in the simulated network has two pointers that indicate the next link specific and lane specific control devices it will meet. These pointers are updated when a vehicle changes lanes or passes a control device. Once a vehicle is close enough to a downstream control device, a procedure of this control device is invoked to update the vehicle's characteristics (e.g., desired speed) or trigger a proper action (e.g., acceleration/deceleration, changing lanes, diverting from scheduled route, etc.).

Traffic signals, ramp meters, lane use signs, and variable speed limits signs have been implemented in the model.

#### 4.2.1 Traffic Signals and Ramp Meters

Traffic signals are characterized by their phases and time settings (generated by the *control strategy model*). Because both fixed time and adaptive traffic controls may be exercised in a simulated network, the traffic simulation model reads initial signal settings from a file, and then it communicates with the control strategy model during the course of the simulation to update them as required. The information that describes the status of a traffic signal and the right-of-way of vehicle movements is the following:

- Number of phases and sequencing;
- Time allocated to each phase;
- Movements allowed during each phase;
- Initial phase.

A countdown clock is used for each traffic signal to count the time left for the current signal phase. Once the clock returns to zero, the signal shifts to the next phase, and the clock is reset. When the last phase is completed, it moves to the first phase and the entire process is repeated. An integer variable in lane object, called *signal status*, is used to indicate which movements to the downstream links are allowed during the current phase.

The variable *signal status* is divided into 4-bit segments. Each segment corresponds to a (turning) movement to a downstream link. The number of segments equals to the number of out-going links of the intersection (for example, an intersection of two-way streets needs 4 segments, an on-ramp to freeway needs one segment). In each of the 4-bit segments, the first 2 bits indicate the color of the traffic light for the (turning) movement represented by the corresponding segment. The values used are defined as follows (the value in parentheses is in binary):

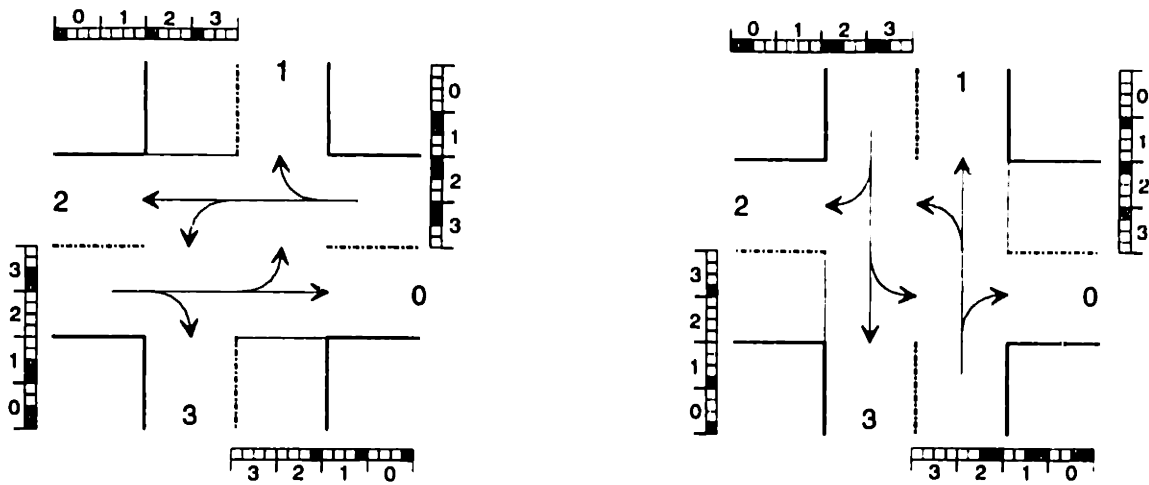
0. Blackout (0000)
1. Red (0001)
2. Yellow (0010)
3. Green (0011)

The third bit (bit 2) indicates if the signal is flashing, and the fourth bit (bit 3) indicates if this signal is unused, i.e.:

4. Flashing (0100)
8. Unused (1000)

If a signal is unused, all other bits are not defined and ignored.

For unsignalized intersections, stop signs are implemented as flashing red lights, and yield signs as flashing yellow lights.



(a) Phase 1 - Green to East and West Bounds

(b) Phase 2 - Green to North and South Bounds

Figure 4.9 A 2-phase Traffic Signal and Corresponding *Signal Status*

In the 2-phase signalized intersection shown in Figure 4.9 there are 4 incoming links and 4 outgoing links. Each link has only a single lane (traffic signals are specified for each lane in MITSIM). During phase 1 the east- and west-bound traffic have green lights, and north- and south-bound traffic have red lights. The *signal status* for east-bound traffic is 0011'0000'0011'0011 in binary or 12,339 in decimal, and indicates that the movements from west (2) to south (3), east (0) and north (1) are allowed. During phase 2 all lights are reversed. In this case, the *signal status* for east-bound traffic is 0001'0000'0001'0001 in binary or 4113 in decimal, and indicates red lights for all the movements.

Vehicles within a certain distance from a traffic signal calculate their acceleration or deceleration rate based on the signal status, their current speed and distance from the stop line (*see Chapter 6 for details*). If traffic lights are green (or yellow and the vehicle is within a certain distance), a vehicle that requires a turning movement will examine the vehicles from other directions before it commits to the movement.

#### 4.2.2 Lane Use Signs (LUS)

Lane use signs are similar to the conventional traffic signals discussed above, but there is no segmentation (because the traffic is for one direction) and 6 bits are used for specifying a signal status. The colors for LUS are indicated by the first 2 bits (bit 0 and 1) of the signal status and are defined as follows:

- 0. Closed (000000);
- 1. Red (000001);
- 2. Yellow (000010);
- 3. Green (000011).

Similar to intersection traffic lights, the third and fourth bits indicate if the signal is flashing or unused.

- 4. Flashing (000100);
- 8. Unused (001000).

If the LUS is unused, all other bits are not defined and ignored.

The fifth and sixth bits (bits 4 and 5) provide additional information on lane changing recommendations. These two bits represent the "arrows" and/or the complementary message sign:

- 16. Merge right (010000);
- 32. Merge left (100000);
- 48. Stay in lane (110000).

A green light with no recommendation for lane changing, for example, has a status value of 3; a yellow light with a flashing "merge left" sign has a status value of 38 (2+4+32).

Vehicles within a certain distance from a LUS will invoke procedures in the car-following and lane changing modules to check their acceleration (deceleration) rates and lane change behavior (*see* Chapter 5).

Implementation of LUS requires as input the following information:

- Location (offset from downstream end of the lane).
- Phase information:
  - Number of phases;
  - Time allocated to each phase;
  - Signal status during each phase.
- Initial phase.

### 4.2.3 Variable Speed Limit Sign (VSLs)

Speed limit signs are considered to be advisory devices because they are not strictly enforced. Each speed limit sign causes the simulated vehicles to recalculate their desired speeds, and these desired speeds will be used for the remaining trip until a new speed limit sign is encountered or the vehicle is moved into a new segment which has a different speed limit and free flow speed. The mapping from the posted speed limit to a driver's desired speed is based on the following formula:

$$V_{nl} = \min \{ r_n \tilde{V}_l, \bar{V}_{nl} \} \quad (4.1)$$

where:

- $V_{nl}$  = desired speed of driver  $n$  on link segment  $l$ ;
- $\tilde{V}_l$  = speed limit on link segment  $l$  as advised by VSLs;
- $\bar{V}_{nl}$  = free flow speed for vehicle  $n$  on link segment  $l$ , a function of vehicle type and road conditions;
- $r_n$  = desired speed ratio, a random variable generated from a known distribution.

Observed data may be used to describe the speed distribution (PDF) and to calculate  $r_n$  based on the accumulated probability (CDF). An example is shown in Table 4.8.

Table 4.8 Distribution of Desired Speed Ratio

Driver Population	Accumulated Population	Desired Speed Ratio
0.05	0.05	80
0.05	0.10	90
0.25	0.35	100
0.35	0.70	110
0.20	0.90	120
0.10	1.00	130

When a driver passes a speed limit sign, a random number uniformly distributed between [0.0, 1.0) is generated and the desired speed ratio is calculated based on the second and third columns in this table. For example, if the random number generated is 0.80,  $r_n = 1.20$ .

### 4.3 Simulation of the Surveillance System

To evaluate "real world" traffic performance and control strategies, it is necessary to simulate realistically the operations of the corresponding surveillance system. This chapter describes the sensor devices which will be supported by the simulation model.

Sensor technologies appropriate for obtaining information about traffic congestion and incidents can be classified into 4 groups according to the type of data they obtain:

- Point data;
- Point to point data;
- Area data; and
- Other data.

To obtain these data, various sensor devices are available or under development. Different sensor devices have different characteristics such as reliability and type of data collected.

The MITSIM model uses two methods for simulating the surveillance system:

- A "generic sensor device" module is used to collect data on traffic counts, occupancy, instantaneous speed, density, vehicle type, length/height, identification, etc. Each sensor device/station in the simulated network may collect any of these data items or combinations based on the type of that sensor device.
- A dummy surveillance module is invoked whenever a vehicle passes a sensor device. The *Surveillance System Model* developer can rewrite this dummy module to simulate the data collection mechanism in detail for the sensor devices of his interest.

A linked list of surveillance objects is maintained for each lane that has been equipped with sensor devices. Any sensor/detector is a specific item in a surveillance object. Each object in this linked list is characterized by the longitudinal location from the downstream end of the segment of the

corresponding device and the data items it collects (see Figure 4.10). A probability of "fail-to-work" and percentage error can also be associated with each type of data that a sensor device collects. Each simulated vehicle in the network has a pointer to the next sensor device. When a vehicle passes or occupies a sensor device, it causes the *surveillance procedure* to output the information that this device collects.

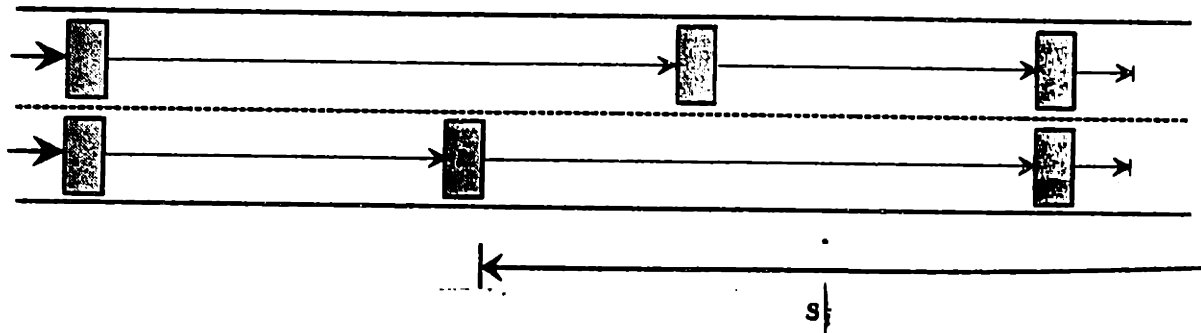


Figure 4.10 Implementation of Surveillance System

In the MITSIM model, mechanism of different sensor devices are not objects of the simulation. Instead, the model analyzes data to be collected by sensor devices. User may add noises to the data collect by different types of detectors to reflect the bias and reliability of that detectors. Furthermore, a simulated detector can be a hypothetical one for which the required technologies is still under development. User may install hypothetical detectors in the simulated network to collect traffic data of his interest. The data that can be collected by the model include following:

1. Flow;
2. Speed;
4. Occupancy;
8. Headway;
128. Vehicle type;
256. Over-height attribute (Yes/No);
2048. Vehicle Identification (ID);
4096. Origin-destination.

More items may be added in this list. Each detector (a single sensor or paired/grouped sensors) in the simulated network can be specified to collect any combination of the above items. For example, to collect flow and occupancy,

a detector is assigned a *task* value of 5 (1+4). To collect vehicle's speed and ID, the *task* value is 2050 (2+2048).

This approach increases the flexibility of the model and facilitates the implementation of a wide range of sensor devices.

### 4.3.1 Point Data Sensors

Point data sensors are used as the principal surveillance tool in the majority of traffic management systems today. Point data sensors include the classic inductive loops, radar detectors, infrared detectors, magnetometers, fiber-optic pressure detectors, ultrasonic detectors, some video image processing applications, and other sensors that collect traffic data at a point. The parameters collected by point data sensors are usually traffic volume, occupancy and, with some configurations, speed, height and type of vehicle class.

Several notations used in calculations of point data are illustrated in Figure 4.11 and described as follows:

- $L$  = vehicle length;
- $x(t)$  = vehicle's position at time  $t$ ;
- $v(t)$  = vehicle's speed at time  $t$ ;
- $a(t, t + dt)$  = vehicle's acceleration (deceleration) rate during the interval of time between  $t$  and  $t + dt$ ;
- $s$  = detector's location;
- $z$  = length of the detection zone.

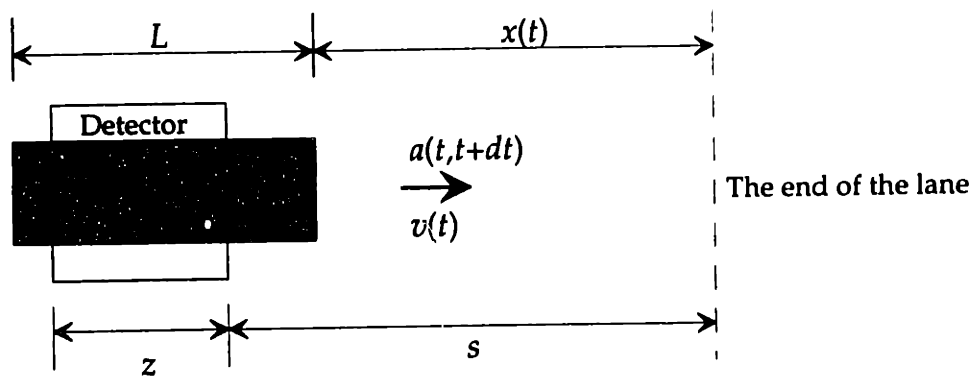


Figure 4.11 Variables in Point Data Calculation

Note that a sensor device in the simulated network may have a specific detection zone. However, the length of the detection zone can be zero. The



position of a detector is measured by the distance from the downstream edge of the detection zone to the downstream end of the lane.

The calculations of point data in the simulation model are described in the followings.

### *Flow*

Flow is defined as the number of vehicles passing a point on the roadway during a specified period of time. Position of each vehicle is scanned by the surveillance module at a prespecified frequency (e.g., 0.1 second). When a vehicle activates a detector (specifically, the front of the vehicle passes the upstream edge of the detection zone), it is counted into the flow of that detector for the specified time period.

### *Speed*

In the simulation model, the speed of a vehicle reported by a point date sensor is the instantaneous velocity that the vehicle passes the sensor device. This speed is computed when the back of the vehicle passes the downstream edge of the detection zone, using interpolation method. Let:

- $dt$  = step size of the simulation clock;
- $v(t)$  = speed at the beginning of the time interval;
- $v(t + dt)$  = speed at the end of the time interval.

Then the average acceleration (deceleration) rate during the time interval  $[t, t + dt]$  is:

$$a(t, t + dt) = \frac{v(t + dt) - v(t)}{dt} \quad (4.2)$$

The distance for the vehicle to travel before it passes the detector is  $x(t) - s + L$ . Since the speed is computed when a vehicle passes the detector, we know that  $x(t) - s + L \geq 0$  and  $x(t + dt) - s + L < 0$ . Given the acceleration (deceleration) rate  $a(t, t + dt)$  and an initial speed  $v(t)$  and position  $x(t)$ , the travel time,  $\tau$ , for the vehicle passing the detector can be obtained from the equation of motion:

$$x(t) - s + L = v(t) \tau + \frac{1}{2} a(t, t + dt) \tau^2 \quad (4.3)$$

Let  $\Delta = [v(t)]^2 + 2 a(t, t + dt) [x(t) - s + L]$ . Solving  $\tau$  from Equation (4.3), we have:

$$\tau = \begin{cases} \frac{\sqrt{\Delta} - v(t)}{a(\cdot, t + dt)} & a(t, t + dt) \neq 0 \\ \frac{x(t) - s + L}{v(t)} & a(t, t + dt) = 0 \end{cases} \quad (4.4)$$

In the first case,  $\Delta \geq 0$  is satisfied because  $\Delta < 0$  means vehicle has decelerated to a stop and does not pass the detector. In the second case,  $v(t) > 0$  is satisfied because a stopped vehicle cannot pass the detector.

Thus the instantaneous speed for a vehicle passing the detector can be calculated by:

$$u = v(t) + a(t, t + dt) \tau \quad (4.5)$$

The speeds of individual vehicles given by Equations 4.5 are accumulated for each detector and used to calculate average speed at the detector location during the specified time period.

### **Occupancy**

Occupancy is defined as the percent of time that a given point on the roadway is occupied by vehicles. Occupancy is usually obtained from inductive loop detectors (ILD), which are currently the most popular traffic sensors for surveillance systems. When a vehicle passes the loop or is stopped within the loop, it decreases the inductance of the loop and activate the detector electronics that detect the passage or presence of this vehicle. When the vehicle leaves the loop, the detector is deactivated. The time difference between a vehicle activates and deactivates the detector is *presence time* ( $\rho$ ). The measured presence time for each vehicle is accumulated for the detector in the *Surveillance System Module* to obtain the occupancy by:

$$\theta = \frac{100}{T} \sum \rho_n \quad (4.6)$$

where  $T$  is the length of the time interval for reporting data by the point data sensors (e.g., 10 seconds, 5, 10, or 15 minutes). In order to compute occupancy

for a detector using this equation, vehicle's presence time in the detection zone needs to be calculated.

A vehicle may require one or more than one time intervals to pass a detector. The relationships between vehicle and detector have 4 cases and they are illustrated in Figure 4.12.

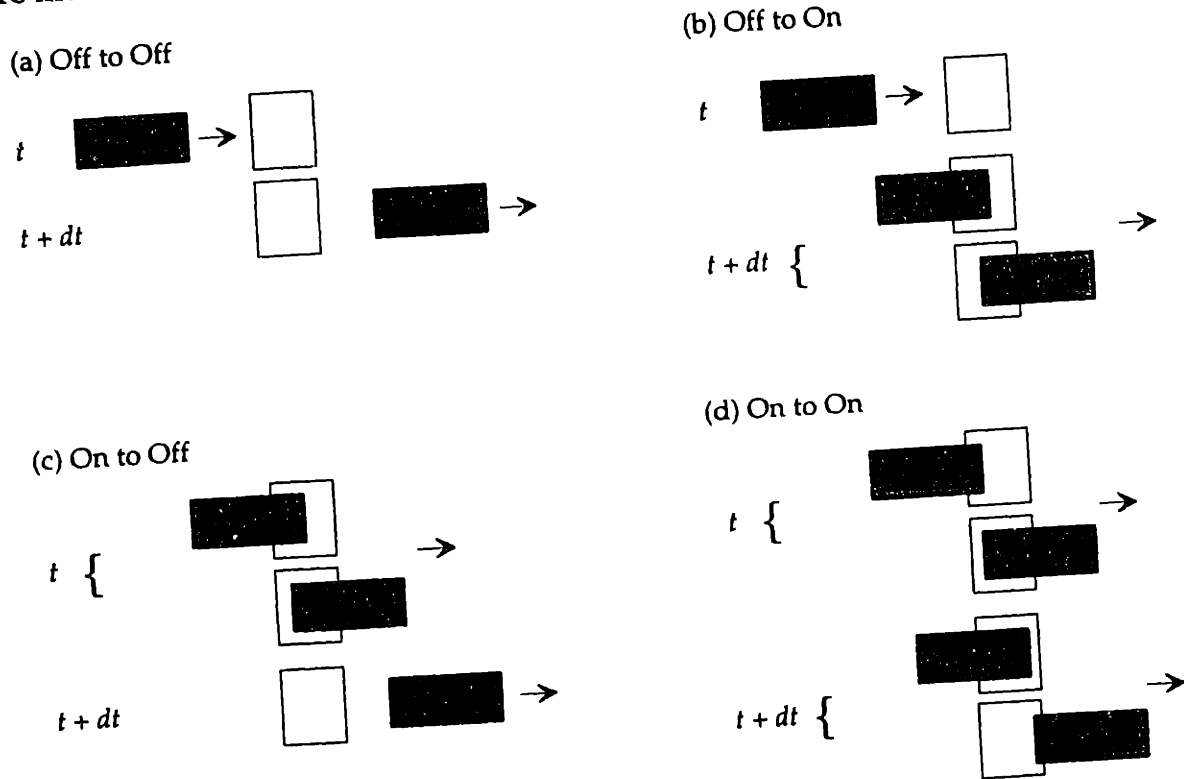


Figure 4.12 Calculation of Vehicle's Presence Time in Detection Zone

In each of the 4 cases in Figure 4.12, a vehicle moves from position  $x(t)$  to  $x(t+dt)$  in a time period  $dt$ . Let  $\omega^{on}$  be the travel time that a vehicle activates the detector, and  $\omega^{off}$  be the travel time that it deactivates the detector. If the detector has already been activated at time  $t$  (see Cases c and d in Figure 4.12),  $\omega^{on}$  is set to 0. If the detector is still active at time  $t$  (see Cases b and d in Figure 4.12),  $\omega^{off}$  is set to  $dt$ . Otherwise,  $\omega^{on}$  and  $\omega^{off}$  is calculated based on following equations:

$$x(t) - s - z = v(t) \omega^{on} + \frac{1}{2} a(t, t+dt) (\omega^{on})^2 \quad (4.7)$$

$$x(t) - s + L = v(t) \omega^{off} + \frac{1}{2} a(t, t+dt) (\omega^{off})^2 \quad (4.8)$$

Using the similar procedure for solving Equation (4.3), we can solve these two equations to obtain  $\omega^{On}$  and  $\omega^{Off}$ . Then the presence time of the vehicle in the detection zone during the time interval  $[t, t + dt]$  is:

$$\rho = \omega^{Off} - \omega^{On} \quad (4.9)$$

To compute occupancy, substitute into Equation (4.6)  $\rho$  calculated for each vehicle present in a detector's detection zone.

### Headway

Headway is defined as the time spacing between the front of successive vehicles in one lane of a road way. It is the time difference between the beginning of successive vehicle detections measured by a detector (see Figure 4.13).

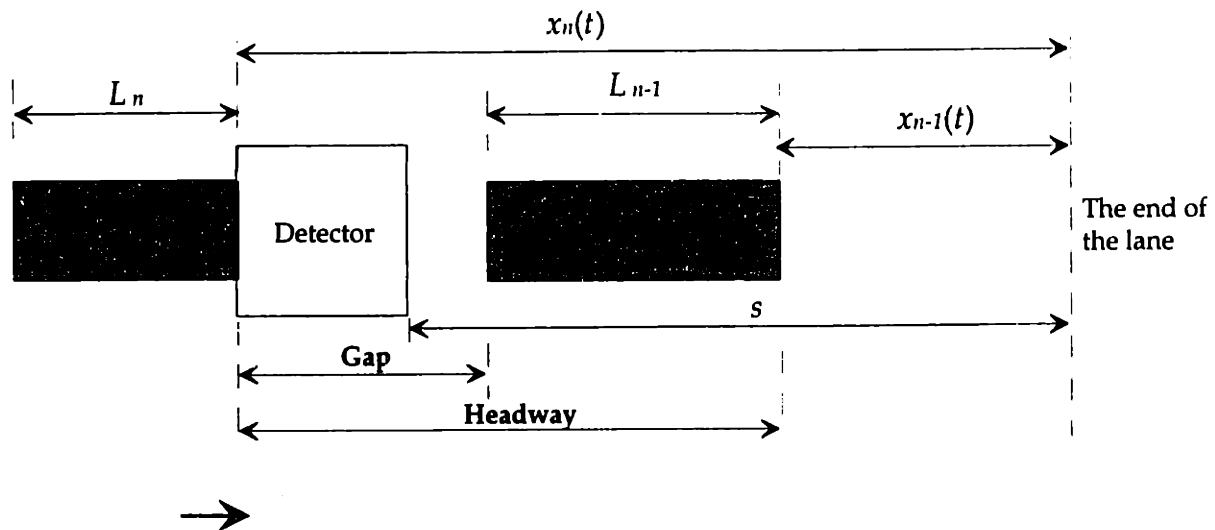


Figure 4.13 Headway Determination

### Over-height Attribute

Over-height detectors check whether a vehicle has exceeded the height limits. Vehicles traveling in the simulated network may be assigned an *overheight* attribute when they enter the network. When an over-height vehicle crosses an over-height detector, it reports the time that this vehicle crosses the detector and sends a message to the *Surveillance System Model*.

### 4.3.2 Point to Point Data Sensors

Point to point data sensors can be used to obtain the travel time between two points. A currently developed and operating point to point data sensor technology is vehicle to roadside communication (VRC), often referred to as automatic vehicle identification (AVI) technology<sup>[69]</sup>. Two methods can be used to simulate point to point data sensors:

- A vehicle equipped with a VRC transponder has (1) a pointer variable that indicates the previous VRC device it has crossed, and (2) a variable that indicates the time it crossed that device. When this vehicle crosses the next VRC device (sensor reader), the travel time between the two devices and other desired variables such as vehicle ID, speed, etc. are reported.
- When a transponder equipped vehicle moves across a VRC device, the time (current clock time), location (device ID), and vehicle identification are reported. This information is recorded or sent to the *Surveillance System Model*, and the vehicle path can be traced, and travel time calculated.

Current implementation of MITSIM uses the second method.

### 4.3.3 Area Data Sensors

Area data sensors refer to the detectors that collect, transmit and analyze video information on traffic conditions. The typical area data sensors are closed circuit television (CCTV), from which the operators at the traffic control center can observe incidents and traffic flow. Processed video information may be used to obtain density, speed and other traffic parameters.

