Title

Simulation of ITS on the Irvine FOT Area Using "Paramics 1.5" Scalable Microscopic Traffic Simulator: Phase I: Model Calibration and Validation

Authors
Abdulhai, Baher
Sheu, Jiuh-Biing
Recker, Will

Publication Date
1999-04-01

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

Simulation of ITS on the Irvine FOT Area Using 'Paramics 1.5' Scalable Microscopic Traffic Simulator: Phase I: Model Calibration and Validation

Baher Abdulhai, Jiuh-Biing Sheu, Will Recker

California PATH Research Report
UCB-ITS-PRR-99-12

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

Report for MOU 359

April 1999

ISSN 1055-1425

Simulation of ITS on the Irvine FOT Area Using the 'Paramics1.5' Scalable Microscopic Traffic Simulator

Phase I: Model Calibration and Validation

Baher Abdulhai

Jiuh-Biing Sheu

Will Recker

PATH

April 12, 1999

Simulation of ITS on the Irvine FOT Area Using the 'Paramics1.5' Scalable Microscopic Traffic Simulator
Phase I: Model Calibration and Validation

Baher Abdulhai, Jiuh-Biing Sheu and Will Recker

EXECUTIVE SUMMARY

In this research, a promising ITS-ready microscopic traffic simulator, Paramics1.5, is thoroughly evaluated both subjectively and objectively in preparation for its use for modeling ITS in Southern California. A general and comprehensive list of requirements (evaluation template) for a microscopic traffic flow simulator to successfully model ITS was first compiled and presented. The evaluation template covered various aspects of modern transportation network modeling, including: [1] supply/control aspects, [2] demand/behavior aspects, [3] environmental-related aspects, and [4] simulator-performance aspects. The template was applied to Paramics1.5, a new and potentially successful ITS-oriented micro-simulator. The evaluation revealed Paramics1.5's capabilities and limitations as an ITS modeling tool. It was concluded that Paramics1.5 is an excellent 'shell' or 'framework' for a comprehensive and extensive transportation simulation laboratory. Paramics1.5 offers two very important and unprecedented features: high performance and scalability. To our knowledge, this is the most promising approach to handle realistic real word traffic networks under ITS. Also, Paramics1.5 offers very plausible detailed modeling for many components of the envisioned 'ideal' simulator. Nevertheless, several limitations were also identified and discussed. Suggestions for improvements were also presented. Improving such limitations requires both direct modifications in the software itself, and maturation of the Application Programming Interface (API), one of Paramics1.5' most important capabilities. The subjective evaluation was also followed by an objective evaluation based calibration of key driving-behavior parameters underlying -among other things- the car following and lane changing models of the simulator. Calibration was conducted using real traffic data from the California Testbed, headquartered at the University of California Irvine. The calibrated model performed well during validation on a freeway link. On a full network, the vehicle release mechanism showed some time-lag in releasing demand onto the network. This is potentially due to stacking of vehicles in memory before adequate headways are found on the road to release the vehicles. Although the problem itself is simple, its effects on the results were notable.

Key Words: Paramics1.5, ITS, Simulation, Dynamic Transportation Networks.

TABLE OF CONTENTS

INTRODUCTION

Intelligent Transportation Systems (ITS) appear to provide a set of intuitively promising tools to improve the increasingly complex and rapidly deteriorating transportation systems of today. Potential success of such ITS is implicit and derived to a large extent form the success of the underlying technology in other fields such as the defense and aerospace industries. Comprehensive research tools for quantifying the expected benefits from ITS are, however, still absent. In order to quantify potential benefits prior to any major investment in development and deployment, the use of traffic simulation is regarded inevitable, at present at least. Simulators are needed, not only to assess the benefits of ITS in a planning mode, but also to generate scenarios, optimize control, and predict network behavior at the operational level.

A fairly large number of traffic simulators are currently under development all over the world covering wide ranges of complexity, comprehensiveness, and potential usefulness. The majority of these simulators, however, have never been closely evaluated, neither calibrated nor validated. In this regard, researchers usually encounter three major obstacles: [1] the unavailability of a manageable network/corridor that is adequately instrumented with <u>functioning</u> vehicle detectors to supply the necessary data, [2] the absence of a comprehensive testbed that is capable of modeling and testing a variety of ITS components and of acquiring real-time traffic data, and [3] the unavailability of a <u>non-limiting</u> traffic simulator that is easy to use and yet capable of ITS modeling in a corridor/network environment of any realistic size.

The research presented in this report is an attempt to compile a list of attributes or a 'wish list' a successful traffic simulator should possess in order to adequately model Intelligent Transportation Systems (ITS), and calibrate, validate and evaluate one promising ITS-capable advanced traffic flow simulator, namely Paramics1.5. The research makes use of the unprecedented capabilities available at the ATMIS testbed and labs of University of California Irvine (UCI) that collectively present a potential solution to the above problems. These capabilities are founded on three pillars: [1] the fairly young, well instrumented and manageable Irvine network, [2] the variety of ITS modeling components of the UCI testbed and its on-line real-time data acquisition capabilities [3] the capabilities of a newly acquired ‘Paramics1.5’ microscopic traffic simulation and visualization tool. The objective is to evaluate Paramics1.5, both subjectively using the complied evaluation template and objectively through calibration and validation, in order to prepare it for a near-future second stage of ITS modeling in the Irvine network, California.

THE IRVINE NETWORK, THE UCI TESTBED AND PARAMICS1.5

The Irvine network is composed of two diverging freeways overlaying a non-grid arterial network and features a highly congested corridor through a rapidly growing employment center. The advantages of the Irvine area could be summarized as:

1. A well defined freeway-arterial corridor,
2. Connected to the UCI testbed with data links from the California Department of Transportation (Caltrans) District 12 Traffic Management Center (TMC),
3. Well instrumented with both loop and video detectors in working condition.

The UCI testbed itself was chosen for this research because of its advanced features. Detailed description of the testbed capabilities is beyond the scope of this writing. Very briefly however, these are:

1. Real-time data acquisition capabilities,
2. High-speed communications,
3. Real-time control capabilities,
4. Remote advanced surveillance and control,
5. Advanced simulation capabilities, and
6. ITS analysis capabilities.

The particular traffic simulator used in this study is ‘Paramics1.5’ by Quadstone, UK. Paramics1.5 is a suite of high performance software tools for microscopic traffic simulation. Individual vehicles are modeled in fine detail for the duration of their entire trip, providing accurate traffic flow, transit time and congestion information, as well as enabling the modeling of the interface between drivers and ITS. The Paramics1.5 software is portable and scalable, allowing a unified approach to traffic modeling across the whole spectrum of network sizes, from single intersections up to national networks.

Performance is a trade-off between number of vehicles simulated and the processing power available. The computing resource can be chosen to match the problem size. Such scalability enables development to start off small, with no concern of hitting a ceiling as models grow. Paramics1.5 interfaces to standard macroscopic data formats, as well as to individual vehicle counts from induction loops and optical sensors data. Paramics1.5 reportedly excels in modeling highly congested networks and ITS infrastructures.

The motivating key features of Paramics1.5 are:

1. High performance microscopic simulation,
2. ITS-capable, featuring integrated simulation of ITS including a variety of traffic management, information and control strategies,
3. Parallel, scalable and portable, making the limits of network size non-existent,
4. Capable of incorporating driver and vehicle performance parameters,
5. Capable of modeling vehicle emissions, public transport, and car parking,
6. Capable of interfacing with real-world inputs,
7. Batch mode operations for statistical studies, and
8. Equipped with advanced visualization tools that adds a complementary subjective traffic behavior evaluation capability under different ITS scenarios.

Paramics1.5 has been designed as a scalable, portable product. The tested version is UNIX based, but a PC Windows version has been recently released. The software uses the OpenGL library for graphics primitives and Motif and X libraries for windowing under UNIX.

At the fundamental level of the Paramics1.5 simulator, a number of functions are used to control vehicle-following behavior and lane-changing behavior. These functions operate on data structures that describe individual vehicles, and take, as input, an array of vehicles that describe the environment around a particular vehicle. A vehicle's data structure contains information that describes the physical characteristics of the vehicle, the behavior parameters of the driver, the vehicle's current position and dynamics, its destination and transit time. Calibration efforts in this research concentrate on driver behavior parameters.

OBJECTIVES AND SCOPE OF THE PROJECT

This project is intended to subjectively evaluate Paramics1.5 on the basis of its capabilities as an ITS tool, as well as to calibrate and test the simulator in order to objectively evaluate its underlying models and to adjust the necessary parameters to reflect local traffic behavior, as opposed to European traffic behavior. The outcome from the project is a calibrated and tested Paramics1.5 for the Irvine sub-network of the Testbed. It is also intended to serve as a basis for a variety of possible future ITS-simulation projects, and will assist in the incorporation of the general Paramics1.5 software both as the framework for the Testbed Workbench ATMS prototype, TRICEPS, and for the visualization of actual traffic conditions within the Testbed. The calibration effort involves the tuning of existing driver and vehicle models and parameters in Paramics1.5 to reproduce local behavior. These models are a small but critical part of any microscopic simulator, including Paramics1.5, which is currently well developed to reflect UK driver and vehicle characteristics.

DATA REQUIREMENTS

Calibration and validation of Paramics1.5 needs:

1. Specification of the network geometry and traffic zones,
2. Description of the infrastructure, including signal locations and settings and loop locations,
3. Specification of demand in the form Origin-Destination (O/D) matrices.
4. Actual field volume and occupancy values from loop detectors all over the network.

Therefore:

- The geometry of the Irvine network has been acquired and coded in Paramics1.5.
- Description of the infrastructure has obtained from the city of Irvine and Caltrans D12, and complemented by actual field visits by the research team.
- The O/D data was obtained from another ongoing UCI testbed project (McNally 1998), the output of which is dynamic O/D data (15 minutes resolution) for the Irvine network. The

procedure used included inputting PM peak O/D data from 1995, obtained form the Irvine Transportation Analysis Model (ITAM) iteratively into Contram and Comset traffic models. Contram assigns the O/D demand onto the network and outputs the number of vehicles (herein referred to as packets) using specific routes. Output from Contram serves as input into Comset, along with observed network link count information, to provide updated 15-minute O/D data for the PM peak. This updated data serves as input to Contram once again, and the cycle is repeated till convergence is attained. The final output is (1) 15-minute O/D data, and (2) packet information with both departure times and identified routes.

- Field data employed in the tests were collected via the Testbed, both for calibration and validation purposes.

SUBJECTIVE EVALUATION

Before embarking on any subjective or objective evaluation of Paramics1.5, it was necessary to envision and define a template of requirements for the ‘ideal’ simulator. Once defined, the template was then used for subsequent evaluation as described next.

THE ROLE OF SIMULATION IN DYNAMIC TRANSPORTATION MANAGEMENT

The prime objective of using traffic simulators within an ITS context is for dynamic transport management. More specifically, simulators can play two distinct roles: [1] as an off-line evaluation/design tool and [2] as an on-line control/guidance tool. Both roles cover numerous ATMIS applications such as provision of traveler information and route guidance, a wide variety of surface street and freeway adaptive control (adaptive signal control, adaptive ramp metering, lane use control ..etc.), incident detection and management, automated toll collection, assessment of environmental impact of transport design and management, to name a few. The off-line role is fairly easier, as real-time operation is not as pressing a need as it is the case for the on-line role. If a simulator is fast enough, however, it could be used for both functions off-line as well as on-line. Since both off-line and on-line functions are inherent objectives of any Traffic Management Center (TMC) and of the UCI-Testbed in our particular case, the efficiency and scalability of the simulator is regarded as a key factor, but not at the expense of losing any detail.

Transportation networks are, by default, dynamic systems that exhibit continuous changes in both the supply/performance and the demand conceptual sides. Unexpected events, incidents for example, inevitably occur and therefore change the supply side of the network. Any intervention by the TMC in charge, possibly in form of updating control measures, further change the supply side. Such changes motivate drivers to change their behavior in several ways, including en-route and/or pre-trip route re-choice, within the day, or from day to day. Similar dynamics take place regarding drivers’ choice of departure times in response to the dynamically changing supply conditions. The collective user behavior and response in this fashion give rise to dynamic demand profiles. Therefore, for any simulator to prove useful for dynamic transport management, it should be capable of:

1. Capturing the dynamics of supply, in terms of the detailed configuration of the transportation network and its performance in response to demands and TMC control functions implementation.
2. Capturing the dynamics of demand, in terms of dynamic user behavior in response to observed supply, either directly or via traveler information systems.
3. Capturing the complex dynamic interaction between supply and demand.
4. Performing faster than real-time to allow for proactive (based on predicted conditions) rather reactive (based on observed conditions) dynamic transport management.

Based on the above condensed conceptualization of dynamic transportation network management, we proceed to define a detailed list of requirements for a simulator to be successful for Advanced Traffic Management and Information Systems (ATMIS) applications.

EVALUATION TEMPLATE AND ITS APPLICATION TO PARAMICS1.5

In the following list, the requirements to be fulfilled by one possible ideal simulator are divided into four categories: [1] supply/control related, [2] demand/behavior related, [3] environmental-aspects related, and [4] simulator-performance related. This categorization is based on judgement in an attempt to follow the natural categorization of transportation networks management issues into supply and demand, and adding a third dimension related to the effect of transportation on the environment, and a fourth dimension related to the performance and user friendliness of the simulator software itself. Components of the following list could be regarded as items that should be explicitly modeled/fulfilled by the simulator to the user's satisfaction, given a realistic application. *It is very important to emphasize that the role of a successful simulator as we discuss here is not restricted to the mere replication of reality in terms of counts, speeds and vehicle movements that match observations. Rather we envision a broader scope and define a simulation-based dynamic traffic modeling and management tool that can replicate observed conditions, forecast future conditions, respond to policy changes, and help design control and guidance strategies as well. i.e. a comprehensive simulation based traffic management laboratory.* For the sake of space, and to avoid being repetitive, each requirement is mentioned and immediately applied to Paramics1.5. Each requirement in the evaluation list could be thought of as an objective and we discuss how close Paramics1.5 is to meeting that objective, inevitably getting into some technical details, limited by the proprietary nature of Paramics1.5.

1. Supply/control related:

1.1. **Capability to model detailed integrated networks:**

1.1.1. **Both freeways and surface streets, and seamless interaction between the two**: roadway network representation in Paramics1.5 is very detailed and comprehensive, covering the entire range of roadway categories, from minor roads, to arterial roads, to freeways. The user can define any number of categories as necessary, each with associated attributes such as the number of lanes, lane width, speed limit, type (urban street, or freeway), grade, cost factors to inflate or deflate the actual cost of travel on any particular road category, headway factors to increase or decrease the target headway on chosen links, tolls if applicable, together with a set of geometry related parameters. Links as well as lanes can been dedicated or barred to a certain class of vehicles (e.g., bus lanes, HOV lanes ..etc). The network editor and the level of user's control over details are commendable. Freeways and surface streets interaction is seamless. Roundabouts (common in Europe) are explicitly modeled. A postscript or GIS map could also be loaded and displayed in the background, over which nodes and links can be placed. This significantly reduces the burden of network coding and editing as well as enhances the match between the resulting model and the real network.

1.1.2. **Transit networks, route, and bus stops:** bus routes, service frequency, bus capacities, the exact locations of bus stops and terminals on links, are all explicitly modeled. Exclusive lanes could be dedicated to buses.

1.2. **Covering different types of Surveillance:**

1.2.1. **Point devices:** users can specify points on the network where statistics can be gathered. This is an emulation of point detector, ranging from conventional loops to more recent types of detectors, such as ultrasonic, microwave, laser, etc. Collectable statistics include: flow, gaps, occupancy and lane changing.

1.2.2. **Point-to-point measurements:** although not explicitly named 'probes', individual vehicles, up to three in version 1.5, could be traced through the network, and statistics gathered such as headway, speed, position, and acceleration. This sounds sufficient to emulate probe vehicles in a network equipped with continuous two-way communication capabilities such as, for instance, GPS-based tracking devices. However, no explicit modeling of vehicle-to-roadside beacons communication is provided. This could hinder some specific study types. For instance, optimizing the locations of road-side communication beacons in a network. The number of traceable vehicles is also very limited. However, larger numbers of different types of vehicles could be traced through the Application Programming Interface, which will be detailed later.

1.2.3. **Area devices, such as video cameras:** not explicitly modeled. This could also hinder some very specific study types, such as optimizing the allocation of video surveillance in a network, and the development of image processing based detection algorithms. This is particularly feasible in the 3-D version of Paramics1.5. Paramics-3D could be later extended to provide a camera view of the road, in which image processing algorithms could detect and track individual vehicles until they disappear from sight.

