CONF-970430--7
ANL/RA/CP--92355

To be presented at the High Performance Computing '97 Conference
sponsored by The Society of Computer Simulation,
April 6 - 10, 1997 Atlanta, Georgia.

Highway Traffic Simulation on Multi-Processor Computers *

by

U.R. Hanebutte, Ezzat Doss and Adrian M. Tentner

ITS Program
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439 USA

*Work supported by the Laboratory Directed Research and Development Program.

DISCLAIMER

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

Highway Traffic Simulation on Multi-Processor Computers *

Ulf R. Hanebutte, Ezzat Doss and Adrian M. Tentner

ITS Program
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439

Keywords: Transportation, Multiprocessors, Discrete simulation, Graphics package.

Abstract

A computer model has been developed to simulate highway traffic for various degrees of automation with a high level of fidelity in regard to driver control and vehicle characteristics. The model simulates vehicle maneuvering in a multi-lane highway traffic system and allows for the use of Intelligent Transportation System (ITS) technologies such as an Automated Intelligent Cruise Control (AICC).

The structure of the computer model facilitates the use of parallel computers for the highway traffic simulation, since domain decomposition techniques can be applied in a straight forward fashion. In this model, the highway system (i.e. a network of road links) is divided into multiple regions; each region is controlled by a separate link manager residing on an individual processor. A graphical user interface augments the computer model by allowing for real-time interactive simulation control and interaction with each individual vehicle and road side infrastructure element on each link. Average speed and traffic volume data is collected at user-specified loop detector locations. Further, as a measure of safety the so-called Time To Collision (TTC) parameter is being recorded.

*Work supported by the Laboratory Directed Research and Development Program

1 Introduction

In 1991, the U.S. Congress through the Intermodal Surface Transportation Efficiency Act (ISTEA) initiated an Intelligent Transportation System (ITS) program to provide improved traffic safety and mobility, efficient transit operation, and to minimize negative environmental impact [1]. The proposed automated vehicle-highway system is expected to help conserve energy resources and to make more efficient use of existing transportation facilities. A common national system architecture for the ITS is currently being developed [2]. Some of the elements of the ITS architecture include in-vehicle navigation systems, Traffic Management Centers (TMC) which provide travel advisory and other information to appropriately instrumented vehicles, probe vehicles, road sensors, real-time adaptive traffic control systems, and communication systems. At Argonne National Laboratory, an advanced simulator has been developed for instrumented smart vehicles with in-vehicle navigation units capable of optimal route planning and Traffic Management Center, Refs. [3] and [4].

The highway traffic simulation discussed below includes realistic modeling of variations to the posted driving speed and lane change behavior. The simulator is designed in such a way, that each vehicle is represented by an autonomous simulation module, while interaction of the vehicles with its surrounding is controlled by an in-vehicle sensor detection system and/or by a link manager process. Prior to performing any maneuvering decision, the model searches for and identifies for each vehicle its surrounding vehicle cluster. Each vehicle module is also linked to an environmental model, that is tracking exhaust emissions

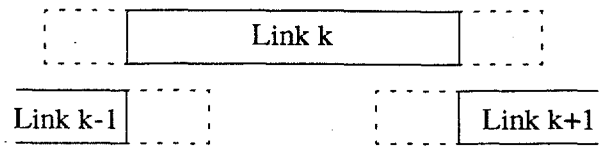

Figure 1: Link-to-Link connectivity.

as well as fuel consumption.

2 Expert Driver - Highway Traffic Simulator

The Highway traffic simulator includes two main components. The first component is a global Expert Driver Model (EDM) which simulates the vehicle traffic through the use of a Link Manager (LM). To facilitate multi-processor computing, the highway system (i.e. a network of road links) is divided into multiple regions, which are controlled by a separate link manager residing on an individual processor. A Link Manager keeps track of information on all vehicles within its domain. It also communicates directly with adjacent link managers and with the vehicles under its jurisdiction. The second component is a local EDM which simulates vehicle-vehicle interaction in a multi-lane highway system. The vehicle maneuvering is based on information received from the Link Manager and/or from simulated on-board instruments and human factor data.

2.1 Global Expert Driver Model (Link Manager)

For ordinary highway traffic, the Link Manager (LM) is artificially introduced to facilitate the simulation. In the case of an Automated Highway System (AHS), the LM is the actual control environment. Many studies utilize a simple link management model, i.e. a single link in isolation, without modeling entrance and exit regions. Although this is sufficient for simple traffic simulations, the LM described in this paper is capable of interacting with surrounding and connecting links, thereby allowing for simulations of complex highway systems.