The macroscopic characteristics of traffic flow such as speed, density, and average headways reported by area data sensors can be estimated from the detailed information generated by the microscopic simulation model. The detailed data, for example, is aggregated for the range that a camera covers, and an error term which represents the reliability of the system is added. The aggregate data may be reported to the computer screen as text or graphical output (e.g., bar charts, etc.) for selected link segments at a frequency specified by the user.

#### 4.3.4 Cellular Phone Call-in and Service Patrol

It is estimated that vehicles involved in freeway incidents can be moved, under their own power, in 75-85% of the cases<sup>[42]</sup>. However, motorists may wait for a police investigation before moving their vehicles from the travel lanes to the shoulder. Therefore, besides the automatic incident detection technologies, freeway service patrol and cellular phone call-in<sup>2</sup> are also important information sources for the identification of incidents, and may reduce incident detection time and help to identify the nature of the problem.

Simulated drivers in MITSIM are assigned a probability to report an incident when they pass an incident location. Some vehicles can be assigned as service patrol vehicles. When a vehicle equipped with cellular phone passes an incident, it invokes a procedure to examine if it will report this incident. When an incident is reported by a vehicle that has a cellular phone, or found by service patrol vehicles, the traffic simulation model sends a message to the *Surveillance System Model*, which will analyze the received information and may inform the *Control Strategy Model* to invoke incident management procedures.

#### 4.3.5 Other Sensors

Other traffic sensors may include vehicle classification, carbon monoxide, and noise detectors. These types of detectors are outside the scope of this thesis and not currently implemented.

---

<sup>2</sup> Phone numbers in the Boston area: 911 for emergency incidents, and \*99 or \*SP for non-emergency incidents.

# Chapter 5

## Vehicle Movements

MITSIM employs a discrete time based simulation approach for moving vehicles from their origins to destinations. The period to be simulated (e.g., 1 or 2 hours) is divided into short time intervals (e.g., 0.1, 0.5 or 1.0 seconds). At each time interval all vehicles in the network are processed in accordance with their desired speeds, destinations inhibited by the immediate traffic, and control environment. The process include three basic steps:

- Making necessary lane changes;
- Calculating acceleration (deceleration) rates in accordance with various constraints;
- Updating vehicles' positions and speeds.

In this chapter, the models that are used for accomplishing the above steps are presented. First the car-following and lane-changing models used in MITSIM are described. Then the simulation of incidents and toll booth operations are explained.

### 5.1 Car-following Logic

As vehicles proceed along a roadway, drivers may desire or be required to accelerate or decelerate their vehicles because of other vehicles in the traffic stream and traffic signals, and/or information received from roadside message signs. Individual vehicles in MITSIM have their own speed, acceleration (deceleration) rates, and position in the simulated network. A vehicle's movement within a lane is governed by its current speed and acceleration (deceleration) rate. The acceleration or deceleration rate of a vehicle, at any given time, is determined based on the following considerations:

- The acceleration and deceleration values observed in actual traffic conditions ( $\bar{a}_n^+$ );
- Stochastic car-following relationships in response to the lead vehicle's acceleration and deceleration ( $a_n^{CarFollowing}$ );
- Response to downstream traffic control devices and incidents ( $a_n^{SignalDetermined}$ );
- Merging and turning movements ( $a_n^{Merging}$ );
- Courtesy yielding ( $a_n^{Yielding}$ ).
- Start-up delay of a stopped vehicle;

This section describes these considerations in detail and discusses their implementation issues.

### 5.1.1 Notations and Definitions

The notations and definitions used below are summarized in Figure 5.1. Two vehicles are moving from left to right with vehicle  $n-1$  as the leading vehicle and vehicle  $n$  as the following vehicle.

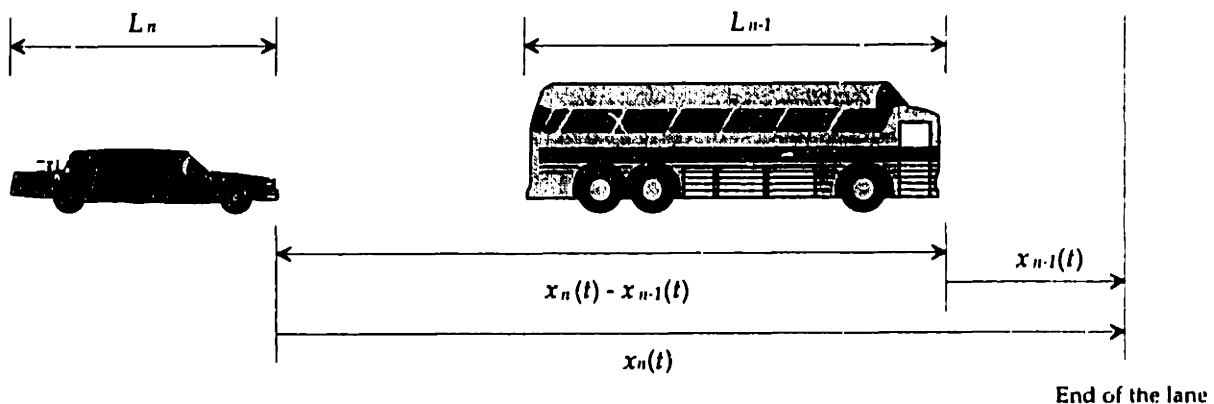


Figure 5.1 Notations and Definitions in Car-Following Models

- $x_{n-1}$  = position of lead vehicle (feet);
- $x_n$  = position of following vehicle (feet);
- $L_{n-1}$  = length of lead vehicle (feet);
- $L_n$  = length of following vehicle (feet);
- $v_{n-1}$  = speed of lead vehicle (feet/second);
- $v_n$  = speed of following vehicle (feet/second);
- $a_{n-1}$  = acceleration rate of lead vehicle (feet/second<sup>2</sup>);
- $a_n$  = acceleration rate of following vehicle (feet/second<sup>2</sup>);



- $V_{n-1}$  = desired speed of lead vehicle (feet/second);
- $V_n$  = desired speed of following vehicle (feet/second);
- $\tau_n$  = reaction time;
- $t$  = time;
- $dt$  = step size of the simulation clock, default is 1/10 seconds;
- $\omega$  = scanning interval of car-following algorithm.

The position of a vehicle ( $x$ ) is the distance from the downstream end of the lane that contains this vehicle.

The space headway at time  $t$  between the lead and following vehicle is given by  $x_n(t) - x_{n-1}(t)$  if the two vehicles are in the same link segment. When the lead vehicle  $n-1$  is in the downstream segment (relative to vehicle  $n$ ), the distance headway is given by  $x_n(t) - x_{n-1}(t) + L$ , where  $L$  is the length of the downstream segment. We assume that all segments are long enough so that the car-following model may ignore vehicles that are beyond the next segment.

The acceleration rate  $a_n$  can be positive or negative. A positive value indicates that a vehicle is accelerating and increasing its speed, while a negative value indicates the reverse.

The acceleration rate of a following vehicle  $a_n$  is specified as occurring during the time period  $[t, t + \omega]$ .  $\Delta t$  represents the interval between the time that a unique car-following situation occurs ( $t$ ) and the time that the driver of the following vehicle responds to a new car-following situation ( $t + \omega$ ).

The acceleration rate  $a_n$  is calculated for a vehicle at time  $t$  and used to move this vehicle during the time interval  $[t, t + \omega]$ , if no emergency events (e.g., too close to the lead vehicle, traffic signal changes to red, etc.) occur. Calculation of acceleration rates is performed at longer intervals than the step size of the simulation clock ( $\omega \geq dt$ ).

Each vehicle in MITSIM maintains a *timer*, which is a count down clock variable used to indicate when the previously calculated acceleration rate expires. The value of the *timer* is decreased by  $dt$  (the clock step) in every iteration or set to zero if an emergency or abnormal situation occurs. When the *timer* becomes zero, the acceleration rate is recalculated based on the new traffic environment and the *timer* is set to  $\omega$ .

## 5.1.2 Review of Representative Car-following Models

Theories describing how one vehicle follows another vehicle were developed primarily in the 1950s and 1960s. Various car-following mechanisms have been discussed in the literature [11, 28, 29, 30, 31, 32, 34, 35, 36, 37, 57, 73]. Most of them are based on the reaction delay and anti-collision concept. This concept assumes that if two vehicles are in a leader-follower relationship, the follower must be in a safe position if the leader suddenly decelerates to a stop.

By reviewing existing car-following models, we selected of them the general car-following model and the CARSIM model for further examination in MITSIM.

Next we will briefly review the two selected car-following models. Then we describe how they are extended and used in MITSIM.

### *General Car-following Model (GM)*

A non-linear car-following model developed by a group of researchers associated with General Motors [11, 19, 33, 34, 36, 77] will be described next. This model is of particular importance because of following reasons:

- Accompanying field experiments have proved its validity;
- It provides a mathematical bridge between the microscopic and macroscopic theories of traffic flow;
- It provides a general representation of car-following relationships in traffic streams;
- It can be easily applied to the MITSIM traffic simulation model with some minor modifications.

In general the car-following relationships can be represented as:

$$\text{Response} = \text{Sensitivity} \times \text{Stimulus} \quad (5.1)$$

In car-following models, the *Stimulus* is often represented by the difference in relative velocity of the lead and following vehicles, i.e.,  $v_{n-1}(t) - v_n(t)$ . Different models vary in their representation of the *Sensitivity* term, which is either a constant, or inversely proportional to the space or time headways<sup>[35]</sup>. A generalized form of this car-following model is:

$$a_n(t, t + \omega) = \alpha \frac{(v_n(t))^\beta}{(x_n(t) - x_{n-1}(t) - L_{n-1})^\gamma} (v_{n-1}(t) - v_n(t)) \quad (5.2)$$

where  $\alpha$ ,  $\beta$ , and  $\gamma$  are model parameters to be calibrated, and  $L_{n-1}$  is the effective length of the lead vehicle. This model was first proposed by Gazis *et al* [36], and further examined by May and Keller [57].

In Equation (5.2),  $\alpha \frac{[v_n(t)]^\beta}{[x_n(t) - x_{n-1}(t) - L_{n-1}]^\gamma}$  is the sensitivity term. When both  $\beta$  and  $\gamma$  are set to 1.0, the car-following sensitivity of a following vehicle is inversely proportional to the time headway from the lead vehicle. As the time headway between two vehicles becomes smaller, the sensitivity term becomes larger. The difference of velocities could be positive, negative or zero, which yields a decision of accelerating, decelerating or keeping current speed. A very interesting property of this microscopic car-following model is that it can generate several macroscopic traffic flow models when the parameters  $\beta$  and  $\gamma$  are assigned appropriate values. Table 5.1 is summary of the relationships between Equation (5.2) and some steady-state flow equations of macroscopic models.

Table 5.1 General Car-following Model and Steady-state Macroscopic Flow Equations  
 $(q = \text{flow}, k = \text{density},$   
 $k_j = \text{jam density}, k_m = \text{density at optimum flow},$   
 $u_f = \text{free flow speed}, u_m = \text{speed at optimum flow})$

$\beta$	$\gamma$	Flow Equation ( $q - k$ )	Reference
0	0	$q = \alpha_0 (1 - k / k_j)$ $\alpha_0 = q_m = 1 / \text{ReactionTime}$	Chandler <i>et al.</i> [11], Pipes [73]
	1	$q = \alpha_0 k \ln(k_j / k)$ $\alpha_0 = u_m$	Greenberg [38], Gazis <i>et al.</i> [36]
	3/2	$q = \alpha_0 k \left[ 1 - (k / k_j)^{\frac{1}{2}} \right]$ $\alpha_0 = u_f$	Drew [17]
	2	$q = \alpha_0 k (1 - k / k_j)$ $\alpha_0 = u_f$	Greenshields [39]
1	2	$q = \alpha_0 k e^{-(k/k_m)}$ $\alpha_0 = u_f$	Edie [19]
	3	$q = \alpha_0 k e^{-\frac{1}{2}(k/k_m)^2}$ $\alpha_0 = u_f$	Drake <i>et al.</i> [18]
0.8	2.8	$q = \alpha_0 k \left[ 1 - (k / k_j)^{1.8} \right]^5$ $\alpha_0 = u_f$	May <i>et al.</i> [57]

Adopted from May and Keller [57].

### CARSIM's Car-following Model

CARSIM (CAR-Following SIMulation) model was developed in 1986 by Benekohal [8]. This model extends the car-following logic used in INTRAS (*see* Appendix) in order to be used both for normal traffic flow and for stop-and-go conditions on freeways. Its extensions include:

- Randomly generated reaction time (INTRAS uses a constant value of 0.3 seconds);
- Start-up delay for stopped vehicles;

- Marginal safe spacings for individual vehicles;
- Dual behavior of traffic in congested and non-congested conditions.

One of the major components in CARSIM is the calculation of acceleration rate, which is performed for every vehicle in 1 second time intervals. The algorithm used for calculation of acceleration rates is summarized below:

1. When a vehicle is moving (or ready to move) but has not reached its speed limit or desired speed, calculate the maximum acceleration rate,  $\bar{a}_n^+$ , constrained by its mechanical ability;
2. When a vehicle has reached its speed limit or desired speed, calculate the normal (or comfortable) deceleration rate,  $a_n^-$  (see Table 5.7).
3. Calculate the acceleration rate,  $a_n^{Space}$ , which satisfies the space headway<sup>3</sup>:

$$x_n - (v_n \omega + \frac{1}{2} a_n^{Space} \omega^2) \geq K + L_{n-1} + x_{n-1} \quad (5.3)$$

where  $\omega$  is scanning time interval<sup>4</sup> and  $K$  is a buffer space. A buffer space of 10 feet was used in CARSIM when density is not very high. For near-jam density conditions, a 5-7 feet buffer space was used.

4. Calculate the acceleration rate,  $a_n^{NonCollision}$ , that satisfies the non-collision constraint:

$$x_n - (v_n \omega + \frac{1}{2} a_n^{NonCollision} \omega^2) - (v_n + a_n^{NonCollision} \omega) \tau_n - \frac{(v_n + a_n^{NonCollision} \omega)^2}{2 \bar{a}_n^-} \geq K + L_{n-1} + x_{n-1} - \frac{v_{n-1}^2}{2 \bar{a}_{n-1}^-} \quad (5.4)$$

where  $\bar{a}_{n-1}^-$  and  $\bar{a}_n^-$  are maximum (in absolute value) deceleration rates for lead and following vehicles, respectively.  $\tau_n$  is the reaction time of the following vehicle. The value of  $a_n^{NonCollision}$  is determined such that after the following vehicle moves to a new position there will be enough space

---

<sup>3</sup> Measurement of vehicle position has been translated into the same notation used in this thesis.

<sup>4</sup> Scanning time interval in CARSIM model is 1 second. Acceleration (deceleration) rates are calculated at the beginning of each interval and used for moving vehicles during the 1 second time interval.

headway for the vehicle to react to a decelerating lead vehicle, and stop or reach a safe driving speed.

5. A start-up delay is applied the vehicle which stopped and has to start from a standing still position. It is assumed that drivers with shorter reaction times will wait less than drivers with longer reaction times. In CARSIM, less than 20% of drivers have a reaction time of 0.68 seconds. These drivers will experience a delay of 1 second and the rest of the drivers will experience a delay of 2 seconds before they move again.

By solving equations (5.3) and (5.4) at equality,  $a_n^{Space}$  and  $a_n^{NonCollision}$  are obtained. To choose the actual acceleration or deceleration rate, CARSIM finds the minimum of  $\bar{a}_n^+$ ,  $a_n^-$ ,  $a_n^{Space}$  and  $a_n^{NonCollision}$ . If the value is positive, it is used as the acceleration rate; otherwise, the maximum of this value and a lower bound of 16 feet/second<sup>2</sup> is used as deceleration rate.

The research by Benekohal indicated that CARSIM's car-following logic replicates vehicle trajectories well [8]. To examine the performance of this model in simulation which allows lane-changing, CARSIM was used in a preliminary version of the MITSIM traffic simulation model. However, we found that vehicles sometimes decrease their speeds suddenly, especially in congested traffic. This causes unreasonable disturbance in the traffic. The calculation of deceleration rate in CARSIM is either based on analytical kinematic equations such as Equations (5.3) and (5.4), or the arbitrary lower bound of 16 feet/second<sup>2</sup>. These constraints do not allow temporary unsafe car-following relationships, and as a result, simulated traffic flow is not smooth. In reality a driver may temporarily accept a "unsafe" position without making emergency deceleration. To support temporary "unsafe" car-following relationships, CARSIM's non-collision constraint can not be readily applied in MITSIM without modification.

### 5.1.3 Car-following Model in MITSIM

The car-following model used in MITSIM combines some elements of GM and CARSIM models. The logic used here is presented as follows.

### *Three Regimes of Car-following Relationships*

Car-following relationships among vehicles in MITSIM are classified into three groups:

- A vehicle is far from the lead vehicle;
- A vehicle is following the lead vehicle;
- A vehicle is too close to the lead vehicle.

Each vehicle is stochastically assigned a *lower bound*  $H_n^{Lower}$  and a *upper bound*  $H_n^{Upper}$ , measured in time headway, for classifying it into one of the three categories. When the time headway,  $H_n$ , from the lead vehicle is less than  $H_n^{Lower}$ , a vehicle is said to be at an *unsafe position* and emergency decelerating is required. When  $H_n$  is greater than  $H_n^{Upper}$ , a vehicle is said to be at a *free flowing position*, and its decision on accelerating, decelerating or maintaining its current speed will depend on whether its current speed is less than, greater than or equal to the driver's desired speed. When  $H_n$  is greater than  $H_n^{Lower}$  and less than  $H_n^{Upper}$ , a vehicle is said to be in *car-following position*, and the general car-following model as described by Equation (5.2) is used for calculating the acceleration rate.

The lower and upper bounds,  $H_n^{Lower}$  and  $H_n^{Upper}$  respectively, are random variables and their distribution may depend on current local traffic condition (i.e., congestion level). The normal probability density functions are used to generate these bounds. Values of the parameters in these functions may be specified by the user. The default values used in the model are given in Table 5.2. However, these parameters are assumed values and further calibration and validation are required.

Table 5.2 Car-following Lower and Upper Bounds

	Density (vehicles per lane per hour)	
	$\leq 80$	$> 80$
Lower Bound	$N(0.50, 0.04)$	$N(0.50, 0.04)$
Upper Bound	$N(4.00, 1.00)$	$N(2.00, 0.25)$

$N(\mu, \sigma^2)$  is the PDF of normal distribution, where  $\mu$  is the mean, and  $\sigma^2$  is the variance.

**Free Flowing Acceleration Rate ( $H_n > H_n^{Upper}$ )**

If there is no vehicle ahead, or the time headway to the lead vehicle is larger than the car-following upper bound  $H_n^{Upper}$ , acceleration rate is determined according to the driver's desired speed. In pseudo code, this can be written as:

1. **if** ( $v_n < V_n$ ), **set**  $a_n^{CarFollowing} = \bar{a}_n^+$ .
2. **else if** ( $v_n > V_n$ ), **set**  $a_n^{CarFollowing} = a_n^-$ .
3. **else set**  $a_n^{CarFollowing} = 0$ .

where:

- $a_n^{CarFollowing}$  = acceleration rate determined by car-following relationships;
- $\bar{a}_n^+$  = maximum acceleration rate, a function of speed and type of the vehicle and grade of the roadway;
- $a_n^-$  = normal deceleration rate, a function of speed and type of the vehicle;
- $V_n$  = desired speed.

In other words, it is assumed that a driver at a "slower" speed will accelerate his vehicle to the desired speed as soon as possible if there is no interaction with other vehicles. If one's speed is higher than his desired speed, he will decelerate his vehicle to the desired speed using the normal (comfortable) deceleration rate. For vehicles that are traveling at their drivers' desired speeds, neither accelerating nor decelerating is required. Desired speed is defined in Chapter 4 (see 4.2.3 VSLS). Maximum acceleration rate and normal deceleration rate are discussed below.



*Maximum Acceleration Rates ( $\bar{a}_n^+$ ):* The MITSIM model assumes vehicles will accelerate to their desired speeds as fast as possible while satisfying all safety and operational constraints. The maximum acceleration rates, which are functions of weight-to-horse-power ratios, grades, and running speeds, provide upper bounds for vehicle accelerating capability. One of the most comprehensive summaries of previous research and field studies on vehicles' maximum acceleration rates can be found in [40]. Maximum acceleration rates for each type of vehicle are read from a file, in which the values used in INTRAS [21, 92] are provided as default (see Tables 5.3-5.6).

Table 5.3 Maximum Acceleration Rates for High Performance Passenger Cars (ft/sec<sup>2</sup>)

Grade	Speed (ft/sec)									
	Freeway					Non-Freeway				
	10	30	50	70	>80	10	30	50	70	>80
-4%	15.0	14.0	14.0	8.0	5.0	9.0	6.0	5.0	5.0	5.0
0%	11.0	11.0	10.0	5.0	3.0	7.0	5.0	3.0	3.0	3.0
2%	10.0	10.0	8.0	4.0	2.0	7.0	5.0	3.0	2.0	2.0
4%	9.0	9.0	5.0	2.0	1.0	6.0	5.0	2.0	2.0	1.0
6%	9.0	9.0	4.0	2.0	1.0	6.0	4.0	1.0	1.0	1.0

Table 5.4 Maximum Acceleration Rates for Low Performance Passenger Cars (ft/sec<sup>2</sup>)

Grade	Speed (ft/sec)									
	Freeway					Non-Freeway				
	10	30	50	70	>80	10	30	50	70	>80
-4%	8.0	8.0	8.0	5.0	3.0	5.0	4.0	3.0	3.0	3.0
0%	6.0	6.0	6.0	3.0	2.0	4.0	3.0	2.0	2.0	2.0
2%	6.0	6.0	5.0	2.0	1.0	4.0	3.0	2.0	1.0	1.0
4%	5.0	5.0	3.0	1.0	1.0	3.0	3.0	1.0	1.0	1.0
6%	5.0	5.0	3.0	1.0	1.0	3.0	3.0	1.0	1.0	1.0

**Table 5.5 Maximum Acceleration Rates for Buses and Heavy Single-Unit Trucks (ft/sec<sup>2</sup>)**

Grade	Speed (ft/sec)									
	Freeway					Non-Freeway				
	10	30	50	70	>80	10	30	50	70	>80
-4%	3.0	2.0	2.0	2.0	1.0	3.0	2.0	2.0	2.0	1.0
0%	3.0	2.0	1.0	1.0	1.0	3.0	2.0	1.0	1.0	1.0
2%	2.0	1.0	1.0	1.0	1.0	2.0	1.0	1.0	1.0	1.0
4%	1.0	1.0	1.0	1.0	0.0	1.0	1.0	1.0	1.0	0.0
6%	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0

**Table 5.6 Maximum Acceleration Rates for Trailer Trucks (ft/sec<sup>2</sup>)**

Grade	Speed (ft/sec)									
	Freeway					Non-Freeway				
	10	30	50	70	>80	10	30	50	70	>80
-4%	2.0	2.0	2.0	1.0	1.0	2.0	2.0	2.0	1.0	1.0
0%	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
2%	1.0	1.0	1.0	0.0	0.0	1.0	1.0	1.0	0.0	0.0
4%	1.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0
6%	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0

*Normal Deceleration Rates ( $a_n^-$ ):* The normal deceleration rate is the average deceleration rate when drivers are not influenced to react rapidly. The normal deceleration rates used in MITSIM are a function of vehicle type (cars, buses, trucks, etc.) and speed, which are provided in an input file to the program. The default values of normal deceleration rates for passenger cars (both high and low performance cars) are adopted from the *Transportation and Traffic Engineering Handbook* [40] (see Table 5.7). For buses and trucks, the values given in Table 5.7 are multiplied by a factor of 0.75.

Table 5.7 Normal Deceleration Rates for Passenger Cars (feet/second<sup>2</sup>)

Speed Change	0-15	15-30	30-40	40-50	50-60	≥ 70
$a_n^-$	7.77	6.74	4.84	4.84	4.84	4.84

Source: *Transportation and Traffic Engineering Handbook* [40].

**Emergency Deceleration Rate ( $H_n < H_n^{Lower}$ )**

If the time headway to the lead vehicle is less than a driver's car-following lower bound  $H_n^{Lower}$ , emergency deceleration is applied to the following vehicle. The emergency deceleration rate is determined by the maximum (in absolute value) of the normal (comfortable) deceleration rate ( $a_n^-$ ) and the deceleration rate given by non-collision constraint. The non-collision constraint is:

$$\begin{aligned}
 x_n - \left( v_n \omega + \frac{1}{2} a_n^{NonCollision} \omega^2 \right) - K_n - L_{n-1} \\
 \geq x_{n-1} - \left( v_{n-1} \omega + \frac{1}{2} a_{n-1} \omega^2 \right)
 \end{aligned}
 \tag{5.5}$$

where:

- $a_n^{NonCollision}$  = non-collision deceleration rate to be calculated for the following vehicle  $n$ ;
- $a_{n-1}$  = current acceleration rate of the lead vehicle;
- $\omega$  = scanning time interval (e.g., 1 second);
- $K_n$  = buffer space for vehicle  $n$ ;
- $L_{n-1}$  = length of the lead vehicle.

Equation (5.5) ensures that the following vehicle will not crash into the lead vehicle after an interval of time  $\omega$ . Buffer space  $K_n$  is randomly assigned a value from 0 to 12 feet according to the distribution in Figure 5.2.