1.3. **Covering different types of control devices:**

1.3.1. **Traffic signals:** Paramics1.5 has a comprehensive Graphical User Interface (GUI) for coding pre-timed signals. Through the GUI, users input the details of cycle lengths, phases, and prioritize movements through each junction. Offsets could also be defined for coordination purposes. Paramics1.5 uses a unique way for modeling actuated signals that is based on associating signal heads with loop detectors. Each group of loop detectors at the intersection-leg is associated with a phase. A phase could be a set of actions that form a 'plan'. A signal plan is a set of conditional statements that vary the timings of the signals associated with that plan. The user defines any number of plans using a set of IF-THEN-ELSE statements which control the cycle length, phase length, gap length ..etc. Although this low-level approach of modeling signals is general and flexible, it puts the burden of designing the control logic in the user's hands. In other words, this is similar to providing a scripting language that the users can utilize to design and model signalization issues. Therefore:

1.3.1.1.**Pre-timed signals:** can be easily modeled, 1.3.1.2. **Actuated signals:** can be modeled with some effort,

1.3.1.3. **Coordinated signals:** Paramics1.5 does not automatically optimize signal coordination. However, external packages could be used and the results could be transferred into Paramics1.5's offsets.

1.3.1.4. **Network-wide dynamic, adaptive signal optimization:** as noted above, the scripting IF-THEN-ELSE capabilities can be used to modify signal timings dynamically. However, no explicit optimization logic is provided in Paramics1.5.

1.3.1.5. **Pedestrian-related signalization issues:** modeled as a phase through the same GUI for pre-timed signals. Pedestrian actuated signals, however, can not be modeled, mainly because there is no way to detect the arrival of pedestrians at a pedestrian-crossing. One possible future extension could incorporate pre-defined arrival rates to randomly actuate a pedestrian signal.

1.3.1.6. **Signal preemption for high occupancy vehicles, transit buses, emergency vehicles:** this could be modeled by associating a phase with a specific loop detector. However, loop detectors in Paramics1.5 can not classify vehicle types, and therefore are unable to distinguish a bus from a car and so on. Therefore, the only way to control a signal for a specific vehicle type such as transit buses is to have a bus-dedicated lane. Loops in such a lane would be actuated only by buses, for which the right of way could be assigned.

1.3.2. **Ramp-metering:** ramp meters are handled in Paramics1.5 as a special case simplified signal, applying the same methodology described above. Users are expected to know the metering rates a priori, i.e. there is no logic provided to set or optimize the metering rates. Using offsets, ramp meters on successive ramps could be coordinated. Adaptive real-time optimization of metering rates is not provided. However, as a possible future extension, the API (detailed later) could be used to feed back the results from an external adaptive ramp metering algorithm into Paramics1.5.

1.3.2.1. **Local:** easily modeled.

1.3.2.2. **Coordinated:** can be modeled.

1.3.2.3. **Adaptive:** not modeled.

1.3.3. **Lane-use control signals:** not modeled.

1.3.4. **Variable-speed sings:** can be modeled. Different speeds can be posted on variable speed signs on different links.

1.4. **Modeling traveler information and route guidance:** The population of drivers or 'Driver/Vehicle Units' (DVU) in Paramics1.5 is divided into "informed/familiar" and "uninformed/unfamiliar" units. The percentage of each is user-controllable. The uninformed/unfamiliar drivers basically choose routes based on the perceived static cost to their destination, on familiar/major routes only. Informed/familiar drivers have access to dynamically updated costs to the destination, and use them to make turning decisions at each decision point or junction in the network. Travel costs are provided at each junction in the form of costs-to-destinations table. The source of the updated information is not explicitly identified, i.e. whether it is from in-vehicle units or otherwise. Nevertheless, the effect of information provision is modeled under the assumption that such information is made available to all informed drivers at all junctions. The rate of information updating, which is defined by the user, controls the ultimate effect on the informed/familiar sub-population of DVU. For instance, if the information update frequency is very low, this could loosely resemble the effect of commuters getting familiar with network as they develop experience with time. On the other hand, if the information update frequency is very high (every few minutes or so), the effect would resemble the case of drivers with in-vehicle information units, displaying dynamic traffic conditions. Informed/familiar drivers are assumed to be compliant and that they always follow their <u>perception</u> of the provided information to make routing decisions. Perception variability (or error) is modeled in Paramics1.5 using random perturbations of the actual costs. If cost perturbation were disabled, there would be no difference in effect between providing information and providing all-or-nothing shortest-route guidance at each and every decision point (junction). Localized traveler information can also be modeled in Paramics1.5 using Variable Message Sings (VMS). A VMS could display information such as "congestion ahead", "take next exit" and the like. Naturally, information via VMS is available to drivers who happen to drive by the sign location only. Such information is accessible to informed/familiar and uninformed/unfamiliar drivers alike, who can choose to follow or ignore the provided information, based on user-defined rules. It should be noted that in the case of frequent information update or feedback, informed/familiar drivers do not need the VMS information, as they have access to more dynamic, more specific destination costs. We will revisit these issues in more details when we discuss demand/behavior later in the chapter.

1.4.1. **Information provision (travel times/congestion information):**

1.4.1.1. **Variable Massage Signs (VMS):** the user can specify localized VMS (or beacons) as described above.

1.4.1.2. **In-Vehicle Units (IVU):** drivers equipped with IVU receive updated information all the time, therefore the Paramics1.5 informed population would adequately capture the effect of IVU, if cost feedback is used frequently.

1.4.1.3. **Highway Advisory Radio (HAR):** HAR-based information is very coarse and can not be as fine as travel times to destinations. Therefore, we consider that HAR is not modeled.

1.4.2. **Guidance provision (actual routing advice):**

1.4.2.1. **User-Optimal guidance (UO), routing users to their shortest paths:** as discussed above, all familiar/informed drivers follow their perception of the provided information. In case of frequent cost feedback and disabling perception errors or perturbations, drivers receive current and exact costs to destination, which, in effect, is all-or-nothing route guidance to the current shortest route. However, no explicit route recommendations are modeled in Paramics1.5. The implication is more serious that it might sound at first. Because of the junction-based destination-cost trees mechanism is used in Paramics1.5 instead of vehicles holding route information, explicit route guidance can not be modeled. Therefore, experimentation with different route choice mechanisms is not possible.

1.4.2.2. **System-Optimal guidance (SO), routing users to paths that minimize the total travel times in the network:** not modeled.

1.4.2.3. **Multiple user class guidance, considering a mix of SO guided users, UO guided users, informed but not guided users, and uniformed, unguided users:** not modeled.

1.5. **Modeling toll roads:** The logic used in Paramics1.5 for modeling traffic signals and ramp metering seems to be usable for modeling manual toll plazas, where every vehicle is required to stop for a few seconds to pay toll and go. Operationally, this is very similar to ramp meters. A specific link or group of links (toll road) can be restricted to a selected type of vehicles (with automated toll payment cards) to emulate automated toll operation. It doesn't seem possible, however, to model mixed toll plazas, where all vehicles stop except tagged ones. Therefore:

1.5.1. **Manual tolling:** can be modeled.

1.5.2. **Exclusive automated tolling:** can be modeled.

1.5.3. **Hybrid tolling:** not modeled.

1.6. **Incident modeling and detection:** An incident is modeled in Paramics1.5 as a stopped vehicle. The user defines the types of incidents on each link, their duration and frequency (rate). The effect of having an incident rate applied to a link is that vehicles entering the link are selected at random at the defined incident frequency, and 'given' an incident. When a vehicle has an incident, it is brought to a halt gradually in the inside lane and at a random point on the link. Other vehicles must subsequently move around it. The stalled vehicle remains in position for the specified incident duration then move again. Incident modeling in Paramics1.5 could be greatly improved if the user is given more control on the incident characteristics. For instance the user should be enabled to control four very important variables which are: [1] the exact incident location on the link, [2] the exact onset time of the incident, [3] the number of blocked lanes and the exact affected lane(s), and [4] incident duration. These variables do not seem to be controllable in Paramics1.5. Moreover, the effect of shoulder incidents in not modeled at all, which is known to slightly affect the main lanes possibly due to rubbernecking or extra driver cautiousness in the vicinity of the shoulder incident. Inclusion of a rubber-necking emulation mechanism in future versions of Paramics1.5 is regarded important. No incident detection logic is embedded in Paramics1.5. Therefore:

1.6.1. **Capacity reduction due to lane closures by mainline incidents:** modeled with limitations (described above).

1.6.2. **Capacity reduction due to shoulder incidents:** not modeled.

1.6.3. **Using incident detection modules:** not modeled.

1.7. **Modeling parking:**

1.7.1. **Curb parking:** effect of on-street curb parking is not modeled.

1.7.2. **Car Parks (lots and garages):** the user can define car park areas within the network. Car park modeling in Paramics1.5 is very detailed, with indicators of initial number of vehicles at the beginning of the run, how full the car park is during simulation, and capacity. Currently the 'capacity' variable has no effect, as the simulator allows the car park usage to exceed capacity. The user can bar some vehicle types from using the car park.

1.8. **Modeling weather effects on supply (road capacities):** not explicitly modeled. Nevertheless, driving speeds are lowered in proportion to rain and fog to simulate adverse weather effects. This will however affect the entire network, and can not be applied to selected parts of the network (such foggy or icy bridge for instance).

2. Demand/behavior related:

2.1. **Capable of using time-dependent Origin-Destination (O/D) profiles:** Paramics1.5 takes as input any number of O/D matrices, typically one for every period of the day. For instance, nighttime, morning peak, inter-peak, and afternoon peak. Each O/D matrix specifies the total demand between O/D pairs in the entire period. The user also specifies a profile for each O/D, which is the distribution of the total period's demand divided into 5 minutes intervals. The time-dependant five-minute dynamic demands are then used to release vehicles onto the network accordingly.

2.2. **Modeling transit demand:** passenger arrival frequency at each stop in passenger per minute and alighting information at each stop (number of passengers getting off at each stop) are taken as inputs.

2.3. **Dynamic O/D prediction:** not modeled.

2.4. **Comprehensive vehicle population composition:** any number of vehicle types can be modeled in Paramics1.5, and the current model uses 16 types. Each vehicle type has a number of physical characteristics, including length, width, rates of acceleration and braking, and maximum speed. The top speed that a vehicle can attain under free flowing traffic condition is function of the maximum speed for the particular type and driver behavior. Some vehicle types are used to denote purpose rather than physical characteristics. Paramics1.5 documentation however does not have adequate details on the characteristics of each vehicle type. Access to traveler information is a characteristic of the Driver/Vehicle Unit (DVU), which can be informed, i.e. having access to continuous updates of destination costs, or not informed, having access only to free-flow destination costs, or typical average destination costs, but not dynamically updated. Therefore:

2.4.1. **Types:** passenger cars, vans, trucks, buses, and fixed route transit vehicles are modeled in detail.

2.4.2. **Performance:** the modeled types cover a wide range of performance.

2.4.3. **Access to information/guidance:** cost-to-destinations information is modeled in a binary format, i.e. DVU can be either dynamically informed at all times everywhere on the network at a user-controlled update frequency, or not informed at all having access to static information only. More macroscopic types of information, such as link closures, or delays is modeled via Variable Message Signs (VMS), which can be located anywhere in the network and is accessible to all passing-by drivers who can selectively follow or ignore the information. This could be useful to ATIS studies that focus of on a number of spatial distribution scenarios of information means such as VMS and their effect on routing and network conditions

2.5. **Modeling driver's 'travel' choice behavior:** in general, travelers are faced with a choice set that include destination, mode, departure/arrival time, route and their combinations. Commuter work-related trips usually have the destination and mode predetermined, and travelers dynamically choose departure/arrival time windows and routes, based of traffic conditions. Such choice dynamics and consequently traffic dynamics can vary in scope as either 'within-day dynamic' or 'day-to-day dynamic'. Further simplification is usually made by fixing the departure/arrival window and limiting the choice to route choice only. Within-day route choice dynamics in response to information can further be classified as either 'pre-trip' or 'en-route'. Paramics1.5 route choice modeling is focused on within-day, en-route route choice dynamics only. As discussed earlier, the population of drivers or driver/vehicle units in Paramics1.5 is divided into "informed/familiar" and "uninformed/unfamiliar" units. The percentage of each is user controllable. The uninformed/unfamiliar drivers basically choose routes based on the perceived static cost to their destination, on familiar (major) routes only. Informed/familiar drivers have access to dynamically updated costs to the destination that reflect the time varying congestion patterns, and use them to make turning decisions at each decision point or junction in the network. Travel costs are provided at each junction in the form of cost-to-destinations table that gets updated on a regular basis, perhaps every 5 minutes. The informed population is assumed to be compliant in such a way that drivers always follow their perception of the provided information to make routing decisions. Perception error or variation in perceiving true travel costs is introduced by adding random perturbation (noise) to the true cost, distributed across the driver population. Unfortunately, the assumption that drivers update and follow their routing choice at every junction is known as being "myopic", and can easily lead to over-reaction and oscillations between alternative routes. Better route choice behavior is definitely needed. For instance, a better approach is known as the "boundedly rational behavior" (Mahmassani, 1998). The essence of such behavior can be summarized as drivers change routes only if the expected benefits exceed a threshold (typically, about 20% of the remaining trip time, or one minute as a minimum). The implementation of such behavior or any other non-myopic behavior requires explicit identification of available alternative routes, their costs, and remaining trip time etc., for every DVU. This capability is not currently available in Paramics1.5. Actually, the original motivation behind using cost-to-destinations tables is to avoid the memory intensive vehicle paths enumeration. Over-reaction is alleviated in Paramics1.5 by using stochastic route choice via using the 'cost perturbation'/'perception error' as mentioned previously. However, the underlying route choice behavior is still myopic. Therefore:

2.5.1. **Plausible within-day pre-trip route/departure time choice modeling:** not modeled.

2.5.2. **Plausible within-day en-route choice modeling:** the used routing logic based on cost-tables is very efficient, but myopic. Path-based route choice needs to be incorporated in Paramics for better flexibility and realism. It is understood however that this will have adverse effects on simulation speed and efficiency. The very least it could be provided as an option.

2.5.3. **Plausible day-to-day dynamic route/departure time choice update:** not modeled.

2.5.4. **Compliance to control and guidance:** all familiar/informed drivers receive cost feedback and are assumed to be compliant. Only information from VMS can be used or ignored by DVU based on user-defined rules.

2.6. **Modeling driver's 'driving' Behavior:** driving behavior, in terms of car following, lane changing, gap acceptance, awareness and aggressiveness, is the heart of any microscopic simulator. Such models in Paramics1.5 matured over a period of 5 years of development with both accurate objective behavior and realistic visual behavior in mind. Only subjective description is presented here. Actual model details can be found in Paramics1.5 documentation. More details will be presented later as needed for calibration and validation purposes. In terms of car following, each Driver-Vehicle Unit (DVU) has a target headway that varies around a mean value depending on other factors such as weather, highway type, vehicle type, driver's aggressiveness and awareness. High aggressiveness, for instance, would cause drivers to adopt smaller headways. Similarly, high awareness value would cause drivers to use longer headways near lane drops as a sort of yielding to merging traffic. On the other hand, if a merging DVU is aggressive, it accepts a smaller gap. DVUs either accelerate, cruise, or brake to maintain the target headway stimulated by the perceived relative speed, acceleration, and brake lights of the vehicle ahead. Perception-reaction lag is taken into account. Under light traffic conditions, DVUs flow unconstrained by other vehicles, limited by lane-specific mean target speed. Lane usage in Paramics1.5 is affected by the vehicle's position relative to its target range of lanes, the latter being consistent with upcoming routing decisions, and the overtaking interactions between nearby vehicles. The target range of lanes is also affected by the DVU aggressiveness and awareness. A higher level of aggression causes a DVU to use the outer high-speed lanes, and a higher level of awareness causes a DVU to adopt the target lane for an impending turn sooner. Overtaking is controlled by varying two tables of thresholds. The lane-awareness threshold table specifies when a vehicle should move out to let a vehicle behind pass. If a DVU's awareness is greater than or equal to the threshold applicable to its current lane, it will move out. The lane-aggressiveness threshold is similar; if a DVU's aggressiveness is greater than the threshold applicable to its lane, it will attempt to overtake. DVU's propensity to change lanes can be controlled by varying the length of time that a vehicle must receive a positive stimulus, i.e. a suitable gap must exist in the target lane for n seconds in order for the vehicle to change lanes. To dampen lane-changing oscillations, a waiting period is specified. A vehicle receiving a negative stimulus must wait for the 'wait on failure' period before attempting to change lanes again. A vehicle successfully changing lanes must wait for the 'wait on success' period before attempting another lane-changing maneuver. Lane changing is therefore affected by both the availability and stability of acceptable gaps in the target lane as well as the history of lane changes of the DVU itself. Gap acceptance itself is function of DVU aggressiveness and awareness. The user defines drivers' aggressiveness and awareness by selecting a distribution for each, across the driver population (normal distribution for instance). Therefore:

2.6.1. **Car following:** modeled in details.

2.6.2. **Lane changing:** modeled in details.

2.6.3. **Gap Acceptance:** static gap acceptance behavior is modeled in details. However, gap acceptance behavior is assumed constant with time. No attempt is made to model driver's increasing impatience with time that could cause some drivers to accept smaller gaps they might have previously rejected.

2.6.4. **Merging, nosing, and yielding modeling:** embedded into car following, gap acceptance and lane changing behavior.

