



PII: S0968-090X(96)00006-X

A MICROSCOPIC TRAFFIC SIMULATOR FOR EVALUATION OF DYNAMIC TRAFFIC MANAGEMENT SYSTEMS

QI YANG¹ and HARIS N. KOUTSOPOULOS²¹Department of Civil and Environmental Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, U.S.A.²Department of Civil and Environmental Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.*(Received 5 June 1995; in revised form 6 June 1996)*

Abstract—A Microscopic Traffic Simulator (MITSIM) has been developed for modeling traffic networks with advanced traffic control, route guidance and surveillance systems. MITSIM represents networks in detail and simulates individual vehicle movements using car following, lane changing, and traffic signal responding logic. A probabilistic route choice model is used to capture drivers' route choice decisions in the presence of real time traffic information provided by route guidance systems. The simulator is a component of a larger system for evaluating traffic management systems and interacts with a surveillance module that can represent a wide variety of sensors (e.g. loop detectors, area sensors, probe vehicles, etc.) and a traffic management module which sets traffic signals and signs, routing recommendations, etc. MITSIM is coded in C++ using object-oriented design and supports distributed implementation. It includes a graphical user interface for animating vehicle movements in the network and displaying aggregate traffic information such as speed and density. Copyright © 1996 Elsevier Science Ltd

1. INTRODUCTION

A number of simulation tools, mostly developed one or two decades ago, exist for assisting the testing, verification, and improvement of traffic management strategies (see Koutsopoulos and Yang, 1992 for a review). However, none of these models can effectively meet the requirements of simulating large-scale traffic networks for ITS applications, especially at an operational level, and are currently undergoing extensive updating and enhancement (Santiago and Kanaan, 1993). The emerging ITS technologies add new functionalities in traffic management systems, such as mainline traffic control, real-time route guidance, and incident management.

New traffic simulation models have been developed or are under development for ITS applications. Examples are DYNASMART (Jayakrishnan *et al.*, 1995; Mahmassani and Jayakrishnan, 1991), INTEGRATION (Van Aerde and Yagar, 1988) and THOREAU (Codelli *et al.*, 1992). The simulation model in DYNASMART and INTEGRATION is mesoscopic, designed mainly for dynamic traffic assignment applications. THOREAU is a microscopic model developed for evaluation. However, it has a very long running time. Other microscopic simulation models are also under development for modeling Automated Highway Systems (Eskafi *et al.*, 1996).

In this paper we present MITSIM (Microscopic Traffic Simulator), a model whose objective is to simulate integrated traffic networks supported by advanced traffic control and surveillance systems. This model is designed to establish a laboratory environment for testing and evaluating designs of Advanced Traffic Management Systems (ATMS) and Advanced Traveller Information Systems (Ben-Akiva *et al.*, 1996). MITSIM represents road networks in detail and simulates vehicle movements using car following, lane changing, and traffic signal responding logic. A probabilistic route choice model is used to capture drivers' route choice decisions in the presence of real time traffic information provided by the route guidance system.

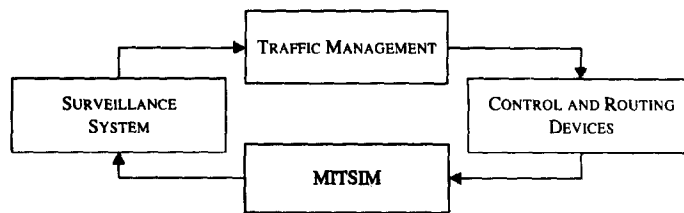


Fig. 1. Simulation framework for dynamic traffic management.

MITSIM is one component of a system developed for evaluating dynamic traffic management systems (Ben-Akiva *et al.*, 1994, 1996). This system consists of four modules: surveillance system, traffic management, control and routing devices, and MITSIM (see Fig. 1). The *traffic management module* represents the candidate traffic control and routing logic under evaluation. The control and routing strategies provided by the traffic management module feed into MITSIM via the traffic control and route guidance devices in the simulated network. MITSIM simulates the “state of the network” in detail. Drivers in the network respond to the various traffic controls and guidance while interacting with one another. The movement of vehicles is recorded by the surveillance system module consisting of traffic sensors and probe vehicles. The surveillance system then provides the traffic management module with the required data for generating control and routing strategies. This design provides flexibility by separating the modeling of the control and surveillance system from the simulation of traffic flow and facilitates distribution of tasks to multiple processors.

MITSIM is coded in C++ using object-oriented design and supports distributed implementation. It includes a graphical user interface for animating vehicle movements in the network and displaying dynamic changes of aggregate traffic information such as speed and density.

2. STRUCTURE AND ELEMENTS

In this section we first describe the overall design of the simulator. Then we present its main elements: network, traffic controls and surveillance sensors, incidents, travel demand, vehicle routing, and vehicle movements. We conclude by discussing the main simulation output.

2.1. Overall Design

The main structure of the traffic simulation model consists of a loop over a set of modules at specified frequencies (time-based) or when certain events occur (event-based). The simulation logic is summarized in the flowchart in Fig. 2.

The simulation starts with loading the simulation parameters, road network, and scenario definition, and initializing the communication channels with the traffic surveillance and management modules. Then, an iterative procedure is initiated with a pre-specified step size. The tasks performed within each iteration include:

1. Update the state of traffic signals, signs, and incidents.
2. Update shortest paths and routing tables.
3. Read new origin–destination (O–D) trip tables and place corresponding vehicles in virtual queues.
4. Load vehicles from the virtual queues into the network.
5. Update vehicles’ acceleration rates and check whether they need to change lane and whether the gaps are acceptable for the desired lane change.
6. Advance vehicles to new positions and update their speeds. If a vehicle activated a sensor, the corresponding measures (speed, occupancy, etc.) are recorded by the surveillance system module. At the end of a lane, a vehicle is either removed from the network (if it arrives at its destination) or handed to the downstream lane.

