Development of an Intelligent Traffic Management System for the UCF Driving Simulator

Dr. Harold Klee
University of Central Florida
School of Electrical Engineering and Computing Science
Box 25000
Orlando, FL 32816
(email:klee@mail.ucf.edu)

Abstract. The UCF Driving Simulator has evolved since the late 1990's into a midrange simulator with the potential to conduct research in transportation, human factors and real-time simulation. A few years ago, the most serious limitation related to the absence of vehicular traffic in the visual data base. In 1999, a multi-year project was initiated to introduce autonomous vehicles for real-time interaction with the simulator vehicle and scenario development. This paper describes the progress during the first two years.

A traffic network consisting of several one and two-way roads, an outer loop and intersections with traffic control devices was created. It is completely defined in terms of a set of nodes and links. The nodes are placed along the network at locations which define the permissible traffic patterns. The nodes are connected by links which allow the vehicles to be properly registered on the roads. A database of link properties stores the entire network topography, i.e. node types at either end (source, sink or intermediate), neighboring links, link geometry, as well as transition probabilities for moving from link to link.

Using a Windows PC, positions and headings of simulated vehicles with respect to the link/node coordinate system of the network were calculated using the MATLAB programming environment. Vehicles are uniquely placed on one of the 140 links at a known distance from the upstream node. The link/node vehicle coordinates are transformed in the MATLAB PC to *x,y* and heading coordinates for communication to the simulation control program running in the host computer. Vehicles are introduced as required at source nodes and removed when and if they arrive at sink nodes.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

INTRODUCTION

The UCF Driving Simulator has evolved since the mid 1990's into a midrange simulator with the potential to conduct research in transportation, human factors and real-time simulation. In 1995, a joint UCF research team began a project to enhance the capabilities of the simulator to incorporate modern hardware and software technology. A Silicon Graphics Onyx Reality Engine with two 200 MHz MIPS processors, two raster managers and 192 MB of memory was acquired and installed as the central host computer for the system. It was upgraded in 1999 with two additional 200 MHz MIPS processors and two more raster managers. The system runs a real-time version of the Unix operating system and is sufficiently powerful to control system I/O, integrate a sophisticated vehicle dynamics model, perform data logging operations, and run a 3-channel color video display with 960 x 680 fixed resolution on each channel. Three video projectors were acquired, and a wrap around curved screen providing a 160-degree field of view was installed. The UCF Driving Simulator is shown in Figure 1. Several views from inside the simulator are shown in Figures 2 and 3.

Certain types of roadway improvement projects related to road geometry, sign placement, traffic calming, etc. could be tested in the simulator. Figures 2 and 3 reveal a significant shortcoming of the UCF Driving Simulator, namely the absence of vehicular traffic in the visual data base. Without moving vehicles, the driver is immersed in a sterile driving environment and the simulator has limited utility as a training device or as a testbed for studies involving traffic operations where traffic interaction is essential.

A multi-phase research study was initiated in 1999 to create external traffic movements viewable by the driver in the simulator and also develop a limited scenario generation capability. In Phase 1, there was no attempt to introduce autonomous vehicles, i.e. vehicles whose movements reflect an awareness of each other's presence as well as that of the simulator vehicle. Instead, the primary focus was assuring that all vehicles remain properly registered with respect to the centerlines of each road.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

TRAFFIC GENERATION – INITIAL PHASE

Microscopic traffic simulations including CORSIM (1), WATSim (2), PARAMICS (3), MITSIM (4), SIMTRAFFIC (5) yield detailed and accurate representations of real traffic flow, however, they are computationally intensive and typically not suitable for real-time implementation in a driving simulator. They are normally run in batch mode to produce macroscopic summaries of traffic operations (6,7,8).

Virtually all traffic simulation programs rely on a link/node description of the roadway network. The nodes are placed along the network at locations which define the permissible traffic patterns. The nodes are connected by links which allow the vehicles to be properly registered on the roads. A link/node description of a roadway network is not unique, however regardless of its definition, it is a starting point for the process of artificially generating vehicles for insertion in the network.

A traffic network consisting of several one and two-way roads, an outer loop and intersections with traffic control devices was created and shown in Figure 4 This roadway network is purely fictitious and was created as a prototype for evaluation of the traffic generation routines.

A database of link properties was created to describe the entire network topography, i.e. node types at either end (source, sink or intermediate), neighboring links, transition probabilities for moving from link to link, and relevant geometrical data. A cell array L with 140 rows, one for each link (see Figure 4), stores the complete set of link properties, some of which are entered directly by the user and others which are calculated internally.

L{i, 1}: Link Identifier ('L1', ... , 'L140')
L{i, 2}: Link Type ('ST', 'RT', 'LT', 'UT')
L{i, 3}: Link Start Node ('N1', ... , 'N209')
L{i, 4}: Link Start Node Type ('SRC', 'INT')
L{i, 5}: Link Start Node Road ('R1', ... , 'R8')
L{i, 6}: Link End Node ('N1', ... , 'N209')

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

L{i, 7}: Link End Node Type ('INT', 'SNK')

L{i, 8}: Link End Node Road ('R1', ... , 'R8')

L{i, 9}: Link Upstream Links ('L1', ... , 'L140'), up to 3

L{i, 10}: Link Downstream Links ('L1', ... , 'L140'), up to 3

L{i, 11}: Link Transition Probabilities ([0], [P1, P2], [P1, P2, P3])

L{i, 12}: Link Length (ft), calculated from link node data for 'ST' links or (h, k, r, θ) for 'RT', 'LT', and 'UT' links

L{i, 13}: Link Speed Limit (mph)

L{i, 14}: Link Start Node x,y coordinates

L{i, 15}: Link End Node x,y coordinates

L{i, 16}: Heading (rad) with respect to positive x axis for 'ST' links or (h, k, r, θ) for 'RT', 'LT', and 'UT' links

Running on a Windows PC platform, MATLAB (9) was used to calculate positions and headings of simulated vehicles with respect to the link/node coordinate system of the network. MATLAB was chosen because of its powerful data analysis and visualization, extensive built-in library of mathematical functions, programming environment, and its capability to produce easily customizable GUI's.

