Co-simulation Framework for Control, Communication and Traffic for Vehicle Platoons

Amr Ibrahim*, Chetan Belagal Math*, Dip Goswami*, Twan Basten*‡, Hong Li†
*Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
†Car Infotainment & Driving Assistance, NXP Semiconductors, Eindhoven, The Netherlands
‡ ESI, TNO, Eindhoven, The Netherlands
Email:{a.ibrahim, c.belagal.math, d.goswami, a.a.basten}@tue.nl, hong.r.li@nxp.com

Abstract*—Vehicle platooning has gained attention for its potential to achieve an increased road capacity and safety, and a higher fuel efficiency. Member vehicles of a platoon wirelessly communicate complying with industrial standards such as IEEE 802.11p. By exchanging information with other members via wireless communication, a platoon member computes its desired acceleration which is then passed on to the engine control system via in-vehicle network to physically realize the acceleration. This leads to a multi-layer control scheme. The upper-layer is influenced by the behavior of 802.11p communication and network congestion due to transmissions by other vehicles in the traffic. The lower-layer engine control loop communicates over the fast and reliable in-vehicle networks (e.g., FlexRay, Ethernet). Design of the overall system therefore depends on (i) the characteristics of 802.11p-based communication (ii) the nature of the traffic (iii) the control algorithms running at the two layers. We present a co-simulation framework consisting of Matlab (for the multi-layer control algorithms), ns-3 (for the 802.11p network) and SUMO (for the traffic behavior). The framework can be used to validate different platooning setups. As an illustrative case study, we consider a multi-layer control strategy where the upper-layer uses Model Predictive Control (MPC) at a rate in compliance with 802.11p and the lower-layer uses state-feedback control at a higher sampling rate in line with in-vehicle networking capabilities. The control strategy is evaluated considering various realistic traffic and network congestion scenarios.

***Index Terms*—V2V communication, vehicle platoons, network simulator, traffic simulator, multi-layer control**

I. INTRODUCTION

Recent technologies in in-vehicle networks, on-board computers and Vehicle-to-Vehicle (V2V) communication allow cooperative driving between vehicles. Vehicle platooning is a promising cooperative driving technique. A platoon of vehicles is defined as a group of autonomous vehicles closely following each other and maintaining a safe inter-vehicle distance. In a platoon, vehicle members communicate with each other using V2V wireless communication complying with industrial standards such as IEEE 802.11p along with other on-board sensors such as radar and lidar. This allows the status of a vehicle (velocity, acceleration, position, etc.) to be made available to the following vehicles. With the help of such richer information sharing, vehicle platoons achieve an increase in road capacity, improved safety and better fuel efficiency [1].

One of the technologies that exists in almost every modern vehicle is Adaptive Cruise Control (ACC) [2]. With ACC, a vehicle automatically adjusts its speed in order to keep a safe distance with respect to its preceding vehicle by sensing its motion using on-board sensors. To ensure safety, in ACC, the *headway* time (i.e. the time needed by the follower vehicle to reach the position of the preceding vehicle) between vehicles should be greater than 1s. This improves road capacity and increases aerodynamics drag force (i.e reduces fuel efficiency). Cooperative Adaptive Cruise Control (CACC) extends ACC functionality by sharing more vehicle information via V2V wireless communication. This allows a faster response by the following vehicles and a significant reduction in headway time compared to ACC [1].

A layered control architecture is adopted in recent studies for control of platoon members both in ACC and CACC [3]. Generally, the acceleration setpoint of a platoon member is computed based on the information received from other platoon members via wireless communication. Next, the acceleration setpoint is realized using the engine control system within the vehicle. Such structure leads to a two-layered control architecture.

In the context of CACC, multiple communication standards are being considered by the automotive Original Equipment Manufacturers (OEMs) for V2V communication, in particular IEEE 802.11p [4] and 5G [5]. The 5G standard aims for ultra high reliability and low latency communication and several releases are being worked on. In this work, we consider IEEE 802.11p. In the IEEE 802.11p standard, Cooperative Awareness Messages (CAMs) or beacons are used for safety applications such as platooning. They are regularly broadcasted over the control channel between vehicles with status information such as acceleration, velocity, position, braking actions, road and intersection status [6]. A large number of vehicles communicating via 802.11p (e.g., urban traffic) causes channel overhead and network congestion leading to packet losses and delays. The level of network congestion depends on the vehicular density (in urban cities or highways), traffic behavior and the number of vehicles equipped with V2V communication devices.