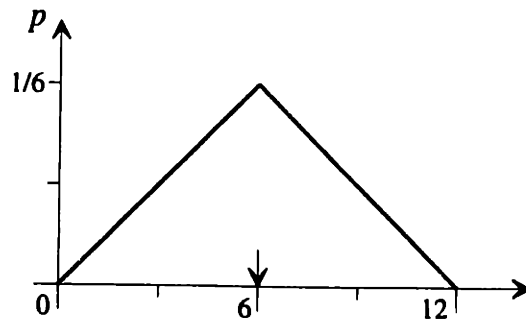


Figure 5.2 Buffer Space,  $K_n$ , between Following and Lead Vehicles

Setting non-collision constraint at equality, we obtain:

$$a_n^{CarFollowing} = a_{n-1} + \frac{2[(x_n - x_{n-1} - K_n - L_{n-1}) - (v_n - v_{n-1})\omega]}{\omega^2} \quad (5.6)$$

### ***Car-following Acceleration Rate ( $H_n^{Lower} \leq H_n \leq H_n^{Upper}$ )***

When the time headway between two vehicles is within the following vehicle's *car-following range* ( $H_n^{Lower} \leq H_n < H_n^{Upper}$ ), the interaction with the leading vehicle dominates the following vehicle's decision on accelerating, decelerating or no changing of speed. In this case, the acceleration rate ( $a_n^{CarFollowing}$ ) for the following vehicle is calculated using Equation (5.2).

The use of the general car-following model (5.2) is limited to the range  $H_n^{Lower} \leq H_n < H_n^{Upper}$  because, if it is applied universally, a following vehicle will not accelerate or decelerate as long as it has the same speed as the lead vehicle, no matter what the distance is between the two vehicles.

The values of the parameters used in Equation (5.2) are input from the user of MITSIM (default values are  $\alpha = 1.25$  and  $\beta = \gamma = 1.0$ , same for all facilities and traffic conditions). As listed in Table 5.1, different values for these parameters correspond to different steady-state flow equations, and will yield different macroscopic characteristics of traffic flow. Appropriate values may depend on facility type (e.g., freeways, urban streets, tunnels, etc.) and flow conditions (e.g., normal, congested, etc.). Further research (e.g., calibrating with field data) is required in order to select the most appropriate values.

## **5.1.4 Response to Traffic Signals**

Traffic control devices are implemented as a linked list. In this list all devices are sorted by their positions (from upstream to downstream). Traffic signals are specific objects in this list and they establish the right of way for vehicle movements. To consider a vehicle's response to traffic signals, the following two steps are used:

- Examine the downstream traffic signals that are within a certain range (e.g., 500 feet). If signal  $i$  is invisible (i.e., distance from the signal is greater than the signal's visibility), unused or green, skip it; otherwise, calculate the acceleration rate ( $a_n^{Signal_i}$ ) in response to the status of traffic signal  $i$ .
- Select the minimum of the acceleration rates calculated in the previous step as the *signal determined acceleration rate*, i.e.:

$$a_n^{SignalDetermined} = \min_i \{ a_n^{Signal_i} \} \quad (5.7)$$

For a traffic signal located at a distance of  $d_i$  from the vehicle, the acceleration rate ( $a_n^{Signal_i}$ ) is calculated by:

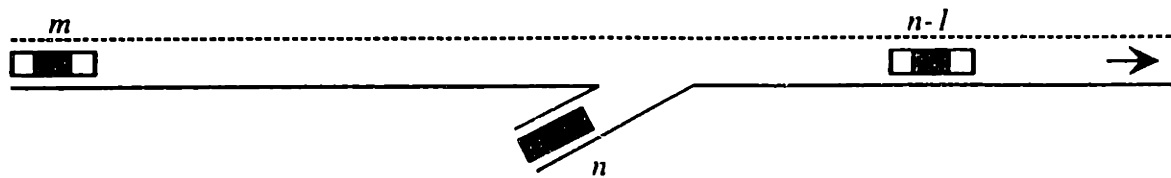
$$a_n^{Signal_i} = \frac{(v_n + V_i^{Target})(v_n - V_i^{Target})}{2 d_i} \quad (5.8)$$

where  $V_i^{Target}$  is the speed regulated by the traffic signal  $i$ . Equation (5.8) will yield an acceleration rate for reaching speed  $V_i^{Target}$  at the signal position.

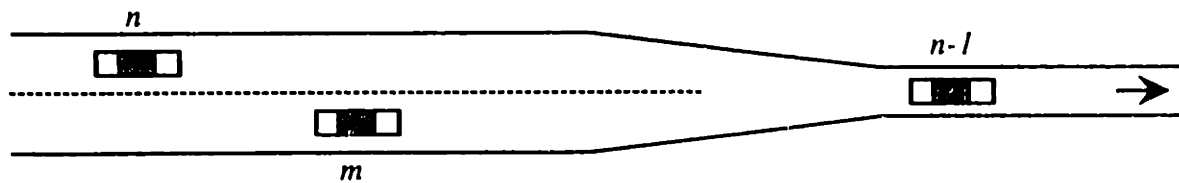
### 5.1.5 Merging

Figure 5.3 shows several cases where two or more upstream lanes are joined into a single downstream lane. When two or more lanes are connected to a single lane, vehicles from upstream lanes may need to compete for use of the downstream lane, and safe merging occurs. For driver  $n$  in Figure 5.3 (case a, b, or c), decision on accelerating, decelerating or maintaining current speed depends not only on the current lead vehicle ( $n-1$ ), but also on the vehicles in other lanes that have the right of way (e.g., vehicle  $m$ ).

a) On-ramp



b) Lane merge



c) Intersection

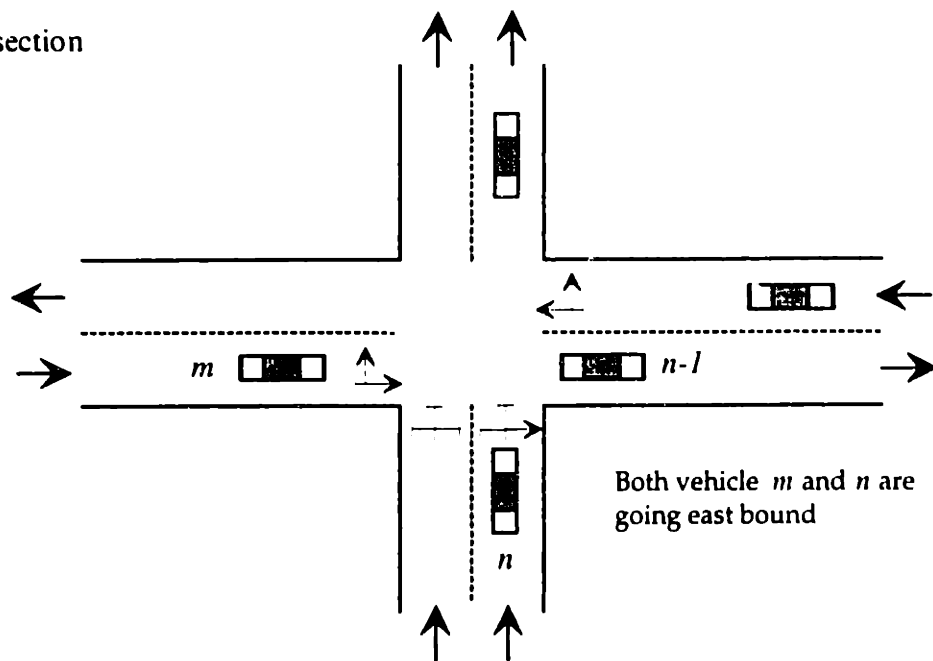


Figure 5.3 Merging

### Merging Range

When a vehicle is close to the end of a lane which merge with other lanes into a single downstream lane, a vehicle is said to be in merging range. The criteria used to characterize a distance as "close" is given by:

$$y_n \leq \frac{v_n^2}{2 a_n^-} \tag{5.9}$$

Within a distance of  $y_n$ , the vehicle with speed  $v_n$  has to use a deceleration rate larger than its normal deceleration rate  $a_n^-$  if it has to stop at the end of the lane.

### ***Merging Acceleration Rate***

A merging algorithm is used to calculate the merging acceleration rate ( $a_n^{Merging}$ ) for a merging vehicle. In pseudo code this algorithm is described as follows:

1. **if** (Vehicle  $n$  has the right of way (e.g., on freeway)) **then**
2.     **if** (Vehicle  $n$  is the first vehicle) **then**
3.          $a_n^{Merging} = \bar{a}_n^+$ ;                             // Maximum acceleration rate
4.     **end**
5.     **else**
6.          $a_n^{Merging} = a_n^{CarFollowing}(m)$              // Car-following acceleration rate with vehicle  $m$  as lead
7.     **end**
8. **end**
9. **else**
10.     **if** (Gap is acceptable) **then**
11.          $a_n^{Merging} = \bar{a}_n^+$                              // Maximum acceleration rate
12.     **end**
13.     **else**
14.          $a_n^{Merging} = -\frac{v_n^2}{2 x_n}$                      // Decelerating to a stop
15.     **end**
16. **end**

Lines 1-8 apply to vehicles which have the right of way (e.g., on freeways, streets with green light at signalized intersections, and streets without stop signs at unsignalized intersections). Lines 9-16 apply to vehicles which do not have the right of way (e.g., merging from on-ramp into freeway, on streets with yield or stop signs). The calculation of car-following acceleration rate in line 6 utilizes Equation (5.2) with vehicle  $m$  as the lead vehicle. In line 10, a gap is acceptable if the following constraint is satisfied (*see* Figure 5.4):

$$g_m \geq \eta_n + \xi_n \tag{5.10}$$

where:

- $g_m$      = time headway to the downstream end of the lane;
- $\eta_n$      = expected travel time before merging;
- $\xi_n$      = merging headway buffer.

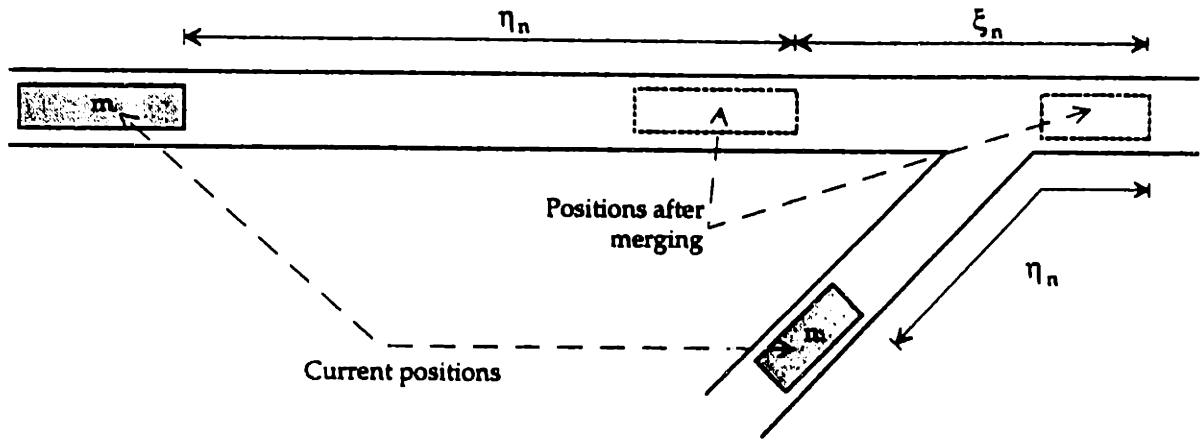


Figure 5.4 Acceptable Merging Time Headway

Merging headway buffer ( $\xi_n$ ) is a random variable (e.g., uniformly distributed between 1.5~3.0 seconds). The expected travel time before merging ( $\eta_n$ ) is the travel time required to move to the beginning of the downstream lane, and it can be calculated based on:

$$x_n + L_n = v_n \eta_n + \frac{1}{2} \bar{a}_n^+ \eta_n^2 \quad (5.11)$$

where  $x_n$  is the distance from the end of current lane, and  $\bar{a}_n^+$  is the maximum acceleration rate given the speed is  $v_n$ . From Equation (5.11) it follows:

$$\eta_n = \frac{\sqrt{v_n^2 + 2 \bar{a}_n^+ (x_n + L_n)} - v_n}{\bar{a}_n^+} \quad (5.12)$$

Equation (5.12) provides an approximation to the time that vehicle  $n$  will take before it merges into the downstream lane, assuming that maximum acceleration is applied.

### 5.1.6 Courtesy Yielding

Courtesy yielding applies to the situation where a driver either decelerates or accelerates but uses a lower acceleration rate than he/she could use in order to permit another driver to merge into his/her lane. In Figure 5.5 driver  $m$  wants to change to the right lane in order to exit at the next off-ramp. According to the car-following relationship between vehicles  $n$  and  $l$ , the following vehicle may maintain its current speed or even accelerate to a higher speed. However, vehicle  $n$  may decelerate or not accelerate in order to allow vehicle  $m$  to merge



into the right lane. The acceleration rate calculated from this type of yielding is called *courtesy yielding acceleration rate* ( $a_n^{Yielding}$ ).

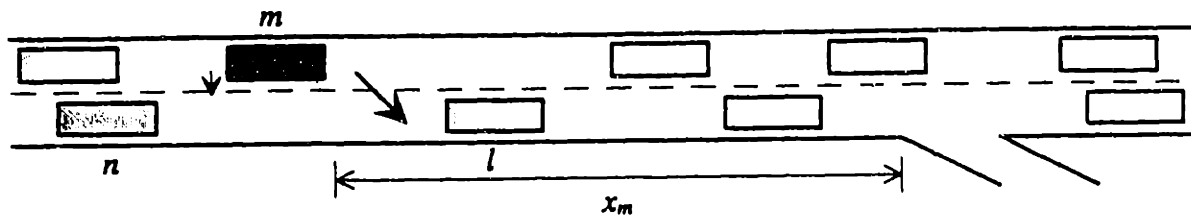


Figure 5.5 Courtesy Yielding

Since not every driver will yield to others, courtesy yielding is treated as a probabilistic event. MITSIM uses two probabilities to determine if a driver will yield to another driver who requires a lane change:

$p_{Yielding}^{Discretionary}$  : probability a driver yields to discretionary lane changing (e.g., 0.5).

$p_{Yielding}^{Mandatory}$  : probability a driver yields to mandatory lane changing (e.g., 0.9);

where  $0.0 \leq p_{Yielding}^{Discretionary} \leq p_{Yielding}^{Mandatory} \leq 1.0$  (see Section 5.2 lane-changing logic);

for the definitions of discretionary and mandatory lane changing). Probability of yielding to a mandatory lane changing is higher than that to a discretionary lane changing because the yielding driver is more aware of the lane change situation.

The algorithm for calculating the courtesy yielding acceleration rate ( $a_n^{Yielding}$ ) is described as follows:

- 1 Check whether the left-front and right-front vehicles (if any) of driver  $n$  require lane changes. If not, skip the courtesy yielding algorithm and return; otherwise, find whether driver  $n$  yields to the vehicle ( $m$ ) that requires the lane changing based on the type of lane changing and the probabilities  $p_{Yielding}^{Discretionary}$  and  $p_{Yielding}^{Mandatory}$ , then go to step 2.
- 2 If driver  $n$  does not yield to other vehicles, skip the courtesy yielding algorithm and return; otherwise, set  $a_n^{Yielding}$  to the car-following acceleration rate using vehicle  $m$  as the lead vehicle.

### 5.1.7 Select a Proper Acceleration Rate

Once all the acceleration or deceleration rates subject to the constraints described above have been calculated, a vehicle's acceleration rate is determined by:

$$a_n = \min \{ a_n^{CurFollowing}, a_n^{SignalDetermined}, a_n^{Merging}, a_n^{Yielding}, \bar{a}_n^+ \} \quad (5.13)$$

In other words, a vehicle's acceleration rate is determined by the most restrictive constraint (including its mechanical ability for accelerating). The acceleration rate calculated from Equation (5.13) will be used by vehicle  $n$  during the next interval of time.

### 5.1.8 Start-up Delay

When a platoon of vehicles is subjected to a kinematic disturbance, the speed of vehicles will decrease gradually and finally they may come to a complete stop. After a lead vehicle moves, the driver of the following vehicle may not perceive this change immediately. In other words, a stopped vehicle incurs a *start-up delay*.

The simulation of start-up delay works as follows:

- Each time a vehicle decelerates to a stop (the speed in the previous iteration is positive, but becomes zero in this iteration), an attribute in its status variable is set to "*stopped*".
- When a stopped vehicle obtains an acceleration rate ( $a_n > 0$ ), the attribute "*stopped*" is cleared from its status variable. Then this vehicle is randomly assigned a start-up delay of 0.0 to 3.0 seconds.
- If a vehicle does incur a start-up delay, its acceleration rate is set to zero ( $a_n = 0$ ), and the variable *timer* is set to a value equal to the start-up delay;
- For a vehicle that does not incur start-up delay (not '*stopped*' or start-up delay returned in step 2 is zero) the variable *timer* is set to  $\omega$ , the step size for updating acceleration rate. Then the acceleration rate calculated by Equation (5.13) is returned.

### 5.1.9 Update Speed and Position

After acceleration rates are calculated for all the vehicles in the network, their speed and position at time  $t+dt$  are updated based on their speeds and positions

at time  $t$ , and acceleration rates  $a_n(t, t + \omega)$  for the interval of time  $[t, t + \omega]$ . The updating of vehicles' speeds and positions use the following two equations:

$$v_n(t + dt) = \max \{ \min \{ v_n(t) + a_n(t, t + \omega) dt, V_n \}, 0 \} \quad (5.14)$$

$$x_n(t + dt) = \max \{ x_n(t) - 0.5 [v_n(t) + v_n(t + dt)] dt, 0 \} \quad (5.15)$$

where:

- $v_n$  = speed, in feet/second;
- $x_n$  = distance from the end of the lane, in feet;
- $a_n$  = acceleration rate, in feet/second<sup>2</sup>;
- $V_n$  = desired speed, in feet/second.

Notice that in Equation (5.14) the speed is limited to be non-negative and less than or equal to the driver's desired speed. When the distance obtained from Equation (5.15) is zero, the vehicle is either moved into its downstream lane or removed from the network if it has arrived at its destination.

## 5.2 Lane-changing Logic

Vehicles in the simulated network may require lane changes in order to:

- move to the *next links* on their paths;
- shift away from lanes that are not allowed for their use according to lane use regulations;
- pass slow lead vehicles to increase speed;
- bypass incidents and closed lanes.

Two types of lane changing behavior are modeled in MITSIM. Figure 5.6 shows a situation where vehicle  $a$  goes to exit  $A$ , vehicle  $b$  goes to exit  $B$ , and vehicle  $c$  and others continue on the freeway to  $C$ . In order to move to the correct downstream link, vehicle  $a$  must make two lane changes to the right, and vehicle  $b$  must make one lane change to the left. This type of lane change is mandatory because it is necessary for the subject vehicles. The second type of lane changing is discretionary. An example is shown by vehicle  $c$  in Figure 5.6,

where the driver is trying to change to the left lane because he is not satisfied with the speed in the central lane and he can drive faster in the left lane.

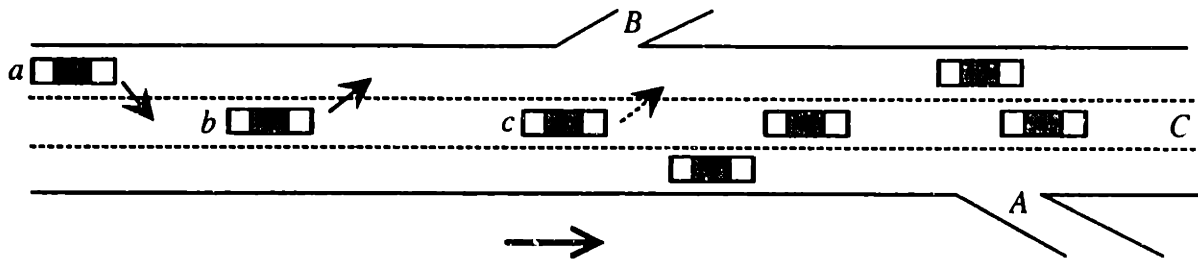


Figure 5.6 Mandatory and Discretionary Lane Changes

The lane-changing algorithm used in MITSIM model includes three steps:

1. examining if a vehicle desires a lane-changing;
2. examining if it is possible to make this change when a lane-changing is desired;
3. moving the vehicle from its current lane to the target lane when a lane-changing is desired and possible.

Periodically each vehicle in the network invokes lane-changing algorithm to determine whether it should make lane changes, and if yes, perform the lane changes. The frequency of invoking the lane changing algorithm is specified differently for mandatory and discretionary lane changes. If a vehicle has started a mandatory lane changing, the first step of above algorithm is skipped and it invoke the second and third step in every iteration of the simulate ( $dt$ ); otherwise, it invokes the algorithm at a user specified frequency ( $\zeta \geq dt$ ). The default value for  $\zeta$  is 1.0 second.

In the followings, the logic and implementation of the lane changing algorithm used in the simulation model are discussed in detail.

### 5.2.1 Lane Change Status Variable

For implementation purpose, six bits of an integer variable  $\kappa_n$  is used to indicate if a vehicle needs a lane change and its current lane changing status. The first 3 bits (bit 0-2) of  $\kappa_n$  indicate if the right, current, and left lanes are the correct lane for this vehicle (in terms of getting to the next link on its path to the destination).

The value for these 3 bits is defined as follows (the value in parentheses is in binary):

1. target lane is right lane (000001)
2. stay in current lane (000010)
3. stay in current lane or move to right lane (000011)
4. target lane is left lane (000100)
6. stay in current lane or move to left lane (000110)
7. stay in current lane or move to either left or right lane (000111)

A vehicle will have one of these 6 values at a given time. The next 2 bits (bits 3 and 4) of  $\kappa_n$  indicate the current status of lane changing, and may have the following values:

0. no lane changing (000000)
8. changing to the right (001000)
16. changing to the left (010000)

Finally, a value of 32 (100000) is used to indicate if the lane change is mandatory or discretionary. If the lane change (or "stay in lane") is mandatory instead of discretionary, bit 5 of  $\kappa_n$  is set to 1. Whenever a vehicle enters a new lane (changing to the left or right lane, or moving into a downstream lane), bits 3-5 are set to zeros. Bits 0-2 of  $\kappa_n$ , which indicate whether the right, current, and left lanes are correct for the vehicle, are set based on whether they are the correct lane for the vehicle. As a vehicle moves within a lane, the first 3 bits (bits 0-2) never change, but the next 3 bits (bits 3-5) may be set or cleared by the lane changing algorithm as described in the next subsection.

## 5.2.2 Determination of Mandatory Lane Changing

Mandatory lane changing applies to vehicles whose current lanes are not aligned to their next links (bit 1 of  $\kappa_n$  is 0). In Figure 5.6, for example,  $\kappa_a$  for vehicle *a* may have a value of  $32+8+1$  (41 in decimal, or 101001 in binary), indicating only the right lane is correct (1) for vehicle *a* and this vehicle is making a mandatory (32) right lane changing (8). Similarly,  $\kappa_b = 32 + 16 + 4$  (52 in decimal, or 110100 in binary),  $\kappa_c = 0 + 16 + 7$  (23 in decimal, or 010111 in binary). Note that because both vehicles *a* and *b* are not in the correct lanes and have to change lanes to reach their destinations, the second bits (bit 1) are zero for these two vehicles.

Vehicle  $c$  is in the correct lane, and the driver has the freedom of selecting any of the three lanes (right, current, and left), although the driver chooses to switch to the left lane in order to increase speed.

A driver may not start mandatory lane changing until he/she is "near" to the end of a link, especially if the link is relatively long. In the simulation model, the individual's decision for mandatory lane changing is made stochastically based on the probability given by the following equation:

$$p_n = \begin{cases} \exp(-\rho x_n) & \rho > 0 & \text{If } x_n \geq \delta \\ 1 & & \text{Otherwise} \end{cases} \quad (5.16)$$

where  $x_n$  is the distance from the downstream node,  $v_n$  is the current speed, and  $\rho$  and  $\delta$  are parameters to be calibrated. This equation assumes that the probability for a driver to make mandatory lane changing increases exponentially as his/her distance from the downstream node decreases; when the distance is less than  $\delta$ , this probability is set to 1 (see Figure 5.7).

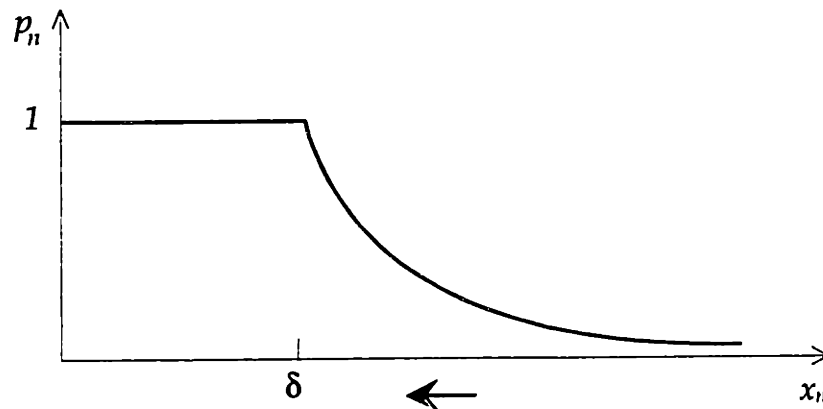


Figure 5.7 Probability of Starting Mandatory Lane Changing

The following process is used to determine whether a vehicle requires a mandatory lane change:

1. **if**  $((\kappa_n \& 32) == 0)$  **then** // This vehicle is not in a mandatory lane changing
2. Calculate probability of making mandatory lane changing,  $p_n$ , using Equation 5.16
3. Generate a random number  $r_n$  uniformly distributed between  $[0,1)$
4. **if**  $(p_n \geq r_n)$  **then**
5.  $\kappa_n \mid= 32$  // A mandatory lane changing (or stay in lane) is required

- 6.     end
- 7.     end

As a vehicle moves closer to the downstream node, the higher the probability to set the vehicle in mandatory lane changing or staying in lane status. When a vehicle is not in mandatory lane changing or staying in lane status, it may change to any lane as long as the change to that lane is allowed by the regulation; otherwise, the lane(s) that this vehicle can change to or stay in must be the correct lanes (corresponding bit in  $\kappa_n$  is 1).

