A Discrete-Event and Hybrid Simulation Framework Based on SimEvents for Intelligent Transportation System Analysis ⋆

Yue Zhang * **Christos G. Cassandras** *,** **Wei Li** *** **Pieter J. Mosterman** ***

* *Division of Systems Engineering*
** *Department of Electrical and Computer Engineering, Boston University, Brookline, MA 02446 USA (e-mail: joycez, cgc@bu.edu)*
*** *MathWorks, 1 Apple Hill Drive, Natick, MA 01760 USA (e-mail: wei.li, pieter.mosterman@mathworks.com)*

Abstract: Intelligent transportation systems combine physical elements with cyber components based on information and communication technologies and the use of control methodologies for Connected Automated Vehicles (CAVs). Intelligent transportation systems, therefore, contain event-driven dynamics along with time-driven dynamics. The hybrid nature of such systems motivates the development of new simulation platforms in order to test and evaluate their effectiveness. A discrete-event and hybrid simulation framework based on SimEvents is introduced within which these systems can be studied at the microscopic level. This framework enables users to apply different control strategies as well as communication protocols for CAVs and to carry out performance analysis of proposed algorithms by authoring customized discrete-event and hybrid systems that include various design paradigms such as entity flow, graphical programming, and object-oriented programming in MATLAB®. These paradigms provide users with the flexibility to select or combine modeling elements for achieving complex goals as the demonstrated scenarios in the paper illustrate. The framework spans multiple toolboxes including MATLAB, Simulink®, and SimEvents®.

Keywords: discrete event systems, hybrid systems, traffic control, intelligent transportation systems, smart cities

1. INTRODUCTION

An Intelligent Transportation System (ITS) combines time-driven dynamics governing its physical components with event-driven dynamics characterizing its cyber elements. The physical elements may include the infrastructure and vehicles, while the cyber components involve a variety of communication technologies, information processing, and control systems methodologies that aim at traffic flow optimization by exploiting vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication and make use of information fusion from multiple traffic sensing modalities.

The advent of CAVs provides the automotive industry with an unprecedented opportunity by enabling users to better monitor a transportation system's conditions, hence improving traffic flow in terms of reducing congestion as well as energy consumption and greenhouse gas emissions, while also improving safety. From a CAV's perspective, the physical domain is defined by vehicle mechanics, motion dynamics, etc., while the cyber domain involves the capability to sense the surroundings, communicate through V2V or V2I, and implement advanced control algorithms.

⋆ Supported in part by NSF under grants ECCS-1509084, CNS-1645681, and IIP-1430145, by AFOSR under grant FA9550-15-1-0471, by DOE under grant DOE-46100, by MathWorks and by Bosch.

Recent advances in CAVs focus on issues such as optimizing powertrain configurations, for instance, in plug-in hybrid electric vehicles (PHEV), as well as improving traffic conditions in terms of reducing travel delay and energy consumption. Examples include Lee and Park (2012) where the overlap between vehicle positions is minimized, and Gilbert (1976),Hooker (1988), Hellström et al. (2010), Li et al. (2012) where the focus is on improving energy economy. To evaluate the effectiveness of emerging proposed methodologies, a good way is to conduct field tests involving actual vehicles as discussed in Shladover et al. (1991) and Kato et al. (2002). Such tests take actual environmental factors into consideration, thus lending them credibility. However, field tests are often infeasible. In view of these factors, a suitable ITS simulation environment is needed.

The focus of this paper is on building microscopic transportation models allowing the evaluation of different information and communication technologies (ICT) and control algorithms. Microscopic models usually track individual transportation elements on a continuous-time basis; for instance, they must track the position of all vehicles. However, transportation systems must respond to events, some of which are random, such as vehicle arrivals or bad weather, while others are controllable, such as routing decisions or traffic light switches. Consequently, traffic models must be both event-driven and time-driven.