The basic multi-lane link is used to describe the design of the LM. However, link manager models for other link types of the highway system are being developed based on the same concepts. They include, among others, highway branching, merging, entrance and exit ramps. A schematic diagram of a LM is shown in Figure 1. Details of a basic multi-lane link and the use of overlapping connectivity regions are given in Figure 2. Only vehicles on the actual link are controlled by the LM. However, the LM has further knowledge of vehicles in the connectivity regions of adjacent links. This enables the LM to provide each vehicle on its link with the correct information regarding close-by vehicles from adjacent links. The connectivity regions at the entrance (denoted by E) and exit (denoted by X) of the link are divided into receiving (E_R, X_R) and sending (E_S, X_S) zones. Here, the letters R and S denote receiving and sending, respectively. Furthermore, vehicles leaving the link during a simulation time step are collected into an overflow buffer region (X_O). This overflow buffer is combined with the X_S region before sending information to the adjacent LM. The length of the receiving and sending zones may differ for each link to accommodate the requirements of the connecting links.

Once the LM receives the information about the vehicles under its control and also those in the connecting zones, the local Expert Driver Model EDM is called to perform vehicle maneuvers for the next time step of the simulation. Upon completion of the vehicle maneuvering phase, the LM generates vehicle information for the connectivity regions at the entrance and exit of the link (E_S, X_S) as well as the overflow zone (X_O). This information is transferred to adjacent links. The link manager receives similar information from connecting links. The simulation advances to the next time step where the whole process is then repeated.

2.2 Local Expert Driver Model (Vehicle-Vehicle Interaction)

The local Expert Driver Model (EDM) is used to determine the maneuvering behavior of all vehicles on a single link, taking vehicle-vehicle interaction into account. Such maneuvers are dependent on the road and weather conditions, the type of vehicle and the type of driver.

The vehicles are first sorted in a one- dimensional array (vector) according to the vehicle's position on the link, followed by the creation of the cluster of vehicles around each vehicle on the link. The driver's behavior model is then called to advance each vehicle to the next time step. In order to advance each ve-

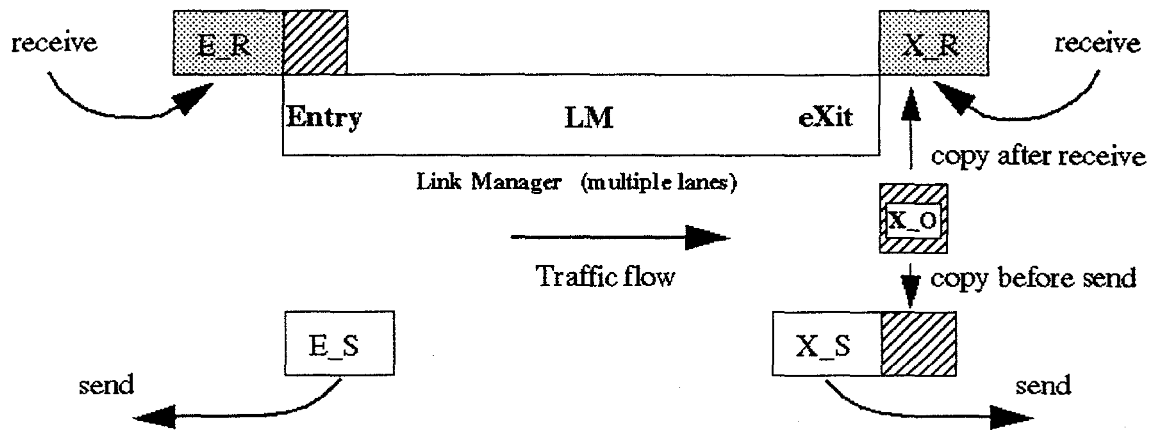

Figure 2: Details of Link Manager (LM) regions: Entry Receive (E_R), Entry Send (E_S), Link Manager (LM), Exit Send (X_S), Exit Receive (X_R) and Exit Overflow (X_O).

hicle to the next simulation time interval, a decision has to be made by the driver's behavior model regarding the vehicle maneuver. The maneuver is based on the geometric structure of the vehicle-cluster and the human factors information concerning the type of vehicle, the type of driver and road/weather conditions. The driver behavior model selects one of the following choices: to change lane (right or left), to slow down, to speed up, or to proceed normally. If the decision is to follow the vehicle in front, a car following model based on a Neural Network Controller[5] is called to provide the appropriate speed and acceleration information.

