Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

A SIMEVENTS MODEL FOR HYBRID TRAFFIC SIMULATION

Yue Zhang

Division of Systems Engineering
Boston University
15 Saint Mary's St.
Brookline, MA 02446, USA

Christos G. Cassandras

Division of Systems Engineering, and
Electrical and Computer Engineering Department
Boston University
15 Saint Mary's St.
Brookline, MA 02446, USA

Wei Li
Pieter J. Mosterman

MathWorks
1 Apple Hill Drive
Natick, MA 01760, USA

ABSTRACT

Intelligent transportation systems are typical Cyber-Physical Systems (CPS) that combine physical components with cyber elements that include communication, information processing and control mechanisms for Connected Automated Vehicles (CAVs). To test and evaluate the efficiency of such systems, new simulation platforms are needed. In this paper, a SimEvents-based framework is introduced for hybrid traffic simulation at the microscopic level. This framework enables users to apply different control strategies for CAVs and carry out performance analysis of proposed algorithms by authoring customized discrete-event and hybrid systems based on *MATLAB Discrete-Event System* using object-oriented MATLAB. The framework spans multiple toolboxes including MATLAB, Simulink, and SimEvents.

1 INTRODUCTION

Intelligent transportation systems (ITS) are typical Cyber-Physical Systems (CPS) that combine time-driven dynamics governing their physical components (infrastructure and vehicles) with event-driven dynamics characterizing their cyber elements (communication, information processing and control mechanism implementation). The interaction between these dynamics leads to the co-evolution of cyber and physical state trajectories. At the heart of an ITS's future are Connected Automated Vehicles (CAVs). From a CAV's perspective, the physical domain is defined by the vehicle motion dynamics, while the cyber domain involves the capability to communicate with other CAVs (V2V) or with the infrastructure (V2I) and to implement advanced control mechanisms. The advent of CAVs provides the automotive industry and transportation network authorities with an unprecedented opportunity to improve the efficiency of the whole transportation system in terms of reducing traffic congestion along with fuel consumption and environmental emissions, as well as drastically improve safety.

Recent advances in CAVs focus on issues such as diversifying powertrain configurations, for instance, in hybrid of battery-powered electric vehicles, as well as connectivity that integrates information and communication technologies (ICT). To evaluate the effectiveness of such new information and communication technology (ICT) approaches and control strategies, a good way is to conduct field tests involving real vehicles as 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; for instance, there may not exist an appropriate infrastructure to test V2I applications. It is also not realistic to use field tests validating a control algorithm intended to coordinate thousands of vehicles in a large scale transportation network. In view of these observations, a suitable traffic simulation environment is needed. Transportation models are normally viewed at different levels of detail and are classified into three types: macroscopic, mesoscopic and microscopic models. Macroscopic models generally deal with transportation elements at an aggregate level and view traffic as an inseparable flow. Microscopic models focus on individual elements and can include driver behavior. Mesoscopic models analyze individual transportation elements in a small group, with all the elements in the group considered homogeneous, for instance, a vehicle platoon. As the evaluation of vehicle behavior under different ICT approaches and control algorithms is based on individual vehicle, the focus of this paper is on microscopic models.

There are many traffic simulation platforms which operate at the microscopic level, such as VISSIM (Fellendorf 1994), PARAMICS (Cameron and Duncan 1996), CARSIM (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 enable complicated but more efficient control mechanisms, testing and validating such functionality requires a large number of different traffic scenarios to be considered. Designing such systems is challenging, since it requires an environment that encompasses all different aspects of traffic dynamics. PreScan is a simulation platform that accommodates CAVs and advanced driver assistance systems (ADAS) based on sensor simulation and flexible scenario definition. After building a scenario out of template components, one can set the vehicle model and implement control algorithms via a MATLAB® (MathWorks® 2016b) and Simulink® (MathWorks® 2016d) interface. The tool ITS Modeller (Versteegt, Klunder, and Arem 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, Malikopoulos, and Cassandras 2016), where an optimal control algorithm is implemented using MATLAB and the resulting vehicle behavior is evaluated based on VISSIM. This illustrates the powerful capabilities of MATLAB and Simulink as a test bed for control algorithms, as long as a traffic simulation platform is provided. This paper takes advantage of the Discrete-Event Simulation framework introduced in SimEvents (MathWorks® 2016c) by MathWorks® (2016). This new framework offers users access to both graphical and textual modeling languages to create customized Discrete Event Systems (DES). Combined with the discrete-event/continuous-time hybrid simulation engine of the original SimEvents® (Clune, Mosterman, and Cassandras 2006), a single simulation model can include both discrete-event components implemented by SimEvents, and continuous-time components implemented by Simulink. This makes MATLAB and Simulink a highly attractive platform for traffic simulation.