Our contributions: We present a co-simulation framework that allows to consider the above three aspects in early design – network behavior, traffic behavior and layered control architecture. In particular, our framework brings together (Fig. 1):

- realistic network simulator ns-3 [7]. ns-3 is a discrete-event network simulator used to simulate V2V communication between platoon vehicles along with other vehicles to characterize different network behaviors. ns-3 implements the communication architecture for IEEE 802.11p.
- microscopic traffic simulator SUMO [8]. SUMO is designed for generating real driving behavior on highways or urban traffic and to provide a graphical user interface (GUI) to observe the motion of vehicles. It allows to consider different road types, traffic lights, intersections, slopes, additional vehicles (e.g. cars, trucks, bicycles, trams, etc.) and pedestrians. By increasing the number of vehicles in SUMO and by simulating them in ns-3, it is possible to create congested networks

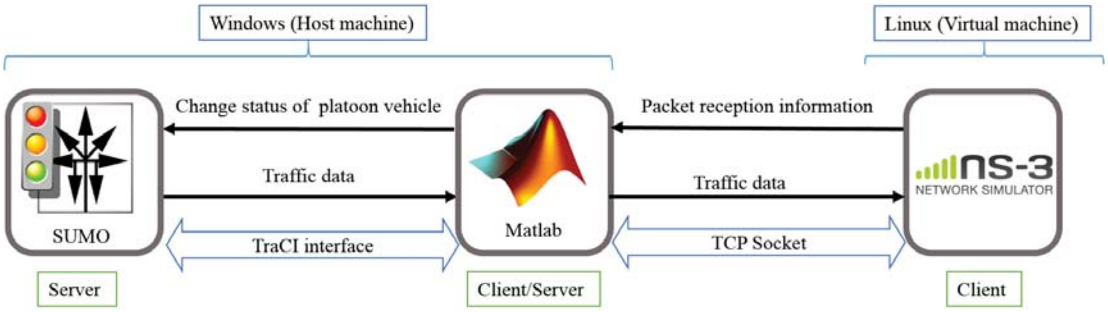


Fig. 1: Proposed co-simulation framework with ns-3, SUMO and Matlab and interaction.

causing packet losses and delays.

- Matlab for design, analysis and evaluation of various control architectures and strategies in presence of packet losses and delays caused by the network congestion and the traffic density.

As a case study, we present a two-layer platoon control architecture. The upper-layer control is responsible for generating the acceleration setpoint ensuring safety and efficiency. We consider Model Predictive Control (MPC) for its ability to handle constraints and its predictive behavior which is relevant to deal with packet losses. The upper-layer is updated at 10Hz in compliance with the 802.11p recommendation considering low channel load. The lower-layer control is responsible to reach the acceleration setpoint as soon as possible. We use state-feedback control at a higher rate (e.g., 50Hz, 20Hz) which is supported by in-vehicle networks. We have considered cases with various vehicular densities and 5 platoon vehicles with all vehicles communicating over 802.11p using CAM messages.

This paper is organized as follows. Related work is discussed in Section II. Section III presents our co-simulation framework. We describe communication, control and traffic behavior in Sections IV, V, VI, respectively. Simulation results of our case study are discussed in Section VII. Section VIII concludes our work.

II. RELATED WORK

Co-simulation frameworks for platoons have been presented in recent literature. In [9], a co-simulation environment has been implemented using VISSIM (for traffic simulation), ns-3 and Matlab. In this framework, the control architecture is simplified with constant desired gap and mainly focuses on upper-layer control (using a simplified PI controller). Moreover, it was assumed that only platoon members communicate over the 802.11p standard. A co-simulation framework based on OMNET++ (for network simulation), SUMO and SIMULINK (for the control application) is reported in [10]. The focus has been on string stability with different packet reception rates and headway times. Packet loss is artificially accomplished by a module which drops the received beacons by using a loss probability with uniform distribution. The controller uses the previously received and stored acceleration values in case of packet losses. Our proposed framework provides a template for multi-layer control architecture with both realistic traffic scenarios and network congestion behavior.

