MICROSCOPIC TRAFFIC SIMULATION: A TOOL FOR THE ANALYSIS AND ASSESSMENT OF ITS SYSTEMS

J. Barceló
Department of Statistics and Operations Research
Universitat Politècnica de Catalunya (UPC)
Scientific Director of TSS-Transport Simulation Systems
barcelo@tss-bcn.com http://www.tss-bcn.com

Abstract
The need to simulate a number of advanced ITS concepts prior to deployment has also necessitated development of high performance microscopic simulators for estimating dynamic traffic assignment, freeway corridor diversion, as well as evaluating driver information systems (including variable message signs), vehicle guidance systems, real time adaptive traffic control strategies and other traffic management concepts. The high performance microscopic simulators capable of achieving such objectives should meet a set of basic requirements regarding the accuracy and performance of the traffic modeling, the ability to deal with the dynamic effects of time dependent traffic demands and, consequently, the time dependent route choices from each origin to each destination. This necessitates employment of behavioral models emulating the route choice processes. Finally, a practical simulator must have easy to use graphical interface allowing employment of all complex modeling components such as graphical editors for the geometry, interactive specification of control strategies, animated output and so on. In this paper a microsimulator specifically designed for assessing ITS systems deployment is presented. The simulator, called AIMSUN2, has undergone significant improvements since its inception in 1987 and it has been used in a number of real life projects. This has resulted in many modeling and functional improvements presented here, as well as acceptance for simulating large scale urban and freeway networks in Europe, Canada and Australasia. The methodology for model implementation is also described.

INTRODUCTION
The deployment of Intelligent Traffic Systems (ITS) requires support of complementary studies clearly showing the feasibility of the systems and what benefits should be expected from their operation. The large investments required have to be justified in a robust way. That means feasibility studies that validate the proposed systems, assess their expected impacts and provide the basis for sound cost benefit analyses. Although there are no doubts that ITS represents the future, and that ITS deployment will dominate the road transport scene in the forthcoming years, there are still some shadow areas concerning the proper implementation in each case, questions on how the users will react and decisions on the best way of operating the systems that need to be answered properly before proceeding to physical installations of ITS.

Microscopic traffic simulation has proven to be a useful tool to achieve these objectives. This is not only due to its ability to capture the full dynamics of time dependent traffic phenomena, but also being capable of dealing with behavioral models accounting for drivers' reactions when exposed to ITS systems. Microscopic traffic simulators are simulation tools that emulate realistically the flow of vehicles on a road network. The origins of microscopic traffic simulation can be traced back to the early stages of digital computers. Although the basic principles were set up many years ago, with the seminal work of, among others, Robert Hermann and the General Motors Group in the early fifties, the computing requirements
made them impractical until hardware and software developments made them affordable even on today’s laptop computers.

The advent of ITS has created new objectives and requirements for micro-simulation models. Quoting from Deliverable D3 of the European Commission Project SMARTEST [1]: “The objective of micro-simulation models is essentially, from the model designer’s point of view, to quantify the benefits of Intelligent Transportation Systems (ITS), primarily Advanced Traveller Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS). Micro-simulation is used for evaluation prior to or in parallel with on-street operation. This covers many objectives such as the study of dynamic traffic control, incident management schemes, real-time route guidance strategies, adaptive intersection signal controls, ramp and mainline metering, etc. Furthermore some models try to assess the impact and sensitivity of alternative design parameters”.

The main traditional basic components of a microscopic traffic simulation model are:

- An accurate representation of the road network geometry
- A detailed modeling of individual vehicles’ behavior
- And an explicit reproduction of traffic control plans.
- Animated output of the simulation runs has proven to be not only a highly desirable feature but also a powerful analysis tool of the simulation results.

To deal with ITS systems, as for example adaptive traffic control systems, automatic incident detection systems, dynamic vehicle guidance systems, advanced traffic management systems, and so on, an advanced microscopic traffic simulator needs also:

- To have the ability to interact on line with external real time applications, as for example SCOOT, SCATS, Real-Time Ramp Metering, or actuated systems (with the ability to preempting public transport) as C-Regelaar in The Netherlands or BALANCE in Germany, among others, in the case of real-time adaptive traffic control systems, or with real-time automatic incident detection algorithms in the case of incident management systems.
- To model explicitly public transport.
- To be able of modeling the Advanced Traffic management Strategies, as for example, re-routings, speed and lane control, etc. based on the use of VMS panels.
- To provide the tools for modeling dynamic vehicle guidance systems, emulating and tracking the equipped vehicles to suitably reproduce the exchange of data and information between the guided vehicle and the Traffic Information Center.

The recent evolution of the microscopic simulators has taken advantages of the state-of-the-art in the development of object-oriented simulators, and graphical user interfaces [2], as well as the new trends in software design and the available tools that support it adapted to traffic modeling requirements. A proper achievement of the basic requirements of a microscopic simulator implies building models as close to the reality as possible. The closer the model is to reality the more data demanding it become. This has been traditionally the main barrier preventing wider use of microscopic simulation. Manual coding of geometric data, turnings movements at intersections, timings and so on, is not only cumbersome and time
consuming but also a potential source of errors. It is also hard to debug if the appropriate tools are not available.

A way of overcoming these drawbacks has been to provide the microsimulators with the proper user friendliness based on the versatility of traffic network graphical editors, which can import the geometric background of the road network to draw the network model on top, as shown on the left part of Figure 1. The background can be imported as a .dxf file from a CAD or GIS system. All objects comprising the road model can be built with the graphic editor. Their attributes and parameters are defined and assigned values by means of windows dialogues such as the one in the right part of figure 1, which shows the definition of the shared movements in a phase of a pre-timed signal control, and the allocation of the timings. Summarising, these software environments for traffic modeling make an easy task of the model building process, ensure accurate geometry, prevent errors and provide powerful debugging tools.

Figure 1: Example of graphic user interface for building microscopic simulation models.

AIMSUN2 (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks) is a microscopic simulator purposely designed and developed at the Laboratorio de Investigación Operativa y Simulación, a research group in the Department of Statistics and Operations Research of the Universitat Politècnica de Catalunya, bearing in mind the requirements of the ITS systems. AIMSUN2 is imbedded in GETRAM (Generic Environment for TRaffic Analysis and Modeling), a simulation environment inspired by modern trends in the design of graphical user interfaces adapted to traffic modelling requirements. References can be found in http://www.tss-bcn.com
Once the simulation model of a basic scenario has been built, before being used for sophisticated applications the microscopic traffic simulator has to prove that it can reproduce to an acceptable degree of significance the observed traffic conditions or, in other words, that it is capable of emulating the reality with enough accuracy. The calibration and validation of the simulator are the required proofing exercises.

A very good example of a data collection process for the calibration and validation of a traffic simulation can be found in [3].

**BACKGROUND**

Most of the currently existing microscopic traffic simulators are based on the family of car-following, lane changing and gap acceptance models to model the vehicle’s behavior. Among the most still used models one can find Helly’s model (4), implemented in SITRA-B+ (5), Herman’s model (6), or its improved version by Wicks (7), implemented in MITSIM, (8), the psycho-physical model of Wiedemann, (9), used in VISSIM (10), or the ad hoc version of Gipps (11), used in AIMSUN2 (12, 13). Other microscopic simulators such as INTEGRATION (14) and PARAMICS employ heuristic or other modeling not publicly available in analytic form.

Helly’s model uses the following expression for the acceleration of the follower car $\ddot{x}_{n+1}(t + T)$:

$$\ddot{x}_{n+1}(t + T) = c_1 \left[ \dot{x}_n(t) - \dot{x}_{n+1}(t) \right] + c_2 \left[ x_n(t) - x_{n+1}(t) - D \right]$$

where $T$ is the reaction time for the vehicle-driver system, $c_1$ and $c_2$ are the relative velocity and headway control parameters and $D$ the desired headway, typically expressed as: $D = l_n + \tau_{n+1} \dot{x}_{n+1}(t)$ with $l_n$ being the length of the leader vehicle $n$ and $\tau_{n+1}$ the time headway for the follower $n+1$.

Herman’s model assumes an acceleration rate given by

$$\ddot{x}_{n+1}(t) = \alpha^+ \frac{\dot{x}_{n+1}(t)}{g_{n+1}^1(t)} (\dot{x}_n(t) - \dot{x}_{n+1}(t))$$

where $\alpha^+, \beta^+$ and $\gamma^+$ are model parameters $\alpha^+, \beta^+, \gamma^+$ are used for acceleration ($\dot{x}_{n+1}(t) \leq \dot{x}_n(t)$), and $\alpha^-, \beta^-, \gamma^-$ for deceleration ($\dot{x}_{n+1}(t) > \dot{x}_n(t)$), and $g_{n+1} = x_{n+1} - x_n - l_n$ represents the gap distance from the leading vehicle.