2.3 Graphic User Interface (GUI)

A graphical user interface has been developed for the Expert Driver Model. A sample screen copy of the color displays is shown in Figure 3. The GUI is written in C and is based on X11/Motif. The default window configuration is shown in Figures 3. The top display window gives a global view of 300 vehicles on a single link (2 miles long). Below this window is a detailed view of the vehicles on a 0.05 mile stretch of the highway. This localized region can be selected by clicking on any region in the global view of the road. The selected region is indicated by two vertical lines in the top window. Loop detector locations are indicated by a square symbol below the distance markers in both road views. Along the top of the screen are two additional windows, one displays the simulation time and the other contains application and display control buttons. One such button, the "Display Vehicle Info" button, allows the user to step through a list of vehicle parameters and to select a parameter which is then displayed for all vehicles in the local road view. As an example, the distance to the preceding vehicle (in feet) is displayed in Figure 3. Any window of the GUI can be resized and individually closed or opened.

It is also possible to retrieve easily vehicle information such as statistical and environmental data for any vehicle currently visible in the local road display. As a measure of safety, the Time-To-Collision (TTC) and the distance to the preceding vehicle are further included in the vehicle information panel. The TTC is a parameter that denotes the time in which a vehicle potentially collides with a slower preceding vehicle and is calculated (reaction time neglected) for each vehicle traveling faster then the preceding vehicle by dividing the distance between both vehicles by their speed differential. The median TTC value (i.e. the 50 percentile) for the total system is a measure for the level of harmonization of the traffic flow.

2.4 Structure of the EDM Simulator

A block diagram for the EDM simulator, which is built upon an object oriented concept, is given in Figure 4. This schematic diagram depicts three link managers, each with a sample of three vehicles and two ITS infrastructure modules, i.e. loop detectors. In the actual EDM simulator, a single link manager will typically be responsible for a few hundreds or

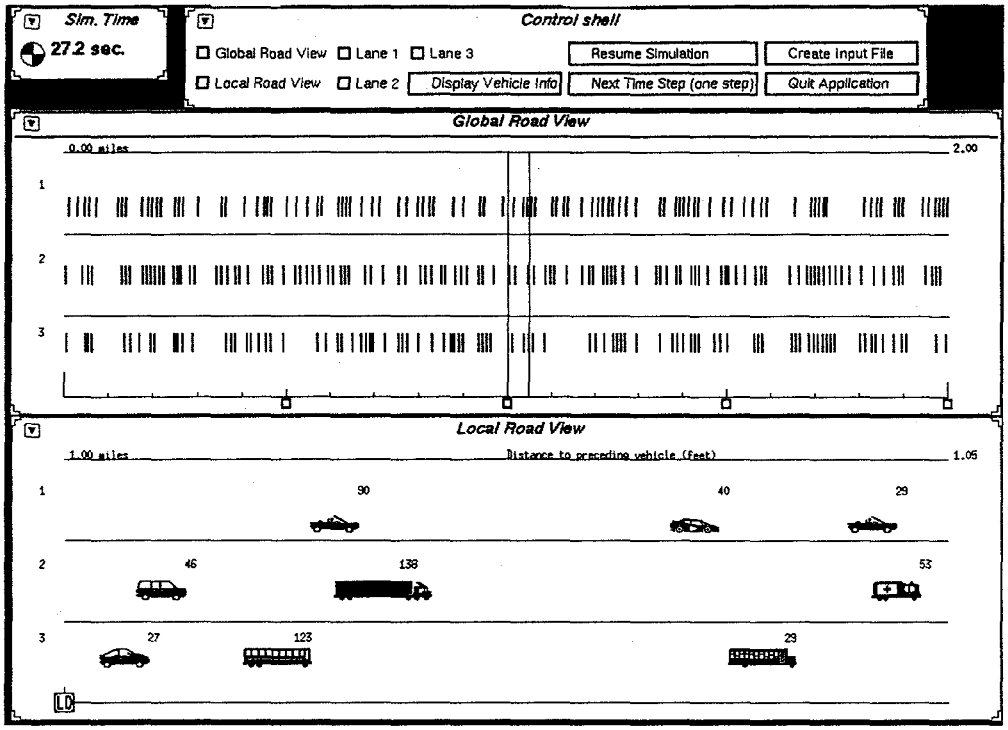

Figure 3: Screen copy of the graphic user interface (GUI) of the EDM highway simulator.

thousands of vehicles. The simulator is further augmented by the Graphic User Interface and data acquisition modules. Not shown in Figure 4 are the details of the link manager and vehicle module. Each link manager module is responsible for *inter link connectivity*, *local vehicle data structure*, *local ITS infrastructure* and *data collection*. The vehicle module consists further of a *current situation*, a *driver's decision*, a *dynamical vehicle system*, an *environmental*, and a *data collection* modules as described before. To support the object-oriented concept of the EDM simulator and its dynamic nature, the code utilizes dynamic data structures such as binary-trees, linked lists and dynamically allocated multi-dimensional arrays. In the parallel implementation, the EDM simulator is further augmented with data exchange, synchronization and control routines.