Plexe [11] is an integrated simulator framework combining OMNET++ and SUMO that extends Veins (vehicular network simulator framework [12]). It focuses on joining maneuvers of platoon vehicles and it adds models to SUMO e.g. cruise control (CC), ACC, CACC. Another framework, VENTOS [13] also combines OMNET++ and SUMO and implements platoon maneuvers e.g. merge, split and lane-change. How to extend the aforementioned frameworks to implement modern controllers (e.g. MPC) is not clear.

III. CO-SIMULATION FRAMEWORK

This section presents the proposed co-simulation framework consisting of ns-3, SUMO and Matlab as shown in Fig. 1. Matlab and SUMO run on the Windows operating system. ns-3 runs on a Linux virtual machine. Matlab is the interface between ns-3 and SUMO. A TCP-based protocol is used to establish interaction between the three tools for reliable data exchange. The TCP connection is established between SUMO and Matlab using TraCI (Traffic Control Interface). Socket programming is used for the TCP connection between Matlab and ns-3. SUMO has two types of vehicles – platoon vehicles and additional regular vehicles. SUMO also has built-in traffic control algorithms imitating human driver behavior on highways which are used to control the additional regular vehicles whereas the platoon vehicles are controlled via the control algorithm running in Matlab. Control of all vehicles (including platoon vehicles) is first generated by SUMO (required for initialization) and then state information of the platoon vehicles is overwritten by TraCI functions using the control algorithm running in Matlab. Our framework assumes that the communicating vehicles to be synchronous since synchronization can be achieved with acceptable accuracy via GPS with the state-of-the-art 802.11p devices e.g. MK5 On-Board Unit (OBU) from Cohda Wireless [14].

Interaction between the three tools happens every 100ms, which is derived from the recommended transmission rate for CAMs in 802.11p and it is based on the client-server model. SUMO waits to receive the new information of the platoon vehicles from Matlab; then it simulates and updates all traffic members. Next, the states of all traffic members are sent to ns-3 through Matlab which simulates V2V communication between vehicles; ns-3 finds and updates Matlab with the packet reception information between platoon vehicles. Then Matlab runs the control algorithm based on the received information from ns-3. Overall interaction between the three tools is as follows.

Tool interactions at the kth iteration:

1) SUMO, as a server, allows Matlab to modify and retrieve states of simulated vehicles. It receives the states of the platoon vehicles computed in Matlab in the $(k-1)^{th}$ iteration and generates new states of all vehicles. Next, SUMO overwrites the states of the platoon vehicles by the information received from Matlab and it displays all the vehicles (i.e., position and velocity) in the SUMO GUI.
2) As a client, Matlab requests the states of the vehicles from SUMO. As an interface between SUMO and ns-3, Matlab forward the received vehicle information to ns-3.

Carrier sensing threshold	-85 dBm	
Transmit power	24 dBm	
(Large scale fading) Dual slope model [19]	Path loss exponent 1 (until 80 m)	1.9
	Path loss exponent 2 (after 80 m)	3.8
(Small scale fading) Nakagami m model [19]	Distance bin in meters	m
	0-50	3
	51-150	1.5
	>150	1
Payload size	300 Bytes	

TABLE I: Communication parameters

3) ns-3 simulates packet broadcast of every vehicle using a uniform velocity mobility model. That is, ns-3 simulates constant speed motion of every vehicle.
4) ns-3 finds packet reception details (packet losses and delay) of each platoon vehicle. ns-3 stores this information in a lookup table that is accessible to Matlab.
5) Based on the packet reception information in the lookup table, Matlab runs the control algorithm for each platoon vehicle. The computed state information is sent to SUMO to overwrite the states of the platoon members.

The process repeats every 100ms.