The Gipps model consists of two components: acceleration and deceleration. The first represents the intention of a vehicle to achieve certain desired speed, while the second reproduces the limitations imposed by the preceding vehicle when trying to drive at the desired speed. This model states that, the maximum speed at which a vehicle $(n)$ can accelerate during a time period $(t, t+T)$ is given by:

$$V_a(n, t + T) = V(n, t) + 2.5a(n)T \left( 1 - \frac{V(n, t)}{V^*(n)} \right) \left[ 0.025 + \frac{V(n, t)}{V^*(n)} \right]$$

where: $V(n, t)$ is the speed of vehicle $n$ at time $t$; $V^*(n)$ is the desired speed of the vehicle $(n)$; $a(n)$ is the maximum acceleration for vehicle $n$; $T$ is the reaction time.

On the other hand, the maximum speed that the same vehicle $(n)$ can reach during the same time interval $(t, t+T)$, according to its own characteristics and the limitations imposed by the presence of the leader vehicle is:
where: \( d(n) \) (< 0) is the maximum deceleration desired by vehicle \( n \); \( x(n,t) \) is position of vehicle \( n \) at time \( t \); \( x(n-1,t) \) is position of preceding vehicle \( (n-1) \) at time \( t \); \( s(n-1) \) is the effective length of vehicle \( (n-1) \); \( d'(n-1) \) is an estimation of vehicle \( (n-1) \) desired deceleration. The final speed for vehicle \( n \) during time interval \((t, t+T)\) is the minimum of those previously defined speeds:

\[
V(n, t + T) = \min \left\{ V_x(n, t + T), V_y(n, t + T) \right\}
\]

The position of vehicle \( n \) inside the current lane is updated by taking the speed into the movement equation:

\[
x(n, t + T) = x(n, t) + V(n, t + T)T
\]

A common drawback of most of these models is that the model parameters are global i.e. constant for the entire network whereas it is well known that driver’s behavior is affected by traffic conditions. Therefore a more realistic car-following modeling for microscopic simulation should account for local behavior. That implies that some of the model parameters must be local depending on local geometric and traffic conditions.

**MODELING IN AIMSUN2**

The AIMSUN2 car following model evolved after the seminal Gipps model, which was improved to meet the local requirements, described earlier. Three main aspects of the model have been enhanced based on the empirical evidence gathered calibrating the model for observed data:

- The way in which is calculated the vehicle speed \( V^*(n) \) used in the Gipps model
- How vehicles in adjacent lanes influence vehicle’s behavior
- Accounting for grade effects in car-following

**Speed calculation**

The first improvement is related to the vehicle speed \( V^*(n) \) used in equation (1). In AIMSUN2 implementation \( V^*(n) \) is the desired speed of vehicle \( n \) for the current section. In car-following a leading vehicle, attempts to drive to its maximum desired speed. Three parameters are used to calculate the maximum speed of leading vehicle \( n \) while driving on a particular section or turning:

1. Maximum desired speed of \( n \): \( \nu_{\text{max}}(n) \). This is a vehicle parameter.
2. Speed acceptance of \( n \): \( \theta(n) \) This is a vehicle parameter measuring the driver’s degree of compliance of the speed limits on the section. \( \theta(n) = 1 \), represents the perfect compliance. \( \theta(n) < 1 \), a driver driving below the speed limits, and \( \theta(n) > 1 \), faster than the speed limits. \( \theta(n) \) is a vehicle parameter that in AIMSUN2 can be sampled from a probability distribution, when such information is available, modeling implicitly in that way the aggressivity of the drivers.
3. Speed limit in section or turning \( s \): \( S_{\text{lim},n}(s) \). This is a section parameter.

The actual speed limit for a vehicle \( n \) on a section or turning \( s \), \( S_{\text{lim},n}(n, s) \), is given by:

\[
S_{\text{lim},n}(n, s) = S_{\text{lim},n}(s) \cdot \theta(n)
\]
The maximum desired speed of vehicle $n$ on a section or turning $s$, $v_{\text{max}}(n,s)$ is:

$$v_{\text{max}}(n,s) = \text{MIN}(s_{\text{lim}r}(n,s), v_{\text{max}}(n))$$

Thus the local maximum desired speed $v_{\text{max}}(n,s)$ equals the desired speed $V^*(n)$ in Eq. (1).

**Influence of adjacent lanes**

When the leading vehicle is driving along a section, the AIMSUN2 car-following model takes into account the potential influence of certain number of vehicles ($N_{\text{vehicles}}$) driving slower in the adjacent right-side lane—or left-side lane, when driving on the left—of the vehicle in the adjacent slower lane ($\text{MeanSpeedVehiclesDown}$). Only vehicles within a certain distance ($\text{MaximumDistance}$) from the current vehicle are taken into account. We distinguish two cases: 1) the adjacent lane is an on-ramp, or acceleration lane, and 2) the adjacent lane is any other type of lane. Apart from $N_{\text{vehicles}}$ and $\text{MaximumDistance}$ parameters, the user can define two additional parameters, $\text{MaximumSpeedDifference}$ and $\text{MaximumSpeedDifferenceOnRamp}$. Then, the final desired speed of a vehicle on a section is given from the following logic:

- **if** (the adjacent slower lane is an On-ramp)
  - $\text{MaximumSpeed} = \text{MeanSpeedVehiclesDown} + \text{MaximumSpeedDifferenceOnRamp}$
- **else** $\text{MaximumSpeed} = \text{MeanSpeedVehiclesDown} + \text{MaximumSpeedDifference}$

$$\text{DesiredSpeed} = \text{Minimum} \left( v_{\text{max}}(n,s), \theta(n) \right) \times \text{MaximumSpeed}$$

This procedure ensures that the differences of speeds between two adjacent lanes will always be lower than $\text{MaximumSpeedDifference}$ or $\text{MaximumSpeedDifferenceOnRamp}$, depending on the case.

**Effects of Grades**

The maximum desired acceleration for a vehicle ($\text{vehicle_acc}$) is a vehicle parameter defined by the modeler in AIMSUN2 for all vehicles belonging to the same class. The influencing of the section grade on the vehicle movement is taken into account by increasing or reducing the acceleration and deceleration rate. The maximum acceleration for a vehicle on a section is a function of the grade and the maximum desired acceleration for the vehicle i.e:

$$\text{accel} = \text{Maximum}(\text{vehicle_acc} - \text{grade} \times 9.81/100.0, \text{vehicle_acc} \times 0.1)$$

In order to avoid zero or negative acceleration values, a minimum value of 10% of the maximum desired acceleration for the vehicle is used.

**Model calibration and testing**

In addition to numerous tests performed by the research team, the car-following model in AIMSUN2 has been tested and calibrated in various real life projects; due to space limitations we only present the benchmark test performed based on the data and the methodology supplied and proposed by a research group from Robert Bosch GmbH (15). This test employed a set of field data and most of the microsimulator developers in Europe and North America were invited to participate. The test for the car-following model was defined as follows:

*The primary task of a car-following model is to reproduce realistic car-following behavior. The reality has been measured with a radar sensor equipped vehicle recording distance and relative speed to the front car (additionally to the own car's speed) in a 100 ms cycle, see (16) for further details. One specific sequence of 5 min length has been chosen to perform the comparison. This sequence has been recorded under stop&go conditions during an*
afternoon peak on a one-lane-per-direction fairly straight road in Stuttgart, Germany. Stop&go is most challenging to the models because the free flow behavior is relatively easy reproducible by any model. During the 5 min sequence several decelerations and accelerations of the front car have been observed. At one moment after 144 sec the front car turned off resulting in a distance step of about 40 m. Because such a maneuver can always happen in real traffic the models have to be able to deal with. Note that it can't be the target of a simulation model to reproduce exactly the measured behavior of this specific driver in the specific test vehicle. Driver and vehicle variations have to be respected. Hence, the main focus lies on qualitative differences. But the fairly good reproduction of the behavior indicates a model's realism. To give an impression of similarity to the measured behavior a quantitative error metric on the distance seems to be reasonable. To avoid overrating discrepancies for large distances a relative metric was chosen weighted by the logarithm and squared. Only the values after each second have been considered.

The error metric used to measure the accuracy of the fitting between measured and simulated values was:

\[ E_m = \sqrt{\sum \left( \log \left( \frac{d_{\text{sim}}}{d_{\text{meas}}} \right) \right)^2} \]

where \(d_{\text{sim}}\) is the distance of the simulated vehicle, \(d_{\text{meas}}\) is the distance measured with the test vehicle, and \(\log\) denotes the logarithm base 10. Figures 2 and 3 display the curves for the observed versus simulated relative distance and speed between leader and follower. The numerical value for the error metric for the AIMSUN2 model was 3.4726. These results show that the AIMSUN2 car-following model is able of a fairly good reproduction of the observed values. The numerical value of the error metric outperforms those provided for most of the currently used models (see (16) for details).

