Proceedings of the IEEE ITSC 2006
2006 IEEE Intelligent Transportation Systems Conference
Toronto, Canada, September 17-20, 2006

WA2.2

Development of a Microscopic Traffic Simulator for Inter-Vehicle Communication Application Research

Keith Yu Kit Leung*, MASc Candidate
Thanh-Son Dao†, PhD Candidate
Christopher M. Clark‡, Assistant Professor
Jan P. Huissoon§, Professor

Lab for Autonomous and Intelligent Robotics
Department of Mechanical Engineering
University of Waterloo
Waterloo, Ontario, Canada, N2L 3G1

Email: *kykleung@lair.uwaterloo.ca, †tsdao@engmail.uwaterloo.ca, ‡cclark@mecheng1.uwaterloo.ca, §jph@uwaterloo.ca

Abstract— **This paper describes the development of a microscopic traffic simulator purposely designed for ITS researchers studying Inter-vehicle communication (IVC) concepts and applications in large traffic networks. The simulator can represent real life vehicles within the simulation by using data from vehicle global positioning system (GPS) receivers, enabling validation of theories with real vehicle data. The software is developed on top of the existing microscopic traffic simulator VISSIM with the added flexibility of modelling and efficiently handling communication between large numbers of vehicles. This along with the software architecture will be discussed in detail.**

I. INTRODUCTION

Inter-vehicle communication (IVC) has been a major component of research and development of intelligent transportation system (ITS) in recent years. Safety, a major incentive for developing IVC technology has been reiterated in numerous occasions and it is a general consensus amongst the ITS community that IVC technology has the potential of improving road safety.

Several groups have previously demonstrated the potential improvement in road safety with the implementation of IVC. In [1], a GPS based slowdown warning system was tested in simulation and in real life with a small number of vehicles driving in one lane. It was shown that drivers have a better chance of avoiding a collision with an abruptly decelerating or stopped downstream vehicle when warnings are broadcasted to approaching vehicles. Furthermore, it was shown that there will be an improvement in safety even when the penetration rate of IVC technology is not widespread. Similar simulation of accident avoidance have been shown in [2], but their simulation was extended to include route changing strategies when traffic congestion is sensed via IVC. Large scale collaborative IVC research projects such as CarTalk 2000 in Europe again shows how vehicle communication can be used for warning, longitudinal control for vehicle platoons and cooperative assistance in various driving maneuvers such as lane merging. The CHAUFFEUR European project aims to use IVC in a similar manner but applies the technology to trucks. In addition to providing driver assistance systems, these European projects also focus on the protocol and details of communication [3]. The goal of the Virtual Reality (VR) Simulator for Collaborative Driving Research presented here is to act as a tool for exploring the effects and benefits of such slow down warning systems, route planning strategies, and other new concepts in a more realistic and larger traffic network setting that includes more vehicles traveling in multiple lanes, multiple roads, highways and multiple directions.

The VR Simulator for Collaborative Driving Research is developed on top of a microscopic traffic simulator, where individual vehicles are modelled as particles controlled by a driver behavior model. The simulator focuses on the use of vehicle to vehicle communication as opposed to road to vehicle communication. The purpose of using GPS data within the driving simulator is to allow vehicles in real life with realistic data from sensory hardware to be represented within the virtual world and interact with other virtual vehicles. This differs from hardware-in-the-loop simulators such as VEHIL [4] where virtual objects from the simulation are brought into the real world to interact with a stationary vehicle within a lab environment through the use of mobile robots and actuators. In the approach presented here, the simulator is more accessible in that it can operate from any computer without high costs associated with major infrastructure.

Currently the simulator is able to read GPS data, which gives the advantage of not having to design an accurate model for generating GPS position readings with realistic bias and error. new IVC concepts are already being tested with this simulator. An example is the use of only GPS data for vehicle localization and lane identification without the use of a geographic information system (GIS) or other sensors (e.g. accelerometers and gyroscopes).

II. The Simulator Development Platform - VISSIM Microscopic Traffic Simulator

The VR Simulator for Collaborative Driving Research is designed on top of the existing commercial microscopic traffic simulator VISSIM. The primary reason of selecting an existing software is to make use of the already established driver model which controls car following and lane changing. VISSIM also provides a means of constructing the infrastructure of any traffic network. The two components listed above are essential to any traffic modelling simulators [5]. Other useful features include the ability to implement vehicle models that govern the dynamic characteristics (such as the engine power curve) for default or user defined vehicle types.