There are many traffic simulation platforms that can operate at the microscopic level, such as VISSIM (Fellendorf (1994)), PARAMICS (Cameron and Duncan (1996)), CAR-

SIM (Benekohal and Treiterer (1988)) and SUMO (Krajzewicz et al. (2002)), all of which offer a wide range of methods to design and evaluate traffic systems. As CAVs make use of more sophisticated and increasingly efficient control algorithms that heavily rely on sensing the transportation environment, we need platforms which are able to consider a large number of different traffic scenarios and encompass all different aspects of ITS operation. An example of such a platform is PreScan, which accommodates CAVs and Advanced Driver Assistance Systems (ADAS) based on sensor simulation and flexible scenario definition. The tool ITS Modeller proposed in Versteegt et al. (2009) complements PreScan in terms of evaluation at a traffic network level.

One common feature of the aforementioned traffic simulation platforms is the integration of MATLAB and Simulink via an interface that allows a user to design ICT methods and control algorithms. Examples can be found in Zhang et al. (2016), where a decentralized optimal control algorithm is implemented using MATLAB and applied to each vehicle, with the resulting vehicle behavior visualized and evaluated through VISSIM. This illustrates the powerful capabilities of MATLAB and Simulink as a test bed for ICT approaches and control algorithms. In some cases, a discrete-event simulation model cannot only capture event-driven behavior, but also abstract continuous-time components through event-driven components. As SimEvents provides users with various paradigms for building a discrete-event simulation model, we take advantage of the Discrete Event System (DES) simulation framework introduced in SimEvents®.

Earlier work in Zhang et al. (2017a) introduced a new traffic simulation framework based on SimEvents in conjunction with MATLAB and Simulink. This framework offers access to both physical components and cyber components which typically involve different ICT approaches and control strategies. Combined with the discrete-event/continuous-time hybrid simulation engine of the original SimEvents (Clune et al. (2006)), the simulation model includes both discrete-event components implemented by SimEvents, and continuous-time components implemented by Simulink. Thus, the overall traffic simulation framework is a hybrid dynamic model.

An important feature of the proposed traffic simulation framework is the capability for users to easily create different scenarios under which users can test various ICT methods and control algorithms. This paper describes and elaborates the capability of this hybrid simulation framework and includes demonstrations of how it can operate under various scenarios.

The paper is structured as follows. Section 2 reviews the hybrid traffic simulation framework introduced in Zhang et al. (2017a). Section 3 discusses in detail the implementation of the hybrid traffic simulation platform using SimEvents in conjunction with MATLAB and Simulink. Section 4 illustrates the effectiveness of the traffic simulation platform by demonstrating several different scenarios. Section 5 concludes with remarks and an outline of further research activities.

2. A DISCRETE-EVENT AND HYBRID TRAFFIC SIMULATION FRAMEWORK

To create a traffic simulation framework for vehicle behavior evaluation, a system consisting of physical elements (infrastructure and vehicles), cyber components (traffic control, communication, sensing technologies), and events is necessary.

Table 1. Infrastructure

Infrastructure	Property	Function
Road segment	length, number of lanes	sensing (sensors, cameras)
Merging zone	length/width, left/right turns	sensing (sensors, cameras)
Controller	control strategy, range	control, communication
Coordinator	range	communication

Table 2. Vehicles

Vehicle	Property & Dynamics
Properties	ID, acceleration, speed, position, lane, mpg, etc.
Motion dynamics	basic model, Kinematic model, Dynamic model, etc.
Control dynamics	optimal control, MPC, etc.
Fuel dynamics	gasoline engine, electric, hybrid, plug-in hybrid, etc.

Table 3. Events

Vehicle-to-Infrastructure Warning	Vehicle-to-Vehicle Warning
Red Light Violation	Emergency Electronic Brake Lights
Curve Speed	Forward Collision
Stop Sign Gap Assist	Intersection Movement Assist
Spot Weather Impact	Left Turn Assist
Reduced Speed/Work Zone	Blind Spot/ Lane Change
Pedestrian in Signalized Crosswalk	Do Not Pass
	Vehicle Turning Right in Front of Bus

The system is designed to comprise various elements of the SimEvents paradigms such as entities, queues, servers, terminators, and customized MATLAB Discrete Event Systems.

The model introduced in Zhang et al. (2017a) is briefly reviewed. There are three basic elements in the hybrid traffic simulation framework: infrastructure, vehicles, and events.