An additional test to analyze the quality of the microscopic simulator is to check the ability to reproduce macroscopic behavior. Also the research team at Bosch proposed in (16) a test to compare various microscopic simulators: “The macroscopic behavior of a microscopic model can be most easily tested by simulating the traffic
on cyclic one lane roads. This excludes any effects of lane changes and node passing and concentrates on the car-following task. For this study, a cyclic road of 1000 m length was used. A fixed number of vehicles has been initially set with speed value 0 km/h at randomly determined positions. All vehicles had the same length of 4.5 m and the drivers had the same free flow speed of 54 km/h. Starting with this initial situation a 10 minute time period was simulated without any measurements to reach traffic conditions which are achievable by the model’s behavior itself. After the starting phase the traffic behavior has been recorded at one local measurement point during a simulation time of 2 hours (exact passing time and speed value of each vehicle). The fixed number of vehicles for the simulation run was varied in discrete steps to realize different traffic densities. To visualize the results the traffic flow has been drawn versus the density (given as the number of initially set vehicles on the 1 km ring). The maximal mean traffic flow value of about 1800 veh/h is known as a quite realistic value for longer periods of measurement time. Under urban traffic conditions this maximal flow is typically reached at higher density values than for freeway traffic. The results of AIMSUN2 for the simulated flow density curve versus the empirical one for the second test are displayed in figure 4, and they appear to be fairly reasonable.

This subjective perception is confirmed by the values of the error metric to measure the fitting between the measured and simulated values as before, that in this case is $E_m = 0.063381$.

**Lane Changing Model**

The lane change model in AIMSUN2 can also be considered as a further evolution of the Gipps lane change model (17). Lane change is modeled as a decision process analyzing the necessity of the lane change (as in the case of turning maneuvers determined by the route), the desirability of the lane change (as for example to reach the desired speed when the leader vehicle is slower), and the feasibility conditions for the lane change that are also
local, depending on the location of the vehicle on the road network. The lane-changing model is a decision model that approximates the driver's behavior as follows:

- Each time a vehicle has to be updated the model draws up the question: *Is it necessary to change lanes?* The answer to this question depends on several factors: the turning feasibility at current lane, the distance to next turning and the traffic conditions in the current lane. The traffic conditions are measured in terms of speed and queue lengths. When a vehicle is driving slower than he wishes, he tries to overtake the preceding vehicle. On the other hand, when he is travelling fast enough, he tends to go back to the slower lane.

- If the answer to the previous question is affirmative, to succeed in the lane changing we have to answer two more questions:

  a) *Is it desirable to change lanes?*

  This requires checking if there will be any improvement in the traffic conditions for the driver as a result of the lane changing. This improvement is measured in terms of speed and distance. If the speed in the future lane is faster (i.e. a user specified threshold is exceeded) than the current lane or if the queue is shorter than a threshold, then it is desirable to change lanes.
b) **Is it possible to change lanes?**

This requires verifying if there is a sufficient gap to do the lane change with complete safety. For this purpose, we calculate both the braking imposed by the next downstream vehicle to the changing vehicle and the braking applied by the changing vehicle to the future upstream vehicle. If both braking ratios are acceptable then lane changing is possible.

In order to achieve a more accurate representation of the driver’s behavior in the lane changing decision process, three different zones inside a section are considered, each one corresponding to a different lane changing motivation. The distance up to the end of the section characterizes these zones and which is the next turning point. The figure 5 depicts the structure of these zones that are defined as follows:

- **Zone 1**: This is the farthest from the next turning point. The lane changing decisions are mainly governed by the traffic conditions of the lanes involved; the feasibility of the next desired turning movement is not yet taken into account. To measure the improvement that the driver will get on changing lanes we consider several parameters: the desired speed of the driver, speed and distance of the current preceding vehicle and speed and distance of the future preceding vehicle.

- **Zone 2**: This is the intermediate zone. Mainly it is the desired turning lane that affects the lane changing decision. Vehicles who are not driving on a valid lane (i.e. a lane where the desired turning movement can be done) tend to get closer to the correct side of the road where the turn is allowed. In this zone vehicles look for a gap and may try to accept it without affecting the behavior of vehicles in the adjacent lanes.

- **Zone 3**: This is the nearest to the next turning lane changing point. Vehicles are forced to reach their desired turning lanes, reducing the speed if necessary and even coming to a complete stop in order to make the lane change possible. Also, vehicles in the adjacent lane can modify their behavior in order to allow a gap big enough for the lane-changing vehicle.

Lane changing zones are defined by two parameters, Distance to Zone 1 and Distance to Zone 2. These parameters are defined in time (seconds) and they are converted into distance whenever it is required for each vehicle at each section using the Vehicle Desired Speed at a Section. This means that these distances are then local parameters their value depending on the current traffic conditions on the section. AIMSUN2 provides default values based on the user’s experience, however it should be noticed that to find the right values for these parameters is part of the model calibration exercise that the user has to perform.
Look Ahead

When traffic conditions are very congested it may happen that some vehicles cannot reach the appropriated turning lane and consequently miss the next turn. This situation could appear either in urban networks where there are short sections or in freeways where weaving sections may be relatively short. It gets then worst as the sections get more congested.

Tuning some modeling parameters such as lane changing zone distances, simulation step, acceleration rates etc., could improve the behavior in order to minimize the number of lost vehicles. Also using polysections in modeling the geometry instead of sections, when feasible, to model streets or weaving areas might help to improve the situation, but it was not enough.

To override these drawbacks a major improvement has been done in the lane change model consisting in modeling a Look Ahead process. The objective is to provide vehicles with the knowledge of various next turning movements and not only one, so they will be able to take decisions not only based on the immediate next turning movement, but on a set of next turning movements.

The Look Ahead consists of four steps:

1. At any time, each vehicle knows the next two turning movements, so the lane changing decisions are influenced by two consecutive turns.
2. The lane changing zones 2 and 3 of any section is extended back beyond the limits of the section, therefore affecting the upstream sections.
3. The next turning movement also influences the turning maneuvers so the selection of destination lane is done based also on the next turn.
4. A greater variability is given to the Lane Changing Zones in order to distribute the lane changing maneuvers along a longer distance.

These points are described with more detail in the following sections.

Influence of two turning movements in the lane changing

When a vehicle is generated and introduced in the network, two turning movements are calculated: the next turn and the ahead turn. That means that the vehicle has knowledge of the first three consecutive sections (or polysections) that will have to follow: the current input section, the second defined by the next turn, and a third section that is defined by the second section and the ahead turn.

Then, each time a vehicle enters a new section (or polysection), the ahead turn becomes the next turn and a new ahead turn is calculated.

The vehicle behavior of a vehicle driving in zones 2 or 3 of a section is mainly governed by the next turn, therefore it will first try to reach a lane where the next turn is feasible. Once it is driving in an appropriate lane with respect to the next turn, it will take into account the ahead turn. It consists in checking whether it is already driving inside the extended zone 2 or 3 of the next section. If that case it will try to find the best lane in the current section that, still allowing next turn, will drive the vehicle either to a lane in the next section where the ahead turn is feasible or to the lane closest to a feasible one.

This is illustrated in the example of figure 6. Vehicle a in section 3 is already locate in the left lane in order to be as close as possible to the left when entering in section 2. In this case the next turn is from 3 to 2 while the ahead turn will be from 2 to 4. Therefore, although the right lane would be OK according to the next turn, it is not according to
the \textit{ahead turn}. Similar behavior for the vehicle \textit{b} that has changed to the rightmost lane in section 1 in order to reach the \textit{ahead turn} to the right.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure6.png}
\caption{Extending back zones 2 and 3}
\end{figure}

In case that Zone 2 or 3 of a section is longer than the length of that section, the zone is extended beyond the limits of the section, therefore entering into the upstream section. In order to make the Look Ahead model work properly it is now very important to define appropriately the lane changing zones. Look Ahead model will not be applied whenever the lane changing zones 2 or 3 are shorter than a section. In the case of polysection the user must take care of properly define the zones for each section of the polysection. The zone extension is also illustrated in figure 6.

\section*{Turning maneuvers}

When a vehicle reaches the end of a lane in a section and enters into a junction it may have the possibility of choosing among different connections, i.e. movements from origin lane (the current one) and destination lanes (the lane in the next section).

For instance, in figure 7 vehicle \textit{a} is going to turn right from the rightmost lane of section 17 to section 11. However it can choose among the different destination lanes in section 11. As this vehicle’s \textit{ahead turn} is again to the right it will enter section 11 through the rightmost lane instead of using the central one.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{figure7.png}
\caption{Variability of Lane Changing Zones}
\end{figure}
When a vehicle crosses from zone 1 to zone 2 there is a change in the vehicle’s behavior, as now it becomes relevant the next turn. Also the crossing from zone 2 to zone 3 produces a change in the behavioral rules of the vehicles, as now reaching the turning lane becomes urgent. In order to distribute these changes of behavior along a longer distance a greater variability is given to the Lane Changing Zones. These zones are calculated particularly for each vehicle according to the following equation:

\[
\text{Distance Zone } n \text{ for vehicle } v \text{ in section } s \text{ (in meters)} = \frac{\text{Distance Zone } n \text{ (in seconds)} \times \text{Speed Limit of Section } s \times \text{Vehicle } v \text{ Coefficient}}{\text{Vehicle } v \text{ Coefficient} = \frac{\text{Speed Limit of Section } s}{\text{Desired speed of Vehicle } v \text{ in section } s}}
\]

This algorithm ensures that for vehicles whose desired speed is slower than the speed limit the lane changing zones will be longer than for vehicles whose desired speed is greater than the speed limit. It means for instance that a heavy truck will try to reach the appropriate turning lane earlier than a speed car.

