Integrated simulation platform for conventional, connected and automated driving: A design from cyber-physical systems perspective

Dongyao Jia[a,*], Jie Sun[a], Anshuman Sharma[a], Zuduo Zheng[a,*], Bingyi Liu[b]

[a]*School of Civil Engineering, University of Queensland, Brisbane, Australia*
[b]*School of computing, Wuhan University of Technology, Wuhan, China*

Abstract

A comprehensive assessment of connected and automated driving is imperative before its large-scale deployment in reality, which can be economically and effectively implemented via a credible simulation platform. Nonetheless, the key components of traffic dynamics, vehicle modeling, and traffic environment are oversimplified in existing simulators. Current traffic simulators normally simplify the function of connected and autonomous vehicles by proposing incremental improvements to the conventional traffic flow modeling methods, which cannot reflect the characteristics of the realistic connected and autonomous vehicles. On the other hand, typical autonomous vehicle simulators only focus on individual function verification in some specific traffic scenarios, omitting the network-level evaluation by integrating both large-scale traffic networks and vehicle-to-anything (V2X) communication. This paper designs a comprehensive simulation platform for conventional, connected and automated driving from a transportation cyber-physical system perspective, which tightly combines the core components of V2X communication, traffic networks, and autonomous/conventional vehicle model. Specifically, three popular open-source simulators SUMO, Omnet++, and Webots are integrated and connected via the traffic control interface, and the whole simulation platform will be deployed in a Client/Server model. As the demonstration, two typical applications, traffic flow optimization and vehicle eco-driving, are implemented in the simulation platform. The proposed platform provides an ideal and credible testbed to explore the potential social/economic impact of connected and automated driving from the individual level to the large-scale network level.

Keywords:
Connected and automated driving, simulation platform, cyber-physical system, mixed traffic flow, V2X communication

1. Introduction

The globe is facing a number of major challenges in the field of transportation, with the need to improve road safety, mitigate traffic congestion, and reduce vehicle emissions. Connected and automated driving is regarded as a promising solution to these transport problems (Fagnant and Kockelman, 2015). For example, the California PATH program (Misener and Shladover, 2006) has demonstrated the efficiency of automated platoon driving in improving traffic throughput, and the FP7 project Drive C2X (Stahlmann et al., 2011) has shown the safety and efficiency benefits brought by cooperative connected systems. Specifically, the connected and autonomous vehicles (CAVs) (Hussain and Zeadally, 2018) integrated with ubiquitous sensing, seamless connection, and artificial intelligence (AI) are paving the way towards the higher level of transport automation, and accordingly, fundamentally transforming the modern transportation.

*Corresponding Authors
Email addresses: jiady@163.com (Dongyao Jia), jie.sun@uq.edu.au (Jie Sun), anshuman.sharma@uq.edu.au (Anshuman Sharma), zuduo.zheng@uq.edu.au (Zuduo Zheng), byliu@whut.edu.cn (Bingyi Liu)

Given the fact that a mixed traffic will appear soon on roads and last for a predictable long time where both CAVs and conventional vehicles, i.e. human-driven vehicles (HDVs), coexist, the complicated interactions between CAVs and conventional vehicles and the wide AI-based applications in connected and automated driving could bring some uncertainties and risks on traffic safety and efficiency. For example, Google's self-driving car was rear-ended by a conventional vehicle while stationary on a public road (Google, 2015). Therefore, a comprehensive assessment of connected and automated driving is essential before its large-scale deployment in reality, which can be economically and effectively implemented via a credible simulation platform.

In essence, traffic simulators are used to test various aspects of driving including trajectory, behavior, traffic scenarios, and lane keeping. There are already many existing simulators developed for specific function verification of connected and automated driving, for example, CARLA for autonomous vehicle (AV) design (Dosovitskiy et al., 2017), SUMO for traffic flow modeling (Behrisch et al., 2011), OMNET++ for V2X communication design (Varga, 2010), etc. Nonetheless, the key components of vehicle modeling, traffic flow modeling, and traffic environment are oversimplified (Hussain and Zeadally, 2018) in existing solutions.

Firstly, current traffic simulators normally simplify the function of CAV by proposing incremental improvements to the conventional traffic flow modeling methods (Tang et al., 2014, Zheng, 2014, Zhu and Zhang, 2018), which cannot reflect the characteristics of realistic CAV, especially in terms of context sensing, information communication, intelligent perceptions and control decisions. Secondly, typical AV simulators only focus on AV's individual function verification in some specific traffic scenarios (Baltodano et al., 2015, Menhour et al., 2017, Shah et al., 2018), omitting the network-level evaluation by integrating both large-scale traffic networks and V2X communication. Specifically, evaluating AV in a connected environment is imperative in the context of currently wide deployment of communication infrastructures and Internet of Things devices. Thirdly, a more realistic description of mixed traffic flow is still limited in current simulators which consists of both traditional vehicles and CAVs with different levels of automation, i.e., the called conventional, connected and automated driving (CCAD). The lack of such traffic assessments may significantly impede the smooth shifting of CAVs from prototypes to industrialized production and wide market adoption.

To summarize, it still lacks a one-stop solution which can provide the unified and credible simulation platform for CCAD assessment from the individual level to the large-scale network level. In this paper, we aim to develop a comprehensive simulation platform for CCAD from a transportation cyber-physical system perspective, in which the core components of V2X communication, traffic networks, and autonomous/conventional vehicle model are tightly combined. Specifically, three popular open-source simulators SUMO, Omnet++, and Webots (Michel, 2004) are integrated and connected via the traffic control interface, and the whole simulation platform is deployed in a Client/Server model. Moreover, the architecture design for multi-user access allows multiple human driving simulators to join the same simulation scenario, which may facilitate the exploration of the complicated interactions between conventional vehicles and CAVs. The proposed simulation platform provides a realistic traffic environment (e.g. map topology, communication and traffic infrastructure, traffic demands, etc.), supports different types of vehicle dynamics and driving behaviours including conventional vehicles and CAVs, and is capable of simulating scalability.

Our main contributions in this paper are as follows. 1) We propose an integrated simulation framework for CCAD from the cyber-physical systems perspective, in which the core components of V2X communication, traffic networks, and autonomous/conventional vehicle model are tightly coupled. 2) Mixed traffic flow is considered in the proposed simulation platform in order to mimic more realistic traffic scenarios especially at the current stage. 3) Two typical CCAD applications, traffic flow optimization and vehicle eco-driving, are demonstrated to showcase the functionality and feasibility of the simulation platform.