IV. V2V COMMUNICATION: IEEE 802.11P

In Europe, a 50MHz spectrum in the 5.9GHz range is licensed to Cooperative Intelligent Transport Systems (C-ITS) to be used exclusively for V2V and V2I (Vehicle-to-Infrastructure) communications. This spectrum is divided into four service channels (SCH) and one control channel (CCH) each of 10MHz bandwidth. CCH is used for safety critical applications whereas SCH is used for infotainment or commercial applications [4]. In our case study, we consider CCH for exchanging CAMs among platoon members. Each vehicle broadcasts CAMs regularly over the control channel of 802.11p for safety applications. These messages contain information on the current status of the vehicle (e.g., acceleration, speed, position, braking action) and other environmental information, e.g., intersection and road status [6]. According to the European Telecommunications Standard Institutes ETSI [15], each 802.11p node sends these beacons regularly at 10Hz, if the communication channel load is low. If the channel load is higher, the beacon message transmission rate may be lower and is controlled by a Decentralized Congestion Control (DCC) algorithm [16]. The current platooning control trials use a dedicated service channel to share message at a higher rate (e.g., > 25Hz) since safety regulations do not allow to use a lower frequency [17]. That requires using a separate transceiver to exchange platoon control messages over a special service channel only for platoons and it experiences less congestion and communication interference since it puts less restrictions on sending rates, message types, MAC protocol [17].

In our case study, we consider communication parameters reported in [18] and they are listed in Table I. The channel model parameters are obtained from the highway scenario specification in the ETSI standard [19].

V. CONTROL STRATEGY: MULTI-LAYER AND MULTI-RATE

As already explained, platoon control often uses a layered control architecture. The upper-layer role is to receive information from the other vehicles and from the environment. Based on this information, the upper-layer control computes the desired acceleration ($a_{des,i}$) of the i^{th} vehicle and provides it to the lower-layer controller (see Fig. 2). In our case study, we use the Predecessor-Follower (PF) topology in which, the upper-layer controller of the i^{th} vehicle receives information from its preceding vehicle i.e., the $(i-1)^{th}$ vehicle. It computes $a_{des,i}$ based on position (q_{i-1}), velocity (v_{i-1}), acceleration (a_{i-1}) of the $(i-1)^{th}$ vehicle received over the 802.11p wireless network and q_i, v_i, a_i of the i^{th} vehicle. Then, $a_{des,i}$ is provided to the engine control system (i.e. the lower-layer controller) via the in-vehicle network. As already stated, the upper-layer controller runs at 10Hz as per 802.11p standard (assuming a channel load lower than 70% [20]).

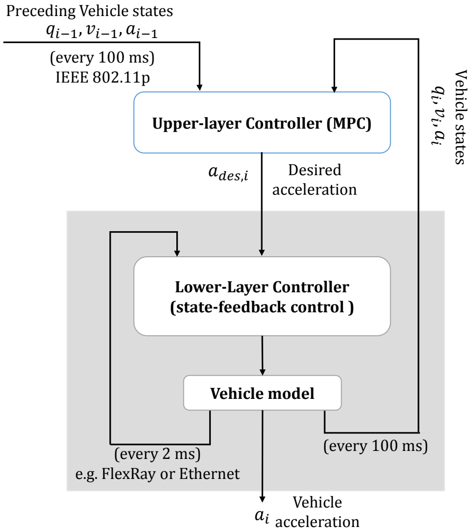


Fig. 2: Multi-layer multi-rate control structure of vehicle i

The lower-layer controller is used as a tracking controller. It computes the required motor duty cycle in % which then changes the throttle angle position so that the energy generated from the engine is enough to move the vehicle forward and reach the desired acceleration set by the upper-layer controller within a short interval. The lower-layer controller receives the desired acceleration over fast and reliable in-vehicle networks (e.g. FlexRay or Ethernet) [21]. The lower layer controller may be implemented with a shorter sampling period (e.g., 2ms, 5ms) considering a typical automotive architecture [22].

A. Vehicle and Platoon Model

The vehicle model of the i^{th} vehicle is given by [23]:

$$\dot{x}_{ll,i} = A_{ll,i}x_{ll,i} + B_{ll,i}u_{ll,i}, \tag{1}$$

where $u_{ll,i}$ is the lower-layer controller (motor duty cycle in %) of the i^{th} vehicle. $A_{ll,i}$ and $B_{ll,i}$ are the state and input matrices of vehicle i, respectively. System states are $x_{ll,i} = [a_i \quad \dot{a}_i]^T$ where a_i and $\dot{a}_i$ denote acceleration and rate of change of acceleration of the i^{th} vehicle, respectively.

