

**Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS**

William G Herbert

March 9, 2009

**1. Introduction and Motivation**

Vehicular route guidance/traffic management [1] and autonomous driving are distinct but related transportation technologies that are likely to be deployed jointly in the future to in order to provide navigation capabilities to driverless vehicles and/or to more accurately simulate transportation networks. While fully autonomous driving systems are still in their infancy, many limited autonomous driving mechanisms, such as lane-keeping, cruise control and crash avoidance systems have been successfully deployed. These systems provide partial control of the vehicle for limited periods, with a human driver supervising the activation, deactivation and operation of the autonomous system. The purpose of this research project was to examine how these two technologies might work together and to assess some of the potential benefits and the challenges involved. This project addressed a specific limited autonomous driving function, lane keeping, the goal of which is to offset or correct driver steering errors and lane departures. In this research, steering error and lane keeping were investigated in relation to TRANSIMS (Transportation Analysis and SIMulation System). This required that steering error/lane-keeping functionality be operational within TRANSIMS. In fact, only steering error had to be added. In its unmodified form, TRANSIMS did not consider steering error, and thus operated as if it already had a working lane keeping function that corrected steering errors.

Thus, for the purposes of this research, a TRANSIMS network modified to incorporate steering error capability can be thought of as a simulation representation of a real-world transportation network in which all vehicles are able to execute steering errors and have no lane keeping functionality. Similarly, a TRANSIMS implementation with no modifications can be thought of as a representation of a transportation network in which all vehicles possess lane keeping functionality that corrects steering errors and maintains a center-lane position for these vehicles. This vision of the TRANSIMS system provided the research framework for this project and is represented in the following table that maps real world transportation methodologies into TRANSIMS components.

**Table 1: Mapping of Real World Transportation Concepts into TRANSIMS Simulation Models**

<b>Real World Representation</b>	<b>TRANSIMS Simulation Model</b>
Vehicles that produce steering errors and do not have lane keeping	TRANSIMS model modified so that vehicles have steering error functionality
Vehicles with lane keeping functionality to inhibit/correct steering errors	TRANSIMS system with no modifications

Within the above research framework, a TRANSIMS system with no steering error capability was compared to a TRANSIMS system with lane keeping (i.e. no modifications) functionality in order to address the following issues:

- Can limited autonomous driving and route guidance/traffic management technologies function together to produce beneficial results?
- What are the required elements of the guidance/traffic management and autonomous driving components that are required for a successful marriage of the two technologies?
- Are there any features of the two components that that may be able to operate independently but not together, and if so, what accommodations can be devised?
- What kind of system controls and interface between the two component technologies would be most useful and easy to use by human operators?

## **2. Steering Errors and Lane Keeping Modeling**

Typically, Lane Keeping systems use vehicle sensors such as cameras with real-time image processing to track features on the road such as lane markings. When these systems detect steering errors or an impending lane departure they either override the driver steering input in order to correct the error (active systems), or else provide an audio, visual or haptic feedback/alarm in time for the driver to take corrective action (passive systems). In passive systems, the driver has the option of accepting or ignoring the lane-keeping input. For this project, it is assumed that lane keeping input is always complied with by vehicles equipped to do so.

