Comparison of PARAMICS and GETRAM/AIMSUN Microscopic Traffic Simulation Tools

Ruey Long Cheu#
Department of Civil Engineering
National University of Singapore
1 Engineering Drive 2, E1A #07-03
Singapore 117576
Telephone: +65-6874-2153
Email: cheu@nus.edu.sg

Yanzhi Tan
Department of Civil Engineering
National University of Singapore
1 Engineering Drive 2, E1A #07-03
Singapore 117576
Telephone: +65-6874-5035
Email: eng90772@nus.edu.sg

Der-Horng Lee
Department of Civil Engineering
National University of Singapore
1 Engineering Drive 2, E1A #07-03
Singapore 117576
Telephone: +65-6874-2153
Email: dhl@nus.edu.sg

Paper #04-2640

Reviewed by

TRB Committee A3A11 – Traffic Flow Theory and Characteristics

for presentation at the 83rd Annual Meeting of the Transportation Research Board

November 2003

# Corresponding author
Abstract: PARAMICS and GETRAM/AIMSUN are two microscopic traffic simulation tools that are capable of modelling the functions of Intelligent Transport Systems (ITS) in a large network. This paper compares the Version 4.0 of PARAMICS with Version 4.0 of GETRAM/AIMSUN, based on the experience gained in using them to model a network with a freeway and arterials. The simulation tools were compared in six aspects: user interface, network modelling capabilities, traffic behaviour, statistical output, run-time performance and other advanced features. Between the two software packages, GETRAM/AIMSUN has shown its strength in the ease of network coding, modelling bus transit systems, and basic functions in freeway traffic management system. On the other hand, PARAMICS has advantage over GETRAM/AIMSUN in its visually realistic graphical animation, better integration of the different functional modules, and offers a much more comprehensive Application Programmer Interface (API) for modelling of advanced and emerging ITS functions.
INTRODUCTION

In the transportation engineering profession, traffic simulation software is an important tool in helping to assess the performance of a traffic network with complex, stochastic but interacting components which are difficult, if not impossible to model analytically. If used correctly, simulation tools allows users to conduct in-laboratory experiments to evaluate the traffic performance of a network under the modeled scenarios in a fast and cost-effective way, without having to disrupt traffic operations in a real network and potentially compromise public safety. With the improvement in computational speed of microprocessors, size of memory and storage devices in computers and reduction in hardware cost in the past two decades, microscopic traffic simulation tools are becoming more popular. In recent years, in respond to the market demand for microscopic traffic simulation software which are capable of simulating many operational and proposed intelligent transportation systems (ITS) functions in a large network, several tools have been developed. Two of them are GETRAM and PARAMICS which are the subject of comparison in this paper.

PARAMICS (PARAllel MICroscopic Simulation) is one of the earliest microscopic traffic simulation tool developed with ITS modeling capabilities. It was originally developed by Quadstone Ltd in Edinburgh, U.K., as part of the European DRIVE project. The current version of PARAMICS is Version 4.0 which runs on most of the personal computers and UNIX operating systems (1,2). Parallel simulation or multi-processor versions exist (3,4). The entire PARAMICS Development Suite comes in six modules: Modeller, Processor, Analyzer, Monitor, Programmer and Estimator. The Modeller, being the core simulation engine, is responsible for network representation, vehicle movement and behavior, and traffic control. The PARAMICS Processor, Analyzer and Monitor assist users in batch simulation, post-simulation analysis of results, and pollution monitoring, respectively. PARAMICS Programmer provides the Application Programmer Interface (API) for users to implement customized simulation functions and connect to an external software. The latest addition is PARAMICS Estimator which back-calculates origin-destination (O-D) matrix from link traffic counts. Unless otherwise specified, the term PARAMICS in this paper may refer to the entire Development Suite, one module (especially the Modeller) or combination of several modules. PARAMICS has been used in many applications worldwide (1), and has been used by the first and third authors of this paper in many research investigations (5-10). However, the second author was new to both PARAMICS and GETRAM at the beginning of this research.