Overall platoon behavior depends on the inter-vehicle longitudinal dynamics which is defined by introducing two state variables: position error $\Delta d = d - d_{des}$ (the error between the actual and the desired inter-vehicle gap) and velocity error $\Delta v = v_{i-1} - v_i$ (the difference in velocity between i^{th} and $(i-1)^{th}$ vehicles). The desired and actual inter-vehicle gap are defined as follows, $d_{des} = d_0 + \tau_h v_i$, $d = q_{i-1} - q_i - L_i$, where d_0 is the gap between vehicles at standstill, $\tau_h = 0.2$m is the constant headway time (the time vehicle i needs to reach the position of vehicle $i-1$ when $d_0 = 0$) and L_i is the length of vehicle i. In our case study, we consider $d_0 = 1$m and $L_i = 4$m. Differentiating Δd and Δv, the inter-vehicle longitudinal dynamics can be written as,

$$\dot{x}_{ld,i} = A_{ld,i} x_{ld,i} + H_{ld,i} x_{ll,i} + G_{ld,i} V_i, \tag{2}$$

where $x_{ld,i} = \begin{bmatrix} \Delta d \\ \Delta v \end{bmatrix}$, $A_{ld,i} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$, $H_{ld,i} = \begin{bmatrix} -\tau_h & 0 \\ -1 & 0 \end{bmatrix}$, $G_{ld,i} = [0 \quad 1]^T$, and $V_i = a_{i-1}$.

Combining the vehicle model in Eq. (1) with the inter-vehicle longitudinal dynamics in Eq. (2), we obtain the platoon model,

$$\dot{x}_{ul,i} = A_{ul,i} x_{ul,i} + B_{ll,i} u_{ll,i} + G_{ld,i} V_i, \tag{3}$$

where $A_{ul,i} = \begin{bmatrix} A_{ll,i} & 0 \\ H_{ld,i} & A_{ld,i} \end{bmatrix}$, $x_{ul,i} = [a_i \quad \dot{a}_i \quad \Delta d \quad \Delta v]^T$.

B. Lower-Layer Controller

The lower-layer controller deals with vehicle model Eq. (1) to achieve the desired acceleration. Discretizing Eq. (1) with lower-layer sampling period h_{ll}, we obtain,

$$x_{ll,i}(k+1) = \Phi_{ll,i} x_{ll,i}(k) + \Gamma_{ll,i} u_{ll,i}(k), \tag{4}$$

where $x_{ll,i}(k) = [a_i(k) \quad \delta a_i(k)]^T$ and $a_i(k)$, $\delta a_i(k)$ are the discretized acceleration and rate of change of acceleration at time step k of the i^{th} vehicle. $\Phi_{ll,i}$, $\Gamma_{ll,i}$ represent the discretized system matrices with period h_{ll}. The choice of h_{ll} is driven by the sampling periods supported by the common automotive operating systems such as OSEK [24]. For example, OSEK supports periods of 2, 3, 5, 10, ... ms. A shorter h_{ll} requires a higher resource usage (in terms of communication and computation) of the in-vehicle electrical and electronic architecture. We consider a sampling period of 2ms in our case study. We consider a state feedback control law in the lower-layer controller of the following form,

$$u_{ll,i}(k) = K_i x_{ll,i}(k) + F_i a_{des,i}, \tag{5}$$

where $a_{des,i}$, K_i, F_i are the desired acceleration, feedback gain and feedforward gain of vehicle i, respectively. By substituting Eq. (5) into Eq. (4) we obtain the closed loop system. Gains K_i, F_i have to be designed such that the discrete-time closed-loop system is stable and reaches $a_{des,i}$.

C. Upper-Layer Controller

The upper-layer controller computes the desired acceleration considering the platoon model in Eq. (3). Overall system behavior depends on the relation between the lower-layer sampling period h_{ll} and the upper-layer sampling period h_{ul}. Generally, $h_{ul} \gg h_{ll}$ and h_{ul} is an integer multiple of h_{ll}. We illustrate the design considering $h_{ul} = 100$ms and $h_{ll} = 2$ms. To this end, we first discretize Eq. (3) at sampling period h_{ll} and substitute $u_{ll,i}(k)$ as per Eq. (5):

$$x_{ul,i}(k+1) = \Phi_{ul,i} x_{ul,i}(k) + \Gamma_{ul,i} u_i(k) + \Psi_{ld,i} V_i(k), \tag{6}$$

where $u_i(k) = a_{des,i}(k)$ and $V_i(k) = a_{i-1}(k)$.

