Implementation of Microscopic Modeling to Traffic Simulation for ITS Applications

Jaime Barceló¹ and Panos G. Michalopoulos²

¹. Departament d'Estadistica i Investigacio Operativa, Universitat Politecnica de Catalunya, Spain, barcelo@eio.upc.es
². Department of Civil Engineering, University of Minnesota, USA, micha001@tc.umn.edu

INTRODUCTION

The need to simulate a number of advanced ITS concepts prior to deployment has necessitated development of high performance microscopic simulators for heuristic dynamic traffic assignment, freeway corridor diversion, driver information systems including variable message signs and vehicle guidance systems, real time adaptive traffic control systems and other traffic management concepts. These needs require treatment of time dependent origin destination information as well as a detailed modeling of merging, diverging, weaving, lane changing, turning and other behavioral characteristics reflecting real life driving conditions. These include the effects of geometric (curves, grades) slow moving vehicles, dynamic route choice and decision-making based on real time information. More detailed modeling is also needed for increasingly sophisticated adaptive traffic control concepts in freeway corridors and urban street networks. Of equal importance is the implementation of this modeling into traffic simulators that can be effectively used in practice.

In this paper a microscopic model treating such complexities is presented along with the methodology for implementation. The modeling has been incorporated into a PC-based computer program called AIMSUN2 (1-2), developed as part of various projects funded through European ITS Programs. AIMSUN2 was implemented in numerous large cities in Europe (Barcelona, London, Amsterdam, Stockholm, Milan and others), Australia and New Zealand (Auckland, Brisbane), South America (Santiago) and North America (Montreal). Since its introduction in 1989 the program underwent numerous tests, which resulted in substantial improvements. These improvements along with numerous recommendations from its current users (over 40 public agencies, Universities and consulting firms) are presented here along with test results from third parties.

One of the most important features of the program is its interactive graphical user interface, which is based on modern CAD developments and human factors design that minimize the data entry effort, and increase the accuracy of the geometry description/representation. The program also includes a set of automatic debugging features, which make it attractive to the user. Other relevant features include the ability to communicate with almost any adaptive control system including actuated signals, and time varying O-D patterns. The latter are resulting from the driver’s need to select alternative routes in real time as traffic conditions change. Depending on the level of information available drivers can dynamically change the route en route. Simulation of such behavior caused by variable message signs or guidance systems is possible with AIMSUN2.

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. 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. The AIMSUN2 car following model evolved after the seminal Gipps (3) model, which was improved to meet these requirements. The first improvement is related to the vehicle speed used in the Gipps model. In AIMSUN2 implementation this is the desired speed of the vehicle for the current section, and is therefore a local parameter. Which is calculated according to a procedure which takes into account the target speed for the vehicle’s driver, his degree of accomplishment of the speed limits on the section, the influence that certain number of vehicles (a user-defined parameter) driving slower in the adjacent right-side lane—or left-side lane, when driving on the left—, may have on the vehicle, and the influence of the section grade in the vehicle movement.

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; specially relevant were the results for the benchmark test performed using the data provided by the research group from Robert Bosch GmbH (4-5). The error metric used to measure the accuracy of the fitting between measured and simulated values was: 

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

where \(d_{sim}\) is the distance of the simulated vehicle, \(d_{meas}\) is the distance measured with the test vehicle, and \(\log\) denotes the logarithm base 10. 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 (5) for details).

Lane Changing Model

The lane change model in AIMSUN2 can also be considered as a further evolution of the Gipps lane change model (6). 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. The parameters governing the lane change process are not only local, depending on the location of the vehicle on the road network, but also variable depending on the traffic conditions (i.e. specified in seconds make that the distances for the lane change maneuvers are shorter or longer as a function of the current section average speed).

A special Lane Changing Modeling is applied at entrance ramps taking into account an additional parameter which represents 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. 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 that determines how long a vehicle is willing to wait before getting impatient. 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.

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. 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 (7). 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. 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.

**Section Cost Functions**

The section cost functions are used for calculating the shortest path trees, the cost function represents the section travel time in seconds, including the penalty of the turning movement, if it exists. Its computation can also take into account the capacity weight, 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 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. 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 (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:

$$Cost_{s,i,vt} = CurrentCost_{s,i,vt} + CurrentCost_{s,i,vt} \times \phi \times (1 - CT_{s,i} / MaxCapacity)$$

The simulation process based on the time dependent routes is as follows:

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. 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.

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 (8) 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.

**A CASE STUDY ON EVALUATION OF RAMP CONTROL LOGIC PERFORMANCE USING AIMSUN2**

An example of a feasibility study (9) 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.