2.6.5. **Aggressiveness and awareness:** modeled with great flexibility.

2.7. **Modeling aggregate traffic behavior:** microscopic simulators in general produce more realistic traffic behavior, in terms of congestion propagation and dissipation, as opposed to mesoscopic, or macroscopic simulators. This is because car following and lane changing are explicitly modeled at the level of individual vehicles. Therefore:

2.7.1. **Queue formation and dissipation relative to reality:** is better captured by micro-simulators in general.

2.7.2. **Shockwave formation and propagation** is also better captured by micro-simulators in general.

2.7.3. **Modeling effect of weather on demand/behavior (effect on both demand for travel and driver behavior):** effect of adverse weather conditions on behavior is modeled in Paramics1.5 in a simplified manner. More rain and or fog reduce speeds proportionally. However, no effect on demand is modeled, as demand is exogenous to Paramics1.5 (taken as input). Weather effects, therefore, could be crudely captured through time varying input O/D demand matrices.

2.8. **Modeling congestion pricing effect on demand:** congestion pricing is a supply parameter that directly influences demand and routing behavior. It is modeled explicitly in Paramics1.5 in the link cost functions. It directly affects route choice behavior, which is based on travel costs. Longer-term effect on demand itself is not modeled.

3. Environmental Aspects:

Paramics1.5 currently uses simple look-up tables of exhaust pollution, noise pollution, and fuel consumption levels as function of vehicle type, speed and acceleration. For instance, the user should provide the pollution data in these tables. Individual vehicle pollution levels are then aggregated to produce link-level pollution. Optional graphical display of pollution levels is also available, in the form a colored band that grows perpendicular to the side of the link. The width of the band indicates the level of pollution for that link. Individual vehicle pollution can also be visualized in the form of a plume of exhaust behind the vehicle, the shade of which indicates the vehicle-specific level of pollution. The procedure is flexible in the sense that the simulator provides individual vehicle speeds and accelerations every time steps, and it is up the user to compute pollution, noise, or fuel consumption accordingly. Several alternative models could be used this way.

3.1. **Modeling exhaust pollution:** data or models can be easily interfaced.

3.2. **Modeling fuel consumption:** data or models can be easily interfaced.

3.3. **Modeling noise pollution:**

3.3.1. **Effect of traffic levels on noise levels:** data or models can be easily interfaced.

3.3.2. **Modeling traffic calming measures:** any measures that would affect speeds and accelerations of individual models can be tested.

4. Simulator performance:

4.1. **Possessing reasonable scalability and high software performance:** The Paramics1.5 software is portable and scalable, allowing a unified approach to traffic modeling across the whole spectrum of network sizes, from single junctions up to national networks. The Paramics1.5 team capitalized on their expertise and skills in parallel computing and high performance software engineering and emphasized high software performance and scalability. Paramics1.5 has two complementary modes of parallelism, and has been designed as high-performance software from the ground up. This enables the real-time simulation of hundreds, thousands or millions of vehicles, with no loss of detail. More vehicles require more processors, and more runs require more processors. This employs a message passing / distributed memory model using the MPI library for communication calls. However Paramics1.5's scalability enables development to start off small, with no risk of hitting a performance ceiling as models grow. The number of vehicles simulated is the most significant factor in performance. The number of links and nodes have a bearing too, but to a much lesser extent. Greater speed could also be achieved if vehicle visualization is turned off, or if the simulator is used in a batch mode. *An important warning is however due; simulation speed is also function of the number of computations per second or the length of the time step employed. More computations per second will, as can be expected, reduce the simulation speed. This was also found to affect the aggregate behavior of congested traffic. The interesting finding is that the lesser the number of computations per second, the higher the possibilities of queue formation in a congested traffic stream. Less computations per second mean less frequent updates of vehicle movements, which translate into cruder behavior of vehicles following each other, higher oscillations, more frequent shockwave formation and more unrealistic queues. This phenomenon disappears asymptotically as the number of computations per second is increased. The conclusion is that number of computations per second should be kept as high as feasible, keeping in mind the natural conflict between the time step size and realism of the output. For a given fixed number of computations per second, the simulator can be made to run faster in other ways such as turning off the graphics or using multiple processors.*

4.1.1. **Capability to model practical and meaningful real-life application size:** the problem reduces to hardware and software costs. The bigger the network, the more expensive the hardware and software will be. The great advantage, however, is that there is no ceiling or limit on network size and demand levels.

4.1.2. **Performing faster than real-time to facilitate usage as a control and guidance tool, and not just an off line evaluation tool:** again, simulation speed depends on the number of processors, which is in the hands of the user. Performance is measured in terms of number of vehicles that can be simulated in real-time using a 0.5 second time-step. Single workstation performance is currently in the order of 3,500 vehicles on a 133Mhz SGI Indy. The new PC version is reported to handle 17,000 vehicles in real time using a Pentium 233 processor. For larger networks (with more vehicles) a multi-processor version is available.

4.2. **Portability across hardware and operating systems:** Both UNIX and Windows version are currently available. The software uses the OpenGL library for graphics primitives and Motif and X libraries for windowing under UNIX.

4.2.1. **Across operating systems such UNIX, Windows etc.:** Both UNIX and Windows are supported.

4.2.2. **Across hardware platforms, such as P.C.s, workstations.. etc.:** Both PCs and workstations are supported. The multiprocessor version employs a message-passing / distributed memory model using the MPI library for communication calls. The MPI library is available for a number of architectures, both SMP and MPP. Currently the Software has been tested on:

- Cray T3D (64 x Alpha)
- SGI Challenge(8 x MIPS R4400))
- Cray CS-6400(16 x Sparc)
- DEC TurboLaser(4 x Alpha)

4.3. **Having a friendly Graphical User Interface (GUI):** Paramics1.5's GUI is one of its most powerful features. Controls are in the form of menus, buttons and sliders. The Paramics1.5 visual interface was created using XDesigner, an application builder. The top-level interface window has a standard ``look and feel" familiar to most target users. Options are selected using pull-down menus, and zooming and panning within the graphics areas of the windows is done using mouse-button combination presses. A number of other sub-windows can be displayed also. In the sub-windows, parameter variation can be done using sliders, as these ensure the validity of values entered. On-line help, is just one of the popup windows that the user can access, and leave up while running the simulation. The on-line help is however similar to the supplied software manuals, in terms of excessive briefness.

4.3.1. **Comprehensive network editors, and input preparation aids:** very detailed.

4.3.2. **Displaying vehicle movements in a bird's eye view:** this is the main visualization mode.

4.3.3. **Displaying driver's view of the road (3-D and virtual reality):** a 3-D view of the network is included in versions higher than 1.5. The user can also select up to three vehicles to trace. The tracer window displays a driver's eye view of the road. This view is intended as a means of evaluating the car following and lane changing models subjectively. Also displayed are many parameters and attributes associated with the vehicle being traced.

4.3.4. **Displaying status of control devices and their relations to vehicle movements:** the status of each signal head at an intersection or a ramp meter is displayed as the simulation proceeds. Also messages on VMS can be shown.

4.3.5. **Network-wide coloring schemes for congestion levels and active incidents status:** the network can be color coded to reflect different speed or density levels in different parts of the network, which is very useful for quick identification of 'hot spots'.

4.3.6. **Network-wide indicators of pollution levels:** As previously mentioned under environmental aspects, optional graphical display of pollution levels is available, in the form a colored band that grows perpendicular to the side of the link. The width of the band indicates the level of pollution for that link. Individual vehicle pollution can also be visualized in the form of a plume of exhaust behind the vehicle, the shade of which indicates the vehicle-specific level of pollution.

4.3.7. **Easy and intuitive window-based software controls to allow changing simulation parameters and examine effects:** available for many parameters such as aggressiveness and awareness, but not available for internal models such car following and lane changing. Internal models are not designed to be directly accessible, but could be overridden using the API.

4.3.8. **Comprehensive graphical output displays (items under 4.5 below):** Paramics1.5 produces a variety of output graphs, such as speed vs. headway, density vs. flow, average speed vs. flow, lane usage vs. flow, and the percentage of drivers rerouting in response to a particular VMS. However, graphical display of detector output vs. time is surprisingly unavailable.

4.4. **Running in batch mode as well, to expedite real time, parallel, multi-scenario design and evaluation:** Paramics1.5 can be run in batch mode without visualization. The advantages of this mode are not only faster execution due to absence of computational burden for graphics, but also the ability to test multiple scenarios at the same time. This could be very useful for the design and testing of different control strategies in response to dynamics event in the network, before selecting one for deployment.

4.5. **Producing comprehensive output and generating of Measures of Effectiveness (MOEs):** Paramics1.5 provides a comprehensive array of output measurements as detailed below, as fine as one second apart. One limitation is notable however: each simulation run produces its own array of outputs, with no capability to aggregate or average outputs over multiple runs. For instance, no tools are available to simulate a given scenario 'n' times, then examine the average behavior/output.

4.5.1. **At a point (loop location):** lane-specific data can be gathered from a loop (a point) such as flow, speed, headway, and acceleration. Data collection could be limited to one or more classes of vehicles (example: trucks only, etc.). Also 20-binned distributions could be requested for data reduction.

4.5.2. **Link level:** the user can request link-aggregated information such as link flow, mean speed, density (the number of vehicles per unit length), the number of lane changing events, and lane usage in the form of the proportion of vehicles using each lane. Queue counts can also be saved. This is the number of vehicles queuing on each link for non-signal controlled links. For signalized links, the number of vehicles queuing is counted at the start of each green phase. The number of vehicles traversing each link can be gathered for every O/D pair.

4.5.3. **Junction level:** the user can collect flows at signal stop lines. Also turn counts can be collected at every junction. For every link in the network, a count is made for the number of vehicles moving from that link to any of its possible exit links.

4.5.4. **Path level:** the user can gather path-specific 'perceived' route costs. This could useful in examining routing patterns in the network. The user must pre-define paths to collect data for, in the form of a series of connected nodes for each path.

4.5.5. **Origin-Destination level:** the user can select to gather total trip times or link by link average travel times between user-specified O/D pairs. As mentioned under 'link level', the number of vehicles traversing each link can be gathered for every O/D pair. O/D release counts can also be saved which is the number of vehicles actually released for each O/D pair. Such release counts can be compared directly to demand O/D matrices.

4.5.6. **Network Level:** Paramics1.5 generates a single network-level figure to indicate delay. The figure gives the total journey time for all vehicles through out the network. No comparison to minimal possible travel times is made (such as relative to System Optimal assignment total travel time). Incident-specific delays can not be collected directly. However, link statistics can be collected for the link that has the incident. No statistics on travel time variability and predictability are available. Snap shots of the network including records of all vehicle locations can saved for later viewing, or for use as a starting point in a future run.

4.5.7. **Vehicle trajectories and time-space diagrams:** the user can trace up to three vehicles, and collect data describing the headway, position, speed and acceleration for the traced vehicles. No graphs however, are directly generated to display vehicle trajectories in time-space diagram format. Also the number of traceable vehicles (three) is very limited. It would be more useful to assign a "trace" attribute for user-chosen vehicles or vehicle classes and enable the user to trace them as well as collect corresponding statistics.

4.5.8. **Transit-related MOEs such as schedule adherence of transit buses:** users can collect average travel times that buses take to traverse a route. Bus routes of interest must be pre-specified by the user. However, no finer data that could facilitate schedule adherence evaluation is directly obtainable.

4.6. **Supporting commonly used database formats (dBase, Access, ..etc.):** Paramics1.5 uses free ASCII text files as its input specification, with any amount of white space between keywords. Comments can be included after the special token ##. Any interaction with standard database software has to be via text files, but not directly. Also text comments could pause problems.

4.7. **Having an Application Programming Interface (API):** it was proposed earlier and now under continuous development by the Paramics team is the idea of providing the software in a 'framework' version. The simulator would be used as a shell, and would provide researchers with the ability to test their own models. We regard this capability to be essential for research. Such an API should have a dual role; first to allow researchers to override the simulator's default models such as car following, lane changing, route choice models for instance, and second, to allow them to interface complementary modules to the simulator. Complementary modules could be any ITS application, such signal optimization, adaptive ramp metering, incident detection and management and so on. The current Paramics1.5's API provides a two-way functional interface with [1] control functions, and [2] callback functions. The control functions allow the user to pass external information to the simulator, controlling such things as speed of each vehicle, likelihood of changing lanes.. etc. Call back functions achieve the opposite, allowing the user to extract information from the simulator such as the attributes of a vehicle and its environment. Currently, the user can override the default car-following and lane-changing logic. The user should define control functions to specify dynamics to control both longitudinal and lateral vehicle movements. Broader functionality of the API is planned in future versions.

CALIBRATION AND VALIDATION OF KEY MODEL PARAMETERS

PRELIMINARY TESTING AND EVALUATION OF RAW MODEL

As a first step, a phase of preliminary testing of the simulator without calibration was conducted. The objectives of this phase include 1) evaluating the raw model's performance on a small freeway section prior to network-wide modeling, in order to avoid errors from demand matrices associated with large networks, 2) exploring the model's transferability for use in the US and 3) identifying critical parameters for calibration.

Field data employed in this series of tests were collected via the UCI on-line intertie to the physical Testbed. Thirty-second traffic data (volume and occupancy) were collected over two hours in a morning-peak period on the southbound I-405 freeway, California, between Culver and Jeffrey off ramps. This site is 1495 feet long, and instrumented with loop detectors as shown in Figure 1-1.

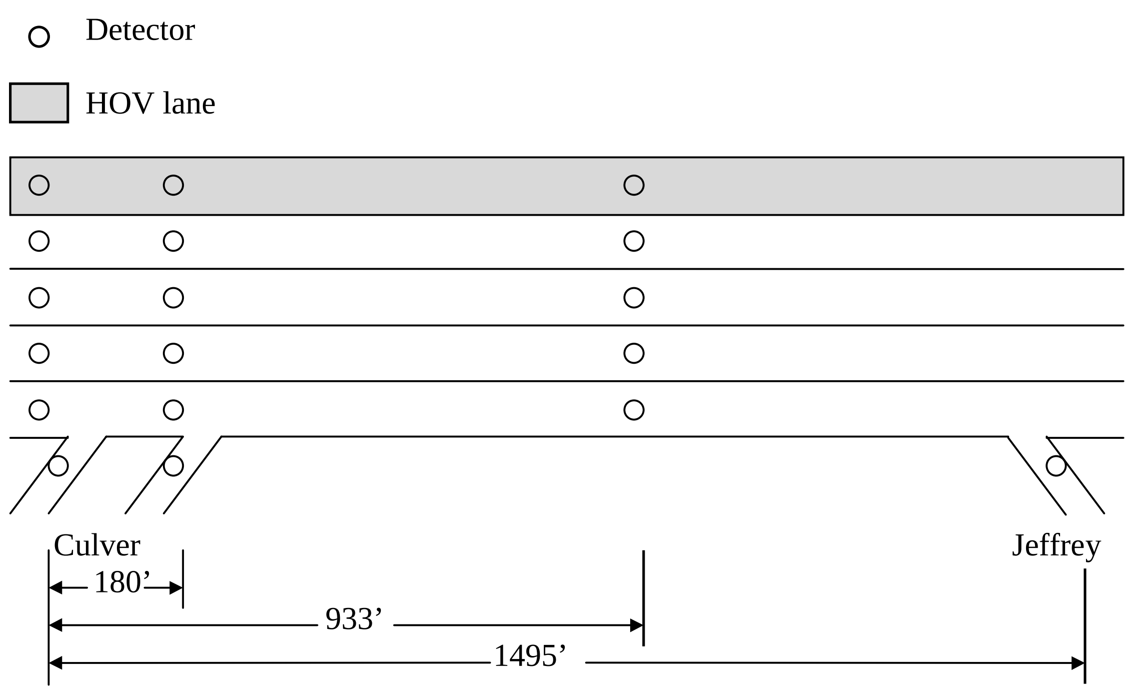

Figure 1-1. Detector Layouts at the Study Site

In Paramics1.5, two types of demand-related data files are necessary 1) *demand*, which records the Origin-Destination (O-D) traffic flows in the network and 2) *profile,* which determines the percentage of total traffic flow released onto the network in each time interval (e.g., 5-minutes). Field data were processed to generate input data files, *demands* and *profile,* as well as output data figures, which were used for comparisons to simulated outputs.

Several simulation runs were conducted in order to identify critical variables that may significantly influence the performance of Paramics. Any critical variables would be regarded as target variables in the calibration phase. In this procedure, the variables related to driver behavior (e.g., the distributions of aggressiveness and awareness) compositions of vehicles, driver's familiarity etc. were examined one-by-one. The numerical and graphical outputs from the simulation were then compared to corresponding real data.