If the current lane is not allowed for use according to lane use regulations, or lane use signals and message signs recommend lane changes, a vehicle also make mandatory lane changes with prespecified probabilities.

### 5.2.3 Determination of Discretionary Lane Changing

A driver may change lanes to increase speed if he/she is following a slow driver. Lane changing for mainly increasing speed is called discretionary lane changing. Discretionary lane changing behavior is simulated by considering the individual's desired speed and the tolerance of following a slow lead vehicle.

The procedure for checking discretionary lane changing is invoked for a vehicle when the following conditions are satisfied:

- Current speed of the vehicle is too low:

$$v_n < \zeta_n V_n \tag{5.17}$$

where  $\zeta_n < 1.0$  is the impatience factor, a random variable that describes the driver's lane changing behavior. When this condition is true, the driver's current speed is considerable lower than his/her desired speed.

- Current acceleration (deceleration) rate is considerably lower than the maximum acceleration rate:

$$a_n \leq \psi \bar{a}_n^+ \tag{5.18}$$

where  $\psi$  is a parameter (e.g.,  $\psi = 0.25$ ). This equation means that the vehicle cannot seriously accelerate if it stays in the current lane.

- Lead vehicle is not accelerating:

$$a_{n-1} \leq 0 \tag{5.19}$$

- The vehicle is not going to a required stop at a traffic control device.

If the above conditions are satisfied, a driver may try to change lanes to increase his/her speed.

### 5.2.4 Target Lane

If a lane change is required, the lanes that the driver can select are determined differently for mandatory and discretionary lane changes:

- Drivers that are currently committing mandatory lane changing can only select among the correct lanes;
- Drivers that are currently committing discretionary lane changing may select any lanes (stay in lane or change to right or left).

To select an appropriate target lane, two more conditions are further examined:

- Changing from current lane to the target lane is allowed according to the lane changing regulations.
- Speed can be significantly increased if the driver shifts into that lane, i.e.:

$$\min \{ a_n^{CurFollowing}, a_n^{SignalDetermined} \} > \psi \bar{a}_n^+ \quad (5.20)$$

where  $\psi$  is the same parameter used in equation 5.18. This equation ensures that both *car-following acceleration rate* ( $a_n^{CurFollowing}$ ) and *signal determined acceleration rate* ( $a_n^{SignalDetermined}$ ) in the new lane are much higher than one's maximum acceleration rate.

If more than one lane is available, a driver will select the one that results in the highest acceleration rate as the *target lane*.

### 5.2.5 Acceptable Gaps in Lane Changing

When a driver desires to shift to the left or right neighbor lane, he/she needs to check whether the lead and lag gaps are large enough for him/her to make this lane change. Figure 5.8 shows that driver  $n$  wishes to move into the gap between driver  $n-1$  and  $n+1$  in the left adjacent lane. To make this lane change, driver  $n$  checks the vehicles in the target lane to assure that both the lead gap and lag gap are acceptable according to his/her risk margins.



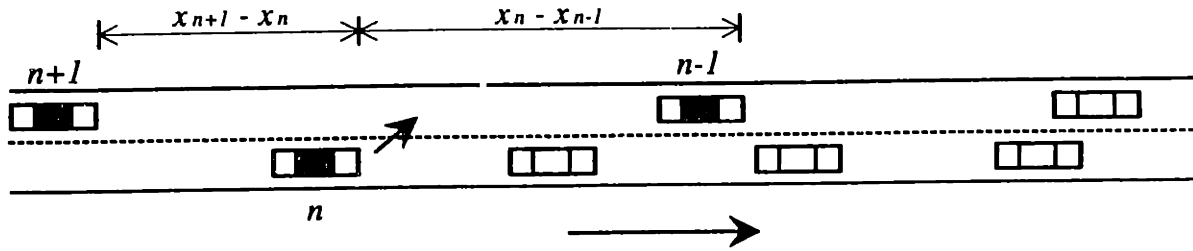


Figure 5.8 Lead and Lag Gaps in Lane Changing

The lead and lag gaps are acceptable for driver  $n$  to make a lane changing if the following constraints are satisfied:

$$x_n - x_{n-1} - L_{n-1} \geq v_n h_n^{lead} \quad (5.21)$$

$$x_{n+1} - x_n - L_n \geq v_{n+1} h_n^{lag} \quad (5.22)$$

where the subscript  $n$  represents the vehicle that wishes to change lanes,  $n-1$  represents the lead vehicle and  $n+1$  represents the following vehicle if vehicle  $n$  shift into the target lane.  $h_n^{lead}$  and  $h_n^{lag}$  are the minimum lead and lag time headways that driver  $n$  accepts in lane changing. Acceptable gaps may differ between individuals and be different for different traffic conditions. We assume that both  $h_n^{lead}$  and  $h_n^{lag}$  are normally distributed random variables (e.g.,  $h_n^{lead} \sim N(0.5, 0.25)$  and  $h_n^{lag} \sim N(2.0, 1.0)$ ), and calibration is required in order to find appropriate means and variances for them.

### 5.3 Toll Booth Operations

In the simulated network, toll booths are located at the end of a link segment, and each toll booth may have any number of toll lanes including electronic toll collection (ETC) lanes. Required information for each toll booth includes as described in Chapter 4:

- Lane use regulations (HOV only, electronic toll collection only, car only, etc.), speed, and mean delay.
- Price tables, which are implemented using a 3-dimensional matrix (Time Periods  $\times$  Origins  $\times$  Vehicle Types).

Vehicles that pass a toll booth are charged based on either their arrival times at toll booths or entry times to the system. Vehicle movements in the presence of toll facilities are still governed by the car-following and lane changing algorithms discussed in the previous sections. Several issues that arise specifically in the context of toll booth operations are discussed below.

### 5.3.1 Lane Selection

When vehicles approach a toll station, they need to select appropriate lanes. Therefore, when vehicles are in a segment with a toll booth installed at the end, additional considerations should be incorporated into their lane changing algorithm. If a lane is not suitable for a vehicle, the first 3 bits (bits 0-2) of its status variable for lane changing ( $\kappa_n$ ) will indicate which direction the vehicle should move. Then a mandatory lane changing procedure is invoked to shift this vehicle to its correct lane in subsequent iterations.

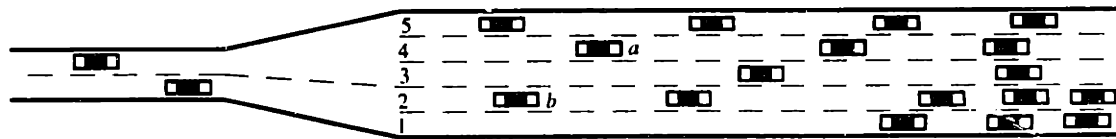


Figure 5.9 Lane Selection at a Toll Booth

Figure 5.9 illustrates a toll booth with 5 lanes. The two left lanes are assigned to vehicles equipped with ETC devices, and the other three lanes are for vehicles without ETC devices. In the figure, vehicle *a* is in the wrong lane ( $(\kappa_n \& 7) = 1$ , indicating only right lane is the correct lane), and it has to make a mandatory lane change to the right. Vehicle *b* is in the correct lane ( $(\kappa_n \& 7) = 7$ , indicating the right, current and left lanes are all correct lanes) but it may make discretionary lane change (choose the left or right lane) because the headways from the lead vehicles in lane 1 and 3 are longer.

### 5.3.2 Delay

When approaching a toll station (the end of a segment), each vehicle decelerates to the required speed if it is in an electronic toll lane or a stop if it is in a conventional toll lane. In a conventional toll lane, as soon as a vehicle stops at the booth, a randomly generated delay is assigned to the count down clock variable *timer*. The value of *timer* decreases by *dt*, the step size of the simulation

clock, in each of the consequent iterations. When *timer* becomes zero (delay has expired), the vehicle is moved into its downstream lane and starts to accelerate.

The delay,  $w_n$ , is a random variable with a negative exponential probability density function (PDF):

$$f(w_n) = \mu \exp(-\mu w_n) \quad \mu > 0, w_n > 0 \quad (5.23)$$

where  $\mu$  is the mean service rate (number of vehicles per second, equal to the inverse of the mean delay). From the PDF given in Equation (5.23) we obtain the delay that a driver may experience at the toll booth:

$$w_n = -\frac{\ln(1-r)}{\mu} \quad (5.24)$$

where  $r$  is a random number uniformly distributed between [0,1). For ETC vehicles,  $w_n$  will always be zero.

Different toll lanes may have different average service rates for each type of vehicles. The default values used in MITSIM are listed Table 5.8.

Table 5.8 Service Rates at Toll Booth (vehicles per lane per hour)

	Cars	Buses	Trucks	Others
Getting Tickets	1,400	900	900	900
Using Pass	1,400	900	900	900
Paying Exact Change	1,500	800	800	800
Other	900	600	600	600

## 5.4 Incidents

Simulated incidents are represented as objects in the linked lists of control devices and events. Each incident is described by the following information:

- Location
  - Link / Segment / Lane;
  - Offset distance ( $d$ );
  - Influenced distance ( $z$ ).
- Start time.

- Maximum duration.
- Type and severity
  0. Road construction and maintenance (00000);
  1. Vehicle break down (00001);
  2. Minor accident (00010);
  3. Major accident (00011);
  16. Lane blocked (10000).
- Maximum speed (if it is not completely blocked).
- Rubber-necking speed of vehicles in adjacent lanes.

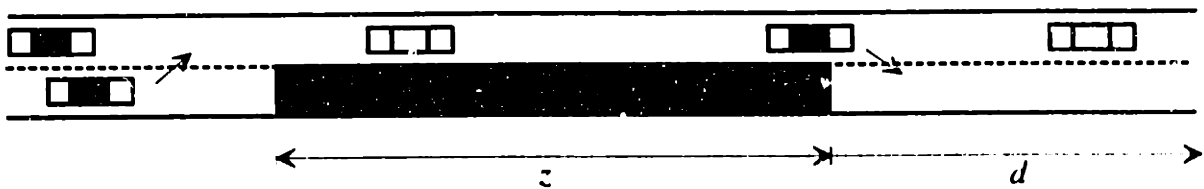


Figure 5.10 Position of an Incident

An incident can be on any lane of a segment and its position is described by offset distance ( $d$ ) and length it blocks ( $z$ ) (see Figure 5.10). If an event blocks two lanes, it is counted as two incidents. Vehicles approaching an incident have to change lanes to bypass it; otherwise they have to decelerate to a stop. When a section of lane is blocked by an incident, no lane changing into this section will be allowed by the lane changing algorithm.

Incidents are treated as exogenous events in the traffic simulation, and their descriptions are read from the incident configuration file during the setup stage of the traffic simulation. An incident can be either *inactive* (it has not occurred yet or has already been cleared) or *active* (it has occurred and has not been cleared yet). An active incident is indicated by bit 5 (corresponding to a value of 32). When the time of the simulation clock is greater than or equal to the start time of an incident, this incident is marked as active and affects the traffic by blocking a section of a lane or slowing down the traffic, depend on type and severity of the incident. The duration of an incident may be changed by the *Control Strategy Model*. If the *Control Strategy Model* never changes the duration, an incident will stay active until the maximum duration expires.

When an incident is active, vehicles moving in the neighbor lanes may have a lower speed because of rubber-necking. When an incident has been cleared, it is marked as inactive and ignored by all vehicles.

# Chapter 6

## Representation of Travel Demand

### 6.1 Origin-Destination Matrices

For the purpose of origin-destination (O-D) flows, MITSIM divides the time period to be simulated into short intervals (e.g., 15 minutes for whole period, or 10 minutes for peak period and 30 minutes for off-peak period). It assumes that within an interval the demand rates remains constant. Hence, travel demand is represented as time-dependent O-D matrix. The entries in these matrices represent the number of vehicles that are scheduled to depart from an origin and travel to a destination during the corresponding time period. Table 6.1 gives an example of O-D flow representation. This table includes two O-D tables, one for 7:00-7:30 and the other for 7:30-8:00am. The first item in each O-D table indicates the time that this O-D table expires, and the second number of O-D pairs with non-zero flow in the O-D matrix. For example, the first part of Table 6.1 indicates that O-D matrix for the period from 7:00am (the simulation starting time) to 7:30am contains 4 O-D pairs, and the information for each entry, including origin node, destination node, vehicle type, and departure rate, are listed subsequently. The second part in Table 6.1 provides similar information for the period from 7:30am to 8:00am.

Table 6.1 O-D Table from 7:00am to 8:00am

Expire Time	# of Entries	Origin	Destination	Vehicle Type	Vehicles/Hour
07:30:00	4	1	4	1	120
		1	4	4	6
		1	6	1	60
		5	4	1	60
08:00:00	5	1	4	1	240
		1	4	4	12
		1	6	1	120
		5	4	1	120
		5	6	1	60

Based on the O-D matrix, individual vehicles are generated and assigned types, origins, destinations, departure times, and pre-trip paths. These vehicles are then sorted by their departure time and stored in a temporary data file. If the simulation clock is greater than or equal to a vehicle's scheduled departure time, the vehicle enters the network and starts its travel toward to its destination. When the current O-D matrix is out-of-date, a new O-D table is read, and vehicles departing during the next time period are generated.

## 6.2 Vehicle Type

As shown in Table 6.1, travel demands between an O-D pair are given for each vehicle type. The travel demand is further segmented into individual driver-vehicle pairs of different behavior and performance. The number of vehicle types to be simulated and the performance data for each vehicle type are input parameters to the simulation model and are read from a file. For example, the following types of vehicles may be simulated:

1. High performance passenger cars (New Cars);
2. Low performance passenger cars (Old Cars);
3. Buses and heavy single-unit trucks (Buses);
4. Trailer trucks (Trucks).

Vehicles of the same type are assumed to possess common characteristics such as length, acceleration and deceleration profile, maximum and desired speed profile, etc.

Simulated vehicles are randomly assigned type prefixes that indicate whether they are equipped with electronic toll collection (ETC) devices, and their access to traffic information sources and their communication capabilities. The type prefixes used in the simulation are defined as follows:

- 128. Vehicles that have ETC devices;
- 256. Guided vehicles;
- 512. VRC transponder equipped vehicles;
- 1024. Vehicles that have cellular phones;
- 2048. Service patrol vehicles.

For example, a new car has a type of 1, while a guided new car is assigned a type of 257 (1+256). The type of a guided and VRC transponder equipped passenger car is 769 (1+256+512). The probabilities used in assigning the type prefix are provided by the user.

### 6.3 Departure Time

During each time interval, the vehicle departing rates for an O-D pair is assumed to be constant, and vehicle departures are assumed to follow a Poisson process. Under this assumption, the time interval between two subsequent departures is negative exponentially distributed, and vehicles' departure times can be computed by:

$$t_n = t_{n-1} - \frac{3600 \ln(1-r)}{\mu} \quad \mu > 0, 0 \leq r < 1, n = 1, 2, \dots \quad (6.1)$$

where:

- $\mu$  = departing rate for a given vehicles type between an O-D pair during the time period (for example, 15 or 30 minutes), in vehicles per hour;
- $r$  = a random number uniformly distributed between [0.0, 1.0);
- $t_n, t_{n-1}$  = departure time of this vehicle and previous vehicle, in seconds (from 00:00:00am);
- $t_0$  = the starting time of the period with the departing rate  $\mu$ , in seconds.

All vehicles departing during the next time period are assigned departure times by applying Equation (6.1) iteratively. This process continues for an O-D pair



until the departure time for the next vehicle is larger than the ending time of the period.

## 6.4 Route Choice

Each simulated vehicle in the network has a scheduled path that leads to its destination. Vehicles are moved from one link to the next along individuals' paths. A vehicle's remaining path can be changed according to the driver's en-route decisions. To model route choice (either pre-trip or en-route), the traffic simulation model provides three types of route choice mechanisms:

- Path enumeration;
- Shortest path;
- Stochastic and adaptive routing.

Individual vehicles can be randomly assigned to one of the three mechanisms based on predefined probabilities. These probabilities may be specified differently for guided and unguided vehicles.

### 6.4.1 Path Enumeration

Each O-D pair can be associated with a set of predefined "reasonable" paths, and the probabilities of selecting these paths. Figure 6.1 shows a simple network with 4 nodes and 4 links. Let us assume that vehicles from node 1 to 4 may use either one of the two paths listed in Table 6.2, with a probability of 0.6 and 0.4 respectively.

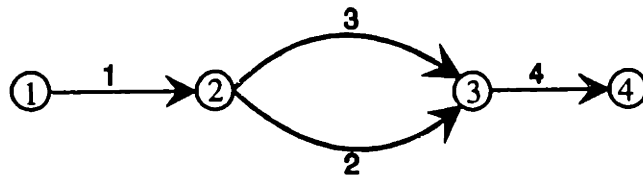


Figure 6.1 Alternative Paths in a Small Network

Vehicles, traveling between the O-D pair 1→4, are randomly assigned one of the predefined paths based on the above (probabilities observed in the field or estimated by a route choice model). In the computer implementation, each vehicle has a variable that points to the next link on its path. If the vehicle has been moved into a downstream link, this pointer variable is updated.

Table 6.2 Paths Associated with the O-D Pair from Node 1 to 4

Path	List of links on the path	Probability
1	1→2→4	0.4
2	1→3→4	0.6

### 6.4.2 Shortest Path

Vehicles are sent to the perceived shortest paths from their current downstream nodes to their destinations. The shortest path calculation can use either historical or predicted travel times (costs), depending on the system one would like to evaluate. The traffic simulation model periodically invokes a function to update link travel times (costs), and computes the least travel time (cost) path tree for each destination node. The link travel time or cost may be time dependent. In this case, a dynamic shortest path algorithm should be used. Vehicles may change their paths based on the information provided.

The shortest path trees are represented in the simulation model by a successor matrix,  $S$ , i.e.,

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1N} \\ s_{21} & s_{22} & \cdots & s_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ s_{M1} & s_{M2} & \cdots & s_{MN} \end{bmatrix} \quad (6.2)$$

where  $N$  is the number of nodes in the network,  $M$  is the number of destination nodes in the network.  $s_{ij}$  is the ID of the first link of the shortest path from node  $j$  to destination node  $i$ . If the routing algorithm to be used is dynamic, the time period to be simulated is discretized into some short intervals (e.g., 10, 15 or 30 minutes). The dynamic shortest path trees are represented by a set of time dependent successor matrices, i.e.,  $S(1)$ ,  $S(2)$ , ...,  $S(t)$ . Each of these matrices corresponds to one time interval. As in the case of path enumeration, each simulated vehicle has a variable that points to the next link it will travel. This pointer variable is changed whenever the vehicle is moved into the downstream link or an en-route decision has been made.

### 6.4.3 Stochastic and Adaptive Routing

This applies in general to the case of guided vehicles (i.e., both shortest path and stochastic routing). We will demonstrate this case with the network is shown in Figure 6.2. A vehicle on link  $l$  has link  $p$  as the next link to travel, and its scheduled path to the destination is  $i, 2, \dots, k, \dots, d$ . The driver may continue his trip on the scheduled route, or change his route based on the information available to him. The stochastic routing model assign each associated link a probability that the vehicle will choose a that link (at a given time and location). Let:

$p_n(l | j, t)$  = the probability that vehicle  $n$  chooses link  $l$ , provided it arrives at node  $j$  at time  $t$ .

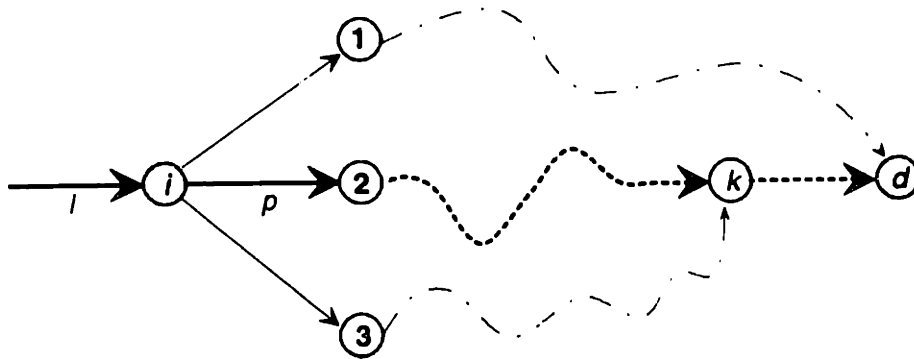


Figure 6.2 Route Choice

When the probabilities,  $p_n(l | j, t)$ , are estimated for every downstream link  $l \in A(j)$  connected to node  $j$ , the simulation model randomly assigns the next link to be traveled, based on these probabilities. The alternative route choice models that can be used for implementing the above scheme may include probit, logit models, and/or some fuzzy heuristic decision rules. For example, if the IIA (indence from irrelevant alternatives) assumption is not violated, a multinomial logit model can be used. In this case, the probabilities of selecting alternative routes at a given time can be defined as follows:

$$p_n(l | j, t) = \frac{\exp[\beta(\hat{c}_l(t) + \hat{C}_k(t + \hat{c}_l(t)) + F_{nl}(t))]}{\sum_{\hat{c}_l(t + \hat{c}_l(t)) \leq \hat{c}_j(t)} \exp[\beta(\hat{c}_l(t) + \hat{C}_k(t + \hat{c}_l(t)) + F_{nl}(t))]} \quad (6.3)$$

where:

- $\hat{c}_l(t)$  = estimated time to traverse link  $l$  for a vehicle that enters that link at time  $t$ ;
- $\hat{C}_k(t)$  = estimated travel time from node  $k$  to the destination for a vehicle that arrives at  $k$  at time  $t$ ;
- $F_{nl}(t)$  = penalty for diverting from current path.  $F_{nl}(t)$  is set to 0 if link  $l$  is on the current path of driver  $n$  or the driver has not been assigned to any path; otherwise,  $F_{nl}(t)$  is set to a positive value (for example, 10% of estimated travel time to the destination).
- $\beta$  = model parameter.

This approach avoids path enumeration at the expense of higher computational cost and maintenance of time dependent link travel times and travel times from all nodes to destinations. Two sets of travel times (costs) will be used in this model, one for guided vehicles and the other for unguided vehicles. In this scenario, the predicted travel times for unguided vehicles are based on historical data, while for guided vehicles the travel times are based on up-to-date data provided by the traffic prediction model (one of the modules that form the control strategy model).

# Chapter 7

## Output and User Interface

As the complexity of simulation models and the size of the networks to be analyzed increase, graphical visualization of the simulated output becomes increasingly important. Unintended results or aberrant model behavior can be more easily detected using graphics. Furthermore, visualization of simulation is very helpful for debugging in the model development stage, as well as checking input data. Computer graphics are used in the MITSIM model to display the temporal and spatial evolution of relevant variables such as density, speed, and travel times, and to animate vehicle movements in selected portions of the network. This chapter presents the implementation of user interfaces in the current version of the program and considerations in the design of user interfaces in the next stage of model development.

### 7.1 Collecting Data in Simulation

The output from the simulation model includes three types of data: (1) simulated measures of effectiveness (based on the actual performance of the system); (2) output from the simulated surveillance system; (3) information for monitoring the simulation.

#### 7.1.1 Standard Output

Since movements of individual vehicles are simulated in detail in this model, various performance measures of the simulated network can be provided at different levels of detail.

In the current version of the program, the following data can be written into files for each segment of selected links at user specified time intervals.

1. Number of vehicles;
2. Density.

The detectors of the surveillance system can provide:

1. Flow;
2. Speed;
4. Occupancy;
8. Headway;
128. Vehicle type;
256. Over-height attribute (Yes/No);
2048. Vehicle Identification (ID);
4096. Origin-destination.

### 7.1.2 Customized Output

Other required data may be easily obtained by adding appropriate procedures into the output module in the simulation program. The following pseudocode, for example, can be used to get information about each vehicle currently in the network:

```
1.  for (every lane i of the network) do
2.      n = first vehicle in lane i
3.      while ( n != NULL) do
4.          Get customized information from vehicle n
5.          n = next vehicle
6.      end
7.  end
```

Standard code is provided with the source files or library files of the program. The user can write his/her own C++ code to substitute line 4 to obtain the information in which he/she is interested. For example, one may write a procedure to obtain maximum queue length, speed distribution, and estimates of fuel consumption and emissions.

## 7.2 Monitoring the Simulation Process

A preliminary graphical user interface has been incorporated in the simulation program. This interface has been implemented on PCs, and is being ported to workstations. Both versions provide command line options for running the program in batch mode (Running time can be improved without graphics and interactive interface).

## 7.2.1 PC Version

This version runs on 386 or 486 PCs in DOS or OS/2 environment. It requires a VGA or better graphical card that supports a resolution of at least 640x480 pixels and 16 colors. The on-screen output can switch among three different display modes: textual, network, and animation.

### *Textual Mode <F4>*

In textual mode the program displays:

- the number of vehicles that
  - have departed;
  - are currently in the network;
  - are queuing at the external nodes (currently there is no storage space in the first link of their paths);
  - have arrived their destinations.
- the average speed of vehicles in the network.

The user can specify the frequency for reporting this information (the default time interval is 1 simulated minute).