GETRAM (Generic Environment for TRaffic Analysis and Modeling) is a relatively new microscopic traffic simulation tool. It is developed and marketed by Transport Simulation Systems (TSS) in Barcelona, Spain (11,12). The GETRAM package consists of AIMSUN (Advanced Interactive Microscopic Simulation for Urban and Non-urban Networks), the Traffic network graphical EDItor (TEDI), a network database, a module for storing simulation results, AIMSUN 3D and GETRAM Extension. AIMSUN is the core simulation module, AIMSUN 3D is the three-dimensional visualization tool while GETRAM Extension provides the API gateway. The term GETRAM refers to the collective simulation environment while AIMSIN refers to only the simulation engine in GETRAM. GETRAM/AIMSUN (including a parallel simulation version) has been used by the developer in many applications in Europe (11). Papers presented in TRB Annual Meetings have so far reported applications in the evaluations of ramp metering (13), signal control (14), traffic safety (15) and combination of ITS strategies (16).

Both GETRAM/AIMSUN and PARAMICS have been highlighted and recommended for use in ITS modeling and evaluation (17). PARAMICS was given the highest score in a separate evaluation of three microscopic traffic simulation software (excluding GETRAM/AIMSUN) (18).

The objective of this paper is to perform a comparative evaluation of Version 4.0 of PARAMICS and Version 4.1 of GETRAM/AIMSUN, the latest version of the respective software packages. The evaluation is based on the authors’ experience in coding and modelling an actual network of integrated freeway and arterials. Both simulation tools were compared in terms of graphical user interface, modelling capabilities, traffic behaviour, statistical output, run-time performance and extensibilities of the applications through the API.

METHODOLOGY

Network Modelled

In this evaluation, the Clementi network in Singapore of approximately 3.0 km by 3.0 km was used as the primary test network. This network consists of a major expressway, four major arterials, four grade separated interchanges and 13 signalized intersections. The network has 12 origin-destination (O-D) zones, including one that represents a bus transit interchange. The O-D matrix was derived based on link volume between 0700 hrs to 0900 hrs on a
typical weekday, provided by the local transportation authority. Traffic signal timings provided by the authority during the same period were also coded into the models.

The Clementi network which was coded in both PARAMICS and GETRAM/AIMSUN was used to accomplish the remaining evaluation tasks. Screenshots of coded networks in PARAMICS and GETRAM are shown in Figures 1 and 2 respectively.

**Evaluation Strategy**

As mentioned, two of the authors was more familiar with PARAMICS than with GETRAM. In order to explore the functionalities of AIM SUN and evaluate its capability against PARAMICS, the following tasks were carried out.

**Task 1:** The first task was to explore the functional capabilities of GETRAM/AIMSUN. This was done through the coding of the test network in TEDI and running of simulations in AIMSUN. Through these processes, the user friendliness and modelling capabilities were assessed.

**Task 2:** The second task was to understand the traffic models used in PARAMICS Modeller and AIMSUN. In particular, the car-following, lane changing models, mechanisms of vehicle generation and traffic assignment models were examined.

**Task 3:** The purpose of this task was to explore the output that PARAMICS Modeller and AIMSUN produce, and when possible, compare their values with simulation results made with the default parameter values.

**Task 4:** The simulation run times of PARAMICS Modeller and AIMSUN were compared in an experiment using the same network and same personal computer.

Six evaluation categories were identified and evaluated in the above tasks. They are user interface, network modelling capabilities, traffic behaviour, statistical output, run-time performance and other advanced features.

**USER INTERFACE**

The evaluation of user interface focused on the ease of coding and preparing a network for simulation.

**Integration of Components**

While PARAMICS Modeller offers integrated network coding, simulation control and visualization, GETRAM users need to first use TEDI for network coding in the Windows environment, save the network in TEDI and then load it into AIMSUN and continue to edit the simulation control and if desired, use AIMSUN 3D for three-dimensional visualization. An example that illustrates PARAMICS’s advantage in network coding is the editing of signal control and immediate visualisation of the changes in the Modeller’s GUI window. PARAMICS also allows users to pause the simulation, edit the network features, and continue with the simulation run. In contrast, the network setting in an AIMSUN simulation run cannot be changed easily. Therefore, in this aspect, PARAMICS is more convenient for users to toggle instantly between editing, running and visualization modes.

**3D Objects and Animation**

In GETRAM, 3D simulations can only be viewed in a separate software (AIMSUN 3D) when the simulation is running, or during post-simulation playback. In PARAMICS Modeller, view can be toggled between 2D and 3D simply by clicking an icon. GETRAM provides build-in cosmetic tools for building and displaying 3D objects. These tools add text, polylines, polygons and circles into the network (via TEDI) to show 3D graphical objects such as buildings and trees. In PARAMICS, such added objects are not available in Modeller, but can be added through the Programmer’s API. Both software packages provide the feature of drive through 3D visualization.

