

MODELLING ADVANCED TRANSPORT TELEMATIC APPLICATIONS WITH MICROSCOPIC SIMULATORS: THE CASE OF AIMSUN2

J. Barceló¹, J. Casas², J.L. Ferrer¹ and D. García²

¹ Laboratori de Simulació i Investigació Operativa, Departament d'Estadística i Investigació Operativa, Universitat Politècnica de Catalunya, Barcelona, Spain, e-mail: lios@eio.upc.es

² TSS-Transport Simulation Systems, Barcelona, Spain, e-mail: info@tss-bcn.com

Abstract

The simulation of Advanced Transport Telematic Applications requires specific modelling features which have not been usually taken into account in the design of microscopic traffic simulation models. This paper discusses the general requirements of some of these applications, and describes how have they been implemented in the microscopic traffic simulator AIMSUN2.

1. Introduction

Microscopic traffic simulators are simulation tools that emulate realistically the flow of vehicles on a road network. The main modelling components of a microscopic traffic simulation model are: an accurate representation of the road network geometry, a detailed modelling of individual vehicles behaviour, and an explicit reproduction of traffic control plans.

The primary attention has been paid usually to the proper modelling and calibration of all these model components, namely the car-following, gap acceptance, lane change, and other internal models which along with other modelling parameters accounting for attributes of the physical system entities, allow the microscopic simulation model reproduce flow, speed, occupancies, travel time, average queue lengths, etc. with enough accuracy to consider the model valid.

The advent of the Advanced Transport Telematic Applications made possible by combining the developments in informatics and telecommunications, and applying them to transportation problems, has created new objectives and requirements for micro-simulation models. Quoting from Deliverable D3 of the SMARTTEST Project (SMARTTEST, 1997): "The objective of micro-simulation models is essentially, from the model designers 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 current trend in the development of Advanced Transport Telematic Applications, either real-time adaptive, or based on other specific approaches, is far from being standardised. It is therefore an exercise of dubious utility to try to integrate them in a fixed way in a microscopic traffic simulator. The relative gain achieved by including any of these, as an in-built function of the microsimulator is limited to simulating, on an easier way, those road networks on which the selected application is operating. However there would be no means of simulating other systems with that microsimulator. This is true whenever we address the problem of simulating adaptive traffic control systems as, for example, SCOOT, SCATS, vehicle actuated, control systems giving priority to public transport, etc., Advanced Traffic Management Systems (using VMS, traffic calming strategies, ramp metering policies, etc), Vehicle Guidance Systems, Public Transport Vehicle Scheduling and Control Systems or applications aimed at estimating the environmental impacts of pollutant emissions, and energy consumes. The main question then is: How can these Advanced Transport Telematic Applications be properly evaluated and tested by simulation?

From a conceptual point of view the operation of these modern systems can be described as follows: for certain applications the road network is suitably equipped with traffic detectors of various technologies (loop detectors, image processing detectors, etc.), with a specific layout depending on the requirements of the control approach. They supply the necessary real-time traffic data (flows, speeds, occupancies, etc) with the required degree of aggregation. These real-time traffic measurements feed the logic of the traffic control or management system which, after suitable processing, makes ad hoc control decisions: e.g. extend the green phase, change to the red phase, apply some traffic calming strategies, etc.. Other applications as, for example vehicle guidance, public

transport monitoring systems, or the evaluation of environmental impacts, require the access to vehicle data (position, speed, acceleration, etc.), to emulate the up-link messages in vehicle guidance applications, the vehicle tracking for the public transport monitoring or fleet management systems, or simply to provide the required data for certain fuel consumption or pollutant emissions models.

To evaluate and test any of these systems a microsimulator must be capable of incorporating in the model the corresponding traffic devices as objects: i.e. detectors, traffic lights, VMS, etc. It must also emulate their functions: provide the specific traffic measurements at the required time intervals, increase the phase timing in a given amount of time, implement a traffic calming strategy (slow down the speed on a road section, recommend an alternative route, etc). How can such evaluations be done by simulation without explicit in-built modelling of the specific Advanced Telematic Application?

2. The GETRAM/AIMSUN2 Microsimulator

2.1 GETRAM

GETRAM (Generic Environment for Traffic Analysis and Modelling), (Grau and Barceló, 1993), (Barceló et al., 1994), is a simulation environment comprising a traffic network graphical editor (TEDI), a microscopic traffic simulator (AIMSUN2), a network data base, a module for storing results and an Application Programming Interface to aid interfacing to assignment models and other simulation models, as for example the macroscopic traffic assignment EMME/2 system, (INRO, 1996).

