Back to the main publication page.

TRAFFIC SIMULATION

BY EDWARD LIEBERMAN[18]
AJAY K. RATHI

[18] President, KLD Associates, Inc. 300 Broadway, Huntington Station, NY 11746

CHAPTER 10 - Frequently used Symbols

a_f	=	acceleration response of follower vehicle to some stimulus
v	=	instantaneous speed of lead vehicle
v_f	=	instantaneous speed of follower vehicle
d	=	projected maximum deceleration rate of lead vehicle
d_f	=	projected maximum deceleration rate of follower vehicle
R_f	=	reaction time lag of driver in follower vehicle
I_i	=	*i*th replicate of specified seed, S_0
R_I	=	*i*th random number
h	=	headway separating vehicles (sec)
H	=	mean headway (sec)
h_{min}	=	minimum headway (sec)
R	=	random number
X_I	=	*i*th observation (sample) of an MOE
μ_x	=	mean of sample
δ^2	=	variance
$\hat{\delta}$	=	estimate of variance
$t_{n-1,1-\alpha/2}$	=	upper 1-α/2 critical point of the *t* distribution with *n*-1 degrees of freedom
$\overline{x}_j(m)$	=	mean of *m* observation of *j*th batch
$\overline{x}$	=	grand sample mean across batches
N_i	=	number of replications of the *i*th strategy
$var(X)$	=	variance of statistic, *X*

10.
TRAFFIC SIMULATION

10.1 Introduction

Simulation modeling is an increasingly popular and effective tool for analyzing a wide variety of dynamical problems which are not amenable to study by other means. These problems are usually associated with complex processes which can not readily be described in analytical terms. Usually, these processes are characterized by the interaction of many system components or *entities*. Often, the behavior of each entity and the interaction of a limited number of entities, may be well understood and can be reliably represented logically and mathematically with acceptable confidence. However, the complex, simultaneous interactions of many system components cannot, in general, be adequately described in mathematical or logical forms.

Simulation models are designed to "mimic" the behavior of such systems. Properly designed models *integrate* these separate entity behaviors and interactions to produce a detailed, quantitative description of system performance. Specifically, simulation models are mathematical/logical representations (or *abstractions*) of real-world systems, which take the form of software executed on a digital computer in an experimental fashion.

The user of traffic simulation software specifies a “scenario” (e.g., highway network configuration, traffic demand) as model inputs. The simulation model results *describe* system operations in two formats: (1) statistical and (2) graphical. The numerical results provide the analyst with detailed quantitative descriptions of *what* is likely to happen. The graphical and animated representations of the system functions can provide insights so that the trained observer can gain an understanding of *why* the system is behaving this way. However, it is the responsibility of the analyst to properly interpret the wealth of information provided by the model to gain an understanding of cause-and-effect relationships.

Traffic simulation models can satisfy a wide range of requirements:

1. *Evaluation of alternative treatments*
   With simulation, the engineer can control the experimental environment and the range of conditions to be explored. Historically, traffic simulation models were used initially to evaluate signal control strategies, and are currently applied for this purpose as an integral element of the ATMS research and development activity.

2. *Testing new designs*
   Transportation facilities are costly investments. Simulation can be applied to quantify traffic performance responding to different geometric designs before the commitment of resources to construction.

3. *As an element of the design process*
   The classical iterative design paradigm of conceptual design followed by the recursive process of evaluation and design refinement, can benefit from the use of simulation. Here, the simulation model can be used for evaluation; the detailed statistics provided can form the basis for identifying design flaws and limitations. These statistics augmented with animation displays can provide invaluable insights guiding the engineer to improve the design and continue the process.

4. *Embed in other tools*
   In addition to its use as a stand-alone tool, simulation *sub-models* can be integrated within software tools designed to perform other functions. Examples include: (1) the flow model within the TRANSYT-7F signal optimization; (2) the DYNASMART simulation model within a dynamic traffic assignment; (3) the simulation component of the INTEGRATION assignment/control model; (4) the CORSIM model within the Traffic Research Laboratory (TreL) developed for FHWA; and (5) the simulation module of the EVIPAS actuated signal optimization program.

5. *Training personnel*
   Simulation can be used in the context of a real-time laboratory to train operators of Traffic Management Centers. Here, the simulation model, which is integrated with a real-time traffic control computer, acts as a surrogate for the real-world surveillance, communication and traffic environments.

6. *Safety Analysis*
   Simulation models to “recreate” accident scenarios have proven to be indispensable tools in the search to build safer vehicles and roadways. An example is the CRASH program used extensively by NHTSA.

This compilation of applications indicates the variety and scope of traffic simulation models and is by no means exhaustive. Simulation models can also be *supportive* of analytical models such as PASSER, and of computational procedures such as the HCS. While these and many other computerized tools do not include simulation sub-models, users of these tools can enhance their value by applying simulation to evaluate their performance.

This chapter is intended for transportation professionals, researchers, students and technical personnel who either currently use simulation models or who wish to explore their potential. Unlike the other chapters of this monograph, we will not focus exclusively on theoretical developments -- although fundamental simulation building blocks will be discussed. Instead, we will describe the properties, types and classes of traffic simulation models, their strengths and pitfalls, user caveats, and model-building fundamentals. We will emphasize how the user can derive the greatest benefits from simulation through proper interpretation of the results, with emphasis on the need to adequately calibrate the model and to apply rigorous statistical analysis of the results.

10.2 When Should the Use of Simulation Models be Considered?

Since simulation models *describe* a dynamical process in statistical and pictorial formats., they can be used to analyze a wide range of applications wherever...

- Mathematical treatment of a problem is infeasible or inadequate due to its temporal or spatial scale, and/or the complexity of the traffic flow process.
- The assumptions underlying a mathematical formulation (e.g., a linear program) or an heuristic procedure (e.g., those in the Highway Capacity Manual) cast some doubt on the accuracy or applicability of the results.
- The mathematical formulation represents the dynamic traffic/control environment as a simpler quasi steady-state system.
- There is a need to view vehicle animation displays to gain an understanding of *how* the system is behaving in order to explain *why* the resulting statistics were produced.
- Congested conditions persist over a significant time.

It must be emphasized that traffic simulation, by itself, cannot be used *in place of* optimization models, capacity estimation procedures, demand modeling activities and design practices. Simulation can be used to *support* such undertakings, either as embedded submodels or as an auxiliary tool to evaluate and extend the results provided by other procedures. Some representative statistics (called Measures of Effectiveness, MOE) provided by traffic simulation models are listed in Table 10.1.

Such statistics can be presented for each specified highway section (network link) and for each specified time period, to yield a level of detail that is both spatially and temporally disaggregated. Aggregations of these data, by subnetwork and network-wide, and over specified time periods, may also be provided.

10.3 Examples of Traffic Simulation Applications

Given the great diversity of applications that are suitable for the use of traffic simulation models, the following limited number of examples provides only a limited representation of past experience.

10.3.1 Evaluation of Signal Control Strategies

This study (Gartner and Hou, 1992) evaluated and compared the performance of two arterial traffic control strategies,

Table 10.1
Simulation Output Statistics: Measures of Effectiveness

Measure for Each Link and for Entire Network	
Travel: Vehicle-Miles	Bus Travel Time
Travel Time: Vehicle-minutes	Bus Moving Time
Moving Time: Vehicle-minutes	Bus Delay
Delay Time: Vehicle-minutes persons-minutes	Bus Efficiency: Moving Time Total Travel Time
Efficiency: Moving Time Total Travel Time	Bus Speed
Mean Travel Time per Vehicle-Mile	Bus Stops
Mean Delay per Vehicle-Mile	Time bus station capacity exceeded
Mean Travel Time per vehicle	Time bus station is empty
Mean Time in Queue	Fuel consumed
Mean Stopped Time	CO Emissions
Mean Speed	HC Emissions
Vehicle Stops	NOX Emissions
Link Volumes Occupancy	
Mean Link Storage Area Consumed	
Number of Signal Phase Failures	
Average Queue Length	
Maximum Queue Length	
Lane Changes	
Bus Trips	
Bus Person Trips	

MULTIBAND and MAXBAND, employing the TRAF-NETSIM simulation model. The paper describes the statistical analysis procedures, the number of simulation replications executed and the resulting 95 percent confidence intervals, and the results of the analysis.