Since, the lower-layer control loop executes 50 times within one upper-layer sampling period, we unroll the loop (i.e. Eq.(6)) 50 times to obtain the upper-layer dynamics. In other words, $x_{ul,i}(k+50)$ is found by recursively solving $x_{ul,i}(k+j)$ for $j = 1, ..., 50$ leading to,

$$x_i(k+1) = \alpha x_i(k) + \beta u_i(k) + \gamma V_i(k), \tag{7}$$

where, $x_i(k+1) := x_{ul,i}(k+50)$, $x_i(k) := x_{ul,i}(k)$, $\alpha := (\Phi_{ul,i})^{50}$, $\beta := \big((\Phi_{ul,i})^{49} + (\Phi_{ul,i})^{48} + .. + \Phi_{ul,i} + I\big)\Gamma_{ul,i}$, $\gamma := \big((\Phi_{ul,i})^{49} + (\Phi_{ul,i})^{48} + .. + \Phi_{ul,i} + I\big)\Psi_{ld,i}$.

Scenario1	600 vehicles, Low congestion, channel load <70%	Average PRR %	83.41
		Average delay(ms)	6.69
Scenario2	800 vehicles, medium congestion, channel load =70%	Average PRR %	67.33
		Average delay(ms)	11.44
Scenario3	1000 vehicles, high congestion, channel load >70%	Average PRR %	51.43
		Average delay(ms)	20.15

TABLE II: Average Packet Reception Ratio (PRR) and average delay in different traffic scenarios

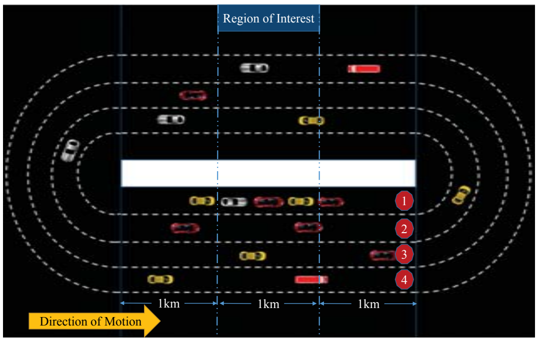


Fig. 3: SUMO traffic scenario

MPC [25] is used in our case study in the upper-layer controller for its ability to handle constraints on inputs and states and satisfy specific objectives e.g. (i) minimizing the gap between the vehicles to achieve a desired and safe inter-vehicle distance, (ii) tracking the speed and acceleration profiles of the preceding vehicle, (iii) minimizing sudden changes in acceleration to maintain passenger comfort. MPC can also deal with packet losses in case of congested network due to its predictive behavior.

VI. TRAFFIC SCENARIOS

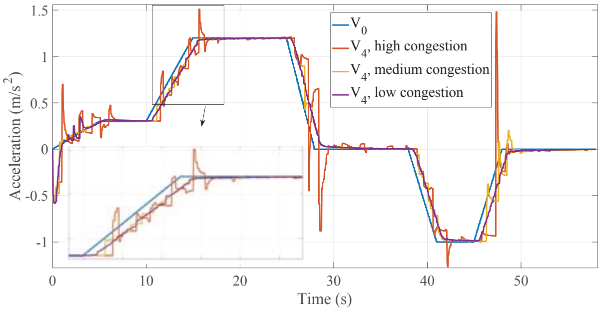
A realistic highway scenario is simulated in SUMO with a road section of 3km length and 4 lanes in each direction where lane 1 is dedicated to platoon vehicles. Platoon vehicles have a length of 4m, the length of additional traffic vehicles is 3m and their speed and acceleration ranges from 0 to 30m/s and from -2 to 3m/s^2, respectively. To eliminate the effects of the network boundaries, we restrict our performance evaluation to 1km in the 3km road (see Fig. 3). All vehicles are communicating via 802.11p and by changing traffic density, we obtain different network behavior. Three traffic scenarios are considered in our case study where the number of platoon vehicles is fixed in each scenario to 5 vehicles. With 600, 800, 1000 V2V enabled vehicles, we represent low, medium, high levels of congestion with channel loads of <70%, ~= 70%, >70%, respectively.