2.2 AIMSUN2 microscopic simulator

AIMSUN2 (Advanced Interactive Microscopic Simulator for Urban and Non-Urban Networks), (Ferrer and Barceló, 1993), (Barceló and Ferrer, 1997), is a microscopic traffic simulator whose main features are:

- AIMSUN2 can deal with different traffic networks: urban networks, freeways, highways, ring roads, arterial and any combination of them.
- Two different types of simulation are involved: one based on input traffic flows and turning proportions, and one based on O-D matrices and route selection models. In the former, vehicles are distributed stochastically around the network, whereas in the latter vehicles are assigned to specific routes from the start of their journey to their destination.
- Different types of traffic control can be modelled: traffic signals, junctions without traffic signals (give way or stop signs) and ramp metering.
- Vehicle behaviour models (car following, lane change, gap acceptance, etc.) are function of several parameters that allow modelling of different types of vehicles: cars, buses, trucks, etc. They can be classified into groups, and reserved lanes for given groups can also be taken into account.
- Due to the detailed modelling of each vehicle in the network, AIMSUN2 can simulate any kind of measurable traffic detector: counts, occupancy and speed.
- AIMSUN2 provides detailed statistical output: flows, speeds, travel times, etc., which may be presented as printouts or plots.
- The model building process is a straightforward task thanks to the graphical network editor, Tedi.
- AIMSUN2 has a user-friendly interface through which the user can define the simulation experiment. It also provides a picture of the network and an animated representation of the vehicles in it. The user has an overview of what is happening in the network that aids performance analysis.
- Through the interface, the user may access any information in the model and define traffic incidents before or during the simulation run. A list of incidents may be stored for use in subsequent simulation runs.

2.3 TEDI network editor

TEDI is a graphical editor for traffic networks. It has been designed with the aim of making the process of network data entry user-friendly. Its main function is the construction of traffic models with which to feed traffic simulators like AIMSUN2. To facilitate this task the editor accepts as a background a graphical description of the network area, so sections and nodes can be built subsequently into the foreground. Its main features are:

- The editor supports both *urban* and *interurban* roads, which means that the level of detail covers elements such as side lanes, entrance and exit ramps, intersections, traffic lights and ramp metering.

- The geometry of the links is specified at the microscopic level, but the editor's ease of use makes it as fast as specifying one-dimensional links in some macroscopic systems, and nodes can be created automatically.
- The richness of parameter available for characterising the different types of objects and traffic conditions means that the only limitation to the precision of the model is the quantity and accuracy of the data collected.
- TEDI has an interface to the EMME/2 DATA BANK, providing the means to complement a macroscopic analysis effortlessly with a microscopic one using the same traffic data (i.e. O/D matrices).
- The user can define a hierarchical tree of views so that a traffic model can be restricted to one of these views.
- The editor is designed for the level of detail required by regional, intermediate and local areas, with increased modelling detail.
- It allows a totally user-defined network state, thus allowing simulations to run in the absence of assignment results.
- A library of high-level, object-based application programming functions, named TDFunctions, assists the development of interactive external applications, and, in general to access any data. The TDFunctions enable objects in the network to be read and manipulated or can restrict the view to a sub-area. Results storage and control plans are also accessed with these functions.

3. GETRAM/AIMSUN2 Extensions

To cope with the requirements of simulating Advanced Transport Telematic Applications specific extensions to GETRAM/AIMSUN2 have been developed. These extensions fall into three categories:

- 3.1 Adaptive Traffic Control, Traffic Management Systems and Incident Management Systems
- 3.2 Vehicle Guidance, Fuel Consumption and Emissions
- 3.3 Public Vehicle Scheduling and Control Systems

The approach taken in GETRAM/AIMSUN2 consists of considering the Advanced Telematic Application to be tested as an EXTERNAL APPLICATION that can communicate with GETRAM/AIMSUN2. An ad hoc version of AIMSUN2 including a set of DLL has been developed for this purpose. This library gives AIMSUN2 the ability to communicate with almost any of the above-mentioned external applications.