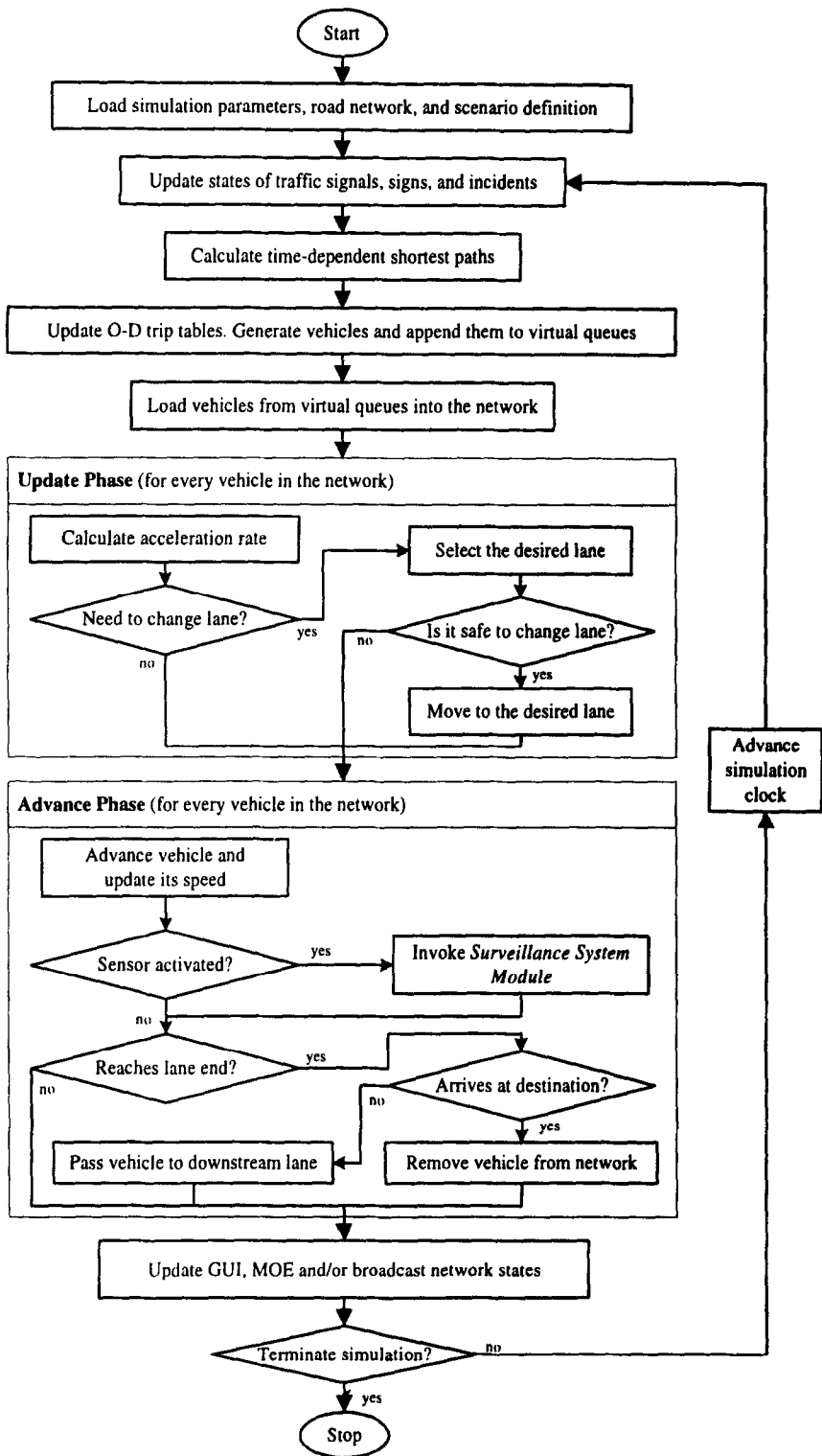


Fig. 2. Flow chart of the traffic simulation model.

7. Update the display if the graphical user interface (GUI) is enabled.
8. Calculate measures of effectiveness (MOEs) or send network states to external MOE or GUI modules.
9. Update the simulation clock and move to next iteration.

MITSIM mainly uses a time-based simulation approach in processing the vehicle movements. The car following, lane changing and event and signal responding functions are invoked for each vehicle every ω seconds (e.g. 1 s). Speeds and positions of the vehicles and states of surveillance sensors are updated every τ s (e.g. 1/10 or 1/2 s). The step size ω has to be greater than or equal to the step size τ and a modulus of τ . These fixed step sizes are adopted for the general flow control. For individual vehicles the step sizes for invoking particular simulation functions may vary (but must be a modulus of τ) as drivers' reaction times differ or vehicles meet particular conditions (e.g. too close to the leading vehicle, delayed at toll booth, etc.).

2.2. Network Representation

MITSIM represents road networks by nodes, links, segments, and lanes. Its structure allows the simulation of traffic operations in integrated networks of freeways and urban streets. The data that describes the simulated network is read from the network database, which can be created using an interactive graphical *road network editor* developed for MITSIM. The network database includes descriptions of all the network objects, lane connections, lane-use privilege, regulation of turning movements at intersections (no left-turns, for example), traffic sensors, control devices, and toll plazas.

Nodes. A node is either an intersection of several roadways or a source or sink where traffic flows enter or leave the simulated network. Each node is represented by its *type* (intersection, source/sink), a unique *identification number*, and an optional *name*.

Links. Links are directional roadways that connect nodes. Each link may consist of one or more segments and is characterized by its *type* (freeway, ramp, urban street, tunnels, etc.), an *identification number*, *starting* and *end nodes*, and the *segments* it consists of. An inbound link and an outbound link of a node are *connected* if there exists at least one lane connection between these two links. Turning restrictions from one link to another can also be specified by a *turning prohibition table*.

Segments. Segments are road sections with uniform geometric characteristics such as number of lanes, grade, curvature, design speed, etc. Each segment is characterized by its *speed limit*, *design speed*, *grade*, *geometry*¹, a unique *identification code*, and the *lanes* it consists of.

Lanes. MITSIM represents the network at the lane level. Each lane is described by two data items: (i) *lane code*, a unique identification code; and (ii) a *lane change regulation* and *lane-use privilege* code. Lane change regulations determine whether lane changes between adjacent lanes are allowed. Lane use privilege specifies the class of vehicles that are allowed to use the lane. For example, a lane may be assigned to high occupancy vehicles (HOV), commercial vehicles, and/or electronic toll collection (ETC) vehicles only. The lane-use privilege code for a particular lane can be any consistent combination of predefined basic types.

Each lane can be connected to one or more upstream and downstream lanes (e.g. at toll plaza) or has no lane connection at all (e.g. lane drop or network boundary). A *lane connection table* is used to represent the connections between upstream and downstream lanes.

2.3. Surveillance Sensors and Traffic Controls

Surveillance sensors. Surveillance sensors in MITSIM represent the various means used to extract data on traffic flows in the network. The general approach is that sensors are represented by their technical capabilities rather than by simulating their operations in

¹The geometry of a segment is referenced to the left curb line by *coordinates* of the end points and the *bulge* of the arc.

detail. For example, a measurement error is assumed instead of being derived by the physics that govern the operations of a sensor. When a vehicle passes a detector the occupancy and speed are interpolated based on its previous and current positions over the update phase (e.g. 0.1 s), assuming constant acceleration.

Representation of the following types of sensors is supported:

- *Point data sensors* for extracting traffic counts, occupancy, speed and headway at a fixed point in the network (e.g. loop detectors).
- *Point data sensors* for extracting information on individual vehicles such as vehicle type, over-height attributes, etc.
- *Point to point data sensors* for extracting information such as travel time from one point in the network to another (e.g. probe vehicles).
- *Area wide sensors* such as radar detectors and video cameras.

Sensors of the same type at the same longitudinal position in a segment are organized into a *sensor station*. Each sensor station is characterized by the *sensor type*, *sensor task* (i.e. the types of data collected by the sensor), length of *detection zone*, *position*, and the *detectors* it consists of (a subset of the lanes may be equipped). In addition, each sensor is assigned a working probability, which specifies the probability that the sensor is operational. The measurement errors for various types of sensors and data items are determined by parameters in the surveillance system module.