The rest of this paper is organized as follows. A brief literature review is given in Section 2. In Section 3, we first generalize the whole architecture of the proposed CCAD simulation platform, then introduce the core design components. In Section 4, we first develop a mixed traffic flow, and then demonstrate the simulation platform via typical applications: mixed traffic flow, traffic flow optimization, and individual eco-driving, followed by the conclusion in Section 5.

2. Literature review

Simulation is considered as an effective tool for CCAD verification since the practical implementation and deployment of such complex applications require high cost and intensive labor. In this section, we review the popular simulators regarding CCAD from different perspectives.

2.1. Microscopic Traffic Simulators

Traditional microscopic traffic simulators have been developed for long time which can emulate the time variability of traffic phenomena from the dynamics of individual vehicles (Barceló et al., 2010). Generally, a traffic simulator consists of three major components: (1) transport network to define road topology at a network level, (2) traffic demand generator to create traffic flow running in the predefined traffic network, and (3) car-following (CF) and lane-changing (LC) models to regulate the vehicle driving behavior.

Some typical simulators include VISSIM (Fellendorf and Vortisch, 2010), Aimsun (Barceló and Casas, 2005), and SUMO (Behrisch et al., 2011). Specifically, VISSIM and Aimsun are two popular commercial simulators distributed worldwide. They can provide three dimensional preview and statistical simulation results. Some recent work (Sun et al., 2015) further integrates VISSIM with driving simulators to explore multi-vehicle interactions. SUMO (Simulation of Urban MObility) is an open source, multi-modal traffic simulator. It allows defining different vehicle types with different CF models and supports diverse network formats from other traffic simulators, for example, VISUM, VISSIM, or MATsim. Specifically, SUMO allows an external application to connect to and interacts with a simulation via a general traffic control interface (TraCI), which could make it possible to bidirectionally couple traffic simulators with other software.

These simulators provide a general evaluation of traffic dynamics at a large-scale network level such as urban and highway traffic scenarios. However, the vehicle model in these simulators cannot meet the requirement of the high fidelity of vehicle dynamics, for example vehicle engine and on-board sensors, at an individual level, which is particularly essential to the connected and automated driving simulation and validation.

2.2. Physics-based (vehicle) simulators

To model and simulate the real physical vehicle, classical engineering tools are used which can incorporate testing of the sensors, vehicular dynamics, controller design, actuators, and so on. Some representatives include CarMaker (Wittenburg, 2007), PreScan (Tideman and Van Noort, 2013), ANSYS (Sovani, 2017), Vires VTD (Zhao et al., 2019), dSPACE (Smith et al., 2003), and Carsim (Dupuy et al., 2001). Based on these simulators, vehicle dynamics models can be easily built up with the integration of different sensor types, including radar, lidar, camera, GPS, etc. By taking advantage of graphics engine, users can also construct more realistic traffic scenarios which combine road sections, infrastructure components, weather conditions, and light sources. The simulators can be used in the design and simulation of Advanced driver-assistance systems via Software-in-the-loop and Hardware-in-the-loop (HIL) validations.

Because automated vehicle driving essentially can be regarded as one type of robot operation, several open-source robotics simulators, such as Webots and Gazebo (Koenig and Howard, 2004), have also been used in the simulation of AVs. They adopt Open Dynamics Engine (Smith et al., 2005) for the detection of collisions and dynamic simulation of the rigid body, and flexibly equip vehicles with a large collection of sensors. They can also create an ideal traffic environment for AVs by 2D/3D rendering engine. Specifically, Webots supports multiple programming languages (C/C++/Python/Matlab, etc.) and allows running external controllers which provide flexible and friendly user interfaces for developers.

More recently, the dedicated open-source simulators on AV have been developed which focus on the advanced AI algorithm implementations, such as CARLA, DeepDrive (Quiter and Ernst, 2018), and AirSim (Shah et al., 2018). They normally can create and render high-resolution 3D traffic environment based on game engine (Unreal or Unity), in which the motion planning of an AV can be designed by taking advantages of context perception and machine learning. The main problem with game engine simulators is that when a high-fidelity simulation is required, correct mathematical representation of subsystem in models or software is imperative to achieve realistic calculations (Rosique et al., 2019). This software is often validated with HIL tests, used to a large extent in the evaluation of computer-based test equipment.

Table 1: Summary of simulators in CCAD

Integrated simulator	Vehicle model		V2X COM.	Traffic model	Visualization		User interface	License
	Physics	Sensor			Graphics	3D map		
Veins	-	-	Omnet++	SUMO	-	-	-	GPL
Prescan	CarSim	X	-	Vissim	Unreal	X	XIL	Commercial
DYNA4	Simulink	X	-	SUMO	X	X	XIL	Commercial
dSPACE	ASM	X	-	X	X	X	XIL	Commercial
rFpro	X	X	-	SUMO	X	X	XIL	Commercial
Webots	X	X	-	SUMO	OpenGL	-	Driving	GPL
CARLA	X	X	-	SUMO	Unreal	X	XIL	GPL
CAT	Gazebo	X	-	SUMO	X	X	HIL	GPL
ANSYS	X	X	-	-	X	X	HIL	Commercial
Our CCAD	Webots	Webots	Omnet++	SUMO+ Mixed flow	OpenGL	X	Driving	GPL

2.3. Integrated simulators

Although current single simulators can explore certain function of connected and automated driving, it is still challenging to address the increasing complexity of the traffic scenarios in the context of emerging mixed traffic. In this regard, some recently integrated simulators have been developed which combine and fully take advantages of existing mature simulators.

To explore the coupled interaction between traffic mobility and V2X communication, a feasible solution is to integrate a traffic simulator with a network simulator. Some typical integrated simulators include TraNS (Piorkowski et al., 2008), iTETRIS (Krajzewicz et al., 2012) and Veins (Sommer et al., 2010), etc. TraNS federates a traffic simulator SUMO and a networking simulator NS-2 (Issariyakul and Hossain, 2009), while iTETRIS integrates SUMO and NS-3 (Riley and Henderson, 2010), and Veins couples SUMO with OMNET++. All the three integrated simulators utilize TraCI as the communication interface which adopts a very similar command-response approach and a TCP connection. These integrated simulators have been widely applied in system validations such as platoon-based cooperative driving (Jia and Ngoduy, 2016, Segata et al., 2014) and communication optimisation (Sommer et al., 2011).