Vissim has an application programming interface (API) but the control over vehicles within the simulator using this method is limited and therefore is not employed. Instead, vehicle parameters are accessed through a dynamic link library (DLL) that is used by Vissim when it refers to an external driver model. This dynamic link library includes three functions to handle reading of vehicle parameters, writing of vehicle parameters, the addition and deletion of vehicles, and initialization of configurations. Adjustable vehicle parameters include desired vehicle velocity, acceleration, and lane position.

In one sense the simulator integrates itself into Vissims external driver model. Essentially the driver model provides a controller that reads the relevant parameters for all vehicles within the simulation and make the appropriate outputs to control vehicle behavior as depicted in figure 1. It is also possible to allow Vissims internal driver model and external driver model to concurrently control different parameters by allowing vehicle parameters read from the output function to directly loop back through the vehicle parameters input function. For instance, the simulator add-on module can indicate a desired velocity but the set velocity will be controlled by Vissims internal driver model which takes into account the presence of other neighboring vehicles to avoid collisions. Another example of internal driver model control is lane changing behavior. By allowing Vissim control over this, overtaking of vehicles does not have to be handled unless it is desired to do so.

III. Software Architecture

The VR Simulator for Collaborative Driving Research is programmed in C++. It works in parallel with the VISSIM simulator engine by creating multiple threads for handling various tasks such as inter-vehicle communication and GPS coordinate reading. Multi-threading allows different parts of the program to run concurrently which is desirable as vehicle parameters and other variables can be updated without waiting for a simulation time step or iteration within the Vissim simulator engine. One benefit of this approach is that vehicles exchanging information will always have the latest vehicle states as Vissim continually updates vehicle parameters. All threads are created when a simulation starts and destroyed when the simulation is stopped.

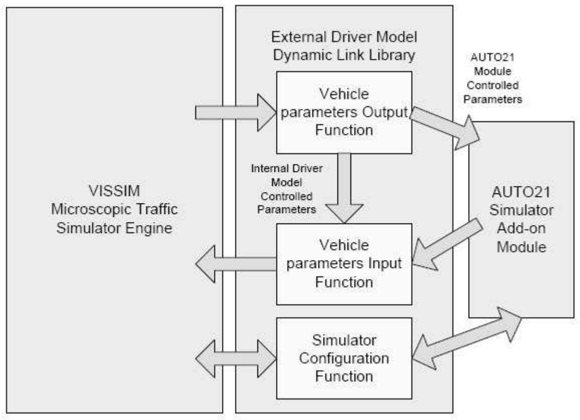
VISSIM Microscopic Traffic Simulator Engine
External Driver Model Dynamic Link Library
Vehicle parameters Output Function
Internal Driver Model Controlled Parameters
Vehicle parameters Input Function
Simulator Configuration Function
AUTO21 Module Controlled Parameters
AUTO21 Simulator Add-on Module

Fig. 1. Accessing Vissim vehicle parameters

Vehicle parameters from the external driver model DLL output function are stored within member variables of a designated vehicle class object. Instances of this class are created at the start of a simulation run and are stored in elements within an object derived from the standard template library (STL) vector class. The vehicle parameter storage class also contains member functions which are used to modify the received vehicle parameters for output to the Vissim simulator engine. Other parameters related to the general simulation environment such as simulation time and the identification of the vehicle currently being updated by Vissim are stored and constantly updated as well. All vehicle parameters are reset at the start of a simulation run. Figure 2 illustrates an overview of the software architecture for the AUTO21 simulator module.