Simulation of other surveillance methods, such as external agency reports, is also possible in MITSIM. For example, when passing an incident, a vehicle equipped with a cellular phone will report the incident to the traffic management simulator with a pre-specified probability.

Traffic control devices. MITSIM supports simulation of a wide range of traffic control and route guidance devices: *intersection controls* (e.g. traffic signals, yield and stop signs); *ramp controls* (e.g. ramp metering); *mainline controls* (e.g. lane use signs (LUS), variable speed limit signs (VSLS), variable message signs (VMS), portal signals at tunnel entrances); and *electronic route guidance devices*.

Control devices can be either link-wide or lane specific. A link-wide device (e.g. VSLS, VMS) controls all the approaching traffic, while a lane specific device (e.g. LUS) controls only the approaching traffic in a particular lane. Control devices are characterized by their *location* (segment, lane, position), *type*, *initial state*, and *visibility*. Visibility represents the maximum distance from which the device is visible by the approaching drivers.

At any given time, the state of a traffic signal is blank, red, yellow, or green. The state may also indicate whether the signal is flashing (stop signs are represented by flashing red and yield signs by flashing yellow).

2.4. Toll Plazas

The description of each toll plaza includes items such as: *visibility*, *segment code*, *location* in the segment, and a list of data items for each toll booth, such as *lane use privilege code*, *default state* (free, open, or closed), *speed limit*, and *toll collection delay*. The lane use privilege code is used to restrict the types of vehicles allowed to use the toll booth. For example, a toll booth can be designated for exact change, electronic toll collection (ETC), or any vehicle. The speed limit sets an upper bound on the speed of the vehicles passing through the toll booth (for booths with manual toll collection, this speed limit is zero).

A vehicle will stop if it enters a booth using manual collection or slow down if it is on an ETC lane. The delay at a manual toll booth is assumed to have a negative exponential distribution with mean μ and truncated at some maximum delay, i.e.:

$$d = \min \{ -\mu \ln(r), d_{\max} \} \quad (1)$$

where d is the delay for paying toll; d_{\max} is the maximum delay; μ is the average delay at the toll booth (a function of toll booth type and vehicle class); r is a random number uniformly distributed between (0,1].

2.5. Incidents

Incidents can be activated and cleared at any particular time and location in the network. Each incident may affect one or multiple lanes. The information for an incident includes: *visibility*, *number of lanes affected*, *segment code* and *position*. Lane specific information for an incident includes the *severity* of the incident and its *length*, *maximum speed*, *start time*, and expected *duration*. The maximum speed of an incident sets an upper bound on the speed of the vehicles passing by and can be used to simulate the rubber-necking effect (partial blockage). The start time of an incident may be different from the time that the incident is detected by the traffic management system.

2.6. Travel Demand and Vehicle Characteristics

MITSIM accepts as input time-dependent origin/destination trip (O-D) tables. Each trip table includes the data items that specify the time that the trip table becomes effective and a list of departure rates for various O-D pairs. The time intervals of the trip tables may have different lengths. This allows the use of shorter time intervals during peak periods and longer time intervals during the off-peak period. The simulator stores elements of the current trip tables only. When the current trip tables expire, updated trip tables are read from the travel demand file.

Trip tables can be specified individually for each vehicle type or, alternatively, the simulator assigns randomly a type to each vehicle based on *fleet mix* specified in the parameter file. Vehicle type is a combination of vehicle class (e.g. high performance passenger cars, low performance passenger cars, intercity buses, trucks, and trailer trucks, etc.), lane-use privilege (e.g. HOV and ETC), and access to information (e.g. guided and unguided).

The inter-departure time between two vehicles is randomly drawn from a distribution which is defined based on traffic conditions. Under low congestion, for example, vehicles are generated according to a Poisson process, with inter-arrival time:

$$t_{n+1} = t_n - \frac{\ln(r)}{\lambda} \quad \lambda > 0, 0 < r \leq 1 \quad (2)$$

where $t_n + 1$ is the departure time for the next vehicle, and t_n is the departure time for the previous vehicle; λ is the demand rate (as specified in the O-D table); and, r is a random number uniformly distributed between (0,1). The most important parameters representing vehicle performance and driver's behavior are described below. The parameters regarding vehicle performance are deterministic, while driver behavior parameters (such as desired speed) are randomly assigned when a vehicle enters the network.

Maximum acceleration rate. The maximum acceleration rate is defined for each vehicle class as a function of the grade of the road segment and current speed of the vehicle. The default values for five speed categories and five grade categories are based on FHWA (1980) and FHWA (1994).

Normal deceleration rate. Normal deceleration rate is used to: (i) slow down a vehicle smoothly in non-emergency situations; and, (ii) compute the *normal stopping distance* required for responding to a downstream event (e.g. traffic signal, incident, and exit). The normal deceleration rate is a function of vehicle class and its speed. The default values for normal deceleration rates are based on Institute of Transportation Engineers (1982).

Desired speed. Driver's desired speed is randomly assigned when a vehicle enters the network and updated whenever the driver views a different speed limit sign or moves into a segment with different *design speed*. The desired speed is calculated as follows:

$$v_n^0 = \min \left\{ v_n^{sign} + v_n, \tilde{v}_n \right\} \quad (3)$$

where v_n^0 is the desired speed of driver n ; v_n^{sign} is the speed limit the driver viewed; v_n , a function of vehicle class and segment geometry (FHWA, 1980), is the maximum speed the vehicle can achieve in the segment; \tilde{v}_n is a driver behavior parameter randomly assigned

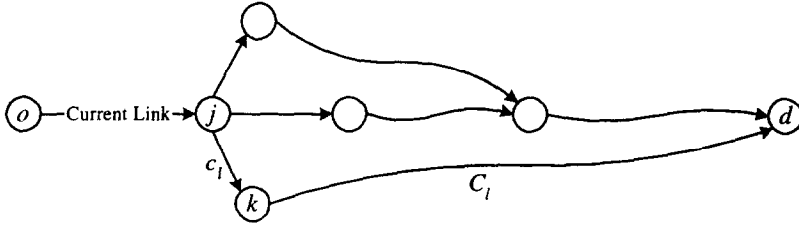


Fig. 3. Route choice.

based on the desired speed distribution. The distribution of v_n for several driver categories as well as the fraction of drivers in each category are given in the parameter file.

Speed distribution across lanes. To capture the speed difference across lanes, a maximum speed is also defined for each *lane* based on the *design speed* and the number of lanes in the segment, i.e.:

$$V_i^{\max} = \kappa_{mi} V^{\text{design}} \quad (4)$$

where V_i is the *maximum speed* vehicles can travel in lane i , V^{design} is the design speed of the segment; κ_{mi} is the speed factor for the i th lane in a segment consisting of m lanes. κ_{mi} captures the fact that, in segments with multiple lanes, the vehicles in the central and left lanes tend to travel faster than the vehicles in the right lanes. Suggested values of κ_{mi} are based on FHWA (1980).

