

# FreeSim – A *Free* Real-Time Freeway Traffic Simulator

Jeffrey Miller  
Department of Computer Science  
University of Southern California  
Jeffrey.Miller@usc.edu

Ellis Horowitz  
Department of Computer Science  
University of Southern California  
horowitz@usc.edu

***Abstract* – In this paper we describe FreeSim, which is a fully-customizable macroscopic and microscopic free-flow traffic simulator. FreeSim allows for multiple freeway systems to be easily represented and loaded into the simulator as a graph data structure with edge weights determined by the current speeds. Traffic and graph algorithms can be created and executed for the entire freeway system or for individual vehicles, and the traffic data used by the simulator can be user-generated or be converted from real-time data gathered by a transportation organization. The vehicles in FreeSim can communicate with the system monitoring the traffic on the freeways, which makes FreeSim ideal for ITS simulation. FreeSim is licensed under the GNU General Public License, and the source code is available for download from <http://www.freewaysimulator.com>.**

## I. INTRODUCTION

Simulators provide a convenient way for testing a specific application before deploying it in a live environment. With intelligent transportation systems (ITS), traffic simulators are used to try to determine what the result will be before a specific application is deployed within a transportation system. Traffic simulators can be classified as macroscopic or microscopic, depending on whether they simulate the overall flow of traffic within a transportation system or are focused on individual vehicles within the system. In this paper, we describe an open-source traffic simulator called FreeSim, which provides both macroscopic and microscopic capabilities.

FreeSim is a freeway traffic simulator licensed under the GNU General Public License (GPL) [1]. FreeSim’s graphical user interface (GUI) runs within a web browser using the Adobe Flash plug-in and connects to a Java-based server application via a socket. The Java code uses a MySQL database for any persistent storage needed.

FreeSim allows for different freeway systems to be easily represented by a graph data structure. The data needed to represent the graph of a freeway system in FreeSim are the distances of the freeway segments (edges) and the nodes to which the freeway segments connect (vertices). From the nodes and vertices, the freeway system will be rendered in a GUI with the amount of time necessary to traverse each freeway segment determined by the distance and the speed of that segment.

FreeSim was created with the idea of an intelligent transportation system that gathers data from the individual vehicles traveling within the freeway system. The vehicles are independent entities that are executing as separate threads, communicating with the simulator as they desire. As the vehicles are traversing freeway segments, they can send their current speed and location to a central server. They can also request the shortest path to their destination or the fastest path based on the current speeds. Once they receive the updated path, the vehicles can make their own decisions as to whether or not to change their current path.

FreeSim provides extensibility by allowing programmers to create their own algorithms to run on the data in the simulator. Shortest path algorithms based on distance and fastest path algorithms based on time can easily be created by implementing two methods. Six fastest path algorithms are already implemented in FreeSim: Dijkstra’s Algorithm [2], Bellman-Ford’s Algorithm [3,4], Johnson’s Algorithm [5], and the three All-Pairs All-Paths Pre-Computed algorithms [6].

Further, FreeSim allows user-generated data or real data, such as that gathered by a transportation organization, to be used to update the edges in the graph during a simulation period. The progress of individual vehicles or the status of the freeway system as a whole can then be tracked to see how the travel times and congestion change with the updated data.

The remainder of the paper is organized as follows. In section II we provide a description of the related work and other traffic simulators, and section III gives an overview of FreeSim. Section IV identifies a number of questions that FreeSim can answer, and the conclusion is given in section V.

## II. RELATED WORK

Many transportation applications currently exist and are still being developed. We have classified seven popular traffic applications (CORSIM/TSIS [7,8], FreeSim, MITSIM [9], PARAMICS [10], RENAISSANCE [11], VATSIM [12], and VISSIM [13]) into eight different categories: platform, source code availability, cost, transportation network, underlying method of use, vehicle model, data input manner, and data gathering manner. Table 1 provides a summary of this data.