2.5 Parallel Implementation

The structure of the global EDM facilitates the use of parallel computers for the highway traffic simulation. Domain decomposition techniques as described in Ref.[6] is applied in a straight forward fashion. The highway traffic system is divided into multiple regions; Each region is controlled by a separate link manager residing on an individual processor.

The Message Passing Interface (MPI)[7] is utilized for implementing the simulator as a Single Instruction Multiple Data (SIMD) code for parallel computing environments. MPI, a de facto standard, is supported on most workstations (network connected or multi-processor shared memory) and parallel super-computers. The portable implementation of MPI called MPICH, which has been developed jointly by Argonne National Laboratory and Mississippi State University, is used here (see internet link: http://www.mcs.anl.gov/mpi/mpich/).

Multi-processor workstations represent the state-of-the-art of high-performance computing in engineering offices. Therefore, a two-processor shared memory SUN HyperSPARC workstation (Solaris 2.5 Operating System) is used for the initial implementation, testing, and evaluation of the parallel simulator. In comparison to a network of workstations, a multi-processor shared-memory computer provides at minimum an order of magnitude faster data exchange between processors, which in turn increases the parallel efficiency due to reduced message pass-

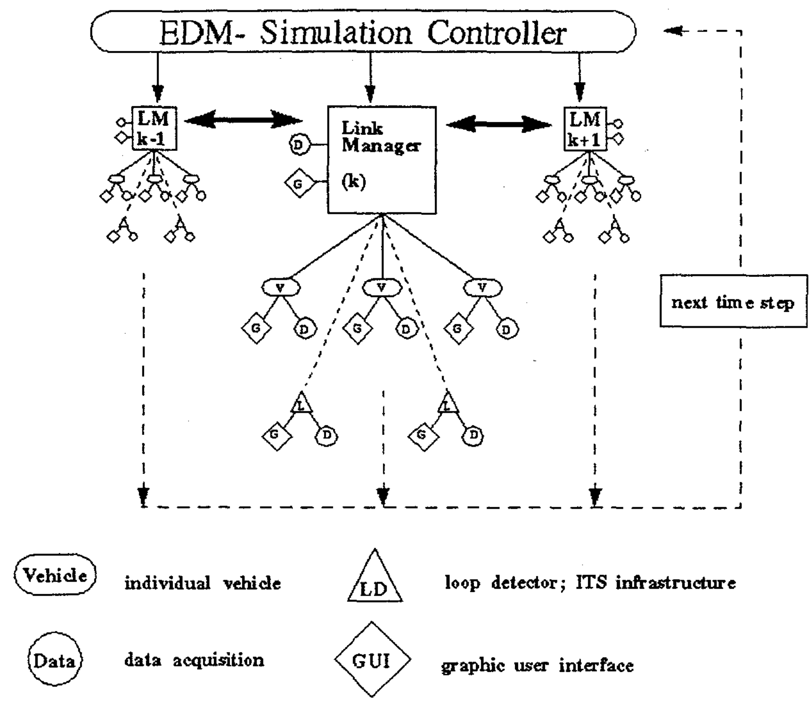

Figure 4: EDM block diagram

ing overhead.

The parallel implementation is based on a version of the simulator for computation in batch mode (i.e. data I/O to files), which does not support the above described graphical user interface. The GUI will be incorporated with the parallel simulator code at a future date. To demonstrate the parallel traffic simulator, a closed loop of 2-mile length is divided into two separate road segments, each controlled by a link manager which is executed on individual processors. The GUI is utilized during the post-processing stage to display the traffic flow, as shown in Figure 5. A situation is depicted in which a stalled truck (located on the second link) is blocking the left most lane. Due to the lane blockage and the occupied middle lane, the sports convertible (located on the first link) had to stop, as indicated in Figure 5. For comparison, a serial calculation is performed utilizing a single link manager. Both, the serial and the two-processor simulations give the exact same results; this demonstrates the correctness of the inter-link boundary treatment. A detailed timing study has not been completed at this time; however, first observations suggest that a linear and in some cases even a super-linear speedup can be achieved on the two-processor SUN HyperSPARC workstation.

3 Conclusions

A computer model has been developed to simulate highway traffic for various degrees of automation with a high level of fidelity in regard to driver control and vehicle characteristics. The model simulates vehicle maneuvering in a multi-lane highway traffic system and allows the use of an Automated Intelligent Cruise Control.