VII. RESULTS AND DISCUSSIONS

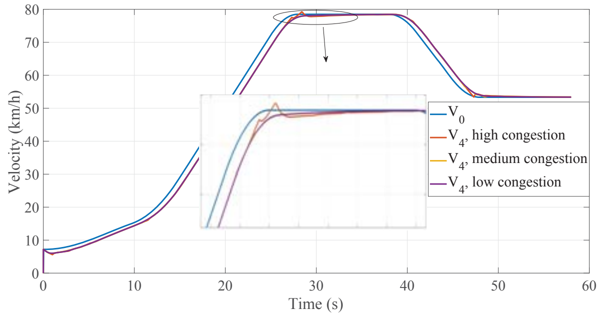
The behavior of the control algorithm depends on the number of packets received at the destination vehicle and the time needed for the packets to be received. In our case study it is assumed that packets are received in event order i.e. no packet interleaving. Packet Reception Ratio (PRR) is defined as the percentage of packets received by the destination vehicle from the transmissions by the source vehicle. Transmission time of a message is the time between the moment that a vehicle wants to send a message and the moment that a message sending is completed; the delay in 802.11p includes: message duration/length < 5ms, and several 10ms of MAC layer delay depending on vehicle density. We considered 100ms delay as the worst case delay of information and the controller was designed based on this consideration.

Average PRR and delay for the wireless link between any two consecutive platoon vehicles are computed and reported in Table II for different scenarios. Table II shows that 83.41% of the packets are received for Scenario1; the average delay = 6.69ms. Scenario2 shows 67.33% PRR and average delay 11.44ms. Average delay might be ignored for low and medium congestion in the control design. Scenario3 shows 50% of lost packets and 20.15ms average delay. That huge delay should be considered and transmission rate must be lowered (i.e. < 10Hz). Results from Table II are nondeterministic i.e. by running the simulation different times we could obtain different results. That is due to: (i) other traffic members controlled by SUMO to mimic human driving behavior (ii) ns-3 finding of different PRR of platoon members due to new positions and velocities of traffic members.

Controller performance under different scenarios is depicted in Fig. 4 and Fig. 5. Fig. 4a shows the given acceleration profile of the leader vehicle (V_0). It also shows that the 5^{th} platoon vehicle (V_4) gently tracks the leader profile in Scenario1. Small fluctuations happen for Scenario2 and fluctuation increases for Scenario3 due to significant packet losses. Fig. 4b shows velocity tracking of V_4 to V_0; our controller guarantees string stable [1] platooning (attenuation of disturbances) except for Scenario3 where V_4 velocity grows beyond V_0 at some points. Fig. 5 shows position error is bounded within 0.5m between V_0 and V_1. Thus vehicles can be driven very close to each other, minimizing air drag while maintaining safety.



(a) Acceleration



(b) Velocity

Fig. 4: Acceleration and velocity profiles of the 5^{th} platoon vehicle (V_4) in different congestion scenarios compared to the platoon leader (V_0)

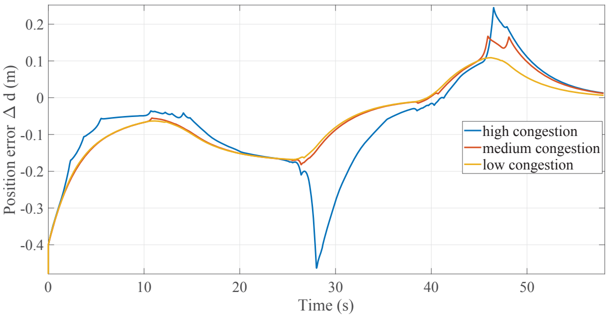


Fig. 5: Position error Δd between the first two platoon vehicles (V_0 & V_1)

VIII. CONCLUSIONS

We presented a co-simulation framework for control, communication and traffic behavior for vehicle platoon applications. The framework provides a design and testing template for all three relevant components. We presented a case study and demonstrated the analysis of various traffic scenarios and their impact on the overall platoon performance. This framework can be used for development, testing and validation of both platoon control and communication in different traffic scenarios.