Maximum speed. The maximum speed a vehicle can actually achieve at a given lane i is given by:

$$v_n^{\max} = \min\{v_n^0, V_i^{\max}\} \quad (5)$$

where v_n^0 and V_i^{\max} are driver's desired speed and the maximum speed of lane i , as defined in eqns 3 and 4 respectively. A driver may seek discretionary lane change (as described in section 2.8.3) if v_n^{\max} is constrained by V_i^{\max} instead of the driver's desired speed.

2.7. Vehicle Routing

A route choice model is used to capture drivers' route choice decisions and response to traffic information. Two driver types are assumed with respect to access to information: informed and uninformed drivers. For a particular driver type the route choice model calculates the probability of choosing an outgoing link at each intersection. Currently, the simulator uses a multinomial logit model and considers only travel time in choosing a route (see Fig. 3):

$$p(l | j, t) = \frac{\exp(u_l(t))}{\sum_{m \in L_j} \exp(u_m(t))} \quad (6)$$

where $p(l | j, t)$ is the probability to choose link l for a vehicle that expects to arrive at node j at time t ; L_j is the set of outgoing links from node j ; and $u_l(t)$ is the utility of choosing a route with link l as the next link, which is further defined as follows:

$$u_l(t) = \beta_1 \hat{c}_l(t) + \beta_2 \hat{C}_k(t + \hat{c}_l(t)) + \beta_3 z_l \quad (7)$$

where $\hat{c}_l(t)$ is the *expected time to traverse link l* for a vehicle that enters that link at time t ; $\hat{C}_k(t)$ is the expected travel time on the *shortest path* from node k (the downstream node of link l) to the destination of a vehicle that arrives at node j at time t ; z_l is a dummy variable that captures freeway bias; β_i are parameters estimated from appropriate data. The choice set, L_j , includes all the connected outgoing links at the downstream node of the current link that may take the vehicle *closer* to its destination. Thus a link must

satisfy $\hat{C}_k(t + \hat{c}_l(t)) \leq \hat{C}_j(t)$ in order to be included in the choice set (modified from Dial, 1971). This constraint also prevents vehicles from following a path that contains cycles. Based on the route choice probabilities individual vehicles choose a next link at each node. Alternatively, only links that belong to a set of predetermined paths may be considered.

The expected travel time to one's destination from each alternative downstream link is time-dependent and depends on the level of information the driver has access to. MITSIM calculates shortest paths from each link to all destination nodes for each driver group. A dynamic shortest path algorithm, which is a modified version of the *label correcting algorithm* (Ahuja *et al.*, 1994) is used. The algorithm takes into account: (i) the link travel times perceived by a particular driver group; (ii) the delays and regulations of turning movements at intersections; and (iii) the lane-use privilege. The travel times used in the shortest path calculations are time-dependent and the values depend on the traffic control system under consideration. For sophisticated ATIS/ATMS systems for example, predicted travel times, obtained from the *traffic management module*, are used and the shortest paths are updated periodically. The frequency at which shortest paths are updated depends on the capabilities and nature of the system to be evaluated.

The updated travel times are transmitted to vehicles equipped with on-board route guidance devices when they "view" communication beacons. Upon receiving the new information, guided vehicles select their routes based on the updated travel times. Any vehicle, including those without on-board route guidance devices, may all respond to VMS according to a compliance rate based the behavioral model.

More sophisticated models that incorporate other considerations in making route choice decisions can also be used (see, for example, Cascetta *et al.*, 1996). In addition, individual drivers can be assigned to pre-defined paths and invoke appropriate behavioral models when switching route becomes necessary. The modular design of the simulator allows for easy substitution of the default models.

2.8. Vehicle Movements

The movement of a vehicle in the network is determined by its interactions with the vehicles ahead, response to traffic controls, desired speed, and lane-use preference. These interactions are manifested in lane change decisions and acceleration and deceleration rate applied at any given time. The simulator maintains a linked list of vehicles in each lane and moves individual vehicles according to the *car following*, *lane changing*, and *event responding* models described in this section. The car following model computes the acceleration or deceleration rate of a vehicle in terms of its relationship with the leading vehicles; the lane changing model represents the behaviour of switching lanes; and the event responding model captures drivers' responses to traffic signals and signs, incidents, and toll booths.

2.8.1. Vehicle loading

Each vehicle enters the network from the upstream end of the first link on its path. Its initial position and speed are determined based on the simulation step size, driver's desired speed, and position and speed of the leading vehicles. If there is no space available in the entrance link, vehicles are stored in a queue and wait to enter the network during subsequent time intervals.

2.8.2. Acceleration/deceleration

The acceleration/deceleration rate of a vehicle is determined by its reaction to the vehicle ahead (car-following), the need to perform a merging maneuver, by responding to events (e.g. incidents), and by the desire to travel at its desired speed. Each one of these individual situations corresponds to a required acceleration/deceleration rate. Among all the factors that affect a vehicle's acceleration decisions, the one with the highest priority (i.e., returning the most constrained acceleration/deceleration rate) determines the effective acceleration rate of the vehicle in the next time step.

The current version of MITSIM, for computational efficiency reasons, does not simulate *explicitly* the driver's reaction time (except for stopped vehicles). The acceleration

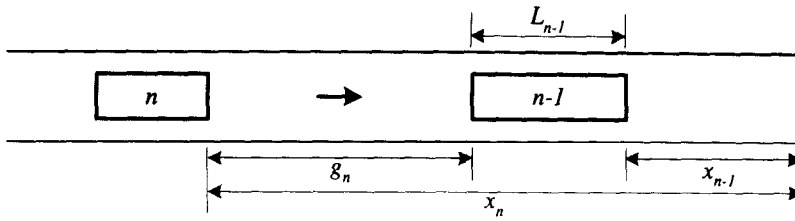


Fig. 4. Car following.

rate applied for the next ω seconds is calculated based on the situation at the end of the previous simulation interval. To make the behavior of vehicles more realistic the following exceptions to the constant reaction time have been implemented:

- When a vehicle enters the emergency regime at any time, the emergency deceleration rate is applied immediately.
- If a vehicle has changed lanes, the car following and event responding models are invoked in the next simulation interval (i.e. τ seconds later) to update the acceleration rate.
- During the advance phase (see Fig. 2), if the speed of a vehicle reaches the driver's desired speed or becomes zero, then the acceleration or deceleration rate is switched to zero.
- If a vehicle is decelerating, the car following and lane changing models are invoked $\omega/2$ s later (instead of ω s later).

In the following we describe in detail the determination of the acceleration rate for each situation.