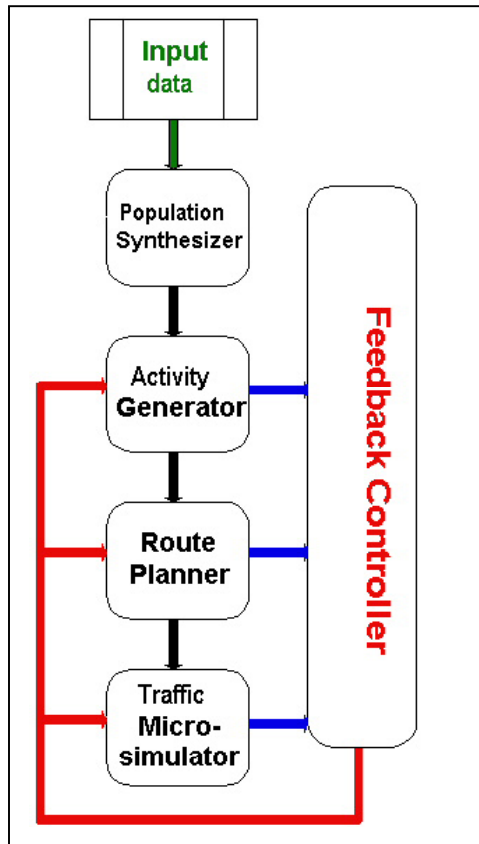
Lane keeping functionality is based on the availability of lane-sensing data. Within the TRANSIMS model, all vehicles know their position and lane boundaries at all times. This is because TRANSIMS vehicles can only be located in the center of a lane. However, in order to for a lane keeping system to function, there must be a steering error to correct. Thus, this project required the incorporation of steering errors into the cellular automata component of TRANSIMS.

Steering errors and lane departures of selected TRANSIMS vehicles (referred to as steering-enabled vehicles) were modeled by causing a vehicle's centroid to depart from center of the grid cell with a designated degree of randomness. The randomness factor represents varying levels of driver distraction. A moderate error would involve a lateral movement within a lane. A major error would cause the vehicle to move into the adjacent lane. If that grid cell is occupied by another vehicle, a collision occurs. Clearly, as network traffic increases, there is increased likelihood that steering errors and lane departures will result in conflict errors and collisions. A probability-based function simulates the possibility that a driver may correct the steering error in time to avoid situations such as collisions, road departure or becoming lost (TRANSIMS terminology describing a vehicle that is unable to move from its current location) in a pocket lane.

### 3. The TRANSIMS Framework

As described in the TRANSIMS background documentation [2], TRANSIMS consists of a series of modules that produce synthetic households, activities for individuals within these households, the choice of routes for movements among these activities, and the microsimulation of these movements to create traffic dynamics on the network. This framework is presented in Figure 1. In this project TRANSIMS modifications involve the microsimulator component only.

**Figure 1: The TRANSIMS Framework**



As referenced in the Installation and Testing How-To document [3], the core TRANSIMS technology includes an array of console-based programs. A console-based program runs in a terminal window (i.e., TRANSIMS does not include a graphical user interface, or GUI). To execute the program, the user typically types the name of the program at the command prompt followed by an application control file. The control file incorporates a number of “user keys,” that allow users to enter simulation parameters without recompiling the program executables. The program streams text messages about its execution status to the screen and to a report file. At the end of the execution the output files specified in the input control file will have been

generated. The console-based approach enables TRANSIMS developers to focus on computational tasks rather than the GUI development.

#### **4. Implementation of Steering Errors within TRANSIMS**

As mentioned previously, TRANSIMS steering errors and lane departures were modeled by causing a vehicle's centroid to depart from center of the grid cell in patterns consistent with publicly available data gathered in previous ITS driving studies. Emory, et.al. [4] found that vehicles on public roads experience 2.8 lane departures or near lane departures per mile, while McQuade, et. al. [5] report that on straight, uninterrupted sections of limited-access highway driving, drivers are able to avoid lane departures in more than 99% of driving. Based on this data, the TRANSIMS steering error generator function was implemented to create 2.8 lane departures per mile at TRANSIMS base level (i.e. default level) of congestion. At its base level (i.e. TRIP\_SCALING\_FACTOR\_1 = 1.0 for Autos), TRANSIMS has a high level of congestion. The Steering Error functionality was investigated at congestion levels ranging from TRIP\_SCALING\_FACTOR\_1 = 1.0 (i.e. heavy congestion) down to TRIP\_SCALING\_FACTOR\_1 = 0.65 (light congestion). As TRANSIMS congestion decreases, the number of steering errors per mile also decreases slightly (down to 2.25 lane departures per mile at congestion 0.65).