The textual mode provides minimal information for monitoring the process of the simulation and runs faster than both the network and animation mode.

### *Network Mode <F5>*

In network mode aggregate information such as average speed and density for the link segments that are visible on screen are displayed. Each segment is shown with a color that corresponds to its current value of the selected measure. The colors used in this display mode are listed below:

#### (1) Speed (miles per hour) <S>

- $\leq 15$  (light red);
- 15-30 (red);
- 30-45 (yellow);
- 45-60 (green);
- $> 60$  (light green).

(2) Density (vehicles per lane per mile) <D>

- $\leq 25$  (light green);
- 25-50 (green);
- 50-75 (yellow);
- 75-100 (red);
- $> 100$  (light red).

Current shortest path tree for a selected destination node can also be displayed.

(3) Shortest path tree of a selected destination node <P>

- Links that are in the shortest path tree (green);
- Links that are not in the shortest path tree (light gray).

Arrow keys are used to select the part of the network to be displayed on the screen. <PgUp> and <PgDn> are used to zoom out and zoom in. In the network display mode, a small box that corresponds to the current range to be animated is also drawn on the screen. The user can use <Ctrl>+arrows keys to move this box over the network to select the range to be animated.

***Animation Mode <F6>***

The animation mode displays vehicle movements and the status of traffic signals in a selected range of the simulated network. Vehicles that are currently visible on the screen are displayed in different colors corresponding to their type or status and their positions on the screen are updated when they move to new locations. Vehicle types and status that can be shown in the current version, as well as the colors used in each display mode are listed below.

(1) Vehicle type <T>:

- New cars (light green);
- Old cars (green);
- Buses and trucks (red);
- Trailer trucks (magenta).

(2) Information availability <I>:

- Guided (red);
- Un-guided (green).



(2) Moving status <A>:

- Accelerating (green);
- Moving with an uniform speed (yellow);
- Deceleration (light red);
- Stopped (red).

(3) Lane changing status <L>:

- No lane changing (yellow);
- A change to the right lane is required (green);
- Currently looking for a change to the right lane (light green);
- A change to the left lane is required (red).
- Currently looking for a change to the left lane (light red);

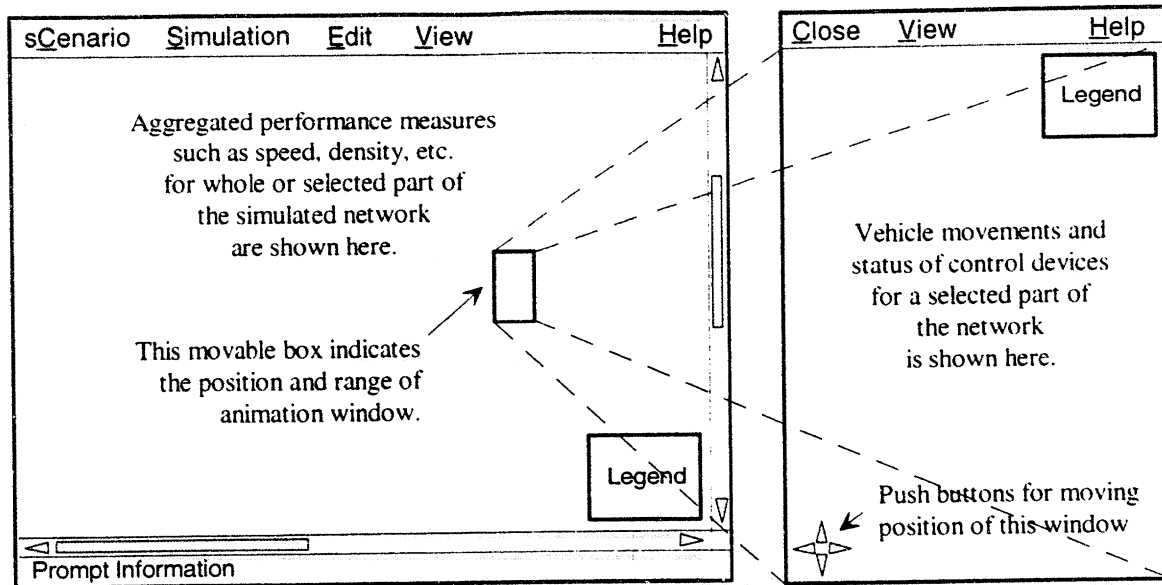
Arrow keys are used to select the part of the network to be displayed on the screen. <PgUp> and <PgDn> are used to zoom out and zoom in the display.

## 7.2.2 Workstation Version

This version is designed for workstations that run X window system in the UNIX environment. The workstation version provides similar features as that in the PC version but with the following additional capabilities:

- Support large networks;
- Provide better user interfaces for editing parameters used in the model and controlling the execution of the program.
- Simultaneously display multiple windows that show:
  - Aggregate measures for link segments such as speed and density, and the shortest path tree of selected destinations;
  - Animation of vehicle movements;
  - Graphical charts for selected statistics.
- Support distributed computation.

Two basic windows are shown in Figure 7.1.



(a) Network Window

(b) Animation Window

Figure 7.1 Layout of Network and Animation Windows.

### *Network Window*

The network window is used to show aggregated performance measures such as speeds, densities, shortest paths to a selected destination node, route choice probabilities for a selected origin and destination pair, etc. This window can be resized, zoomed in and zoomed out. The bottom and right scroll bars can be used to select the visible area of the network. A box is used to indicate the current position and range of the animation window. This box can be moved with a mouse, and once it has been moved, the animation window are redrawn.

The network window also serves as a control panel of the simulation program. The menu bar at the top of the window can be used to choose the scenario (and corresponding data files) to be simulated, start/stop and pause/resume the execution of the simulation program, edit simulation parameters (using dialog boxes), select the display mode (speed, density, etc.) and zooming scales.

### *Animation Window*

Same as the PC version, the animation window is used to display vehicle movements and the status of traffic signals in a selected range of the simulated network. This window can also be resized, zoomed in and zoomed out. When

its size or resolution has been changed, the roadways are redrawn and the corresponding position box in the network window is updated. Display mode (vehicle types, moving status, lane change, etc.) and zooming scale can be chosen from the menu bar.

The workstation version is currently being implemented using Xlib, Xt intrinsic and Motif of X window system.

# Chapter 8

## A Case Study

The MITSIM model is at an early stage of development. Debugging, testing and validation are still underway at the time this thesis is written. In this chapter, we use a small network to demonstrate the use of MITSIM. We first establish the simulation network and describe the required data files. Next we run the simulation model by loading a small number of vehicles and examine its output in detail. Then we test the program by loading vehicles at flow rates that represent a near capacity scenario, and compare the results with field data and that from INTRAS.

### 8.1 The Network

The network used in this case study is shown in Figure 8.1. There are 10 detector stations in a 1.153 miles long freeway located approximately 600 feet apart.

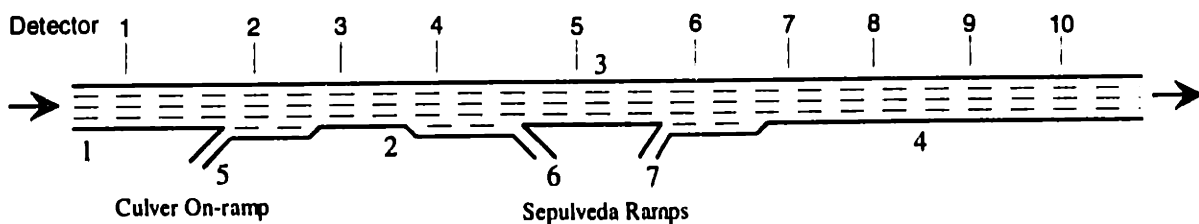


Figure 8.1 Schematic of San Diego Freeway in Los Angeles

The road network shown in Figure 8.1 is represented by 8 nodes, 7 links, and 14 segments. The coded network is shown in Figure 8.2.

A dummy segment (1000 feet) is appended at the upstream end of link 1. Vehicles originating from node 1 are first placed in this dummy segment when they enter the simulated network. The input data files that describe this road network and the surveillance devices are presented in Tables 8.1-8.5. Maximum

acceleration and normal deceleration rates for each vehicles type are the same as in INTRAS. All other parameters in the simulation model are the default values specified in this thesis.

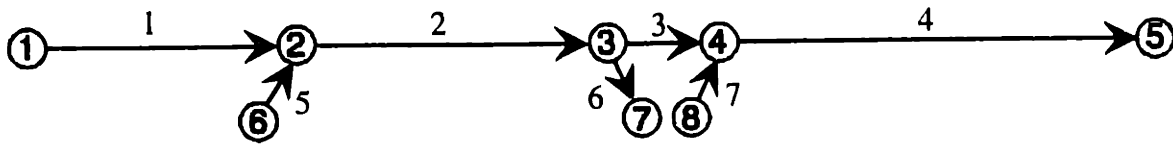


Figure 8.2 Node and Link Codes of the Network

### 8.1.1 Data Files for Road Network

**Link Data File:** The network includes 4 freeway links and 3 non-freeway links. Information of these links is provided in Table 8.1. Link 1 consists of two segments, one is the actual link from node 1 to node 2, and the other is the added dummy segment. Link 2 is divided into 3 segments, and link 4 is divided into 5 segments. All other links consist of a single segment.

Table 8.1 Link Data

Link Id	Adjacent Nodes		Reverse Link	Type	# of Segments
	Upstream	Downstream			
1	1	2	0	1	2
2	2	3	0	1	3
3	3	4	0	1	1
4	4	5	0	1	5
5	6	2	0	2	1
6	3	7	0	2	1
7	8	4	0	2	1

**Segment Data File:** Information of all the segments are described in Table 8.2. Speed limits for all the segments are set to 55 miles per hour (mph). Free flow speeds are 60 mph for freeway segments, and 55 mph for non-freeway segments.

Table 8.2 Segment Data

Segment ID		Length (feet)	Fixed Cost (\$)	# of Lanes	Grade (%)	Speed (mph)	Free Speed (mph)
Link	Segment						
1	1	1000	0	4	0	55	60
1	2	750	0	4	0	55	60
2	1	498	0	5	0	55	60
2	2	498	0	4	0	55	60
2	3	663	0	5	0	55	60
3	1	708	0	4	0	55	60
4	1	474	0	5	0	55	60
4	2	498	0	4	0	55	60
4	3	600	0	4	0	55	60
4	4	798	0	4	0	55	60
4	5	600	0	4	0	55	55
5	1	300	0	1	0	55	55
6	1	300	0	1	0	55	55
7	1	300	0	1	0	55	55

**Lane Data File:** Lane information is described in Table 8.3. This table also provides information on regulations for lane changes (column 4, see Chapter 4 for definition).

Table 8.3 Lane Data

Lane ID		Lane	Lane Changing	Lane Alignment	
Link	Segment		Regulations	# of Lanes	Upstream Lanes (link, lane)
1	1	1	2	0	
1	1	2	3	0	
1	1	3	3	0	
1	1	4	1	0	
1	2	1	2	1	1,1
1	2	2	3	1	1,2
1	2	3	3	1	1,3
1	2	4	1	1	1,4
2	1	1	2	1	5,1
2	1	2	3	1	1,1
2	1	3	3	1	1,2
2	1	4	3	1	1,3
2	1	5	1	1	1,4
2	2	1	2	2	2,1; 2,2
2	2	2	3	1	2,3
2	2	3	3	1	2,4
2	2	4	1	1	2,5

Table 8.3 Lane Data (continued)

Lane ID			Lane Changing	Lane Alignment	
Link	Segment	Lane	Regulations	# of Lanes	Upstream Lanes (link, lane)
2	3	1	2	1	2,1
2	3	2	3	1	2,1
2	3	3	3	1	2,2
2	3	4	3	1	2,3
2	3	5	1	1	2,4
3	1	1	2	1	2,2
3	1	2	3	1	2,3
3	1	3	3	1	2,4
3	1	4	1	1	2,5
4	1	1	2	1	7,1
4	1	2	3	1	3,1
4	1	3	3	1	3,2
4	1	4	3	1	3,3
4	1	5	1	1	3,4
4	2	1	2	2	4,1; 4,2
4	2	2	3	1	4,3
4	2	3	3	1	4,4
4	2	4	1	1	4,5
4	3	1	2	1	4,1
4	3	2	3	1	4,2
4	3	3	3	1	4,3
4	3	4	1	1	4,4
4	4	1	2	1	4,1
4	4	2	3	1	4,2
4	4	3	3	1	4,3
4	4	4	1	1	4,4
4	5	1	2	1	4,1
4	5	2	3	1	4,2
4	5	3	3	1	4,3
4	5	4	1	1	4,4
5	1	1	0	0	
6	1	1	0	1	2,1
7	1	1	0	0	

*Geometric Data File:* Each segment contains two curbs. Each curb is represented by a set of connect line-segments, whose geometry is determined by the coordinates of their vertices. The geometric data for all the segments in Figure 8.1 are listed in Table 8.4.

Table 8.4 Geometric Data

Link	Segment	# of Lines, Vertices
1	1	1,(-1000,0),(0,0) 1,(-1000,48),(0,48)
1	2	1,(0,0),(750,0) 1,(0,48),(750,48)
2	1	1,(750,-12),(1248,-12) 1,(750,48),(1248,48)
2	2	3,(1248,-12),(1448,0),(1646,0),(1746,-12) 1,(1248,48),(1746,48)
2	3	1,(1746,-12),(2409,-12) 1,(1746,48),(2409,48)
3	1	1,(2409,0),(3117,0) 1,(2409,48),(3117,48)
4	1	1,(3117,-12),(3591,-12) 1,(3117,48),(3591,48)
4	2	2,(3591,-12),(3691,0),(4089,0) 1,(3591,48),(4089,48)
4	3	1,(4089,0),(4689,0) 1,(4089,48),(4689,48)
4	4	1,(4689,0),(5487,0) 1,(4689,48),(5487,48)
4	5	1,(5487,0),(6087,0) 1,(5487,48),(6087,48)
5	1	1,(550,-200),(750,0) 1,(562,-200),(750,-12)
6	1	1,(2409,-12),(2609,-200) 1,(2409,0),(2621,-200)
7	1	1,(2917,-200),(3117,0) 1,(2929,-200),(3117,-12)

### 8.1.2 Data File for the Surveillance System

There are 10 detector stations (43 detectors) in the simulated network. All these detectors, except the ones in acceleration and deceleration lanes connected to on- and off-ramps, represent the actual surveillance system in this network. 5-minute aggregated speeds for each detector station are available (*see* <sup>[22]</sup>), and will be used for validation purpose. The specification of the detectors in this network are given in Table 8.5.



Table 8.5 Detector Data

Station No	Location				Task	Working Probability
	Link	Segment	Lane	Distance (feet)		
1	1	2	1	504	2	1.00
1	1	2	2	504	2	1.00
1	1	2	3	504	2	1.00
1	1	2	4	504	2	1.00
2	2	1	1	312	2	0.00
2	2	1	2	312	2	1.00
2	2	1	3	312	2	1.00
2	2	1	4	312	2	1.00
2	2	1	5	312	2	1.00
3	2	2	1	302	2	1.00
3	2	2	2	302	2	1.00
3	2	2	3	302	2	1.00
3	2	2	4	302	2	1.00
4	2	3	1	477	2	0.00
4	2	3	2	477	2	1.00
4	2	3	3	477	2	1.00
4	2	3	4	477	2	1.00
4	2	3	5	477	2	1.00
5	3	1	1	462	2	1.00
5	3	1	2	462	2	1.00
5	3	1	3	462	2	1.00
5	3	1	4	462	2	1.00
6	4	1	1	358	2	0.00
6	4	1	2	358	2	1.00
6	4	1	3	358	2	1.00
6	4	1	4	358	2	1.00
6	4	1	5	358	2	1.00
7	4	2	1	237	2	1.00
7	4	2	2	237	2	1.00
7	4	2	3	237	2	1.00
7	4	2	4	237	2	1.00
8	4	3	1	254	2	1.00
8	4	3	2	254	2	1.00
8	4	3	3	254	2	1.00
8	4	3	4	254	2	1.00
9	4	4	1	432	2	1.00
9	4	4	2	432	2	1.00
9	4	4	3	432	2	1.00
9	4	4	4	432	2	1.00
10	4	5	1	224	2	1.00
10	4	5	2	224	2	1.00
10	4	5	3	224	2	1.00
10	4	5	4	224	2	1.00

## 8.2 A Debugging Scenario

In this scenario, only 5 vehicles depart during the period to be simulated (12:30:00~13:30:00). Information on departure time, origin, destination and type of each vehicle is listed in Table 8.6. Note that the first vehicle (a trailer truck) has to change lanes at least once in order to leave the network using the Sepulveda off-ramp (link 6). This simple scenario is used to verify that vehicles in the simulated network will complete their trips, and the model results in reasonable values for the various parameters of interest.

Table 8.6 Five Generated Vehicles

Vehicle ID	Departure Time	Origin	Destination	Vehicle Type
1	45000	1	7	4
2	45000	1	5	3
3	45008	6	5	2
4	45008	6	5	1
5	45010	1	5	1

It takes 2 seconds on a 486PC (with graphics) and less than 1 second on a DEC 5000 workstation (without graphics) to complete the simulation. The five vehicles all arrived at their own destinations and a summary of their type, travel times, and average speed for each vehicle is shown in Table 8.7.

Table 8.7 Travel Time and Space Mean Speed

Vehicle ID	Vehicle Type	Ori. Node	Des. Node	Dep. Time	Arrival Time	Travel Time (minutes)	Mileage (miles)	Speed (mph)
1	516	1	7	45000	45040	0.7	0.535	48.2
3	2	6	5	45008	45075	1.1	1.034	55.5
4	1	6	5	45008	45077	1.1	1.059	55.3
2	3	1	5	45000	45078	1.3	1.176	54.3
5	1	1	5	45010	45086	1.3	1.176	55.7

We assume that all of the 43 detectors (including the 3 dummy detectors) in the 10 stations collect all the default point data and point to point data except headways. In this case the *task* assigned to each detector is 6535. The data collected by point data sensors is reported in one-minute intervals (see Table 8.8).

Table 8.8 Results from Point Data Sensors

Time Interval	Detector		Occupancy (%)	Speed (mph)	Flow (vehicles)
	Station	Lane			
12:30:00-12:31:00	1	3	1.53	53.86	2
	1	4	0.83	56.59	1
	2	1	0.69	57.98	2
	2	4	1.53	53.86	2
	2	5	0.83	56.59	1
	3	1	0.67	60.00	2
	3	3	1.53	53.86	2
	3	4	0.83	56.59	1
	4	1	1.17	53.18	1
	4	2	0.67	60.00	2
	4	4	0.35	54.55	1
	4	5	0.83	56.59	1
	5	1	0.67	60.00	2
	5	3	0.35	54.55	1
	5	4	0.83	56.59	1
	6	2	0.67	60.00	2
	6	4	0.35	54.55	1
	6	5	0.83	56.59	1
	7	1	0.67	60.00	2
	7	3	0.35	54.55	1
7	4	0.83	56.59	1	
8	1	0.67	60.00	2	
8	4	0.83	56.59	1	
12:31:00-12:31:27	8	3	0.35	54.55	1
	9	1	0.67	60.00	2
	9	3	0.35	54.55	1
	9	4	0.83	56.59	1
	10	1	0.71	54.57	2
	10	3	0.35	54.55	1
	10	4	0.94	51.14	1

One vehicle is a truck equipped with a VRC transponder (vehicle #1 in Table 8.7). The data reported by the VRC readers as well as over-height detectors are listed in Table 8.9. Note that vehicle #1 changed lanes before it arrived at detector station 4. VRC data can be used to calculate point to point data such as travel time and average speed. It takes, for example, 9 seconds for the vehicle to travel from station 1 to 2, 8 seconds from station 2 to 3, and 7 seconds from station 3 to 4. The corresponding average speed between station 1 and 4 is 47.90 mph.

Table 8.9 Results from VRC Readers and Over-Height Detectors

Station	Lane	Time	Vehicle ID	Ori. Node	Des. Node	Vehicle Type	Over-Height	Speed (mph)
1	3	45005	1	1	7	516	No	53
2	4	45014	1	1	7	516	No	53
3	3	45022	1	1	7	516	No	53
4	1	45029	1	1	7	516	No	53

### 8.3 A Comparison Scenario

The trip table used in this scenario is given in Table 8.10. In this table, travel demand for each of the 5 O-D pairs is provided in 5-minute intervals for a one-hour period. Based on this trip table, 7084 vehicles are generated. The flow rate in this scenario is approximately 1700~1800 vehicles per lane per hour. High performance passenger and lower performance passenger cars are the two types of vehicles simulated. Each type accounts for 50% of the total vehicles.

Table 8.10 Travel Demand

5-minute Time Period	By Origin-Destination Pairs					By Links			
	1→5	1→7	6→5	6→7	8→5	1	2	3	4
1	6324	264	323	13	660	6588	6924	6647	7307
2	6146	190	372	12	744	6336	6720	6518	7262
3	5888	376	226	14	684	6264	6504	6114	6798
4	5956	248	0	0	660	6204	6204	5956	6616
5	6042	318	376	20	624	6360	6756	6418	7042
6	6006	186	12	0	600	6192	6204	6018	6618
7	6094	254	127	5	624	6348	6480	6221	6845
8	5841	243	207	9	444	6084	6300	6048	6492
9	6236	328	308	16	660	6564	6888	6544	7204
10	5982	450	11	1	600	6432	6444	5993	6593
11	5426	226	334	14	756	5652	6000	5760	6516
12	5233	455	0	0	504	5688	5688	5233	5737

The program takes 17-21 minutes on a 486PC (with graphics) and 3-5 minutes on a DEC 5000 workstation (without graphics) to complete an one-hour simulated time. Average speeds at each of the 10 detector station for 12 time periods are listed in Table 8.11. Figure 8.4 compares the field data and the simulation outputs by MITSIM and INTRAS.

Table 8.11 Comparison with Field Data and INTRAS Generated Data

Time Period	Data Source	Detector Station									
		1	2	3	4	5	6	7	8	9	10
1	MITSIM	44.3	41.0	41.0	41.1	40.2	39.5	43.1	44.6	46.2	50.5
	Field	18.9	23.4	28.9	19.8	23.8	22.0	25.0	25.5	30.5	33.6
	INTRAS	51.5	46.4	37.3	41.1	45.5	33.2	36.0	45.0	44.7	44.6
2	MITSIM	36.5	31.7	31.7	30.2	27.8	28.6	39.0	41.9	43.8	49.6
	Field	38.9	39.6	41.8	21.5	18.7	24.7	30.4	33.7	36.0	37.7
	INTRAS	51.4	43.4	34.2	42.5	35.0	24.0	33.3	44.7	43.9	44.0
3	MITSIM	24.8	27.6	31.7	33.4	33.6	35.2	40.2	41.9	43.6	49.0
	Field	41.4	42.4	45.2	28.7	39.7	40.9	32.2	33.7	37.7	39.6
	INTRAS	52.7	45.5	37.1	40.6	33.1	25.6	33.9	44.1	43.1	44.0
4	MITSIM	34.3	32.4	32.0	33.1	33.4	33.5	39.8	41.2	42.8	48.5
	Field	42.3	43.1	46.1	29.1	40.1	44.6	43.8	45.1	47.3	42.4
	INTRAS	52.9	49.7	46.4	45.2	41.2	29.6	34.8	46.3	45.3	45.0
5	MITSIM	30.9	30.9	32.9	33.4	32.7	33.2	39.5	41.5	43.2	49.1
	Field	41.3	42.2	46.2	29.9	39.4	44.6	43.3	45.1	46.4	46.7
	INTRAS	51.9	46.3	36.6	42.8	48.9	37.4	36.5	45.4	45.2	43.4
6	MITSIM	30.7	30.9	33.5	34.9	35.0	36.1	40.5	41.4	42.2	48.3
	Field	41.0	42.1	45.2	27.3	39.0	44.7	43.3	44.8	47.8	47.3
	INTRAS	52.8	48.9	43.5	44.7	47.6	32.8	34.7	45.6	44.7	44.8
7	MITSIM	40.6	39.2	39.0	39.6	39.3	39.8	41.8	42.5	44.4	50.7
	Field	41.6	42.4	44.9	29.4	39.9	42.7	41.6	44.0	46.0	45.0
	INTRAS	51.7	48.6	43.9	44.7	49.0	42.2	37.5	45.8	45.2	46.0
8	MITSIM	40.5	36.7	38.8	39.7	40.9	41.3	42.5	42.0	43.3	48.6
	Field	41.5	42.3	45.1	29.1	39.5	44.4	43.1	44.0	45.7	47.0
	INTRAS	52.2	48.7	44.0	45.3	40.1	42.7	38.5	45.7	44.8	43.2
9	MITSIM	41.1	39.2	39.0	39.9	40.3	40.1	42.0	43.3	44.9	50.8
	Field	41.1	42.4	45.5	28.1	39.8	44.6	43.2	45.0	46.8	46.8
	INTRAS	51.8	45.1	38.2	43.6	48.8	39.9	37.2	45.6	45.3	44.3
10	MITSIM	40.0	37.5	37.9	38.7	39.9	40.2	41.4	42.8	44.4	50.1
	Field	40.2	40.3	43.8	27.9	38.2	43.3	42.1	44.3	46.4	47.4
	INTRAS	51.9	47.2	41.4	42.1	46.5	32.1	34.0	44.8	44.8	45.0
11	MITSIM	46.8	43.2	43.5	44.9	43.3	42.7	43.7	44.8	46.2	51.4
	Field	41.3	42.4	46.3	30.4	40.3	44.6	43.0	43.7	44.5	44.8
	INTRAS	52.7	49.5	43.9	45.0	49.9	41.1	36.0	45.8	45.4	45.3
12	MITSIM	51.0	50.6	50.7	51.3	51.4	49.9	49.9	49.7	49.2	51.9
	Field	42.2	43.1	46.1	30.5	39.7	44.6	44.2	46.9	48.2	48.5
	INTRAS	52.4	49.8	46.5	45.7	50.6	45.8	40.6	47.1	46.9	47.4

The average speed of the 120 data points is 39.7 mph for field data, 43.6 mph for INTRAS, and 40.7 mph for MITSIM. The average speed simulated by MITSIM is closer to field data than INTRAS. The distribution of measured speeds is shown in Figure 8.3.