Uniquely identified vehicles are initially placed on each of the 140 links according to a default or user's initial density specification. Link/node vehicle coordinates, consisting of the link number, distance along the link from the upstream node, and heading are transformed in the MATLAB PC to *x*,*y* and heading values in the coordinate system of the 3D modeling software used to create the visual database. This transformation guarantees the vehicles will be properly registered with respect to the centerlines of the roads. These coordinates are communicated over a network to the simulation control program running in the host computer, a Silicon Graphics ONYX computer. New link/node vehicle coordinates are established each frame, i.e. a complete pass through the traffic generation loop. The situation is illustrated in Figure 5. The cab I/O routines run in a separate PC which receives driver input signals over ethernet from a collection of analog and digital I/O modules under the hood. The vehicle dynamics model runs on the host computer, however a separate PC networked to the host computer will soon assume this function.

New vehicles entering the network are introduced at source nodes 'N100', 'N101', ... ,'N109' (see Figure 4) in a stochastic fashion based on default or user defined average arrival rates. Current vehicles are removed when and if they arrive at sink nodes 'N200', 'N201', ..., 'N209'. On the PC side, an onscreen map of the network displaying several different views of the simulated traffic movement, frame by frame, is useful for off-line software development purposes. The view selection menu allows the user to track a specific vehicle, view selected links, view the entire network, or track the simulator vehicle. An individual frame snap shot showing traffic at one of the intersections is shown in Figure 6. The network map is not rendered when the simulator is running in order to minimize the traffic generation cycle time on the MATLAB PC.

Currently, the simulator's position from the vehicle dynamics model is not communicated to the traffic generation software on the PC. Consequently, the externally generated traffic does not respond to the presence of the simulator vehicle. Instead, for each frame of traffic generation, previously recorded simulator vehicle profile data (t,x,y,heading) is read from an external file to account for the location of the simulator vehicle. This was done to test various procedures for locating the nearest link to the simulator vehicle, a requirement for future integration of the real simulator vehicle and intelligent traffic generation. Initially there was no intelligence in the simulated traffic as the vehicles were either randomly advanced along the links each frame or deterministically placed for purposes of creating specific scenarios. Several snapshots of the simulated traffic from different views are shown in Figures 7-10.

The traffic generation frame rate is a function of the number of cars and other factors which affect the computational load on the PC. Initial results of timing the traffic generation loop subject to various loads indicate a sustainable frame rate in the neighborhood of 1 Hz on a relatively slow PC with a 333 MHz processor. Interpolation is used to produce vehicle coordinates synchronized with the basic simulator frame rate to assure smooth trajectories of the rendered vehicles. A single frame lag is the penalty associated with using interpolated values based on "old" traffic coordinates to determine the current positions of the simulated vehicles.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

TRAFFIC GENERATION – SECOND PHASE

The goal of Phase 1 was acquiring the ability to populate the simulator database with moving vehicles whose paths were confined to the existing roadway network. In the second phase of traffic generation, the vehicles are being programmed to behave in an intelligent fashion. That is, instead of random movements executed by the external vehicles, simplified car-following algorithms, supplemented with additional logic to control vehicle behavior in the vicinity of intersections, are being implemented to govern traffic behavior.

Car-following algorithms have been used extensively (10,11) to model the behavior of a driver/vehicle in response to the motion of a lead vehicle as shown in Figure 11. The algorithms vary in complexity depending on the degree of realism and the range of conditions for which they apply. For an interactive real-time simulator, simplicity is of paramount importance.

Figure 12 illustrates a simple car-following algorithm based on gap control (12). The driver is modeled as a controller designed to maintain a commanded gap (G) between the driver's vehicle and the preceding vehicle. It is applicable for routine driving conditions where the actual gap (g) is below a nominal value. This eliminates excessive headways where car-following is not even applicable and very low speeds of the following vehicle such as when approaching a stopped vehicle. Driver control functions $K_1(e_1)$ and $K_g(e_g)$, as well as limiting vehicle performance parameters must be determined to produce realistic traffic movements.

Figures 13 and 14 show the results of some rudimentary testing using a trapezoidal speed profile of a lead vehicle followed by two trailing vehicles attempting to maintain a 2 sec gap in the traffic. Other scenarios will be considered as will other car-following algorithms in order to obtain a robust model for the intended application.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

Traffic generation at intersections poses a number of problems. Work is currently in progress to simulate vehicle movements at intersections with and without traffic control devices. Approaching vehicles are assigned one of three actions, namely 'Go', 'Stop' or 'Yield' depending on the type and number of approach lanes, presence of traffic, existence of traffic control devices and the cycle phase for a traffic light. Transition probabilities associated with the downstream node of each link control the movements (left, straight, or right turn) of vehicles at an intersection.

Video clips of the PC traffic generation output and the view of traffic from the simulator are available for downloading at the CATSS (Center for Advanced Transportation Simulation Systems) website (http://www.catss.ucf.edu).

FUTURE RESEARCH

Further refinements are necessary before the simulator is capable of supporting applications involving driver training and traffic operations. The following research activities are still to be addressed.

1. Ambient traffic vehicles must have knowledge of the position and speed of the simulator vehicle to properly position themselves when in the vicinity of the simulator. Software designed to establish the simulator's link coordinates is being tested so that it (the simulator) can be viewed as simply another vehicle in the traffic mix (albeit its position is generated external to the traffic generation model).
2. Vehicles must have the abilities to switch lanes. Traffic flows are more realistic if vehicles are not constrained to a link, e.g. are able to change lanes when passing or attempting to avoid congestion.
3. Ambient driver/vehicle entities should exhibit individualized characteristics so vehicle movements are less predictable. This requires the use of randomized parameters for factors such as driver aggressiveness and vehicle performance.