Some commercial simulators also extend their capabilities to interact with other software in improving the scalability and fidelity of connected and automated driving simulation. DYNA4 (Kaths et al., 2019) which originally focuses on operational decision making level is extended by SUMO's tactical driver decisions, aiming at virtual test drives in complex surrounding traffic with realistic reaction on traffic and traffic control and reduced parametrization effort. rFpro (Cottignies et al., 2017), one commercial game engine, can also provide flexible interface to external software. For example, it allows you to populate the virtual test world with intelligent traffic, from swarm tools such as SUMO and VISSIM. It also allows you to create specific scenarios, such as a potential collision at an intersection, using tools such as CarMaker Traffic, or even under direct control via Simulink IO Block. Similar popular solutions also include dSPACE and Prescan.

To explore the advantage of AI applications in traffic optimization, an open-source project called Flow (Wu et al., 2017) is developed which integrates the deep reinforcement learning framework with microscopic simulators such as SUMO and VISSIM. Bhadani et al. (2018) proposed the CAT (Cognitive and Autonomous Test) Vehicle Testbed to mimic dynamics of a real vehicle in simulation and then transition to reproduction of use cases with HIL. They adopt Gazebo to build up the virtual traffic environment and create ROS-based controllers with the help of MATLAB/Simulink toolbox.

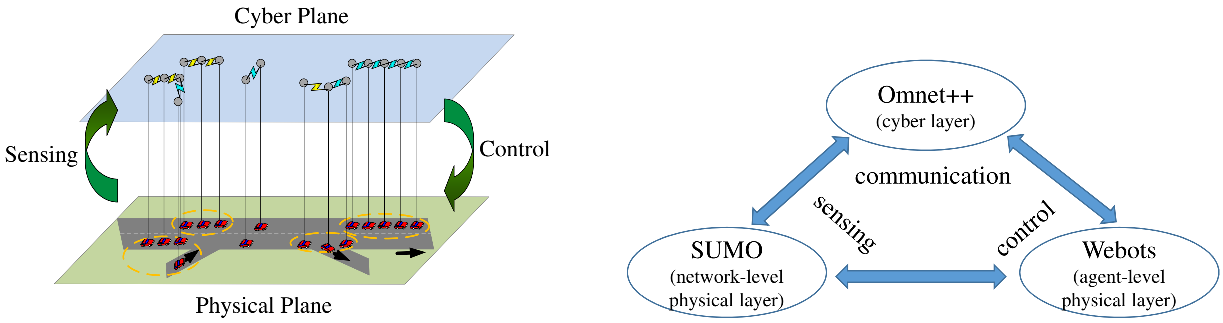
Table 1 summarizes the recent popular integrated simulators in the area of CCAD, featured by the criteria of vehicle model, traffic model, visualization, user interface, etc. It is clearly that although the existing simulators demonstrate their advantages in simulating and validating some aspects of connected and automated driving, a general simulation platform for CCAD is still missing which integrates the core component of V2X communication, traffic networks, traffic model and vehicle model, and can provide the comprehensive assessment of CCAD from the individual level to the large-scale network level.

3. Platform structure and design

3.1. Design principles and platform architecture

In principle, a deep understanding of how to appropriately describe CCAD is critical for the simulation platform design. In this part, we model CCAD from the vehicle (individual) level and system level, respectively.

In terms of vehicle modeling, we describe a conventional/CA vehicle as an agent which includes four major components (Jia and Ngoduy, 2016): i) the physical dynamics which inherently characterize vehicle behaviour stemming from manufacture, e.g., engine and actuator; ii) the sensors to provide information exchanged among vehicles, e.g., the vehicle kinetic information (based on GPS) and environment context (via Lidar/Radar); iii) the communication component which can describe the connectivity topology of vehicular networks, such as predecessor-follower, leader-follower and bidirectional; iv) the controller such as control algorithms or CF/LC rules to be implemented on each vehicle.



(a) Concept of cyber-physical system (Jia et al., 2016)

(b) Implementation in simulation

Figure 1: Design simulation platform from transportation cyber-physical systems perspective

In terms of system modeling, CCAD can be regarded as a typical transportation cyber-physical system (TCPS) which is characterized by the tight coupling between vehicles' physical dynamics (mobility) and the behaviors of vehicular networks (Jia et al., 2016). As illustrated in Fig. 1(a), the TCPS consists of two planes, a physical plane to describes the traffic mobility under the constraints of traffic environment, and a cyber plane to describe the behaviors of vehicular networks formed by connected vehicles (CVs). The sensing and control functions can be implemented by on-board sensors and driving controller (automatically or manually), respectively.

Apparently, the envisioned CCAD simulation platform consists of three core components: conventional/CA vehicle model, traffic dynamics/environment, and V2X communication. To implement these components, we select the corresponding popular open-source simulators: Webots, SUMO, Omnet++ as the backbone of the simulation platform (shown in Fig. 1(b)), wherein Webots is used for building physical vehicle model and specific controller design of CAV/AV, SUMO is used for generating traffic networks and realistic traffic demands, and Omnet++ helps to model vehicular networks for CVs. In particular, Webots also supports the unmanned aerial vehicle (UAV) modelling, by which a more complicated traffic scenario can be easily simulated with the integration of UAV application. It is noteworthy that different from the existing ones that concentrate on some specific functions, the main purpose of the proposed simulation platform is to deploy CAVs with fair fidelity into realistic traffic environment and explore their impacts in terms of social/economic benefit.

The corresponding detailed architecture of CCAD simulation platform is shown in Fig. 2. Webots can easily functionalize the core components of a typical AV including Lidars, cameras, powertrain, and the implementation of common AI algorithms. In particular, the hardware of driving force wheel and pedals set (e.g., Logitech G29 supported by Webots) can be installed via `Joystick API` of Webots to provide the operational terminal of the simulated vehicle for a driver, to let the simulation platform work as a typical driving simulator. VR (Virtual Reality) headset (e.g. HTC Vive) will also be equipped in the platform to help