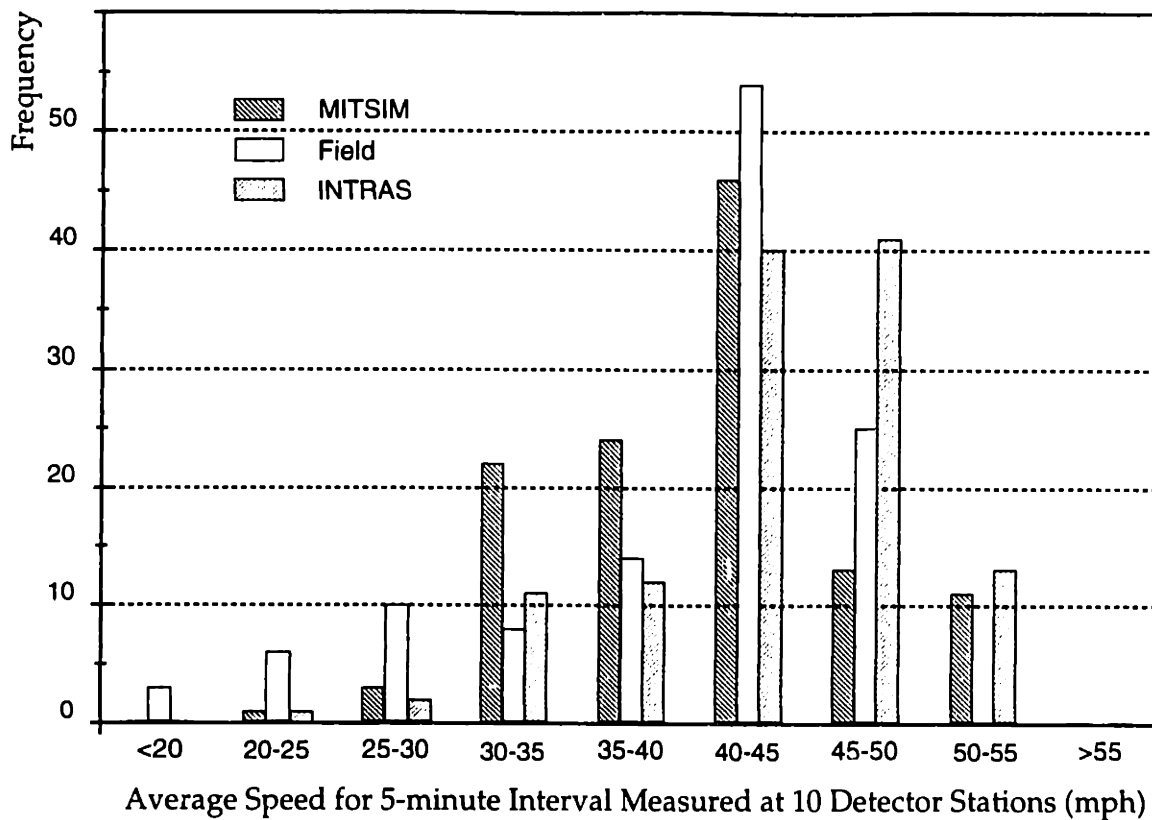


Figure 8.3 Distribution of Measured Speed

The speeds by individual detectors and time periods in Table 8.11 indicate that both simulation models over-estimated the speeds at all stations during the first period, at stations 4-10 during the second period, and at stations 7-10 during the third period. This is probably due to the fact that the simulation started with an empty network, but some congestion was present in the network when the simulation started and did not dissipate until the third time period ended. Both MITSIM and INTRAS also consistently over-estimated the speeds at station 4 for all time periods. This could be caused either by defective detectors, or by the presence of point congestion in the vicinity of station 4.

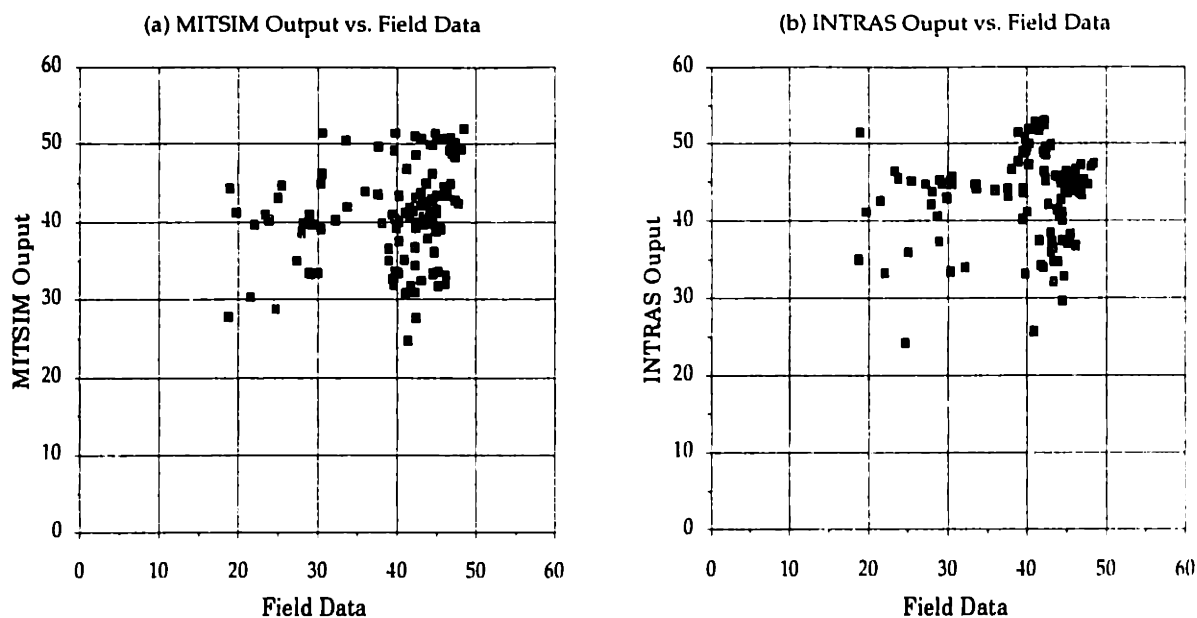


Figure 8.4 Comparison with Field Data

Figures 8.4 (a) and 8.4 (b) are scatter plots of field data and simulated results by MITSIM and INTRAS respectively. In these figures, x-coordinate of each point represents average speeds of field data, and y-coordinate represents simulation output. These points tend to distribute around the 45° diagonal line. However, the fact that some points distribute away from the diagonal line indicates that the simulated data compares poorly with the field data for some stations during certain periods. This might be partially caused by the randomness of field data because of unknown factors and/or by uncalibrated parameters used in the model.

# Chapter 9

## Conclusion

In the final chapter of this thesis, research directions for further development of the traffic simulation model are discussed. The chapter concludes with an appraisal of the current research work.

### 9.1 Next Steps

In previous chapters the framework and basic components of the MITSIM model are described. This model is complex and should be built incrementally. In the first stage of its development, the focus is on traffic operations on freeways, and only the major components of the system has been implemented. This section considers the next steps in the development of the model.

#### 9.1.1 Implementation of Measures of Effectiveness

The current version of the MITSIM model generates only basic network performance measures such as speed, density and flow on selected links, and travel times and mean speeds for selected origin-destination pairs. However, this model is designed to support comprehensive system evaluation tasks, and more statistics and measures of effectiveness (MOE) can be generated by incorporating appropriate code into the program. This task is to define specifically what kind of MOE's are required for the evaluation of IVHS applications, and to implement them in the traffic simulation model.

#### 9.1.2 Communications with the Control and Surveillance System Models

Input to the MITSIM model includes the status of various control devices and events. The status of control devices and events are output from the *Control System Model* and they are based on "real-time" traffic information collected by



the *Surveillance System Model* from the MITSIM model. Since these three models need to exchange information dynamically, two problems require detailed consideration:

- How are the data flows between the traffic simulation model and the Control and Surveillance System models going to be transferred?
- Should the three models be implemented in a distributed environment (each model runs on a different processor) or be compiled as one program and run on a single machine?

Using multiple processors for different models may significantly improve the speed of the simulation system, but it requires careful coordination and special implementation efforts.

### 9.1.3 Incorporation of Urban Streets

The MITSIM model contains most of the functionality required for supporting an integrated traffic networks. However, the following elements that need to be incorporated to simulate traffic operations in urban streets:

- Turning movements at signalized and unsignalized intersections;
- Bus operations;
- Pedestrian interrupts;
- Parking interrupts;
- Passing (overtaking) using the opposite direction lane on two-way streets.

### 9.1.4 Emergency Vehicle Operations

Simulation of emergency vehicle operations requires special algorithms:

- **Dispatching:** When the *Control System Model* dispatches one or more emergency vehicles in response to an incident that occurred in the network, the required number of vehicles are generated and they enter the network immediately.
- **Movements:** The movements of emergency vehicles may be influenced by the density of vehicles on their path. Algorithms for moving emergency vehicles need to be developed.
- **Impacts:** Emergency vehicles affect speeds and lane changes of the vehicles on their path. The car-following and lane-changing logic and

vehicle behavior in general may need modification in the presence of emergency vehicles.

- **Response Time:** When emergency vehicles arrive at their destination, the simulation model may record the arrival time, and then calculate the response time.

### **9.1.5 Reversible Lanes**

Furthermore, reversible lanes may be used as part of the control action under consideration (e.g. in peak period). In response to incidents occurred and/or detected in a lane, a lane can be reversed so that emergency vehicles can access the incident location quickly. Similarly, in special facilities, such as tunnels, reversible lanes may be used for vehicle dispersing in the case of an accident.

### **9.1.6 Calibration and Validation**

The MITSIM model uses a large number of parameters that characterize driver behavior and network performance. These parameters were assigned default values based on information found in the literature or our best guess if no such information is available. The model also provides the users with flexibility for changing the value of the parameters used in the model. Nevertheless, these parameters require considerable calibration and validation effort. Two basic approaches for calibration and validation are:

- Observe the behavior of the simulation model using hypothetical small networks. Simulations for different scenarios, since the networks are small and scenarios are controlled, may reveal counter intuitive behavior of the model.
- Obtain data on existing facilities, run the simulation model and compare its results to the field observations. If there is unreasonable difference between simulated data and field data, appropriate parameters and algorithms used in the model should be modified.

The parameters to be calibrated include the parameters in the car-following, lane-changing, and route choice algorithms as well as the step sizes used to control the frequency of invoking these algorithms.

## 9.2 Future Research

The traffic simulation model developed in this thesis contains the basic functions for supporting system evaluation of IVHS applications. The simulation model will eventually be the engine behind a system model. An important element of this system model is the control strategy model(s). The integrated system model, in order to be useful for IVHS evaluations, requires research and development in the following areas:

### 9.2.1 Control Strategy Model

The control strategy model should perform three major tasks:

***Prediction of Travel Times and Congestion:*** This is one of the most important tasks for implementing ATIS and ATMS. The data used for traffic prediction includes (1) historical link travel times and traffic information, and (2) current traffic information collected by the surveillance system. The predicted travel times and travel demand are used for generating traffic control strategies.

***Generation of Responsive Signal Controls:*** This task includes algorithms that optimize the actions of the traffic control system. These algorithms use real-time sensor data from the local field controllers and predicted traffic conditions to update the status of various traffic control devices such as variable speed limits signs, lane use signs and ramp meters. The MITSIM model can be used in this research to test and evaluate the performance of these algorithms.

***Provision of Dynamic Route Guidance:*** This task is to develop an algorithm that connects congestion prediction to ATIS. In particular, this task will focus on how to provide dynamic and consistent route guidance to motorists using variable message signs, highway advisory radio and in-vehicle route guidance systems.

### 9.2.2 Computational Considerations

The highest resolution of the microscopic traffic simulation model is obtained at the expense of extensive requirements in terms of running time and memory. Thus the model may not be very useful for large networks without

compromising its resolution. To support the simulation of traffic operations in large networks, we may consider the following alternatives.

**Macro Links and Nodes:** Links and nodes in the simulated network are divided into two groups: (1) micro links and nodes; and (2) macro links and nodes. Traffic operations on micro links and nodes are simulated in detail using the car-following and lane-changing algorithms described in this thesis. On macro links and nodes, vehicle movements are governed by speed-density relationships and queue dispersion equations. In other words, microscopic and macroscopic simulation approaches are integrated into one single model, and applied to the network selectively.

**Distributed Implementation:** This approach divides the large network into several smaller subnetworks, and runs simulations for these subnetworks simultaneously using multiple computer processors. There are three main issues in the distributed implementation: (1) defining the *common nodes* among subnetworks; (2) transferring vehicles between subnetworks via the common nodes; and (3) synchronization and coordination of the different subnetwork simulations.

**Parallel Implementation:** In this approach, the simulation model is implemented on a massively parallel computing architecture, such as the Connection Machine by Thinking Machine Corporation<sup>[80]</sup>. The Connection Machine has 65,536 processors and provides 2 Gbytes of memory (CM-2). In this implementation, nodes, links, segments, traffic control devices, surveillance devices and vehicles are represented as different *Shapes* in C\*<sup>[81]</sup> (similar to arrays in C and C++). Since computation involved in each *Shape* and communication between *Shapes* can be performed by 64K processors simultaneously, efficiency of the simulation is expected to be significantly improved<sup>[44]</sup>. Although parallel computing has not yet become widespread, given the characteristics of traffic simulation algorithms and the expected continuing improvement in hardware and software of the Connection Machine, massively parallel computers may be the ultimate platform for large network simulation models, especially when the simulation needs to be done in real time.

## 9.3 Conclusion

The primary objective of this work was to develop a research tool for system evaluation of IVHS applications. Towards this end, a microscopic traffic simulation model was developed. This model may output measures of effectiveness (MOE's) that are required in the system evaluation, such as speed, travel times, queues, delay, incident response time, etc. It is the heart of the framework proposed for evaluation of ATIS and ATMS systems. This framework can be used to test algorithms for traffic management (e.g., incident detection, responsive signal controls, etc.), evaluate system integration (e.g., interactions between different components), and assess innovative ideas for traffic management (e.g., congestion pricing, flexible work hours, etc.). Other components of this framework (e.g., the *Control System Model* and the *Surveillance System Model*, etc.) are not finished yet. However, the traffic simulation model developed in this research may act as a base on which these components can be incrementally developed and incorporated.

# Appendix

In this appendix, we present some representative traffic simulation models and their advantages and limitation. Elements of these models are used in the development of the microscopic traffic simulation model presented in this thesis.

## NETSIM

NETSIM (NETwork SIMulation Model) is a detailed microscopic simulation model for urban streets. It represents traffic conditions on surface streets in detail and is well suited for analysis of urban traffic control systems and evaluation of network performance. The model has been the subject of extensive field testing, evaluation, and validation. NETSIM is designed as a tool for analyzing and evaluating a wide range of traffic control and surveillance concepts for complex street networks, including:

- "Stop" and "yield" signs;
- Turn controls;
- Parking controls;
- Fixed-time signals;
- Vehicle-actuated signals;
- Surveillance systems.

The modular structure incorporated in this microscopic simulation model enables the detailed treatment of:

- Car-following behavior;
- Network geometry;
- Grades;
- Bus traffic;
- Queue formation;
- Intersection discharge;
- Lane blockages;
- Pedestrian-vehicular conflicts.

NETSIM provides a flexible mix of standard performance measures. Inputs to NETSIM include the coded street network and its geometry, flow rates of source/sink nodes (no O/D information for individual vehicles), flow characteristics such as percentages of turning movements, gap distribution, discharge rate, traffic composition, type of controller and control logic (signal time settings), and surveillance system. NETSIM also handles the effects of buses, street parking, and pedestrians on the flow of traffic. Some "rare events" can also be specified as input to NETSIM and simulated stochastically.

NETSIM is a microscopic, time-based stochastic traffic simulation model. It represents the street system as a network of interconnected directional links and nodes. Each link represents a particular approach to a node, and changes in link characteristics can be modeled by inserting mid-block nodes. A link may have up to 5 lanes plus right and left turn pockets. Traffic generators, such as parking lots and minor streets, are modeled as sink or source nodes. Vehicles are processed individually once per second, responding to current traffic status and controls specified at the nodes (yield or stop signs, signals with pre-timed or traffic-actuated controllers, etc.). The vehicle motion is governed by a series of car-following, queue-discharging, and lane-changing algorithms. The simulation structure in NETSIM consists of the following modules:

- **Process Queues:** all vehicles that were located in queues at the beginning of the time step are processed first, based on a queue-discharging algorithm;
- **Move Vehicles:** all remaining vehicles already on the network, but not "in-queue", are processed next, based on car-following and lane-changing algorithms;
- **Input New Vehicles:** new vehicles are admitted to the network from their origin nodes (both interior and exterior);
- **Update Signal Status:** status of all traffic lights is updated in every iteration of simulation (1 second);
- **Record Statistics:** a set of standard vehicle and link statistics are accumulated;
- **Output Reports:** if a point has been reached in the simulation run where a statistical output is called for, the necessary results are reported.

The car following logic in NETSIM, for cases with vehicles within 200 feet (61m), is represented by the following equation:

$$a_f = \frac{7(x_l - x_f - v_f - L_l) + \frac{1}{3}(v_l^2 - 2v_f^2)}{v_f + 3} \quad (10.1)$$

where:

- $a_f$  = acceleration of follower at the end of the time slice;
- $x$  = location from the beginning of link;
- $v$  = speed;
- $L$  = vehicle length, including 3 feet (1m) clearance;
- $l$  = subscript indicating lead vehicle;
- $f$  = subscript indicating following vehicle.

The standard NETSIM output includes a variety of measures of effectiveness for each link, such as travel time, queue length, total vehicle miles, travel time and delay per vehicle (total delay includes both delay due to stops and "delay" due to speeds reduced below some specified target level, etc.), number of stops per vehicle, fuel consumption and vehicle emission. Other supplementary outputs include the O-D pattern of all vehicles, types and locations of all detectors, all randomly generated "rare events", and bus performance.

NETSIM is particularly applicable to the analysis of complex urban traffic networks, and it is one of the most widely used microscopic traffic simulation models. The model was programmed in FORTRAN 77, and versions are available for both mainframe and PC computers. To facilitate data preparation and output interpretation, a data input and output graphic display manager, called GTRAF, was developed for NETSIM.

Former versions of NETSIM can model up to 99 nodes, 160 links and 1600 vehicles. Although the maximum network size and computing time for the newer versions is only limited by the available computer resources, high memory requirements and very extensive computation may restrict the use of NETSIM on large networks.

Traffic control systems can be modeled in NETSIM, but the control algorithms must be inserted by the user. NETSIM does not support vehicle routing; turning movements are set randomly at each intersection, using



observed turning fractions as probabilities. These probabilities may also be obtained through the traffic assignment model TRAFFIC.

## **INTRAS/FRESIM**

INTRAS (INtegrated TRAffic Simulation) is a microscopic freeway and freeway corridor simulation model developed under the sponsorship of the Federal Highway Administration [20, 10, 92, 93]. The model extends features supported by its predecessor models (UTCS-1 and SCOT [24]) to support car-following and lane-changing freeway logic. It can be used to simulate vehicle movements on freeways in detail. INTRAS is designed to answer "what if" questions with respect to network geometries, traffic control strategies and traffic characteristics. It can also be used for evaluating incident detection methods and alleviation of incident effects through control and detector placement.

INTRAS requires as input detailed data on the network geometry, traffic demand and the traffic control system. The network data describes the geometry and topology of links and nodes (link length, number of lanes, lane channelization, type of link, grade, radius of curvature, percentage of super elevation, and pavement type). Surface roads and homogeneous freeway sections (with the same geometric characteristics, such as grade, curvature and number of lanes) are represented by links. Each link can include up to five through lanes and two additional turning pockets or auxiliary lanes.

Up to five types of vehicles can be considered in INTRAS. For each vehicle type, information on acceleration, deceleration, top speed, length, fuel-consumption, and emission characteristics are required. Travel demand is represented as time-varied flow rates (number of vehicles entering the network by vehicle type), and time-varied percentages of vehicles turning at each intersection.

Radar-based detection devices and magnetic induction loop detectors are the two sensor types that can be simulated in the surveillance system. Incidents and their effect on lane capacity and delays may also be simulated by INTRAS.

INTRAS is a time-based microscopic simulation model. Vehicles traversing freeway links are moved at one-second intervals. A car-following algorithm was

developed in which the acceleration of the subject vehicle is computed on the basis of its current speed, the speed of the leading vehicle and the spatial separation of the two vehicles. A collision avoidance (emergency deceleration) module is applied, when necessary, to override the car-following model described above. The lane-changing algorithm adopted in INTRAS consists of two processes. First, each vehicle is polled to determine whether a change is desirable in order to pass a slow leader or exit from the freeway. Second, the immediate environment is examined for each vehicle desiring to change lanes to determine if the change may be accomplished safely. When it changes lanes, a vehicle occupies two lanes simultaneously for a period of several time steps (seconds).

INTRAS has no traffic assignment capabilities. Vehicles emerge into and depart from the network according to time-based flow rates, and they move around based on turn percentages. In its most recent version, an algorithm which determines the origin-destination matrix for the freeway subnetworks was added. This algorithm uses ramp volumes to estimate O-D flows. The user is given the option of entering a partial, or full O-D matrix for the freeway. Vehicles are assigned destinations as they enter the freeway, which are selected stochastically so as to replicate the specified set of destinations derived from the O-D matrix.

INTRAS can simulate fixed-time and traffic-actuated signals and sign controls. Intersections with fixed-time signals may have up to six pre-determined plans and nine phases per cycle. INTRAS allows the user to simulate an incident at any location on a freeway link and for any given length of time. The incident may block one or more lanes during its existence.

A fuel-consumption and vehicle-emission evaluation model are built into INTRAS, following a similar module developed for the NETSIM simulation model.

The standard output in INTRAS includes:

- Vehicle miles and minutes;
- Volume, density and speed;
- Delay per vehicle;
- Lane changing;

- Fuel consumption and emissions by vehicle type;
- Speed and time headway distributions.

The summary statistics are normally reported at the end of each simulation subinterval, on either a link specific or network-wide basis. The user may also select those reports at specified time intervals within each subinterval. These statistics are cumulative either from the start of the simulation or, optionally, from the beginning of each subinterval. The output of surveillance detector data may be restricted to individual links or inhibited altogether. Furthermore, data may be output for later processing by plotting modules. Through this option, vehicle trajectories and/or measures of effectiveness contour graphs may be created for selected freeway links or groups of links.

Both the traffic simulation model and the incident detection algorithm of INTRAS were validated against a number of data sets from the field covering a range of flow conditions with and without incidents. The results indicate that INTRAS provides a realistic representation of traffic (compared to actual data obtained from existing surveillance system) [21].

INTRAS was programmed in FORTRAN and runs on PCs and mainframes. After its original development, further development of INTRAS led to FRESIM which will be incorporated, when completed, in the TRAF family of simulation models [60, 9].

## **TRAFLO**

TRAFLO, which consists of two macroscopic models, one for urban networks (NETFLO) and the other for freeways (FREFLO), is a subsystem of the TRAF family [72, 25, 26, 53]. It is a tool for use in transportation planning and traffic engineering. TRAFLO is used to test and evaluate traffic management strategies applied over large roadway networks. The model is capable of simulating the traffic environment of networks consisting of freeways, arterials, and grid surface streets in regions of approximately 2,000 intersections. TRAFLO provides a hierarchy of macroscopic simulation models at different levels of detail, and permits the user to choose the optimal trade-off between the accuracy required and the available computer resources.

The input required by TRAFLO depends on the degree of detail required. Input data include:

- Characteristics of urban and freeway links;
- Turning movements or trip tables;
- Incident specifications;
- Fixed-time signal controls;
- Actuated signal controls;
- Bus transit.

TRAFLO consists of four component models that interface with one another (see Figure 10.1) The first three models simulate traffic operations in urban subnetworks at different levels of detail, and the last model simulates freeway traffic. If the network consists of freeways and urban streets, it must be partitioned into freeway and urban subnetworks. Various subnetworks are interconnected through the "interface nodes." A more detailed description of the simulation models NETFLO and FREFLO follows.