Car following. The car following model calculates a vehicle's acceleration rate in terms of its relationship with the leading vehicle. This model is also used as a sub-model in a number of circumstances for calculating appropriate acceleration rate in order to make decisions on: (i) preparing to follow another vehicle if two or more lanes merge into a single downstream lane; and (ii) yielding to another vehicle shifting into the same lane. The car following model used in MITSIM draws upon previous research (see, for example, Herman *et al.*, 1959; Herman and Rothery, 1963; Wicks, 1977). The model is based on the headway and relative speed of the leading and the following vehicles. Depending on the magnitude of this headway, a vehicle is classified into one of three regimes: *free flowing*, *car following*, and *emergency decelerating*.

Free flowing regime. If the time headway is larger than a pre-determined threshold h^{upper} , the vehicle does not interact with the leading vehicle. In this case, if the vehicle's current speed is lower than its maximum speed (see eq. 5), it accelerates at the *maximum acceleration rate* to achieve its desired speed as quickly as possible; if current speed is higher than the maximum speed, the vehicle decelerates with the *normal deceleration rate* in order to slow down:

$$a_n = \begin{cases} a_n^+ & \text{if } v_n < v_n^{\max} \\ 0 & \text{if } v_n = v_n^{\max} \\ a_n^- & \text{if } v_n > v_n^{\max} \end{cases} \quad (8)$$

where a_n is the acceleration rate; a_n^+ is the maximum acceleration rate; a_n^- is the normal deceleration rate; v_n is the current speed; v_n^{\max} is the maximum speed.

Emergency regime. If a vehicle has a headway smaller than a pre-determined threshold h^{lower} , it is in emergency regime. In this case the vehicle uses an appropriate deceleration rate to avoid collision and extend its headway. Let x_n and x_{n-1} be the longitudinal

positions of the two vehicles (position is measured as the distance from the downstream end of the segment, see Fig. 4); L_{n-1} the length of the leading vehicle; $g_n = x_n - x_{n-1} - L_{n-1}$ the gap distance from the leading vehicle; and a_{n-1} the acceleration rate of the leading vehicle. The acceleration rate a_n for the following vehicle n is given by:

$$a_n = \begin{cases} \min\left\{a_{n-1} - \frac{(v_n - v_{n-1})^2}{2g_n}, a_n^-\right\} & \text{if } v_n > v_{n-1} \\ \min\{a_{n-1}, a_n^-\} & \text{if } v_n \leq v_{n-1} \end{cases} \quad (9)$$

Equation 9 guarantees that the following vehicle, in an emergency regime, will always decelerate to extend the headway to a safe range.

Car following regime. Finally, if a vehicle has a headway between h^{lower} and h^{upper} , it is in the car following regime. In this case the acceleration rate is calculated based on Herman's general car-following model (Herman *et al.*, 1959):

$$a_n = \alpha^\pm \frac{v_n^{\beta^\pm}}{g_n^{\gamma^\pm}} (v_{n-1} - v_n) \quad (10)$$

where α^\pm , β^\pm and γ^\pm are model parameters related to driver behavior. α^+ , β^+ , γ^+ are used for accelerating ($v_n \leq v_{n-1}$), and α^- , β^- , γ^- for decelerating ($v_n > v_{n-1}$) cases.

Merging. Merging operations may affect the effective acceleration. For the purposes of the simulator, merging is classified into: (i) priority-based merging; and (ii) merging without priority. Priority-based merging includes merging from ramps to freeways, from dropped lanes to mainline, and from minor streets to major streets. Merging without priority may occur at the downstream of toll plazas.

For priority-based merging, a vehicle *without* the right-of-way has to check whether there is any vehicle from the competing upstream lanes and executes the merge only if the projected *headway gap* is acceptable. If the headway gap is not acceptable, the vehicle either calculates the car-following acceleration rate by treating the vehicle which has the right-of-way as leader or prepares to stop at the end of the lane (depending on which case is the critical one).

For merging without priority, the right-of-way for merging is determined according to a "first come, first served" principle. In other words, among all the vehicles coming from competing upstream lanes, the one with the shortest headway from the downstream lane is chosen as the first vehicle to merge. Other vehicles will either follow appropriate leaders or prepare to stop at the end of their current lanes.

Event responding. The car following and merging models described in the preceding sections capture the behavior of the vehicle in response to the behavior of the leading vehicle or vehicles from competing upstream lanes. Another set of constraints in making acceleration decisions is imposed by various downstream events within driver's view. Such events include: (i) traffic signals and signs; (ii) incidents; (iii) making connection to the next link at the downstream node; and, (iv) yielding to another vehicle shifting into the same lane. These events may also influence drivers' lane change decisions, which are discussed in the next section.

Traffic signals and signs. Each control device in the network is assigned a visibility parameter (based on roadway geometry, weather conditions, etc.). When calculating the acceleration rate and making decisions on lane changes, a vehicle checks all the downstream signals and signs (including those in the next link) and reacts to the controls whose distances are less than or equal to a *normal stopping distance* defined as:

$$y_n = \max\left(-\frac{v_n^2}{2a_n^-}, y_{\min}\right) \quad (11)$$

where y_n is the distance required for a vehicle with speed v_n to decelerate to a stop by applying the normal deceleration rate a_n^- ; y_{\min} is a lower bound of the normal stopping distance.

If a viewed control device is within the above distance, the signal is red or yellow, and the travel time to the signal is longer than the expected remaining yellow time, the vehicle prepares to stop using a deceleration rate given by:

$$a_n = -v_n^2 / (2x_n) \quad (12)$$

where a_n is the deceleration rate to be applied in order to stop before the traffic light; x_n is the distance from the stop line; and v_n is the vehicle's current speed.

If the state of the control device is blank, green, or yellow and the time headway between the vehicle and the device is shorter than the expected remaining yellow time, the constraint defined by eq. (12) is ignored. For a flashing red control (including the stop sign), this constraint is used when the vehicle is moving and ignored when the vehicle is stopped at the stop line. For a flashing yellow control (including the yield sign), a vehicle may need to slow down using the following acceleration rate:

$$a_n = \frac{(v_n^{\max})^2 - v_n^2}{2x_n} \quad (13)$$

where v_n^{\max} is the maximum speed the driver can use at the control location (see eq. 5).

A vehicle calculates the corresponding acceleration rates for all the viewed controls within the normal stop distance (y_n) and uses the one with the minimum value as the *signal constrained acceleration rate*.

When the state of a traffic signal changes from red to green and/or the leading vehicle has moved, queued vehicles may not begin to move immediately because of the driver's response delay. The startup delay, defined as a function of the vehicle's *position in queue*, is used to model this problem.

Finally, the simulator allows the user to specify the probability that drivers comply with traffic controls. The response to signals discussed above applies only to complied drivers.

Incidents. An incident can be either a complete blockage of a lane (assigned a speed of zero), or a rubber necking effect where vehicles slow down to a particular speed. When an incident is within view and the vehicle cannot make lane changes to bypass it, the vehicle slows down and prepares to stop when the distance from the incident is less than the *normal stopping distance* (y_n) according to eq. (12).

Connection to downstream link. This case occurs when a vehicle has to change lane to follow the path leading to its destination (for example, to take an exit ramp, make a left turn, etc.). If the immediate situation prevents a vehicle from shifting into the desired lane, the vehicle will decelerate and prepare to stop before the downstream node to avoid missing its exit or moving into a wrong link.