Figure 10.1 is taken from this paper and illustrates how simulation can provide objective, accurate data sufficient to distinguish between the performance of alternative analytical models, within the framework of a controlled experiment.

10.3.2 Analysis of Equilibrium Dynamic Assignments (Mahmassani and Peeta, 1993)

This large-scale study used the DYNASMART simulation-assignment model to perform both user equilibrium (UE) and system optimal (SO) equilibrium calculations for a specified network, over a range of traffic loading conditions from unsaturated to oversaturated. This is an example of traffic simulation used as a component of a larger model to perform a complex analysis of an ITS initiative.

Figure 10.2 which is taken from the cited paper illustrates how simulation can produce internally consistent results for large scale projects, of sufficient resolution to distinguish between two comparable equilibrium assignment approaches.

10.3.3 Analysis of Corridor Design Alternatives (Korve Engineers 1996)

This analysis employed the WATSim simulation model to evaluate alternative scenarios for increasing capacity and improving traffic flow on a freeway connection, SR242, in

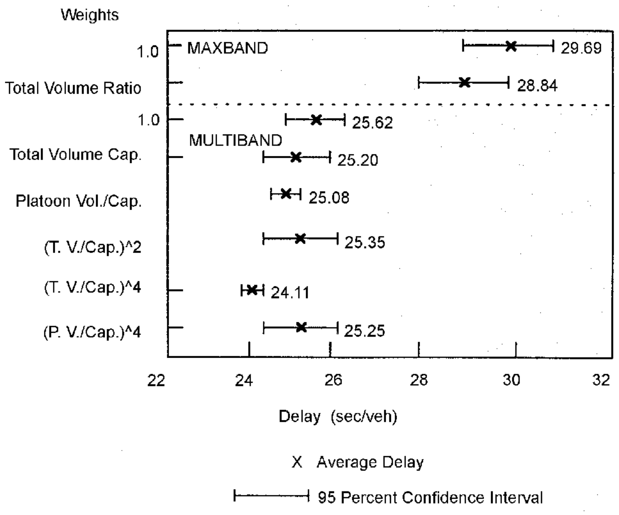


Figure 10.1
Average Delay Comparison, Canal Street, MULTIBAND & MAXBAND (KLD-242).

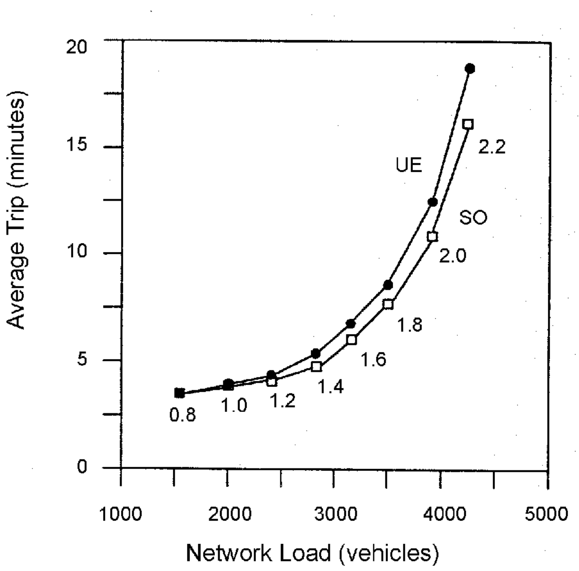


Figure 10.2
Comparison of average trip times (minutes) of SO & UE...(KLD-243).

California and ensuring a balanced design relative to freeway SR4 on the north and I680 to the south. Design alternatives considered for three future periods (years 2000, 2010, 2020) included geometric changes, widening, HOV lanes and ramp metering. Given the scale of this 20-mile corridor and the strong interactions of projected design changes for the three highways, the use of simulation provided a statistical basis for quantifying the operational performance of the corridor sections for each alternative.

This example illustrates the use of simulation as an element of the design process with the capability of analyzing candidate designs of large-scale highway systems in a manner that lies beyond the capabilities of a straight-forward HCM analysis.

10.3.4 Testing New Concepts

The TRAF-NETSIM simulation model was used by Rathi and Lieberman (1989) to determine whether the application of metering control along the periphery of a congested urban area could mitigate the extent and duration of congestion within the area, thereby improving performance and productivity. Here, a new control concept was tested on a real-world test-bed: a section of Manhattan. This example illustrates the value in testing new “high risk” ideas with simulation without exposing the public to possible adverse consequences, and prior to expending resources to implement these concepts.

These examples certainly do not represent the full range of traffic simulation applications. Yet, they demonstrate the application of traffic simulation in the areas of (1) traffic control; (2) transportation planning; (3) design; and (4) research.

10.4 Classification of Simulation Models

Almost all traffic simulation models describe *dynamical* systems -- *time* is always the basic independent variable. *Continuous* simulation models describe how the elements of a system change *state* continuously over time in response to continuous stimuli. *Discrete* simulation models represent real-world systems (that are either continuous or discrete) by asserting that their *states* change abruptly at points in time. There are generally two types of discrete models:

- Discrete time
- Discrete event

The first, segments time into a succession of known time intervals. Within each such interval, the simulation model computes the *activities* which change the *states* of selected system elements. This approach is analogous to representing an initial-value differential equation in the form of a finite-difference expression with the independent variable, Δt.

Some systems are characterized by entities that are "idle" much of the time. For example, the state of a traffic signal indication (say, green) remains constant for many seconds until its state changes *instantaneously* to yellow. This abrupt change in state is called an *event*. Since it is possible to accurately describe the operation of the signal by recording its changes in state as a succession of [known or computed] timed events, considerable savings in computer time can be realized by only executing these events rather than computing the state of the signal second-by-second. For systems of limited size or those representing entities whose states change infrequently, discrete event simulations are more appropriate than are discrete time simulation models, and are far more economical in execution time. However, for systems where most entities experience a continuous change in state (*e.g.*, a traffic environment) and where the model objectives require very detailed descriptions, the discrete time model is likely to be the better choice.

Simulation models may also be classified according to the level of detail with which they represent the system to be studied:

- Microscopic (high fidelity)
- Mesoscopic (mixed fidelity)
- Macroscopic (low fidelity)

A *microscopic* model describes both the system entities and their interactions at a high level of detail. For example, a lane-change maneuver at this level could invoke the car-following law for the subject vehicle with respect to its current leader, then with respect to its putative leader *and* its putative follower in the target lane, as well as representing other detailed driver decision processes. The duration of the lane-change maneuver can also be calculated.

A *mesoscopic* model generally represents most entities at a high level of detail but describes their activities and interactions at a much lower level of detail than would a microscopic model. For example, the lane-change maneuver could be represented for individual vehicles as an instantaneous event with the decision based, say, on relative lane densities, rather than detailed vehicle interactions.

A *macroscopic* model describes entities and their activities and interactions at a low level of detail. For example, the traffic stream may be represented in some aggregate manner such as a statistical histogram or by scalar values of flow rate, density and speed. Lane change maneuvers would probably not be represented at all; the model may assert that the traffic stream is properly allocated to lanes or employ an approximation to this end.

High-fidelity microscopic models, and the resulting software, are costly to develop, execute and to maintain, relative to the lower fidelity models. While these detailed models possess the *potential* to be more accurate than their less detailed counterparts, this potential may not always be realized due to the complexity of their logic and the larger number of parameters that need to be calibrated.

Lower-fidelity models are easier and less costly to develop, execute and to maintain. They carry a risk that their representation of the real-world system may be less accurate, less valid or perhaps, inadequate. Use of lower-fidelity simulations is appropriate if:

- The results are not sensitive to microscopic details.
- The scale of the application cannot accommodate the higher execution time of the microscopic model.
- The available model development time and resources are limited.

Within each level of detail, the developer has wide latitude in designing the simulation model. The developer must identify the sensitivity of the model's performance to the underlying features of the real-world process. For example, if the model is to be used to analyze weaving sections, then a detailed treatment of lane-change interactions would be required, implying the need for a micro- or mesoscopic model. On the other hand, if the model is designed for freeways characterized by limited merging and no weaving, describing the lane-change interactions in great detail is of lesser importance, and a macroscopic model may be the suitable choice.