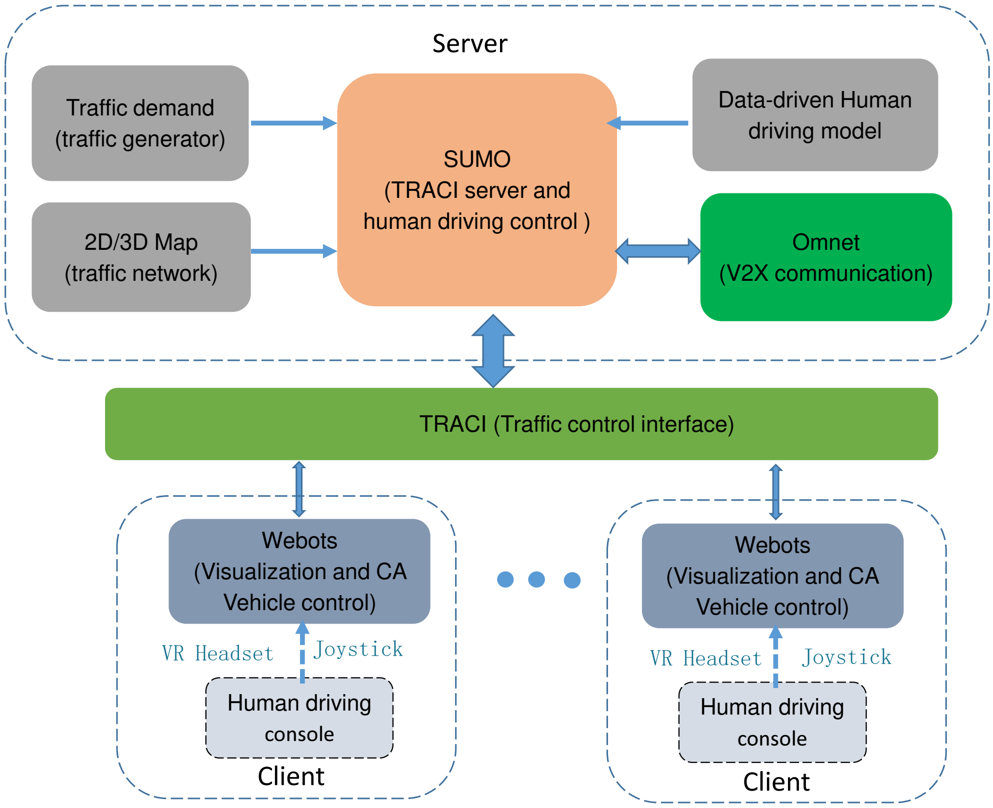


Figure 2: System architecture of simulation platform

the driver experience the high-immersion interaction with the virtual vehicles in traffic simulation scenarios. To build up a 3D traffic environment, the elevation information from the external resource (e.g. Nasa SRTM or Google Map) will be added to OpenStreetMap data which then are imported to both SUMO to create 2D map with altitude information and Webots to create 3D environment by OpenStreetMap Importer. To extend CCAD simulation from the individual level to the network level with mixed traffic, SUMO will be integrated with Webots which can flexibly build up large-scale traffic networks with different traffic conditions. OMNeT++ constructs a connected environment which supports popular V2X communication protocols such as IEEE 802.11p and LTE-V2X (It is noteworthy that 5G technologies is envisioned as the next generation solution for vehicular communication and may be introduced as appropriate in the future). Consequently, a CAV model with full functions of sensing, communication, and control is implemented by the combination of Webots, Omnet++ and SUMO.

All of the three modules are connected via TraCI. The whole simulation platform is deployed in a Client/Server model, where SUMO+OMNET++ works in a high-performance server computer, and Webots works in a client desktop computer. This architecture allows multiple end users (human drivers) to access and play in the same simulation simultaneously. This is the first demonstration that introduces multiple human driving simulators into the simulation platform in which conventional vehicles and autonomous vehicles can directly interact with each other. This type of simulation mode provides a realistic mixed traffic scenario for the verification of CCAD.

3.2. Vehicle modeling

To create a realistic mixed traffic scenario, four general types of vehicles need to be modeled in the CCAD simulation platform: conventional vehicle, CV, AV, and CAV.

3.2.1. Conventional vehicle

Two sub-types can be further defined in terms of control mode: (1) the ***abstract*** conventional vehicles created by SUMO and controlled via CF/LC models, and (2) the ***virtual*** conventional vehicle created by Webots and fully controlled by human driver via joystick. Specifically, the ***abstract*** vehicle is only designed as an node that follows the second-order kinetic formulas (e.g. speed, acceleration, etc.) and can be efficiently deployed in SUMO to form the large background traffic flow. The ***virtual*** vehicle, however, considers the realistic kinetic model (e.g. Ackermann steering geometry) and engine model (e.g. engine type, transmission, gearbox, etc.) in Webots so that it can mimic the real vehicle dynamics to be controlled by a human driver or automated driving strategy. Fig. 3(a) and Fig. 3(b) show the typical descriptions of the two types of conventional vehicle.

3.2.2. CV

A CV in this paper is modeled as the integration of a conventional vehicle with the communication function implemented by Omnet++. Accordingly, accurate modeling of V2X communication is critical to the authentic verification of individual CA and network-level connected environment. In the simulation platform, we adopt Omnet++/Veins to model the *de facto* vehicular networking standard, IEEE 802.11p and IEEE 1609.4 DSRC/WAVE.

Specifically, MixiM framework is adopted in Omnet++/Veins to model physical layer effects, such as radio wave propagation, interference estimation, radio transceiver power consumption and wireless MAC protocols. Veins also includes higher layer models of the DSRC/WAVE stack for channel hopping according to the standard (i.e., switching between CCH and SCH, that is, control channel and service channel), if this is desired. In addition, the application framework of Wave Short Message (WSM) and beaconing is defined and handled in Veins.

The implementation of V2X communication for CV is shown in Fig. 3(c). The typical communication function is described by a NED file which models a ***cyber car*** (cpsCar) and consists of an 802.11p Network Interface Card (NIC) to be able to communicate with others, an application layer directly connecting to this NIC, a scenario module describing traffic scenario, and the mobility module responsible for updating the position of the car. It shall be noted that the human driving behaviour may change since the dirver's perception of surrounding environment can be improved by the accessible information via V2X communication. In Section 4.1, we will introduce a new type of CF model named CVDS-IDM for CV which can capture the impact of information perception on human driving behaviour.