Using the TEDI & AIMSUN2 functions the detector, VMS and traffic lights can be modelled and their attributes defined. The process of information exchange between AIMSUN2 and the external application is shown in Figure 1:

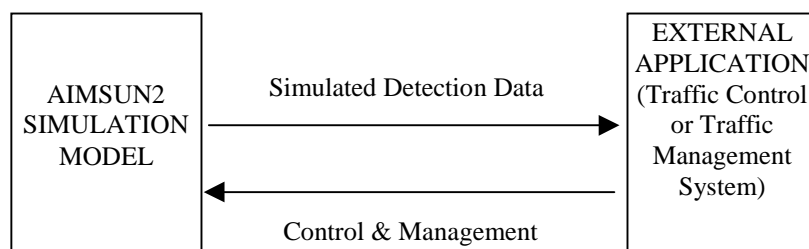


Figure 1: Process of information exchange

The AIMSUN2 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. Another set of DLL function enables the user to access the information on each vehicle state (position, speed, acceleration, etc.) at each simulation cycle.

3.1 Simulating Management and Control actions

Three main types of actions, as a result of the actuation of EXTERNAL APPLICATION on the simulation, are taken into account:

- 1) Actuate control of traffic lights and ramp metering

- 2) Actuate control of ramp metering
- 3) Supply information to the driver using Variable Message Signs (VMS)

3.1.1 - Traffic Control Actions

AIMSUN2 takes into account two types of traffic control: traffic lights and ramp metering. The first is considered for urban type intersection nodes, while the second one is for controlling freeway entrance ramps.

3.1.2. - Traffic Lights

For the intersection control, a phase-based approach is applied in which the cycle of the junction is divided into phases where each one has a particular set of signal groups with right of way (a signal group is considered as one traffic light). During the simulation of a scenario, AIMSUN2 executes a fixed control plan taking into account the phase modelling for each junction. An EXTERNAL APPLICATION can modify this execution by means of different actions. The available actions are to:

1. Change the duration of each phase: The EXTERNAL APPLICATION can increase or decrease the duration, but the control plan structure is not modified.
2. Disable the fixed control plan structure: The EXTERNAL APPLICATION disables the structure of the control plan and completely controls the phase changing.
3. Change the current phase: The EXTERNAL APPLICATION can change the current phase to another. If the fixed control plan (timings) of the junction have not been disabled, AIMSUN2 programmes the next changing of phase taking into account the duration of the phases. Otherwise AIMSUN2 holds the new phase until the EXTERNAL APPLICATION changes it to another.

3.1.3. - Ramp Metering

AIMSUN2 also incorporates ramp-metering control. This type of control is used to limit the input flow to certain roads or freeways in order to maintain certain smooth traffic conditions. The objective is to ensure that entrance demand never surpasses the capacity of the main road.

AIMSUN2 considers three types of ramp metering depending on the implementation and the parameters that characterise it:

1. Green time metering, with parameters green time and cycle time. It is modelled as a traffic light.
2. Flow metering, with parameters platoon length and flow (veh/h). The meter is automatically regulated in order to permit the entrance of a certain maximum number of vehicles per hour.
3. Delay metering, with parameters mean delay time and its standard deviation. It is used to model the stopped vehicles due to some control facility, such as a toll or a customs checkpoint.

The EXTERNAL APPLICATION can modify this modelling by different actions. It can:

1. Change the parameters of a metering: The EXTERNAL APPLICATION can dynamically modify the parameters that define a ramp metering.
2. Disable the control structure: EXTERNAL APPLICATION disable the structure of the ramp metering and completely controls the state changing.
3. Change the state of a metering: The EXTERNAL APPLICATION can change the current state to another. If the metering has not disabled the control, AIMSUN2 programmes the next changing of state taking into account the parameters, which define the control. Otherwise AIMSUN2 holds the new state until the EXTERNAL APPLICATION changes it to another.

3.1.4 Variable Message Signs

Providing information to drivers is a possible action of a Traffic Management System on a road network equipped with Variable Message Sign infrastructure. Messages may inform drivers about the presence of incidents, congestion ahead or suggest alternative routes. AIMSUN2 takes into account the modelling of

Variable Message Sign (VMS) as defined in TEDI by means of a dialogue including the following information:

- VMS name (a string of characters) or identification code.
- Position in the section as the distance from the entrance point of the section to the VMS.
- Activated message, if any.
- List of feasible messages for this VMS.
- List of all Actions available for this network associated to the messages.