**Navigation**

In GETRAM, the GUI in TEDI and AIMSUN has similar navigation style as Microsoft Windows. For instance, scroll bars are used for navigation with similar functions of the scroll bars in the Windows. On the other hand, navigation in PARAMICS requires practice for new users because the control actions via the mouse buttons do not follow the normal Windows convention. However, once a user has mastered the mouse control actions, navigation within a network, whether in 2D or 3D mode becomes very convenient.
Background Image
Both software packages allow of a set of imported background images as an aid for coding large traffic networks. TEDI and AIMSUN gives greater flexibility by accepting formats in AutoCAD dxf, bitmap bmp, jpg, tif and gif while PARAMICS supports overlay in only bmp and dxf formats.

Editor Options
The ease of coding and editing a network in PARAMCIS Modeller and TEDI is compared in three areas, namely network editing, data input and shortcut keys.

Network Editing
Network coding in TEDI is made simple with a column of buttons on the toolbar located at the left-hand-side of the window. Similar feature has also been provided in AIMSUN (see Figure 2). In addition, editing of a network object starts by double clicking on the object. In PARAMCIS Modeller, modifications of the network can only be done in the editing mode. The icons on the toolbars (that appears in the editing mode) and extended right mouse menu support provide some ease in network editing. Overall, network editing in TEDI is easier and faster.

Data Input
In TEDI, users will be entering these input data through dialog windows. Directing editing of the text files is possible, but users have to follow the file format strictly. PARAMICS users will find it easier to edit text files directly.

Shortcut Keys
In TEDI, some very useful shortcut keys are Ctrl+X, Ctrl+C and Ctrl+V which can cut, copy and paste objects. Pasted objects can be positioned by dragging to the required position using the left button mouse. In comparison, PARAMICS offers a whole series of hotkeys ranging from Ctrl+C for Edit Configuration, Ctrl+D for View Demands to Ctrl+O for Open Network and Ctrl+S for Save Network. There are more hotkeys that serve as short cuts, but the key strokes may not be intuitive for a new user who is familiar with the Windows environment.

NETWORK MODELLING CAPABILITIES
In this category, the modeling capabilities of AIMSUN and PARAMICS are compared.

Network Size Limit
Both AIMSUN and PARAMICS do not put any limit to the size of the network that can be modelled. The running speed of the programs will obviously be slower if the network size and vehicle numbers exceeds the available memory of the machine.

Simulation Mode
Both PARAMICS Modeller and AIMSUN can be executed in interactive (real-time) and in batch modes. Functions to execute batch simulations are provided within the AIMSUN program. The PARAMICS Modeller can be executed in batch mode in the MS-DOS environment through the PARAMCIS Processor. In the interactive simulation mode, both programs provide options for users to adjust the simulation speed, by specifying a delay between clock ticks in PARAMICS Modeller and by specifying the desired speed in AIMSUN.

Nodes and Intersections
In microscopic traffic simulation models, nodes are points that connect two or more links. Nodes may also be objects for the modeling of intersections where vehicle turning movements and signal timings are defined. PARAMICS allows users to define the lane to lane connections between two adjacent links. These lane specific
connections define the permitted or prohibited traffic movements at the node or intersection. This allows more detailed and realistic modeling of traffic movements at an intersection. In GETRAM/AIMSUN, the program will automatically make the lane assignments. This has provided some convenience in network coding but lack the flexibility in detailed modeling of traffic movements.

Links

Links in PARAMICS (and many other traffic simulation software) are known as sections in AIMSUN. In both PARAMICS and AIMSUN, link characteristics such as speed limit, gradient and visibility (sight) distance can be specified. In addition, lane changing parameters such as signposting distance (in PARAMICS) and zone distances (in AIMSUN) can be defined by the users. Lanes can also be closed or restricted for certain vehicle types. Links in AIMSUN and PARAMICS has differences in the following aspects.

One-Way Versus Two-Way Links

Sections created by AIMSUN are only for one-way traffic. In contrast, links in PARAMICS are two-way by default, but users may change them to one-way links by modifying the link attributes. In AIMSUN, the minimum section length is 6 m, but there is no such restriction in PARAMICS.