Outputs used for comparison were classified into two groups: Origin/Destination related (total demand generated at each zone and total flow reaching the destination) and lane-specific measures (lane outflow, lane usage and lane-specific headway distribution). Default values for all the variables in files *Vehicles* and *Driver* of Paramics1.5 were kept unchanged (including vehicle population composition, car pool eligible vehicles, and driver aggressiveness and awareness distributions). Other demand data including *demands* and *profile* were formulated on the basis of field data. Thirty simulation runs were executed and outputs were aggregated and analyzed using Microsoft Excel. Outputs aggregated in this step for further comparison include 1) total generation (total vehicles released in 2 hours), 2) total attractions (total number of vehicles completing the short journey in 2 hours), 3) lane-by-lane outflow (the number of vehicles collected from each lane in 2 hours), and 4) % lane-by-lane usage.

Results in Table 1-1 indicate a shortage in the number of vehicles being released to the section, the number of vehicles traversing the section, and a severe lane usage problem. Therefore, parameter values other than the default need to be explored, as explained in the next section.

Table 1-1. Output Data Comparison based on the raw Paramics1.5

Total Demand	Paramics1.5		VDS*		Error (%)
	11906		12869		-7
Total Outflow	Paramics1.5		VDS		Error (%)
	9000		12025		-25
Lane Outflow	Lane-1	Lane-2	Lane-3	Lane-4	
Paramics1.5	1052	1960	2695	3293	
VDS	2170	2900	3330	3625	
Error (%)	-52%	-32%	-19%	-1%	
Lane Usage	Lane-1	Lane-2	Lane-3	Lane-4	
Paramics1.5	12%	22%	30%	36%	
VDS	18%	24%	28%	30%	
Error (%)	-6%	-2%	2%	6%	

* VDS: Vehicle Detection System

EXPERIMENTING WITH PARAMETER SELECTION

In the *vehicles* file, the user can specify the types of vehicles using the network and their percentages, the types and percentages of vehicles eligible for using the HOV lane. It was found that setting the percentage of HOV usage equal the observed percentage from field data does not produce acceptable results. A possible explanation is that vehicles 'eligible' for using the HOV lane do not necessarily use it. It seemed that the types of HOV-eligible vehicles and their percentages need empirical experimentation in order for the correct percentage actually using the HOV lane. However, there is no obvious and systematic way for doing that. In our case, several combinations of the types of vehicles and their HOV-eligibility were experimented with. The final combinations in the *Vehicle* file are shown in Table 1-2.

Table 1-2 The Calibrated Parameters in the file *Vehicle* of Paramics1.5

Vehicle Composition			
Type-1	car		Proportion 71
Type-2	car	HOV	Proportion 4
Type-3	car	HOV	Proportion 3
Type-4	car		Proportion 2
Type-12	lgv		Proportion 4.5
Type-13	ogv1		Proportion 7.5
Type-14	ogv2		Proportion 8.0

In addition, and to improve lane usage, parameters concerned with driver behavior were investigated. Default values for these parameters were based on research at the Transportation Research Laboratory (Jeffreys, 1994). Two parameter sets in particular, aggressiveness and awareness, were examined since these two significantly influence the behavior of drivers. The distributions of aggressiveness and awareness levels across the population of drivers on the simulation network can be varied at any time during the simulation. There are four default types of distributions associated with each of these two parameters in the file *Driver.* In this test step, numerous runs with different combinations of distributions were conducted until the outputs of lane use were improved further. The selected parameters in the file Driver are shown in Table 1-3.

Table 1-3 Aggressiveness and Awareness Parameters in the file *Driver* of Paramics1.5

aggressiveness	multiple 3	slides	1	4	11	21	28	21	11	4	1	(normal distribution)
awareness	multiple 1	slides	1	4	11	21	28	21	11	4	1	(normal distribution)

Another important insight gained from the above experimentation is that the usage of lane-1 (the outside lane) and lane-2 is affected by the shaping of the destination zone. For instance, the proportion of vehicles using lane-1 increases notably in case of the destination zone covering the off ramp of the study segment of the freeway where only one destination zone is used. To examine the effect of the zone shape, a simple experiment was conducted. A new destination zone was added separately to the off-ramp, i.e. one zone for the mainline segment and another for the off-ramp respectively. This way, the proportions of vehicles leaving the freeway and those staying on the freeway are under control. Compared to the single-zone case, the lane usage was remarkably improved.

Using the above adjustments combined, ten simulation runs were conducted and improved results summarized in table 1-4.

Table 1-4. Improved Results

Total Demand	Paramics1.5	VDS		Error (%)
	12623	12869		-1.91
Total Outflow	Paramics1.5	VDS		Error (%)
	11793	12025		-1.93
Lane Outflow	Lane-1	Lane-2	Lane-3	Lane-4
Paramics1.5	2092	2517	3399	3785
VDS	2170	2900	3330	3625
Error (%)	-3.59%	-13.2%	-2.07%	4.41%
Lane Usage	Lane-1	Lane-2	Lane-3	Lane-4
Paramics1.5	18%	22%	29%	31%
VDS	18%	24%	28%	30%
Error (%)	-0%	-2%	1%	1%

Graphically, Figures 1-2 and 1-3 present the distributions of lane-by usage generated using real VDS and simulation data respectively. Each data point in either figure represents the ratio of the lane traffic count in a given lane to the sum of lane traffic counts collected from the midstream detectors in a given 30-second time step. Compared to VDS data, the simulated data points shown in Figure 1-3 distribute more densely, mostly in the range between 5000veh/hr and 7000veh/hr. However, cluster centers match in both figures.

Figures 1-4 and 1-5 summarize the headway distributions based on real and simulation data respectively. Vehicular headway distribution is classified into eleven integer categories ranging from 1 to 11. Each of these numbers represents the center of a bin that is 1 second wide. For example, vehicles with headway values between 1.5 seconds and 2.5 seconds are regarded as the headway group "2-second." Compared to Figure 1-4, the headway distributions shown in Figure 1-5 indicate a good match except for lane-1 (the outside lane) for which the peak headway simulated using Paramics1.5 is about 4 seconds and is only 3 seconds in the field.

Figure 1-7 shows that the simulated flow-density relationship closely approximates the real one shown in Figure 1-6 indicating that Paramics1.5 properly captures the fundamental flow-density relationship. In the following section, we examine, calibrate and test the microscopic parameters behind the car following and lane changing models of the simulator in order to enhance performance further.

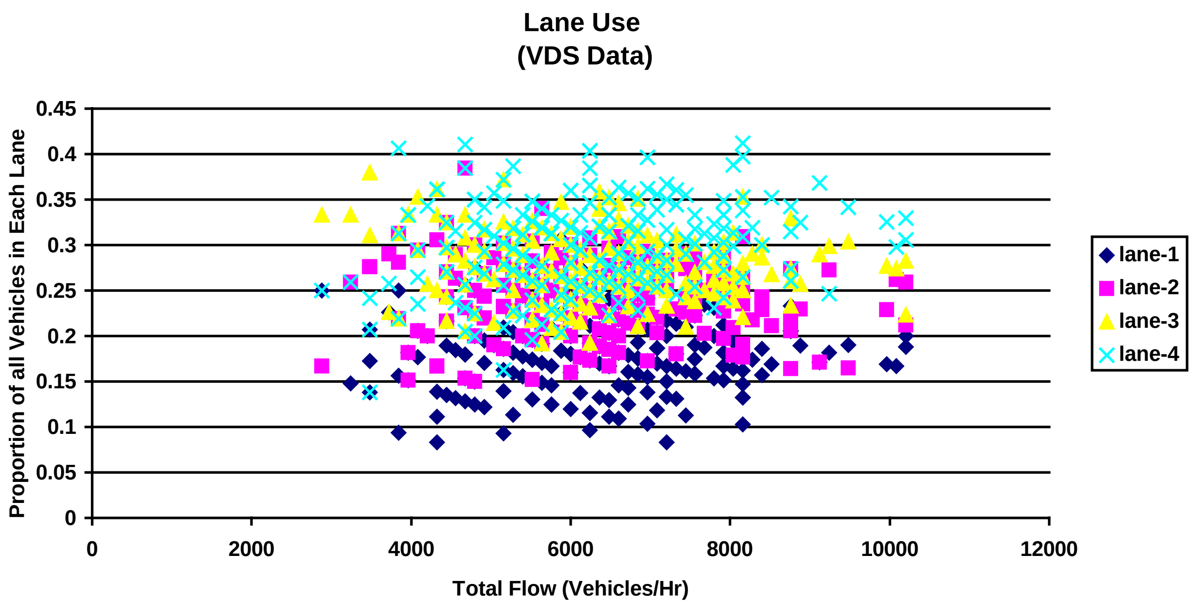

Figure 1-2. Lane Usage on a 4-Lane Freeway Segment (VDS Data)

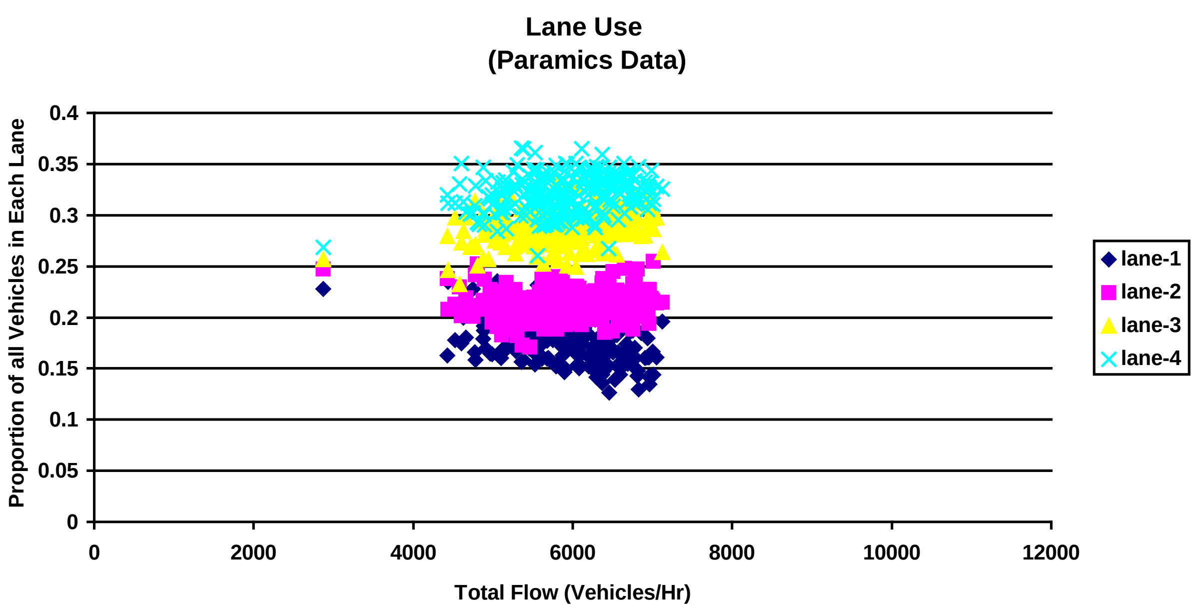

Figure 1-3 Lane Usage on a 4-Lane Freeway Segment (Paramics1.5 Data)

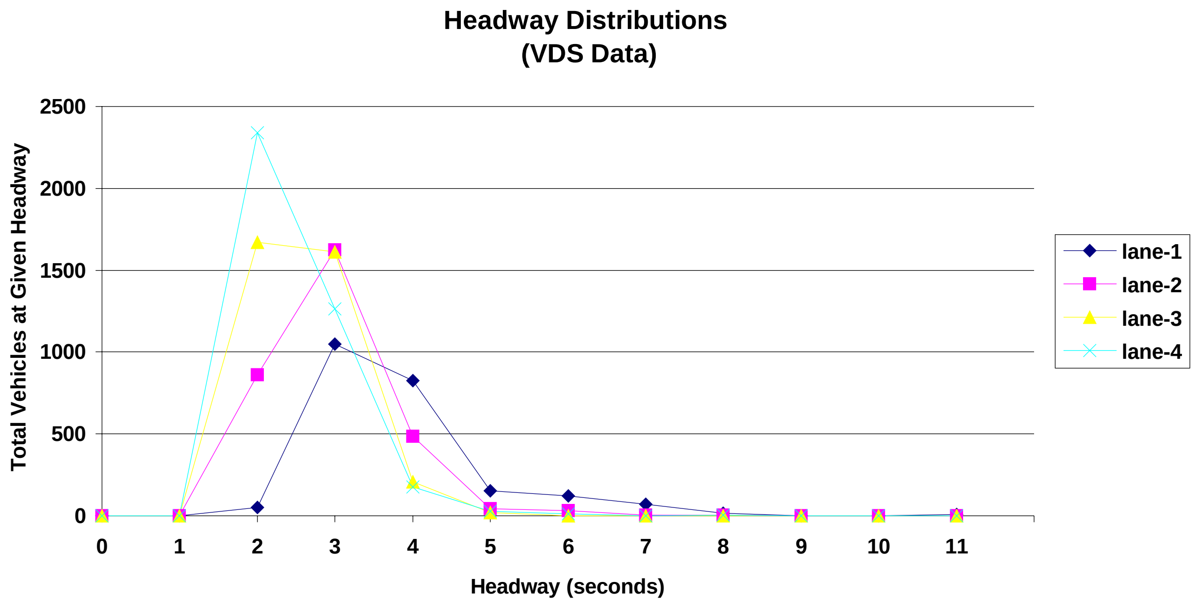

Figure 1-4. Headway Distributions on a 4-Lane Freeway Segment (VDS Data)

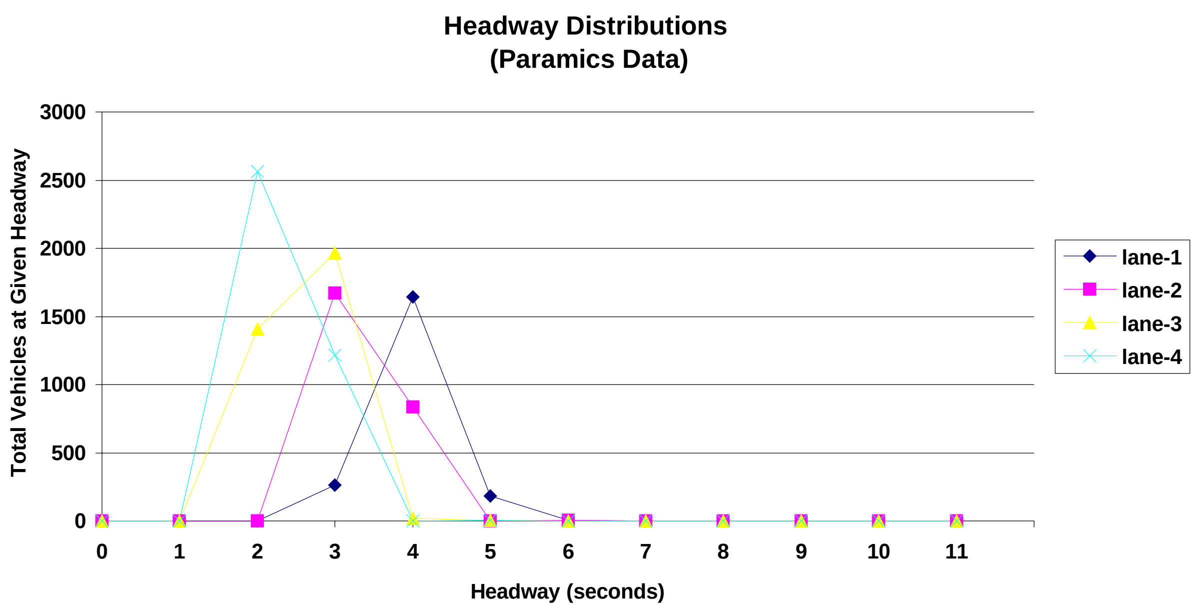

Figure 1-5 Headway Distributions on a 4-Lane Freeway Segment (Paramics1.5 Data)