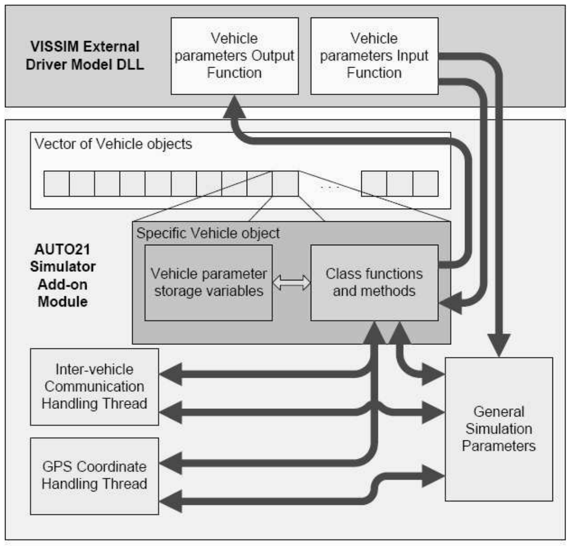
VISSIM External Driver Model DLL
Vehicle parameters Output Function
Vehicle parameters Input Function
Vector of Vehicle objects
Specific Vehicle object
AUTO21 Simulator Add-on Module
Vehicle parameter storage variables
Class functions and methods
Inter-vehicle Communication Handling Thread
GPS Coordinate Handling Thread
General Simulation Parameters

Fig. 2. Software architecture for the AUTO21 simulator module

A. Handling of Inter-vehicle Communication

The thread for handling IVC is the heart of the VR Simulator for Collaborative Driving Research. Its purpose is to manage the exchange of information between vehicles using two levels of operation. On the higher level, the IVC routine pairs up vehicles that are within communication range with each other. The actual exchange of information takes place on the lower level, where messages are passed between vehicles and stored in a buffer prior to processing. The information contained within a message is programmable and has been left open ended at the moment to maintain flexibility for users of the simulator. Likewise, there is currently no specific communication protocol that is modelled within the IVC routine. Specific protocols such as the one proposed in [6] are expected to be implemented in the future that will model aspects of communication such as success rate and information dissemination speed.

Efficiency is important because the simulator is intended for testing effects of IVC on large traffic networks with a large number of vehicles. To limit the number of communication distance checks a vehicle has to make, the simulator module divides the two dimensional cartesian simulation space into square sectors with lengths equal to two times the maximum communication range. Essentially a square grid can be inscribed by a circle that represents a space of communication coverage as shown in figure 3.

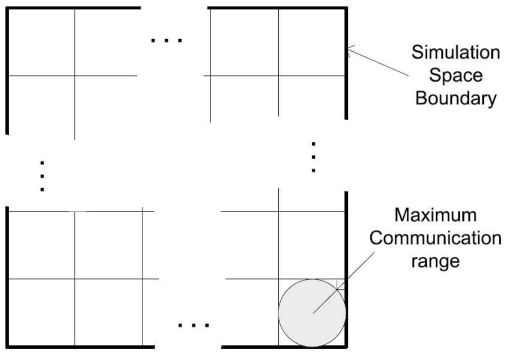

Fig. 3. Sectors for efficient handling of IVC

Each sector is represented by a sector class object in the program. A sector object keeps track of vehicles currently within itself as well as its adjacent sectors and a STL vector is used to store all sector objects. The number of sectors required to fill the simulation space is a function of the rectangular simulation space boundary coordinates, and the maximum communication range. The number of sectors is calculated at the start of a simulation run by taking the product of the number of rows n_r and columns n_c required to construct the grid. These quantities can be determined by knowing the boundary of the simulation space (x_{max}, x_{min}, y_{max}, y_{min}) as well as the communication range r_{comm}.

$$n_c = \left\lceil \frac{x_{max} - x_{min}}{r_{comm}} \right\rceil \tag{1}$$

$$n_r = \left\lceil \frac{y_{max} - y_{min}}{r_{comm}} \right\rceil \tag{2}$$

An index for identification k is also assigned to each sector with reference to its corresponding row r and column c within the grid.

$$k = c + (r-1)(n_c) \tag{3}$$

Each time a vehicle updates its state (x,y), it also determines which sector it currently belongs to and reports its presence to the appropriate sector object.

$$k = \left\lceil \frac{x - x_{min}}{r_{comm}} \right\rceil + \left\lceil \frac{y - y_{min}}{r_{comm}} - 1 \right\rceil (n_c); \tag{4}$$

The IVC routine iterates through all sectors one by one. For a given sector, it will instruct all vehicles within to exchange information with other vehicles in the same sector that is within communication range. For n vehicles within a sector, it is possible to limit the number of information exchange events to $\frac{1}{2}n(n-1)$. After intra-sector communication events have completed, each vehicle will try to communicate with vehicles in adjacent sectors in four specific directions if they exist as shown in figure 4. The number of communication events for any adjacent sector is $n(n_{adj})$. This method of handling communication will ensure that a pair of vehicles will only exchange information once per iteration of the IVC routine to allow for efficient simulation and computation.