**Lane Changing Modeling at ON-Ramps**

A special Lane Changing Modeling is applied at entrance ramps. An additional zone parameter may be defined, TimeDistanceOnRamp. This is the Distance (in seconds, converted into distance as before) from those lateral lanes considered to be on-ramp lanes, in order to distinguish between a common lateral lane, that is a long lane used for overtaking which drops down, from the proper on-ramp lanes, which are never used for overtaking. Vehicles driving on a lateral lane that are farther than TimeDistanceOnRamp from the end behave as if they were in the Zone 1 of a normal lane. When they are closer than TimeDistanceOnRamp to the end of the lane, they behave as having to merge from an on-ramp. Merge from on-ramp model takes into account whether a vehicle is stopped or not, if it is stopped whether it is at the beginning of the on-ramp queue or not and how long has been waiting. There is another vehicle parameter, Maximum Waiting Time, that determines how long a vehicle is willing to wait before getting impatient.

**Lane Change Prohibitions**

Prohibitions to change lanes are modelled through the definition of continuous lines between lanes. Lane changing is not allowed wherever there is a continuous line between two lanes. Through this feature the user may avoid undesired lane changing in some conflicting points, as it may be the nearby of an on-ramp. For each pair of adjacent lanes the user may define a continuous line at the right, at the left or in both sides. Continuous line at the right means that no lane change is allowed from the right lane towards the left lane. Continuous line at the left means that no lane change is allowed from the left lane towards the right lane. Continuous line at both sides means that no lane change at all is allowed between both lanes. Continuous lines can be defined for the whole lane length or only part of it.

![Figure 8: Example of continuous lines](image)

Figure 8 shows an example where there is a short continuous line between the on-ramp lane (lane number 1) and the rightmost main lane (lane number 2), so vehicles can not merge from the on-ramp during the first 20 meters, as
is the case of vehicle 2. There is also a continuous line between the left lane (lane number 3) and the right lane with the purpose of creating more gaps in the nearby of the on-ramp. This is the case of vehicle 4, which is not allowed to change to the right lane, which would disturb the merging of vehicles 1 and 2.

Overtaking Maneuver

An overtaking maneuver takes place mainly on zone 1, although it can also take place in zones 2 and 3 when the vehicle is in the appropriate turning lane. In order to promote or discourage the overtaking, there are two parameters that the user can define: PercentOvertake and PercentRecover. PercentOvertake is the percentage of the desired speed of a vehicle below what the vehicle may decide to overtake. This implies that whenever the leader vehicle is driving slower than PercentOvertake% of the follower desired speed, the follower vehicle will try to overtake. The default value is 0.90. PercentRecover is the percentage of the desired speed of a vehicle above what a vehicle may decide return to the slower lane. It means that whenever the leader vehicle is driving faster than PercentRecover% of the follower desired speed, the follower vehicle will try to recover the rightmost (or leftmost) lane. The default value is 0.95.

The whole lane change process is modeled formally as a decision tree model whose logic structure is depicted in the diagram of figure 9. The process identifies the type of entity (central lane, off-ramp lane, junction, on-ramp etc.) in which the maneuver is going to be done, next determines how the zone modeling should be applied. The current traffic conditions are analyzed and the level at which the lane change can be performed is determined, and then the corresponding model is applied.

Figure 9: Lane changing decision tree
AIMSUN2 Route Based

In this model, vehicles are fed into the network according to the demand data defined as an O/D matrix and they drive along the network following specific paths in order to reach their destination. In the main Route Based simulation new routes are to be calculated periodically during the simulation, and a Route Choice model is needed, when alternative routes are available. Two alternative Route Choice models are currently available in AIMSUN2: Binomial and Multinomial Logit. In the variable mode, regardless of the Route Choice model used, there are two types of driver’s behavior with respect to the route assignment: Static and Dynamic, which refers to whether or not a vehicle can modify the actual path en-route as new paths become available during the trip.

Shortest Route Component

This route-based version of AIMSUN2 becomes a simulation platform for networks containing an ATMS/ATIS component in which traffic management centers provide real time information and the drivers react by possibly using different alternative routes. As ITS technology is being deployed Advanced Traffic Information Systems provide such information for Vehicle Guidance, next generation ATM systems require this type of dynamic tools (18). A route based microscopic simulation implies that the simulator needs to store the current shortest routes from the beginning of every section to all destinations (whether these are sections or centroids). It is also needed to store the routes that different vehicles wish to follow that is all previously generated routes as long as there are vehicles using them (12). For each destination and instant in time, the routes are stored as a tree that allows knowing how to reach the destination from any section of the network. The shortest routes component takes into account turning penalties as the different turning movements at the end of a section have in general unequal travel times (e.g. left turn, going straight, etc). The procedure implemented to compute the shortest routes to a destination uses a network where an arc, connecting two nodes, models a section. A special arc connecting the beginning of the turning to its end models a turning movement. The computation of shortest routes uses a label setting method, where the labels are associated with an arc. The network is constructed only once before the start of the simulation. During the simulation, the computation of shortest routes is launched at certain time steps. The shortest route routine is a variation of Dijkstra’s label setting algorithm. It provides the shortest routes from the start of every section to all destinations. The penalties associated with turning movements are taken into account.

Default Section Cost Functions

Two types of section cost functions are used for calculating the shortest path trees, depending on whether or not there are simulated data available to be used for. These are the Initial Cost Function and the Current Cost Function. In both cases, the cost function represents section travel time in seconds, including the penalty of the turning movement, if it exists. The Initial Cost Function is applied at the beginning of simulation when there is no simulated data gathered to calculate the travel times. In this case, the cost of each section is calculated as a function of the travel time in free flow conditions and the capacity of the section. The travel time in free flow conditions is the time it would take a vehicle, travelling at the maximum allowed speed of the section, to cross the section. The initial cost of section \( s \), \( IniCost_{s,i} \), taking into account the penalty for the i-th turning at the end of the section, and the capacity \( CT_{s,i} \), of the i-th turning on section \( s \) is calculated as follows:

\[
IniCost_{s,i} = TravelTFF_{s,i} + TravelTFF_{s,i} * \phi * (1 - CT_{s,i}/MaxCapacity)
\]

where \( TravelTFF_{s,i} = \text{link length}_s/\text{MaximumSpeed}_s + \text{turning length}_{s,i}/\text{MaximumSpeed}_{s,i} \)

\( MaxCapacity \) is the Maximum section capacity in the network

\( \phi \) is the capacity weight

\( MaxCapacity \) is the Maximum section capacity in the network
The capacity weight is a user-defined parameter that allows the user to control the influence that the section capacity has in the cost in relation with the travel time. The **Current Cost Function** is applied when there is simulated travel time data available. The current cost for each section is the mean travel time, in seconds, for all simulated vehicles that have crossed the section during the last statistical gathering period as depicted in Figure 10. Cycle represents the time interval for updating the shortest paths trees based on the current costs. The statistical data used for cost calculations consider a given number of cycles (a user defined parameter) according to a rolling horizon technique (in the example of the figure only the two last cycles are used). In case that no vehicle has crossed the section we distinguish the case of a totally congested section from the case of an empty section. In the first case, the cost is calculated as the maximum between the Initial Cost and the average waiting time for the vehicles in front of the queue in the section. In the second case, the cost is taken as the initial cost. Taking into account that AIMSUN2 can distinguish between different vehicle types, and that lanes can be reserved only for certain classes.

$$\text{Cost}_{s,i,vt} = \text{CurrentCost}_{s,i,vt} + \text{CurrentCost}_{s,i,vt} \cdot \varphi \cdot (1 - \frac{\text{CT}_{s,i}}{\text{MaxCapacity}})$$

(HOV or public transport, for example), in these cases may be different costs for different classes.

The cost of vehicle type vt in turning i-th in section s is:

1. Calculate initial shortest routes for each O/D pair using the defined initial costs.
2. Simulate for a predefined period (e.g. 5 minutes) assigning to the available routes the fraction of the trips between each O/D pair for that period according to the selected route choice model and obtain new average link travel times as a result of the simulation.
3. Recalculate shortest routes, taking into account the current average link travel times.
4. If there are guided vehicles, or variable message signs suggesting rerouting, provide the information calculated in 3 to the drivers that are dynamically allowed to reroute.
5. Go to step 2.