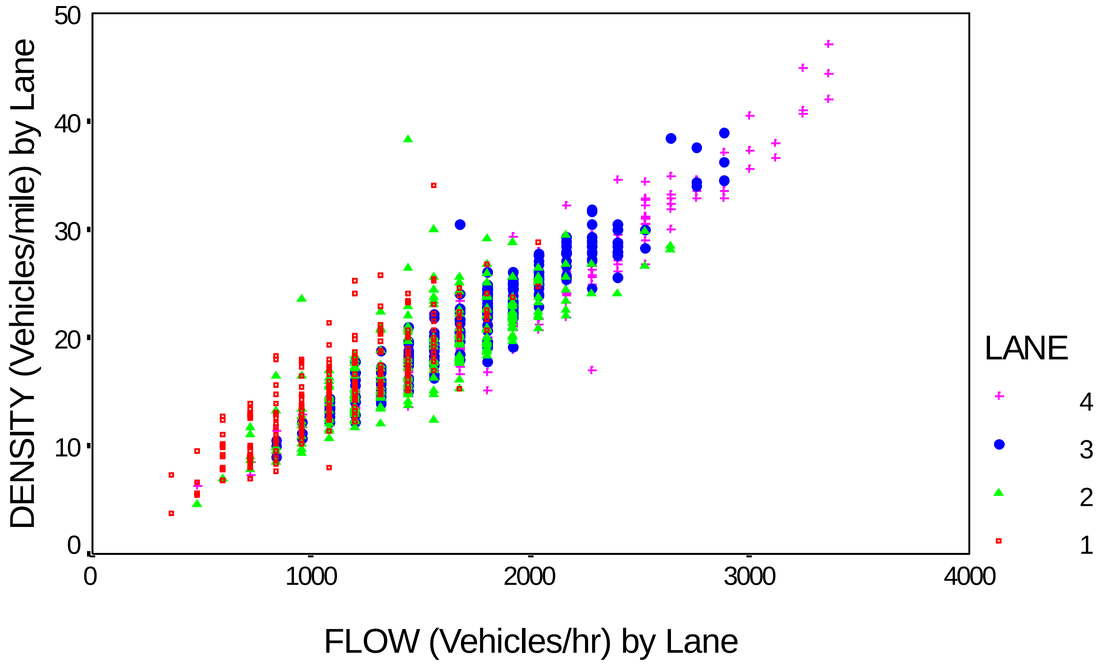

Figure 1-6. VDS Flow-Density Characteristics

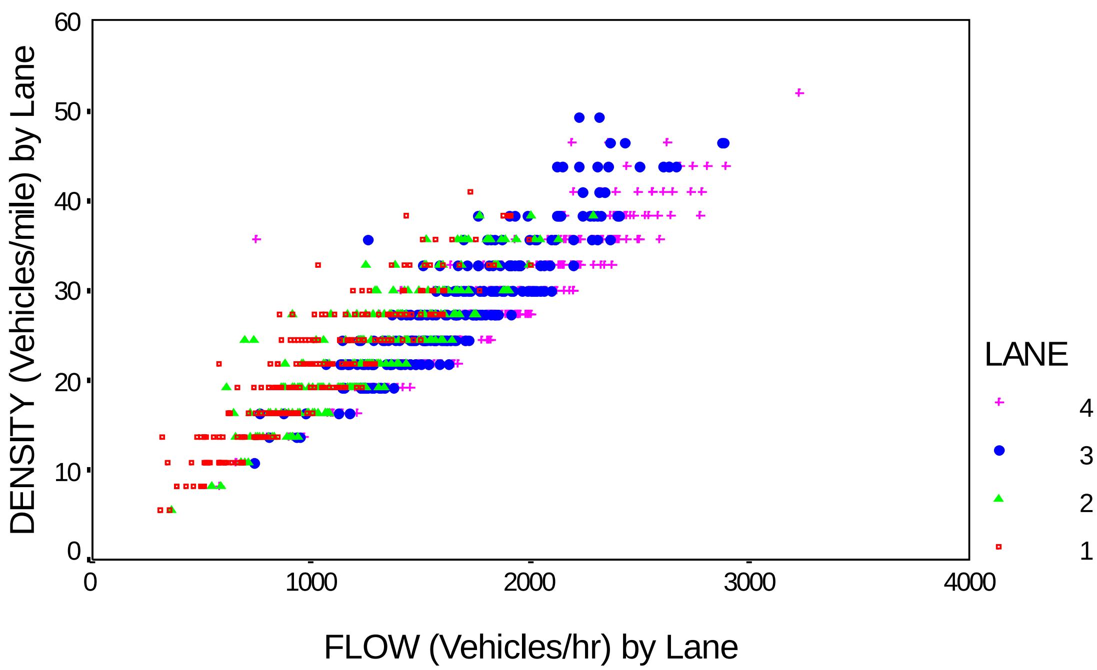

Figure 1-7. Paramics1.5 Flow-Density Characteristics

CALIBRATION OF KEY MICROSCOPIC PARAMETERS

CAR FOLLOWING AND LANE CHANGING: MEAN HEADWAY AND REACTION TIME

In this section, we examine the heart of the micro-simulator, more specifically, the car following and lane changing models. The objective is to identify key parameters for calibration and validation using California data. Other parameters already adjusted up till this stage are kept the same.

The car following and lane changing models in Paramics1.5 were developed over 5 years to show close correlation to British Data as well as visual observation using video and the 'mind's eye' judgement. These models control the behavior of each Driver Vehicle Unit (DVU). In this effort, we accept the structure of models, the details of which can be found in the software documentation. We only attempt to calibrate two key parameters: mean headway and reaction time.

Each DVU has a target headway h, the mean value of which defaults to 1 second. The target headway for each DVU varies around the mean depending on an array of other parameters including driver's aggressiveness and awareness, vehicle time, road geometry and so on. High aggression for instance will cause drivers to accept a smaller headway, and high awareness will effect the use of a longer headway when approaching a lane drop in order to allow DVUs in other lanes to merge more easily. If stimulated to change gap acceptance behavior, a DVU varies it speed to attain its target headway while following the lead DVU. In absence of a lead DVU, the target speed comes into effect. The reaction-time captures the time lag before the driver of the following vehicle comprehends the action of the lead vehicle and reacts to it. Therefore, the acceleration for the following vehicle is based on the action of the lead vehicle some time in the past. It is the use of reaction time that results in effective modeling of shock waves. Also, intelligent cruise control systems (in Automated Highway Systems) can be modeled by reducing the reaction time to near zero, causing shockwaves to disappear and causing DVUs to follow each other with a constant gap. Similar to car following, lane changing is effected –in part- by the availability of acceptable gaps, which is turn affected by the mean headway and reaction time.

The default values for the mean headway and the mean reaction time in Paramics1.5 are both set to 1 second. Early related studies (Greenshields, 1947; King et al., 1973; Kunzman, 1978), recommend headway values around 2 seconds in the fields of urban traffic control and related applications. The value suggested by the Highway Capacity Manual, 1985 (HCM) is 2 also seconds. For reaction time, numerous early experiments had been conducted to measure the range of reaction time. The range of 0.3-2.0 seconds was suggested in a study by Johansson et al. (1971) where 0.66 is the median in their study. It is noticeable, however, that most of these studies are fairly old, and they indicate ranges rather than pinpoint specific values. Therefore, derivation of local mean headway and reaction time from field traffic flow data seems appropriate, especially in the presence of abundant field data as in case of the California testbed. The focus of the calibration effort, described next, is on mean headway and reaction time of DVUs.

CALIBRATION PROCEDURE AND RESULTS

Finding appropriate values for the mean headway and reaction time is a two dimensional search process that minimizes the discrepancy between the simulation output and field observations, and that requires numerous simulation runs for every point in the search space. Since the shape the search space is unknown, we used our own empirical iterative search process that searches in one direction only at a time. One parameter, say , is fixed at its best known value, possibly from the previous search iterations, while one-directional search is performed on the other parameter, say , within reasonable upper and lower bounds. Once the optimal value is found, it gets fixed and its range empirically reduced around this value, and the search for starts. The process continues until improvements diminish. The primary steps of the calibration procedure are summarized as follows:

Step 1: Select one target parameter ( ) for calibration at the current iteration and fix the other parameter to its best known value.

Step 2: Specify the increment-size associated with the target parameter, $_i$, where the subscript "i" represents the i^{th} time of calibration on the parameter , such that $_i$ $_{i-1}$.

Step 3: Given $_i$, test the simulation model using the increasing sequence of values: L_α^i, L_α^i + $_i$, L_α^i +2 $_i$,...., U_α^i where L_α^i and U_α^i represent the lower and upper bounds of at the i^{th} time of calibration on the parameter respectively.

Step 4: Select and Update the optimal value of .

Step 5: Subjectively truncate the acceptable range of around the current optimal value.

Step 6: Repeat for . Stop if the expected model performance has been achieved, or the rate of improvement diminishes. Otherwise, go back to Step 1 and repeat.

It should be noted that this procedure is purely empirical and simple. If warranted, more sophisticated search algorithms such as Genetic Algorithms could be used.

The mean headway (H) and the mean reaction time (R) were calibrated through six iterations of the proposed procedure. The average absolute error of simulated traffic volumes was employed as an index to evaluate the system's performance. The following summarizes the results from each iteration in the calibration procedure:

Iteration	Target Parameter	Fixed Parameter		L	U	Optimal
1	H	R=0.66	H=0.25	1.00	3.00	H=1.75
2	R	H=1.75	R=0.20	0.30	1.90	R=0.50
3	H	R=0.50	H=0.10	1.25	2.25	H=1.75
4	R	H=1.75	R=0.10	0.30	1.50	R=0.4
5	H	R=0.40	H=0.01	1.65	1.85	H=1.65
6	R	H=1.65	R=0.02	0.30	0.80	R=0.42

The previously obtained mean headway and reaction time were then used to compare simulation output to actual field data. The simulated output shown in the following tables and figures is based on aggregated results from ten simulation runs. The following tables summarize the improvements from using the calibrated parameters:

Table 2-1 shows an improvement in the total demand actually generated by the simulator. In Paramics1.5, if DVUs can't find suitable gaps on the link on which they are to be generated, the DVU's queue up in the computer's memory. Improving the mean headway and reaction time might have reduced the virtual queue allowing more of the actual demand to be generated.

Table 2-1 Comparison of Total Demand

	VDS	Paramics1.5 (un-calibrated)	Paramics1.5 (calibrated)
total demand	12869	12623	12949
error (%)		-1.91	0.62

Similarly, Table 2-2 supports the same conclusion. More DVU's make it to and through the test site, improving the total flow as detected at the downstream end of the site. The lane-by-lane outflows and percentage usage also improved as shown in Table 2-3 and Table 2-4.

Table 2-2 Comparison of Total Outflow

	VDS	Paramics1.5 (un-calibrated)	Paramics1.5 (calibrated)
total demand	12025	11793	12132
error (%)		-1.93	0.89

Table 2-3 Comparison of Lane Outflow

	Lane-1	Lane-2	Lane-3	Lane-4
VDS	2170	2900	3330	3625
Paramics1.5 (un-calibrated)	2092	2517	3399	3785
Error (%)	-3.59	-13.2	2.07	4.41
Paramics1.5 (calibrated)	2160	2849	3351	3772
Error (%)	-0.46	-1.75	0.63	4.05

Table 2-4 Comparison of Lane Usage

	Lane-1	Lane-2	Lane-3	Lane-4
VDS	18%	24%	28%	30%
Paramics1.5 (un-calibrated)	18%	22%	29%	31%
error (%)	0	-2	1	1
Paramics1.5 (calibrated)	17%	24%	28%	31%
error (%)	1	0	0	1

The lane-use scatter diagrams in Figures 2-1 and 2-2 show good correspondence between simulated and real data. The simulated clusters, although smaller than the real ones, are better than the case before calibration. Figures 2-3 through 2-6 also show good correspondence between the headway distributions and flow-density relationships of the simulated output and the real data respectively.

Next we attempt to validate the performance of Paramics1.5 using new data sets, previously unseen in the calibration phase.

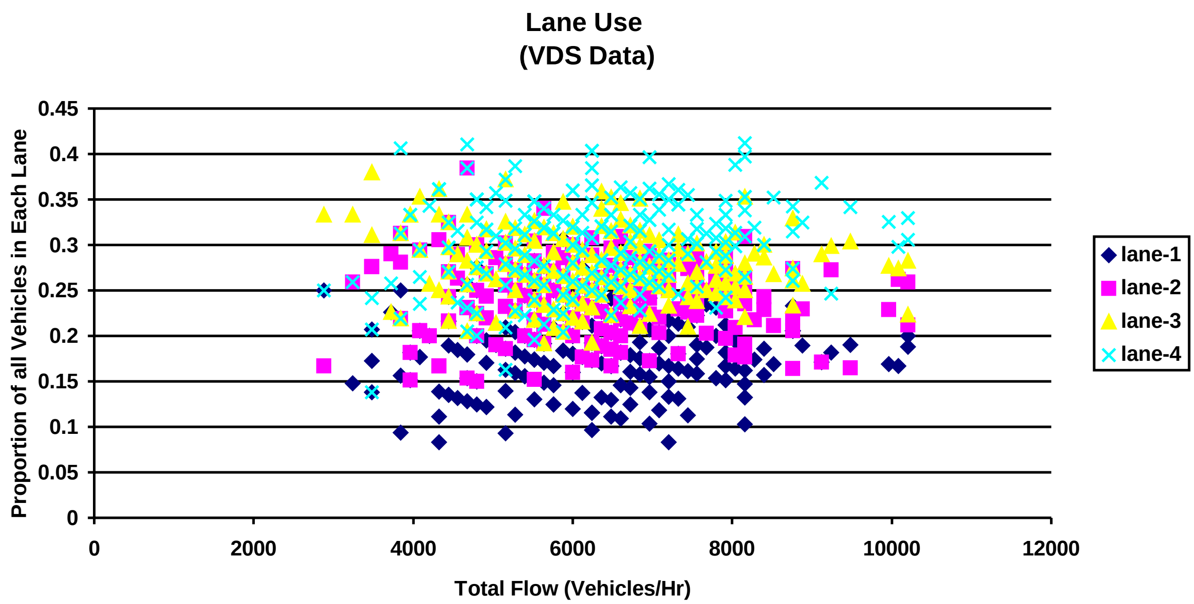

Figure 2-1 Lane Usage on a 4-Lane Freeway Segment (VDS Data)

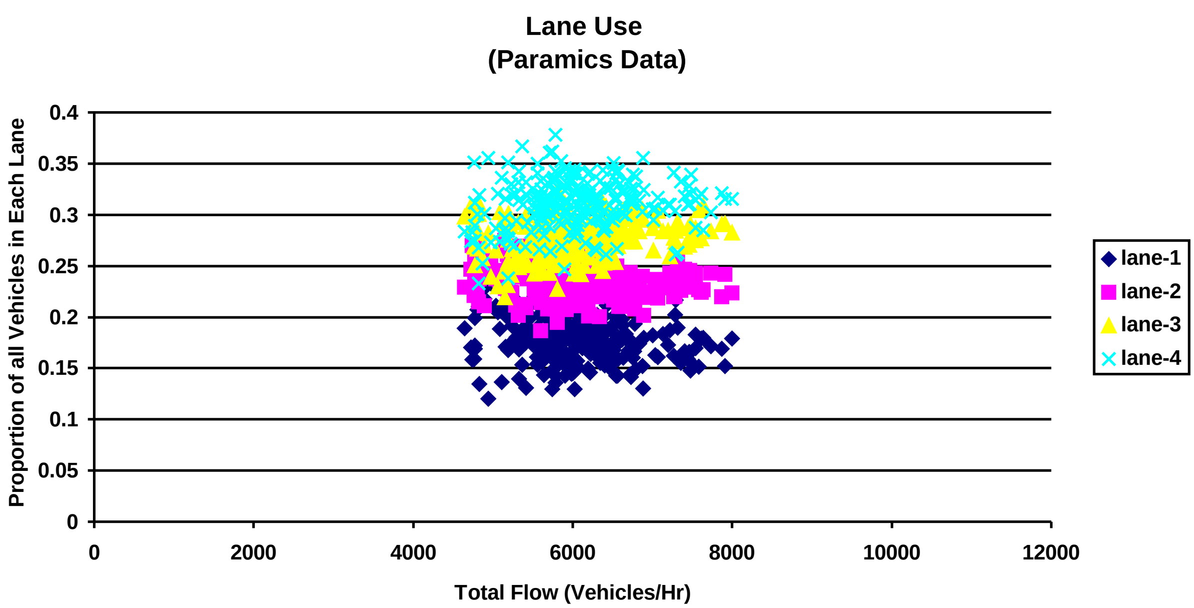

Figure 2-2 Lane Usage on a 4-Lane Freeway Segment (Paramics1.5 Data)

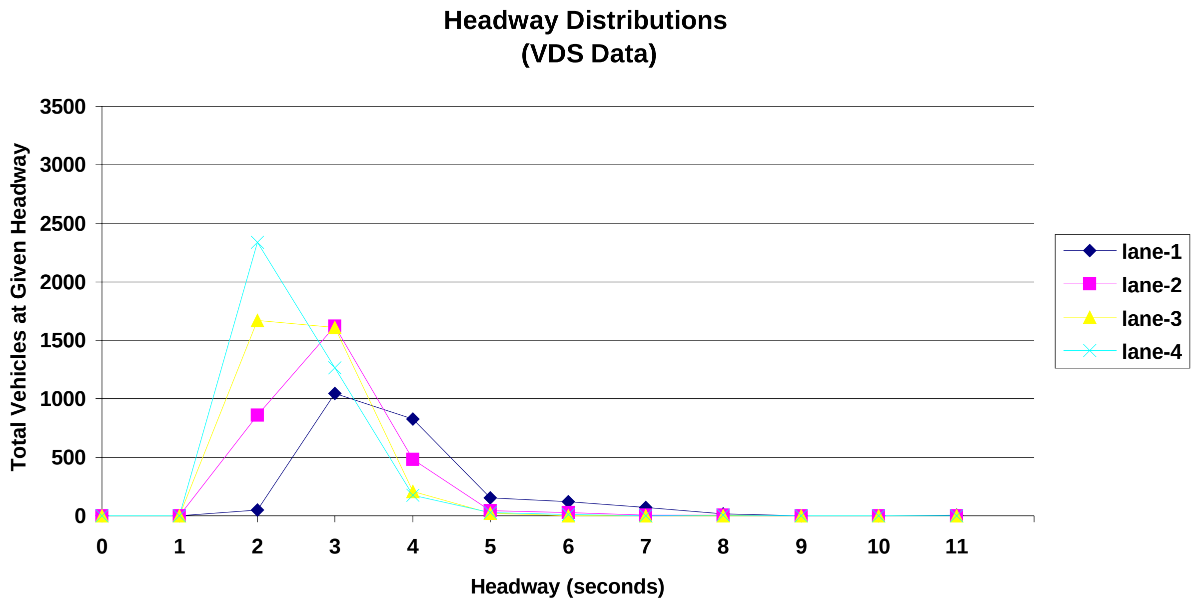

Figure 2-3 Headway Distributions on a 4-Lane Freeway Segment (VDS Data)