Messages in a VMS may be activated in two different ways:

1. Directly through the user interface, clicking on it and pressing the 'Activate Message' button. Any message from the messages list box may be activated.
2. Through the communication interface, any external system may activate a message on any VMS of the network, by sending the corresponding command, which consists of the VMS identifier and the message text.

In both cases, it will cause the message to be displayed as Activated Message and the Actions associated with it to be implemented. Each message has a list of Actions associated with it which appear in the list box named 'Mess Actions' in the VMS Information Window. The list box named 'Actions' contains all actions available for this network. The user may add/remove action to/from the list of actions associated with a message using the bottom part of the window. An Action represents the expected impact a message has on driver's behaviour. When simulating on the input flows and turning proportions basis, the user must define actions for each possible message, as modelling assumptions. The user may access the Actions information through the AIMSUN2 user interface.

The Actions Information Window has four areas. The first area contains the list of sections where the action will have effect. The user may add or remove sections using the 'Add' and 'Remove' buttons. The other three correspond to the three types of actions:

1. Modifications of the speed limit
2. Modification of the input flow
3. Modifications of the turning proportions

When simulating with the Route Based option, the user may access the Actions information through the ad hoc AIMSUN2 interface. An action has effect on a set of sections. In each section it is possible to define the impact in two ways:

- Modification of Speed Limit of the section
- Re-routing: The possibility of altering the vehicle's path. This effect is accomplished by defining the next turn and/or defining a new destination

The re-routing effect is defined by the following for each modality independently:

- **Compliance level (δ):** This parameter gives the compliance level of the action. If $\delta=1$ then it causes the re-routing to be followed by all vehicles (i.e. it is obligatory). If $\delta=0$ then the re-routing action will be followed depending on the driver behavioural parameter (a local parameter of it), i.e. it is an information only.
When $0 < \delta < 1$, δ gives the level of acceptance, e.g. it is advice
- **Modify the next turning:** Change, which is the next turning that the vehicle must follow. This action is defined taking into account the destination
- **Modify destination:** When a vehicle enters into a section affected by an action, the simulator changes its destination

The EXTERNAL APPLICATION can actuate in the following way:

1. Consult the current message activated in a VMS: EXTERNAL APPLICATION can consult which message is activated in a VMS.
2. Activate a message in a VMS: EXTERNAL APPLICATION can activate a message, AIMSUN2 executes all actions defined by the message.

3.1.5 Detector Measurements

Detection output data is produced by AIMSUN2 periodically, provided that there are detectors defined in the network and the user has set the Model Detection on in the Detection Parameter Window before running the simulation experiment and has defined a detection interval. Currently there are two main types of Detection implemented: Common Detection Model and EXTERNAL APPLICATION type Detection Model.

In the Common Detection Model, the data produced depends on the measuring capabilities of the detectors. There is a data line for each detector, which contains the detector identifier and the list of measures gathered. They may be Count (number of vehicles per interval), Occupancy (percentage of time the vehicle is on the detector), Speed (mean speed for vehicles crossing the detector) and Presence (if a vehicle has been on the detector, it is set to 1). These data are stored in ASCII files.

In the EXTERNAL APPLICATION type Detection Model, the measures are given at every simulation step or aggregated each detection interval. The gathered measures are:

- Count: Number of vehicles
- Speed: Mean speed for vehicles crossing the detector.
- Occupancy: Occupancy (percentage of time the detector is pressed).
- Presence: Whether a vehicle is over the detector or not.

The EXTERNAL APPLICATION can undertake the following actions with detectors:

1. Consult the number of detectors in the network.
2. Consult the name of each detector.
3. Consult the detection interval.
4. Consult the detector measures gathered in each simulation step.
5. Consult the detector measures aggregated.

3.2 Simulating incidents and incident management

The diagram in figure 2 schematises the methodological procedure proposed for the simulation of incident detection and management based on the EXTERNAL APPLICATIONS. The procedure works in the following way:

- A microscopic simulation model of a site emulates 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 a standard request for most automatic incident detection algorithms), with the format proper of the technology used at the site.
- These traffic data feed the Incident Warning, Incident Detection and Traffic Management Modules implementing the corresponding External Applications. Information exchange is based on the DLL functions.
- The Incident Warning applications estimate an incident probability (Wilmink and Immers, 1995) 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, as an external application, in the way described above.
- 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 as described above.