The “Application” field provides a list of the different traffic applications we are comparing with the organization

TABLE I. SIMULATOR CHARACTERISTICS

Application	Platform	Source Code Availability	Cost	Transportation Networks	Underlying Method of Use	Vehicle Model	Data Input Manner	Data Gathering Manner
CORSIM/TSIS (FHWA)	Windows	Closed	Paid	Free-Flowing (FRESIM), Regulated (NETSIM)	Simulator	Micro	Discrete	Traditional
FreeSim (USC)	Any	Open	Free	Free-Flowing	Simulator	Macro, Micro	Continuous, Discrete	Individual Vehicles, Traditional
MITSIM (MIT)	Linux	Open	Free	Free-Flowing, Regulated	Simulator	Micro	Discrete Transformed	Traditional
PARAMICS (Quadstone Ltd.)	Linux, Solaris, Windows	Closed	Paid	Free-Flowing, Regulated	Simulator	Micro	Discrete	Traditional
RENAISSANCE (Technical Univ. de Crete)	Windows	Closed	Paid	Free-Flowing	Simulator	Macro	Discrete Transformed	Traditional
VATSIM (Ohio State Univ.)	Linux	N/A	N/A	Free-Flowing, Regulated	Emulator	Micro	Simulated Continuous	Traditional
VISSIM (PTV)	Windows	Closed	Paid	Free-Flowing, Regulated, Railways	Simulator	Macro (VISUM), Micro	Discrete	Traditional

that created the application in parentheses. CORSIM is the only application created for a government organization, which is the United States Federal Highway Administration (FHWA). FreeSim, MITSIM, RENAISSANCE, and VATSIM were created by universities, and PARAMICS and VISSIM were created and are supported by private companies.

The “Platform” shows on which operating system the application will run. There is close to an even split between the Windows and Linux applications, and FreeSim allows execution on any operating system since the entire application is written in Flash and Java, which are platform independent.

The “Source Code Availability” is critical for determining how extensible the application is. FreeSim and MITSIM are both open-source, licensed under the GNU GPL [1] and the MITSIMLab Open Source License [14], respectively. The other applications are extensible, but only based on the application programming interface (API) provided by the developers of the software.

The “Cost” data happens to correspond directly with the “Source Code Availability” data, where applications that are open-source are also provided for free, whereas applications that do not provide source code require a fee for licensing.

The “Transportation Networks” field describes the types of roadways that are available in the applications. All of the applications support free-flowing roadways (such as freeways, highways, interstates, etc.) and CORSIM, MITSIM, PARAMICS, VATSIM, and VISSIM all support traffic-regulated roadways, such as roads with traffic signals, stop signs, toll booths, etc. VISSIM also allows railways, bicycle traffic, and pedestrians.

The “Underlying Method of Use” allows users to know where the application can be used. Although they all could be used in a driving emulator, VATSIM is the only tool specifically designed for that. The rest of the applications are considered simulators since they are attempting to model a transportation system and show the system based on a certain vehicle model. Maroto, et.al., provide a good overview of other driving emulators, as well as providing another driving emulator with a new driver model in [15].

The “Vehicle Model” specifies whether the application shows the transportation system from a bird’s eye view (macroscopic) or as viewed by an individual vehicle (microscopic). RENAISSANCE is the only application that solely models the vehicles macroscopically, and FreeSim and VISSIM both allow for macroscopic and microscopic modeling of vehicles. An overall view of the transportation system can be seen, and an individual vehicle’s progress can be tracked in those simulators.

The “Data Input Manner” describes whether the data used to determine the locations and speeds of the vehicles is done continuously or just by giving their speed at a discrete location and then having the vehicle maintain that speed until the next discrete location in its path. VATSIM simulates a continuous flow since the vehicles in the application change speeds using algorithms based on the vehicles surrounding them coupled with a psychological driver-behavior algorithm. MITSIM and RENAISSANCE use discrete data; however they extend the discrete data to be uniformly continuous through an algorithm such as the traffic state estimator algorithm. For more information on how MITSIM and RENAISSANCE convert discrete data into uniformly continuous data, refer to [9] and [11],

respectively. FreeSim allows the data being sent to the vehicles to be continuously updated by the application, and then each individual vehicle can decide whether or not to respond to the received information.