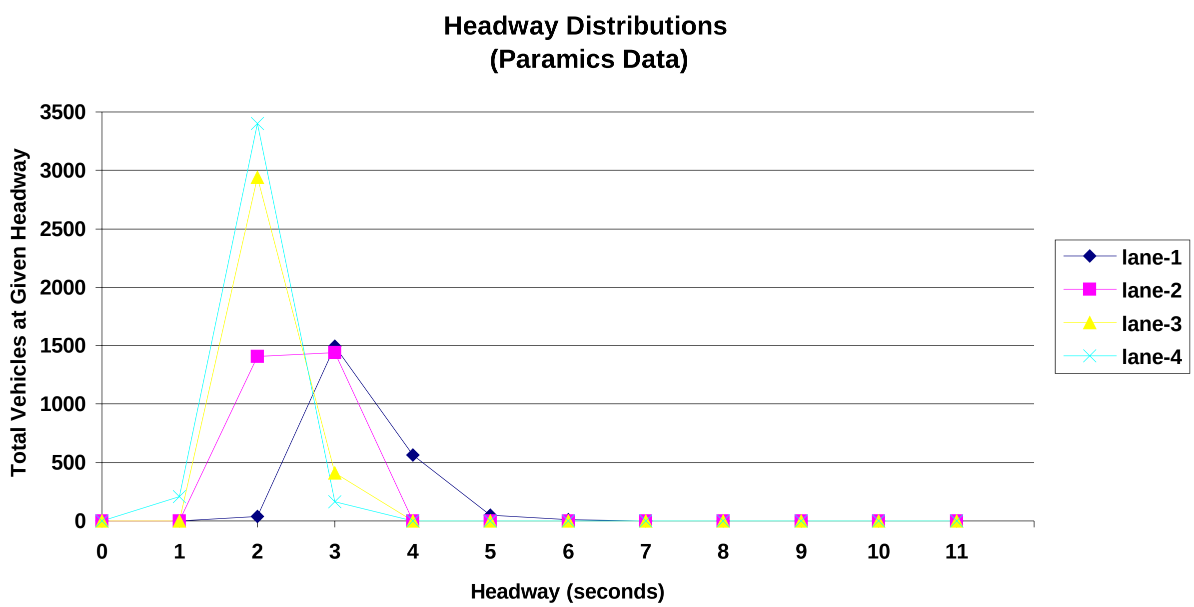

Figure 2-4 Headway Distributions on a 4-Lane Freeway Segment (Paramics1.5 Data)

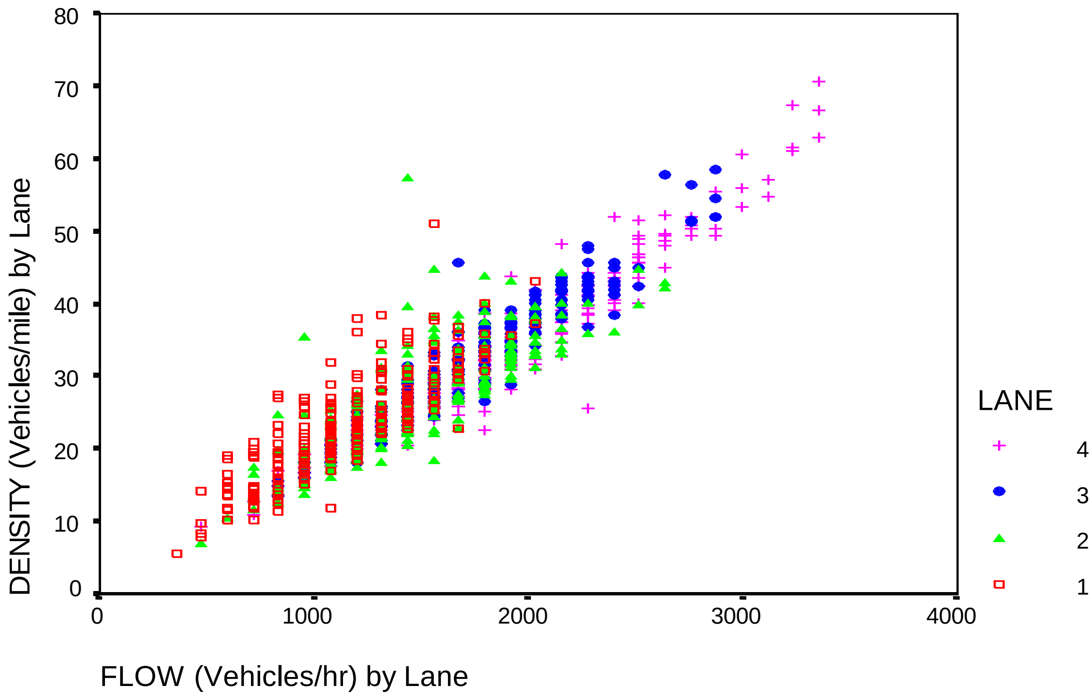

Figure 2-5 VDS Flow-Density Characteristics

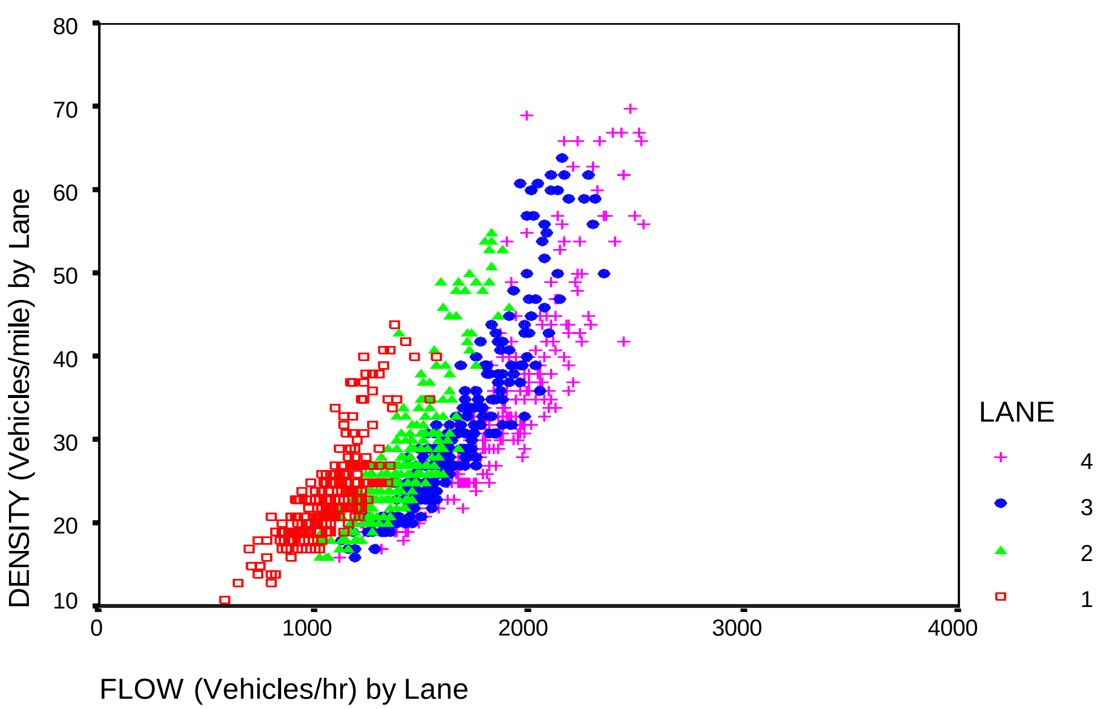

Figure 2-6 Paramics1.5 Flow-Density Characteristics

MODEL VALIDATION

To validate the previous results, the calibrated model including aggressiveness, awareness, mean headway and reaction time, was then subjected to two phases of validation. The first calibration phase focused on the section of the I405 freeway that was used for calibration. The second phase of validation involved not only this section but rather the entire network. Once an entire network is used, the effect of the Origin-Destination (O-D) data comes into play as will be discussed in due course. In all cases, real demand data were used to drive the simulator, and the aggregated output from multiple runs was compared to real detector data collected from the field.

VALIDATION ON THE I405 FREEWAY SECTION

The same section of the I405 freeway, shown in Figure 1, was used for the first phase of validation. Five field data sets were collected covering varying traffic conditions in different days, in the morning peak, afternoon peak and the in between off-peak periods. Each data set is two hours long. For each of the time periods during which real data were collected, five simulation runs were conducted to compare the aggregated results to the real data. i.e. twenty-five simulation runs in total were conducted in this phase, for two hours long each. In all runs, the real data were used to formulate the demand and profile files in Paramics1.5.

Table 3-1 presents a summary of the total demand comparison. In addition to the case-by-case outputs, the average value of the total demand volumes obtained from the five data sets was calculated, and then compared to the result obtained previously in the calibration scenario. Results indicate that the average error in the validation phase is greater than the case of testing after calibration, which is expected. Nevertheless, error magnitudes seem reasonable. Also the error magnitudes seem to increase as demand increases pointing again to the issue of vehicle stacking in memory awaiting suitable gap to be released onto the network. Results of total outflow shown in Table 3-2 also support this observation. In Table 3-3, the lane usage percentages compare favorably to field observations. Errors are slightly higher than the case of testing on calibration data. The occasional higher errors in the case of lane 4 is again related to the percentages of usage of the HOV lane which does not seem to be directly controllable in Paramics1.5 as pointed out previously. Figures 3-1 to 3-6 compare the output of Paramics1.5 to the real data during the validation phase.

Table 3-1 Comparison of Total Demand

Case 1: 07:00-09:00 (08/04/97)		
VDS	Paramics1.5	Error (%)
13613	14150	3.94
Case 2: 16:00-18:00 (09/09/97)		
VDS	Paramics1.5	Error (%)
19511	17818	-8.67
Case 3: 11:00-13:00 (10/22/97)		
VDS	Paramics1.5	Error (%)
11279	11561	2.50
Case 4: 07:00-09:00 (07/17/97)		
VDS	Paramics1.5	Error (%)
14440	14936	3.43
Case 5: 16:00-18:00 (09/26/97)		
VDS	Paramics1.5	Error (%)
20426	18038	-11.69
Average		
VDS	Paramics1.5	Error (%)
15853	15301	-3.48
Model Calibration (previous scenario)		
VDS	Paramics1.5	Error (%)
12869	12949	0.62

Table 3-2 Comparison of Total Outflow

Case 1: 07:00-09:00 (08/04/97)		
VDS	Paramics1.5	Error (%)
12381	12695	2.53
Case 2: 16:00-18:00 (09/09/97)		
VDS	Paramics1.5	Error (%)
17023	15290	-10.18
Case 3: 11:00-13:00 (10/22/97)		
VDS	Paramics1.5	Error (%)
10719	10390	-3.06
Case 4: 07:00-09:00 (07/17/97)		
VDS	Paramics1.5	Error (%)
13004	13221	1.67
Case 5: 16:00-18:00 (09/26/97)		
VDS	Paramics1.5	Error (%)
17187	15404	-10.37
Average		
VDS	Paramics1.5	Error (%)
14026	13400	-4.46
Model Calibration (previous scenario)		
VDS	Paramics1.5	Error (%)
12025	12132	0.89

Table 3-3 Comparison of Lane Usage

Case 1: 07:00-09:00 (08/04/97)				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.17	0.25	0.28	0.30
Paramics1.5	0.16	0.23	0.28	0.33
Error (%)	-1	-2	0	3
Case 2: 16:00-18:00 (09/09/97)				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.22	0.24	0.26	0.28
Paramics1.5	0.17	0.24	0.29	0.30
Error (%)	-5	0	3	2
Case 3: 11:00-13:00 (10/22/97)				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.22	0.24	0.27	0.27
Paramics1.5	0.17	0.19	0.26	0.38
Error (%)	-5	-5	-1	11
Case 4: 07:00-09:00 (07/17/97)				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.17	0.24	0.28	0.31
Paramics1.5	0.17	0.22	0.28	0.33
Error (%)	0	-2	0	2
Case 5: 16:00-18:00 (09/26/97)				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.22	0.24	0.26	0.28
Paramics1.5	0.17	0.24	0.29	0.30
Error (%)	-5	0	3	2
Average				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.20	0.24	0.27	0.29
Paramics1.5	0.17	0.23	0.28	0.32
Error (%)	-3	-1	1	3
Model Calibration (previous scenario)				
	Lane-1	Lane-2	Lane-3	Lane-4
VDS	0.18	0.24	0.28	0.30
Paramics1.5	0.17	0.24	0.28	0.31
Error (%)	-1	0	0	1

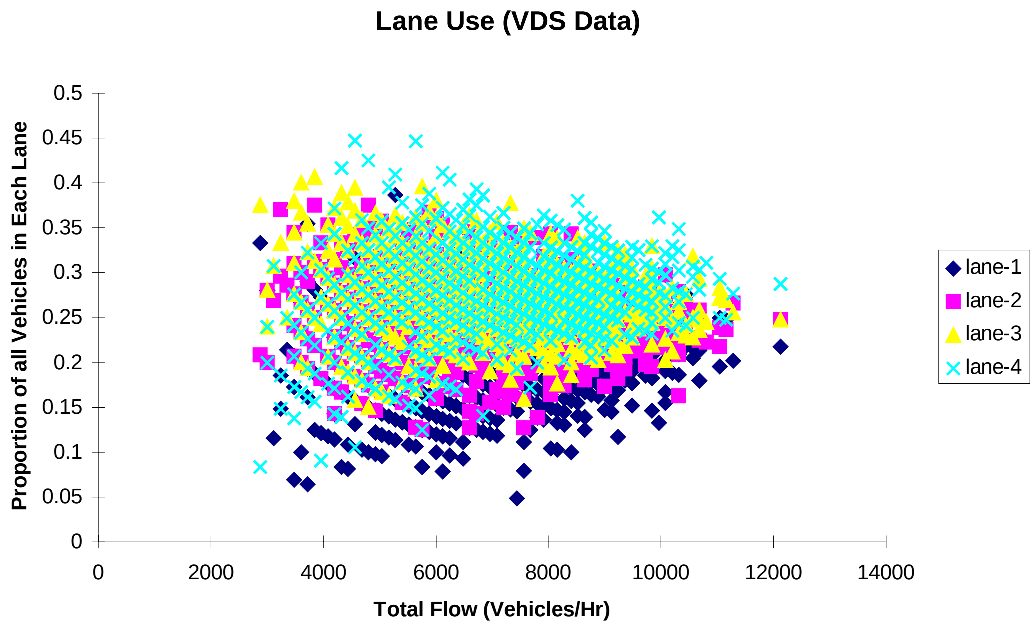

Figure 3-1 Lane Usage Based on VDS Data (for Model Validation)

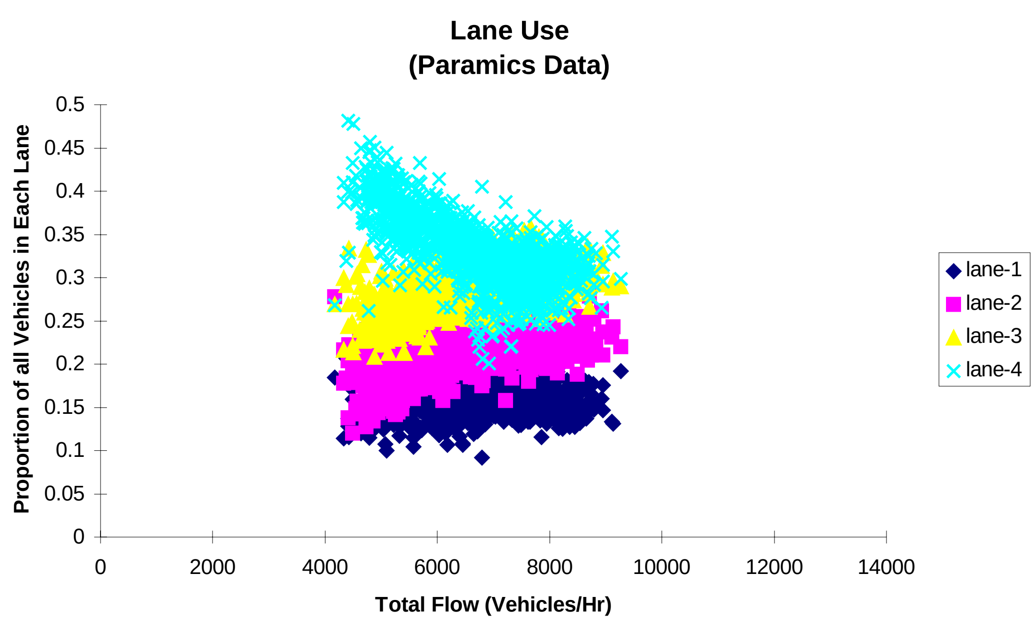

Figure 3-2 Lane Usage Based on Paramics1.5 Data (for Model Validation)