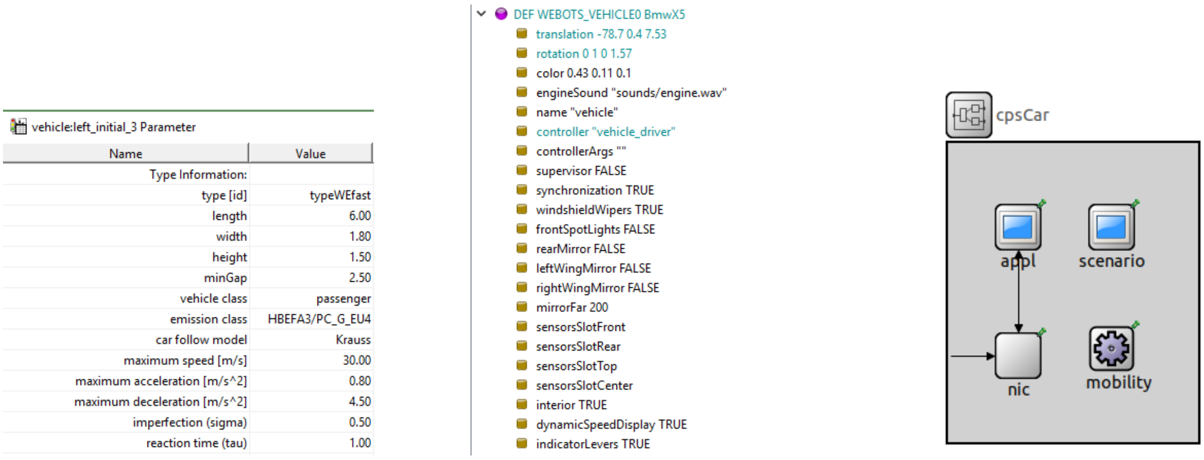
3.2.3. AV and CAV

Both the AV and CAV are created and fully controlled by Webots with thephysical dynamic model same to the ***virtual*** conventional vehicle. In addition, the AV/CAV is normally equipped with diverse on-board sensors (e.g. lidar, GPS, camera, etc.) to collect the environment information and CAV also has the communication component implemented in Omnet++. The possible control strategies may include traditional control algorithms such as model predictive control and consensus control, and advanced AI algorithms, e.g., reinforcement learning, fuzzy control, etc.

It should be noted that although the three types of vehicles, ***virtual*** vehicle, AV and CAV, are fully controlled by Webots, the *shadow* CF model is still required in SUMO which is used to synchronize the vehicle's motion and location in both SUMO's and Webots's coordinate system. Specifically, the shadow CF model does not calculate the vehicle's acceleration and speed for the next step (which is normally conducted in traditional CF models, e.g. IDM), and only periodically updates the vehicle's mobility and position based on the information received from Webots.

3.3. Building up traffic scenario

Building up realistic traffic scenarios is critical to the credible CCAD simulation. A typical traffic scenario for CCAD testing includes traffic environment construction (static scenario) and traffic demand generation (dynamic scenario).



vehicle:left_initial_3 Parameter

Name	Value
Type Information:	
type [id]	typeWEfast
length	6.00
width	1.80
height	1.50
minGap	2.50
vehicle class	passenger
emission class	HBEFA3/PC_G_EU4
car follow model	Krauss
maximum speed [m/s]	30.00
maximum acceleration [m/s^2]	0.80
maximum deceleration [m/s^2]	4.50
imperfection (sigma)	0.50
reaction time (tau)	1.00

(a) Abstract vehicle (b) Virtual vehicle (c) CV

Figure 3: Modeling vehicle types

3.3.1. Traffic environment construction

High-quality virtual traffic environment is required for control strategy design and visualization of CCAD, which can be constructed based on a real road network map, i.e., OSM file, imported directly from OpenStreetMap and then translated into the 3D-world description file by Webots. With the help of the tool NETCONVERT, the OSM file can also be converted into a SUMO network file. To represent a realistic physical movement in road networks, Webots uses Open Dynamics Engine (ODE), a high-performance, open source library, for detecting vehicle collisions and simulating rigid body dynamics.

Besides the road networks, the indispensable real-world elements can be well modeled via the world description file in Webots, which is a 3D description of the properties of robots and of their environment. It contains a description of every object: position, orientation, geometry, appearance (like color or brightness), physical properties, type of object, etc. Worlds are organized as hierarchical structures where objects can contain other objects (like in VRML97). Webots provides advanced 3D rendering capabilities by utilizing openGL engine which can generate very realistic simulation images. In addition, since the CCAD platform integrates the function of Omnet++, the communication infrastructures, for example, roadside units, can be easily deployed in the simulation to construct a fully connected environment.

3.3.2. Traffic demand generation

Reproducing realistic background traffic flow in the simulation is critical for CCAD validation. To associate the simulation with the real-world traffic demand, several methods are provided in SUMO: (1) ACTIVITYGEN to generate demand from a description of the population in the net, (2) OD2TRIPS to convert the real Origin-Destination-Matrices obtained from traffic authorities to single vehicle trips, (3) DFROUTER to use induction loop data to compute vehicle routes, and (4) DUAROUTER to import demand data given by source and destination edges to obtain routes through the shortest path computation.

3.4. Communication interface and information synchronization

3.4.1. Communication interface

Undoubtedly, the timely and seamless information synchronization among SUMO, Omnet++, and Webots is very important to the CCAD simulation, which can be designed by taking the advantage of TraCI. TraCI uses a TCP based client/server architecture to provide access to SUMO. Specifically, TraCI supports multiple clients (Omnet++ and Webots in this study) and executes all commands of a client in a sequence. In order to have a predefined execution order, every client should issue a `SetOrder` command before the first simulation step which assigns a number to the client and commands from different clients during the same simulation step will be executed in the order of that numbering. TraCI also support variable subscription mechanism which allows you to ask once for a set of a structure's variables (for example, vehicle-related information) and retrieve them periodically. SUMO develops python/C++ API for external applications to call TraCI function.