The “Data Gathering Manner” describes how the data is gathered while the simulation is executing. In FreeSim, the individual vehicles communicate their speed and location back to the application, which is then fed into the algorithms for updating the fastest paths for all of the vehicles. As in the other applications, the data is also gathered in a traditional manner, similar to the way in which transportation organizations use loop detectors, video cameras, or other devices measuring data at discrete locations.

Although each of the applications described have certain advantages over other applications, FreeSim provides three main advantages. First of all, FreeSim is open-source and free to download. Second, the data used to simulate the vehicles in the freeway system can be sent in a continuous or a discrete manner. And finally, which is the main reason we created our own simulator rather than using an existing one, FreeSim allows communication between a central system and each individual vehicle. Much of the research concerning intelligent transportation systems assume that the vehicles are able to communicate with a central server, and FreeSim, unlike other simulators, has this ability built into the framework.

### III. FreeSim OVERVIEW

FreeSim is a traffic simulator that models free-flowing transportation systems as a weighted directed graph. The edges of the graph are the freeway segments that the user would like to monitor, and the nodes are the connections between the segments. There is no limit as to how many edges can emanate from a node, although in most freeway systems this number will not exceed more than about eight (i.e. a 4-freeway interchange like the 4-level interchange in Los Angeles that connects the 5, 10, 60, and 110 freeways). A freeway system is stored in a database with FreeSim, though there is a second program bundled with FreeSim for reading a list of nodes and a list of edges that define a freeway system from text files and populating the database accordingly. Multiple freeway systems can be stored in the same database with unique identifying names.

The graphical user interface for FreeSim renders a freeway system in a browser via the Adobe Flash 8 plug-in (see Figure 1). Over a socket connection, the Flash front-end connects to a Java-based server program for all of the simulator functionality. The Flash interface merely provides a light-weight GUI for displaying the output of the simulator.

FreeSim allows shortest paths and fastest paths to be determined based on distance or current speeds, respectively. The shortest paths between all pairs of nodes are determined at start-up using Johnson’s algorithm, although the fastest path must be determined at query time

since it will change based on the current speeds. Fastest path algorithms can be created and executed in FreeSim quite easily by registering a Java class with the traffic simulator that implements two methods – one for retrieving a fastest path from a source node to a destination node, and one for updating the weight of an edge with a new speed. A graph data structure containing the freeway system is also accessible by the class. The method for retrieving the fastest path is called when a user or vehicle requests a fastest path, and the method for updating the weight of an edge with a new speed is called when the speed on an edge is updated to a speed for which the difference is greater than a certain threshold, as defined in [6]. FreeSim already has six fastest path algorithms implemented: Dijkstra’s Algorithm [2], Bellman-Ford’s Algorithm [3,4], Johnson’s Algorithm [5], and the three All-Pairs All-Paths Pre-Computed algorithms [6].

In the FreeSim interface, the speed on an edge can be updated, which causes the method for updating the weight of an edge to be called for all of the fastest path algorithms that have been registered with the traffic simulator. The speed and time to traverse each edge at that speed can be viewed by placing the mouse on the top of a node. Those speeds and times will change in real-time as the simulator receives speed updates from vehicles.