Another classification addresses the processes represented by the model: (1) Deterministic; and (2) Stochastic. *Deterministic* models have no random variables; all entity interactions are defined by exact relationships (mathematical, statistical or logical). *Stochastic* models have processes which include probability functions. For example, a car-following model can be formulated either as a deterministic or stochastic relationship by defining the driver's reaction time as a constant value or as a random variable, respectively.

Traffic simulation models have taken many forms depending on their anticipated uses. Table 10.2 lists the TRAF family of models developed for the Federal Highway Administration (FHWA), along with other prominent models, and indicates their respective classifications. This listing is necessarily limited. Some traffic simulation models consider a single facility (NETSIM, NETFLO 1 and 2: surface streets; FRESIM, FREFLO: freeways; ROADSIM: two-lane rural roads; (CORSIM) integrates two other simulation models, FRESIM and NETSIM; INTEGRATION, DYNASMART, TRANSIMS are components of larger systems which include demand models and control policies; while CARSIM is a stand-alone simulation of a car-following model. It is seen that traffic simulation models take many forms, each of which satisfies a specific area of application.

Table 10.2
Representative Traffic Simulation Models

Name	Discrete Time	Discrete Event	Micro	Mesoscopic	Macro	Deterministic	Stochastic
NETSIM	X		X				X
NETFLO 1		X		X			X
NETFLO 2	X				X	X	
FREFLO	X				X	X	
ROADSIM	X		X				X
FRESIM	X		X				X
CORSIM	X		X				X
INTEGRATION	X		X				X
DYNASMART	X			X		X	
CARSIM	X		X				X
TRANSIMS	X			X			X

10.5 Building Traffic Simulation Models

The development of a traffic simulation model involves the following activities:

1) Define the Problem and the Model Objectives
   - State the purpose for which the model is being developed.
   - Define the information that the model must provide.

2) Define the System to be Studied
   - Disaggregate the system to identify its major components.
   - Define the major interactions of these components.
   - Identify the information needed as inputs.
   - Bound the domain of the system to be modeled.

3) Develop the Model
   - Identify the level of complexity needed to satisfy the stated objectives.
   - Classify the model and define its inputs and outputs.
   - Define the flow of data within the model.
   - Define the functions and processes of the model components.
   - Determine the calibration requirements and form: scalars, statistical distributions, parametric dependencies.
   - Develop abstractions (*i.e.*, mathematical-logical-statistical algorithms) of each major system component, their activities and interactions.
   - Create a logical structure for integrating these model components to support the flow of data among them.
   - Select the software development paradigm, programming language(s), user interface, presentation formats of model results.
   - Design the software: simulation, structured or object-oriented programming language; database, relational/object oriented.
   - Document the logic and all computational procedures.
   - Develop the software code and debug.

4) Calibrate the Model
   - Collect/acquire data to calibrate the model.
   - Introduce this data into the model.

5) Model Verification
   - Establish that the software executes in accord with the design specification.
   - Perform verification at the model component level.

6) Model Validation
   - Collect, reduce, organize data for purposes of validation.
   - Establish that the model describes the real system at an acceptable level of accuracy over its entire domain of operation; apply rigorous statistical testing methods.

7) Documentation
   - Executive Summary
   - Users Manual
   - Model documentation: algorithms and software

The development of a traffic simulation model is not a "single-pass" process. At each step in the above sequence, the analyst must review the activities completed earlier to determine whether a revision/extension is required before proceeding further. For example, in step 5 the analyst may verify that the software is replicating a model component properly as designed, but that its performance is at a variance with theoretical expectations or with empirical observations. The analyst must then determine whether the calibration is adequate and accurate (step 4); whether the model's logical/mathematical design is correct and complete (step 3); whether all interactions with other model components are properly accounted for and that the specified inputs are adequate in number and accuracy (step 2). This continual feedback is essential; clearly, it would be pointless to proceed with validation (step 6) if it is *known* that the verification activity is incomplete.

Step 3 may be viewed as the most creative activity of the development process. The simulation logic must represent all relevant interactions by suitably exploring the universe of possibilities and representing the likely outcome. These combinations of interactions are called *processes* which represent specified *functions* and utilize *component models*. A small sample of these is presented below.

10.5.1 Car-Following

One fundamental interaction present in all microscopic traffic simulation models is that between a leader-follower pair of vehicles traveling in the same lane. This interaction takes the form of a stimulus-response mechanism:

$$a_f = F\ (v_l,v_f,s,d_l,d_f,R_f,P_i) \tag{10.1}$$

where a_f, the acceleration (response) of the follower vehicle, is dependent on a number of (stimulus) factors including:

v_l, v_f = Speeds of leader, follower vehicles, respectively.
s = Separation distance.
d_l, d_f = Projected deceleration rates of the leader, follower vehicles, respectively.
R_f = Reaction time of the driver in the following vehicle.
P_i = Other parameters specific to the car-following model.
$F(\bullet)$ = A mathematical and logical formulation relating the response parameter to the stimulus factors.

This behavioral model can be referenced (i.e., executed) to support other behavioral models such as lane-changing, merging, etc.

10.5.2 Random Number Generation

All stochastic models must have the ability to generate random numbers. Generation of random numbers has historically been an area of interest for researchers and practitioners. Before computers were invented, people relied on mechanical devices and their observations to generate random numbers. While numerous methods in terms of computer programs have been devised to generate random numbers, these numbers only "appear" to be random. This is the reason why some call them *pseudo-random* numbers. The most popular approach for random number generation is the "linear congruential method" which employs a recursive equation to produce a sequence of random integers *S* as:

$$S_i = (aS_{i-1} + b) \bmod c.$$

where the integers chosen are defined as,
c is the modulus, such that $c > 0$,
a is the multiplier such that $0 < a < c$,
b is the increment such that $0 < b < m$, and
S_o is the starting value or the *Seed* of the random number generator, such that $0 < S_o < c$.

The *i*th random number denoted by R_i is then generated as

$$R_i = \frac{S_i}{c}.$$

These random number generations are typically used to generate random numbers between 0 and 1. That is, a Uniform (0,1) random number is generated. *Random variates* are usually referred to as the sample generated from a distribution other than the Uniform (0,1). More often than not, these random variates are generated from the Uniform (0,1) random number. A simulation usually needs random variates during its execution. Based on the distribution specified, there are various analytical methods employed by the simulation models to generate the random variates. The reader is referred to Law and Kelton (1991) or Roberts (1983) for a detailed treatment on this topic. As an example, random variates in traffic simulation are used to generate a stream of vehicles.

10.5.3 Vehicle Generation

At the outset of a simulation run, the system is "empty". Vehicles are generated at origin points, usually at the periphery of the analysis network, according to some headway distribution based on specified volumes. For example, the shifted negative exponential distribution will yield the following expression:

$$h = (H - h_{\text{min}})\ [-\ \ln(1-R)] + H - h_{\text{min}}$$

where

h = Headway (sec) separating vehicle emissions

H = Mean headway = 3600/V, where V is the specified volume, vph

h_{min} = Specified minimum headway (e.g., 1.2 sec/veh)

R = Random number in the range (0 to 1.0), obtained from a pseudo-random number generator.

Suppose the specified volume, V (vph), applies for a 15-minute period. If the user elects to *guarantee* that V is explicitly satisfied by the simulation model, it is necessary to generate N values of h using the above formula repeatedly, generating a new random number each time. Here, $N = V/4$ is the *expected* number of vehicles to be emitted in 15 minutes. The model could then calculate the factor, K:

$$K = \frac{15 x 60}{\sum_{i=1}^{N} h_i} \qquad (10.2)$$

The model would then multiply each of the N values of h_i, by K, so that the resulting sum of (the revised) h_i will be exactly 15 minutes, ensuring that the user's specification of demand volume are satisfied. However, if $K \neq 1.0$, then the resulting distribution of generated vehicles is altered and one element of stochasticity (*i.e.*, the actual number of generated vehicles) is removed. The model developer must either include this treatment (*i.e.*, eq.10.2), exclude it; or offer it as a user option with appropriate documentation.

10.5.4 A Representative Model Component

Consider two *elements* of every traffic environment: (1) a vehicle and (2) its driver. Each element can be defined in terms of its relevant *attributes*:

Vehicle: Length; width; acceleration limits; deceleration limit; maximum speed; type (auto, bus, truck, ...); maximum turn radius, etc.