This paper presents a new hybrid traffic simulation framework based on SimEvents in conjunction with MATLAB and Simulink. The traffic simulation framework offers access to both the physical components, for instance, vehicle motion dynamics, and the cyber components, which may involve different ICT approaches and control strategies. Different scenarios can be built based on infrastructure settings and random events. The simulation model consists of two parts: (1) the continuous part for vehicle simulation, for instance, vehicle status tracking; and (2) the discrete event-driven part, for instance, a change in driving behavior when traffic lights turn red. Hence, the traffic simulation framework is a *hybrid* dynamic model.

This paper is structured as follows. Section 2 introduces the key elements of the proposed hybrid traffic simulation framework for evaluating ICT approaches and control algorithms. Section 3 shows the implementation of the traffic simulation platform using SimeEvents in conjunction with MATLAB and Simulink. Section 4 demonstrates the effectiveness of the traffic simulation platform by adopting two different scenarios: (1) CAVs under decentralized optimal control as in (Zhang, Malikopoulos, and

Table 1: Infrastructure.

Infrastructure	Property	Function
Road segment (control zone)	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 & Dynamic
Property	ID, acceleration, speed, position, lane, destination, mpg, etc.
Motion dynamic	basic model, Kinematic model, Dynamic model, etc.
Control dynamic	optimal control, MPC, etc.
Fuel consumption dynamic	EV, Hybrid vehicles, Internal combustion engine vehicles

Cassandras 2016) without explicit traffic signaling; and (2) CAVs under traffic light control. Finally, Section 5 gives a conclusion and outlook on further research activities regarding this platform.

2 A HYBRID TRAFFIC SIMULATION FRAMEWORK

For ICT methods and control algorithms, a traffic simulator offers a test bed for analyzing vehicle behavior under various traffic scenarios, for instance, avoiding a potential rear-end collision, stopping before a red light, or following the preceding vehicle. To create a traffic simulation framework for vehicle behavior evaluation, a closed loop system consisting of the roadside infrastructure, vehicles, traffic management, and random events is necessary. The simulation framework should consist of two parts: the continuous (time-driven) part for vehicle simulation and the discrete part for event-driven system components. The system is designed to comprise entities, queues, servers, and storage, where an entity may represent a vehicle, a communication packet, or an event.

There are three basic physical elements in the hybrid traffic simulation framework: infrastructure, vehicles, and random events.

- **Infrastructure**. The infrastructure consists of roadside facilities that may enable communication and carry out traffic management.
- **Vehicles**. Vehicles in the presented framework, assumed to be CAVs, should possess the ability to communicate with the controller, the coordinator, and other vehicles in the network. CAVs can differ in driver models, for instance, motion dynamics, control mechanism, and fuel consumption dynamics.
- **Random Events**: Random events do not only refer to vehicle arrival, lane change, and turn, but they also include events that may cause safety issues. Generally, all such random events can be categorized into two classes, depending on whether they are caused by vehicles or by the infrastructure/environment Bettisworth et al. (2015).

Figure 1 depicts the architecture for the traffic simulation framework and shows how different physical elements are connected. Different system elements should be capable of communicating with each other and exchanging information. *Vehicles* (CAVs) can talk to other CAVs (V2V) or the *coordinator* (V2I). For the time being, it is assumed that only V2I communication is active, as V2V communication can be achieved through vehicle-to-infrastructure-to-vehicle (V2I2V) communication. An important element of the proposed framework is the inclusion of the communication delays in order to study the effects of latency

Table 3: Random events.

Vehicle-to-Infrastructure (V2I)	Vehicle-to-Vehicle (V2V)
Red Light Violation Warning	Emergency Electronic Brake Lights (EEBL)
Curve Speed Warning	Forward Collision Warning (FCW)
Stop Sign Gap Assist	Intersection Movement Assist (IMA)
Spot Weather Impact Warning	Left Turn Assist (LTA)
Reduced Speed/Work Zone Warning	Blind Spot/ Lane Change Warning (BSW/LCW)
Pedestrian in Signalized Crosswalk Warning	Do Not Pass Warning (DNPW)
	Vehicle Turning Right in Front of Bus Warning

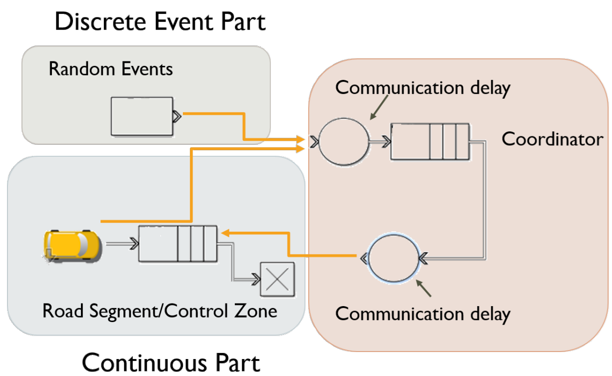

Figure 1: Architecture of the hybrid traffic simulation framework.

in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications, as low packet delays are necessary to implement the control algorithms employed by CAVs. In Fig. 1, an entity server is used to model the communication delays.