Infrastructure consists of roadside facilities that enable communication and carry out traffic management.

Vehicles can differ in motion dynamics, driver behavior models, fuel dynamics, and so on. For CAVs, they should also possess the ability to communicate with each other.

Events can be categorized into two classes in the hybrid systems: exogenous and endogenous events. Exogenous events include those originating from the outside world and force certain elements to change the behavior, for instance, an unexpected storm. Endogenous events occur when a time-driven state variable enters a particular set, for instance, the inter-vehicle distance falling below the minimum safe following distance, which may indicate a possible impending rear-end crash. Depending on whether they occur among vehicles or between vehicles and infrastructure (Bettisworth et al. (2015)), the events can also be categorized as listed in Table 3.

Figure 1 depicts the architecture of the traffic simulation framework and shows how different elements are connected. Certain elements should be capable of communicating with others. In this paper, it is assumed that only V2I communication is active, as V2V communication can be achieved through V2I2V communication. An important feature of the proposed framework is the inclusion of communication delay, as low packet delays are necessary for implementing the control algorithms employed by CAVs. In Fig. 1, the servers are used to model the communication delays.

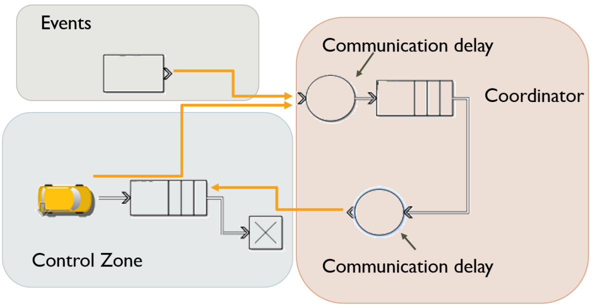

Fig. 1. Architecture of the hybrid traffic simulation framework.

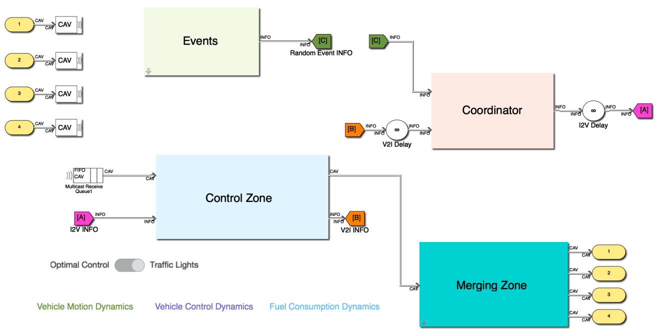

Fig. 2. Simulink® model of an intersection with four road segments (input/outputs).

The output of *Events* into the coordinator indicates that the coordinator is aware of all the event-based information by means of sensing and communication. Once a certain event occurs the coordinator will broadcast or send the information to the vehicles that might be affected, so that the vehicles can make appropriate decisions.

The *continuous* part in this framework includes vehicle motion dynamics, control dynamics, and fuel consumption dynamics that should be tracked continuously. For the *discrete-event* part, events are considered as they can affect vehicle behavior.

3. IMPLEMENTATION

The proposed hybrid traffic simulation framework is built based on MATLAB and Simulink that include various programming paradigms (Li et al. (2016)). The incorporation of SimEvents offers tools to work with discrete event components. The various programming options offer users a platform for rapid prototyping that is widely used in the automotive industry. The paradigms used in the proposed model include *Entity Flow*, *Graphical Programming*, and *Textual Programming*. The model structure of a single intersection is shown in Fig. 2.

3.1 Entity Flow

Entities are the discrete items of interest carrying a rich set of attributes, which can pass through a network of queues and servers during the discrete-event simulations. In the transportation modeling context, an entity can represent a vehicle, whose attributes may include ID, acceleration, speed, position, lane, destination, and so on (see the example shown in the yellow rectangle of Fig. 3).

3.2 Graphical Programming

The graphical programming paradigm enables users to work with discrete event components, whereby users can specify various functions associated with events such as entity entry and exit. These functions are event-driven actions, that is, they can only be triggered by a different class of events. For instance, in Fig. 3, a series of functions are defined in the red rectangle that can only be executed when the CAV is generated. These functions specify how the attributes, for instance, speed, of vehicles are initialized when they are generated.