Courtesy yielding. Courtesy yielding refers to the cases where a driver decelerates to make space for another vehicle shifting into its lane. The simulator assigns a probability of courtesy yielding to drivers. When a driver has decided to yield, the deceleration rate is calculated using the car-following model by treating the vehicle in the neighboring lane as the leader.

2.8.3. Lane changing

The lane changing model in MITSIM is implemented in three steps: (a) checking if a change is necessary and defining the type of change; (b) selecting the desired lane; and (c) executing the desired lane change if gap distances are acceptable. This model is based on Gipps (1986).

Checking the need and defining the type of the change. MITSIM classifies lane changes into mandatory and discretionary. Mandatory lane changing occurs when drivers have to change lanes in order to:

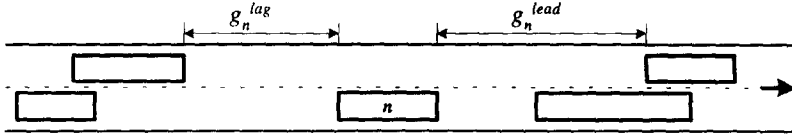


Fig. 5. Lead and lag gaps for lane changing.

- (a) connect to the next link on their path;
- (b) bypass a lane blockage downstream;
- (c) avoid entering a restricted use lane; or,
- (d) respond to LUS or VMS (e.g. warning of lane drop).

Discretionary lane changing refers to cases in which drivers change lane in order to increase speed, bypass a heavy vehicle, avoid the lane connected to a ramp, etc.

In the case of mandatory lane changing, a driver will start the lane change at a distance x_n from the downstream node (or incident, lane drop, red LUS) with probability:

$$p_n = \begin{cases} \exp[(x_n - x_0)^2 / \sigma_n^2] & x_n > x_0 \\ 1 & x_n \leq x_0 \end{cases} \quad (14)$$

where p_n is the probability that vehicle n starts a mandatory lane change maneuver; x_n is the distance from the downstream node or lane blockage; x_0 is a critical distance, which may be associated to the position of a particular message sign (such as final exit warning); and σ_n is a variable defined as follows:

$$\sigma_n = \alpha_0 + \alpha_1 m_n + \alpha_2 K \quad (15)$$

where m_n is the number of lanes that the vehicle needs to cross in order to be in the target lane; K is the traffic density of the segment; α_0 , α_1 , and α_2 are parameters. When a vehicle has been tagged with a status of mandatory lane change, it keeps that status until it has performed the desired lane change or moved into the downstream link.

For discretionary lane changing, the decision to change is based on traffic conditions of both the current lane and adjacent lanes. If a vehicle has a speed lower than the driver's desired speed due to a slow vehicle in front or the maximum speed of that lane, it checks the neighboring lanes for opportunities to increase its speed. Several parameters, such as an *impatient factor* and a *speed indifference factor*, are used to determine whether the current speed is low enough and the speeds in adjacent lanes are high enough for considering a lane change.

Selection of desired lane. When selecting the desired lane, the vehicle first determines the set of admissible lanes. A lane is defined as *admissible* based on several criteria including lane changing regulation, lane use privilege, lane connection, signal state and incident, prevailing traffic conditions, driver's desired speed, and lane's maximum speed V_i^{\max} .

Gap acceptance. Once a vehicle has decided to change lanes, it examines the lead and lag gaps in the target lane to determine whether the desired change can be executed. If both the lead and lag gaps are acceptable, the desired lane change is executed instantaneously.

For discretionary lane changes, the minimum acceptable gaps are given by (see Fig. 5):

$$\tilde{g}_n^i = \bar{g}^i + \tilde{\epsilon}_n^i \quad i = \text{lead, lag} \quad (16)$$

where \tilde{g}_n^i is the minimum gap that driver n considers to be acceptable for a discretionary lane change; \bar{g}^i is the average acceptable gap; and, $\tilde{\epsilon}_n^i$ is an error term. The parameter \bar{g}^i and the distribution of $\tilde{\epsilon}_n^i$ are provided in the parameter file for both the lead and lag gaps. Acceptable gaps for mandatory lane changes may decrease as the vehicle approaches the

downstream node (same for incidents and lane drops). In other words, it is assumed that drivers tend to accept smaller gaps as they get closer to the last location where the lane change has to take place. The current gap acceptance model for mandatory lane changes has the following form:

$$\tilde{g}_n^i = \tilde{\epsilon}_n + \begin{cases} \tilde{g}_{\max}^i & x_n \geq x_{\max} \\ \tilde{g}_{\min}^i + (\tilde{g}_{\max}^i - \tilde{g}_{\min}^i) \frac{(x_n - x_{\min})}{(x_{\max} - x_{\min})} & x_{\min} < x_n < x_{\max} \\ \tilde{g}_{\min}^i & x_n \leq x_{\min} \end{cases} \quad (17)$$

where i is an index indicating whether the parameter is for the *lead* or *lag* gap; \tilde{g}_n^i is the minimum gap driver n accepts for a mandatory lane change; \tilde{g}_{\min}^i and \tilde{g}_{\max}^i are respectively lower and upper bounds; x_n is the vehicle's current position; x_{\min} and x_{\max} are distances that define the range within which the critical gap varies from \tilde{g}_{\min}^i to \tilde{g}_{\max}^i , and $\tilde{\epsilon}_n^i$ is an error term.

2.9. Simulation Output

A typical output from MITSIM includes: (i) vehicle specific information, such as total travel time, miles traveled, and average speed (reported when vehicles reach their destination nodes); (ii) readings from traffic sensors, such as traffic counts, occupancy, speed, point to point travel times, vehicle classification, and incident information (reported at either a fixed frequency or when a vehicle passes a detector or incident); (iii) segment specific traffic data such as density, average speed, and travel times (reported at a fixed frequency or when simulation is completed); (iv) warnings and error messages of the simulation run; and, (v) graphical display of the vehicle movements and segment specific traffic data (e.g. average density and speed).

The collected disaggregate data on vehicle movements provide all the necessary information for development of measures of effectiveness (MOEs) useful for evaluation purposes (e.g. travel times, queue lengths, fuel consumption, emissions, safety, etc.).

3. IMPLEMENTATION

3.1. Hardware and software environment

MITSIM is implemented in C++ using the object-oriented programming paradigm. This provides the flexibility to easily modify and extend the code as the model is continuously enhanced and validated. The simulator code has been compiled using a number of compilers including the GNU C++ compiler and runs on HP, SGI, DEC, SUN and IBM workstations with UNIX and X Windows and PCs with Linux and XFree86.

The implementation of MITSIM does not impose restrictions on the number of network and vehicle objects. However, the size of the network and number of vehicles are limited by available computer memory. Furthermore, the running time increases as the size of network increases, and heavily depends on the number of vehicles that exist simultaneously in the simulated network.