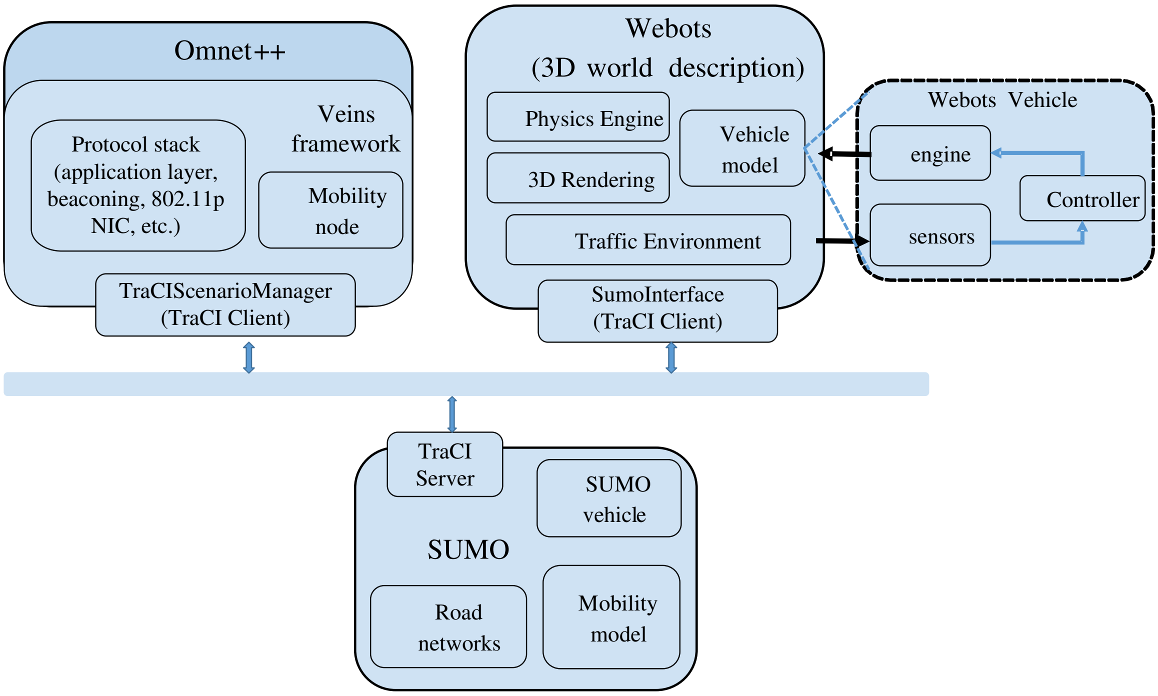


Figure 4: Communication interface among components

It is noteworthy that the clients Omnet++ and Webots are not allowed to communicate to each other directly in the proposed simulation platform. Consequently, the two communication interfaces of SUMO-to-Omnet++ and SUMO-to-Webots need to be implemented in the platform, as shown in Fig. 4.

To interact Omnet++ with SUMO, we apply Veins which designs a module of `TraCIScenarioManager` to bi-directionally couples the two components. By using this module, Veins can obtain the information about the movements of vehicles in the simulated road traffic scenario and updates their mobility information within OMNeT++. We also extend the module function to send the surrounding vehicle information collected by V2X communication to SUMO.

To build up the communication between Webots and SUMO, the `SUMOInterface` has been developed in Webots which is written in Python in a Supervisor controller. The interface with SUMO ensures the introduction of a large number of vehicles in Webots which recreate a real traffic situation. SUMO vehicles are created and moved to Webots by using vehicle information retrieved from SUMO with TraCI and the Webots Supervisor API. Moreover, the Supervisor API is helpful to restart or end the simulation, to read or change the position of the various actors of the simulation, and to read the status red lights. On the other hand, the AV/CAV controlled in Webots is moved to SUMO using TraCI. We also introduce some new functions in `SUMOInterface` to better support V2X information exchange as well as the 3D visualization of vehicle's movement.

3.4.2. Information synchronization

The V2X information type supported in present simulation platform is the ***beacon***, i.e., the periodical message dissemination among CVs to inform the current kinematic status to neighbors. Specifically, the information may include a vehicle's 3D position, speed, direction, and acceleration, the message's timestamp, etc. According to the system architecture, the received neighbour's beacon will be stored in SUMO, in which the new variables can be easily extended by `VehicleVariables` class defined in CF models of SUMO.

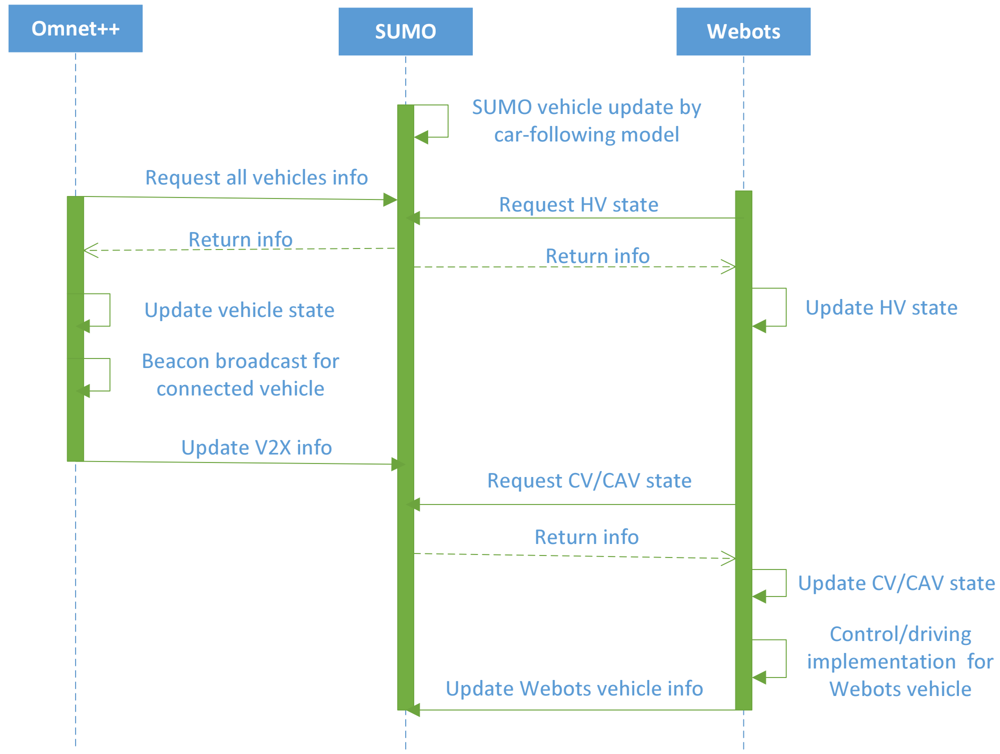


Figure 5: Sequence diagram for information synchronization among components

In addition, we associate the new vehicle types, CV and CAV, with the dedicated CF models in SUMO for the purpose of processing the additional beacon inforJoystick mation. For a CV, we create new CF models which allow the storage of V2X information but still comply with traditional models such as IDM model (Treiber et al., 2000) and Krauss model (Krauß et al., 1997). While for a CAV, since it is fully controlled by Webots, its shadow CF model in SUMO is only used for processing the V2X information.