At the beginning of the simulation, shortest path trees are calculated from every section to each destination centroid, taking as section costs the Initial Cost Function. These paths are used at the beginning of the warm-up initialization period. During the simulation, new routes are recomputed every time interval taking as section costs the ones calculated by the Current Cost Function explained before. The user may define the time interval for recalculation of paths and the maximum number of path trees that wishes to maintain during the simulation. When the maximum number of path trees (k) is reached, the oldest paths will be removed as soon as no vehicle is using them. It is assumed that vehicles only choose among the most recent k path trees, therefore, the oldest ones will become obsolete and unused.

Static versus Dynamic Route assignment Models

Vehicles are initially assigned to a route from a set of available routes on a probabilistic way. Apart from the initial assignment of route, which is made at the vehicle’s departing time, there is the possibility of making a route reassignment during the trip. This is the Dynamic route choice model in which a guided, or informed, vehicle can make a new decision about what route to follow at any time along its trip, whenever there are new shortest routes available. In the Static model, a vehicle will always follow its initially selected route until reaching the destination, although new shortest route could be available during the trip. In the Dynamic model only guided vehicles can take the decision of changing to a new shortest route during the trip, assuming that this information is only available to them. Regarding this, there is a parameter for each vehicle type that gives the percentage of guided vehicles. The behavior of the driver in response to information acquisition may be modeled in different ways: as a shortest route follower, a boundedly rational user or route choice based on discrete path choice models. This opens the way to a wide variety of applications in the context of ATMS/ATIS systems.

Route Choice Models

Currently there are three default route choice models implemented in AIMSUN2, which are used either when assigning the initial path for a vehicle at the beginning of its trip or when having to decide whether or not to change path en-route in the dynamic modeling. These models are the Binomial, the Multinomial Logit and the Modified Logit or C-Logit models. In the Binomial Model, a Binomial (k-1, p) distribution is taken to find the probability of selecting each path. Parameter k is the number of available paths and p is the “success” probability. This model does not consider the travel costs in the decision process, but only the time at which the path was calculated. Selecting a small p implies that oldest paths will be more likely used while selecting high values of p implies that the most recent paths will be more frequently taken. The Multinomial Logit Model assumes that the utility $U_{k}^{rs}$ of route k between origin r and destination s is given by:

$$U_{k}^{rs} = -\theta t_{k}^{rs} + \epsilon_{k}^{rs}$$

Where: $\theta$ is a shape or scale factor parameter, $t_{k}^{rs}$ is the expected travel time on route k from r to s, calculated as the sum of the current costs of all the sections composing the path (CurrCost(s) function as explained above) and $\epsilon_{k}^{rs}$ is a random term. The underlying modeling hypothesis is that random terms $\epsilon_{k}^{rs}$ are independent identically distributed GUMBEL variates. Under these conditions the probability of choosing route k amongst all alternative routes from r to s is given by the logistic distribution:
The scale factor $\theta$ plays a twofold role making independent of the measurement units the decision based on differences between utilities, and influencing the standard error of the distribution of expected travel times. If $\theta < 1$ (high perception of the variance) there is a trend to utilize many alternative routes, conversely if $\theta > 1$ alternative choices are concentrated in very few routes.

Logit models can show unrealistic path choice probabilities when there is a significant overlapping among the alternative routes, which unfortunately is a quite common situation in transport applications. This is a fact observed by many researchers as well as by practitioners. This problem could be partially overcome through explicit path enumeration of heavily overlapping paths, but this becomes unpractical in large networks with many alternative paths. To overcome these drawbacks the Modified Multinomial Logit Model, or in short the C-Logit Model has been proposed (19-20). The C-Logit deals with similarities among overlapping paths through an additional “cost” attribute: the commonality factor. The utility $U_{rs}^k$ associated to path $k$ between origin $r$ and destination $s$ can be redefined as:

$$U_{rs}^k = \tilde{V}_{rs}^k + \varepsilon_{rs}^k \quad \forall k \in I_{rs} \{\text{available paths connecting the OD pair } (r, s)\}$$

and making: $\tilde{V}_{rs}^k = V_{rs}^k - CF_k$, where $V_{rs}^k$ is the average or systematic utility on path $k$ (i.e. the travel time $t_{rs}^k$), and $CF_k$ is the commonality factor on path $k$.

The probability distribution (2) becomes now:

$$P_{rs}^k = \frac{\exp[\theta(V_{rs}^k - CF_k)]}{\sum_{h \in I_{rs}} \exp[\theta(V_{rs}^h - CF_h)]}$$

The commonality factor $CF_k$ is directly proportional to the degree of similarity (or overlapping) of path $k$ with other paths belonging to $I_{rs}$. This implies that heavily overlapping paths have larger commonality factors and thus a smaller systematic utility (larger generalized cost) with respect to similar but independent paths.

An example of such commonality factor could be:

$$CF_k = \beta_k \ln \sum_{h \in I_{rs}} \left[ \frac{L_{hk}}{L_h^{1/2}L_k^{1/2}} \right]^\gamma$$

Where: $L_{hk}$ is the length of links common to paths $h$ and $k$, and $L_h$, $L_k$ are the overall lengths of paths $h$ and $k$. Lengths can either be physical or the link additive part of generalized costs.

Behavioral dynamic route choice models are still an open field of research; for this reason in addition to the default route choice models AIMSUN2 also includes a function editor allowing the user to define his own route choice function using the currently available arguments provided by the simulation such as links cost as defined, link lengths, experienced travel times in the past time periods, etc.
TRAVEL DEMAND

AIMSUN2 accepts as input time-dependent Origin-destination matrices defined as sets of matrices for each time interval and each vehicle type. Although a good estimation of time dependent Origin-Destination matrices is still a problem, far for being satisfactorily solved in all cases, it is also widely accepted that this is a main requirement for most of the ITS applications, especially for Advanced Traffic Management. Therefore any simulation tool aimed at assessing ITS applications and policies must be able of dealing with time-dependent Origin-destination matrices; this was the main reason for implementing this function in AIMSUN2. As a complement, in order to provide the model at least an acceptable input, we have developed a heuristic approximate procedure combining the information of an existing target matrix, possibly one used for transport planning purposes, with complementary information (specific cordon surveys, for example). This is used for refining the target matrix combined with traffic flow measurements for each time interval. When real-time measurements are also available a further refinement is still possible using neural networks. The heuristic procedure (21) can be summarized as follows:

**Step 1.** Assuming that a global O/D matrix is available, possibly from a transport planning study, a “traversal” matrix for the simulated sub-network is extracted from the global Origin-Destination matrix using, for example, the process defined in the EMME/2 package (22).

**Step 2.** The Traversal Matrix is combined with complementary available information on the time distribution of the total number of trips on the network (possibly generated by cordon or license plate surveys on the subnetwork modeled microscopically), to generate an Traversal Time Sliced O/D Matrix consistent with time variation of the link flows in the time horizon, and the time distribution of the total number of trips.

**Step 3.** Each time slice of the traversal matrix, and the link flow measurements for that time interval provided by the data collection, are the input to the heuristic matrix adjustment whose output is the Adjusted O/D Matrix for the corresponding time interval. This process has proven in practice its capability to adjust trip matrices to the flow variations in the time horizon considered, reflecting in that way the time variability of the traffic demand. We have adopted the following bilevel formulation of the matrix adjustment problem as a non-linear optimization problem:

\[
\begin{align*}
\text{Min } F(v(g), v) &= \frac{1}{2} \sum_{a \in A} (v(g)_a - \hat{v}_a)^2 \\
v(g) &= \arg \min \sum_{a \in A} \int_0^{v_a(g)} s_a(x) dx \\
\text{s.t.} & \\
& \sum_{k \in K_i} h_k = g_i, \quad \forall i \in I \\
& h_k \geq 0, \quad \forall k \in K_i, \quad \forall i \in I \\
& v_a = \sum_{i \in I} \sum_{k \in K_i} \delta_{ak} h_k, \quad \forall a \in A
\end{align*}
\]

where \( v_a(g) \) is the flow on link \( a \) estimated by the lower level traffic assignment problem with the adjusted trip matrix \( g \), \( h_k \) is the flow on the \( k \)-th path for the \( i \)-th O-D pair, and \( \hat{v}_a \) is the measured flow on link \( a \). \( I \) is the set of all Origin-Destination pairs in the network, and \( K \) is the set of paths connecting the \( i \)-th O-D pair. The algorithm used to solve the problem, based on a proposal by Spiess (23) is heuristic in nature, of steepest descent type, and does not guarantee that a global optimum to the formulated problem will be found. The iterative process is as follows:

**At iteration \( k \):**

- Given a solution \( z^k \), an equilibrium assignment is solved giving link flows \( v^k \), and proportions \( \{p^k_{ih}\} \)

satisfying the relationship...
\[ v_a^k = \sum_{i=1}^{p_a} g_i^k, \quad \forall a \in A \]

Note: the target matrix is used in the first iteration (i.e. \( g_1^1 = \hat{g}_1, \quad \forall i \in I \))

- The gradient of the objective function \( F(v(g)) \) is computed. For a more realistic approach the gradient is based on the relative change in the demand, written as:

\[
g_{i}^{k+1} = \begin{cases} 
\hat{g}_{i} & \text{for } k = 0 \\
da \left( 1 - \lambda^k \left[ \frac{\partial F_2(g)}{\partial g_i} \right]_{k^*} \right) & \text{for } k = 1, 2, 3, \ldots 
\end{cases}
\]

(Then a change in the demand is proportional to the demand in the initial matrix and zeroes will be preserved in the process).