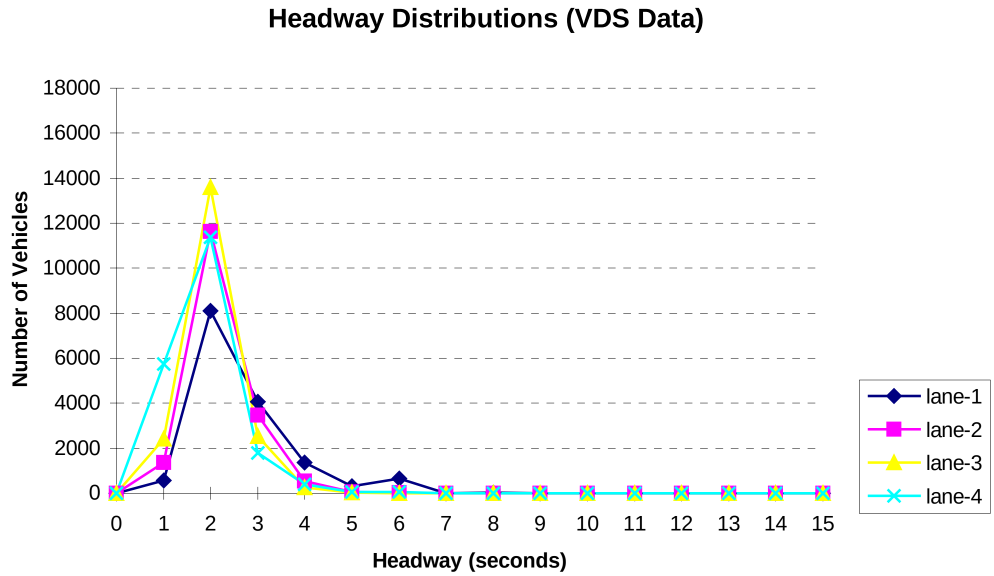

Figure 3-3 Headway Distributions Based on VDS Data (for Model Validation)

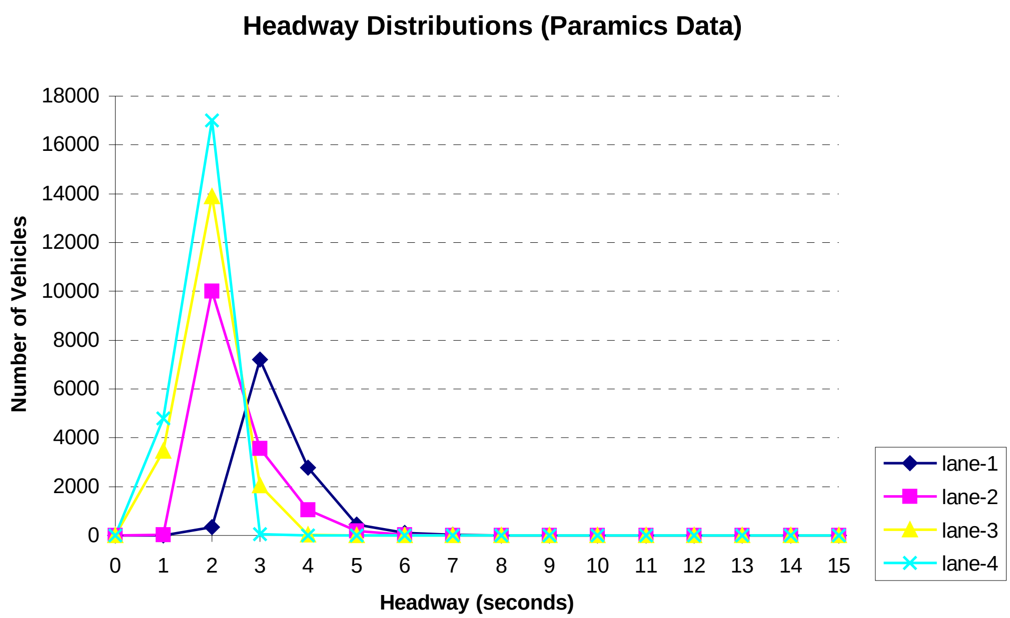

Figure 3-4 Headway Distributions Based on Paramics1.5 Data (for Model Validation)

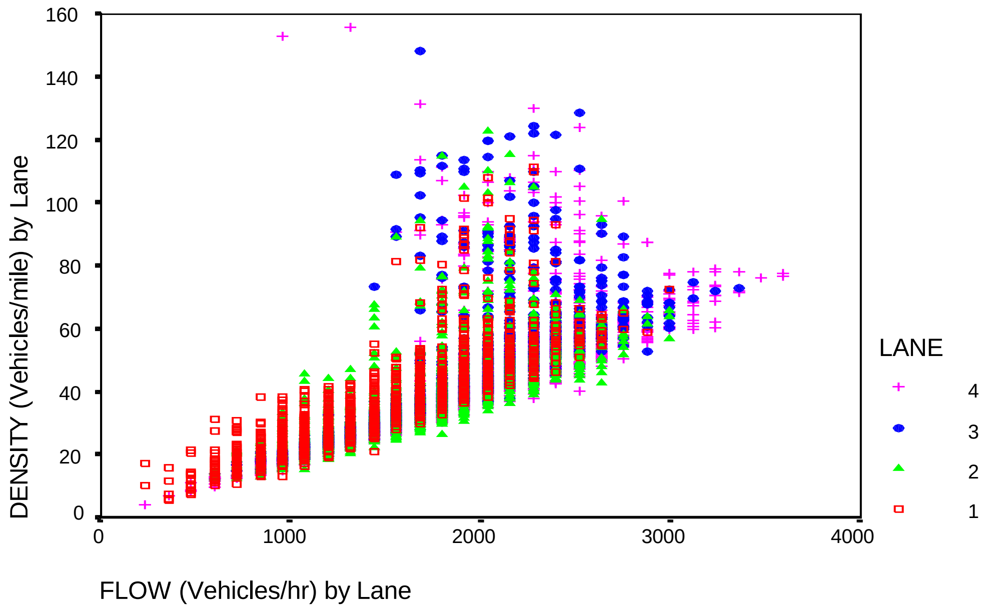

Figure 3-5 VDS Flow-Density Characteristics (for Model Validation)

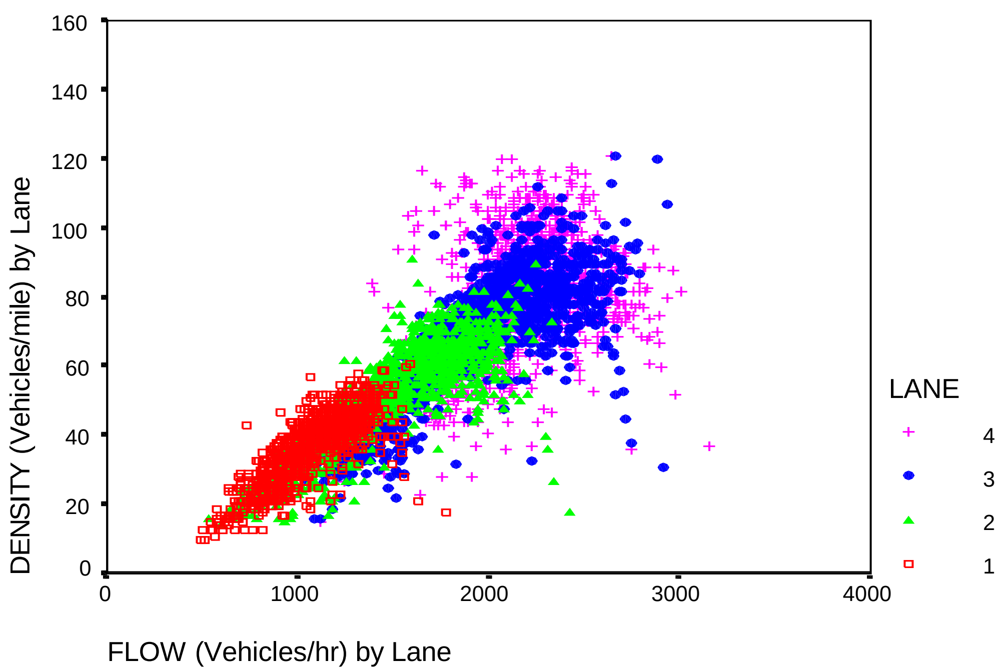

Figure 3-6 Paramics1.5 Flow-Density Characteristics (for Model Validation)

VALIDATION ON THE IRVINE NETWORK

So far, Paramics1.5 has been tested only on a single segment of the I405 freeway. Testing on such a single segment has pros and cons. The main advantage is that allows for the evaluation of the underlying models without exogenous errors from Origin-Destination (O-D) estimation, signal timing coding and the like. The goal, however, is examine the performance of the simulator on a full network, including freeways as well as surface streets, and for which real demand and performance data are available. Therefore, in this phase of validation, we use the parameters calibrated previously and simulate a portion of the California testbed network, known as the Irvine network shown in Figure 4-1. The network is located in the city of Irvine, bounded by a triangle of two freeways (I-405 and I-5) and one major arterial (Jeffrey Road).

Coding the network in Paramics1.5 involved coding the geometry of freeways and surface streets. The elements coded for the geometry of freeways include 1) mainline segments, 2) on-ramps and off-ramps and 3) interchanges. Link characteristics, such as the number of lanes, the lengths of on-ramps and off-ramps, lane width, lane restrictions for the HOV lanes, were imported from a TRANPLAN network and manually refined based on real geometry data. Coding the surface street portion of the network involved: 1) the numbers of lanes for all the approaches to intersections, 2) curb location within each intersection, 3) lane composition for each approach, 4) stop-line location, and 5) signal timing plans. Fine geometry details are emphasized in Paramics1.5. Obtaining geometry information for the surface streets was more difficult than freeways, mainly because intersection layouts change more frequently and updated information is rarely available. The research team, in many cases, resorted to making field visits to visually update geometry information and intersection layouts. Another surface-street coding obstacle was signal-timing plans at intersections. In Irvine, signalized intersections are controlled by either fully actuated or semi- actuated signals. Paramics1.5, however, only allowed for fixed-time signalization. Using the maximum greens for approximating actuated signal operation during peek traffic did not always perform well and created unrealistic queues at some intersections. Therefore, using experimentation in a calibration-like manner, different signal timing plans in the study network were tested and adjusted until anomalous traffic congestion was removed from each intersection in the network. The final signal timing plans were then used for simulating the network.

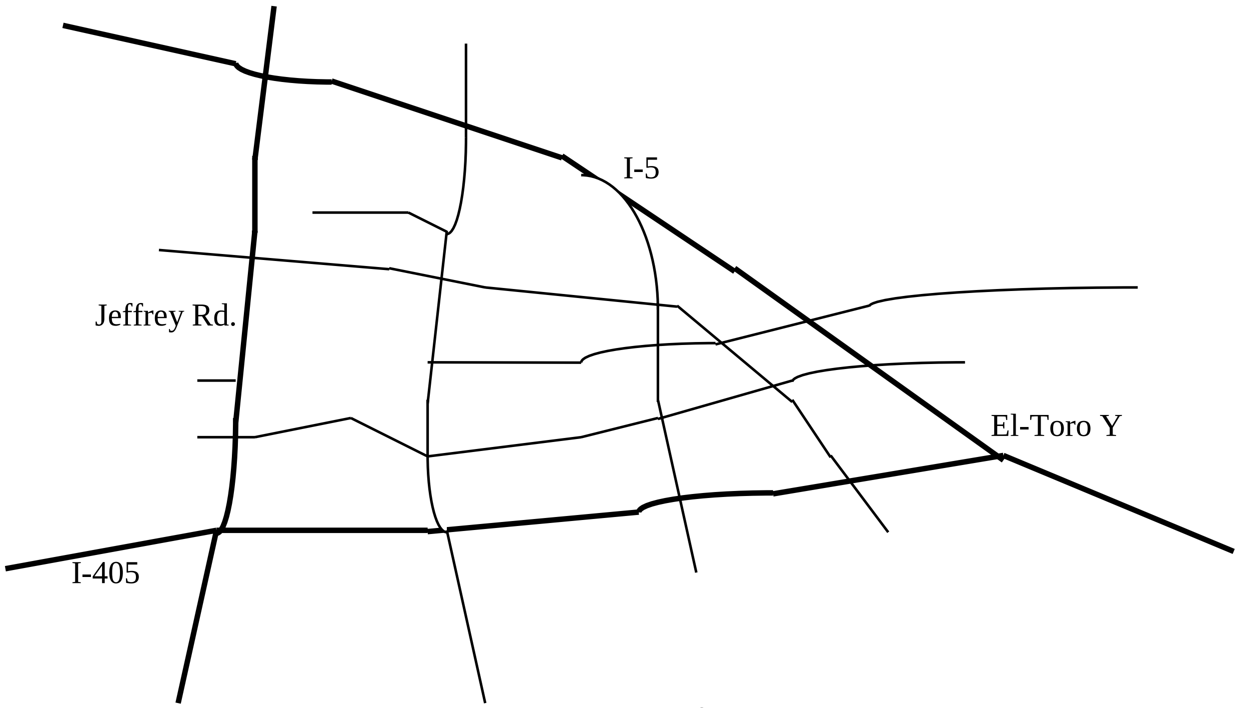

Figure 4-1 A schematic of the Irvine network

The O/D data were obtained from another ongoing UCI testbed project (McNally 1998), the output of which is dynamic O/D data (15 minutes) for the Irvine network. The procedure used included inputting PM peak O/D data from 1995, obtained form the Irvine Transportation Analysis Model (ITAM) iteratively into Contram and Comset traffic models. Contram assigns the O/D demand onto the network and outputs the number of vehicles (herein referred to as packets) using specific routes. Output from Contram serves as input into Comset, along with observed network link count information, to provide updated 15-minute O/D data for the PM peak. This updated data serves as input to Contram once again, and the cycle is repeated till convergence is attained. The final output is (1) 15-minute O/D data, and (2) packet information with both departure times and identified routes. In Paramics1.5, 106 O/D zones were coded, matching the O/D data from McNally (1998).

Field data availability is an important factor taken into account when determining the performance measures to be used while testing. In this study, field data corresponding to the O/D matrices that have been used, are limited to 1) total demand and supply volumes associated with each zone, and 2) volumes collected at some specific links within the network.

Taking into account the available field data, loop detectors were simulated in Paramics1.5 to collect simulated output that corresponds to the real data. First, to measure the total demand associated with each zone, 212 loop detectors were set in the simulated network. Two detectors were assigned to each zone to collect the total volume attracted to and generated by the zone. Second, seven extra loop detectors were created on different links within the network to collect simulated link volumes at the same locations as the in the field. Locations of these detectors are as follows (Figure 4-2):

(1) I-405-NB
(2) I-5-NB
(3) Jeffrey-NB
(4) San Canyon-SB
(5) I-133-NB
(6) Irvine-EB
(7) Alton-EB

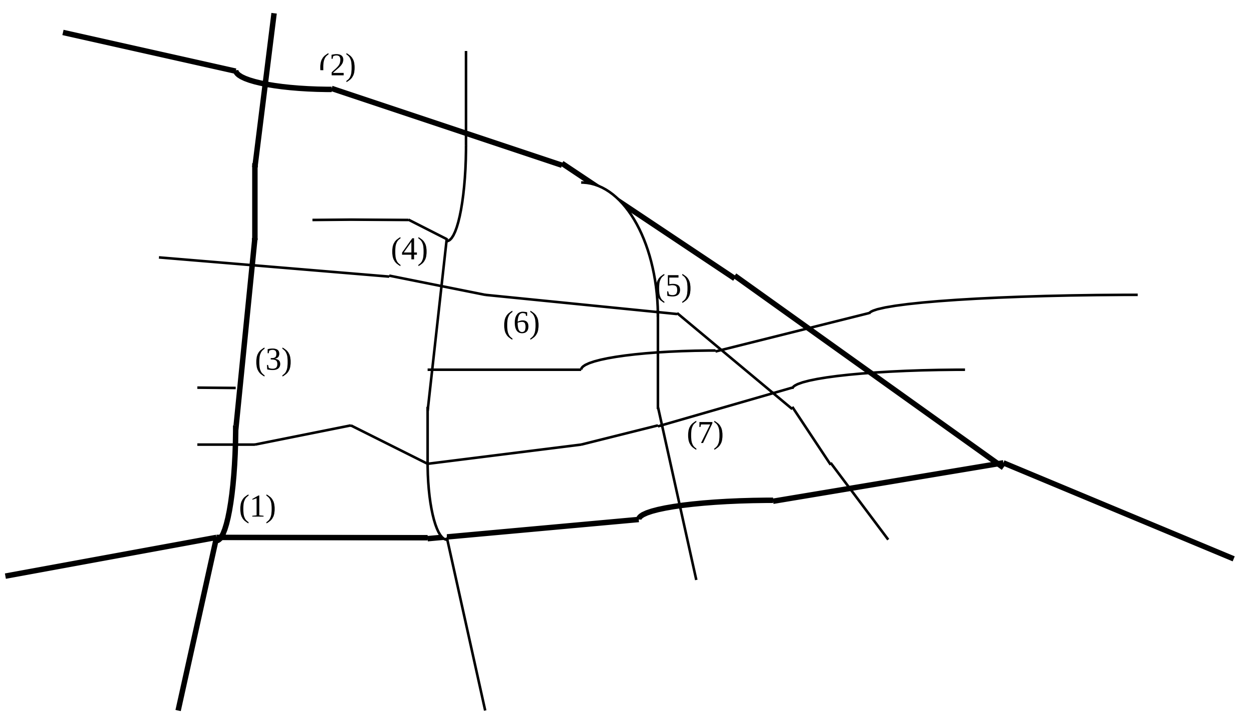

Figure 4-2 The Detector Layouts for Collection of Transition Link Volumes

Using the above detector layout, three simulation runs, 3-hours long each, were executed. For each 3-hour simulation run, lane volumes were measured from each loop detector and updated every one hour. Data associated with each run were aggregated, and the average volumes were used for comparison to field data.