Driver: Aggressiveness; responsiveness to stimuli; destination (route); other behavioral and decision processes.

Each attribute must be represented by the analyst, some by scalars (e.g., vehicle length); some by a functional relationship (e.g., maximum vehicle acceleration as a function of its current speed); some by a probability distribution (e.g., driver gap acceptance behavior). All must be calibrated.

The driver-vehicle combination forms a model component, or entity. This component is defined in terms of its own elemental attributes and its functionality is defined in terms of the interactions between these elements. For example, the driver's decision to accelerate at a certain rate may be constrained by the vehicle's operational limitations. In addition, this system component interacts with other model entities representing the environment under study, including:

- roadway geometrics
- intersection configurations
- nearby driver-vehicle entities
- control devices
- lane channelization
- conflicting vehicle movements

As an example, the driver-vehicle entity's interaction with a control device depends on the type and current state of the device (*e.g.*, a signal with a red indication), the vehicle's speed, its distance from stop-bar, the driver's aggressiveness, etc. It is the developer's responsibility to design the model components and their interactions in a manner that satisfies the model objectives and is consistent with its fidelity.

10.5.5 Programming Considerations

Programming languages, in the context of this chapter, may be classified as *simulation* and *general-purpose* languages. Simulation languages such as SIMSCRIPT and GPSS/H greatly ease the task of developing simulation software by incorporating many features which compile statistics and perform queuing and other functions common to discrete simulation modeling.

General-purpose languages may be classified as *procedural* (*e.g.*, FORTRAN, PASCAL, C, BASIC), or *object-oriented* (*e.g.*, SMALLTALK, C++, JAVA). Object-oriented languages are gaining prominence since they support the concept of reusable software defining *objects* which communicate with one another to solve a programming task. Unlike procedural languages, where the functions are separated from, and operate upon, the data base, objects *encapsulate* both data describing its state, as well as operations (or "methods") which can change its state and interact with other objects.

While object-oriented languages can produce more reliable software, they require a higher level of programming skill than do procedural languages. The developer should select a language which is hardware independent, is supported by the major operating systems and is expected to have a long life, given the rapid changes in the world of software engineering. Other factors which can influence the language selection process include: (1) the expected life of the simulation model; (2) the skills of the user community; (3) available budget (time and resources) to develop and maintain the software; and (4) a realistic assessment of available software development skills.

10.6 An Illustration of Simulation Model Building

Given the confines of this chapter, we will illustrate the model development process by presenting the highlights of a sample problem, but avoiding exhaustive detail.

1) ***Define the Problem and Model Objectives*** - An existing microscopic stochastic simulation model of freeway traffic does not consider lane-change operations. It has been determined that this model's results are unreliable as a result. The purpose of this project is to introduce additional logic into the model to represent lane-changing operations. This addition should provide improved accuracy in estimating speed and delay; in addition it will compute estimates of lane changes by lane, by vehicle type and by direction (to the left and to the right).

2) ***Define the System -***

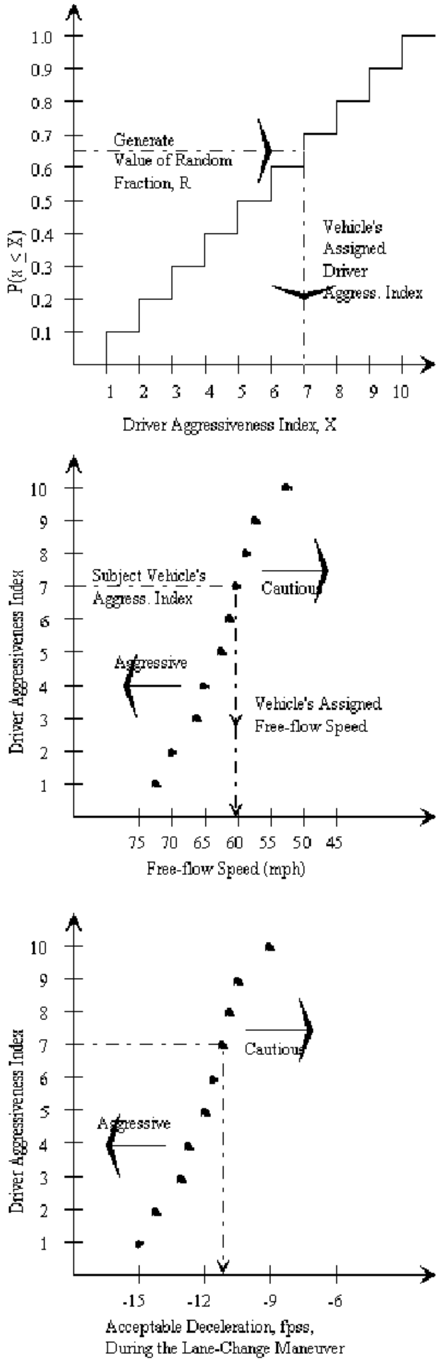
   a) A freeway of up to six lanes -- level tangent
   b) Three vehicle types: passenger car; single-unit truck; tractor-trailer truck
   c) Required inputs: traffic volume (varies with time); distributions of free-speed, of acceptable risk (expressed in terms of deceleration rates if lead vehicle brakes), of motivation to change lanes, all disaggregated by vehicle type
   d) Drivers are randomly assigned an "aggressiveness index" ranging from 1 (very aggressive) to 10 (very cautious) drawn from a uniform distribution to represent the range of human behavior.

   It must be emphasized that model development is an *iterative* process. For example, the need for the indicated input distributions may not have been recognized during this definition phase, but may have emerged later during the logical design. Note also that the problem is bounded -- no grades or horizontal curves are to be considered at this time. See Figure 10.3 for the form of these distributions.

3) ***Develop the Model*** - Since this lane-change model is to be introduced into an existing microscopic stochastic model, using a procedural language, it will be designed to utilize the existing software. The model logic moves each vehicle, each time-step, Δt, starting with the farthest downstream vehicle, then moving the closest upstream vehicle regardless of lane position, etc. At time, t_o, the vehicle *states* are shown in Figure 10.4(a).

   In developing the model, it is essential to identify the independent *functions* that need to be performed and to segregate each function into a separate software module, or routine. Figure 10.5 depicts the *structure* -- not the flow -- of the software. This structure shows which routines are logically connected, with data flowing between them. Some routines reference others more than once, demonstrating the benefits of disaggregating the software into functionally independent modules.

   As indicated in Table 10.3 which presents the algorithm for the Lane Change Executive Routine in both "Structured English" or "pseudo-code" and as a flow chart, traffic



Note:
Random-number generator produces a value, R, for each emitted vehicle. Application of the *inverse transformation method* to this distribution assigns a value of X to the vehicle, defining the driver's aggressiveness.

(a) Discretized Uniform Distribution

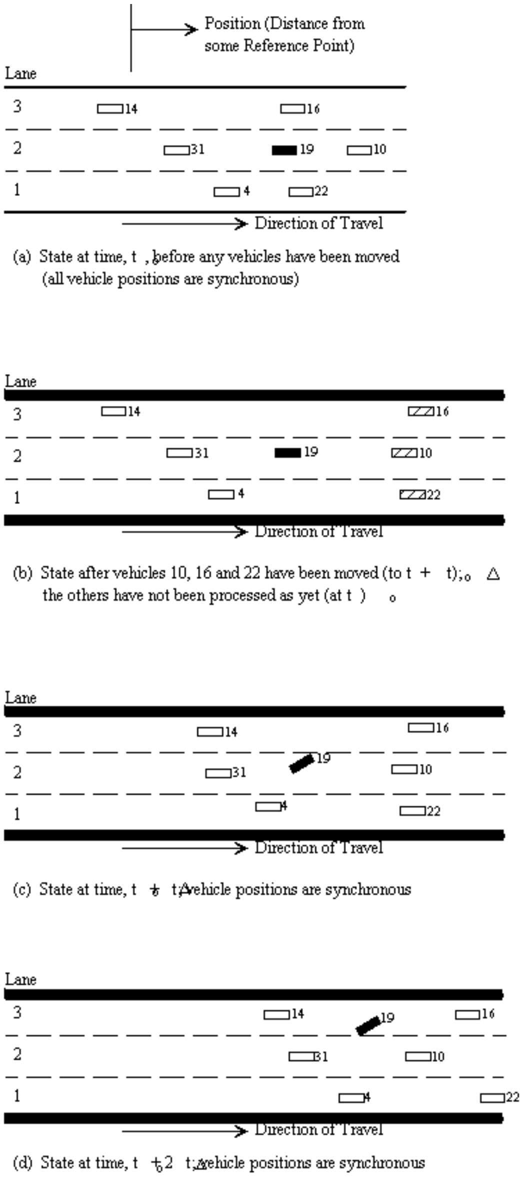
Note:
This distribution implies that aggressive drivers select higher free speeds

(b) Discretized Distribution relating Driver Aggressiveness Index to Desired Free-flow Speed

Note:
This distribution implies that aggressive drivers accept higher risks -- they are willing to accept situations that could require high deceleration to avoid a collision

(c) Discretized Distribution relating Driver Aggressiveness Index to Acceptable Deceleration during Lane-Change Maneuver

Figure 10.3
Several Statistical Distributions.