- The gradient is approximated by

\[
\frac{\partial F_2(g)}{\partial g_i} = \sum_{k \in K_i} h_k \sum_{a \in A} \delta_{ak} (v_a - \hat{v}_a), \quad \forall i \in I
\]

(where \( \hat{A} \subset A \) is the subset of links with flow counts).

- The step length is approximated as:

\[
\lambda^* = \frac{\sum_{a \in A} v_a \cdot (\hat{v}_a - v_a)}{\sum_{a \in A} v_a^2}
\]

where

\[
v_a^* = -\sum_{i \in I} g_i \left( \sum_{k \in K_i} \sum_{a \in A} \delta_{ak} (v_a - \hat{v}_a) \right) \left( \sum_{k \in K_i} \delta_{ak} h_k \right)
\]

Step 4. This Adjusted Time Sliced O/D Matrix is the input to the Route Based AIMSUN2

IMPLEMENTATION

AIMSUN2 is embedded in GETRAM (Generic Environment for Traffic Analysis and Modeling) a simulation environment comprising a traffic network graphical editor (TEDI), a network data base, a module for storing results, and an Application Programming Interface to aid interfacing to other simulation or assignment models. An additional library of DLL functions, the GETRAM EXTENSIONS, enables the system to communicate with external applications, as for example real-time control logic. The functional structure of the system is depicted in figure 11.

TEDI is a graphical editor for traffic networks. It has been designed with the aim of making the process of network data entry and model building user-friendly. Its main function is the easy entry of the network feeding the traffic simulators like AIMSUN2. To facilitate this task the editor accepts as a background a graphical description of the network area, in terms of a DXF file from a GIS or an AutoCAD system, so sections and nodes can be built subsequently into the foreground. The editor supports both urban and interurban roads, which means that the level of detail covers elements such as surface roads, entrance and exit ramps, intersections, traffic lights and ramp metering. Figure 1 illustrates the process of using the graphical editor to build an urban model on the top of a background.
A CASE STUDY ON THE USE OF AN AIMSUN2 MODEL TO REPRODUCE TRAFFIC FLOWS: STUDY AREA AND SCOPE

Auckland is located towards the north of New Zealand’s North Island. With a population of some 1.2 million people it is the country’s largest urban centre and is growing at 2.5% per year. The study section (see Figure 12) is a 9.7 km length of the Southern Motorway extending south from the Central Business District (CBD). It carries bi-directional Average Annual Daily Traffic (ADDT) volumes along its length ranging from 109,000 to about 200,000 vehicles per day (vpd). To obtain information that was comprehensive, yet at a sufficiently high level of detail, many different types of data were collected simultaneously at varying intervals along the motorway [4]. Automatic count data were collected over a full 7-day week. Resource-intensive data collection methods, including video taping, aerial photography and laser-gun speed profiles, were measured on a single day. Most of the data were used to calibrate the AIMSUN2 model. Some data not used in the calibration were used as benchmarks to which simulation output was compared in order to validate the model. The work reported in this study was a preliminary investigation into the ability of the model to reproduce traffic flows in the northbound direction of this motorway corridor. The general approach was to define the motorway links, apply field-measured traffic flows at the network boundaries and then calibrate the model by fine-tuning sensitive parameters to seek agreement with measured data at intermediate points within the corridor.
Figure 12 Auckland Study Area

Motorway Model

A model was constructed in GETRAM (Version 3.2) of the northbound motorway lanes. The model consists of three motorway through lanes and a short length of each ramp. Traffic was applied to this network as three vehicle types (cars, light commercial and heavy commercial vehicles). A trip matrix was produced for each vehicle type during each 15 minute time period for an extended morning commuter peak from 6.00am to 10.00am. The traffic flows applied to the model were an approximation to the actual flows that existed on the motorway on Friday 26 September 1997. Some gaps in the field data were filled by comparison with flow data from other days and missing length classification percentages were assumed from a 1992 postcard survey. After inputting the road sections and trip matrices the model was run and some parameters adjusted by trial and error to replicate traffic conditions observed in the field.

Model Outputs

Quantitative analysis for average speeds and flows over 5-minute intervals comparing field data with the model data shows a good agreement, at most of the sites [24]. It should be noted that subsequent runs with the new parameters available in version 3.3 of GETRAM have shown closer agreement. Better performance should be expected with the improved new GETRAM version 4.0. Figure 13 displays some of these results for a weaving area.
Figure 13

Figure 14 shows an aerial photo of the motorway north of the Greenlane on-ramp and the graphical model displays at the same moment in the simulation. The bunching of traffic as on-ramp vehicles merge into the through lane flows can be clearly seen in both the photo and the simulated outputs. There is a good agreement between the field and modelled vehicle lane densities.
THE ASSESSMENT OF ADVANCED TRAFFIC CONTROL SYSTEMS

The implementation of the requirements for a microscopic simulator be able of simulating ITS systems means that further than the already described abilities the simulation software must be able of interacting with a wide variety of traffic control and management applications, as well as reproducing the impacts of the management strategies on the traffic conditions. A general description of these requirements can be found in [25]. From the point of view of the software design a common solution to the first problem implemented by some of the currently available microsimulators consist of providing a library of DLL functions enabling the simulator to interface the external application of interest. This library gives the simulator the ability to communicate with almost any real-time traffic control or management applications. The process of information exchange between the simulator and the external application is conceptually illustrated in Figure 15.

The microsimulation model of the road network emulates the detection process providing the external application with the required “Simulation Detection Data”. The external application (user provided) decides which control and/or management actions have to be applied on the road network and sends the corresponding information to the simulation model which then emulates their operation through the corresponding model components such as traffic lights, VMS, etc.
EVALUATION OF RAMP CONTROL LOGIC PERFORMANCE

Once the calibration of a simulation model has proven its ability for reproducing the actual traffic conditions on the road network, within an acceptable degree of accuracy, the calibrated model is ready to conduct the simulation experiments that will provide the answers to the what if questions necessary for the decision makers. An example of a feasibility study on the suitability of integrated ramp control strategies for freeway corridors is the simulation test of ramp control logic developed by the Minnesota Department of Transportation (MnDOT) on a segment of the I-35W in Minneapolis. The work was done at the ITS Laboratory of the University of Minnesota (26).

Test Site Description

Following discussions with the MnDOT engineers in charge of the Transport Management Centre, a 24 km long section of I-35W going south was selected for testing purposes. This section was specifically chosen as it includes most of the common geometry configurations found in the Twin Cities. This section begins at Downtown Minneapolis and ends at the interchange with Highway 13. It includes 20 exit and 22 entrance ramps, which are controlled during PM peak hours. Four entrances are freeway to freeway ramps, carrying very high volumes in the range of 1200 veh/hr with long spill-back queues. The geometry includes 6 weaving sections and also has a lane drop section. The test site is divided into three zones and has three bottleneck locations. It also has a single HOV lane from I-494 interchange to Highway 13 that is about 10 km long. The total experiment was based on data collected during a 60 day period during May and June 1999. Most of this data was used to calibrate simulation model parameters.

Calibration and Validation of the Simulation Model

From a methodological point of view it is widely accepted that simulation is a useful technique to provide an experimental test bed to compare alternate system designs, replacing the experiments on the physical system by experiments on its formal representation in a computer in terms of a simulation model. The outcomes of the computer experiment provide in this way the basis for a quantitative support to decision-makers.

The reliability of this decision making process depends on the ability to produce a simulation model representing the system behavior closely enough for the purpose of using the model as a substitute of the actual system for experimental purposes. This is true for any simulation analysis in general and obviously for traffic simulation. The process of determining whether the simulation model is close enough to the actual system is usually achieved through the validation of the model, an iterative process involving the calibration of the model parameters and comparing the model to the actual system behavior and using the discrepancies between the two, and the insight gained, to improve the model until the accuracy is judged to be acceptable.
In the case of the traffic systems the behavior of the actual system is usually defined in terms of the traffic variables, flows, speeds, occupancies, queue lengths, and so on, which can be measured by traffic detectors at specific locations in the road network. To validate the traffic simulation model the simulator should be able of emulating the traffic detection process and produce series of simulated observations which comparison to the actual measurements will be used to determine whether the desired accuracy in reproducing the system behavior is achieved. Statistical techniques will be used for such determination.