Curved Links

In AIMSUN, curved links are formed by a group of piecewise linear sections. In PARAMICS, when a link is identified as being curved, the Edit Curves mode enables the link to be drawn as an arc, as part of a circle. On such curved links, vehicle motion follows the curvature of the defined links. In this aspect, PARAMICS Modeller provides a precise modeling of the road geometry and smoother movement of vehicles.

Periodic Reload of Link Information

Within PARAMICS, there are time dependent files of links. They allow periodic reload of node and link data as a simulation run progresses. Lane closures, speed limits, link cost factors and turn restrictions can be modified during a simulation. This is not possible in AIMSUN as network attributes are fixed throughout a simulation run.

Lane Change Restriction

In AIMSUN, a user can impose restrictions on lane changing between two adjacent lanes in a section. Using this feature the user may avoid undesired lane changing to avoid potential vehicle conflicts. Such restriction is not available in PARAMICS. However, PARAMICS user may use a combination of turn restriction and vehicle type restriction to model the same effect.

Link Cost Functions

In both PARAMICS and AIMSUN, the default cost used for route choice is the free flow link travel time. In PARAMICS, the link cost function may include travel time, distance, toll or any weighted linear combination of them. Link travel times may be updated and shortest paths recalculated at specified intervals. In AIMSUN, an arc is defined as a series of sections between two intersections. The arc cost is a function of travel time, volume, turn penalty and vehicle type. Travel time and shortest path may also be updated at regular intervals during a simulation.

Incident Simulation

Incidents can be simulated in both PARAMICS and AIMSUN. Both software packages only allow the simulation of one-lane incident. To simulate an incident, a user has to specify the starting time and end time of the incident, link (or section) and lane number, distance from upstream node, number of vehicles (in PARAMICS) or length affected (in AIMSUN). In PARAMICS, an incident is simulated by stopping the correct number of vehicles, at the specified time and location. In AIMSUN, an incident is simulated by closure of a lane.
Public Transport Modelling

Bus Stops and Bays

Bus stops in AIMSUN can be denoted as Normal Bus Stops or Bus Bay Stops. This is a useful feature in some urban networks where bus bays are common. AIMSUN also provide a Bus Terminal feature, with fixed capacity. PARAMICS does not offer the bus bay and bus terminal features. User will have to manipulate nodes, links and vehicle type attributes to model such functions. For example, a bus bay can be modelled with additional links created only for buses.

Dwell Time

PARAMICS offers two methods for determining dwell time at bus stops. The first method uses a passenger arrival rate to calculate the stop time at a bus stop. Alternatively, dwell time may be modelled by a normal distribution, with the mean and standard deviation specified explicitly by the user. In AIMSUN, dwell time of a bus at a bus stop is modelled as a normal distribution with user defined mean and standard deviation values.

Network Equipment

Two of the most commonly used ITS equipment on freeways are variable message signs (VMS) and ramp meters. AIMSUN provides some basic features of VMS and ramp meters that can be used to influence traffic flow. In PARAMICS, limited functions are provided and users are expected to implement the customized functions through the Programmer’s API.

Variable Message Signs

The simulation of VMS has been provided in Version 4.0 of PARAMICS Modeller. However, drivers’ response to a VMS has to be explicitly model through a user developed API plug-in. In AIMSUN, VMS can be simulated, and three types of VMS responses or actions can be defined. They are modifications to the speed limit, turning proportions and re-routing actions.

Ramp Metering

AIMSUN incorporates three ramp-metering options: Green Time Metering (for a fixed time metering signal), Flow Metering (for a fixed desired entry volume) and Delay Metering (which forces a vehicle to wait at the meter with a specified delay). In PARAMICS, ramp metering is not provided with the Modeller. A user has to use a traffic signal as a ramp meter, or write his own control logic in an API plug-in.

TRAFFIC MODELLING

Traffic Demand

The traffic input into AIMSUN may be defined in two different ways: (1) by input volumes at the boundary sections, plus turning proportions at all the intersections; (2) by an O-D matrix. In PARAMICS, traffic demand can only be defined by means of an O-D matrix. However, the newly released PARAMICS Estimator may be used to convert link volume counts and turning percentages into an O-D matrix.

Both PARAMICS Modeller and AIMSUN support multiple periods within a simulation run, with each period having a different input volume (for AIMSUN) or O-D matrix. This feature is useful for long simulation hours that cover peak and off-peak traffic demand.