(a) State at time, t_o, before any vehicles have been moved (all vehicle positions are synchronous)

(b) State after vehicles 10, 16 and 22 have been moved (to $t_o + \Delta t$); the others have not been processed as yet (at t_o)

(c) State at time, $t_o + \Delta t$; vehicle positions are synchronous

(d) State at time, $t_o + 2\Delta t$; vehicle positions are synchronous

	Longitudinal/Lane Position at time		
Veh. ID	t_o	$t + \Delta t$	$t_o + 2\Delta t$
10	216/2	181/2	145/2
16	240/3	169/3	119/3
22	243/1	176/1	111/1
19	250/2	216/2+3	174/3
4	276/1	230/1	184/1
31	301/2	255/2	208/2
14	343/3	278/3	215/3

□ Vehicle position at indicated times

■ Subject (lane-change) vehicle position at indicated times

▨ Vehicle position, one time-step, Δt, into the future, relative to indicated time

Figure 10.4
Vehicle Positions during Lane-Change Maneuver

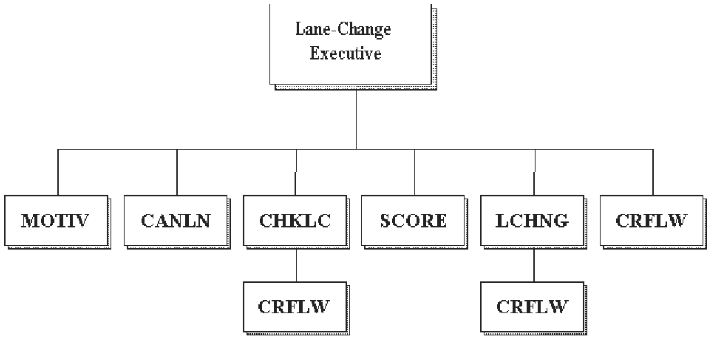


Figure 10.5
Structure Chart of Simulation Modules

Table 10.3
Executive Routine

```
For each vehicle, I:
   CALL routine MOTIV to determine whether this driver is “motivated” to change lanes, now
   IF so, THEN
      CALL routine CANLN to identify which of neighboring lanes (if either) are
             acceptable as potential target lanes
      IF the lane to the right is acceptable, THEN
         CALL routine CHKLC to determine whether a lane-change is feasible, now.
             Set flag if so.
      ENDIF
      IF the lane to the left is acceptable, THEN
             CALL routine CHKLC to determine whether a lane-change is feasible, now.
               Set flag, if so.
      ENDIF
      IF both lane-change flags are set (lane-change is feasible in either direction), THEN
             CALL routine SCORE to determine more favorable target lane
      ELSE IF one lane-change flag is set, THEN
             Identify that lane
      ENDIF
      IF a [favored] target lane exists, THEN
             CALL routine LCHNG to execute the lane-change
             Update lane-change statistics
      ELSE
             CALL routine CRFLW to move vehicle within this lane
             Set vehicle’s process code (to indicate vehicle has been moved this time-step)
      ENDIF
   ELSE (no lane-change desired)
      CALL routine CRFLW to move vehicle within its current lane
      Set vehicle’s process code
   ENDIF
continue...
```

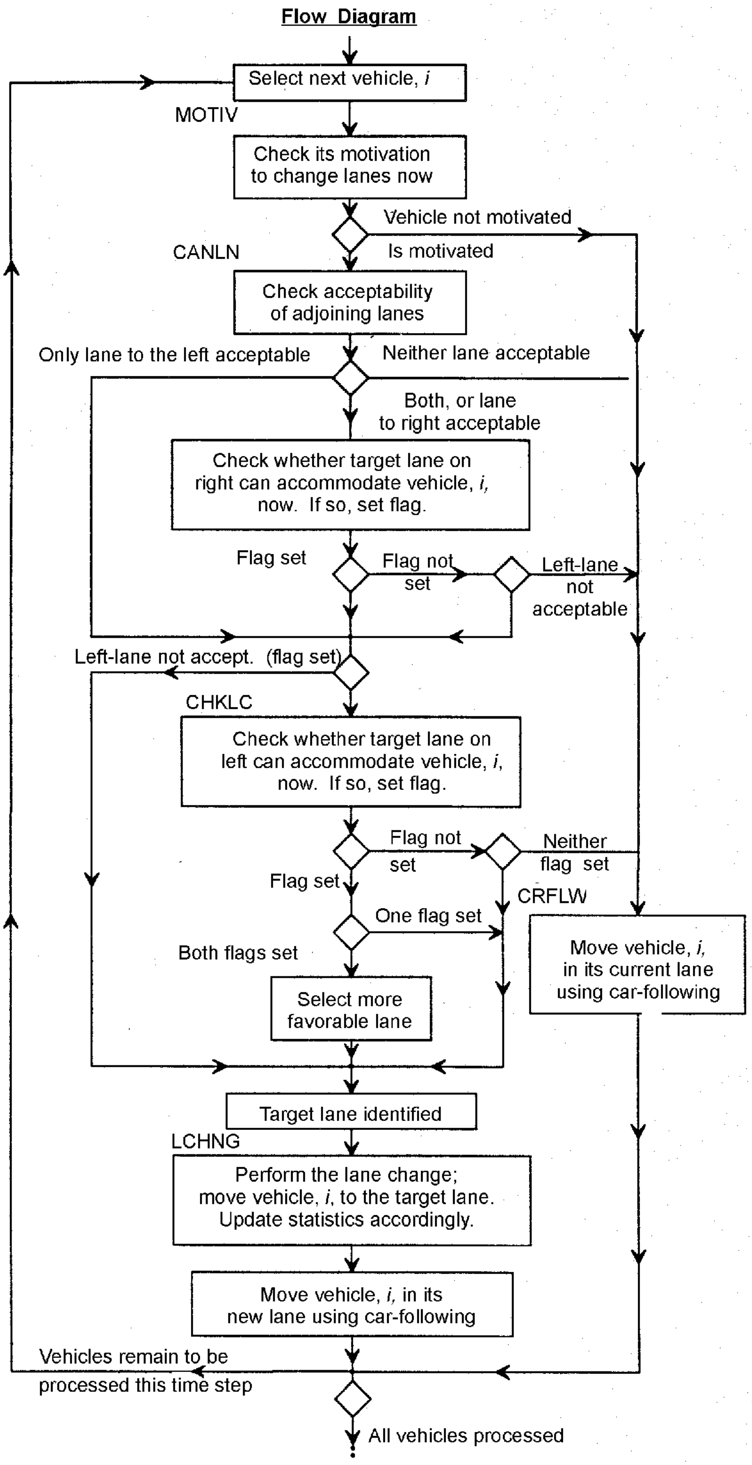

Flow Diagram
Select next vehicle, i
MOTIV
Check its motivation to change lanes now
Vehicle not motivated
Is motivated
CANLN
Check acceptability of adjoining lanes
Only lane to the left acceptable
Neither lane acceptable
Both, or lane to right acceptable
Check whether target lane on right can accommodate vehicle, i, now. If so, set flag.
Flag set
Flag not set
Left-lane not acceptable
Left-lane not accept. (flag set)
CHKLC
Check whether target lane on left can accommodate vehicle, i, now. If so, set flag.
Flag not set
Neither flag set
Flag set
CRFLW
One flag set
Both flags set
Move vehicle, i, in its current lane using car-following
Select more favorable lane
Target lane identified
LCHNG
Perform the lane change; move vehicle, i, to the target lane. Update statistics accordingly.
Move vehicle, i, in its new lane using car-following
Vehicles remain to be processed this time step
All vehicles processed

Figure 10.6
Flow Diagram.

simulation models are primarily *logical*, rather than *computational* in context. This property reflects the fact that traffic operations are largely the outcome of driver decisions which themselves are [hopefully!] logical in context. As the vehicles are processed by the model logic, they transition from one *state* to the next. The reader should reference Table 10.3 and Figures 10.3, 10.4, 10.5 and 10.6 to follow the discussion given below for each routine. The "subject vehicle" is shown as number 19 in Figure 10.4.