The highway traffic simulation includes realistic modeling of variations to the posted driving speed which are based on human factor studies that take into consideration weather and road conditions, driver's personality and behavior, vehicle type, and safe-passing zones.

The initial implementation of the parallel code on a two-processor shared memory workstation (utilizing the message passing interface, MPI) demonstrates the correctness of the inter-link boundary treatment. First observations suggest that a linear and in some cases even a super-linear speedup can be achieved on the two-processor SUN HyperSPARC workstation.

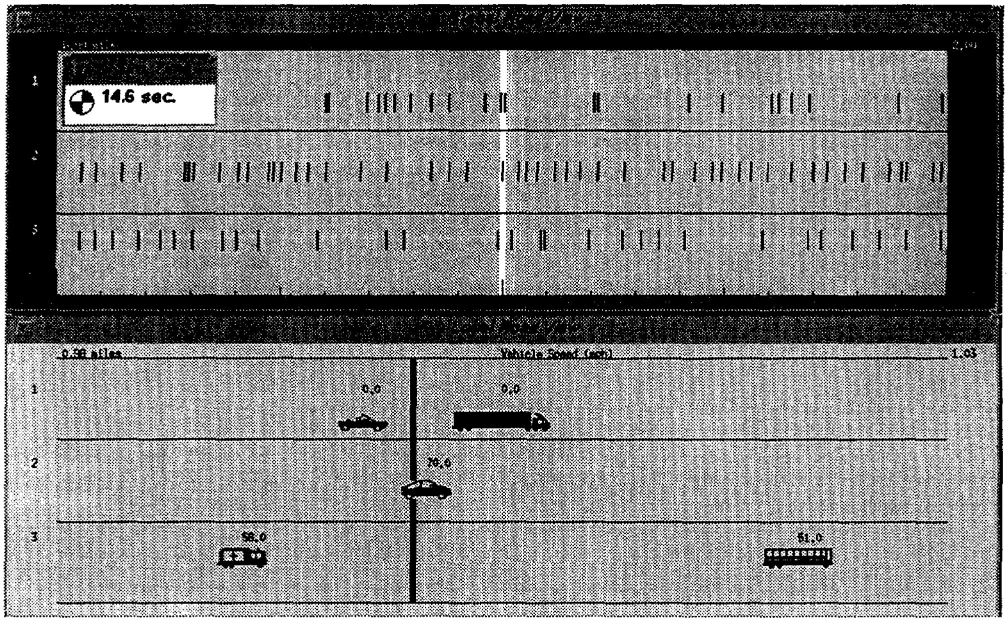

Figure 5: View of the link-link interface for the parallel test case.

Work is progress to implement the parallel code on the 128 processor IBM SP at Argonne National Laboratory in order to perform detailed performance studies on larger processor clusters.

4 Acknowledgment

The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. W-31-109-ENG-38. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

[1] National Automated Highway System Consortium (1995). *Automated Highway System (AHS)-System Objectives and Characteristics.* National Automated Highway System Consortium, Troy, Michigan, November 3, 1995.

[2] Hall, R.W., Lo, H.K. and Minge, E. (1995). *Organization for IVHS: Computer Integrated Transportation.* Transportation Research Board 74th Annual Meeting, Washington, DC January 22-28, 1995.

[3] Ewing, T.F., Canfield, T., Hanebutte, U.R., Levine, D. and Tentner, A.M. (1995). *Large-Scale Intelligent Transportation Systems Simulation.* Proc. High Performance Computing '95, Phoenix, Arizona, April 9-13, 1995, p. 253-257.

[4] Ewing, T.F., Doss, E., Hanebutte, U.R., Canfield, T., Brown-VanHoozer, A. and Tenter, A.M. (1996). *Argonne Simulation Framework for Intelligent Transportation Systems.* Proc. of the ITS America Six Annual Meeting, Houston, Texas, April 1996.

[5] Vitela, J.E., Hanebutte, U.R. and Reifman, J. (1996). *An Artificial Neural Network Controller for Intelligent Transportation Systems Applications.* Proc. Neural Network Applications in Highway and Vehicle Engineering, George Washington University, Ashburn, VA, April 8-10, 1996.

[6] Hanebutte, U.R. and Tenter, A.M. (1995). *Traffic Simulations on Parallel Computers Using Domain Decomposition Techniques.* Proc. Second World Congress on Intelligent Transportation Systems, November 9 - 11, 1995, Yokohama, Japan, p. 1934-1939.

[7] Gropp, W., Lusk, E., Skjellum, A. (1994). Using MPI. MIT Press, Cambridge, MA, 1994.