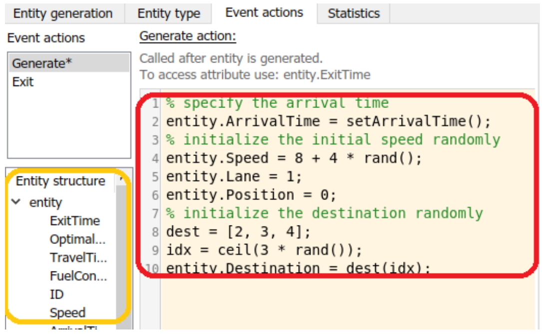

Fig. 3. Customized event actions of CAV generator block.

3.3 Textual Programming

The MATLAB Discrete Event System provides maximal flexibility as it offers users the capability to author an event-driven entity-flow system arbitrarily using object-oriented programming in MATLAB, whose functionality is expandable by incorporating functions from other MATLAB toolboxes.

```
% CAV arrives at the control zone
function [entity, events] = CAVEntryImpl(obj, storage, entity, ~)...
% CAV generates an info packet and sends it to the coordinator
function [entity, events] = INFOGenerateImpl(obj, ~, entity, tag)...
% CAV receives info from the coordinator and (re-)computes the
% control
function [entity, events] = INFOEntryImpl(obj, storage, entity, ~)...

% persistently monitor and track the status of CAVs by setting
% timers and iterators for evaluation purposes
function [entity, events] = CAVTimerImpl(obj, ~, entity, tag)...
function [entity, events, next] = CAVIterateImpl(obj, storage, ...
        entity, tag, ~)...
```

Fig. 4. The MATLAB® Discrete Event System for Control Zone (partial codes).

For instance, the control zone shown in Fig. 2 was designed using a MATLAB Discrete Event System in the proposed traffic model. The program specifies the properties of the control zone as well as the definition of different storages that contain user-defined entities, that is, CAVs and information packets (INFOs). In addition, different event actions are defined through methods as shown in Fig. 4.

4. DEMONSTRATIONS

The effectiveness of the proposed hybrid framework has been demonstrated through simulation under different traffic scenarios: 1) a scenario with only CAVs, 2) a scenario with only non-CAVs, 3) a mixed-traffic scenario where CAVs and non-CAVs co-exist. For each scenario, the control methodology is introduced first and then the demonstration examples are presented.

4.1 A Scenario with Only CAVs

In the first scenario, only CAVs are being considered. As shown in Fig. 5, the region at the center of each intersection, called *Merging Zone* (MZ) is the area of potential lateral CAV collision, which is taken to be a square. Each intersection has

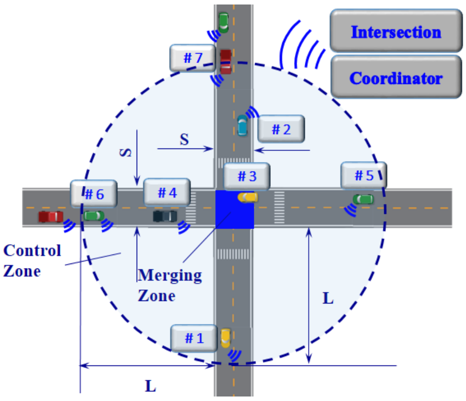

Fig. 5. Connected Automated Vehicles (CAVs) crossing an urban intersection.

a *Control Zone* (CZ) and a coordinator that can communicate with the CAVs traveling within the CZ. A M/M/1 queueing system (Cassandras and Lafortune (2009)) following a first-in-first-out order is assumed for CAVs that have entered the CZ.

The decentralized optimal control framework, introduced in Zhang et al. (2016), is used for optimally controlling CAVs crossing a signal-free intersection with the objective of minimizing energy consumption inside the CZ, subject to a throughput maximization requirement formulated in Malikopoulos et al. (2017). The rear-end safety can be ensured through an appropriately designed Feasibility Enforcement Zone (FEZ) that precedes the CZ, as discussed in Zhang et al. (2017b).