Executive: Controls the flow of processing, activating (through CALLs) routines to perform the necessary functions. Also updates lane-change statistics.

MOTIV: Determines whether a lane-change is *required* to position the subject vehicle for a downstream maneuver or is *desired* to improve the vehicle's operation (increase its speed).

CANLN: Determines whether either or both adjoining lanes (1 and 3) are suitable for servicing the subject vehicle.

CHKLC: Identifies vehicles 22 and 4 as the leader and follower, respectively, in target Lane 1; and vehicles 16 and 14, respectively, in target Lane 3. The car-following dynamics between the pairs of vehicles, 19 and 22, then 4 and 19 are quantified to assess the prospects for a lane-change to Lane 1. Subsequently, the process is repeated between the pairs of vehicles, 19 and 16; then 14 and 19; for a lane-change to Lane 3. If the gap is inadequate in Lane 1, causing an excessive, and possibly impossible deceleration by either the subject vehicle, 19, or the target follower, 4, to avoid a collision, then Lane 3, would be identified as the only feasible target lane. In any case, CHKLC would identify either Lane 1 or Lane 3, or both, or neither, as acceptable target lanes at this time, depending on safety considerations.

SCORE: If both adjoining target lanes are acceptable, then this *heuristic* algorithm emulates driver reasoning to select the preferable target lane. It is reasonable to expect that the target lane with a higher-speed leader and fewer vehicles -- especially trucks and buses -- would be more attractive. Such *reasonableness* algorithms (which are expressed as "rules" in expert systems), are also common in models which simulate human decisions.

LCHNG: After executing the subject vehicle's lane-change activity, the logic performs some needed bookkeeping:

At time, $t_o + \Delta t$, vehicle 19 acts as the leader for both vehicles 14 and 31, who must "follow" (and are constrained by) its presence.

At time, $t_o + 2\Delta t$, the logic asserts that the lane changer (no. 19) has committed to the lane-change and no longer influences its former follower, vehicle no. 31. Of course, vehicle 14 now follows the lane-change vehicle, 19.

4) ***Calibrate the Model*** - Figures 10.3(b) and (c) are distributions which represent the outcome of a calibration activity. The distribution of free-flow speed is site-specific and can be quantified by direct observation (using paired loop detectors or radar) when traffic conditions are light -- LOS A.

The distribution of acceptable decelerations would be very difficult to quantify by direct observation -- if not infeasible. Therefore, alternative approaches should be considered. For example...

- Gather video data (speeds, distance headway) of lane-change maneuvers. Then apply the car-following model with these data to "back-out" the implied acceptable decelerations. From a sample of adequate size, develop the distribution.

- On a more macro level, gather statistics of lane-changes for a section of highway. Execute the simulation model and adjust this distribution of acceptable deceleration rates until agreement is attained between the lane-changes executed by the model, and those observed in the real world. This is tenuous since it is confounded by the other model features, but may be the best viable approach.

It is seen that calibration -- the process of quantifying model parameters using real-world data -- is often a difficult and costly undertaking. Nevertheless, it is a *necessary* undertaking that must be pursued with some creativity and tenacity.

5) ***Model Verification*** - Following *de-bugging*, verification is a structured regimen to provide assurance that the software performs as intended (Note: verification does not address the question, "Are the model components and their interactions *correct*?"). Since simulation models are primarily *logical* constructs, rather than computational ones, the analyst must perform detailed logical path analyses.

Verification is performed at two levels and generally in the sequence given below:

- Each software routine (bottom-up testing)
- Integration of "trees" (top-down testing)...

When completed, the model *developer* should be convinced that the model is performing in accord with expectations over its entire domain of application.

6) ***Model Validation*** - Validation establishes that the model behavior accurately and reliably represents the real-world system being simulated, over the range of conditions anticipated (i.e., the model's "domain"). Model validation involves the following activities:

- Acquiring real-world data which, to the extent possible, extends over the model's domain.

- Reducing and structuring these empirical data so that they are in the same format as the data generated by the model.

- Establishing validation criteria, stating the underlying hypotheses and selecting the statistical tests to be applied.

- Developing the experimental design of the validation study, including a variety of "scenarios" to be examined.

- Performing the validation study:
  - Executing the model using input data and calibration data representing the real-world conditions.
  - Performing the hypothesis testing.

- Identifying the causes for any failure to satisfy the validation tests and repairing the model accordingly.

- Validation should be performed at the component system level as well as for the model as a whole. For example, Figure 10.7 compares the results produced by a car-following model, with field data collected with aerial photographs. Such *face* validation offers strong assurance that the model is valid. This validation activity is *iterative* -- as differences between the model results and the real-world data emerge, the developer must "repair" the model, then revalidate. Considerable skill (and persistence) are needed to successfully validate a traffic simulation model.

7) ***Documentation*** - Traffic simulation models, as is the case for virtually all transportation models, are *data intensive*. This implies that users must invest effort in data acquisition and input preparation to make use of these models. Consequently, it is *essential* that the model be documented for...

- The end user, to provide a "friendly" interface to ease the burden of model application.
- Software maintenance personnel.
- Supervisory personnel who must assess the potential benefits of using the model.

10.7 Applying Traffic Simulation Models

Considerable skill and attention to detail must be exercised by the user in order to derive accurate and reliable results from a simulation-based analysis. The following procedure is recommended:

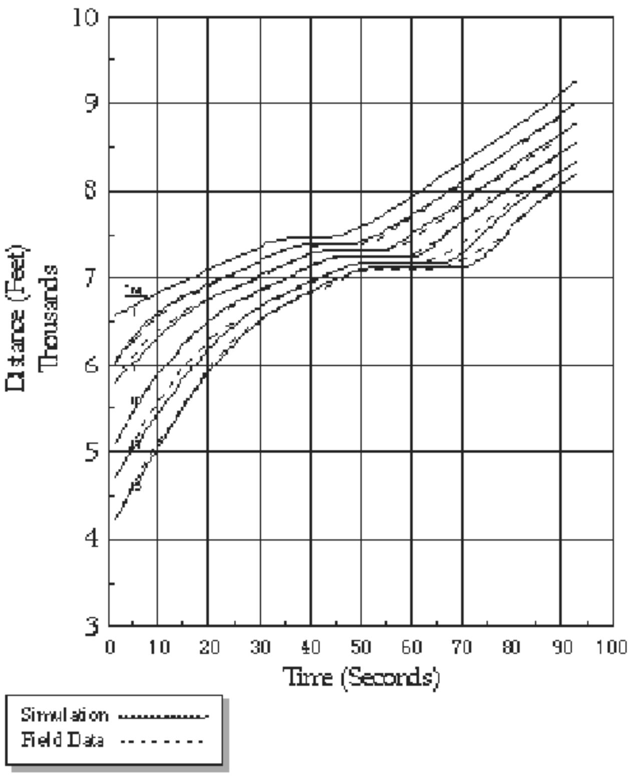


Figure 10.7
Comparison of Trajectories of Vehicles from Simulation Versus Field Data for Platoon 123.

Identify the Problem Domain

What highway facilities are involved?

- Surface streets (grid, arterial, both), freeways, rural roads, toll plaza.

What is the traffic environment?

- Autos, trucks, buses, LRT, HOV...
- Unsaturated, oversaturated conditions.

What is the control environment?

- Signals (fixed time, actuated, computer-controlled), signs
- Route guidance

What is the size of the network and duration of the analysis period?

Define the Purpose of the Study

What information is needed from the simulation model?

- Statistical: MOE sought, level of detail.
- Pictorial: static graphics, animation.

What information is available as input and calibration data?

- Consider expected accuracy and reliability.
- Consider available budget for data acquisition.