TRANSIMS modifications were implemented within the *Microsimulator* Project in accordance with the Software Requirements Statement – version I entitled *Simulating Lane Keeping Functionality in TRANSIMS*, dated 15-August-2008. The implementation involved the creation of the following new programs:

- **Steer\_Error.cpp** – Provides the logic for the steering error functionality. For all network vehicles, this function is invoked within Check\_Ahead.cpp. Within Steer\_Error, a steering error is generated for each vehicle. If the error involves a lane departure, checks are made to determine whether the lane departure took the vehicle into a pocket lane, an adjacent lane or off of the road. If to an adjacent lane, a check is made to determine whether the lane is occupied. If so, and the driver is not able to recover (see random recovery, below), the event is deemed a collision and the vehicle removed from the network. Certain other situations, such as getting lost in a pocket lane, also cause the vehicle to be removed from the network.
- **Random\_Error.cpp** – Generates random steering errors ranging from -1.0 to +1.0 where:
  - -1.0 represents a lane departure to the adjacent lane on the left
  - +1.0 represents a lane departure to the adjacent lane on the right
  - Values between these limits represent within-lane steering errors

# Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS

## Final Implementation Report

---

- **Random Recovery** – Generates random recovery probabilities that drivers can recover from various steering errors such as, erroneously driving into a pocket lane, recovering from a lane departure, recovering from a road departure and avoiding a collision when encroaching into an occupied lane.

In order to support the new functions, the following programs were modified:

- Control.cpp
- Execute.cpp
- Check\_Ahead.cpp
- Vehicle\_Process.cpp
- Microsimulator.hpp
- Microsimulator.cpp
- Msim\_Vehicle.hpp
- Msim\_Vehicle.cpp

The changes outlined above were implemented in TRANSIMS system code version 4.0 downloaded from:

<https://transims.svn.sourceforge.net/svnroot/transims/version4/trunk/src/Transims40/> using TortoiseSVN. The development environment was MS Visual Studio 2005 Express.

The modified Microsimulator code was tested as reported in section 5 below. The final version of the code is uploaded to the TRANSIMS repository:

<https://transims.svn.sourceforge.net/svnroot/transims/version4/branches/billherbert/>. The TRANSIMS code with steering error functionality is open source code that is available for use by all interested researchers. A Microsimulator.exe can be built from the code at ../branches/billherbert and run with other TRANSIMS modules as directed in *Installation and Testing How-To.doc* found at: [https://transims.svn.sourceforge.net/svnroot/transims/version4/trunk/doc/how\\_to/](https://transims.svn.sourceforge.net/svnroot/transims/version4/trunk/doc/how_to/).

The Microsimulator code was modified to include an additional user Key, STEER\_PERCENT that allows users to specify a percentage 0 – 100% of the network vehicles that have steering error enabled.

## 5. Analysis of Steering-Error/Lane Keeping Test Results

Several case studies have been developed and used for research/testing of TRANSIMS performance. Many of these case studies have been developed using census demographic data from U.S. cities such as Alexandria, VA and Portland, OR. While case studies such as these provide insights into real-world problems, the data files are extremely large because they model the entire traffic network of large cities and simulate traffic movement in one second time steps over a 24 hour day. Thus, these models require very long simulation run times.