Demonstration examples **A group of CAVs crossing a single urban intersection** is considered, where the length of the CZ is $L = 400$m and the length of the MZ is $S = 30$m. For each direction, only one lane is considered. The minimum safe inter-vehicle distance is set to $\delta = 10$m. The vehicle arrivals are assumed to be given by a Poisson process and the initial speeds are uniformly distributed over $[8, 12]$m/s.

In the current framework, simple information exchange is assumed between vehicles and the coordinator. Every time a CAV enters the CZ, it sends information to the coordinator indicating its arrival. After a certain period of communication delay, which is modeled using servers, the coordinator sends relevant information back and based on that, the CAV can make decisions regarding the remaining trip.

For evaluation purposes, the position of the CAV must be continuously monitored and tracked. This represents the continuous time-driven component of the simulation framework. Combined with the discrete event-driven component, such as vehicle arrivals, the two together form the *hybrid* nature of the simulation framework.

A snapshot of the demonstration example is shown in Fig. 6, where the color represents the direction that the vehicle comes from. Note that the proposed simulation framework is capable of generating state displays, which can easily be achieved by incorporating functions from other MATLAB toolboxes. For instance, the optimal control output and the corresponding speed trajectories are shown in Fig. 7.

This scenario can be easily modified, for instance, by **including left the right turns** (Fig. 8). The solution to account for left and right turns under hard safety constraints is provided in Zhang et al. (2017c). Note that the proposed simulation

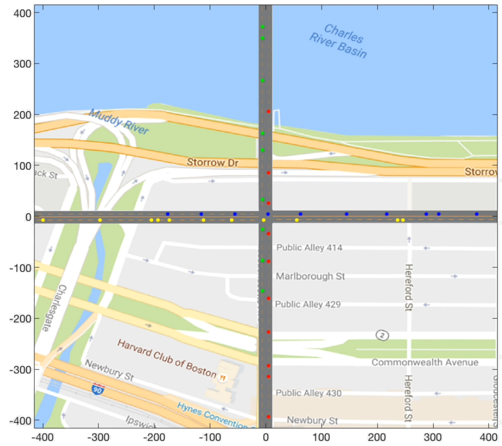

Fig. 6. Optimal control of CAVs crossing an urban intersection.

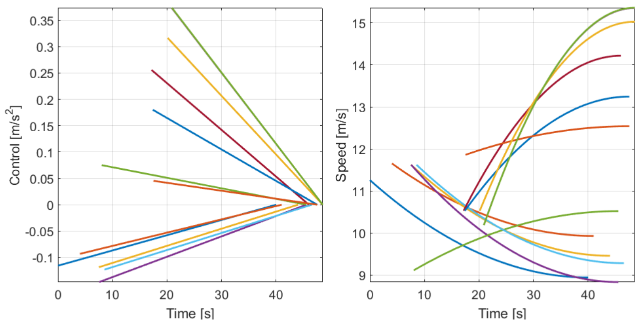

Fig. 7. Control and speed trajectories of the first 10 CAVs under decentralized optimal control framework.

framework is also equipped with performance reports. Fig. 8 shows how average fuel consumption and average travel time can be observed in real time.

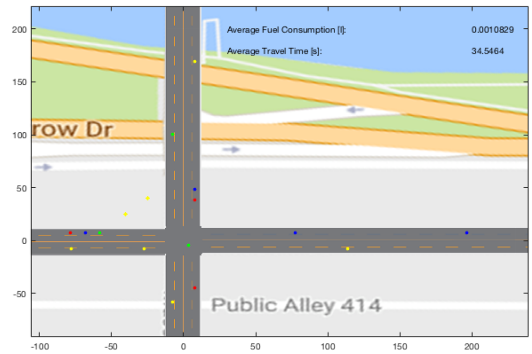

Fig. 8. Optimal control of CAVs including left and right turns.