The output of *Random Events* into the coordinator indicates that the latter is aware of all the event-based information by means of sensing and communication. Once a certain random event occurs the coordinator will send or broadcast the information to the CAVs traveling on the road segments that might be affected, so that the CAVs can make appropriate decisions. For example, if a rear-end collision occurs near the merging zone of an intersection, the coordinator will receive this information through sensors, cameras, or the information sent by other vehicles. As part of safety considerations, the coordinator will broadcast the information and the CAVs traveling towards this area should decelerate or detour.

The *continuous* part in this framework includes all vehicle models, which should be evaluated in terms of the motion dynamics, control dynamics, and fuel consumption dynamics; for the *discrete event* part, random events (both vehicle-related and infrastructure/environment-related) are considered as well, since they can affect vehicle behavior. For instance, the CAVs may need information regarding other vehicles and/or the environment to determine the optimal driving decision. However, due to communication latency, a CAV is forced to cruise until the relevant information is received. When such an event occurs, the CAV will determine an optimal driving decision. Similarly, if the coordinator is aware of an upcoming storm, it will broadcast the information to the vehicles traveling inside the network and then the vehicles can make decisions accordingly. Another case may happen when the vehicle approaches to the intersection and the traffic lights turn red. Such an event forces a CAV to change the driving behavior to ensure the absence of any traffic law violations.

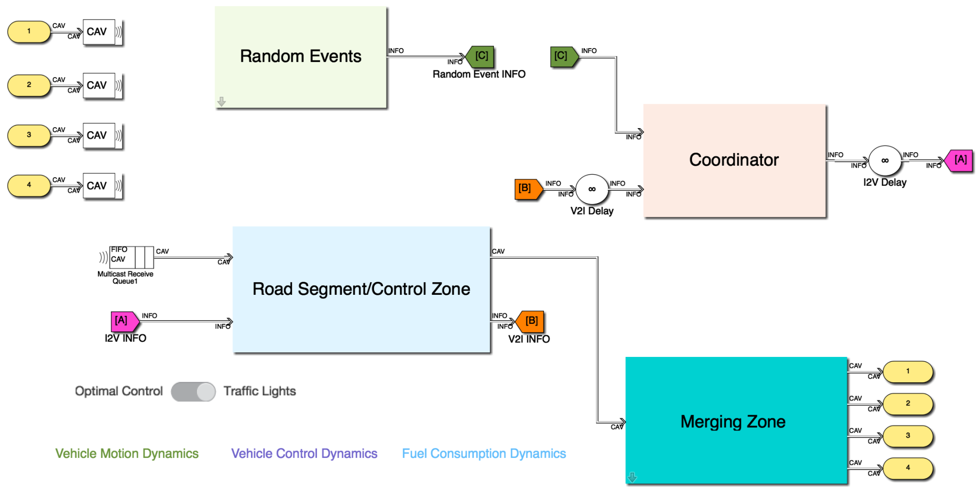

Figure 2: Simulink® model of an intersection with four road segments (input/outputs).

3 IMPLEMENTATION

The presented hybrid traffic simulation framework is built based on MATLAB and Simulink, which offers a platform for implementing controllers for rapid prototyping that is widely used in the automotive industry. The incorporation of SimEvents offers tools to build discrete event components. The model structure of a single intersection is shown in Fig. 2. Instead of simply using queues and servers to model the road segments, a *MATLAB Discrete-Event System* (Li, Mani, and Mosterman 2016) is created. Though the model structure does not follow the physical layout of the real intersection (as shown in Fig. 4), it provides flexibility as the MATLAB Discrete-Event System allows users to author an event-driven entity-flow system using object-oriented MATLAB (MathWorks® 2016a) and use it with Simulink as a block. The MATLAB Discrete-Event System features the following modeling and simulation capabilities:

- The MATLAB Discrete-Event System can contain multiple entity storage, while each storage can contain multiple SimEvents entities with a specific type, and is configured to sort entities in a certain order.
- An entity or a storage can schedule and execute multiple types of events such as creating and destroying an entity, or iterate over multiple entities in the storage.
- The MATLAB Discrete-Event System can take either entities or signals (data) as inputs or outputs and both built-in data types and structured/bus data types are supported.
- The MATLAB Discrete-Event System can be authored via a set of MATLAB methods. By implementing these methods users can define both structural properties (e.g., entity types and storage) and dynamic behavior of the system (e.g., event triggering conditions and actions taken when an event happens).