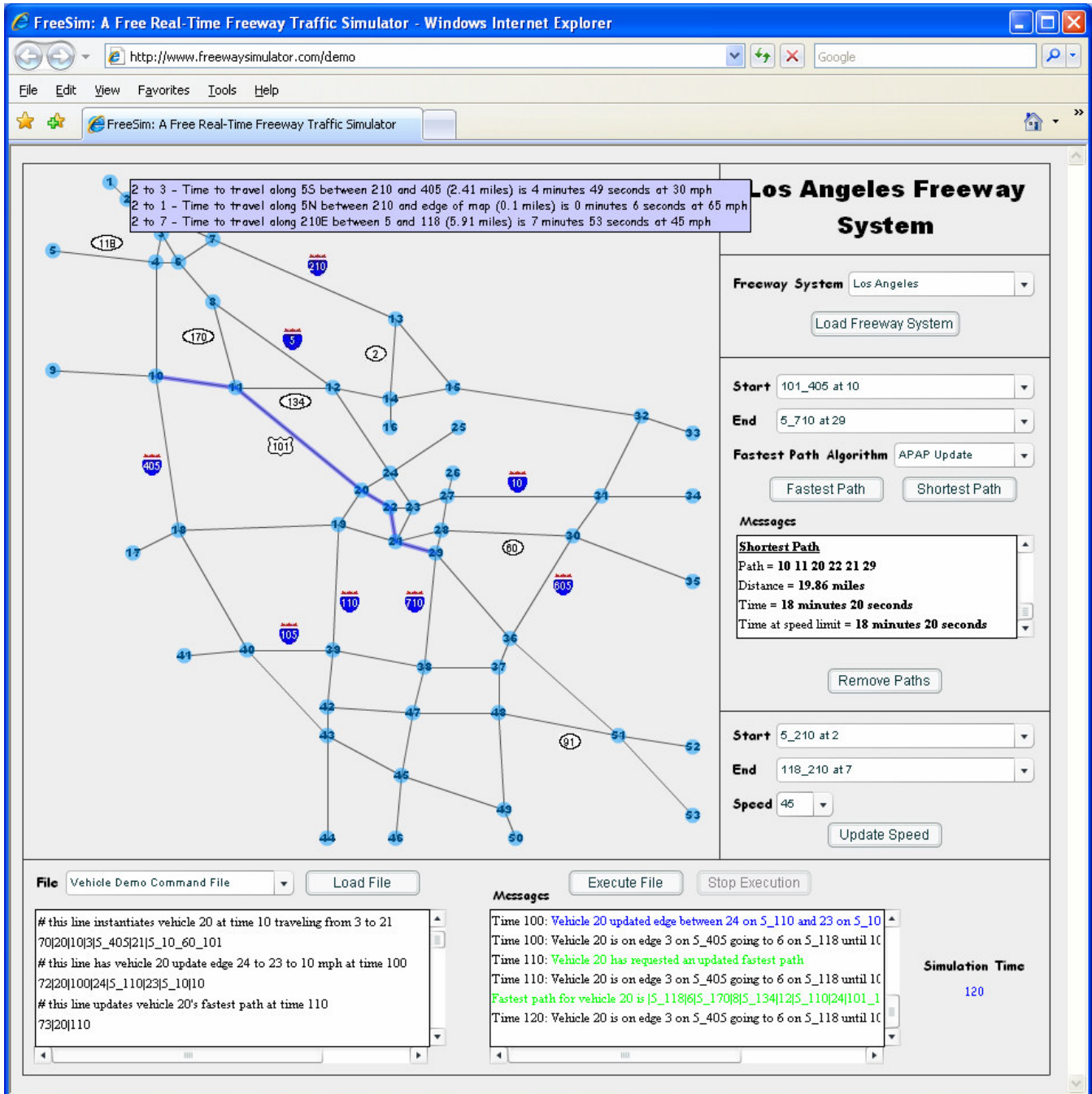
In addition to the user being able to find shortest and fastest paths and update the speed on a freeway segment, FreeSim also allows a user to have a series of commands executed simultaneously during a simulation. A file can be created that allows a user to specify different events that will occur at discrete times during the course of a simulation. Among these events are vehicles being inserted into the freeway system at a certain source node bound for a destination node, vehicles sending updated speeds for a specific edge, vehicles requesting a new fastest path based on the current speeds, and messages that allow a user to track the progress of a vehicle along its path.