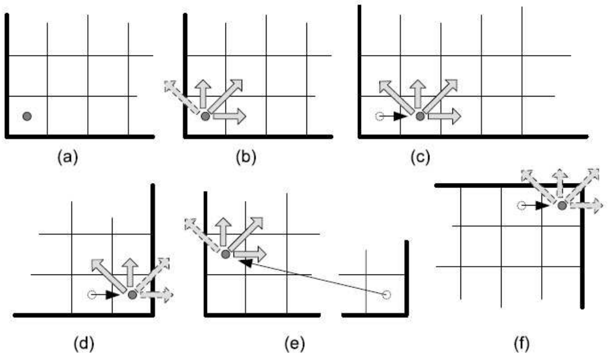

Fig. 4. IVC routine walkthrough: a) Starting from the first sector, vehicles within this sector communicate with each other b) Vehicles within the first sector communicate with adjacent sectors in 4 directions c) Moving across to sector 2, the intra-sector and inter-sector communication repeats d) The end of the first sector row is reached after a number of iterations e) The process continues starting in the second row f) The process continues until the last sector of the last row, then the IVC routine restarts from sector 1

IV. GPS Integration

The purpose of adding the functionality of allowing vehicles within the simulator to follow real GPS coordinate is to allow real vehicles to be represented within the simulator. Currently the simulator does not follow GPS readings in real time. Instead, real data is logged to a file first and the log is read during subsequent simulation runs. A vehicle within the simulator will not be able to distinguish whether the GPS coordinates being used are real time data or not because the logged GPS data is fed into the simulator in a real time manner. That is, a vehicle will not know its future GPS readings even though they already exist in the log file. There are four components that are necessary to achieve GPS coordinate following so that a real vehicle can be represented in the virtual world: GPS Receiver Interfacing, coordinate transformation, traffic network construction, and coordinate prediction and following algorithm.

A. GPS Receiver Interfacing

Two different low cost GPS receivers have been used with the simulator. The first is the LocSense 40-CM, and the second is a Garmin 18-5Hz, both of which can be interfaced with a RS-232 connection. Both GPS receivers output National Marine Electronics Association (NMEA) 0183 standard messages which are sentences with coordinate information embedded within. As such, the first step is to extract the useful information such as time, latitude, and longitude.

B. Coordinate Transformation

The simulator world is a two dimensional cartesian plane, while GPS readings of latitude ϕ,longitude θ, and height h are in a geodetic reference frame which can be looked at as the polar form of the Earth centered Earth fixed (ECEF) coordinate system. Therefore, coordinate transformations are necessary to bring GPS coordinates into the simulator. To proceed, the geodetic coordinates are transformed into ECEF cartesian coordinates using Helmerts Formulation [7]:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} (v+h)\cos\theta\cos\phi \\ (v+h)\sin\theta\cos\phi \\ \big((1-e^2)v+h\big)\sin\phi \end{pmatrix} \tag{5}$$

Where,

$$v = \frac{a}{\sqrt{1-e^2\sin^2\phi}} \tag{6}$$

$$e^2 = 2f - f^2 \tag{7}$$

And, a and f are geometric constants with values of $6378137[m]$ and 298.257223563^{-1} respectively.

The simulation 2D world is assumed to be parallel to a plane that is tangent to a point on the surface of the WGS-84 reference ellipsoid, a current standard for global coordination. Height is therefore not important and does not have any effect. The location of this tangential point (x,y,z) should be within the simulation boundary when it is projected into the simulation space to minimize error in position calculations due to the curvature of the Earth. This point will also serve as the origin for coordinate frames that will result from required transformations. The concepts described above are shown in figure 5.