The MATLAB Discrete-Event System for "Road Segment" is shown in Fig. 3. An interactive window can also be created where the properties of the road segment/control zone can be defined without going into the actual code. For evaluation purposes, vehicle and driver models can be modified in terms of motion dynamics, control mechanisms, and fuel consumption dynamics. This is important as one of the future directions is the interaction between different vehicle types, for instance, CAVs and human drivers.

CAVs are modeled as entities coming from four directions in an intersection. They have several properties: ID, acceleration, speed, position, lane, destination, and so forth, and they are capable of communicating with the coordinator that is also built as a MATLAB Discrete-Event System. Packets are

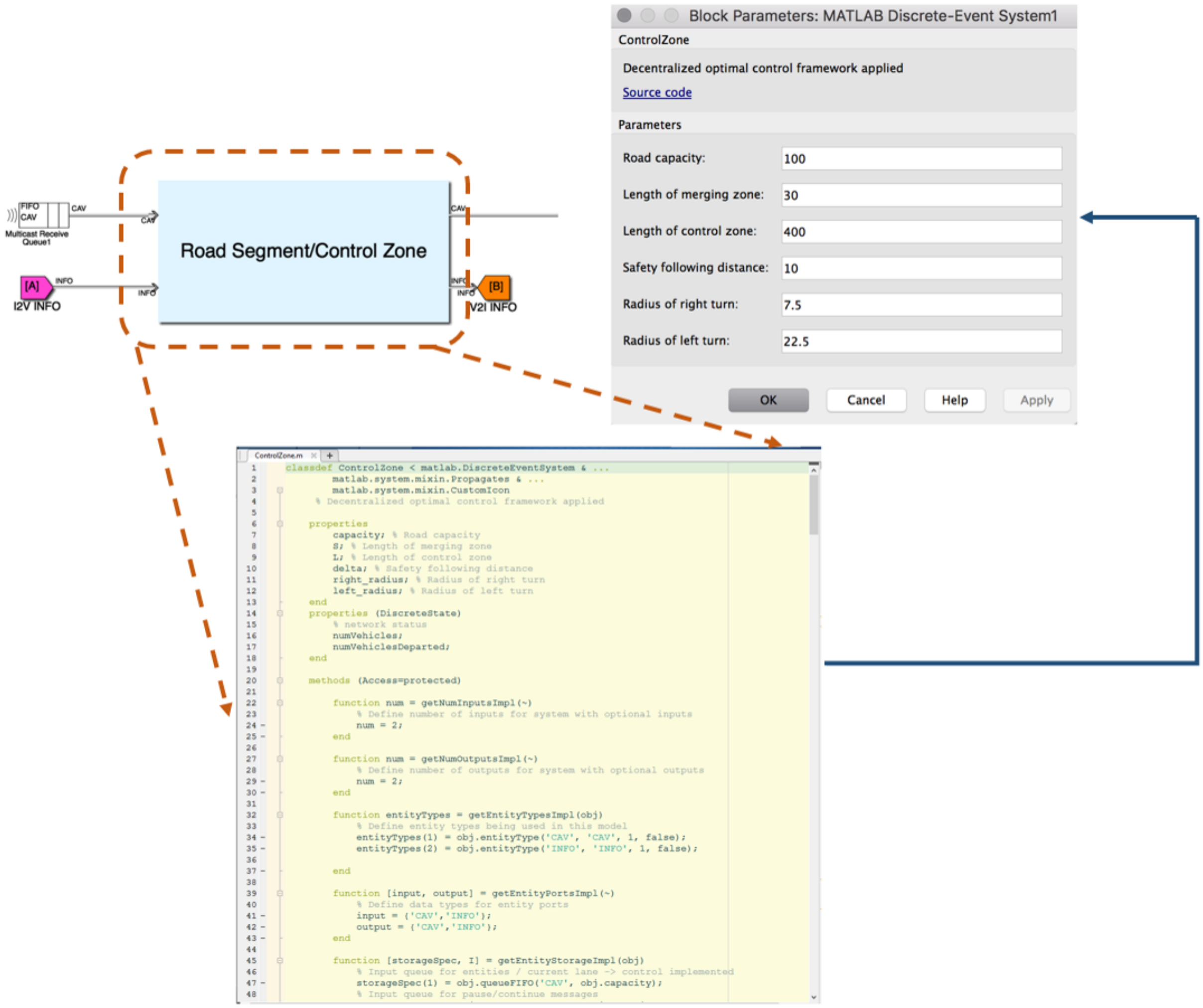

Figure 3: Simulink model with MATLAB-authored Discrete-Event System block.

modeled as entities as well, which may contain different kinds of information depending on the purpose of communication.

The random events part is simply modeled as an event generator, which can produce different scenarios for evaluating ICT approaches and control algorithms.