One possible use of this input file is to load the freeway system with live speed data, such as that gathered by a transportation organization. The data obtained by loop detectors can be used to obtain the speed of the vehicles crossing that detector during a given time period [16, 17], which can then be used to create the input file to be executed by FreeSim. This gives FreeSim the ability to simulate the traffic within a freeway system based on live data. With this approach, FreeSim does not generate any traffic data of its own, which removes the concern that it does not accurately simulate real traffic within a freeway system, although it does allow vehicles to travel within the system using the live data. Based on having an accurate simulation of real traffic data, there are many potential applications and research questions that can be answered.

### IV. FreeSim SIMULATIONS

FreeSim has been executed with user-generated data and live data gathered from the California Department of

FIGURE 1. SCREENSHOT OF FreeSim GUI



Transportation (CalTrans). Any research based on vehicle travel times would be relatively easy to conduct within FreeSim, such as determining the optimal time of the day to travel based on the speeds or occupancy of the freeway system or observing the change in time to travel based on the time of the day at which a vehicle departs from a location. Determining the optimal speed of vehicles to provide the maximum throughput for a specific freeway

system could also be measured, as was reported in [20] for the Los Angeles freeway system.

In addition, any research concerned with the distance traveled during a certain time frame based on changing speeds could be determined. Using the amount of time a vehicle is traveling with the distance traveled during a certain time frame, coupled with average gas mileage of vehicles and price per gallon of gas, the number of gallons of fuel and the amount of money that could have

been saved by avoiding traffic could be calculated. All of this data can be used to aid in improving the public transportation systems of a city by providing alternate routes during times of heavy traffic or by giving the systems more accurate arrival and departure times.

Based on the live data gathered by a transportation organization, the added traffic due to an incident (such as a collision, road construction, etc.) can be determined. Alternatively, as an event that attracts a significant number of vehicles begins or ends (such as a sporting event), the effect on the traffic of a freeway system of inserting or removing that many vehicles at a certain node can be measured. Similarly, if a new building were proposed to be built that would attract a number of vehicles each day during a certain time interval, the impact of the added traffic within a freeway system could be simulated. As new freeways are proposed, the overall improvement in the flow of traffic in the entire freeway system could be measured.

From a more theoretical standpoint, graph and traffic algorithms can be tested on user-generated or real traffic data. Shortest path algorithms with changing edge weights (also known as dynamic shortest path algorithms) can be implemented through an API in FreeSim. Traffic prediction algorithms, such as those in [18] and [19], can not only be tested, but can also be verified against live traffic data. Incident identification algorithms can be executed to determine how long it takes to determine that an incident has occurred based on real traffic data.

Focusing more on ITS technology, figuring out how many (or what percentage of) vehicles need to transmit their speed and location to a central traffic server to be able to accurately route vehicles along fastest paths can be determined. With individual vehicles having the ability to communicate as autonomous entities, aggregation algorithms or peer-to-peer approaches can be implemented among the vehicle objects. Having more granular information such as speeds of vehicles in specific lanes gives the ability to determine if lanes are traveling faster than other lanes and therefore saving time for those vehicles.

Although we can not possibly enumerate all of the questions that can be answered using a traffic simulator, this at least provides a few of the open research questions that can be solved using FreeSim.

## V. CONCLUSION

In this paper, we have described FreeSim, which is an open-source simulator that can be downloaded for free from <http://www.freewaysimulator.com>. It offers many advantages over other simulators, with one such advantage being the ability to have vehicles traveling within a transportation system remain in communication with a central system. With many intelligent transportation systems assuming that vehicle speed and

location data will be available, FreeSim provides an ideal test-bed for ITS applications.

FreeSim allows for real traffic data, such as that gathered by transportation organizations, to be used within a simulation, which enables more accurate and credible analyses of traffic scenarios. The GUI provided with FreeSim is platform-independent and can be run within a browser. Although there are many traffic simulators available, FreeSim is free, open-source (offered under the GNU GPL), fully-extensible, operates on real traffic data, and provides the necessary framework for current and future ITS applications.