4. Traffic scenarios involving the ambient vehicles must be easily programmed to test drivers reactions and accident avoidance maneuvers. Authoring tools enabling expeditious creation of traffic scenarios is essential for driver training, certification and transportation research.

5. MATLAB is an interpretive programming environment. Traffic generation execution is relatively slow. After sufficient testing and evaluation, the traffic generation software will be converted to C++ to capitalize on the its object oriented environment and benefit from the reduced run time of the executable code.

Figure 1 The UCF Driving Simulator

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

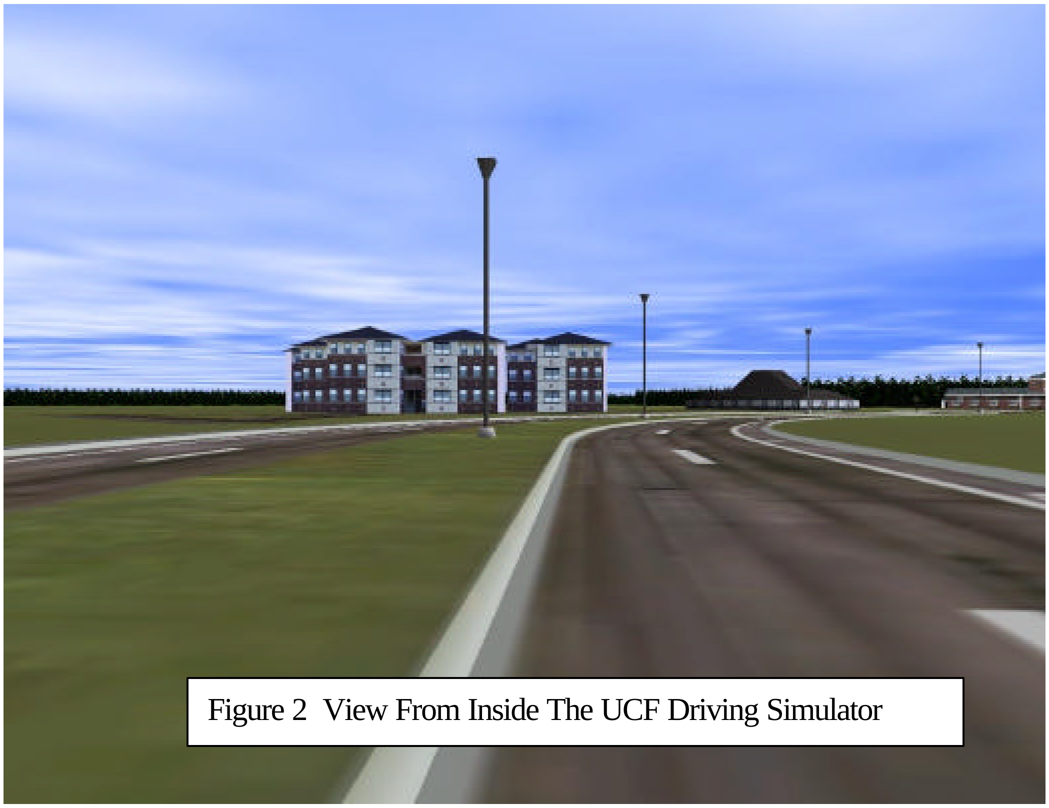

Figure 2 View From Inside The UCF Driving Simulator

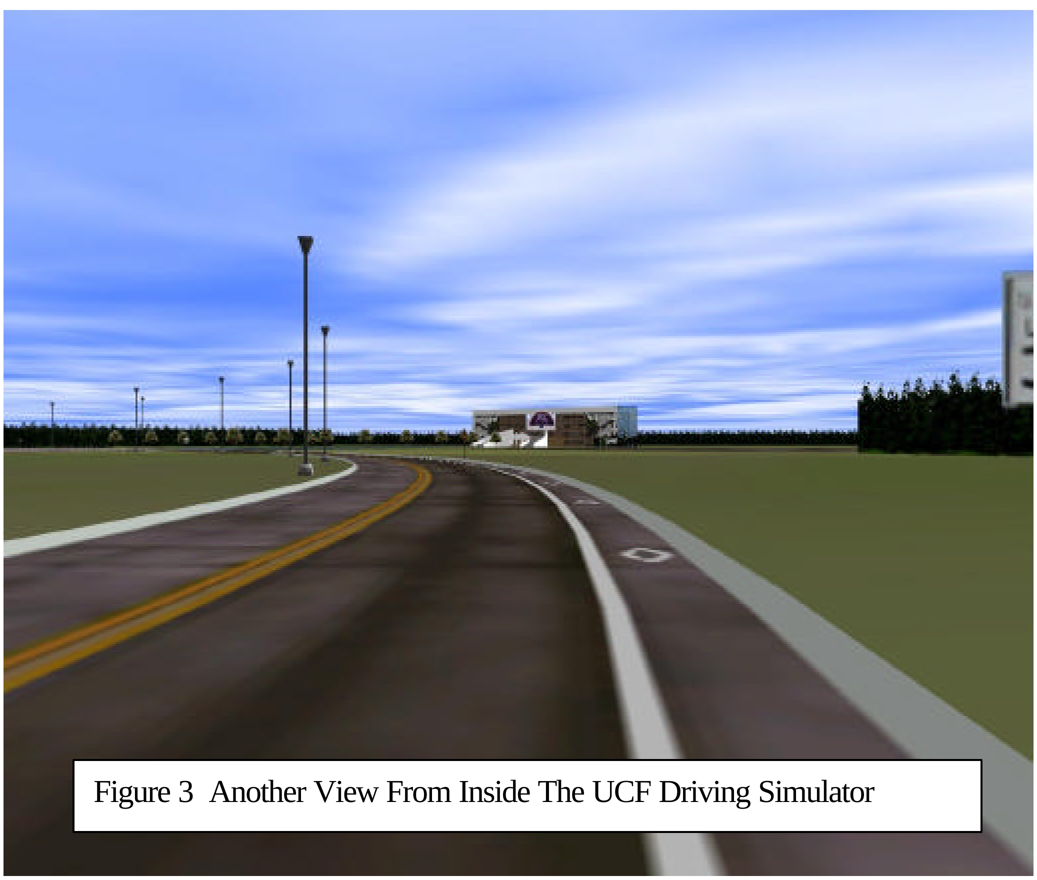

Figure 3 Another View From Inside The UCF Driving Simulator

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

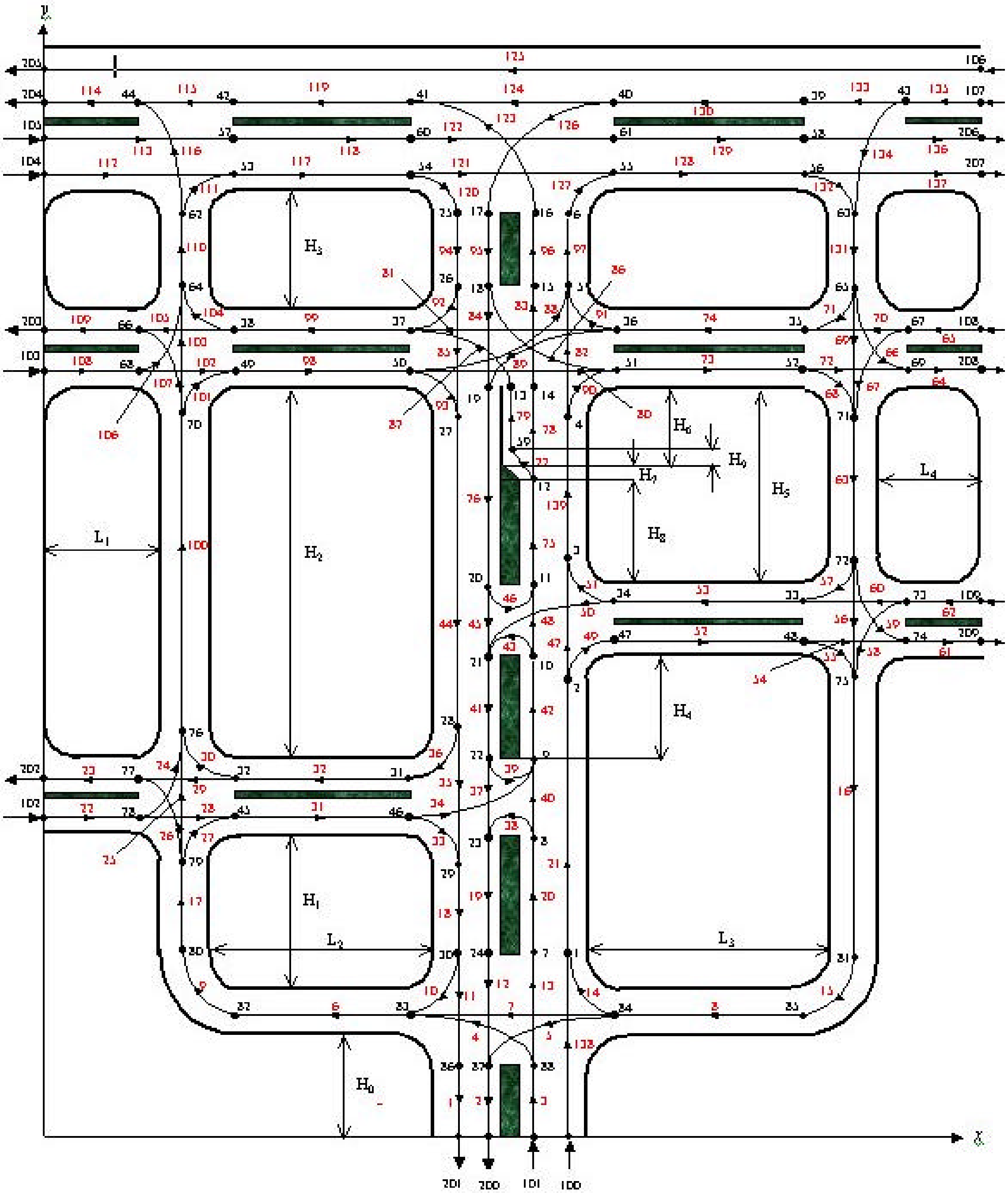