4 EVALUATION

The effectiveness of the traffic simulation platform is illustrated under two different scenarios, (1) the baseline scenario with traffic light control and (2) the decentralized optimal control framework without explicit traffic signaling, which is briefly discussed in the following subsection.

4.1 Decentralized Optimal Control for CAVs at Intersections

The decentralized optimal control framework is used for optimally controlling CAVs crossing an urban intersection without any explicit traffic signaling, so as to minimize energy consumption subject to a throughput maximization requirement.

The model introduced in (Zhang, Malikopoulos, and Cassandras 2016) is briefly reviewed. As shown in Fig. 4, the region at the center of each intersection, called *Merging Zone* (MZ) is the area of potential

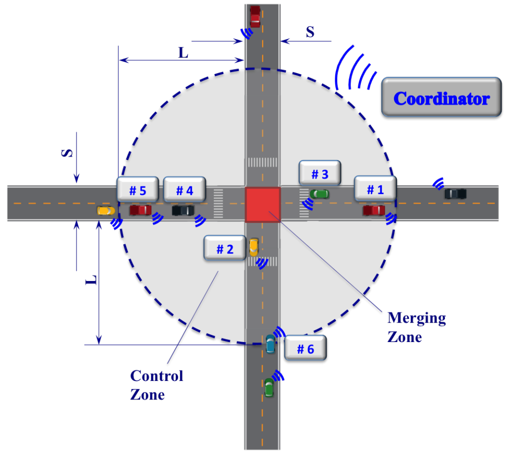

Figure 4: Intersection with Connected Automated Vehicles.

lateral CAV collision. Each intersection has a *Control Zone* (CZ) and a coordinator that can communicate with the CAVs traveling within it. Let $M_z(t) \in \mathbb{N}$ be the cumulative number of CAVs that have entered the CZ and formed a queue by time t, $z = 1,2$. The way the queue is formed is not restrictive. Here, a M/M/1 queueing system (Cassandras and Lafortune 2009) following first-in-first-out order is assumed. When a CAV reaches the CZ of intersection z, the coordinator assigns it an integer value $i = M_z(t) + 1$.

For simplicity, each CAV is assumed to be governed by second order dynamics

$$\dot{p}_i = v_i(t), \;\; p_i(t_i^0) = 0; \;\; \dot{v}_i = u_i(t), \; v_i(t_i^0) \text{ given} \tag{1}$$

where $p_i(t) \in \mathscr{P}_i$, $v_i(t) \in \mathscr{V}_i$, and $u_i(t) \in \mathscr{U}_i$ denote the position (i.e., travel distance since the entry of the CZ), speed and acceleration/deceleration (control input) of each CAV i. The sets $\mathscr{P}_i$, $\mathscr{V}_i$ and $\mathscr{U}_i$ are complete and totally bounded sets of $\mathbb{R}$. These dynamics are in force over an interval $[t_i^0, t_i^f]$, where t_i^0 and t_i^f are the times that the vehicle i enters the CZ and exits the MZ of intersection z, respectively.

To ensure that the control input and vehicle speed are within a given admissible range, the following constraints are imposed:

$$\begin{aligned} u_{i,min} \leq u_i(t) \leq u_{i,max}, \quad & \text{and} \\ 0 \leq v_{min} \leq v_i(t) \leq v_{max}, \quad & \forall t \in [t_i^0, t_i^m], \end{aligned} \tag{2}$$

where t_i^m is the time that the vehicle i enters the MZ. To ensure the absence of any rear-end collision throughout the CZ, the *rear-end safety* constraint is imposed

$$s_i(t) = p_k(t) - p_i(t) \geq \delta, \quad \forall t \in [t_i^0, t_i^m] \tag{3}$$

where δ is the *minimal safe distance* allowable and k is the CAV physically ahead of i.

The objective of each CAV is to derive an optimal acceleration/deceleration, in terms of fuel consumption, inside the CZ, that is,

$$\begin{aligned} & \min_{u_i \in U_i} \frac{1}{2} \int_{t_i^0}^{t_i^m} K_i \cdot u_i^2 \, dt \\ & \text{subject to} : (1), (2), t_i^m, \;\; p_i(t_i^0) = 0, \; p_i(t_i^m) = L, \\ & z = 1,2, \text{and given } t_i^0, \; v_i(t_i^0), \end{aligned} \tag{4}$$

where K_i is a factor to capture CAV diversity (for simplicity, $K_i = 1$ in the remainder of this paper). Note that this formulation does not include the safety constraint (3). In addition, safety constraints are imposed to avoid either rear-end collision or lateral collision inside the MZ.