3.2. Computational experience

Two test scenarios have been used to determine the capabilities of the simulator and its limits from a computational point of view. All experiments presented here were conducted on a SGI Indigo2 R4400 workstation. The simulation step sizes were 1 s for car-following and lane-changing and 0.1 s for updating of vehicles' position and speed (for more details, see Ben-Akiva *et al.*, 1996).

The first test was on an 8 km freeway consisting of 20 nodes, 19 links, 25 segments, 4-6 lanes per segment, and 94 loop detectors, with close to capacity travel demand. Simulating 2 h 20 min of operations on this small network, with a total of 22,000 vehicles, required 15 min of running time.

In the second test, a freeway network of 143 nodes, 171 links, and 700 lanes with a total length of approx. 170 lane-km was used. The simulator collected extensive data during the simulation (1.16 Gbytes) and performed faster than real time, with a time factor (execution time divided by simulated time) equal to 0.48.

3.3. User interface

A graphical user interface (GUI) is used to display network data (e.g. geometry and connectivity, link type, positions of surveillance and control devices), vehicle movements, traffic data (e.g. average speed and density of each segment), and operations of the traffic control and surveillance system (e.g. state of traffic signals and signs, detection of vehicle passage by detectors) during the simulation.

Use of the on-screen graphical display slows down the simulation, but it provides a very useful tool for checking the correctness of input data and visualizing the simulation process. During the development phase, the GUI has been used for detecting abnormal behavior and debugging the code.

Traffic network. The road network is color coded by direction, facility type (e.g. tunnels and no-tunnels), and average density or speed of segments. Lane marks are shown in colors to indicate the lane change regulations. Dedicated symbols drawn in the lane indicate the lane-use privilege. The network graphics are updated whenever a redraw command (e.g. zooming, panning, or changing display mode) is received. If the GUI is currently displaying information that changes dynamically (e.g. speed or density), the information are updated at a fixed frequency (e.g. 60 s).

Traffic sensors. The state of surveillance sensors (counts, average speed and occupancy) is displayed with colors and refreshed at fixed time intervals.

Traffic controls and incidents. The traffic control devices are shown by dedicated symbols indicating their current state. Incidents are also shown with dedicated symbols and are color coded by their state (active or cleared).

Animation of vehicle movements. Vehicles are shown as colored rectangles with dimensions proportional to their size. The following information can be displayed: *vehicle type* (class, information availability, lane-use privilege, etc), *car-following regime*, *speed*, *acceleration*, *lane changing*, and *turning movement* status. The user can also pause the simulation at any time to query information on individual vehicles.

3.4. Communication with supporting modules

The entire system (traffic simulator and other modules) is implemented in a distributed mode. A typical configuration uses two processes, one for MITSIM and the other for the traffic management simulator. A *master controller* is used to synchronize the execution of all modules. Information exchanges between the modules are performed via inter-processor communication. More specifically, at the beginning of each iteration, the simulator checks whether any message has been received from the traffic management simulator and updates the state of traffic control devices and incidents if necessary. At the end of each iteration, it checks whether the sensor readings, as well as the vehicle state vectors, should be sent to the traffic management simulator and other supporting modules running on the same or different processors. MITSIM uses PVM (Geist *et al.*, 1994) for the implementation of the communication functions.

4. EXAMPLE

In this section, we present some preliminary results from the application of MITSIM using a data set acquired from 10 detector stations on an 8 km stretch of I-880 around Hayward, California. This network has 4 on-ramps and 5 off-ramps, two of them connecting I-880 with SR-92 (see Fig. 6). The traffic counts and mean speeds aggregated over 10-min time intervals are available and used in this study.

Using the observed traffic counts and speeds during a 4-h time period for a number of days, time dependent O-D matrices were first estimated using the method of Ashok and

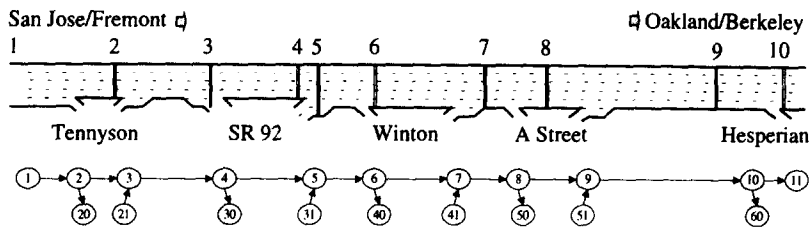


Fig. 6. An example freeway network.

Ben-Akiva (1993). Given the estimated O-D matrices, 5 simulation runs were conducted for a period of 2 h 20 min. Traffic counts and speeds were collected at each detector station for 10-min time intervals and averaged over the 5 simulation runs. The first 2 time intervals were treated as “warm-up” periods and excluded from the calculation of the statistics.

Compared with the field data, the root mean square error (RMSE) of the simulation output is 156 vph for the flows and 10.3 kph (5.8 mph) for the speeds. If the last two detector stations are excluded, the RMSE for speed is 8 kph (5 mph) compared to field data. Figures 7(a) and (b) show scatter plots of simulated and field data. The points in the scatter plot for traffic counts tend to distribute around the 45° diagonal line while the simulator tends to over estimates speeds at some points. Further analysis indicates that the simulated speeds compare poorly with the field data at the detector station near the Hesperian off-ramp (where the simulated speed is considerably higher than the observed field speed over the first 40 min). This is probably due to congestion occurring at

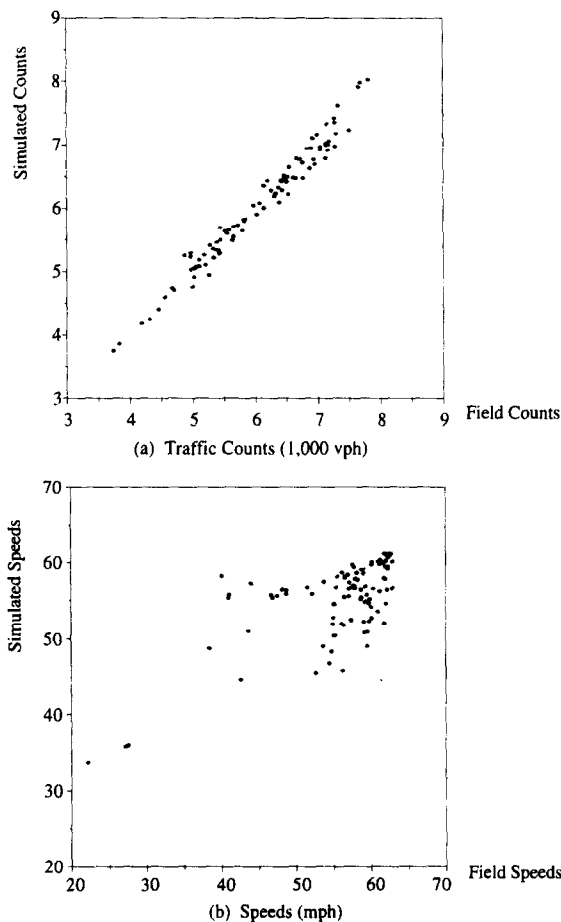


Fig. 7. Comparison between simulation output and field data.

the downstream section outside the boundary of the simulated network (and may explain some of the overestimated speeds in the scatter plot of Fig. 7).