In the case of a microsimulation model in general, and specifically for an AIMSUN2 traffic simulation model, the model behavior depends on a rich variety of model parameters most of which have already been described in the previous sections. Summarizing, if one considers the model composed of entities, i.e. vehicles, sections, junctions, intersections, and so on, each of them described by a set of attributes, i.e. parameters of the car-following, the lane change, gap acceptance, speed limits and speed acceptance on sections, an so on, are the numerical values of these parameters which will determine the model behavior. The calibration process will have as objective to find the values of these parameters that will produce a valid model.

Some examples will help to illustrate this dependency between parameter values and model behavior. Vehicle lengths have a clear influence in flows: as the vehicle lengths increase flows decrease and queue lengths increase. As we have shown when discussing the AIMSUN2 car-following model the target speed, the section speed limit and the speed acceptance, among others, define the desired speed for each vehicle on each section. The higher the target speed, the higher the desired speed for any given section, resulting in an increase in flow according to the flow-speed relationships. In this way, as part of the calibration process one should establish for a particular model the influence of acceleration and breaking parameters in the capacity of the sections, namely for weaving sections. Similarly, the effects of lengths of Zones 1 and 2 in the lane change model influence the capacity of the weaving sections as well as the percent overtake and percent recover parameters influence the lane distribution, and so on.

To validate the simulation model we should be able of using as input data the same input data as in the actual model. For example, in the case of the I-35W model, inputs to the actual model are the time dependent flows at input sections and the turning percentages at exit sections. Assuming that these input data are known (i.e. measurements of input flows every five minutes) these will be the data used to define the inputs to the simulation model.

The statistical methods and techniques for validating simulation models are clearly exposed in most textbooks and specialized papers (27-28). What follows is an adaptation of the general process to our problem of validating a microscopic simulation model. The measured data in the actual system should be split in two data sets: the data set that will be used to develop and calibrate the model, and a separate data set that will be used for the validation test.

At each step in the iterative validation process a simulation experiment will be conducted. Each of these simulation experiments will be defined by the data input to the simulation model and the set of values of the model parameters that identify the experiment. The output of the simulation experiment will be a set of simulated values of the variables of interest, in this case study the flows measured at each traffic detector in the road network at each sampling interval. For example, assuming that in the definition of the simulation experiment the sampling interval is five minutes, that is the model statistics are gathered every five minutes, and that the sampling variable is the simulated flow \( w \), the output of the simulation model will be characterized by the set of values \( w_{ij} \) of the simulated flow at detector \( i \) at time \( j \), where index \( i \) identifies the detector (\( i=1,2,\ldots,n \), being \( n \) the number of detectors), and index \( j \) the sampling interval (\( j=1,2,\ldots,m \), being \( m \) the number of sampling intervals in the simulation horizon \( T \)). If \( v_{ij} \) are the corresponding actual model measures for detector \( i \) at sampling interval \( j \), a typical statistical technique to validate the model would be compare both series of observations to determine if they are close enough. For detector \( i \) the comparison could be based on testing whether the difference

\[
d_i = w_{ij} - v_{ij}, \quad j=1,\ldots,m
\]
has a mean \( \overline{d}_i \) significantly different from zero or not. This can be determined using the t-statistics:

\[
\hat{t}_{m-1} = \frac{\overline{d}_i - \delta_i}{\overline{s}_d / \sqrt{m}}
\]

where \( \delta_i \) is the expected value of \( \overline{d}_i \) and \( \overline{s}_d \) the standard deviation of \( \overline{d}_i \), for testing the null hypothesis:

\[H_0 : \delta_i = 0 \quad |\hat{t}_{m-1}| > t_{m-1;\alpha/2}\]

a. If for \( \delta_i = 0 \) the calculated value \( \hat{t}_{m-1} \) of the Student's t distribution is significant to the specified significance level \( \alpha \) then we have to conclude that the model is not reproducing close enough the system behavior and then we have to reject the model.

b. If \( \delta_i = 0 \) gives a non-significant \( \hat{t}_{m-1} \) then we conclude that the simulated and the real means are “practically” the same so the simulation is “valid enough”.

This process will be repeated for each of the \( n \) detectors. The model is accepted when all detectors (or a specific subset of detectors, depending on the model purposes and taking into account that the simulation is only a model, and therefore an approximation, so \( \delta_i \) will never be exactly zero) pass the test.

So far the statistical method however, there is some special considerations to take into account (28) namely in the case of the traffic simulation analysis.

1. The statistical procedure assumes identically and independently distributed (i.i.d) observations whereas the actual system measures and the corresponding simulated output for a time series. Therefore it would be desirable that at least the \( m \) paired (correlated) differences \( d_i = w_{ij} - v_{ij}, j=1,\ldots,m \) are i.i.d. This can be achieved when the \( w_{ij} \) and the \( v_{ij} \) are average values of independently replicated experiments.

2. The bigger the sample is, the smaller the critical value \( \hat{t}_{m-1;\alpha/2} \) is, and this implies that a simulation model has a higher chance of being rejected at the sample grows bigger. Therefore the t statistics may be significant and yet unimportant if the sample is very large, and the simulation model be good enough for practical purposes.

These considerations lead to recommend not relay in only one type of statistical test for validating the simulation model. An alternative test is to check whether \( w \) and \( v \) are positively correlated, that is test the significance of the null hypothesis:

\[H_0 : \rho > 0 \quad (\rho \text{ linear correlation coefficient})\]

This represents a less stringent validation tests accepting that simulated an real responses do not necessarily have the same mean and that what is significant is whether or not they are positively correlated. The test can be implemented using the ordinary least squares technique to estimate the regression model:

\[E(v|w) = \beta_0 + \beta_1 w + \varepsilon \quad (\varepsilon \text{ random error term})\]

The test concerns then the one-sided hypothesis \( H_0 : \beta_1 \leq 0 \). The null hypothesis is rejected and the simulation model accepted if there is strong evidence that the simulated and the real responses are positively correlated. The variance analysis of the regression model is the usual way of implementing this test. This test may be strengthened, becoming equivalent to the first test if this hypothesis is replaced by the composite hypothesis \( H_0 :\)
\( \beta_0 = 0 \) and \( \beta_1 = 1 \), implying that the means of the actual measurements and the simulated responses are identical and when a systems measurement exceeds its mean then the simulated observation exceeds its mean too.

A third family of statistical tests for the validation of simulation model is rooted in the former observation that the measured and the simulated series, \( v_i \) and \( w_i \) respectively, are time series. In this case the measured series could be interpreted as the original one and the simulated series the “prediction” of the observed series. In that case the quality of the simulation model could be established in terms of the quality of the prediction, and that would mean to resort to time series forecasting techniques for that purpose. If consider that what is observed as output of the system as well as output of the model representing the system is dependent on two type of components: the functional relationships governing the system (the pattern) and the randomness (the error), and that the measured as well as the observed data are related to these components by the relationship:

\[
\text{Data} = \text{pattern} + \text{error}
\]

Then the critical task in forecasting can be interpreted in terms of separating the pattern from the error component so that the former can be used for forecasting. The general procedure for estimating the pattern of a relationship is through fitting some functional form in such a way as to minimize the error component. A way of achieving that could be through the regression analysis as in the former test.

If for detector \( i \) the error of the \( j \)-th “prediction” is \( d_i = w_i - v_i, j=1,\ldots,m \), then a typical way of estimating the error of the predictions for the detector \( i \) is “Root Mean Square Error”, \( \text{rms}_i \), defined by:

\[
\text{rms}_i = \sqrt{\frac{1}{m} \sum_{j=1}^{m} (w_i - v_i)^2}
\]

This error estimate has been perhaps the most used in traffic simulation, and although obviously the smaller \( \text{rms}_i \) is the better the model is, it has a quite important drawback, as far as it squares the error it emphasizes large errors. Therefore it would be helpful to have a measure that considers both, the disproportionate weight of large errors and provides a basis for comparison with other methods.

Theil’s U-Statistic is the measure achieving these objectives. In general, if \( X_i \) is the observed and \( Y_i \) the forecasted series, \( j = 1,\ldots,m \), then Theil’s U-Statistic is defined as:

\[
U = \frac{\sum_{j=1}^{m-1} (\text{FRC}_{j+1} - \text{ARC}_{j+1})^2}{(m-1) \sum_{j=1}^{m-1} (\text{ARC}_{j+1})^2} - \frac{\sum_{j=1}^{m-1} (\text{ARC}_{j+1})^2}{(m-1) \sum_{j=1}^{m-1} (\text{ARC}_{j+1})^2}
\]

where: \( \text{FRC}_{j+1} = \frac{Y_{j+1} - X_j}{X_j} \) is the forecasted relative change, and \( \text{ARC}_{j+1} = \frac{X_{j+1} - X_j}{X_j} \) the actual relative change. An immediate interpretation of Theil’s U-Statistic, is the following:

\( U = 0 \iff \text{FRC}_{j+1} = \text{ARC}_{j+1} \), and then the forecast is perfect

\( U = 1 \iff \text{FRC}_{j+1} = 0 \), and the forecast is as bad as possible
Therefore the closer to zero the Theil’s U-Statistics is the better the forecasted series is or, in other words, the better the simulation model. When Theil’s U-statistic is close to 1 the forecasted series, and therefore the simulation model, should be rejected.