An analytical solution of problem (4) may be obtained through a Hamiltonian analysis. Assuming that all constraints are satisfied upon entering the CZ and that they remain inactive throughout $[t_i^0, t_i^m]$, the optimal control input (acceleration/deceleration) over $t \in [t_i^0, t_i^m]$ is given by

$$u_i^*(t) = a_i t + b_i \tag{5}$$

where a_i and b_i are constants. Using (5) in the CAV dynamics (1), the optimal speed and position are obtained:

$$v_i^*(t) = \frac{1}{2} a_i t^2 + b_i t + c_i \tag{6}$$

$$p_i^*(t) = \frac{1}{6} a_i t^3 + \frac{1}{2} b_i t^2 + c_i t + d_i, \tag{7}$$

where c_i and d_i are constants of integration. The coefficients a_i, b_i, c_i, d_i can be obtained given initial and terminal conditions as follows:

$$\begin{bmatrix} \frac{1}{6}t^3 & \frac{1}{2}t^2 & t & 1 \\ \frac{1}{2}t^2 & t & 1 & 0 \\ \frac{1}{6}(t_i^m)^3 & \frac{1}{2}(t_i^m)^2 & t_i^m & 1 \\ -t_i^m & -1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} p_i(t) \\ v_i(t) \\ p_i(t_i^m) \\ 0 \end{bmatrix}$$

The conditions under which the feasible solutions exist are derived in (Zhang, Cassandras, and Malikopoulos 2017), where it is also shown how they can be enforced through an appropriately designed *Feasibility Enforcement Zone* that precedes the control zone. The complete analytical solution satisfying all safety constraints is presented in (Malikopoulos, Cassandras, and Zhang 2017). The optimal control framework is also extended to account for left and right turns under hard safety constraints in (Zhang, Malikopoulos, and Cassandras 2017).

4.2 Simulation and Experiments

A single intersection of two roads with four lanes is simulated, with each lane corresponding to a specific direction. The length of the MZ, S, is 30m and the length of the CZ, L, is 400m. The minimum safety following distance δ is 10m.

Regarding the information and communication approaches, simple information exchange is assumed between vehicles and the coordinator for the time being. Every time a CAV enters the network, it sends information to the coordinator indicating its arrival. Note that the vehicle arrival actually serves as a "random event", which must be taken care of by the coordinator. After a certain period of communication latency, which is modeled using servers, the coordinator will send information back and this information may be used by the CAV to make decisions. For example, a CAV will provide its current speed and destination to the coordinator, and the coordinator will return the speed limit.

To test the decentralized optimal control algorithm, a scenario without explicit traffic signaling is considered. One snapshot of the simulation is shown in Fig. 5, where the color represents the direction the vehicle comes from. The output control input and speed trajectories as shown in Fig. 6 are consistent with expectation, which shows it to be a useful tool to evaluate control algorithms.

In order to show that the traffic simulation framework is adaptable to different scenarios, the situation is considered where traffic light control is applied. One snapshot is shown in Fig. 7. Note that only green and red phases are considered, with each lasting for 30 seconds.

Based on the simulation platform, both the optimal control algorithm and the traffic light control work very well. In fact, the platform can easily be adapted to any kind of scenario. Since the modules in the

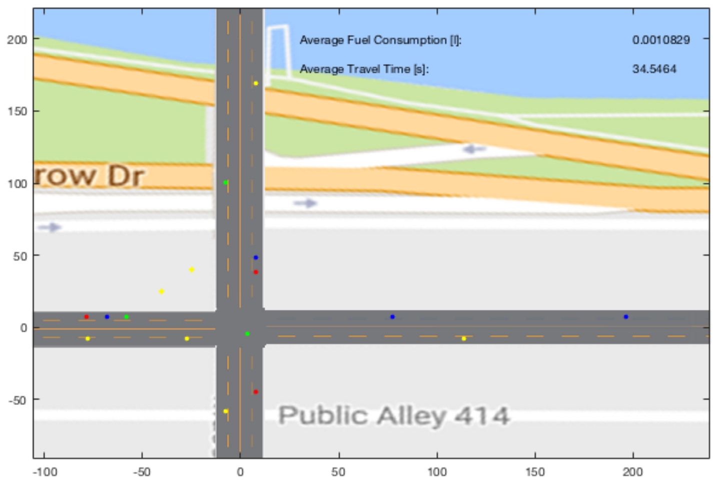

Figure 5: A snapshot of the scenario under decentralized optimal control framework.