The sequence diagram for information synchronization within platform at each simulation step is shown in Fig. 5. Initially, SUMO update each ***abstract*** (conventional) vehicle by running certain CF model. Then Omnet++ request all vehicles' information in SUMO and update their position in Omnet++. Based on the obtained latest vehicle information, the CV broadcast/receive the beacon to/from neighbours via V2X communication. The V2X information will be sent to SUMO for updating the corresponding variables stored in SUMO vehicles. All these steps will be executed within the fixed time duration predefined by the platform. On the other side, Webots will first request all HVs' states from SUMO at the beginning of simulation step and update their position in Webots. After Omnet++ implements the V2X communication, Webots will request and update CVs/CAVs' states including V2X information from SUMO. Then the ***virtual*** vehicles will be driven by users and CAVs are controlled by control algorithms in Webots, respectively. Finally, these two types of Webots-controlled vehicles will update their state to SUMO. It should be noted that all three components are time synchronized, which means the three components adopt the same simulation clock. Therefore, the information update for each component can be synchronized by setting the predefined time offset in order to avoid information inconsistency. In addition, the `SetOrder` command also guarantees the accurate order of both Omnet++ and Webots accessing to SUMO, as defined in Fig. 5

The screenshot of Fig. 6 demonstrates that the three simulators can run in parallel via the information synchronization. In particular, each vehicle is represented in both the cyber plane (Omnet++) and the physical plane (SUMO/Webots), and has the corresponding reflection in all three simulators.

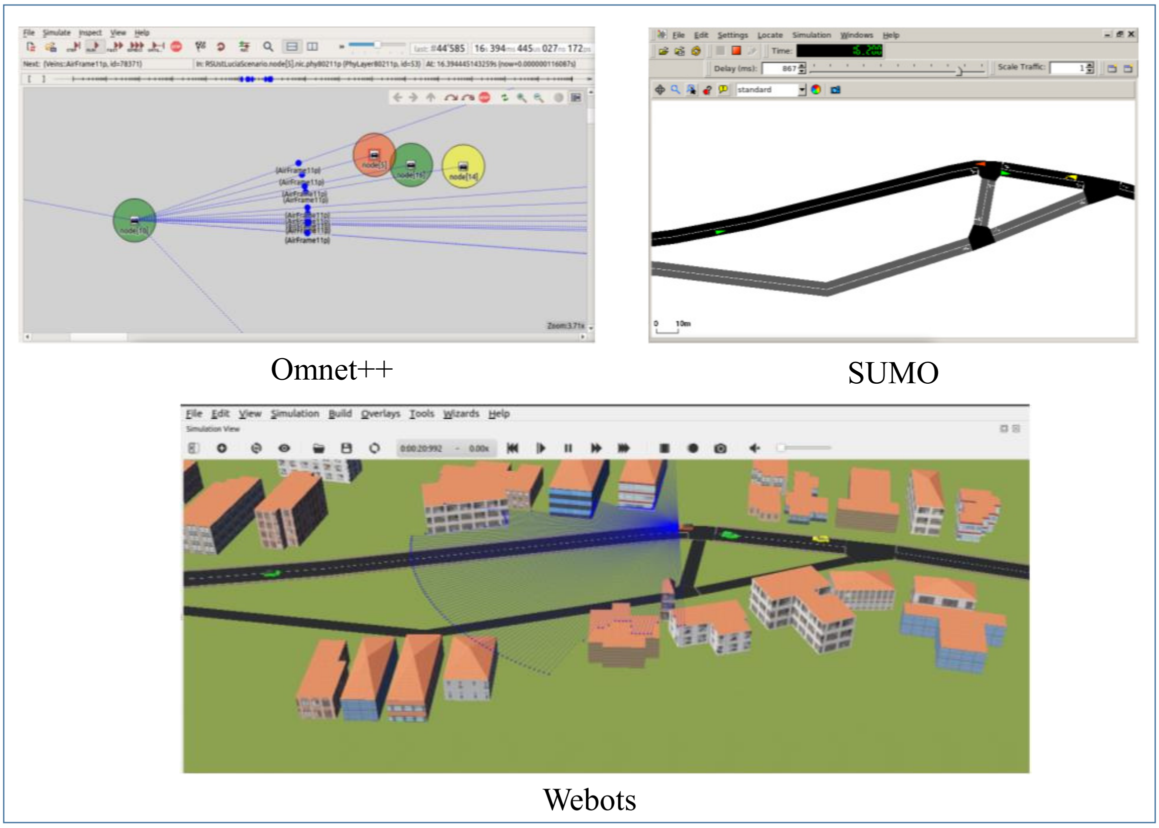


Figure 6: Simulation screenshot

4. Case Study

In this section, we first introduce the mixed traffic flow model adopted in the simulation platform, then implement two use cases, traffic flow optimization and vehicle eco-driving, to demonstrate how to apply the proposed simulation platform in CCAD evaluation and assessment.

4.1. Build up realistic mixed traffic flow

In the mixed traffic flow, the driving behaviour of conventional vehicle can be modeled by typical CF models such as IDM model, and CAV will be fully controlled by Webots. To describe the CV's driving behaviour, we adopt the Connected Vehicle Driving Strategy (CVDS) (Sharma et al., 2019b) which explicitly models the driver compliance to the continuous information and the advanced event-triggered information. CVDS is a general driving strategy that can be integrated with any existing CF models to describe the CV's CF behaviour. Moreover, CVDS includes two components: (a) part I incorporates the impact of continuous information into the model; (b) part II incorporates the impact of the advanced event-triggered information (currently limited to modeling the hard braking cases only). The information that is always presented on the windscreen/ driver assistance devices is called the continuous information whereas the information that is delivered to drivers a few seconds before an event occurs is called the advanced event-triggered information. For simplicity, in this study, we only consider the continuous information and its impact on CF behaviour in connected environment. The driver compliance in CVDS is modeled using prospect theory (Kahneman and Tversky, 1979) and it is an integral element of both the parts.

4.1.1. CVDS Part I: Modeling the driver's response to the continuous information

When modeling the driver response to the continuous information, the stimuli i.e., the relative speed and the spacing are considered as estimation error-free since connected environment provides such information to drivers. The assumption here is that drivers devote their undivided attention to such information. The time gap parameter in CF models is multiplied with $(1 + UT(h_{obs}))$ to accommodate the impact of driver compliance on the CV's CF behaviour.

The mathematical formulations of CVDS-IDM part I are presented in Eq. (1) and Eq. (2):

$$a_n(s_n, v_n, \Delta v_n) = a\Big[1 - \Big(\frac{v_n}{v_0}\Big)^\delta - \Big(\frac{s^*(v_n, \Delta v_n)}{s_n}\Big)^2\Big] \tag{1}$$

$$s^*(v_n, \Delta v_n) = s_0 + (1 + UT(h_{obs}))T_0 v_n + \frac{v_n \Delta v_n}{2\sqrt{ab}} \tag{2}$$

where $(UT(h_{obs}))$ is thee utility value calculated at h_{obs} using prospect theory shape parameters. The parameter h_{obs} is the observed headway between the follower and the leader measured at the time when the messages are received by the followers. In the case of continuous information, the h_{obs} measured at each time since messages are available all the time. A detailed discussion is referred to Sharma et al. (2019b) on how h_{obs} is evaluated using prospect theory.