Are results needed on a *relative* or *absolute* basis?

What other functions and tools are involved?

- Capacity analysis
- Design
- Demand modeling
- Signal optimization

Is the application real-time or off-line?

Investigate Candidate Traffic Simulation Models

Identify strengths and limitations of each.

- Underlying assumptions

- Computing requirements
- Availability, clarity, completeness of documentation
- Availability, reliability, timeliness of software support

Estimate extent and cost of data collection for calibration and input preparation.

Determine whether model features match problem needs.

Assess level of skill needed to properly apply model.

Determine compatibility with other tools/procedures needed for the analysis.

Assess the Need to Use a Traffic Simulation Model

Is traffic simulation necessary to perform the analysis of the problem?

- Are other tools adequate but less costly?
- Are your skills adequate to properly apply simulation?
- Can the data needed by traffic simulation be acquired?

Is traffic simulation highly beneficial even though not necessary?

- Simulation results can confirm results obtained by other tools.
- Animation displays needed as a presentation medium.

If it is determined that traffic simulation is needed/advisable, continue.

Select Traffic Simulation Model

Relate relevant model attributes to problem needs.

Determine which model satisfies problem needs to the greatest extent. Consider technical, and cost, time, available skills and support, and risks factors.

Data Acquisition

Obtain reliable records of required information.

- Design drawings for geometrics.
- Signal timing plans, actuated controller settings.
- Traffic volume and patterns; traffic composition.
- Transit schedules
- Other, as required.

Confirm the accuracy of these data through field observation.

Undertake field data collection for input and for calibration, as required.

Identify need for accurate operational traffic data: based on model's sensitivity site-specific features; accuracy requirements.

- Select representative locations to acquire these field data.
- Collect data using video or other methods as required: saturation flow rates at intersections; free-flow speeds; acceptable gaps for permitted left-turns, etc.
- Accept model default values or other data from the literature with great care if data collection is infeasible or limited by cost considerations.

Model Calibration

Calibration is the activity of specifying data describing traffic operations and other features that are site-specific. These data may take the form of scalar elemets and of statistical distributions that are referenced by the logic of stochastic simulation models. While traffic simulation models generally provide default value which represent average conditions for these calibration data elements, it is the responsibility of the analyst to quantify these data with field observations to the extent practicable rather than to accept these default values.

Model Execution

The application of a simulation model should be viewed as performing a rigorous statistical experiment. The model must first be executed to *initialize* its database so that the data properly represents the initial state of the traffic environment. This requirement can reliably be realized if the environment is initially at equilibrium.

Thus, to perform an analysis of congested conditions, the analyst should design the experiment so that the initial state of the traffic environment is undersaturated, and then specify the changing conditions which, over time, censors the congested state which is of interest. Similarly, the final state of the traffic environment should likewise be undersaturated, if feasible.

It is also essential that the analyst properly specify the dynamic (*i.e.*, changing) input conditions which describe the traffic environment. For example, if one-hour of traffic is to be simulated, the analyst should always specify the variation in demand volumes -- and in other variables -- over that hour at an appropriate level of detail rather than specifying average, constant values of volume.

Finally, if animation displays are provided by the model, this option should always be exercised, as discussed below.

Interpretation of Simulation Results

Quite possibly, this activity may be the most critical. It is the analyst who must determine whether the model results constitute a reasonable and valid representation of the traffic environment under study, and who is responsible for any inferences drawn from these results. Given the complex processes taking place in the real-world traffic environment, the analyst must be alert to the possibility that (1) the model's features may be deficient in adequately representing some important process; (2) the input data and/or calibration specified is inaccurate or inadequate; (3) the results provided are of insufficient detail to meet the project objectives; (4) the statistical analysis of the results are flawed (as discussed in the following section); or (5) the model has "bugs" or some of its algorithms are incorrect. Animation displays of the traffic environment (if available) are a most powerful tool for analyzing simulation results. A *careful and thorough* review of this animation can be crucial to the analyst in identifying:

- Cause-and-effect relationships. Specifically the *origins* of congested conditions in the form of growing queues can be observed and related to the factors that caused it.
- Anomolous results (e.g., the creation and growth of queues when conditions are believed to be undersaturated) can be examined and traced to valid, incongested behavior; to errant input specifications; or to model deficiencies.

If the selected traffic simulation lacks an animation feature or if questions remain after viewing the animation, then the following procedures may be applied:

- Execute the model to replicate existing real-world conditions and compare its results with observed behavior. This "face validation", which is recommended regardless of the model selected, can identify model or implementation deficiencies.

- Perform "sensitivity" tests on the study network by varying key variables and observing model responses in a carefully designed succession of model executions.
- Plot these results. A review will probably uncover the perceived anomalies.

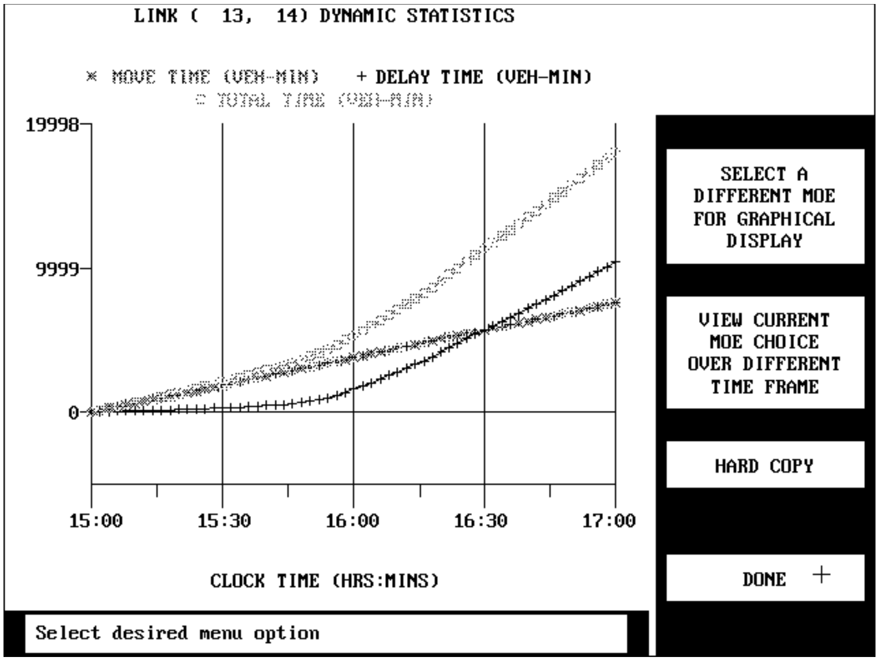
Table 10.1 lists representative data elements provided by traffic simulation models. Figure 10.8 shows typical graph displays while Figure 10.9 displays a "snapshot" of an animation screen.

Note that all the graphical displays can be accessed interactively by the user, thus affording the user an efficient means for extracting the sought *information* and *insights* from the mass of *data* compiled by the simulation model.

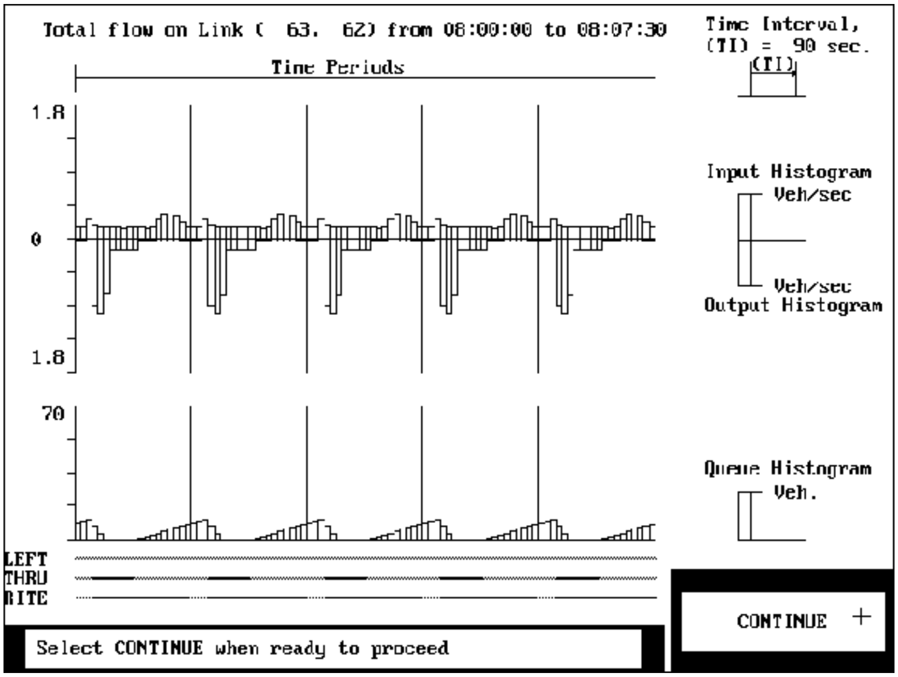
Proper output analysis is one of the most important aspects of any simulation study. A variety of techniques are used, particularly for stochastic model output, to arrive at inferences that are supportable by the output. A brief exposition to output analysis of simulation data is paresented next.