Vehicle Generation

In PARAMICS Modeller, the time between two successive vehicle releases is governed by the negative binomial distribution. In AIMSUN, the headway of vehicles being released from a centroid into an input section may follow an exponential, uniform or normal distribution. A user may also specify a constant headway, or “ASAP”. The
default distributions are exponential when specifying traffic demand with input volume and turning proportions, and uniform distribution when using an O-D matrix.

**Car-Following**

AIMSUN and PARAMICS use car-following and lane changing models to emulate the movements of individual vehicles once they are released into the network.

**PARAMICS Car-Following Model**

The PARAMICS car-following model has been developed over a period of 5 years from 1992 to 1997. It is based on a number of other models, but most of the components were created from scratch. The car-following model has been validated for the driving conditions in the U.K..

In PARAMICS, each vehicle is called a Driver-Vehicle Unit (DVU). Each DVU has a car-following target headway. The target headway may be adjusted by the driver’s aggression. An aggressive driver will have a smaller target headway.

The PARAMICS car-following model divides the actions taken by a DVU into three modes: braking, cruising and acceleration modes. A following vehicle is said to be in acceleration and braking modes if its lead vehicle is accelerating or decelerating at a rate beyond their respective thresholds. In both modes, the vehicle’s acceleration or deceleration follows the respective values preset by the Modeller. In the cruising mode, vehicle acceleration $a$ is governed by one of three equations:

$$a = k_2 \Delta V$$

$$a_B = k_2 \Delta V + k_1 \frac{g - t}{t}$$

$$a_C = c - \frac{(\Delta V)^2}{g - t}$$

where $\Delta V$ is the relative speed (speed of lead DVU minus the following DVU);

$c$ is termed “bunching” acceleration, $g$ is the current gap (in m);

$t$ is the distance to a target point (a traffic control device) or lead vehicle; and

$k_1$ and $k_2$ are coefficients.

Equation (1) is used when the DVU is following too closely than the target headway, and need to reduce its speed and move away from the lead vehicle. Equation (2) is applied when the lead vehicle is accelerating away, and (3) is used when the two DVUs are at a constant separation or closing in, but still larger than the target headway.

**AIMSUN Car-Following Model**

AIMSUN’s car-following model is based on the Gipps model (12). It basically has two components. The first component represents the maximum speed to which a vehicle can accelerate

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

where $V(n, t)$ is the speed of vehicle $n$ at time $t$;

$a(n)$ is the maximum acceleration of vehicle $n$;

$T$ is the reaction time; and

$V^*(n)$ is the desired speed of vehicle $n$. 

TRB 2004 Annual Meeting CD-ROM  
Paper revised from original submittal.
The second component represents the limitation imposed by the lead (\((n-1)\)th) vehicle

\[
V_b(n,t + T) = d(n)T = \sqrt{d(n)^2T^2 - d(n)} \left\{ x(n-1,t) - s(n-1) - x(n,t) \right\} - V(n,t)T - \frac{V(n-1,t)^2}{d'(n-1)}
\]  

where \( d(n) < 0 \) is the maximum deceleration desired by vehicle \( n \);
\( x(n,t) \) is the position of vehicle \( n \) at time \( t \);
\( s(n) \) is the effective length of vehicle \( n \);
\( d'(n) \) is an estimation of the \( n \)th vehicle’s desired deceleration;

At each simulation time step, \( V_{\text{max}}(n,t+T) \), the maximum desired speed of vehicle \( n \) at time \( t+T \) is the minimum of \( V_d(n,t+T) \) and \( V_b(n,t+T) \). However, the actual speed of vehicles \( v(n,t+T) \) is calculated as

\[
v(n,t+T) = \min\{V_{\text{max}}(n,t+T), S_{\text{limit}}\theta(n)\}
\]  

where \( S_{\text{limit}} \) is speed limit of the section; and
\( \theta(n) \) is the “speed acceptance” factor of vehicle \( n \), which reflects the driver’s compliance with the section speed limit. \( \theta(n) = 1 \).