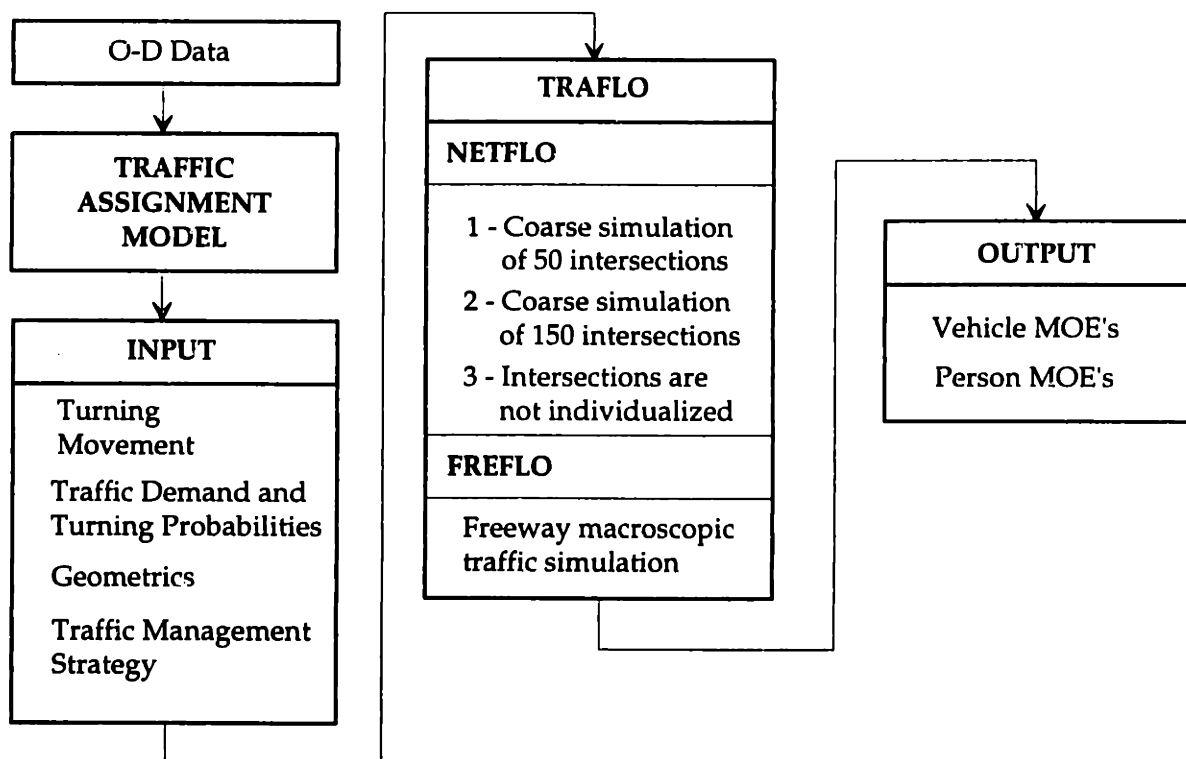


Figure 10.1 Structure of the TRAFLO Model  
(Source: FHWA, 1982)

# NETFLO

NETFLO is macroscopic in nature and may operate in three separate levels of detail.

Level 1 is the most detailed level of traffic representation in NETFLO. The representation of traffic flows at this level is almost microscopic since individual vehicles are identified; different types of vehicles are treated separately according to their respective operational characteristics. Actuated signal controls, right-turn-on-red, and pedestrian interference are modeled.

Vehicle movements are simulated with an event-based process. Associated with each vehicle is an "activation time" (*AT*), the time at which the vehicle will be processed. When the simulation clock equals the vehicle's *AT*, the vehicle is processed, and its new location, speed and *AT* are calculated based on the conditions downstream of its starting point. Then this vehicle remains "dormant" until the simulation clock advances to its new *AT*, whereupon the vehicle is again processed.

At this level of detail, NETFLO is able to simulate the interaction of the various vehicle types and the impact of various traffic management strategies in detail (such as lane channelization of bus-only or truck-only streets, carpool priority, etc.).

Level 2 is less detailed and includes fewer features than level 1, but it is computationally faster. The flow model in this level represents the traffic stream as movement-specific statistical histograms where the flow rate is expressed as a function of time. A total of five such histograms are used for each of the various turning movements on each link. These five histograms describe, respectively:

- The entry of flow from upstream links;
- Platoon flow arriving at the stopline (obtained from entry flows according to specified turn ratios for the subject link);
- Service volume for each turn movement (reflecting the type of control device);
- Queue formation and dispatching at the stop line (derived from demand and service rates); and

- Flow discharged to downstream links.

This approach gives NETFLO the ability to identify the characteristics of each turn-movement-specific component of the traffic flow, and to properly model the evolution of spill-backs. At this level, buses are still represented as separate entities (although in less detail than level 1).

Level 3, which is the fastest computationally, is the least detailed and is applicable only to arterials. At this level, traffic flows and signal controls are described with aggregated variables. Instead of representing detailed behavior of traffic at intersections explicitly, the model uses associated impedances. Vehicle delays are estimated using an extension of Webster's formula [91].

NETFLO provides the performance measures at different level of detail based on the specification of the simulation. System wide and individual vehicle based statistics can be provided.

NETFLO is programmed in FORTRAN and has been statistically validated by comparing model results with field data [53]. The computing time for the level 1 model is a linear function of the number of vehicles that travel in the network. The running time for levels 2 and 3 models is more sensitive to the size of the network rather than the traffic volume [54].

## **FREFLO**

FREFLO, an extension of MACK, is a macroscopic simulation model for freeway traffic [72]. It is particularly useful for evaluating freeway operations, such as control strategies in response to freeway incidents, ramp metering, etc. Traffic controls or ramp metering in FREFLO may be specified as a function of the time of the day, or depend on traffic conditions as measured by the surveillance system. FREFLO does not provide for the incorporation of signalized surface streets, and there is no concept of vehicle routing and O/D information. Demand for freeway traffic is modeled by time-dependent on-ramp and off-ramp volumes.

FREFLO requires as input freeway geometric data such as the number of lanes and section lengths, on-ramp and off-ramp locations, and normal section capacities. The required traffic data includes on-ramp demand volumes, off-

ramp rates and the initial state of the densities and speeds. It is generally necessary to develop speed-density relationships for each distinct freeway facility, however, the required calibration of the various parameters may be extensive. The TRAFFIC model is interfaced internally with FREFLO to facilitate flow pattern and turn movements of traffic.

FREFLO accepts three vehicle categories: (1) automobiles and trucks, (2) buses, and (3) carpools. In addition, a number of lanes can be designated for special purposes (for example, buses and/or carpools only).

The freeway segment is divided into homogenous sections, and the time period of analysis is divided into uniform time intervals. Traffic is modeled in terms of aggregate measures (flow, density and space mean speed) based on a fluid-flow analogy to traffic operations. The simulation logic of FREFLO can be expressed as several basic equations. The following notation is used:

- $j$  = freeway section;
- $n$  = time period;
- $l_j$  = number of lanes;
- $\Delta x_j$  = section length in kilometers;
- $\rho_j^n$  = section density;
- $u_j^n$  = section space-mean-speed;
- $q_j^n$  = flow rate (or volume);
- $f_j^{ON,n}$  = on-ramp volume; and
- $f_j^{OFF,n}$  = off-ramp volume.

The first equation expresses the conservation of vehicles in each section  $j$  and the time period  $n$ :

$$\rho_j^{n+1} = \rho_j^n + \frac{\Delta t}{l_j \Delta x_j} (l_{j-1} q_j^{n+1} - l_j q_{j+1}^{n+1} + f_j^{ON,n+1} - f_j^{OFF,n+1}) \quad (10.2)$$

The off-ramp volume is given by:

$$f_j^{OFF,n+1} = \beta_j q_j^{n+1} \quad (10.3)$$

where  $\beta_j$  is the fraction of exiting vehicles in section  $j$ .

Under uniform conditions, the volume, density, and speed are related by:

$$q_{j+1}^{n+1} = \rho_j^n u_j^n \quad (10.4)$$

Speed is given as a function of density:

$$u_{j+1}^{n+1} = u_j^n - \Delta t \left\{ u_j^n \left( \frac{u_j^n - u_{j-1}^n}{\Delta x_j} \right) + \frac{1}{\Delta T_j} \left[ (u_j^n - U_e(\rho_j^n)) + \frac{v_j}{\rho_j^n} \left( \frac{\rho_{j+1}^n - \rho_j^n}{\Delta x_j} \right) \right] \right\} \quad (10.5)$$

where  $T_j = K_r x_j$  and  $v_j = K_v x_j$ .  $U_e(\cdot)$  is the equilibrium speed-density relationship,  $K_r$  and  $K_v$  are parameters to be calibrated, known as the relaxation time and anticipation coefficients, respectively. A good choice for  $K_r$  and  $K_v$  is 46 second/km and 40 km/hour, respectively [72].

The underlying logic in the above equation is that vehicles traveling at speed  $u_{j-1}$  in the upstream section tend to continue to travel at that speed as they enter section  $j$ ; drivers have the tendency to adjust their speeds to an equilibrium speed-density relationship and tend to slow down if the density in the section ahead seems to be increasing.

FREFLO simulates incidents by their effect on capacity, that is, by reducing the number of available lanes and the flow passing through the incident site, or by altering some model parameters such as  $K_r$  and  $K_v$ .

The flow in the freeway,  $f_j^{ON,n}$ , at ramp  $j$ , is given by:

$$f_j^{ON,n} = \min \left\{ r_j^n, \frac{\lambda_j^n}{\Delta t} + d_j^n \right\} \quad (10.6)$$

where:

- $r_j^n$  = metering rate;
- $\lambda_j^n$  = queue length in vehicles;
- $\Delta t$  = time interval;
- $d_j^n$  = demand.

The queue length is estimated by the expression:



$$\lambda_j^{n+1} = \lambda_j^n + (d_j^n - f_j^{ON,n}) \Delta t \quad (10.7)$$

where  $\lambda_j^n$  is the queue length in vehicles on the ramp entering section  $j$  in the  $n^{th}$  time interval.

The performance measures produced from FREFLO include service rate<sup>5</sup>, travel time, speed, on-ramp metering rates, pollution emissions, fuel consumption, etc.

FREFLO represents a typical macroscopic simulation model which sacrifices a great deal of detail but gains efficiency and ability to deal with problems of large networks. This model was programmed in FORTRAN. FREFLO has undergone only limited calibration and validation but has shown considerable promise [71]. It has been used in several studies for the development and evaluation of ramp-metering strategies and response to freeway incidents.

## CORQ

The CORQ (CORridor Queuing) model was developed for modeling flows and queues in freeway corridors based on time-varying traffic demands and traffic assignment logic [94]. It was designed as a tool to help traffic analysts assess system-wide effects of traffic control strategies. CORQ can also be used for design purposes since it has a partial optimization module for selecting ramp metering rates.

CORQ was designed for modeling the critical elements of a corridor in terms of traffic flow, capacity, queuing and delay. Ramp metering rates can be set as input or simulated as output using an optimization module. Traffic lights and signals are not simulated directly; instead, intersection approaches and turning movements are assigned time costs according to traffic density. Travel demands are treated as time dependent flows. CORQ simulates traffic flow as fluid and calculates travel times as simple step function of flows. The data requirements for CORQ include:

---

<sup>5</sup> Service rate for a freeway section, measured in vehicle miles per hour, is the product of the number of lanes, length of the section and the corresponding flow in each time interval.

- Capacities;
- Time cost for intersection approaches and movements;
- Flow-cost relationships;
- Time dependent O/D trip tables;
- Queue sizes and traffic counts.

CORQ divides the peak period into short time slices so that the rates of demand between various O/D pairs can be considered constant (approximately 15 minutes intervals). Vehicles corresponding to an O/D matrix are assigned to the network sequentially in time from their origins and are taken out at their destinations. The assignment is based on the principle of minimizing individual's travel time. Over-saturation may occur on some links temporarily. That is, in any time slice, certain links may have more demand than they can serve, and queues are formed. Queues dissipate at a constant rate over the time slice. If a queue exists at the end of a time slice, it is taken out of the network and fed back in as new demand (originating from the downstream node) in the next time slice.

CORQ has the capability of changing network characteristics at the beginning of each time slice. That is, capacities can also vary as flows vary, especially in weave sections and at merges. CORQ assumes that drivers know the travel times of all the links for the present time slice and choose the current best path. If a queue occurs on that path, the driver may re-select the path based on new information when he/she is ready to leave the queue. This traffic assignment approach in CORQ may force long trips that should require travel for more than one time slice to arrive at their destination in one time slice (if there is no queue on their paths).

The outputs of CORQ are the link flows, queues, delay and travel time estimates for the entire freeway corridor. CORQ can also be used to select ramp metering rates.

## **KRONOS**

KRONOS is a macroscopic freeway simulation model developed at the University of Minnesota <sup>[63]</sup>. This model is microcomputer-based and features simulation of merging, diverging and weaving traffic dynamics in freeways.

Due to the macroscopic nature of the program and the model simplicity, input requirements were kept to a minimum. Depending on the amount of detail desired, one can select a reduced or an extended input version. The former employs a predetermined default set of values for the model parameters, while in the extended input version, these parameters can be user specified. In general, the input to KRONOS includes:

- Geometries of the freeway under consideration
- Freeway characteristics
- Speed-density model
- Free-flow speed
- Minimum speed for estimating delays
- Jam density and the capacity of each section and ramp
- Initial conditions
- Time slice traffic arrival and departure data at the beginning and end of the mainline section as well as for the on and off ramps.

KRONOS divides time and space into short increments  $dt$  and  $dx$  respectively, and employs simple continuum flow models. In other words, this model is based on the conservation equation and equilibrium speed-density relationships (similar to FREFLO).

KRONOS can generate intermediate outputs after each time slice and summarize the output after the simulation is completed. The reports provide measures of effectiveness in terms of flow, density, delay and queue length. Environmental parameters are also available for estimating noise, emissions and fuel consumption. The output data with respect to speed, flow or density can be imported to graphics software and displayed in 3-dimensional contours (time, space, value).

KRONOS was programmed in Pascal and runs on PCs. The model can simulate freeway sections up to six lanes wide, approximately 10 miles long with up to 20 on and 20 off ramps.

## **CONTRAM**

CONTRAM (CONtinuous TRAffic Assignment Model) was developed by the Transport and Research Laboratory (TRRL) for modeling time varying traffic

demands on urban networks subject to capacity restraints and transient overload. The model predicts the variation through time of the resulting routes, queues and delay. The latest version, CONTRAM 5, was released in October 1988.

CONTRAM provides analysts with the choice of minimum travel-time or generalized cost assignment. The data requirements for CONTRAM can be grouped into three categories: network data, traffic demand, and control parameters. The network is described with origins, destinations, intersections, and links. More than one link can be used between adjacent intersections to model different streams of traffic, and three types of links, namely, give way, signal controlled and uncontrolled links, can be represented. CONTRAM supports a full range of intersection types including signal controlled, major/minor intersection and roundabouts. Each intersection can be treated as isolated or as a part of a coordinated signal system along an arterial. The signal coordination model in CONTRAM can adopt fixed time plans for signal control or "optimized" signal plans for individual intersections at the current iteration, based on the flows in the proceeding iteration.

In CONTRAM, traffic demands are converted into passenger car unit (pcu) equivalents and vehicles are grouped into packets by time intervals and vehicle types. Three vehicle classes are specified: cars, buses (fixed route) and trucks. The packet information includes entry time, origin, destination, packet size, vehicle class, and routing information.

CONTRAM uses speed-flow relationships whose general form consists of two linear sections of different slopes. The exact form is determined by entering as data three points -- the free flow speed (where flow is zero), "break point" where the slope changes, and the capacity speed. The speed-flow model in CONTRAM may be specified for light and heavy vehicles separately.

The generalized cost of travel along a link can be used if the required data is available. This function is defined as a linear combination of distance, travel time, the square of the speed, queuing delay, toll attached to the link, probability to be stopped at an intersection, link type and "incidence factor" attached to the link.

CONTRAM provides a data entry system for preparing and modifying the input data interactively. This preprocessor accepts free format entry of data. It checks for logical errors and values that are outside predefined ranges, calculates saturation flow values using geometric data and sets up the time varying flow profiles for the O/D demand matrix.

The basic approach employed by CONTRAM is to divide the peak period into time intervals (usually 10-20 minutes long). Traffic demands are defined as a time-varying set of flows, one for each O/D movement at a given time interval. The vehicles for each O/D movement are grouped into packets and sequentially assigned to the network at a uniform rate for each time interval. Each such packet is indivisible and travels along its own individual minimum path to its destination.

The link travel times used to update the path tree include two components: cruise time, which is estimated based on assumed "reasonable" speed and the length of the link, and delay at downstream stopline, which is estimated by:

$$t^{delay} = \frac{L+1}{M} \times 3600 \quad (\text{seconds}) \quad (10.8)$$

where  $L$  is the number of queuing vehicles (pcu), and  $M$  (pcu/hr) is the throughput capacity. The throughput capacity is defined as the average flow rate at which vehicles discharge from a queue on a link. The throughput capacity takes into account the mix of different classes of vehicles, and it is calculated separately for signal controlled links, give way links and uncontrolled links.

Flows and queues on links are calculated for each time interval in the assignment procedure and are carried over from one time interval to the next. For each time interval, route choice is modeled iteratively until the procedure has converged satisfactorily. This assignment procedure for modeling route choices considers each driver as a marginal user and iterates until a traffic equilibrium is achieved. The procedure is summarized as follows:

- Remove a packet from the flows stored for each link in the appropriate time interval;
- Calculate queues and travel times on links affected;

- Assign removed packets to next link according to the new minimum travel time paths;
- Take the next packet and repeat the steps.

This process is repeated until packet routing is stabilized. Usually 5 to 10 iterations are adequate for practical purposes. The assignment procedure requires that all vehicles reach their destinations before the end of the simulation period, therefore, a final clearance interval with zero demand and sufficient length is designed.

A fuel consumption model is built in CONTRAM, and it takes into account the following components:

- Fuel consumed at steady cruise speed;
- Additional fuel consumed while accelerating;
- Fuel saved when decelerating, and;
- Fuel consumed while stationary or moving up in a queue.

The parameters of the fuel consumption model can be input separately for each of the three vehicle classes.

The output of CONTRAM includes aggregate performance measures and detailed (time, space or vehicle type specific) information.

- Overall summaries of travel times, distance traveled, average speed and fuel consumption;
- Time variation in flows, queues, delays, percentage saturation, total time spent and average speed on links;
- Identification of blocking when it occurs;
- Summary tables, for each time interval, of flows, queues, mean queue times and average speeds on links;
- Turning movements at intersections for each time interval;
- Travel times, distance traveled and fuel consumption, for different classes of vehicles;
- Point-to-point average speeds for selected O-D movements;
- Vehicle route information;

- Convergence parameters.

Furthermore, a post-analysis module is provided which analyzes CONTRAM's outputs, and facilities:

- Calculation of simple statistics;
- Comparison of up to 5 different runs of CONTRAM (scenarios);
- Graphic representation of the selected parameters in the form of histograms or annotated network diagrams.

CONTRAM was programmed in FORTRAN. The main advantage of CONTRAM is its dynamic traffic assignment technique. The main drawbacks are the lack of freeway routines, the simplified representation of traffic signals, the need to explicitly store all vehicle packet routes, and the simplistic user behavior logic. CONTRAM has been used extensively for IVHS applications in Europe.

## **DYNASMART**

DYNASMART is a macroscopic traffic simulation model developed for modeling real-time route guidance systems in general urban traffic networks at the University of Texas at Austin<sup>[55]</sup>. The model provides the capability of traffic simulation and assignment with explicit consideration of user route and departure time choices. It is derived from an earlier model named MPSM (Macroparticle Simulation Model) which is a macroscopic highway corridor traffic simulation model specially designed for studies of commuter departure times. Initially the model represented traffic as discrete vehicle packets that were moved according to the speeds defined by local concentrations. The new model (still under development) tracks individual vehicles (while it still uses macroscopic relations of speed and concentration). It also has traffic signal timing and coordination capabilities.

The model is a time-based macroscopic simulation model. The traffic flow is represented as a set of particles. Each particle is a collection of vehicles with the same destination and departure time. The model keeps track of the physical position of these particles using a prespecified speed-density relationship.

The model consists of two principal components. The first component, or vehicle generation component, processes the daily decisions of users into discretized time-dependent demand patterns. The generation functions may directly specify the demand pattern for a particular simulation day. Alternatively the model may formulate the time-dependent departure patterns endogenously by using specified decision rules to determine vehicles' departure time choices on a given simulation day (in response to the service levels experienced on previous days). The second component actually simulates the flow of traffic, including queuing at entrance ramps and vehicle movement along the highway. A deterministic queuing approximation is used to determine delays. Vehicles leaving the queue are subsequently grouped in packets for moving on the highway. Packets are moved at each iteration based on the speed determined by speed-concentration relationships. The concentration in freeway section  $i$  is calculated by:

$$K_i^{t+1} = K_i^t + \frac{M_i^{t+1} - M_{i+1}^{t+1} + N_i^{t+1}}{L_i \Delta x_i} \quad (10.9)$$

where:

- $K_i^t$  = concentration in section  $i$  during the  $t^{\text{th}}$  time interval, in vehicles per lane-mile;
- $\Delta x_i$  = length of section  $i$ ;
- $L_i$  = number of lanes of section  $i$ ;
- $N_i^t$  = number of vehicles generated in section  $i$  minus those exiting from section  $i$  during the  $t^{\text{th}}$  time interval;
- $M_i^t$  = the vehicles that enter section  $i$  from the preceding section;

The concentration in each section is updated at the beginning of every time interval, and it is assumed to remain constant over the interval. The updated concentration is used to compute the mean speed prevailing during this time interval, according to a speed-concentration relation, such as:

$$V_i^t = V_0 + (V_f - V_0) \left( 1 - \frac{K_i^t}{K_0} \right)^\alpha \quad (10.10)$$

where:



- $V_i'$  = mean speed in section  $i$  during the  $t^{\text{th}}$  time interval;
- $V_f$  = mean free speed;
- $V_0$  = minimum speeds;
- $K_0$  = maximum or jam concentration;
- $\alpha$  = a parameter to be calibrated.

The simulation proceeds from one day to the next, as the individual decisions determined in the user-decision subroutine are internally transformed into time-dependent departure patterns. The process iterates a prespecified maximum number of days or until steady state is reached, whatever occurs first.

Inputs to the model include the physical and operational features of the highway facility, such as the total length, the number of sections the highway is subdivided into, the number of lanes, the free flow speed in each section, the parameters of the speed-density model. The model outputs the following information:

- The concentration fluctuation in each section along the highway corridor for a given time-dependent demand;
- The variation of travel time versus departure time for a given origin-destination pattern;
- The day-to-day evolution of the system's performance under various postulated rules that might govern users' departure time choice behavior.

To keep a balance between simulation accuracy and computational efficiency, typically the simulation time step  $\Delta t$  is 1 minute, freeway section length  $\Delta x$  is 1 mile, and particle size is 5 to 10 vehicles. The model was programmed in FORTRAN and runs on a CRAY supercomputer. No validation of the model has been reported in the literature but some applications on realistic size networks have taken place.

## THOREAU

THOREAU (Traffic and Highway Objects for Research, Analysis, and Understanding) [13, 14] is a microscopic simulation model specifically designed for comparative evaluation of ATMS and ATIS operating in multiple, diverse environments. This model is still under development, using the object-oriented

paradigm, by MITRE Corp. THOREAU integrates both microscopic and mesoscopic simulation approaches into one model, and different parts of the network can be simulated with different levels of detail.

For mesoscopic simulation (a link and/or a node is specified as macro-link), each vehicle is moved from a link segment to its neighboring segment according to analytic speed-flow-density and queue dispersion equations. For the microscopic simulation, individual vehicles are processed by an event-based routine, in which vehicle movements are maneuvered by actions defined by their current positions, speeds, and available headways. More specifically, four vehicle states are defined:

- Stopped (S)
- Uniformly accelerating (A)
- Traveling with uniform speed (U)
- Uniformly decelerating (D)

The transition from one state to another is based on comparison of the distance of the vehicle under consideration from the vehicle ahead to the minimum distance headway. For a leading vehicle, this distance is set equal to the distance from the downstream node of the link. The minimum headway used in state transition is a function of speed. Turns, lane changes, and merging at intersections are processed as required.

THOREAU can be used to assess network performance under IVHS, and provide measures such as trip delay, stop time, average travel speed, etc. Its vehicle routing algorithm distinguishes guided and unguided vehicles. For vehicles without route guidance devices installed, a vehicle always follows its predetermined static path, while vehicles with route guidance devices installed are rerouted whenever a significant reduction in remaining trip time can be achieved. The shortest remaining path is determined by the Floyd algorithm [27].

In addition, THOREAU can also simulate incidents such as lane blockages or slow downs. Shortest paths are recalculated when an incident occurs or when it is cleared, and ATIS-equipped vehicles are notified.

THOREAU has graphics capabilities to animate vehicle movement and presents color map for user-selectable performance parameters (such as traffic

volume, average speed, volume/capacity ratio, etc.) during the simulation process. These graphic features help debugging and monitoring of the simulation process, but, if used, they slow down the simulation (by a factor of three).

THOREAU is implemented with the MODSIM II simulation language. Instead of using fixed formats for input files, this model uses a key-word match strategy, a feature that provides flexibility and expandability for input data and implementing strategies to be tested. A major drawback of THOREAU is its inefficiency with respect to running time and memory requirements (especially in modeling larger urban networks). It is reported that simulating a 12-minute time period for the town of Troy, Michigan (roughly 400 links and 100 nodes) consumes 40-50 MB of virtual memory and may take a few days of running time [14].

The inputs to THOREAU include node, link, route data (for vehicles with fixed paths), and signal plans; origin/destination traffic demand matrices; and incident data. The ATMS and/or ATIS strategies to be examined are not directly supported. The user has to implement them using the functions provided in THOREAU.

The current version of the model provides various performance measures including average travel time and delay for each O/D pair, average speed on each link, average stop time at each node, etc. The distribution of these data, which might be used to calculate energy consumption, pollution, or congestion cost in post-processor routines, is collected using the statistics routine built into the simulation language.