The traffic simulation model is inherently scalable. For instance, the single intersection scenario can be easily extended to **a multi-intersection scenario** by adding more queues, servers, the MATLAB Discrete Event Systems or a combination of them. A snapshot of the demonstration example for CAVs crossing two adjacent urban intersections is shown in Fig. 9. Note that the two intersections can be coupled in different ways, which mainly depends on the distance between the two intersections D. In Fig. 9, the distance between the two intersections is set to $D = L = 400$m, which indicates that once the CAV exits the upstream MZ, it immediately enters the downstream CZ.

To explore the event-driven feature of the transportation systems, a scenario is created by **randomly generating an event**

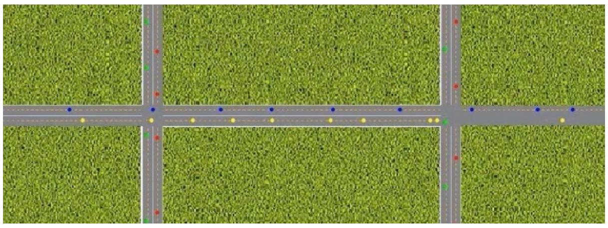

Fig. 9. Optimal control of CAVs crossing two adjacent urban intersections.

from *Events* blocks, for instance, a storm event. Note that the optimal control in Zhang et al. (2016) actually remains unchanged until an event occurs. As the storm may reduce the friction of the road surface, vehicles are forced to decelerate under this scenario and re-calculate the speed profiles. The speed trajectories of the first 3 CAVs are shown in Fig. 10, where the dashed lines and solid lines represent the projected speed trajectories before and after the event (storm) occurs, respectively. Observe that the re-calculation of the speed profiles is only triggered by the storm event. Otherwise, vehicles would proceed according to the original optimal control profiles (dashed lines).

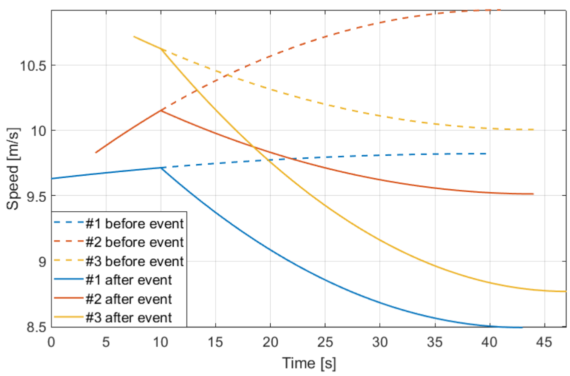

Fig. 10. The speed trajectories of the first 3 CAVs before and after a storm event occurs.

4.2 A Scenario with Only Non-CAVs

To demonstrate the efficiency of new control algorithms, comparisons are usually required. Therefore, it is of great importance that a traffic simulation framework can be adapted to different traffic scenarios without much effort, so that users can make comparisons among different approaches. Here, a baseline scenario is built and tested on the traffic simulation platform where the non-CAVs are controlled by the fixed-cycle traffic lights. In this paper, simple control policies are assumed for the non-CAVs, that is, a non-CAV (1) keeps cruising unless an event occurs that would affect its behavior; (2) decelerates when it approaches a red light; (3) accelerates to the desired speed when the red light turns green.

Demonstration examples A snapshot of the demonstration example for the non-CAVs crossing a signalized intersection under traffic light control is shown in Fig. 11, where both the green and red phases last for 30 seconds. In Fig. 11, queues can be observed that gradually formed in front of the red lights. It was shown in Zhang et al. (2017b) that compared with the baseline scenario, the optimal control of CAVs can achieve 42% improvement in reducing fuel consumption and 37% improvement in reducing travel delay.

Fig. 11. A snapshot of the scenario under traffic light control.

4.3 A Mixed-Traffic Scenario

As challenges remain before massive deployment of fully autonomous vehicles, the mixed-traffic scenario where both CAVs and non-CAVs travel on the roads must be considered. To model this scenario on the proposed traffic simulation platform, different control algorithms are implemented for CAVs and non-CAVs, respectively. For non-CAVs, simple control policies are assumed, that is, a non-CAV keeps cruising if no event occurs that would affect its driving behavior. For CAVs, a two-mode optimal control framework, introduced in Zhang and Cassandras (2018) is applied to minimize fuel consumption white adaptively maintaining a safe inter-vehicle distance if it is constrained by a preceding non-CAV.