3.3 Simulation of Vehicle Guidance

Vehicle Guidance can also be modelled as an External Application that can be properly simulated with AIMSUN2 by means of an exchange of information using the suitable DLL functions. Essentially the simulation of a Vehicle Guidance system, (Jeffery, 1987), consists of an exchange of information between the equipped car and the Traffic Information Centre. The on board equipment collects information on the vehicle's position, travel time, speed, experienced delay, number of stops, etc. which is sent to the Traffic Information

Centre, (SOCRATES, 1990), (SOCRATES, 1992), by means of the telecommunication technology on which the system is based, as for example, beacons or GSM, or both.

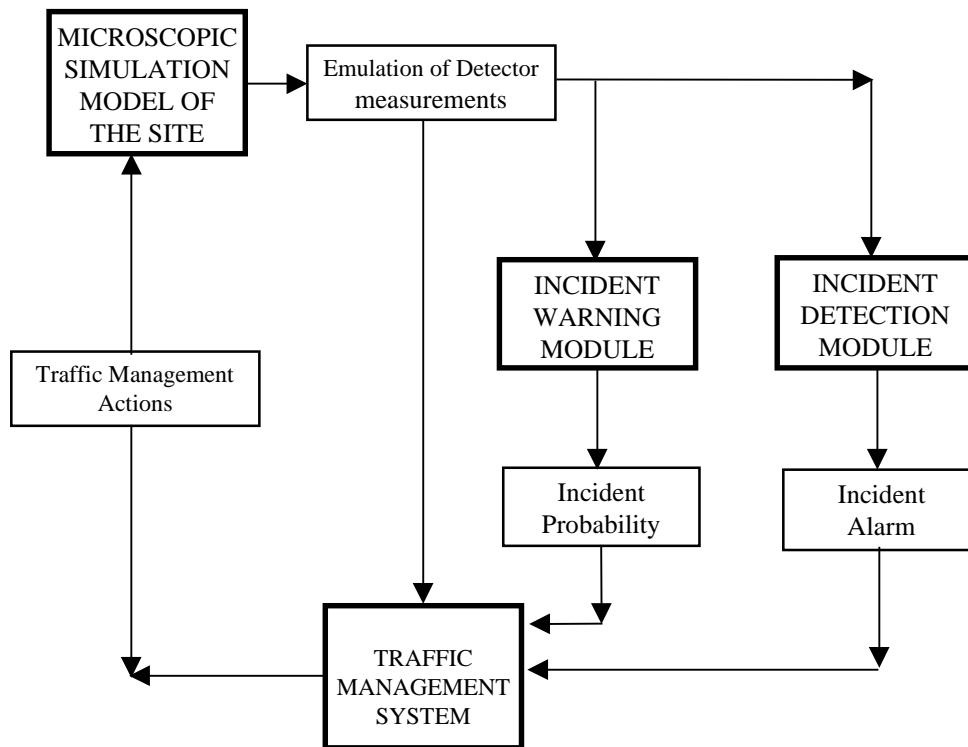


Figure 2.

The vehicle data collection can be modelled as follows: each equipped vehicle sends the information to the system when passing through certain points on the network. In the network description the definition of these Data Collection Points (DCP) has been included, they are identified by the section (i.e link) and position in the section. A DCP works like a detector. Each time the position of a guided vehicle is updated it is checked whether or not it has passed through a DCP. If so, a message to the information centre is sent and the vehicle information is updated. Alternative data collection procedures can be implemented for the simulation, as for example assuming that there are a set of fixed DCP which correspond to the end of each section, or that DCP are variable and their position depends on the behaviour of each guided vehicle, for instance, if it takes more than certain time for a vehicle to cross a section or if it has to stop during a section journey. Examples of floating car data collection sampling procedures are:

- (1) Section basis: a vehicle sends a message whenever he reach the end of any section in the network
- (2) Time basis: a vehicle sends a message every certain time interval whose length can be selected by the user as a simulation input parameter.
- (3) Dual mode sampling:
 - Time and section basis: a vehicle sends a message every time it reaches the end of a section and every time interval. It means that, if a vehicle takes more than one time interval to travel one section, a message is sent for each time interval.
 - Speed and section basis: a vehicle sends a message at the end of each section. Besides, if during the trip along a section the vehicle speed falls below a minimum value (which is set by the user as a simulation input parameter), a message is sent every time interval after this incident until the end of section is reached.