To move to the right-handed ECEF coordinate frame onto the surface of the reference ellipsoid located by latitude and longitude, the following homogeneous transformation matrix can be used:

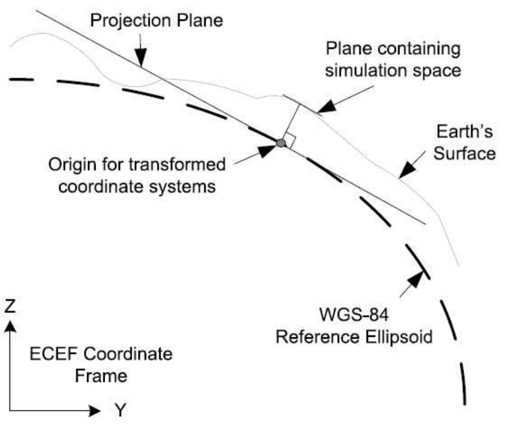
Projection Plane
Plane containing simulation space
Earth's Surface
Origin for transformed coordinate systems
Z
ECEF Coordinate Frame
Y
WGS-84 Reference Ellipsoid

Fig. 5. Coordinate frame transformations and simulation space projection

$$H_{ECEF,1} = \begin{pmatrix} 1 & 0 & 0 & v\cos\theta\cos\phi \\ 0 & 1 & 0 & v\sin\theta\cos\phi \\ 0 & 0 & 1 & v(1-e^2)\sin\phi \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{8}$$

The translation is followed by rotations to align the axes of the coordinate system to the north, east and upward (NEU) directions.

$$H_{1,2} = \begin{pmatrix} cos\theta & -sin\theta & 0 & 0 \\ sin\theta & cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

$$H_{2,NEU} = \begin{pmatrix} cos\phi & 0 & -sin\phi & 0 \\ 0 & 1 & 0 & 0 \\ sin\phi & 0 & cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{10}$$

Combining all the transformations together yields

$$H_{ECEF,NEU} =$$

$$\begin{pmatrix} \cos\theta\cos\phi & -\sin\theta & -\cos\theta\sin\phi & v\cos\theta\cos\phi \\ \sin\theta\cos\phi & cos\theta & -\sin\theta\sin\phi & v\sin\theta\cos\phi \\ \sin\phi & 0 & \cos\phi & v(1-e^2)\sin\phi \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{11}$$

This result is similar to transformations to NED reference frames commonly seen with inertial navigation systems. Depending on the orientation and position of the origin in simulation space with respect to the NEU frame, an additional homogeneous transformation matrix can be used to translate (Δx,Δy) and rotate coordinates about the upward pointing axis by an angle α.

$$H_{NEU,SIM} = \begin{pmatrix} cos\alpha & -sin\alpha & 0 & \Delta x \\ sin\alpha & cos\alpha & 0 & \Delta y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

Therefore, any GPS readings can be mapped into the simulator with the mathematical operation:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_{SIM} = H^{-1}_{ECEF,SIM} \begin{pmatrix} x \\ y \\ z \end{pmatrix}_{ECEF} \quad (13)$$

C. Traffic Network Construction

With GPS coordinates, a Vissim traffic network or infrastructure of the corresponding area where readings were or will be taken is required. Vehicles in the simulation will travel within this defined network. It is important for this traffic network to be an accurate representation of the roads in real life in terms of position and scale because real GPS readings are used. Unfortunately, traffic network generation is a manual process but there are tools available to provide a gauge for accuracy. Satellite images from geographic information system (GIS) resources are used as references in generating the traffic network. Satellite images can be imported into Vissim and used as a tracing template for roads. Furthermore, Vissim provides a scaling function for the image provided the distance between any two points on the image is known.

D. Coordinate Following

As mentioned previously, the purpose for implementing a GPS coordinate following functionality is so that vehicles in real life can be represented in real time. Since most low cost GPS update at a low frequency of between 1 - 5 Hz, there will be time lag between when a vehicle receives the latest coordinate and when the vehicle moves to the appropriate location within the simulator. For a vehicle traveling straight on a highway at 100 km/h, a one second time lag corresponds to a position difference of 27.8 m. To minimize this time lag effect, a prediction is made on the coordinate of the upcoming GPS reading and the vehicle in the simulator is accelerated toward this coordinate such that it will be reached when the next GPS reading is expected to be received.

Prediction of future GPS coordinates $(x_{gps},y_{gps})_{k+1}$ is done using linear extrapolation on previous GPS readings $(x_{gps},y_{gps})_k$ and $(x_{gps},y_{gps})_{k-1}$. This simple prediction method is possible assumung vehicles are non-holonomic and do not change direction or accelerate abruptly while traveling down a highway. Prediction results on a particular road section are shown in figure 6.