IX. ACKNOWLEDGMENT

This research was partially funded by the European Union's Horizon 2020 Framework Programme for Research and Innovation under grant agreement no 674875 (oCPS Marie Curie Network).

REFERENCES

[1] Ploeg, Jeroen, et al. "Design and experimental evaluation of cooperative adaptive cruise control." ITSC, 14th International IEEE Conference, 2011.
[2] Marsden, Greg, Mike McDonald, and Mark Brackstone. "Towards an understanding of adaptive cruise control." Transportation Research Part C: Emerging Technologies 9.1 (2001): 33-51.
[3] Li, Shengbo, et al. "Model predictive multi-objective vehicular adaptive cruise control." IEEE Transactions on Control Systems Technology 19.3 , 2011.
[4] Jiang, Daniel, and Luca Delgrossi. "IEEE 802.11 p: Towards an international standard for wireless access in vehicular environments." VTC Spring, IEEE, 2008.
[5] Wang, Cheng-Xiang, et al. "Cellular architecture and key technologies for 5G wireless communication networks." IEEE Communications Magazine 52.2, 2014.
[6] ETSI TR 102 638, "Intelligent transport systems (ITS); vehicular communications; basic set of applications; definitions," June 2009.
[7] https://www.nsnam.org/
[8] Krajzewicz, Daniel, et al. "Recent development and applications of SUMO-Simulation of Urban MObility." International Journal On Advances in Systems and Measurements 5.3&4, 2012.
[9] Choudhury, Apratim, et al. "An integrated V2X simulator with applications in vehicle platooning." ITSC, IEEE 19th International Conference on, 2016.
[10] Lei, Chenxi, et al. "Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++." EURASIP Journal, 2012.
[11] Segata, Michele, et al. "Plexe: A platooning extension for Veins." VNC, IEEE, 2014.
[12] Sommer, Christoph, Reinhard German, and Falko Dressler. "Bidirectionally coupled network and road traffic simulation for improved IVC analysis." IEEE Transactions on Mobile Computing 10.1, 2011.
[13] Amoozadeh, Mani, et al. "Platoon management with cooperative adaptive cruise control enabled by VANET." Vehicular communications 2.2 (2015): 110-123.
[14] Cohda Wireless Pty Ltd, MK5 V2X On Board Unit.
[15] http://www.etsi.org/
[16] Wischhof, Lars, and Hermann Rohling. "Congestion control in vehicular ad hoc networks." Vehicular Electronics and Safety, IEEE, 2005.
[17] Böhm, Annette, Magnus Jonsson, and Elisabeth Uhlemann. "Performance evaluation of a platooning application using the IEEE 802.11 p MAC on a control channel vs. a centralized real-time MAC on a service channel." (2013).
[18] "Cross Layer DCC Management Entity for operation in the ITS G5A and ITS G5B medium; Validation set-up and results", ETSI ITS, Tech. Rep., 2015.
[19] "STDMA Recommended Parameters and Settings for Cooperative ITS; Access Layer Part." ETSI ITS, Tech. Rep., 2012.
[20] Fallah, Yaser P., et al. "Analysis of information dissemination in vehicular ad-hoc networks with application to cooperative vehicle safety systems." IEEE Transactions on Vehicular Technology 60.1, 2011.
[21] Makowitz, Rainer, and Christopher Temple. "FlexRay-a communication network for automotive control systems." IEEE International Workshop on Factory Communication Systems, 2006.
[22] Goswami, Dip, et al. "Multirate controller design for resource-and schedule-constrained automotive ECUs." DATE, IEEE, 2013.
[23] Ulsoy, A. Galip, Huei Peng, and Melih Çakmakci. Automotive control systems. Cambridge University Press, 2012.
[24] OSEK Consortium, OSEK/VDX operating system, version 2.2. 3, February 2005.
[25] Camacho, Eduardo F., and Carlos Bordons Alba. Model predictive control. Springer Science & Business Media, 2013.