An important feature of this framework is the potential to model and simulate communication protocols in order to study the effects of delay in V2V and V2I communications on safety. To cooperate with the control algorithms employed by the CAVs, low packet delay/loss/error is necessary for maintaining safety.

Demonstration examples The following example explores the influence of communication delay on inter-vehicle safety, where a non-CAV #1 is assumed to cruise at its initial speed and CAV #2 enters the CZ immediately after #1 on the same lane. If the inter-vehicle distance between vehicles #2 and #1, denoted as $s_2(t)$, falls below the minimum safe following distance $\delta = 10$m, CAV #2 simply forgoes the optimal control and adaptively follows the non-CAV #1, that is, maintaining the minimum safe following distance and the same speed as #1. Varying the service time of the server, which is used for simulating the communication delay between the CAVs and the coordinator, allows easy experimentation with the influence of large communication delay on the implementation of control algorithms and inter-vehicle safety.

The speed trajectories under different communication delays and the corresponding inter-vehicle distance between vehicles #2 and #1 are shown in Fig. 12. When the communication delay is low, that is, delay = 0.01s, CAV #2 is able to make adjustments in time so that the inter-vehicle distance $s_2(t)$ (red curve) does not fall too much below $\delta = 10$m; after a short period, CAV #2 starts to follow the non-CAV #1 while maintaining a distance of δ with the non-CAV #1. When the communication delay is high, that is, delay = 3s, the inter-vehicle distance $s_2(t)$ falls to 5.5m, which is undesired in terms of maintaining safety; while the communication delay increases to 6s, the inter-vehicle distance falls below 0, which indicates an accident. Using the simulation setting, the operation of the communication protocols can be easily examined and if a crash occurs, the protocols may need to be re-designed.

As shown in the previous demonstrations, the framework can be easily adapted to different scenarios. Since the modules in the

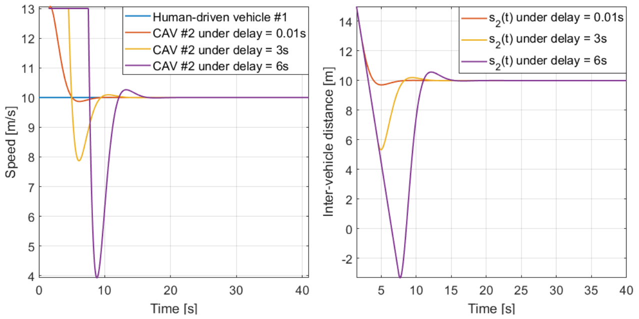

Fig. 12. The speed trajectories and the corresponding inter-vehicle distance under different communication delays.

framework are relatively independent, only the corresponding modules need modification.

5. CONCLUDING REMARKS AND FUTURE WORK

This paper proposed a discrete-event and hybrid simulation framework based on SimEvents for ITS analysis. SimEvents has become a valuable tool for discrete-event and hybrid simulations that fits the goal that the traffic model should be both time-driven and event-driven. The benefits of the simulation framework demonstrated in this paper include (1) abstraction of continuous-time components based on discrete event systems, (2) a modular architecture that allows different system configurations, (3) a framework that can be easily adapted to different traffic scenarios, (4) direct comparison among different simulations by introducing an event, (5) expandable functionality by incorporating functions from other MATLAB toolboxes, (6) scalability by simply adding more queues, servers, and storages. In addition, MATLAB provides users with full access to model details and flexibility to manipulate the model elements, and real-time state displays and performance reports as well.

Ongoing research includes the incorporation of the Dedicated Short Range Communication (DSRC) protocols as it is the key technology for V2V safety communications. Furthermore, more elaborate and diversified non-CAV models are required so as to study the interactions between CAVs and non-CAVs.

REFERENCES

Benekohal, R. and Treiterer, J. (1988). Carsim: Car-following model for simulation of traffic in normal and stop-and-go conditions. *Transportation Research Record*, (1194).