10.8 Statistical Analysis of Simulation Data

In most efforts on simulation studies, more often than not a large amount of time and money is spent on model development and coding, but little on analyzing the output data. Simulation practitioners therefore have more confidence in their results than is justified. Unfortunately, many simulation studies begin with a heuristic model, which is then programmed on the computer, and conclude with a single run of the program to yield *an answer*. This is a result of overlooking the fact that a simulation is a sampling experiment implemented on a computer and therefore needs appropriate statistical techniques for its design,

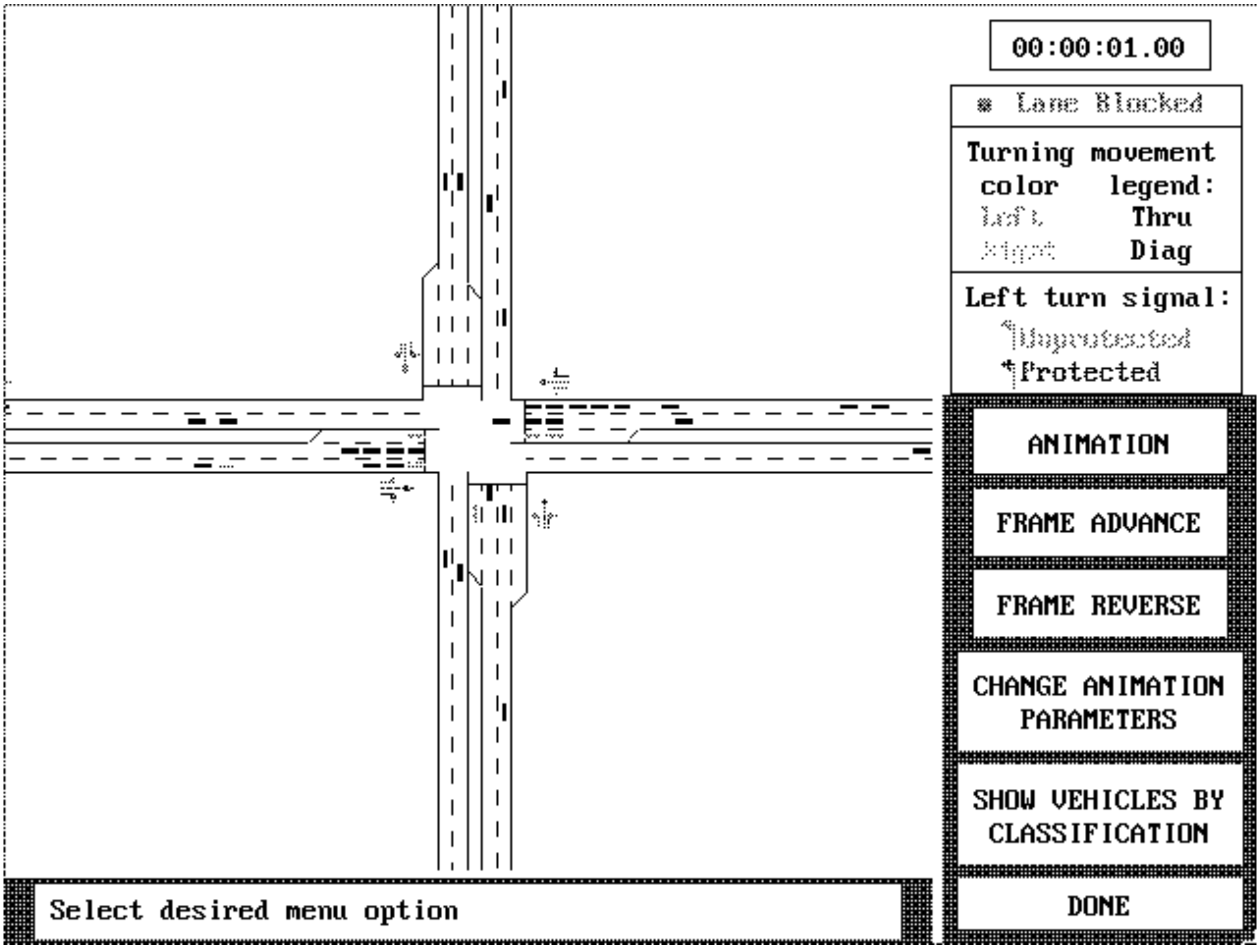


Select desired menu option



Select CONTINUE when ready to proceed

Figure 10.8
Graphical Displays.



Select desired menu option

Figure 10.9
Animation Snapshot**.**

analysis, and implementation. Also, more often than not, output data from simulation experiments are auto correlated and nonstationary. This precludes its analysis using classical statistical techniques which are based on independent and identically distributed (IID) observations.

Typical goals of analyzing output data from simulation experiments are to present point estimates of the measures of effectiveness (MOE) and form confidence intervals around these estimates for one particular system design, or to establish which simulated system is the best among different alternate

Back to the main publication page.

configurations with respect to some specified MOE. Point estimates and confidence intervals for the MOEs can be obtained either from one simulation run or a set of replications of the system using independent random number streams across replications.

10.9 Looking to the Future

With the traffic simulation models now mounted on high-performance PCS, and with new graphical user interfaces (GUI) becoming available to ease the burden of input preparation, it is reasonable to expect that usage of these models will continue to increase significantly over the coming years.

In addition, technology-driven advances in computers, combined with the expanding needs of the Intelligent Transportation Systems (ITS) program, suggest that the new applications of traffic simulation can contribute importantly to this program. Specifically:

- ■ Simulation support systems of Advanced Traffic Management Systems (ATMS) in the form of:

  - Off-line emulation to test, refine, evaluate, new real-time control policies.
  - On-line support to evaluate candidate ad-hoc responses to unscheduled events and to advise the operators at the Traffic Management Center (TMC) as to the "best" response. -Real-time component of an advanced control/guidance strategy. That is, the simulation model would be a component of the on-line strategy software.
  - Combining simulation with Artificial Intelligence (AI) software. The simulation can provide the knowledge base in real-time or generate it in advance as an off-line activity.
  - Integrating traffic simulation models with other tools such as: transportation demand models, signal optimization models, GIS, office suites, etc.
  - Providing Internet access.

- ■ Real-time simulators which replicate the performance of TMC operations. These simulators must rely upon simulation models as "drivers" to provide the real-world stimuli to ITS real-time software being tested. Such simulators are invaluable for:

  - Testing new ATMS concepts prior to deployment.
  - Testing interfaces among neighboring TMCs.
  - Training TMC operating personnel.
  - Evaluating different ATMS architectures.
  - Educating practitioners and student.
  - Demonstrating the benefits of ITS programs to state and municipal officials and to the public through animated graphical displays and virtual reality.

References

Gartner, N.H. and D. L. Hou (1992). *Comparative Evaluation of Alternative Traffic Control Strategies,*" Transportation Research Record 1360, Transportation Research Board.

Mahmassani, H.S. and S. Peeta (1993). *Network Performance Under System Optimal and User Equilibrium Dynamic Assignments: Implications for Advanced Traveler Information Systems* Transportation Research Record 1408, Transportation Research Board.

Korve Engineers (1996). *State Route 242 Widening Project - Operations Analysis Report* to Contra Costa Transportation Authority.

Rathi, A.K. and E.B. Lieberman (1989). *Effectiveness of Traffic Restraint for a Congested Urban Network: A Simulation Study*. Transportation Research Record 1232.