Figure 4 Roadway Network for Traffic Simulation

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

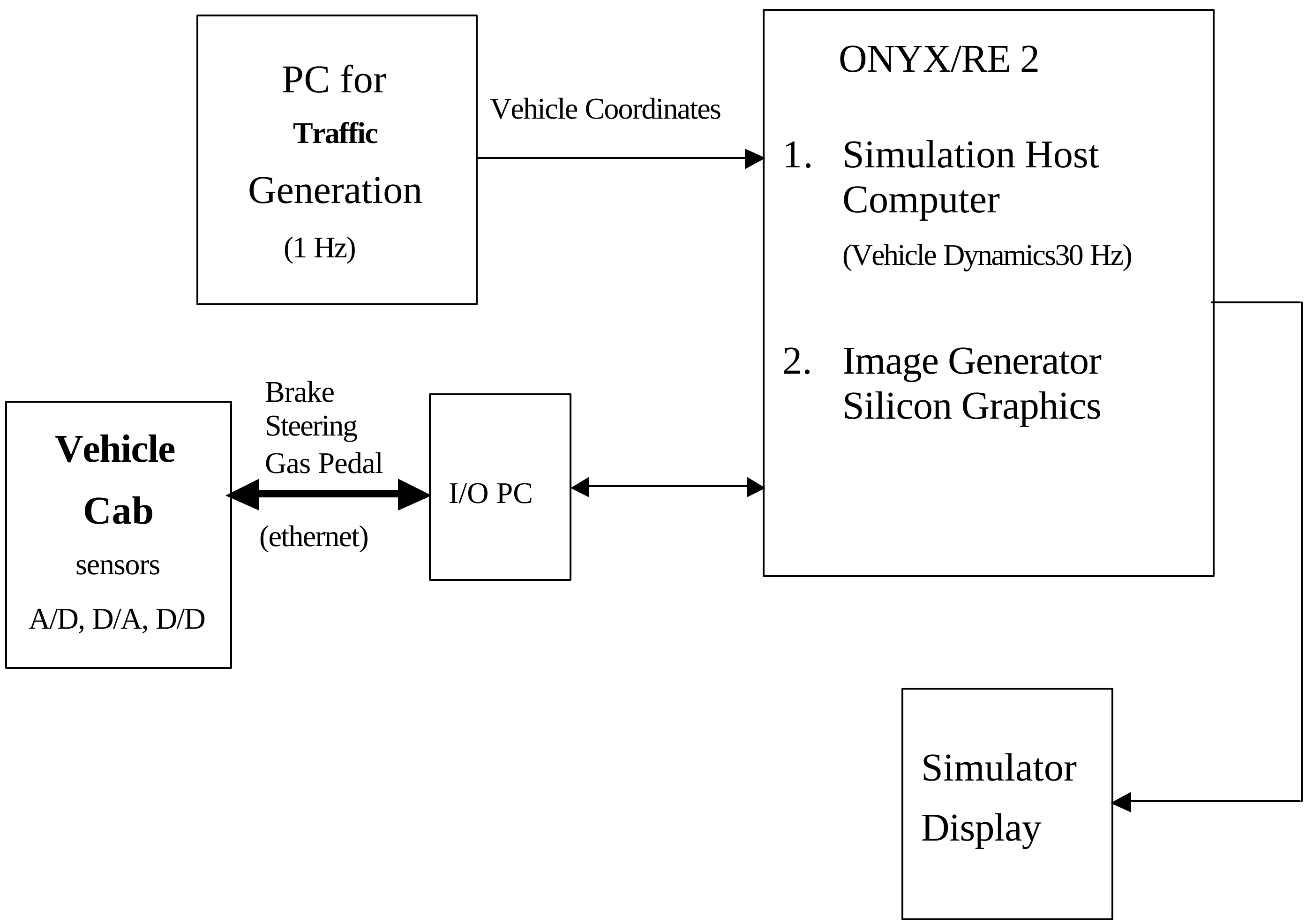

Figure 5 Simulator Hardware Configuration

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

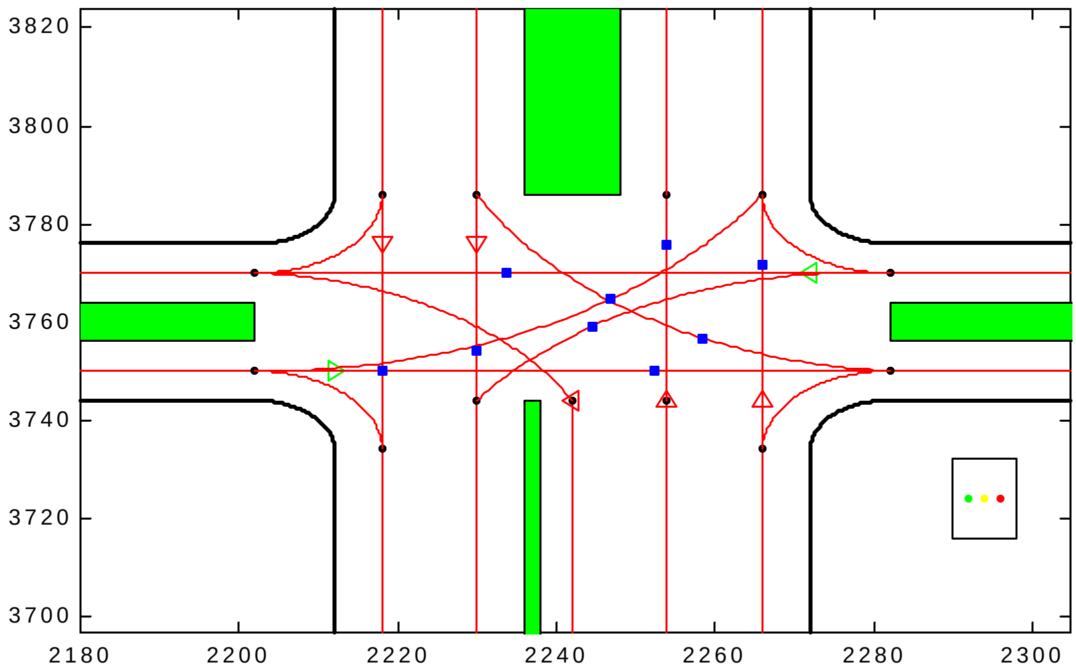

Figure 6 Snapshot of Roadway Network Intersection from PC Traffic Generation

Figure 7 View of External Traffic From Fixed Location

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