The sampling data collection process is characterised by the following parameters, which must be defined for each simulation experiment: Percentage of guided vehicles, Sampling procedure, Time interval (for the time basis and dual mode sampling procedures), Minimum Speed (for the dual mode sampling, speed basis).

Guided vehicles, which are a percentage of the total, are identified at generation time and the data collection process is initialised. For each guided vehicle the corresponding DLL functions collect the information on the vehicle data. Additionally, each time a guided vehicle is updated, it is checked whether or not it is time to transmit data. If so, the following information is updated by means of the ad hoc DLL: Number of messages sent, Number of data blocks transmitted, Sections and distance travelled since last message, Time for the next transmission. In a similar way every time a guided vehicle reaches the end of a section, the following information is updated: Number of messages sent, Number of data blocks transmitted, Distance travelled since last message, Time since last transmission.

The simulation can thus provide the following information (Barceló et al.1998):

1. Traffic volumes (guided and unguided): mean number of vehicles present in the network and total number of trips per hour.
2. Mean vehicles speed (km/h).
3. Number of stops per vehicle/kilometer.
4. Communications established: total number of transmissions between vehicles and information center, per time unit.
5. Amount of information transmitted: total number of blocks transmitted per time unit.
6. Mean time between transmissions per vehicle: how often a vehicle transmits the data collected to the center.
7. Distance travelled between transmissions per vehicle.
8. Communications per section per vehicle: for each section, how many transmissions has a vehicle done.

Items 1,2 and 3 determine the level of network congestion in the simulation experiment. Items 4 and 5 represent the communications requirements for the system. Items 6, 7 and 8 are measures about the quality of the overall information received by the centre.

The Traffic Information Centre collects the individual information from the equipped vehicles and, after a suitable processing produces the guidance information which is transmitted to the guided vehicles. The broadcasting of this information and the guided vehicle reactions can be simulated using the DLL in a similar way as the simulation of the VMS. It should be noticed that, as in the previous case, this implies that the simulator has to have the capability of dynamically re-route the guided vehicles en-route.

References

- Barceló, J., Ferrer, J.L., and Martín, R, 1998, Simulation Assisted design and Assessment of Vehicle Guidance Systems, Accepted for publication in International Transactions on Operations Research.
- Barceló, J., J.L. Ferrer (1997). *AIMSUN2: Advanced Interactive Microscopic Simulator for Urban Networks. User's Manual*. Departamento de Estadística e Investigación Operativa. Facultad de Informática. Universidad Politécnica de Cataluña.
- Barceló, J., J.L. Ferrer and R. Grau (1994). *AIMSUN2 and the GETRAM Simulation Environment. Internal Report*. Departamento de Estadística e Investigación Operativa. Facultad de Informática. Universidad Politécnica de Cataluña.
- Ferrer, J.L. and J. Barceló (1993). *AIMSUN2: Advanced Interactive Microscopic Simulator for Urban and non-urban Networks*. Internal Report. Departamento de Estadística e Investigación Operativa. Facultad de Informática. Universidad Politécnica de Cataluña.
- Grau, R. and J. Barceló (1993). *The design of GETRAM: A Generic Environment for Traffic Analysis and Modeling. Research Report DR 93/02*. Departamento de Estadística e Investigación Operativa. Facultad de Informática. Universidad Politécnica de Cataluña.
- Jeffery D.(1987). Route guidance and In-vehicle Information Systems. In: *Information Technology. Applications in Transport*, P. Bonsall and M. Bell (eds), VNU Science Press, Utrecht, pp. 319-351.
- SMARTTEST Project Deliverable D3, August 1997, European Commission, 4th Framework Programme, Transport RTD Programme, Contract N^o: RO-97-SC.1059.
- SOCRATES, 1990, DRIVE I Project V1007, Commission of the European Communities, Report on WP1.2.3, 'Floating Car Data', Prepared by Hoffmann Leiter, Universitat Politècnica de Catalunya, and BAST, responsables G.Hoffmann and J.Barceló.
- SOCRATES KERNEL 1992, DRIVE II Project V2013, Commission of the European Communities, Report SCKN/UPC 03.43.92, *Equipped Vehicle Fleet Requirements for Monitoring Network Conditions in a Dynamic Route Guidance System* Prepared by Universitat Politècnica de Catalunya, responsable J.Barceló.

Wilmink, I.R. and Immers, L.H., 1995, Deriving Incident Management Measures using Incident Probability Models and Simulation, TNO Research Report 95/NV/172, The Netherlands.