**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.

Control Plan Interface (CPI) Description
The CPI is an interface that integrates AIMSUN2 with an external user defined ramp control logic. It
facilitates the exchange of information between the simulator and an external control scheme. This is
especially needed for simulating adaptive ramp control strategies which use real time traffic
measurements to determine current metering rates. Such measurements might be volume, occupancy,
speed on the mainline as well as queue lengths on the ramps. The simulator provides the necessary
measurements (Simulation Detection Data) which the CPI transfers to the external control logic. In its
turn, the control logic calculates the new rates and transfers them to the simulator through the CPI.The
process of information exchange between the simulator and the external application is conceptually
illustrated in Figure 1.

![Control & Management Actions](Simulated Detection Data) - CPI (DLL) INTERFACE - EXTERNAL CONTROL LOGIC (Traffic Control or Traffic Management System)

Figure 1: Process of information exchange between the traffic simulation model and a real-time traffic
control system

In general, most of the simulators have been designed to include one or more known ramp control
schemes. This allows the user to compare the available schemes within a single simulator, but it doesn’t
allow comparison with other non-supported schemes. With the CPI, the user has the capability of easily
programming any control logic without having to change the core of the simulator.

AIMSUN2 is capable of communicating with an external application. The CPI enhances this ability by
grouping the necessary information specifically needed for ramp control schemes. In addition, new
functions were added to allow easier access to the simulator. This makes the job of the end user easier
because he has more tools available to integrate his control logic with the simulator. Specifically, in the
CPI the notion of detector stations was added as most of the current ramp control strategies require
measurements over all lanes of the mainline instead of lane by lane. Additionally, the user is now able
to define the collection rate of measurements which can now be different from the one specified in the
simulator. For example, the simulator may collect lane by lane detector data every 30 seconds but the
user’s ramp control logic could request and receive 5 minute detector station data. With the original
external interface the user’s logic had to be designed specifically for the network under consideration.
With the CPI the user may access information at run-time about the road geometry, traffic detection and
control devices as well as their mode of operation. This allows the user to design his logic in a more
generic way and be able to use it on a variety of different networks. Finally, the implementation allows
for customized output to be saved including information specific to the operation, effectiveness and
general performance of the control logic. The CPI has been developed on Visual C++, using Microsoft
Foundation Classes (MFC) and it runs on Windows 95/NT machines. It has the form of a Dynamic Link Library (DLL) that the user attaches to the simulator.

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. With the improvements described earlier in this paper, the volume entry took under half an hour versus 72 hours manually.

Table 1 summarizes the effectiveness of ramp control for normal and heavy congestion. As can be seen, 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 (Table 1). 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 2. 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.

REFERENCES

<table>
<thead>
<tr>
<th>MOEs</th>
<th>Case</th>
<th></th>
<th></th>
<th>%Difference</th>
<th></th>
<th></th>
<th>%Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Normal Congestion</strong></td>
<td><strong>Heavy Congestion</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Travel Time</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td></td>
</tr>
<tr>
<td>TTT (veh-hrs)</td>
<td>29037</td>
<td>15627</td>
<td>-46.18</td>
<td>41231</td>
<td>27501</td>
<td>-33.30</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1948</td>
<td>4634</td>
<td>137.86</td>
<td>3425</td>
<td>6263</td>
<td>82.86</td>
<td></td>
</tr>
<tr>
<td></td>
<td>30985</td>
<td>20261</td>
<td>-34.61</td>
<td>44656</td>
<td>33764</td>
<td>-24.39</td>
<td></td>
</tr>
<tr>
<td>Total Delay</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td></td>
</tr>
<tr>
<td>TD (veh-hrs)</td>
<td>16844</td>
<td>3128</td>
<td>-81.43</td>
<td>29720</td>
<td>14124</td>
<td>-52.48</td>
<td></td>
</tr>
<tr>
<td></td>
<td>935</td>
<td>3667</td>
<td>292.36</td>
<td>2730</td>
<td>5535</td>
<td>102.77</td>
<td></td>
</tr>
<tr>
<td></td>
<td>17779</td>
<td>6975</td>
<td>-61.78</td>
<td>32450</td>
<td>19659</td>
<td>-39.41</td>
<td></td>
</tr>
<tr>
<td>Speed (km/hr)</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td></td>
</tr>
<tr>
<td></td>
<td>59.6</td>
<td>88.55</td>
<td>47.85</td>
<td>45</td>
<td>68</td>
<td>52.16</td>
<td></td>
</tr>
<tr>
<td># of Stops</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td>No Control</td>
<td>With Control</td>
<td>%Difference</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2933764</td>
<td>266572</td>
<td>-90.91</td>
<td>6163727</td>
<td>1913605</td>
<td>-68.95</td>
<td></td>
</tr>
<tr>
<td>Total Travel</td>
<td>System</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TT (veh-km)</td>
<td>698960</td>
<td>716591</td>
<td>2.52</td>
<td>656250</td>
<td>761870</td>
<td>16.09</td>
<td></td>
</tr>
</tbody>
</table>

*Table 1. Ramp Control Strategy Evaluation Results*