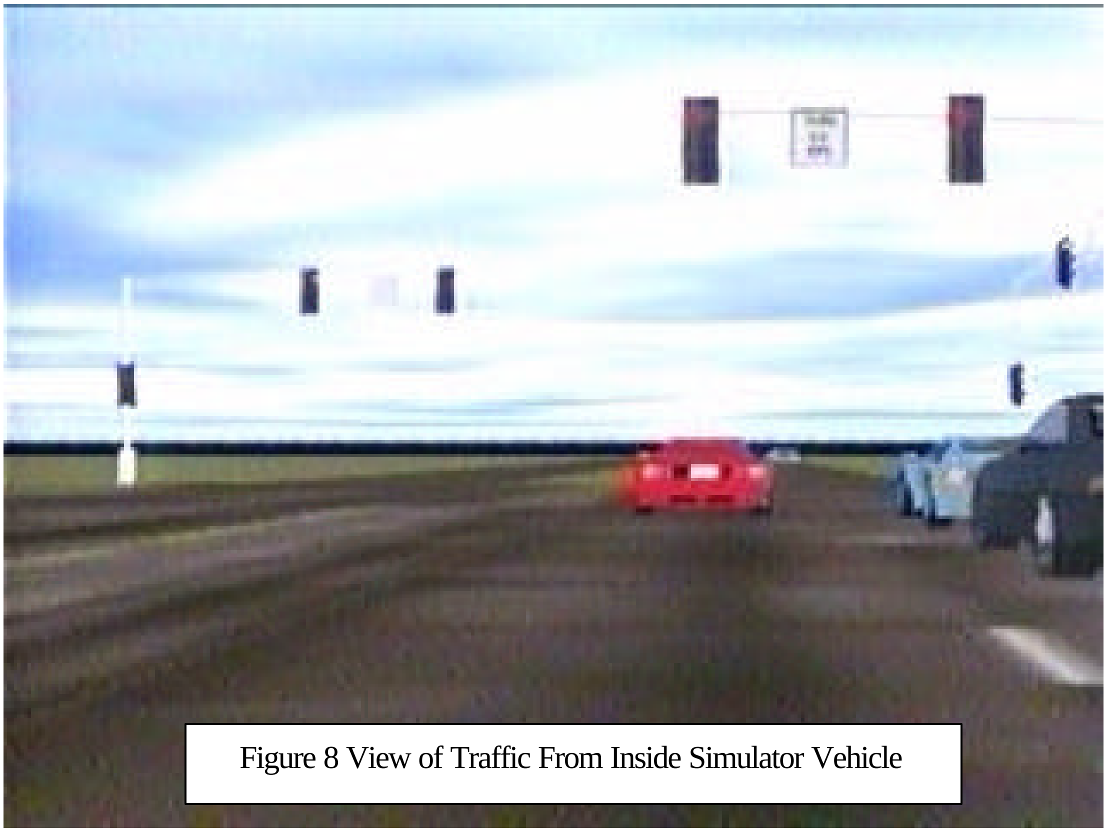

Figure 8 View of Traffic From Inside Simulator Vehicle

Figure 9 Another View of External Traffic From Fixed Location

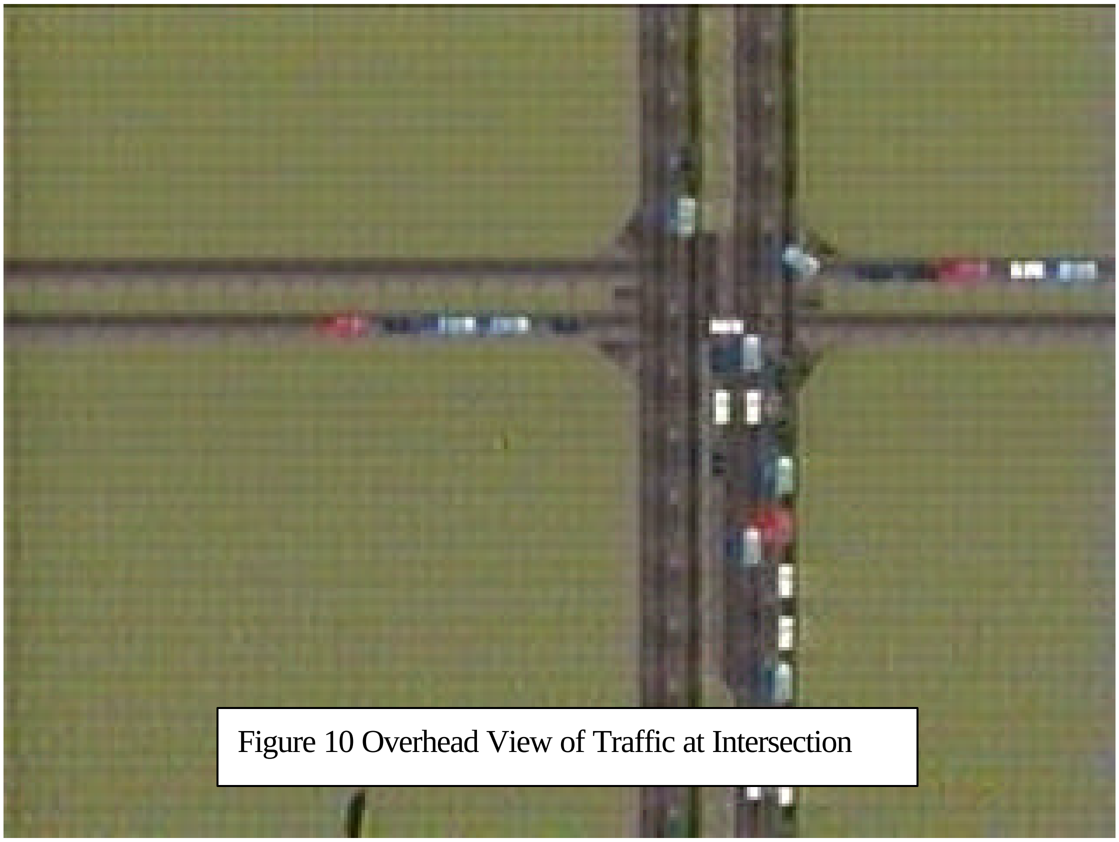

Figure 10 Overhead View of Traffic at Intersection

L	Vehicle length
x_{n-1}	Position of lead vehicle
x_n	Position of following vehicle
$h=(x_{n-1}-L)-x_n$	Headway
$\dot{x}_{n-1}$	Speed of lead vehicle
$\dot{x}_n$	Speed of following vehicle
$g=\dfrac{(x_{n-1}-L)-x_n}{\dot{x}_n}$	Gap