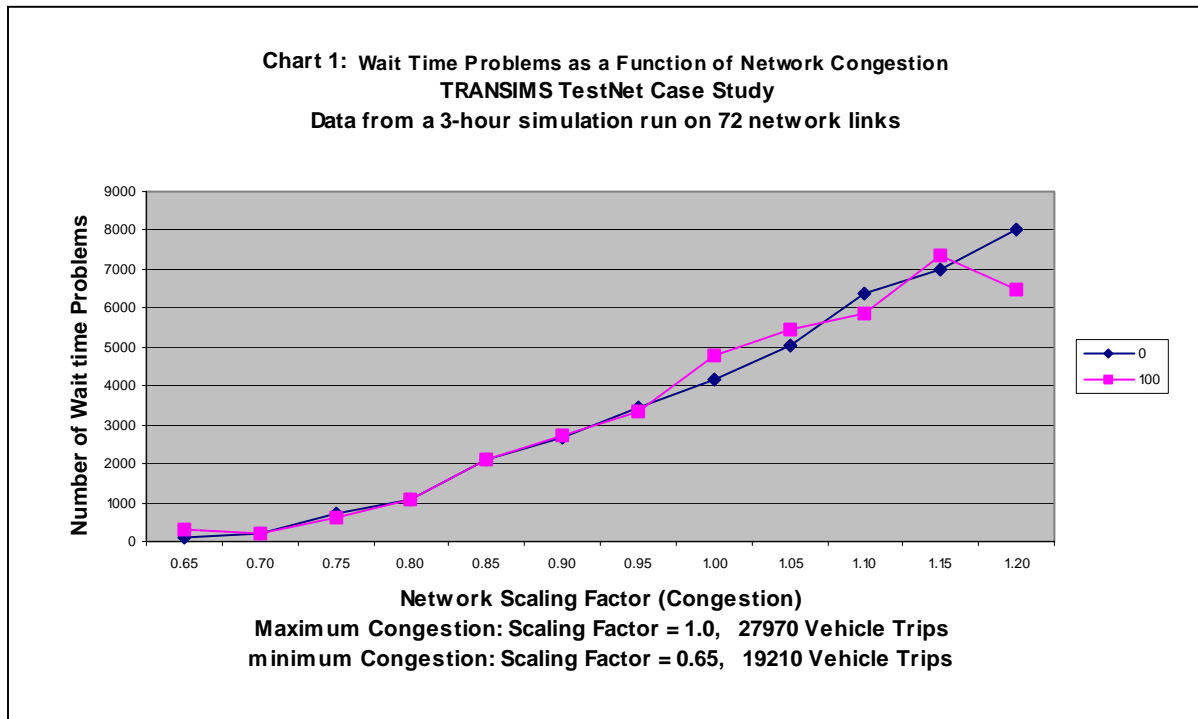
# Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS

## Final Implementation Report

This project used the TestNet case study, a small case study utilizing only 72 traffic links in which traffic was simulated in one-second time steps for only three hours. TestNet is large enough to provide a realistic load for TRANSIMS simulation, but small enough for an entire run to be completed in less than 15 minutes on a small desktop computer. This makes TestNet a very useful tool for traffic researchers who make frequent runs during iterative development and testing of new functionality.

One of the first tasks in the analysis phase of the project was to determine upper and lower levels of congestion for satisfactory performance of TRANSIMS. As mentioned above, the steering error functionality has been calibrated to generate 2.8 lane departure errors per mile. Some of these lane departures will result in a collision if the target lane is occupied by another vehicle. For any given setting of lane departures/mile, the number of collisions/mile should be expected to increase as congestion increases. However, this will not occur if, beyond some limits, the network fails to perform or experiences gridlock or some other problem. Thus, it is important to know the network's useful range in terms of congestion, to ensure that accurate system performance data is generated.

It was also important to determine ensure that the implementation of steering-error functionality did not degrade existing TRANSIMS performance or have an impact system performance in ways unrelated to steering behavior.



# Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS

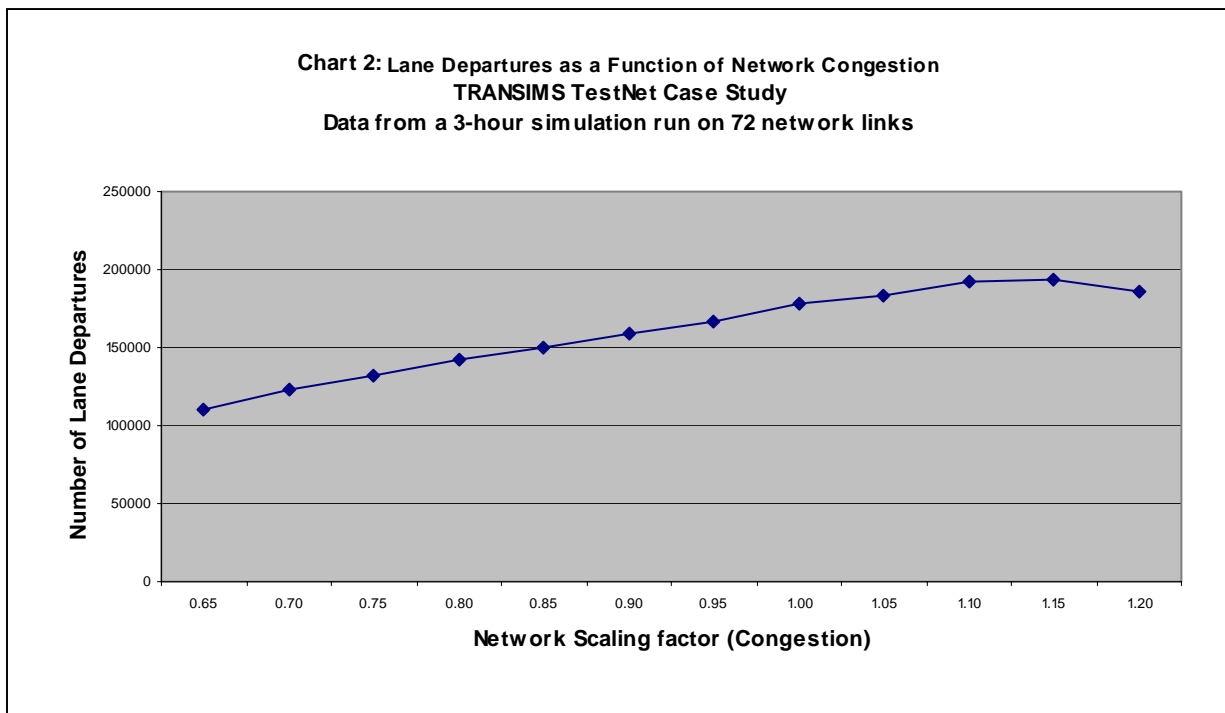
## Final Implementation Report

---

Chart 1 correlates “wait-time” problems with network congestion and compares TRANSIMS results for 0% and 100% of steering-enabled vehicles. A wait time problem is any situation in which a vehicle does not move its position for more than 120 seconds. Wait time is one of many problems that can occur during a TRANSIMS simulation run. It is a user-controlled parameter that has a default setting of 120. Chart 1 shows that wait time problems are comparable for steering-enabled and steering-not-enabled. This suggests that the implementation of steering error functionality has not adversely impacted system performance.

Chart 1 also shows the TRANSIMS system usable range in terms of congestion. For trip scaling factors < 0.65, TRANSIMS does not run. Above scaling factors of 1.0, the number of wait time problems showed great inconsistency between TRANSIMS simulation runs in which steering error was enabled and those in which it was not enabled. This situation suggests that the network congestion was near gridlock, causing the unstable results.

Chart 2 shows the number of lane departures as a function of network congestion. As would be expected, this metric increases consistently between the lower and upper bounds of network congestion. Although the number of lane departures/mile is intended to remain constant at 2.8, we expect the number of lane departures to increase with increased congestion because there are more vehicle trips as congestion increases. Confirming this, we find that, at a scaling factor of .65 (minimum congestion) there are 19210 vehicle trips while at scaling factor = 1.0 (maximum congestion) there are 27970 vehicle trips.