Table 4-1 and Figure 4.3 show the simulated versus real zone traffic generation (departure) and attraction (arrival) over three hours. Table 4-2 compares loop detector reading from the simulator to those from the field. Error levels in Table 4.1 are notably high. Most likely, these errors are caused by the vehicle release mechanism onto the network. If vehicles get stacked in memory, the simulation period terminates without releasing all the vehicles, which directly affects the number of vehicles propagating through the network and ultimately reaching the destination. It should be noted that if the simulator is allowed to run longer than three hours, all vehicles eventually get generated onto and traversed through the network, but after the 3 hours. The length of time after the 3 hours depends on the level of ‘virtual congestion’ in memory at each zone; some zones take longer than others. Error levels in Table 4-2 are consequently high as well. Visual inspection of the simulator during operation indicate that the routing of vehicles, and their behavior near and within intersections seem normal. Therefore, we attribute the above error levels mainly to the vehicle releasing problem.

Table 4-1 Simulated versus Real Zone Traffic Generation and Attraction

zone	simulated generation	real generation	relative error (%)	simulated attraction	real attraction	relative error (%)	mean absolute error (%)
1	1176	1889	-37.7	1838	2317	-20.6	29.1
2	0	0	0	0	0	0	0
3	4342	3908	11.1	1798	1997	-9.6	10.4
4	799	949	-15.8	2025	2823	-28.2	22
5	1250	1727	-27.6	661	487	35.7	31.6
6	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0
8	264	340	-22.3	43	97	-55.6	38.9
9	0	0	0	0	0	0	0
10	11	12	-8.3	24	64	-62.5	35.4
11	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0
13	446	664	-32.8	701	1013	-30.9	31.7
14	882	1225	-28	60	127	-52.7	40.3
15	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0
29	289	458	-36.8	526	794	-33.7	35.2
30	0	0	0	0	0	0	0

Table 4-1(cont.) Simulated versus Real Zone Traffic Generation and Attraction

zone	simulated generation	real generation	relative error (%)	simulated attraction	real attraction	relative error (%)	mean absolute error (%)
31	0	0	0	0	0	0	0
32	2485	2548	-2.4	292	426	-31.4	16.9
33	159	191	-16.7	60	135	-55.5	36.1
34	69	154	-55.1	51	197	-74.1	64.6
35	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0
37	1455	1069	36.1	229	525	-56.3	46.2
38	1193	738	61.6	480	602	-20.2	40.9
39	741	1024	-27.6	169	271	-37.6	32.6
40	0	0	0	0	0	0	0
41	133	161	-17.3	115	239	-51.8	34.5
42	525	558	-5.9	0	0	0	2.9
43	0	0	0	0	0	0	0
44	291	444	-34.4	209	372	-43.8	39.1
45	919	1285	-28.4	103	169	-39.5	33.7
46	1398	1115	25.3	221	321	-31.1	28.2
47	0	0	0	0	0	0	0
48	0	0	0	0	0	0	0
49	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0
52	451	564	-20.1	7	17	-58.8	39.4
53	659	932	-29.2	19	62	-69.3	49.2
54	691	906	-23.7	6	9	-33.3	28.5
55	27	33	-15.1	0	0	0	0
56	696	892	-21.9	16	45	-64.4	43.1
57	112	152	-26.3	18	36	50.0	38.1
58	518	734	-29.4	3	4	-25.0	27.2
59	1295	1356	-4.4	495	355	39.4	21.9
60	493	682	-27.7	50	113	-55.7	41.7

Table 4-1(cont.) Simulated versus Real Zone Traffic Generation and Attraction

zone	simulated generation	real generation	relative error (%)	simulated attraction	real attraction	relative error (%)	mean absolute error (%)
61	570	832	-31.4	332	557	-40.3	35.8
62	1793	2673	-33.2	229	434	-47.2	40.2
63	0	0	0	0	0	0	0
64	77	132	-41.6	314	495	-36.5	39.1
65	0	0	0	0	0	0	0
66	1	1	0	44	68	-35.9	17.6
67	0	0	0	0	0	0	0
68	192	240	-20.0	59	96	-38.5	29.2
69	139	180	-22.7	0	0	0	0
70	87	133	-34.5	168	275	-38.9	36.7
71	0	0	0	0	0	0	0
72	0	0	0	0	0	0	0
73	0	0	0	0	0	0	0
74	0	0	0	0	0	0	0
75	0	0	0	0	0	0	0
76	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0
78	318	456	-30.2	634	499	27.5	28.6
79	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0
83	0	0	0	0	0	0	0
84	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0
86	14901	20912	-28.7	0	0	0	14.3
87	5816	19862	-70.7	0	0	0	35.3
88	3536	3170	11.5	3542	5303	-33.2	22.1
89	264	359	-26.4	6514	7230	-9.9	18.1
90	24139	33362	-27.6	0	0	0	13.8

Table 4-1(cont.) Simulated versus Real Zone Traffic Generation and Attraction

zone	simulated generation	real generation	relative error (%)	simulated attraction	real attraction	relative error (%)	mean absolute error (%)
91	0	0	0	24527	40712	-39.7	19.8
92	6030	5106	18.9	2975	4627	-35.7	27.3
93	2903	3996	-27.3	4332	5225	-17.7	22.1
94	0	0	0	0	0	0	0
95	994	1032	-3.6	909	1452	-37.3	20.4
96	0	0	0	14056	19073	-26.3	13.1
97	973	1335	-27.1	564	881	-35.9	31.5
98	2258	2277	-0.8	794	1103	-28.0	14.4
99	3026	3436	-11.9	3522	3586	-1.7	6.8
100	0	0	0	18388	21492	-14.4	7.2
101	4435	4979	-10.9	1349	915	47.4	29.1
102	1368	1235	10.7	5660	5827	-2.8	6.75
103	4563	4699	-2.8	3657	5307	-31.1	16.5
104	360	489	-26.3	2570	3656	-29.7	28.0
105	0	0	0	0	0	0	0
106	0	0	0	0	0	0	0

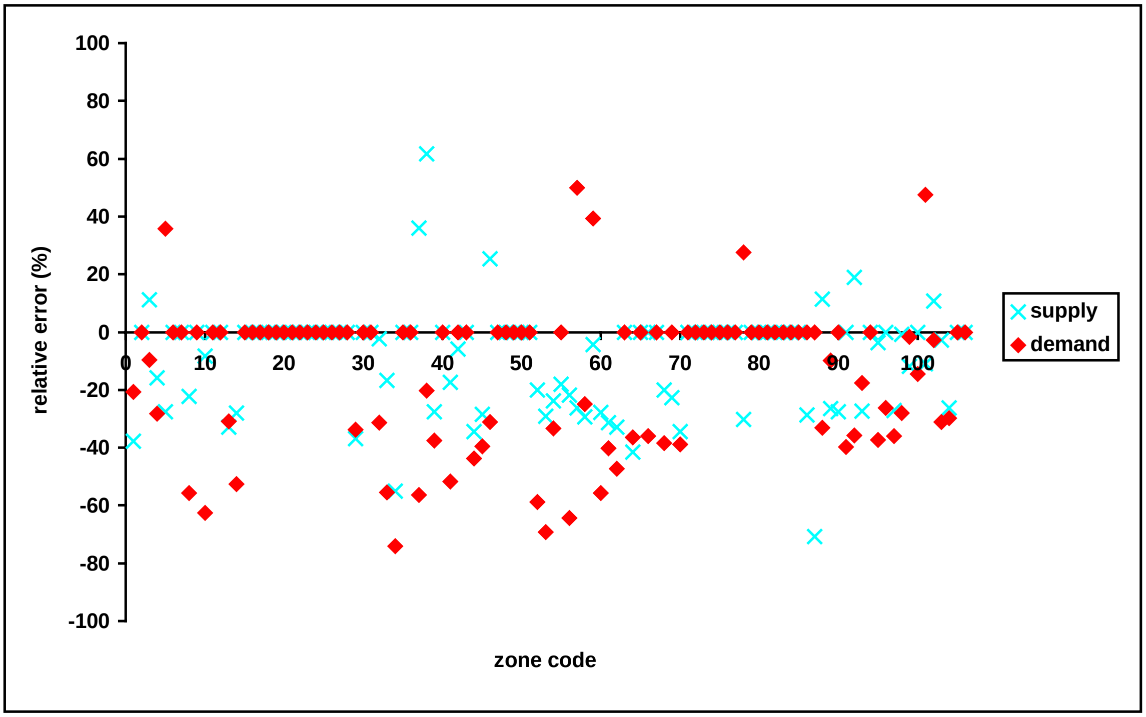

Figure 4-3 Distributions of the Relative Errors in Generation and Attraction Volumes

Table 4-2 Comparisons of Link Volumes

detector code	simulation value	real value	relative error (%)
(1) I-405-NB	16152	19172	-15.8
(2) I-5-NB	12109	17006	-28.8
(3) Jeffrey-NB	5018	4708	6.6
(4) San Canyon-SB	2040	2210	-7.7
(5) I-133-NB	3807	5349	-28.8
(6) Irvine-EB	1406	2189	-35.8
(7) Alton-EB	2037	2292	-11.1
mean relative absolute error (%)	19.2		

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

In this study, a comprehensive list of requirements for a microscopic traffic flow simulator to successfully model ITS was envisioned and presented. The evaluation template covered [1] supply/control aspects, [2] demand/behavior aspects, [3] environmental-related aspects, and [4] simulator-performance aspects. The template was used to subjectively evaluate Paramics1.5, a new and potentially successful ITS-oriented micro-simulator. The subjective evaluation was then followed by calibration of selected key parameters underlying the simulators car-following and lane changing models. The performance of the simulator was validated in two phases, on a small freeway section and on complete network of freeways and surface streets in the City of Irvine, California.

It was evident from the detailed description of Paramics1.5' capabilities and limitations that the software is an excellent 'shell' or 'framework' for a comprehensive and extensive transportation simulation laboratory. Paramics1.5 offers two very important and unprecedented features: high performance and scalability. To our knowledge, this is the most promising approach to handle realistic real word traffic networks under ITS. Also, Paramics1.5 offers very plausible detailed modeling for many components of the 'desired ideal' simulator. Nevertheless, limitations were also numerous as the subjective evaluation revealed. Improving and eliminating all limitations is not envisioned to be a simple task for the Paramics company. Therefore, there must be a mechanism for researchers to use Paramics1.5 in a "framework" or "shell" mode and complement it with custom features of their own. Consequently, the Application Programming Interface (API), offered and being matured by Paramics, is regarded to be the most important capability of the software. A truly comprehensive API would be the gateway for researchers to the heart of the software, without having to deal with the underlying proprietary code. The API should have a dual role: (1) to allow researchers to override the simulator's default models such as car following, lane changing, route choice models and the like, and (2) to allow them to interface complementary modules to the simulator. Complementary modules could be any ITS application such signal optimization, adaptive ramp metering, incident detection and management and any other important ITS application.

The following is a list of our recommendations for further Paramics1.5 improvements, based on the subjective evaluation:

1. Incorporate path-based routing for better travel choice behavior. It is understood that path-based routing will involve extensive book keeping and memory intensive tracking of individual vehicle's set of alternative paths, remaining trip times and so on. However, for the sake of realistic modeling of travel choice behavior, it seems necessary to incorporate path based routing. Details of implementation, and whether to code it internally or via the API, still need to be worked out. The entire path-based routing could also be provided as a research tool option that the user could turn on and off,
2. Developing the capability to trace any number of selected vehicles or vehicle classes based on a vehicle-specific attribute. For instance, a variable 'trace' could be defined and toggled on and off for each vehicle type. When trace is on, the simulator would collect and aggregate statistics for marked vehicles, such as routes taken, travel times, delays, etc,
3. Providing turning fractions as output,
4. Improving incident modeling, to enable the user to control [1] the exact incident location on the link, [2] the exact onset time of the incident, [3] the number of blocked lanes and the exact affected lane(s), and [4] incident duration. Also incorporate a rubber-necking factor or similar for modeling shoulder incidents. Collecting incident-specific delays is important as well,
5. Providing access to and redefining of vehicle types' attributes. For instance, the user should be able to know the attributes of vehicle type 'x' and re-define them as needed,
6. Providing access to existing car-following and lane changing model parameters for calibration purposes (without the need to replace the models through the API),
7. Providing graphical display for loop output vs. time,
8. Providing aggregation tools for multiple run outputs,
9. Maturing the API to enable easy interfacing of external modules such as:
   - O/D estimation and prediction using real-time counts,
   - incident detection modules,
   - Incident management and network management modules,
   - Adaptive signal control,
   - Adaptive ramp metering and freeway control,
   - Dynamic user equilibrium and system optimal routing models,

- Etc.

10. Expanding existing documentation, and providing ‘usage’ as well as ‘calibration’ tutorials and guidelines.
11. Modeling of pedestrian-actuated signals,
12. Providing lane-use control signals,
13. Modeling of driver compliance to continuous information,
14. Providing direct interface with Geographical Information Systems (GIS) to enable automated network editing as well as direct use of GIS information in transportation modeling,
15. Modeling coarse traveler information systems such as Highway Advisory Radio (HAR),
16. Modeling hybrid toll plazas, and proving a vehicle-specific ‘toll pass” variable to enable selected vehicles to pass toll plazas without stopping,
17. Modeling on street curb parking,
18. Activating the effect of car-park capacity and adding re-routing logic if a car park is full,
19. Defining weather zones that would affect driving behavior parameters on the enclosed links accordingly,
20. Providing graphical display of time-space diagrams,
21. Providing schedule adherence assessment tools for transit buses,

In terms of objective performance, the calibrated model performed well on a small freeway section, but errors were relatively high when applied to the Irvine network. It was concluded that these errors are caused by a single problem, which is the stacking of vehicles in memory, if they can not find a suitable gap to be released onto the network. This ‘virtual congestion’ causes the simulator to release fewer vehicles than in real life, and consequently fewer vehicles traverse the network and reach the destination within the simulation period. If the simulator is allowed to run for longer than the demand period, the remaining vehicles eventually make it to and through the network, taking longer than the specified demand period. It is not clear to us how to overcome this problem. One possibility that has not been tested here is to model more road space on which vehicles can be created within each zone, possibly by modeling minor roads as well and not only major arterial roads. Another possibility is to use large contiguous zones that correspond to actual geographical zones. This way it might be possible to avoid releasing high demand levels on a few links, causing virtual congestion.

ACKNOWLEDGMENT

This research was funded by the California Partners for Advanced Transit and Highways (PATH) headquartered at the University of California Berkeley, and the California Department of Transportation (Caltrans). It was also facilitated for by the California ATMIS Testbed, headquartered at the University of California Irvine.

REFERENCES

Greenshields, B.D. (1947), “Traffic Performance at Intersections”, Yale Technical Report #1, 1947.

Jeffreys, W. (1994), “Motorway Lane Discipline: Motorway Traffic Modeling Study” Technical Report, Transportation Research Laboratory, March 1994.

Johansson, G. and Rumar, H. (1971), “Drivers’ Brake Reaction Times”, Human Fact., 13(1), 23-27.

King, G. and Wilkinson, M. (1973), “Relationships of Signal Design to Discharge Headway”, Transportation Research Laboratory, report 615, 1973.

Kunzman, W. (1978), “Another Look at Signalized Intersection Capacity”, ITE Journal, August 1978.

McNally, M. G. (1998), “Evaluation of the Irvine Field Operation Test”, Ongoing Research Project.

Paramics Traffic Simulation Ltd., (1997), Paramics v1.5 manuals.