Due to error and bias of the GPS receiver, the reported coordinates do not always fall on the road that the corresponding vehicle is travelling. Often, coordinate readings end up in the opposite lane or can be off the road by greater than 10m seen from experiments. Figure 7 is an example of this observation.

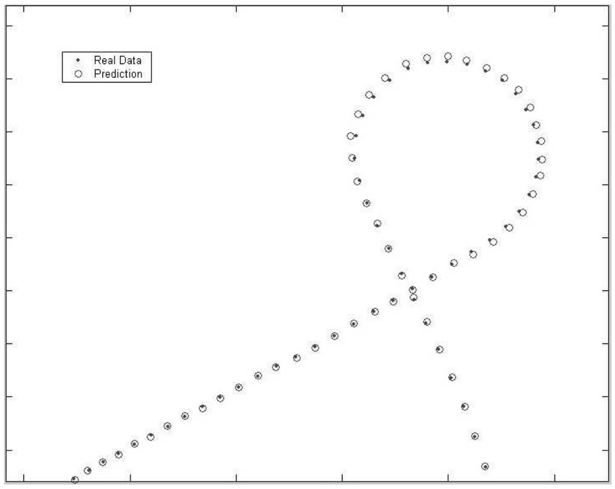

Fig. 6. GPS reading prediction on real data obtained from a highway off-ramp

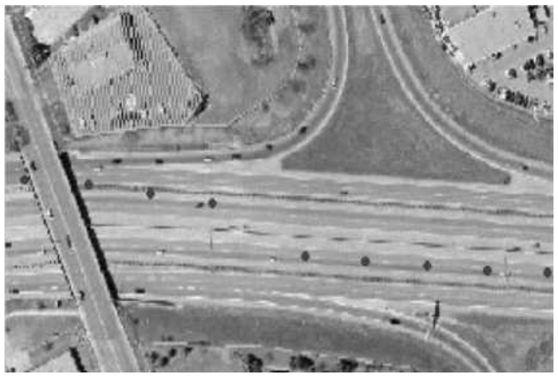

Fig. 7. An example of GPS Error while traveling on a highway (Dark dots represent GPS readings)

A method for correcting this is by projecting the coordinate onto a velocity vector calculated using the current position and the position from the last simulation time step. This has the effect of shifting GPS readings toward the road as illustrated in figure 8

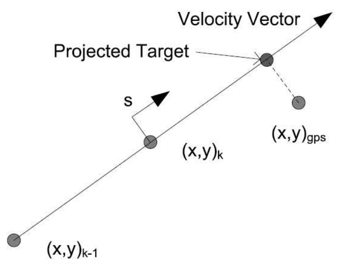

Fig. 8. Correcting for GPS reading prediction that stray from the path of vehicle travel)

Assuming that the vehicle will maintain its heading, the target position can be determined mathematically using geometry and the desired acceleration can be calculated by knowing the time remaining until the upcoming GPS reading and the current velocity. With reference to figure 8, let

$$a = \sqrt{(x_{gps} - x_k)^2 + (y_{gps} - y_k)^2} \tag{14}$$

$$b = \sqrt{(x_k - x_{k-1})^2 + (y_k - y_{k-1})^2} \tag{15}$$

$$c = \sqrt{(x_{gps} - x_{k-1})^2 + (y_{gps} - y_{k-1})^2} \tag{16}$$

The distance to target s is

$$s = a \cos\left[\pi - \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)\right] \tag{17}$$

The required acceleration $\ddot{s}$ given the time remaining until the next GPS reading t and the current velocity $\dot{s}$ is

$$\ddot{s} = \frac{2(s - \dot{s}t)}{t^2} \tag{18}$$

Note that negative acceleration is possible depending on the coordinate of the GPS reading estimation. The complete process of GPS reading estimation, correction, and determination of desired acceleration is repeated for every iteration of the simulator. The routine was tested on a traffic network modeled after a section of highway 85 in Waterloo, Ontario, Canada as shown in figure 9.