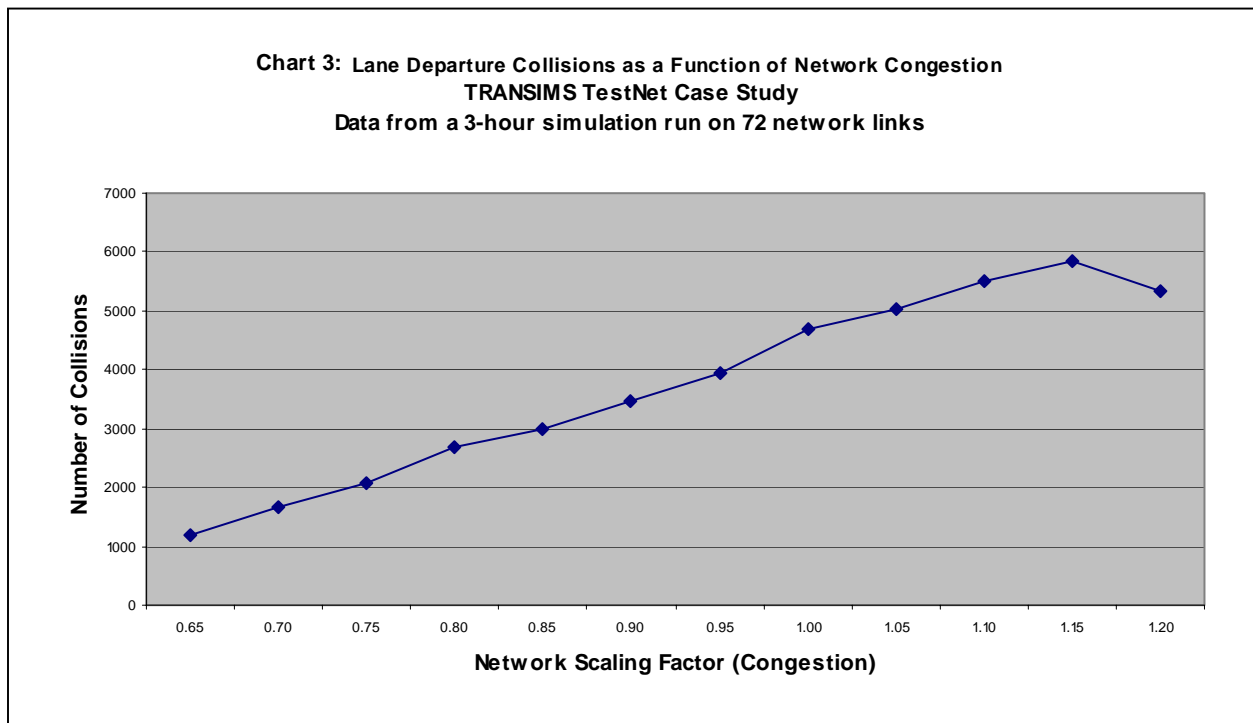
# Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS

## Final Implementation Report

---

Steer\_Error was coded to provide 2.8 lane departures/mile. The system was calibrated at a scaling factor of 1.0 and, in fact, produces 2.8 lane departures/mile at that scaling factor. Although lane departures/mile is intended to be a constant, the current implementation somehow causes this value to decrease slightly as congestion decreases. At a scaling factor of .65, the number of lane departures/mile decreases by 19% to 2.25. This problem, the cause of which has not yet been determined, but will require further investigation.

One of the practical uses of the TRANSIMS Steering Error/Lane Keeping functionality is to predict the number of traffic accidents involving lane departures in which a vehicle (the subject vehicle) not equipped with lane keeping collides with another vehicle that is occupying the target lane into which the subject vehicle has strayed. Chart 3 presents the number of collisions resulting from steering error lane departures that can be expected at various levels of network congestion for vehicles that have steering error capability but no lane keeping.



The number of collisions is based on the following parameters:

- 2.8 lane departures/mile
- 100% of network vehicles are enabled with steering error and 0% are enabled with lane keeping (a situation similar to the current state in real-world driving). By means of a STEER\_PERCENTAGE user-key parameter, users are able to control the percentage of vehicles that are steering error enabled vs. lane keeping enabled. As the percentage of



lane keeping enabled vehicles increases, the number of vehicles with steering error only capability decreases. If 100% of vehicles are modeled as lane keeping enabled, no steering errors, lane departures, lane departure collisions or road departures can occur.

- If a subject vehicle moves into a target lane occupied by another vehicle, there is a 25% likelihood that the driver can correct the error and return to the original lane without a collision, and a 75% likelihood that a collision will occur.