When the forecast efficiency is based on the regression model \( \mathbb{E}(v|w) = \beta_0 + \beta_1 w + \varepsilon \) (\( \varepsilon \) random error term) the most efficient forecast would correspond to \( \beta_0 = 0 \) and \( \beta_1 = 1 \), that can be tested by the application of variance analysis to the regression model as indicated earlier. But taking into account that the average squared forecast error:

\[
D_m^2 = \frac{1}{m} \sum_{j=1}^{m} (Y_j - X_j)^2
\]

can be decomposed (Theil) in the following way:

\[
D_m^2 = \frac{1}{m} \sum_{j=1}^{m} (Y_j - X_j)^2 = (\bar{Y} - \bar{X})^2 + (S_Y - S_X)^2 + 2(1-G)S_Y S_X
\]

where \( \bar{Y} \) and \( \bar{X} \) are the sample means of the forecasted and the observed series respectively, \( S_Y \) and \( S_X \) are the sample standard deviations and \( G \) is the sample correlation coefficient between the two series, the following indices can be defined:

\[
\begin{align*}
U_M &= \frac{(\bar{Y} - \bar{X})^2}{D_m^2} \\
U_S &= \frac{(S_Y - S_X)^2}{D_m^2} \\
U_C &= \frac{2(1-G)S_Y S_X}{D_m^2}
\end{align*}
\]

\[\Rightarrow U_M + U_S + U_C = 1\]

\( U_M \) is the “Bias proportion” index and can be interpreted in terms of a measure of systematic error, \( U_S \) is the “variance proportion” index and provides an indication of the forecasted series ability to replicate the degree of variability of the original series or, in other words, the simulation model’s ability to replicate the variable of interest of the actual system. Finally \( U_C \) or “Covariance Proportion” index is a measure of the unsystematic error. The best forecasts, and hence the best simulation model, are those for which \( U_M \) and \( U_S \) do not differ significantly from zero and \( U_C \) is close to unity. It can be shown that this happens when \( \beta_0 \) and \( \beta_1 \) in the regression do not differ significantly from zero and unity respectively.

Applying these validation criteria to the I-35W motorway, the final calibration process leads to a model for which the validation process gives the following results for some selected detectors.

**Detector 429:**

The graphical comparison between the Observed and the Simulated flows measured every five minutes over the simulation horizon of six hours is depicted in Figure 17.
The visual inspection of both series reveals a high degree of agreement that is corroborated by the Theil's coefficients:

\[ U = 0.038169 \]
\[ U_M = 0.166091 \]
\[ U_S = 0.061009 \]
\[ U_C = 0.784646 \]

And a correlation coefficient \( \rho = 0.97295 \).

**Simulation experiments and results analysis**

After it was deemed that the simulator was working as close as possible to real life conditions, one day's worth of data from the above period was used for the evaluation. The experiment consisted of two test cases, one involving normal congestion levels and the other where the previous demands were uniformly increased by 20%. In each case, two simulations were performed with and without ramp control. The Measures of Effectiveness (MOEs) collected included Total Travel Time (TTT) in veh-hrs and Total Delay (TD) in veh-hrs separately for the mainline and the ramps and Total Travel (TT) in veh-km for the whole network.

**Test Results**

Before presenting the test results it should be recognized that due to the lack of sufficient data, the entire corridor was not simulated i.e. only the freeway and the ramps were included assuming no diversion. Delays were estimated by assuming a minimum speed of 10 mph above the posted speed limit in each section of the freeway, which varied from 45 to 55 mph and 45 mph on the ramps.

The effectiveness of ramp control for normal and heavy congestion was established. TTT in the mainline decreased by 46% when control was introduced under normal congestion. This can be explained by the fact that
with ramp control density remains below critical at the bottleneck. As a result, higher speeds were achieved. Total ramp delays increased substantially as expected but overall system TTT and delays were reduced by 34.61% and 61.78% respectively. For the heavy congestion case, the system TTT decreased by 24.39% and TD by 39.41%. Similar improvements were also realized in the remaining MOE's. In general, in both cases with control, higher speeds were achieved and the flow was smoother throughout the freeway as can be appreciated in figure 18 displaying the evolution of the simulated average speeds during a 160 minutes with and without adaptive ramp metering. The results of this testing simply confirmed that the ramp control strategy, improved the operating conditions on the freeway significantly on the overall system, especially with heavy congestion. This of course, was not unexpected. However, quantification of the results became a much easier task. The results of this testing simply confirmed that the ramp control strategy, improved the operating conditions on the freeway significantly on the overall system, especially with heavy congestion. This of course, was not unexpected. However, quantification of the results became a much easier task thanks to AIMSUN2.
THE SIMULATION OF VEHICLE GUIDANCE SYSTEMS

The use of microsimulation for the assessment of dynamic vehicle guidance systems has specific requirements (see [18] and [29] for details). Vehicles should follow paths between origin and destinations on the road network model, selected according to route choice models. Paths are computed using the current link travel times as link costs. Link travel times change over time according to changes in traffic conditions, and paths are recomputed timely according to the user-defined policy. A given percentage of vehicles, whose value is should be defined by the user are allowed to change dynamically the route on the route from their current position to their destinations when a better route is identified, these are the guided vehicles.

Figure 19: Probe guided vehicle (in green) tracking and available information

A microsimulator suitable for the assessment of guidance systems should have available tools to analyze the traffic flows on the used paths and should enable the user to track selected vehicles, in that way the routes used by unguided vehicles can be compared to those of the guided vehicles to draw conclusions on the benefits of guidance under various conditions. When tracking vehicles the simulator has access to very detailed vehicle information as the current position, speed, acceleration, etc. Figure 19 displays a snapshot of the animation of a simulation in which a guided vehicle approaching congestion caused by an incident (highlighted in violet) is being tracked. This information can be used to emulate the floating car data generated by the equipped vehicles. This is a key information for the assessment of vehicle guidance strategies based on the guidance information broadcast to the equipped vehicles.

In this example the vehicle tracking has been combined with the simulation of an incident, its automatic detection, management of the conflict generated by the incident and information to the users based either on an in-car equipment or on the use of VMS. The diagram in figure 20 schematizes the methodological procedure proposed for
the simulation of incident detection and management based on the EXTERNAL APPLICATIONS DLL Library. The microscopic simulation model of the site emulates the traffic conditions on the site, and generates traffic data: flows, occupancies, speeds, (travel times when required), at the sampling rate requested by the external applications (for example 30 seconds is an standard request for most automatic incident detection algorithms, [30]), with the format suitable for the technology used at the site. These traffic data feed the Incident Warning, Incident Detection and Traffic Management Modules implementing the corresponding External Applications.

The Incident Warning applications estimate an incident probability [31] that is sent as a warning to the Traffic Management System that may take it into account. The Simulation model dialogues with the Management System. Once the Incident Detection Module detects an incident, it generates an incident alarm, which is sent to the Traffic Management System. The management decisions are communicated to the simulation model through the proper dialogue.

![Diagram](image)

To conclude this summary view on the requirements of microscopic simulators for the analysis and assessment of ITS systems, we should mention that it is desirable that the simulator has available tools for the analysis of the simulation results. This comprises the in-built tools for visualizing the simulation results, as well as the possibility of storing the simulation results in an standard way (an ODBC output, for example) accessible by any of the most common statistical packages.

6. CONCLUSIONS

The examples described in this paper have been selected to illustrate the main aspects concerning the use of microscopic simulation for the analysis of traffic systems, namely those accounting for ITS applications. The examples have been selected from the many resulting from the use of AIMSUN2 in the current professional practice of its users. The study of the Auckland motorway provides a good example of the data intensiveness of microscopic simulation and how to match these requirements and verify that the model suitably fits the reality. This is an indispensable requisite before proceeding to proper simulation studies. The I-35W in Minnesota model provides a view of what can be achieved with the simulation of traffic systems involving ITS applications such as ramp control or adaptive control systems which prioritise public transport. The results are a demonstration of the abilities of AIMSUN2 to deal with these types of traffic systems.
REFERENCES


Acknowledgements

This paper summarises the work initially developed by a research team at the Laboratori d’Investigacio Operativa i Simulacio, of the Department of Statistics and Operations Research of the Universitat Politècnica de Catalunya, continued at TSS-Transport Simulation Systems. The members who have contributed to the development, implementation and testing of the simulator and the graphical user interface have been Jaume L. Ferrer, David Garcia, Jordi Casas and Pablo Barceló. The author is very grateful to Transit New Zealand and Mr. John Hughes for the kind permission to use the material from his calibration study of the AIMSUN2 model for Auckland. And to Professor Michalopoulos, John Hurdakis and Muralidhar Koka of the Department of Civil Engineering of the University of Minnesota for the data from Minnesota.