We should point out that the above results were obtained using the default values for various parameters in the simulation model. The only calibration attempt, at the aggregate level, was made with respect to the parameters α^\pm , β^\pm and γ^\pm of the car-following model. The simpler version of the model (e.g. $\beta^\pm = 0$ and $\gamma^\pm = 1$) did not perform well. This was also the case when the values of the parameters suggested by May and Keller (1967) were used ($\beta^\pm = 0.8$ and $\gamma^\pm = 2.8$). Recently, Ozaki (1993) calibrated Herman's general car-following model using detailed microscopic data collected through an experiment involving three cars driving on a test track. He suggests different values of the parameters under acceleration and deceleration conditions ($\beta^- = 0.9$ and $\gamma^- = 1.0$ in the case of deceleration and $\beta^+ = 0.2$ and $\gamma^+ = -0.2$ in the case of acceleration). The importance of considering different parameters for acceleration and deceleration was verified by our analysis as well. The model provided a better representation of existing conditions for the following values of the parameters: $\alpha^+ = 0.50$, $\beta^+ = -1.00$, $\gamma^+ = -1.00$, and $\alpha^- = 1.25$, $\beta^- = 1.00$, $\gamma^- = 1.00$ (distance is measured in meters, speed in m/s, and acceleration in m/s²).

5. CONCLUSION

A microscopic traffic simulator supporting evaluation of advanced traffic surveillance and control systems is presented. This simulator models integrated traffic networks in detail and uses car following, lane changing, signal and event responding logic in modeling vehicle movements. The model supports distributed implementation and features a graphical user interface and communications with other modules such as traffic control and surveillance system simulators. MITSIM is currently undergoing extensive calibration and validation using detailed data on driver behavior and traffic conditions on various facilities (Ahmed *et al.*, 1996).

Acknowledgements—The authors would like to thank M. Ben-Akiva, T. Hotz, R. Mishalani, and the members of the ITS group at MIT for many useful discussions; M. Hickman, K. Petty and H. Noemi for providing the I-880 data set; and K. Ashok, who estimated the O-D matrices and helped in preparing the network for the case study.

REFERENCES

- Ahmed, K., Ben-Akiva, M., Koutsopoulos, H. N. and Mishalani, R. G. (1996) Models for freeway lane changing and gap acceptance behavior, *Proceedings of 13th International Symposium on Transportation and Traffic Theory*, Lyon France.
- Ahuja, R. K., Magnanti, T. L. and Orlin, J. B. (1994) *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall, Englewood Cliffs, New Jersey.
- Ashok, K. and Ben-Akiva, M. (1993) Dynamic O-D matrix estimation and prediction for real-time traffic management systems. In C. F. Daganzo (ed.), *Transportation and Traffic Flow Theory*, Elsevier, pp. 465–484.
- Ben-Akiva, M. E., Koutsopoulos, H. N., Mishalani, R. G. and Yang, Q. (1994) Integrated simulation framework for evaluating dynamic traffic management systems, *Proceedings of the First World Congress on Applications of Transport Telematics and Intelligent Transportation Systems*, Paris.
- Ben-Akiva, M. E., Koutsopoulos, H. N., Mishalani, R. G. and Yang, Q. (1996) Simulation laboratory for evaluating dynamic traffic management systems. *ASCE Journal in Transportation Engineering* (forthcoming).
- Cascetta, E., Nuzzolo, A., Russo, F. and Vitetta, A. (1996) A modified logit route choice model overcoming path overlapping problems: Specification and some calibration results for interurban networks, *Proceedings of 13th International Symposium on Transportation and Traffic Theory*, Lyon France.
- Codelli, D. K., Niedringhaus, W. P. and Wang, P. (1992) The Traffic and Highway Objects for REsearch, Analysis, and Understanding (THOREAU) IVHS model, Vol. I, *Technical Report MTR 92W208V1*, MITRE, McLean, Virginia.
- Dial, R. B. (1971) A probabilistic multipath traffic assignment algorithm which obviates path enumeration. *Transportation Research*, 5(2), 83–111.
- Eskafi, F., Khorramabadi, D. and Varaiya, R. (1996) An automated highway system simulator. *Transportation Research*, 3(1), 1–17.
- FHWA (1980) Development and testing of INTRAS: a microscopic freeway simulation model—Vol. 1–3, *Report FHWA/RD-80/106-108*, Federal Highway Administration, US-DOT, McLean, Virginia.
- FHWA (1994) FRESIM User Guide, *Technical Report Version 4.5*, Federal Highway Administration, US-DOT, McLean, Virginia.

- Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R. and Sunderam, V. (1994) *PVM: Parallel Virtual Machine — A Users' Guide and Tutorial for Networked Parallel Computing*. MIT Press, Cambridge, Massachusetts.
- Gipps, P. G. (1986) A model for the structure of lane changing decisions. *Transportation Research -B*, **20**(5), 403–414.
- Herman, R. and Rothery, R. W. (1963) Car-following and steady state flow, *Theory of Traffic Flow Symposium Proceedings*, pp. 1–13.
- Herman, R., Montroll, E. W., Potts, R. and Rothery, R. W. (1959) Traffic dynamics: Analysis of stability in car-following. *Operations Research*, **1**(7), 86–106.
- Institute of Transportation Engineers (1982) *Transportation and Traffic Engineering Handbook*, 2nd edn. Prentice-Hall Inc., Englewood Cliffs, N.J.
- Jayakrishnan, R., Mahamassani, H. and Hu, T. (1995) An evaluation tool for advanced traffic information and management systems in urban networks. *Transportation Research -C*, **2**(3), 129–147.
- Koutsopoulos, H. N. and Yang, Q. (1992) Modeling requirements for emerging transportation system operating environments, *Report DTRS-57-88-C-0078, TTD 29*, John A. Volpe National Transportation Systems Center, Cambridge.
- Mahmassani, H. and Jayakrishnan, R. (1991) System performance and user response under real-time information in a congested traffic corridor. *Transportation Research -A*, **25**(5), 293–307.
- May, A. and Keller, H. E. (1967) Non-integer car-following models. *Highway Research Board*, **199**, 19–32.
- Ozaki, H. (1993) Reaction and anticipation in the car-following behavior. In C. F. Daganzo, (ed.), *Transportation and Traffic Flow Theory*. Elsevier, pp. 346–349.
- Santiago, A. J. and Kanaan, A. (1993) ATMS laboratories: A requirement for program delivery, The 3rd Annual Meeting of IVHS America.
- Van Aerde, M. and Yagar, S. (1988) Dynamic integrated freeway/traffic signal networks: a routing-based modeling approach. *Transportation Research -A*, **22**(6), 445–453.
- Wicks, D. A. (1977) INTRAS—a microscopic freeway corridor simulation model. *Overview of simulation in highway transportation*, **1**, 95–107.