## 6. Summary and Conclusions

The new TRANSIMS steering error functionality implemented in this project provides a framework for transportation researchers to evaluate new steering error driving data by modeling that data within TRANSIMS. It also provides a base model for comparisons to new steering error/lane keeping algorithms and concepts. Based on the new TRANSIMS functionality, and the subsequent analysis, the research questions posed in section one of this report are addressed as follows:

- *Can limited autonomous driving and route guidance/traffic management technologies function together to produce beneficial results?*

Within the simplified vision of the TRANSIMS environment as presented in Table 1 above, the various technologies worked well together with no conflict and no degradation of performance. Manual steering and steering error modeling involved using many of the same algorithms that are used in TRANSIMS for dynamic traffic management. For example the Check\_Ahead and Check\_Behind functions within TRANSIMS are used in simulating lane changes to determine whether a vehicle can safely be moved (by the traffic management system) to a target cell. This same function is now used by the steering error function to determine whether a driver can manually steer a vehicle into a target lane and to compute the probability that a collision can be avoided if the target cell is already occupied.

- *What are the required elements of the guidance/traffic management and autonomous driving components that are required for a successful marriage of the two technologies?*

As stated above, this research demonstrates that traffic management can be modeled with steering enabled vehicles in a manner that shares underlying technologies. However, as noted in [1], Traffic management and route guidance are related but distinct technologies. Route guidance introduces many new complexities since it involves real-world drivers, vehicles and traffic networks. A few of these systems are active (such as the autonomous driving vehicles that compete in the annual DARPA challenge) but most guidance systems in use today are passive, in that they provide guidance to the driver, who may or may not follow the guidance. As mentioned above, lane keeping systems have not been widely deployed for public use, but can be implement as active or passive systems.

# Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS

## Final Implementation Report

---

Passive systems introduce a human element that involves new problems, particularly if more than one system is involved (e.g. both route guidance and lane keeping). For example, suppose that a vehicle is enabled with a passive route guidance system and an active lane keeping system. If we assume that in most cases, route guidance takes precedence over lane keeping, a driver will receive route guidance, decide whether to use it, and if so, implement it possibly creating a need to overrule/disengage the lane keeping system and then re-engage it again later. Thus, it can be reasonably observed that marrying guidance/traffic management and lane keeping is simpler in a simulated environment than in a real-world environment, especially when human factors and multiple systems are involved.

- *Are there any features of the two components that that may be able to operate independently but not together, and if so, what accommodations can be devised? For example, is there any aspect of lane keeping that would prevent the successful execution of a turn or lane change, while the lane keeping function is engaged?*

In this research project, there did not appear to be any aspects of the component technologies that prevented them from working together in a simulation environment.

- *What kind of system controls and interface between the two component technologies would be most useful and easy to use by human operators?*

In the simulation environment, the user interface described in section three is adequate. However, TRANSIMS is a complicated system and the development of a GUI would make the system easier to learn and use. In a real world implementation, to address human overload, it will likely be beneficial to incorporate a console/head-up display that centralizes control for both applications

In order to use this module or any other TRANSIMS modules, The TRANSIMS open source community maintains an active web-based project site with the goal of which has the goal of establishing TRANSIMS as an ongoing public resource available to the transportation community.

## 7. References:

[1] Route Guidance: State of the Art vs. State of the Practice, W. Herbert, F. Mili, IEEE Intelligent Vehicle Symposium 2008, 1/10/2008

[2]

<https://transims.svn.sourceforge.net/svnroot/transims/version4/trunk/doc/Background/transim%20Intro.pdf>

[3]

[https://transims.svn.sourceforge.net/svnroot/transims/version4/trunk/doc/how\\_to/Installation%20and%20Testing%20How-To.doc](https://transims.svn.sourceforge.net/svnroot/transims/version4/trunk/doc/how_to/Installation%20and%20Testing%20How-To.doc)

**Simulating Limited Autonomous Driving (Lane Keeping) in TRANSIMS**  
Final Implementation Report

---

[4] Status Report on USDOT Project “An Intelligent Vehicle Initiative Road Departure Crash Warning Field Operational Test,” L. Emory, et.al., 07/2005

[5] Method For Naturalistic Measurement of Lane-Keeping Behavior, G. McQuade, et. al., University of Michigan Transportation Research Institute, 10/1998