Bettisworth, C., Burt, M., Chachich, A., Harrington, R., Hassol, J., Kim, A., Lamoureux, K., LaFrance-Linden, D., Maloney, C., Perlman, D., Ritter, G., Sloan, S.M., and Wallischeck, E. (2015). Status of the Dedicated Short-Range Communications technology and applications: Report to Congress. Technical report.

Cameron, G.D. and Duncan, G.I. (1996). Paramics-parallel microscopic simulation of road traffic. *The Journal of Supercomputing*, 10(1), 25–53.

Cassandras, C.G. and Lafortune, S. (2009). *Introduction to Discrete Event Systems*. Springer Science & Business Media.

Clune, M.I., Mosterman, P.J., and Cassandras, C.G. (2006). Discrete event and hybrid system simulation with SimEvents. In *Proceedings of the 8th International Workshop on Discrete Event Systems*, 386–387.

Fellendorf, M. (1994). Vissim: A microscopic simulation tool to evaluate actuated signal control including bus priority. In *64th Institute of Transportation Engineers Annual Meeting*, 1–9. Springer.

Gilbert, E.G. (1976). Vehicle cruise: Improved fuel economy by periodic control. *Automatica*, 12(2), 159–166.

Hellström, E., Åslund, J., and Nielsen, L. (2010). Design of an efficient algorithm for fuel-optimal look-ahead control. *Control Engineering Practice*, 18(11), 1318–1327.

Hooker, J. (1988). Optimal driving for single-vehicle fuel economy. *Transportation Research Part A: General*, 22(3), 183–201.

Kato, S., Tsugawa, S., Tokuda, K., Matsui, T., and Fujii, H. (2002). Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *IEEE Transactions on Intelligent Transportation Systems*, 3(3), 155–161.

Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002). Sumo (simulation of urban mobility)-an open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002)*, 183–187.

Lee, J. and Park, B. (2012). Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(1), 81–90.

Li, S.E., Peng, H., Li, K., and Wang, J. (2012). Minimum fuel control strategy in automated car-following scenarios. *IEEE Transactions on Vehicular Technology*, 61(3), 998–1007.

Li, W., Mani, R., and Mosterman, P.J. (2016). Extensible discrete-event simulation framework in SimEvents. In *Proceedings of the 2016 Winter Simulation Conference*, 943–954. IEEE Press.

Malikopoulos, A.A., Cassandras, C.G., and Zhang, Y. (2017). A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. *Automatica*. (to appear).

Shladover, S.E., Desoer, C.A., Hedrick, J.K., Tomizuka, M., Walrand, J., Zhang, W.B., McMahon, D.H., Peng, H., Sheikholeslam, S., and McKeown, N. (1991). Automated vehicle control developments in the path program. *IEEE Transactions on Vehicular Technology*, 40(1), 114–130.

Versteegt, E., Klunder, G., and Arem, B.v. (2009). Modelling cooperative roadside and in-vehicle intelligent transport systems using the its modeller. In *12th World Congress on Intelligent Transport Systems 2005, 6 November 2005 through 10 November 2005, San Francisco, CA, 1558-1567*.

Zhang, Y. and Cassandras, C.G. (2018). The penetration effect of connected automated vehicles in urban traffic: an energy impact study. ArXiv: 1803.05577.

Zhang, Y., Cassandras, C.G., Li, W., and Mosterman, P.J. (2017a). A SimEvents model for hybrid traffic simulation. In *Proceedings of the 2017 Winter Simulation Conference*, 1455–1466. IEEE Press.

Zhang, Y., Cassandras, C.G., and Malikopoulos, A.A. (2017b). Optimal control of connected automated vehicles at urban traffic intersections: A feasibility enforcement analysis. In *Proceedings of the 2017 American Control Conference*, 3548–3553.

Zhang, Y., Malikopoulos, A.A., and Cassandras, C.G. (2016). Optimal control and coordination of connected and automated vehicles at urban traffic intersections. In *Proceedings of the 2016 American Control Conference*, 6227–6232.

Zhang, Y., Malikopoulos, A.A., and Cassandras, C.G. (2017c). Decentralized optimal control for connected automated vehicles at intersections including left and right turns. In *56th IEEE Conference on Decision and Control*, 4428–4433.