## **INTEGRATION**

INTEGRATION is a mesoscopic time-based traffic simulation model for modeling time varying traffic demand on general networks. This model was developed at Queen's University through the sponsorship of Ministry of Transportation, Ontario, Canada. The first version of the model was developed and tested in 1984, and it has been extensively revised since then [85, 88]. The model can be used for both freeway/arterial and urban street networks, and can incorporate user-specified signal timing plans and ramp metering (fixed-time or

traffic actuated plans). The model supports routines for O/D estimation, equilibrium assignment, signal optimization, and an AUTOLISP plotting interface. INTEGRATION also provides animation of vehicle movements, routes, incidents and controls.

Traffic flows in INTEGRATION are modeled in terms of individual vehicles which have self-assignment capabilities and are moved through the network in accordance with speed and minimum headways. Traffic demands are represented with time-dependent departures. The randomness of departure times is captured by using stochastic departure rate functions.

Vehicles enter the network from their origins at scheduled departure times, and select appropriate links based on the minimum path trees. Following its entry onto a link, a vehicle stays on that link for a period of time which is at least equal to that link's travel time. When this time has expired, the vehicle is moved onto the next appropriate link towards its destination according to the shortest path. Upon arrival to its destination, a vehicle is removed from the network and the associated statistics are recorded.

Different routing algorithms can be used for vehicles in INTEGRATION. These routing algorithms are based on:

- Externally specified path trees. If no external path trees are provided, sampling of link travel times are performed based on historical information, and stochastic shortest path trees are generated.
- Virtually continuous updated real-time link travel times.
- Anticipatory knowledge of expected future link travel times (a combination of real-time and historical information).
- Accessibility to high occupancy vehicle (HOV) links.

Capacity of links is determined in terms of a minimum headway, which is a function of speed. When a vehicle's expected departure time is expired but it can not be served because of capacity constraints, this vehicle queues at the corresponding link. The queue could spillback if the number of vehicles exceeds the link's storage capacity.

The simulation clock is advanced in deciseconds. The minimum path tree is updated when it is necessary (every 6 seconds). The basic modeling approach essentially manipulates two stacks of vehicles. One is the list of vehicles in departure sequence, and the other is a sorted list of all vehicles on each link. Each time a vehicle enters a new link, its departure time from the link is estimated as the current clock time plus the prevailing link travel time (exclusive of queuing). The first vehicle in each link stack is checked every clock increment to determine if it is eligible for departure. This check considers the vehicle's departure time, capacity or traffic signal restrictions, and queue spillbacks.

INTEGRATION was initially implemented in both C and compiled BASIC. Since its initial development in 1984, the model has undergone extensive revisions to deal with larger and more sophisticated networks and be useful for IVHS applications. The current version of the model has been implemented in FORTRAN, and it can handle networks up to 10,000 nodes, 20,000 links, and 1,000 traffic signals on 486/64MB PCs. Up to 1,000,000 vehicles (5 vehicle types) and 20 concurrent incidents can be modeled. In addition, several models were developed and attached as satellites of INTEGRATION:

- **ASSIGN:** provides routing information for guided vehicles using a pseudo-dynamic equilibrium traffic assignment algorithm;
- **QUEENLQD:** estimates dynamic synthetic O-D matrix;
- **MULTIPATH:** generates multiple "roughly equal shortest" routes and computes weights associated with the paths identified;
- **Q-PROBE:** simulates the two-way communications between the vehicle and a traffic management center, and provides INTEGRATION with the estimated seeds of link travel time, O-D matrix and link flow required by other modules;
- **REAL-TRAN:** simulates the surveillance system and traffic controls. The model is basically a modification of TRANSYT-7F, and it simulates optimized signal timing settings of a SCOOT-like traffic signal system.

In its basic form INTEGRATION can be considered as a vehicle-mover model.

# DYNAMIC TRAFFIC MODEL

The Dynamic Traffic Model (DTM) is currently under development at MIT [3, 66]. The model will be used to support (in real time) integrated ATMS/ATIS operations.

Successful operation of ATMS and ATIS requires decision support systems that operate in real time and are able to:

- Obtain sensor data collected by the *surveillance system* to assess the current state of the network and predict future traffic conditions using a *Congestion Prediction* model; and
- Develop, through a *Control Strategy Generator*, routing recommendations and traffic signal plans on the predicted traffic conditions.

The DTM incorporates real-time sensor data, congestion prediction, and routing and control strategy generation into a single framework. Driver behavior, origin/destination flow prediction, and dynamic traffic assignment models are also included to provide the information (in addition to sensor data) required in order for the congestion prediction and control strategy generator models to operate properly.

The various components of the Dynamic Traffic Model and their interactions, are shown schematically in Figure 10.2. The driver behavior model predicts path choices as well as response to guidance information for various driver classes (e.g., guided, unguided). The OD Estimation Model provides real time estimates of expected time-dependent OD flows, using historical OD matrices, historical and actual link counts and actual travel times between OD pairs.

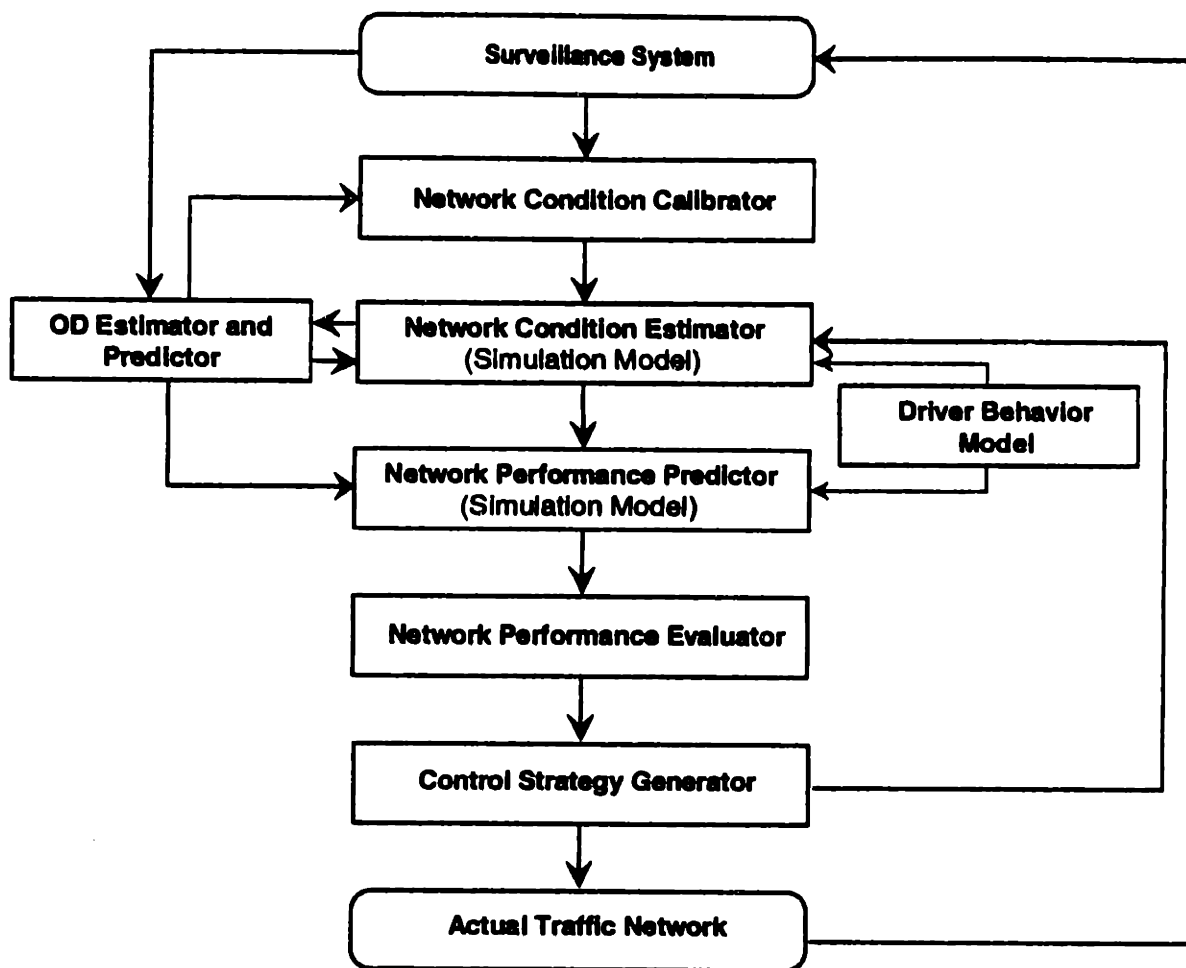


Figure 10.2 Overall Structure of DTM  
(source: Ben-Akiva *et al*, 1992b)

The Network Condition Estimator (NCE) is a real time simulation model that estimates the current state of the network. The output from NCE is a combination of surveillance data and simulation data for the links and intersections where surveillance data is not available. The Network Condition Calibrator (NCC) is used to calibrate the functions and parameters of NCE by detecting and correcting discrepancies between the simulated state of the network and the sensor data. NCC uses as input the *Control Strategies* (route guidance and traffic signal controls) currently implemented in the actual network.

The Network Performance Predictor (NPP) projects the current network state estimate obtained from the NCE into a fixed time period in the future. This prediction is based on the future estimates of OD flows obtained from the OD Estimation Model and the control strategies implemented in the network. The

NPP also uses the Driver Behavior Model for predicting driver path choices and driver responses to guidance information.

The objective of the Control Strategy Generator is to generate new *Control Strategies*. Instrumental in the development of appropriate control strategies is the Network Performance Evaluator (NPE). NPE has as input the state of the network, as predicted by NPP. Based on user defined performance measures NPP evaluates the performance on the network and decides whether or not the control strategy currently evaluated is acceptable. If the control strategy is not acceptable a new strategy is generated by the Control Strategy Generator. The NPP again predicts the future state based on the new control strategy. This loop continues until an acceptable control strategy is obtained. This control strategy is then implemented in the actual network and is also fed back to the NCE which uses it for updating the network state, during the next iteration.

The main components of the NPP and NCE modules is a flexible and efficient traffic simulation model. The simulation is discrete time-based in which vehicles (or group of vehicles) are handled as individual units. The performance of the network is simulated in detail with emphasis on the representation of queues and spill-backs. Links consist of two parts: a running part and a queuing part. The length of each part depends on the current length of the queue, if any, that is formed on the links. Deterministic queuing models are used to estimate delays on the queuing part, while macroscopic speed-density relationships are employed to estimate travel times on the running part.

The simulation model provides a flexible modeling tool that satisfies the requirements of both NCE and NPP. The NCE simulates the network in real time while providing an accurate estimate of the actual network conditions. The NPP, on the other, needs to simulate faster than real time but does not need the same level of accuracy as the NPP. To satisfy the above requirements the granularity of the simulation can be adjusted by defining the time step increments and grouping vehicles with similar travel characteristics together as single units. Furthermore, in order to improve its real time capabilities, the Dynamic Traffic Model supports multiprocessing and distributed processing computational environments.



The model is at an early stage of development and no applications have been reported yet.

# Bibliography

- [1] BDM International, Inc., "Project Gridlock Report," report prepared for Ford Motor Company, 1991.
- [2] Beckman, M.J., C.B. McGuire, and C.B. Winsten, *Studies in the Economics of Transportation*, Yale University Press, New Haven, Conn.
- [3] Ben-Akiva, M., H.N. Koutsopoulos and A. Mukundan, "A Dynamic Traffic Model System," in *Proceedings of the Second International Seminar on Urban Traffic Networks*, Capri, Italy, July 1992.
- [4] Ben-Akiva, M., D. Bernstein, A. Hotz, H. Koutsopoulos, and J. Susman, "The Case for Smart Highways." *Technology Review*, Vol. 95, No. 5, pp. 39-47, 1992.
- [5] Ben-Akiva, M., M. Cyna and A. de Palma, "Dynamic Model of Peak Period Congestion," *Transportation Research*, Vol. 18B, No. 4/5, 1984, pp. 339-355.
- [6] Ben-Akiva, M. and S.R. Lerman, *Discrete Choice Analysis - theory and Application to travel Demand*, The MIT Press, Cambridge, Massachusetts, 1985.
- [7] Ben-Akiva, M., A. de Palma, and I. Kaysi, "Dynamic Network Models and Driver Information Systems," *Transportation Research*, Vol. 25A, No.5, 1991.
- [8] Benekohal, R.F. and Joseph Treiterer, "CARSIM: Car-Following Model for Simulation of Traffic in Normal and Stop-and-Go Condition," *Transportation Research Record 1194*, TRB, pp. 99-111.
- [9] Bullen, A.G., "Development of Compact Microsimulation for Analyzing Freeway Operations and Design," *Transportation Research Record 841*, TRB, 1982.
- [10] Bullen, A.G., P. Athol, "Development and Testing of INTRAS - A Microscopic Freeway Simulation Model," *Freeway Dynamics Component Development*, Vol. 2.
- [11] Chandler, R.E., R. Herman, and E.W. Montroll, *Traffic Dynamics: Studies in Car-Following*, *Operation Research*, Vol. 6, No. 2, 1958, pp. 165-184.
- [12] Chang, G., H.S. Mahmassani, and R. Herman, "Macroparticle Traffic Simulation Model to Investigate Peak-Period Commuter Decision Dynamic," *Transportation Research Record 1005*, pp. 107-121.

- [13] Codelli, D.K., W.P. Niedringhaus, and P. Wang, "The Traffic and Highway Objects for REsearch, Analysis, and Understanding (THOREAU) IVHS Model," Volume I, MITRE, McLean, Virginia, MTR 92W0000208V1, November 1992.
- [14] Codelli, D.K., W.P. Niedringhaus, and P. Wang, "User Manual for Traffic and Highway Objects for REsearch, Analysis, and Understanding (THOREAU) IVHS Model," Volume II, MITRE, McLean, Virginia, MTR 92W0000208V2, November 1992.
- [15] Coombe, R.D., T.J. Annesley and R.P. Goodwin, "The Use of CONTRAM in Bahrain," *Traffic Engineering and Control*, March 1983.
- [16] Daganzo, C.F. and Y. Sheffi, "On Stochastic Models of Traffic Assignment," *Transportation Science*, Vol. 11, No. 3, pp. 253-274.
- [17] Drew, D.R., "Deterministic Aspect of Freeway Operation and Control," *Highway Research Record* 99, 1965, pp. 48-58.
- [18] Drake, J.S., J.L. Schoefer and A.D. May, "A Statistical Analysis of Speed Density Hypotheses," *Highway Research Record* 154, 1967, pp. 53-87.
- [19] Edie, L.C., "Car-Following and Steady-State Theory for Non-Congested Traffic," *Operation Research*, Vol. 9, 1961, pp. 66-76.
- [20] FHWA, Development and Testing of INTRAS, a Microscopic Freeway Simulation Model, Vol. 1. Program Design, Parameter Calibration and Freeway Dynamics Component Development, Report No. FHWA/RD-80/106, October, 1980.
- [21] FHWA, Development and Testing of INTRAS, a Microscopic Freeway Simulation Model, Vol. 2. User's Manual, Report No. FHWA/RD-80/107, October, 1980.
- [22] FHWA, Development and Testing of INTRAS, a Microscopic Freeway Simulation Model, Vol. 3. Validation and Application, Report No. FHWA/RD-80/108, October, 1980.
- [23] FHWA, *Traffic Control Systems Handbook*, U.S. DOT, FHWA-IP-85-11, April 1985.
- [24] FHWA, SCOT Model, User's Manual and Program Documentation, Working Document, 1975.
- [25] FHWA, Computer Programs for Urban Transportation Planning: PLANPAC/BACKPAC General Information Manual, U.S. DOT, Washington, D.C.
- [26] FHWA, *Handbook of Computer Models for Traffic Operations Analysis*, U.S. DOT, FHWA-TS-82-213, 1982.

- [27] Floyd, R. L., Algorithm 97: Shortest Path, *Communication of ACM* 5, 1962, pp.345.
- [28] Forbes, T.W., H.J. Zagorski, E.L. Holshouser, and W.A. Deterline, Measurement of Driver's Reaction to Tunnel Conditions, *Highway Research Board, Proceedings*, Vol. 37, 1958, pp. 345-357.
- [29] Forbes, T.W., *Human Factor Consideration in Traffic Flow Theory*, Highway Research Board, Record 15, HRB, Washinton D.C., 1963, pp. 60-66.
- [30] Forbes, T.W., and M.E. Simpson, Driver and Vehicle Response in Freeway Deceleration Waves, *Transportation Science*, Vol. 2, No. 1, 1968, pp. 77-104.
- [31] Herman, R., E.W. Montroll, R. Potts, and R.W. Rothery, Traffic Dynamics: Analysis of Stability in Car-Following, *Operation Research*, Vol. 1, No. 7, 1959, pp. 86-106.
- [32] Herman, R., and R.B. Potts, Single-Lane Traffic Theory and Experiment, *Theory of Traffic Flow Symposium Proceedings*, 1961, pp. 120-146.
- [33] Herman R., and R.W. Rothery, Microscopic and Macroscopic Aspects of Single-Lane Traffic Flow, *Journal of Operations Research Society of Japan*, Vol. 5, 1962, pp. 74-93.
- [34] Herman, R., and R.W. Rothery, Car-Following and Steady State Flow, *Theory of Traffic Flow Symposium Proceedings*, 1963, pp. 1-13.
- [35] Gazis, D.C., R. Herman, and R.B. Potts, Car-Following Theory of Steady State Flow, *Operation Research*, Vol. 7, No. 4, 1959, pp. 499-505.
- [36] Gazis, D.C., R. Herman, and R.W. Rothery, Non-Linear Following-the-Leader Models of Traffic Flow, *Operation research*, Vol. 9, No. 4, 1961, pp. 545-567.
- [37] Gipps, P.G., A Behavioral Car-Following Model for Computer Simulation, *Transportation Research 15B*, 1981, pp. 105-111.
- [38] Greenberg, H., "An Analysis of Traffic Flow," *Operation Research*, Vol. 7, No. 1, 1959, pp. 79-85.
- [39] Greenshields, B. D., "A study in Highway Capacity," Highway Research Board Proceeding 14, 1934, pp. 448-477.
- [40] Institute of Transportation Engineers, *Transportation and Traffic Engineering Handbook*, 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1982, 883 pages.
- [41] IVHS America, "Strategic Plan for Intelligent Vehicle Highway Systems in the United States," Summer 1992.
- [42] JHK and Associates, IVHS Functions and Technologies Deminar: IVHS Program for Metropolitan Boston, Massachusetts Highway Department.

- [43] Jones, E.R. and R.T. Hennessy, *Human Factor Aspects of Simulation*, National Academy Press, Washington, D.C., 1985.
- [44] Junchaya, T, G.L. Chang and A. Santiago, "ATMS: Real-time Network Traffic Simulation Methodology with a Massively Parallel Computing Architecture," Transportation Research Board, 71st Annual Meeting, Washington, D.C., Jan, 1992.
- [45] Kaysi, I.A., "Framework and Models for the Provision of Real-Time Driver Information," Ph.D Dissertation, Department of Civil Engineering, MIT, February 1992.
- [46] Kornhauser, A.L., "Exploring Transportation Problems Using Interactive Computer Graphics," *Computer and Graphics*, Vol. 11, No. 3, 1987.
- [47] Koutsopoulos, N.K. and Q. Yang, "Modeling Requirements for Emerging Transportation System Operation Environments," a report to John A. Volpe National Transportation Systems Center, Department of Civil and Environmental Engineering, MIT. DTRS-57-88-C-0078, TTD#29, 1992.
- [48] Larrabee, D. and R. Shull, "Maximizing Your Use of Interactive Graphics in Transportation Simulation Modeling," *Microcomputers in Transportation III*, Vol.2, pp.1229-1231, 1989.
- [49] Leonard, D. *et al*, "CONTRAM: A Traffic Assignment Model for Predicting Flows and Queues During Peak Periods," *TRRL Report LR 841*, 1978.
- [50] Leonard, D. and P. Gower, "User Guide to CONTRAM Version 4," *TRRL Report SR 735*, 1982.
- [51] Leonard, D., P. Gower and N. Taylor, "CONTRAM: Structure of the Model," *TRRL Research Report 178*, 1989.
- [52] Lieberman, E., "Simulation of Corridor Traffic: the SCOT Model," *Highway Research Record 409*, 1972.
- [53] Lieberman, E., *et al*, "Macroscopic Simulation for Urban Traffic Management: The TRAFLO Model," Federal Highway Administration, U.S. Department of Transportation, Vols. 1-7, 1980.
- [54] Lieberman, E., and B. Andrews, "TRAFLO: A New Tool to Evaluate Transportation System Management Strategies," *Transportation Research Record 772*, TRB, 1980, pp. 9-15.
- [55] Mahmassani, H.S., "DYNASMART", IVHS Dynamic Traffic Assignment and Simulation Workshop Attendee Response Form, The University of Texas at Austin, 1992.
- [56] Manheim, M.L., *Fundamentals of Transportation Systems analysis, Volume 1: Basic Concepts*, The MIT Press, Cambridge, Massachusetts, 1979.

- [57] May, A., and H.E. Keller, *Non-integer Car-Following models*, Highway Research Board, Record 199, HRB, Washington D.C., 1967, pp. 19-32.
- [58] May, A., *Traffic Flow Fundamentals*, Prentice Hall, Englewood Cliffs, New Jersey, 1990.
- [59] May, A., "Freeway Simulation Models Revisited," *Transportation Research Record* 1132, pp. 94-99.
- [60] May, A., "Models for Freeway Corridor Analysis," *TRB Special Report* 194, 1981.
- [61] McGurrin, M.F. and P. Wang, "An Object-Oriented Traffic Simulation with IVHS Applications," pp. 551-560.
- [62] McFadden, D., "Conditional Logit Analysis of Qualitative Choice Behavior," in P. Zarembka (Ed.), *Frontiers in Econometrics*, Academic Press, New York, pp. 105-142, 1973.
- [63] Michalopoulos, P., "A Dynamic Freeway Simulation Program for Personal Computers," *Transportation Research Record* 971, TRB, 1984.
- [64] Minnesota DOT and University of Minnesota, "KRONOS - The development and Operation of Freeway Simulation," 2nd International ASCE Conference on Applications of Advanced Technologies in Transportation Engineering.
- [65] Morlok, E.K., *Introduction to Transportation Engineering and Planning*, McGraw-Hill, New York, 1978.
- [66] Mukundan, Anil, "A Real-Time System for IVHS Traffic Modeling", Master Thesis, Civil and Environmental Engineering, MIT, 1993.
- [67] Nakatsuji, T., N. Kuwajima, and M. Aihara, "Traffic Simulation and Its Graphic Presentations," *Microcomputers in Transportation III*, Vol. 1, pp. 617-627, 1989.
- [68] Papageorgiou, M., B. Posch, and G. Schmidt, "Comparison of Macroscopic Models for Control of Freeway Traffic," *Transportation Research B*, Vol. 17B, No. 2, pp. 107-116, 1983.
- [69] Parkany, Emily, "Using Vehicle to Roadside Communications Data for Incident Detection", Master Thesis, Civil and Environmental Engineering, MIT, 1993.
- [70] Payne, H.J., "Models of Freeway Traffic Control," *Simulation Councils Proceedings*, Vol.1, No.1, 1971.
- [71] Payne, H.J. and H.M. Koble, "A Comparison of Macroscopic Simulation Models (MACK and INTRAS)," ORINCON Corp. Rept. OCR-78-11-9367-1, March 1978.

- [72] Payne, H.J., "FREFLO - A Macroscopic Simulation Model of Freeway Traffic," *Transportation Research Record 722*, TRB, 1979, pp. 68-75.
- [73] Pipes, L.A., An Operational Analysis of Traffic Dynamics, *Journal of Applied Physics*, Vol. 24, No. 3, 1953, pp. 274-287.
- [74] Richetta, A., "A Literature Review on the Use of Computer Graphics as Tools to Provide Interactive Use, Data Display and Model Testing for Computerized Traffic Control Systems," Traffic Control System, Term Project Report, MIT, 1992.
- [75] Roden, D., W. Okitsu, and A. May, "FREQ7PE - A Freeway Corridor Simulation Model," ITS, University of California, Berkeley, 1980.
- [76] Ross, P. and D. Gibson, "Survey of Models for Simulating Traffic," *Overview of Simulation in Highway Transportation*, Vol.1, 1977, pp. 36-48.
- [77] Rothery, R.W., R. Silver, R. Herman, and C. Tornor, Analysis of Experiments of Single-Lane Traffic Flow, *Operation Research*, Vol. 12, No. 6, 1964, pp. 913-933.
- [78] Skabardonis, A. and P. Loubal, "Applications of Computer Graphics in Traffic Control," Microcomputer Applications in *The Urban Transportation Environment*, pp.342-350, 1985.
- [79] Stephanedes, Y. J., E. Kwon, and P. Michalopoulos, "On-Line Division Prediction for Dynamic Control and Vehicle Guidance in Freeway Corridors," *Transportation Research Record 1287*, pp.11-19.
- [80] Thinking Machine Corp., "Connection Machine®: Model CM-2 Technical Summary (5.1)," Thinking Machine Corporation, Cambridge, Massachusetts, May 1989.
- [81] Thinking Machine Corp., "C\* Programming Guide (6.02)," Thinking Machine Corporation, Cambridge, Massachusetts, June 1991.
- [82] Tolle, J.E., Composite Car-Following Models, *Transportation research*, Vol. 8, 1974, pp. 91-96.
- [83] U.S. Bureau of Public Roads, *Traffic Assignment Manual*, U.S. Department of Commerce, Washington, D.C.
- [84] Van Aerde, M. and Y. Blum, "Electronic Route Guidance in Congested Networks," *ITE Journal*, August 1988.
- [85] Van Aerde, M. and S. Yagar, "Dynamic Integrated Freeway/Traffic Signal Networks: Problems and Proposed Solutions," *Transportation Research A*, Vol. 22A, No. 6, 1988, pp. 435-443.