## VI. REFERENCES

- [1] GNU General Public License. <http://www.gnu.org/licenses/gpl.html>.
- [2] Dijkstra, E.W. "A note on two problems in connexion with graphs." *Numerische Mathematik*, 1959.
- [3] Bellman, Richard. "On a Routing Problem", *Quarterly of Applied Mathematics*. Volume 16, Issue 1, 1958.
- [4] Ford Jr., Lester R., D.R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.
- [5] Johnson, Donald. "Efficient Algorithms for Shortest Paths in Sparse Networks." *Journal of the ACM*, 1977.
- [6] Miller, Jeffrey, Ellis Horowitz. "Algorithms for Real-Time Gathering and Analysis of Continuous-Flow Traffic Data." *IEEE 9<sup>th</sup> International Intelligent Transportation Systems Conference*, September 17-20, 2006.
- [7] Corsim. <http://ops.fhwa.dot.gov/trafficanalysisitools/corsim.htm>
- [8] Owen, Larry E., Yunlong Zhang, Lei Rao, Gene McHale. "Traffic Flow Simulation Using CORSIM." *Proceedings of the 2000 Winter Simulation Conference*, December 10-13, 2000.
- [9] Yang, Qi, Haris N. Koutsopoulos. "A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems." *Transpn Res., Part C*, Volume 4, Number 3, 1996.
- [10] Cameron, Gordon, Brian J.N. Wylie, David McArthur. "Paramics: Moving Vehicles on the Connection Machine." *Proceedings of the 1994 ACM/IEEE Conference on Supercomputing*, November 14-19, 1994.
- [11] Wang, Yibing, Markos Papageorgiou, Albert Messmer. "RENAISSANCE: A Real-Time Freeway Network Traffic Surveillance Tool." *IEEE 9<sup>th</sup> International Intelligent Transportation Systems Conference*, September 17-20, 2006.
- [12] Redmill, Keith A., Umit Ozguner. "VATSIM: A Vehicle and Traffic Simulator." *IEEE 2<sup>nd</sup> International Intelligent Transportation Systems Conference*, October 5-8, 2001.
- [13] Fellendorf, Martin. "VISSIM: A Microscopic Simulation Tool to Evaluate Actuated Signal Control

Including Bus Priority.” *64<sup>th</sup> Institute of Transportation Engineers Annual Meeting*, October 1994.

[14] MITSIMLab Open Source License. <http://mit.edu/its/MITSIMLabOSnew.html#license>.

[15] Maroto, Joaquin, Eduardo Delso, Jesus Felez, Jose Ma. Cabanellas. “Real-Time Traffic Simulation With a Microscopic Model.” *IEEE Transactions on Intelligent Transportation Systems*. Volume 7, Number 4, December 2006.

[16] Zhanfeng, Jia, Chao Chen, Ben Coifman, Pravin Varaiya. “The PeMS algorithms for accurate, real-time estimates of g-factors and speeds from single-loop detectors.” *IEEE 4<sup>th</sup> International Intelligent Transportation Systems Conference*, February 12, 2001.

[17] Petty, Karl F., Peter Bickel, Jiming Jiang, Michael Ostland, John Rice, Ya’acov Ritov, Frederic Schoenberg. “Accurate Estimation of Travel Times from Single-Loop Detectors.” *Transportation Research, Part 1 (Policy and Practice)*, Volume 32A, #1, January 1998.

[18] Vlahogianni, Eleni, Matthew Karlaftis, John Golias, Nikolaos Kourbelis. “Pattern-Based Short-Term Urban Traffic Predictor.” *IEEE 9<sup>th</sup> International Intelligent Transportation Systems Conference*, September 17-20, 2006.

[19] Quek, Chai, Michel Pasquier, Bernard Boon Seng Lim. “POP-TRAFFIC: A Novel Fuzzy Neural Approach to Road Traffic Analysis and Prediction.” *IEEE Transactions on Intelligent Transportation Systems*, Volume 7, Number 2, June 2006.

[20] Zhanfeng, Jia, Pravin Varaiya, Chao Chen, Karl Petty, Alex Skabardonis. “Maximum throughput in LA freeways occurs at 60 mph.” *Berkeley Technical Report*, January 16, 2001.