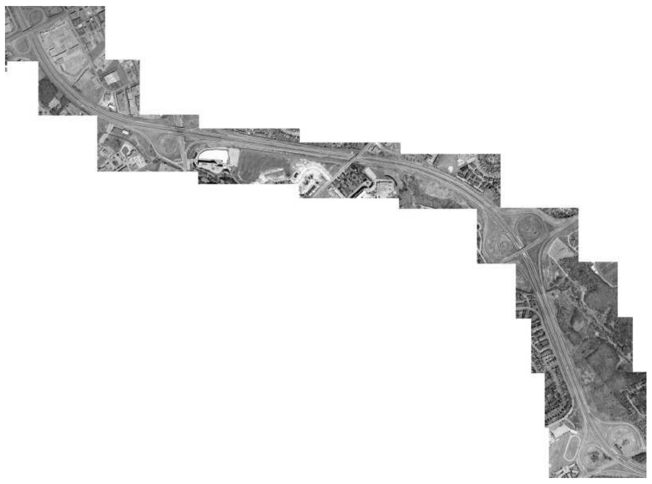

Fig. 9. The Vissim traffic network used for testing the GPS coordinate following routine

The result is that position discrepancy due to time delay is greatly reduced. For a vehicle traveling at highway speeds of about 100 to 120 km/h, the position discrepancy is reduced to 5m or less. This result is sufficient for allowing a vehicle to be represented within the simulation in a real time manner.

V. Conclusions

This paper has presented a new IVC traffic simulator including its purpose, its flexibility, and where it differs from other simulators in terms of scale. The simulator is built on top of an existing simulator VISSIM to make use of previously established components. Details of the software architecture were discussed with special focus on the efficient handling of IVC and the implementation of a GPS coordinate following module for simulation vehicles. The simulator is already being used for testing new IVC and ITS concepts. Hopefully in the future, more research projects in the ITS field will benefit from the software.

VI. Future Work

The VR Simulator for Collaborative Driving Research is now at the stage where it can be used to test novel ideas. Currently, GPS data coupled with IVC is being tested for use in vehicle state estimation and lane prediction which differs from systems that rely on computer vision. In the future, it is desired to not only perform state information on individual vehicles, but on the macroscopic states of local traffic flows. It is believed that state estimation of a group of local vehicles will allow the coordination of maneuvers such as car following, lane changing, and merging. While estimations of macroscopic flow states will lead to optimization of travel route selection and dynamic route planning. Another idea being tested currently is roadway estimation using IVC without relying on GIS. Future research will determine if local estimates together with macroscopic state estimation can be maintained in a traffic system without the use of stationary roadside beacons to aid with IVC.

Focusing on the communication details, it is necessary to determine the properties or parameters that vehicles should exchange with each other to achieve accurate state estimates. The amount of information being exchanged should be minimized however to reduce process time and increase efficiency.

Acknowledgment

The VR Simulator for Collaborative Driving Development is a project funded by AUTO21, a national research incentive funded by the government of Canada. This project is classified under the Intelligent Systems and Sensors research theme and is a part of the Dynamic Collaborative Driving project supported by various academic institutes.

References

[1] A. Chakravarthy, K. Y. Song, and E. Feron, "A gps-based slowdown warning system for automotive safety," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 489–494.

[2] Y. Liu and U. Özgüner", "Quantitative evaluation of inter-vehicle communication effect in highway transportation from micro and macro views," *Internation Journal of Vehicle Information and Communication Systems*, vol. 1, pp. 152–180, 2005.

[3] D. Reichardt, M. Migietta, L. Moretti, P. Morsink, and W. Schulz, "Cartalk 2000 safe and comfortable driving based upon inter-vehicle communication," in *IEEE Intelligent Vehicles Symposium*, 2002, pp. 545–550.

[4] L. Verhoeff, D. J. Verburg, H. A. Lupker, and L. J. J. Kusters, "VEHIL: A full-scale test methodology for intelligent transport systems, vehicles and subsystems," in *IEEE Intelligent Vehicles Symposium*, 2000, pp. 369–375.

[5] Y. Hörmann, H. P. Groβmann, W. H. Khalifa, M. Salah, and O. H. Karam, "Simulator for inter-vehicle communication based on traffic modeling," in *IEEE Intelligent Vehicles Symposium*, 2004, pp. 99–104.

[6] G. Korkmaz, E. Ekici, F. Özgüner", and U. Özgüner", "Urban multi-hop broadcast protocol for inter-vehicle communication systems," in *Association for Computing Machinery VANET 2004 Workshop*, 2004.

[7] *WGS 84 Implementation Manuel*, European Organization for the Saftey of Air Navigation, Institute of Geodesy and Navigation, Feb 1998.