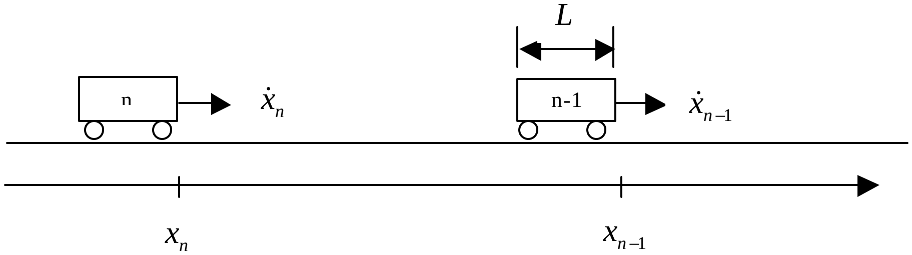

Figure 11 Car-Following Configuration and Notation

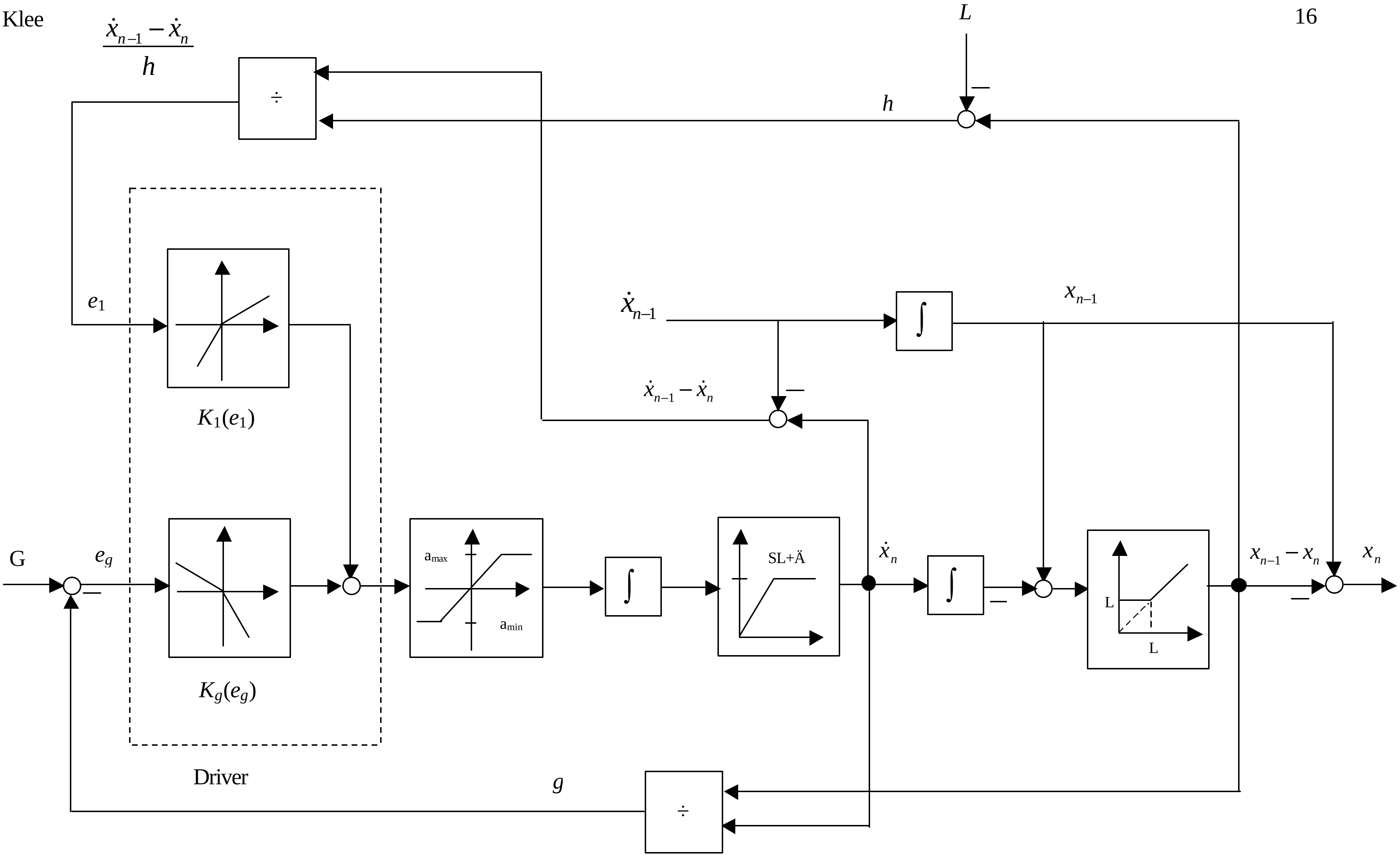

Figure 12 Block Diagram of Proposed Car-Following Algorithm

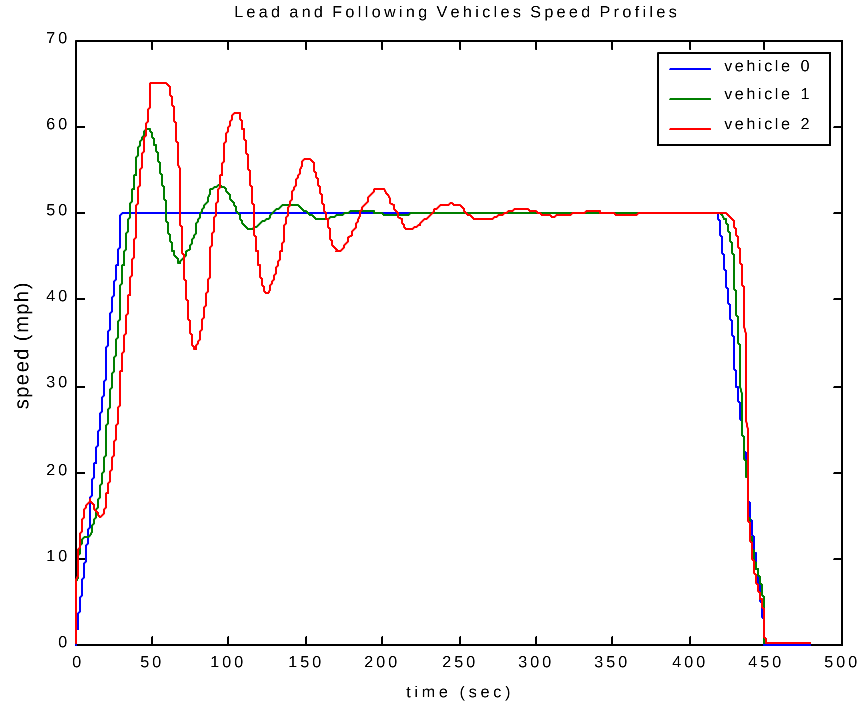

Figure 13 Simulation of Vehicle Speeds

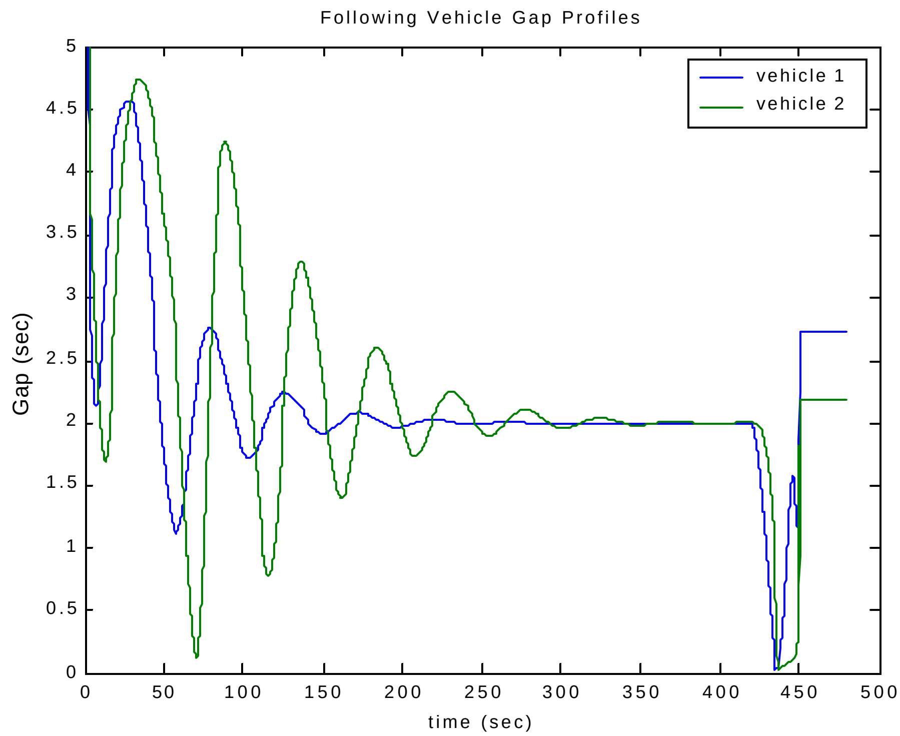

Figure 14 Simulation of Vehicle Gaps

REFERENCES

1. Federal Highway Administration, *CORSIM User's Manual*, Version 1.04, 1998.
2. KLD Associates, Inc., *WATSim Model: User Guide*, 1966.
3. Quadstone Limited, *Paramics V3.0 System Overview*, http://www.paramics-online.com, UK, 2000.
4. Yang, Q., *Microscopic Traffic Simulator*, http://its.mit.edu/products/simlab/, MIT, 1997.
5. TrafficWare, *SimTraffic*, http://www.trafficware.com, 2000.
6. Prevedouros, P. and Y. Wang, "Simulation of a Large Freeway/Arterial Network with CORSIM, INTEGRATION and WATSim", *Transportation Research Record*, 78th Annual Meeting, Washington, D.C., Jan 10-15, 1999.
7. Yang, Q. and H.N. Koutsopoulos, "A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems", *Transportation Research Record*, 1996.
8. Daigle, G., Thomas, M. and M. Vasudevan, "Field application of CORSIM: I-40 Freeway Design Evaluation, Oklahoma City, OK. *Proceedings of the 1998 Winter Simulation Conference*, pp 1161-1166.
9. The MathWorks, Inc., *MATLAB Version 6*, Natick, MA.
10. Gipps, P.G., "A Behavioral Car-following Model for Computer Simulation", *Transport Research Board*, Volume 15-B, pp 105-111, 1981.
11. Simonson, S.O., "Car-following as a Tool in Road Traffic Simulation", IEEE – IEE Vehicle Navigation & Information Systems Conference, Ottawa, pp 150-153, 1993.
12. Zhang, X., "A Computational Model of Autonomous Intelligent Vehicles for Traffic Simulation and Scenario Generation in a Virtual Driving Environment", Ph.D. Candidacy Proposal, University of Central Florida, 2001.

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).

Proceedings of the 1st Human-Centered Transportation Simulation Conference, The University of Iowa, Iowa City, Iowa, November 4-7, 2001 (ISSN 1538-3288).