**Lane Changing**

**PARAMICS Lane Changing Model**

In PARAMICS Modeller, a DVU’s turning movement is determined when it is two links upstream of an intersection. The internal logic divides each link or a series of links into 2 zones for the purpose of lane changing: Zone 1 at the upstream portion and Zone 2 at downstream portion. In Zone 1, a vehicle is not under pressure to change lane and the only objective of the lane change is to overtake a slow vehicle. In Zone 2, no overtaking is allowed, but a DVU is under pressure to move into the correct lane in time to make an appropriate turn. The boundary of these two zones in a link may be established by a signpost. The default signpost distance is 250 m for an arterial link and 750 m for a freeway link. Lane changing action follows the established gap acceptance theory and a historical record of suitable gap availability in a link. A driver’s aggression is taken into account in the lane changing model by means of adjusting his signpost distance.

**AIMSUN Lane Changing Model**

AIMSUN uses the lane changing model proposed by Gipps (12). The car following model make a decision to change lane when the three conditions of the necessity, desirability and possibility are all satisfied. When a vehicle is approaching a node where turning action is required, the lane changing model makes decisions by classifying the location of the vehicle into one of three zones, with each zone corresponds to a different urgency of lane changing motivation. The zones are:
- **Zone 1**, where lane changing decisions are governed by traffic conditions: desired speed, speeds and gaps of neighbour vehicles
- **Zone 2**, where turning priority increases and require the vehicle to look for a gap and make a lane change
- **Zone 3**, where a vehicle are forced to reach the desired lane, reducing speed to look for gap

**Route Choice**

Both PARAMICS Modeller and AIMSUN offer three types of traffic assignment methods: all-or-nothing, stochastic and dynamic feedback assignment methods. The shortest path of vehicle may be re-calculated at user defined intervals, using updated link travel times. In terms route choice, both simulation tools apparently provide the same flexibility. However, there are two differences. First, PARAMICS Modeller allows different types of vehicles to have different traffic assignment methods. This is apparently not available in AIMSUN, but a user may overcome this by changing the arc cost function that is applied to a particular vehicle type. Second, the link or arc cost functions used by PARAMICS Modeller and AIMSUN are slightly different. These have been discussed earlier in this paper.
SIMULATION OUTPUT

Format of Output Statistics

By default, both PARAMICS Modeller and AIMSUN give approximately the same type of network, nodes and link statistics in text files at user-defined intervals. However, when the ASCII file format is specified for AIMSUN, the program gives an output file for each period of accumulation of statistics, resulting in multiple ASCII files in a long simulation run. A more convenient method is to store files in Access database or in ODBC database. PARAMICS does not have the problem of multiple ASCII output files. But it does not have the option to output files in Access or ODBC format. On the other hand, PARAMICS users can make use of the Excel Wizard (in the form of Excel Add-In macro) to import the Modeller’s output file conveniently into Excel for plotting graphs and further analysis.

Colour Representation of Network Statistics

AIMSUN can colour the sections according to ranges of values for a user specified traffic measure (such as flow, speed, density, queue length, delay, and etc). In this aspect, AIMSUN provides more options as compared to PARAMICS Modeller. However, users of PARAMICS, can through the Analyzer, display graphical representations of such traffic measures and make comparisons of the output between simulation runs. The PARAMICS Analyzer apparently provides more analysis options that what is available in AIMSUN.

Dynamic Graphs

While simulation is running, dynamic graphs will be updated continuously as new data becomes available. PARAMICS Modeller offers more options in the display of a dynamic graph. Users can set the x and y-axis for different parameters, such as flow, speed and density, in a link. While in AIMSUN, only time dependent plots (i.e., time in the x-axis) are available. However, in AIMSUN, statistics of several sections can be plotted on the same graph. Multiple time plots with different variables can also be displayed at the same time in different windows.

Speed-Flow Relationship

For a comparison, the speed-flow curves of an expressway link were derived from PARAMICS Modeller and AIMSUN output, respectively. The network’s traffic demand was loaded incrementally in a two-hour simulation. The expressway link of interest had a speed limit of 90 km/h, and passenger cars were set to have a top speed of 110 km/h. Loop detector data was recorded every 60 seconds (average across all lanes). Figure 3 shows the plots of speed-flow data from PARAMICS Modeller and AIMSUN. The general shapes of speed-flow derived from PARAMICS Modeller and AIMSUN have similar shape. PARAMICS appears to have a higher capacity based on the raw detector data, but the fitted curves of PARAMICS and AIMSUN have the maximum capacity at the same point on the graph. In AIMSUN, vehicle speed at low flow is approximately at the speed limit of 90 km/h. The mean of the speed acceptance factors is 1.0. Therefore, vehicles travelled at 90 km/h at low densities as illustrated by the upper part of curve. PARAMICS Modeller gives a relatively higher speed at low flow as its car-following logic allows a vehicle to travel higher than the link speed limit, if the driver has a higher aggression value. The authors caution that the comparison is based on the default car-following and lane changing parameter values. One way of calibrating the simulation model is to adjust the model parameters such that the macroscopic traffic flow relationships (such as Figure 3) generated by simulation and field detector data matches well. However, this is beyond the scope of this paper.