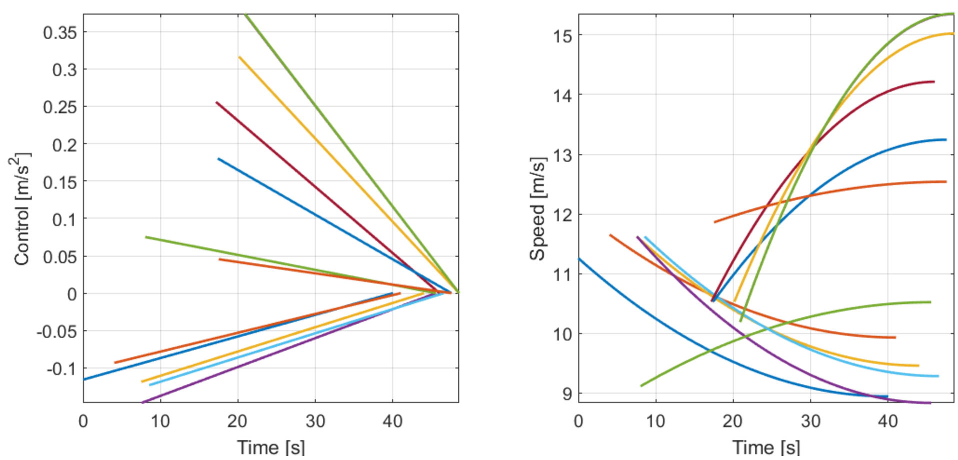

Figure 6: Control input (acceleration/deceleration) and speed trajectories of CAVs under decentralized optimal control framework.

framework are relatively independent, it is only necessary to modify associated modules and replace the previous control algorithm with a new one, as long as they share same motion dynamics.

5 CONCLUDING REMARKS AND FUTURE WORK

In this paper, a hybrid traffic simulation framework has been proposed with the purpose of evaluating different ICT approaches and control algorithms. The hybrid nature of SimEvents fits the goal in the sense that the traffic simulation does not only contains the continuous (time-driven) but also the discrete (event-driven) components. The effectiveness of the simulation framework was demonstrated using two different scenarios, where a decentralized optimal control algorithm and one based on traffic lights are applied, respectively.

Other advantages include scalability, that is, we can expand the system by simply adding more storage/queues. The nature of MATLAB provides full access to model details. For example, it is easy to

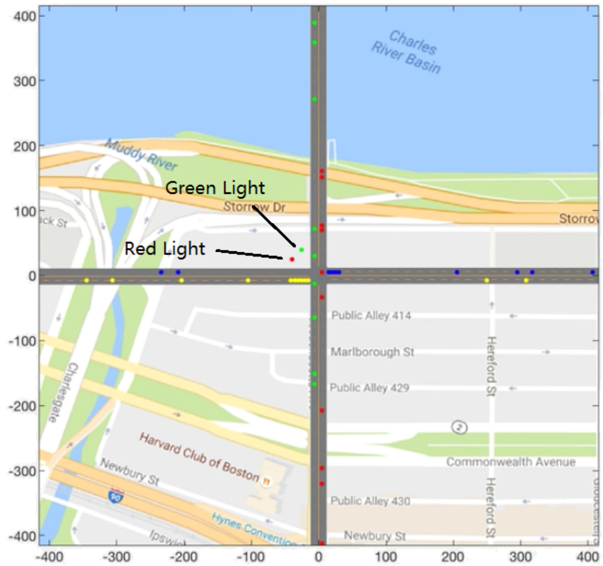

Figure 7: A snapshot of the scenario under traffic light control.

obtain and modify the acceleration of a CAV. It is convenient to manipulate the model elements such as specifying the road length and adding and/or deleting CAV properties.

Ongoing research includes the incorporation of the Dedicated Short Range Communication (DSRC) protocols and the investigation of random events and the associated vehicle reaction behavior. As CAVs may not be the only vehicle type traveling on the road, interactions with vehicles controlled by human drivers must also be considered. Ultimately, the framework should work for various traffic scenarios and accommodate vehicles with different control strategies, conditions, and preferences. Furthermore, the Dedicated Short Range Communication (DSRC) protocol should be incorporated, as it is the key technology for V2V safety communications.

REFERENCES

Benekohal, R., and J. Treiterer. 1988. “CARSIM: Car-Following Model for Simulation of Traffic in Normal and Stop-and-Go Conditions”. *Transportation Research Record* (1194).

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

Cameron, G. D., and G. I. Duncan. 1996. “PARAMICS—Parallel Microscopic Simulation of Road Traffic”. *The Journal of Supercomputing* 10 (1): 25–53.

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

Clune, M. I., P. J. Mosterman, and C. G. Cassandras. 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.

Kato, S., S. Tsugawa, K. Tokuda, T. Matsui, and H. Fujii. 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., G. Hertkorn, C. Rössel, and P. Wagner. 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.

Li, W., R. Mani, and P. J. Mosterman. 2016. "Extensible Discrete-Event Simulation Framework in Simevents". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. Huschka, S. Chick, J. Jimenez, P. Frazier, T. Roeder, R. Szechtman, and E. Zhou, 943–954. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Malikopoulos, A. A., C. G. Cassandras, and Y. Zhang. 2017. "A Decentralized Energy-Optimal Control Framework for Connected Automated Vehicles at Signal-Free Intersections". *Automatica*. (to appear).