RUN-TIME PERFORMANCE

An experiment was conducted to compare the relative speed of simulations made by PARAMICS Modeller and AIMSUN. The experiment was performed on the same personal computer with an Intel Pentium III 733 MHz processor and 512MB RAM. A two hour simulation of the traffic operations in the Clementi network was made, with simulation time step set to 0.5 sec. The base O-D matrix was based on the morning peak traffic demand from 7:00 a.m. to 9:00 a.m. on a typical weekday. The total number of vehicles in the base matrix (released in two hours of simulation) was 19,500 pcu. Simulation runs were repeated with 50%, 100% and 200% of the base O-D matrix. Within each run, the rate of vehicle released from each centroid or zone remained uniform throughout the simulation period. Fixed time traffic signals were used and all the graphical and statistical functions were disabled. The CPU
times to complete the simulation runs in batch mode were recorded. Figure 4 shows the results of this experiment. The results show that the speed of AIMSUN simulation is relatively faster than PARAMICS Modeller by as much as 25%. However, when the network was relatively uncongested, both tools ran at approximately the same speed.

OTHER CONSIDERATIONS

Application Programming Interface

By using the API, a user can extract data from, or write data onto the simulation model’s internal memory. He can also define plug-in functions to implement certain control actions. As many ITS functions have their own characteristics, and many of them are new concepts that are still in the experimental phases, the provision of API is one of the main reasons for the use or potential use of PARAMICS in modelling road networks with ITS functions. The PARAMICS Programmer is the API gateway that links the PARAMICS Modeller to the external functions or programmes. Many API functions and plug-ins have been used by researchers in earlier versions of PARAMICS (4-10,19,20). The Version 4.0 of PARAMCIS Programmer has provided approximately 800 API functions (2). In GETRAM/AIMSUN, the provision of API is through the GETRAM Extension. The current version of GETRAM Extension has a relatively limited list of approximately 130 functions (21). This is perhaps due to the fact that GETRAM is newer to the commercial market, and some ITS functions (such as drivers’ response to VMS, ramp metering algorithms) have been provided in AIMSUN. Many functions appear like plug-ins, i.e., higher level actions. These may be simple to use, but in the authors’ opinion, lack the flexibility for users to customize detail logic based on fundamental function calls. It should also be noted that, unlike PRARMICS Programmer, override functions for car-following and lane changing models are not available in the current version of GETRAM Extensions.

Technical Support

Both PARAMICS and AIMSUN have websites (1,11) where information on their software and downloads can be obtained. Discussion groups are also available for users to share ideas and post messages or questions.

Documentation

To majority of the users, program documentation is the most important source of reference. GETRAM/AIMSUN provides comprehensive information on the functions and traffic models in its manual (12). The PARAMICS manuals are divided by its modules, and the manual of each module is further divided into two parts: user guide and reference manual (2). The user guide is designed as a tutorial but theories for traffic models are not included. The reference manual is more straightforward in organization and explains the functionality of each of the menus, buttons, sliders and other controls available to the user. Documentations on the PARAMICS car-following and lane changing models is not included in the reference manual but may be requested from the vendor.

SUMMARY

The comparisons of the different aspects of GETRAM/AIMSUN and PARAMICS have been summarized in Table 1. Each component is given a rating in the scale of 1 to 3. Depending on the different requirements, users can choose the model that suits their functional requirement and objectives.

Because of the similarity of the TEDI GUI with the Windows environment, GETRAM/AIMSUN is for a new user to learn, enter and edit the network features. It is also easier for a user to model the basic ITS functions related to VMS and ramp metering. This may be appealing to users who have no prior experience in using a microscopic traffic simulation tool. The use of piecewise linear sections in the modelling of a curve link may not be visually attractive. But this can perhaps be compensated by the 3D object view. On the other hand, PARAMICS provides a more detailed two-dimensional simulation animation. After careful adjustment of the link and node attributes, PARAMICS Modeller can replicate the actual observed site geometry and traffic movements more realistically. This has the obvious advantage of giving impressive demonstration on ITS modelling. Although PARAMICS Modeller lacks some basic ITS functions, the hundreds more API functions provided by the PRRAMICS Programmer has provided numerous avenues for an advanced user to develop customized plug-in that perform the necessary modelling tasks. Therefore, GETRAM/AIMSUM may be the choice if a user wants to set up a model quickly, and is satisfied with the readily provided modelling functions. However, PARAMICS should be
more attractive to researchers who wish to simulate sophisticated or new ITS functions, or users who are not facing time constraints and are willing to put in efforts in model building and API programming.

Having said so, the authors would like to note that the comparison and conclusion is based on the authors’ experience in using GETRAM/AIMSUN in modelling one network. Due to the page limit, not all the detail items are reported here. Both PARAMICS Modeller and AIMSUN were run with the default parameter values. The ITS simulation software market is very dynamic. Updated versions and additional plug-ins are constantly being released by developers. Had the evaluation been conducted at a later time, or for a project with a different expectation or evaluation criteria, a different choice of simulation tool or conclusion may be possible.

REFERENCES

LIST OF TABLES

TABLE 1  Summary of Evaluation Results.

LIST OF FIGURES

FIGURE 1  Clementi network loaded in Version 4.0 of PARAMCS Modeller.
FIGURE 2  Clementi network loaded in Version 4.1 of AIMSUN.
FIGURE 3  Speed-flow relationships from detectors on an expressway link.
FIGURE 4  Comparison of run times at different levels of traffic demand.
FIGURE 1 Clementi network loaded in Version 4.0 of PARAMICS Modeller.
FIGURE 2 Clementi network loaded in Version 4.1 of AIMSUN.
FIGURE 3  Speed-flow relationships from detectors on an expressway link.
FIGURE 4  Comparison of run times at different levels of traffic demand.
<table>
<thead>
<tr>
<th>Evaluation Criteria</th>
<th>AIMSUN</th>
<th>PARAMICS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>A. Graphical User Interface</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. System Integration</td>
<td></td>
<td>★★★★</td>
</tr>
<tr>
<td>2. Navigation</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td>3. Background Graphics</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td>4. Editing Options</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td><strong>B. Network Modelling Capabilities</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Network Size Limit</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>2. Simulation Mode</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td>3. Nodes and Intersection</td>
<td>★</td>
<td>★★★★</td>
</tr>
<tr>
<td>4. Links</td>
<td>★</td>
<td>★★★★</td>
</tr>
<tr>
<td>5. Public Transport Modelling</td>
<td></td>
<td>★</td>
</tr>
<tr>
<td>6. Network Equipment</td>
<td>★★</td>
<td>★</td>
</tr>
<tr>
<td><strong>C. Traffic Modelling</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Credibility of vehicle movement models</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>2. Traffic Demand Data</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td>3. Traffic Generation</td>
<td>★★</td>
<td>★★</td>
</tr>
<tr>
<td>4. Route Choice Models</td>
<td>★★★★</td>
<td>★★★★</td>
</tr>
<tr>
<td><strong>D. Simulation Output</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Detail of Output</td>
<td>★★</td>
<td>★★★★</td>
</tr>
<tr>
<td>2. Format of Print-out Output</td>
<td>★</td>
<td>★★★★</td>
</tr>
<tr>
<td>3. Colour Representation of Network Statistics</td>
<td>★★★★</td>
<td>★</td>
</tr>
<tr>
<td>4. Dynamic Graphs</td>
<td>★</td>
<td>★★★★</td>
</tr>
<tr>
<td><strong>E. Performance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Run Time</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td>2. Program Integrity</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td><strong>F. Other Considerations</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1. Application Programming Interfaces (API)</td>
<td>★</td>
<td>★★★★</td>
</tr>
<tr>
<td>2. Technical Support</td>
<td>★★★★</td>
<td>★★</td>
</tr>
<tr>
<td>3. Documentation</td>
<td>★★★★</td>
<td>★★</td>
</tr>
</tbody>
</table>

Legend:

- ★★★★ Very Good
- ★★★ Acceptable
- ★ Needs Improvement