MathWorks® 2016a. *MATLAB® Object-Oriented Programming (R2016b)*. MathWorks®, Natick, MA.

MathWorks® 2016b. *MATLAB®, Primer (R2016b)*. MathWorks®, Natick, MA.

MathWorks® 2016c. *SimEvents®, User's Guide (R2016b)*. MathWorks®, Natick, MA.

MathWorks® 2016d. *Simulink®, User's Guide (R2016b)*. MathWorks®, Natick, MA.

Shladover, S. E., C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown. 1991. "Automated Vehicle Control Developments in the PATH Program". *IEEE Transactions on Vehicular Technology* 40 (1): 114–130.

Versteegt, E., G. Klunder, and B. v. Arem. 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., C. G. Cassandras, and A. A. Malikopoulos. 2017. "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., A. A. Malikopoulos, and C. G. Cassandras. 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., A. A. Malikopoulos, and C. G. Cassandras. 2017. "Decentralized Optimal Control for Connected Automated Vehicles at Intersections Including Left and Right Turns". In *56th IEEE Conference on Decision and Control*. (to appear).

AUTHOR BIOGRAPHIES

YUE ZHANG received the B.S. degree in Electronics and Information Engineering from Huazhong University of Science and Technology, Wuhan, China in 2013. She is currently pursuing the Ph.D. degree with the Division of Systems Engineering and the Center for Information and Systems Engineering (CISE) at Boston University, working with Professor Christos G. Cassandras. Her research interests include optimization, optimal control of hybrid systems and big data analytics, with applications to intelligent transportation systems. She is also involved in Boston Street Bump project, working in collaboration with the city of Boston, to apply machine learning techniques on detecting and identifying fixable bumps on city streets. During the summer of 2015, she is working as a research intern with the National Transportation Research Center at the Oak Ridge National Laboratory, Knoxville, TN, USA. Her email address is joycez@bu.edu.

CHRISTOS G. CASSANDRAS received the B.S. degree from Yale University, New Haven, CT, USA, in 1977, the M.S.E.E. degree from Stanford University, Stanford, CA, USA, in 1978, and the M.S. and Ph.D. degrees from Harvard University, Cambridge, MA, USA, in 1979 and 1982, respectively. He was with ITP Boston, Inc., Cambridge, from 1982 to 1984, where he was involved in the design of automated manufacturing systems. From 1984 to 1996, he was a Faculty Member with the Department of Electrical and Computer Engineering, University of Massachusetts Amherst, Amherst, MA, USA. He is currently a Distinguished Professor of Engineering with Boston University, Brookline, MA, USA, the Head of the

Division of Systems Engineering, and a Professor of Electrical and Computer Engineering. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has authored over 380 refereed papers in these areas, and six books. Dr. Cassandras is a member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the IEE and a Fellow of the International Federation of Automatic Control (IFAC). He was a recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the 2006 Distinguished Member Award of the IEEE Control Systems Society, the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook), a 2011 prize and a 2014 prize for the IBM/IEEE Smarter Planet Challenge competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, a 1991 Lilly Fellowship, and a 2012 Kern Fellowship. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL from 1998 to 2009. He serves on several editorial boards and has been a Guest Editor for various journals. He was the President of the IEEE Control Systems Society in 2012. His email address is cgc@bu.edu.

WEI LI received his B.S. (1998) and M.S. (2001) degrees in Control Engineering from Tsinghua University, Beijing, China, and his Ph.D. degree (2005) in Systems Engineering from Boston University. From 2006 he has been working in software development at MathWorks, Inc. His current interests include discrete-event and hybrid simulation technologies, quantitative analysis, formal verification and optimization of discrete-event systems, embedded system architecture modeling, network simulation, and operations research. His email address is Wei.Li@mathworks.com .

PIETER J. MOSTERMAN is Chief Research Scientist and Director of the MathWorks Advanced Research & Technology Office in Natick, Massachusetts. He also holds an adjunct professor position at the School of Computer Science of McGill University. Prior to this, he was a research associate at the German Aerospace Center (DLR) in Oberpfaffenhofen. He earned his Ph.D. in Electrical and Computer Engineering from Vanderbilt University in Nashville, Tennessee, and his M.Sc. in Electrical Engineering from the University of Twente, the Netherlands. His primary research interests are in Computer Automated Multiparadigm Modeling (CAMPaM) with principal applications in design automation, training systems, and fault detection, isolation, and reconfiguration. In 2009, he received the Distinguished Service Award of *The Society for Modeling and Simulation International* (SCS) for his services as editor in chief of *SIMULATION: Transactions of SCS*. Dr. Mosterman also has been guest editor for special issues on CAMPaM of *SIMULATION, IEEE Transactions on Control Systems Technology*, and *ACM Transactions on Modeling and Computer Simulation*. His email address is pieter.mosterman@mathworks.com .