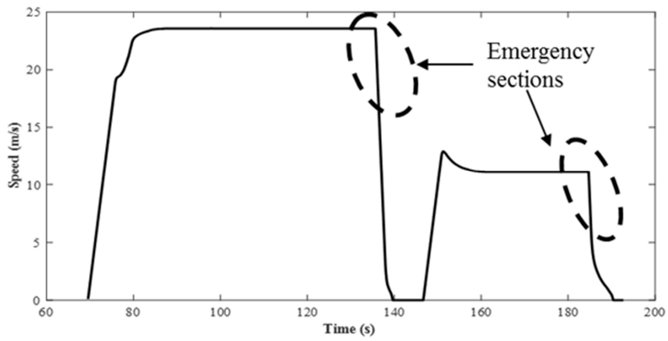
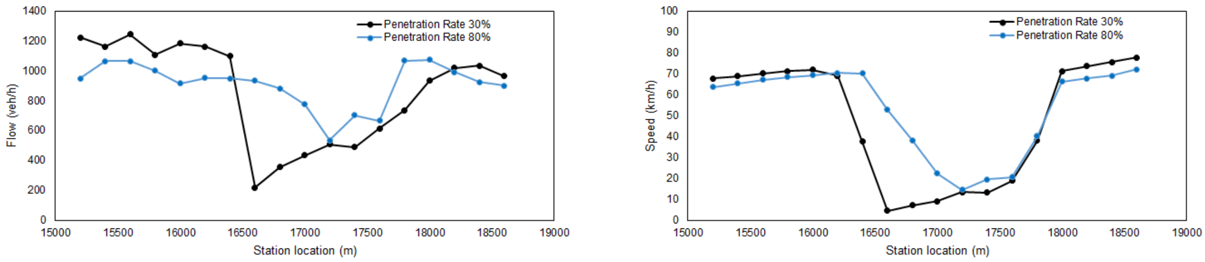


Figure 7: Experiment scenario

To calibrate the $UT(h_{obs})$, a dedicated driving simulator experiment was designed in which the high-quality trajectory data were collected from two specific scenarios, namely baseline (representing TE) and connected (representing CE) (Ali et al., 2020). In each scenario, participants had to follow a platoon of vehicles on a single-lane motorway for 3 km. In the baseline scenario, no information assistance was provided to the participants, and in the connected scenario, to simulate vehicle-to-vehicle and vehicle-to-infrastructure information dissemination, continuous and uninterrupted information assistance was provided to participants using a driving aid. In the CF section of the experiment, the leading cars (programmed vehicles) followed the speed profile displayed in Fig. 7, whereas the speed profile of the driven car's (car driven by the participant) was governed by the participant's CF behavior. The vehicular interactions are designed such that drivers undergo all the driving regimes transitions, which leads to a dataset containing complete trajectories. A trajectory is complete if it constitutes all the six driving regimes, i.e., free acceleration, cruising at the desired speed, following the leader at a constant speed, accelerating behind a leader, decelerating behind a leader, and standing behind a leader (Sharma et al., 2019a). Completeness of the driving regimes is an important aspect of the trajectory data quality and critical for the purpose of CF model calibration and validation. Note that the vehicle interactions remain the same in the baseline and in the connected scenario.

4.1.2. Modeling the mixed traffic of CVs and conventional vehicles

To simulate the mixed traffic, a platoon of 61 vehicles is generated on a single-lane facility of 5 km. The first vehicle's trajectory is generated manually. The following vehicles are the mixed vehicles, in which IDM



(a) Effect of penetration rate of CVs on traffic flow

(b) Effect of penetration rate of CVs average speed

Figure 8: A comparison of fundamental traffic flow parameters at different penetration rates of CVs in the mixed traffic stream of connected and conventional vehicles.

with estimation errors (HIDM) (Treiber et al., 2006) and CVDS integrated with IDM are utilized to model conventional vehicles and CVs, respectively.

The platoon leader is at standstill for t = 20 s at a traffic signal placed near 2500 m from the starting point (15000 m). The fundamental traffic flow parameters, namely flow and average speed are estimated using Edie's generalized definitions (Edie, 1963). Heavy congestion can be observed at station locations between 16000 m and 18000 m. The penetration rate of CVs is varied from 0% to 100%. For the demonstration purpose, results corresponding to the penetration rates of 30% and 80% are presented in Fig. 8. It can be observed that the flow values and average speed values are large for the CVs' penetration rate of 80% as compared to 30% in the heavy congestion section of the road.

It shall be mentioned that several recent studies have taken into account the stochasticity in modeling human driving behaviours (Treiber and Kesting, 2018), or proposed novel data-driven driving models by utilizing deep learning (Kuefler et al., 2017), which will be further implemented in our future work. In addition, some geometric features, such as road surface and gradient, which may significantly affect driving behaviour, will be considered in both CF and LC models.

4.2. Traffic flow optimization

Traffic oscillation, also known as stop/slow and go wave, has significantly negative impact on traffic flow and road users such as riding comfort, crash risk, and energy consumption (Chen et al., 2014). Several studies (Orosz et al., 2010, Wilson and Ward, 2011, Zheng et al., 2010) have shown that the instability and diversity in CF behaviour is the inherent cause of oscillation. Perturbations propagate upstream along the platoon of vehicles can lead to severe oscillations. With the development of communication and control technology, CAV has attracted much attention because of its potential positive effect on traffic flow operation. Therefore, in this case study, we aim to demonstrate how CF behaviour impacts traffic oscillation and how to optimize the traffic flow with AV/CAVs.

4.2.1. Vehicle types and control strategy

Three types of vehicles are modeled in this case, i.e., conventional vehicle, AV, and CAV, where conventional vehicle is modeled within SUMO (i.e. ***abstract*** vehicle), while AV and CAV (***virtual*** vehicle) are controlled through Webots.

Conventional vehicle is modeled with the typical IDM model, in which the parameters is typical for vehicles on freeway as per Treiber and Kesting (2013): desired speed v_0=120km/h; desired time gap T=1s; maximum acceleration a=1 m/s^2; comfortable deceleration β=1.5m/s^2; minimum gap s_0=2m.

AV and CAV are modelled with the physical dynamics of BmwX5 and equipped with the camera which is used to collect the information of the preceding vehicle, i.e., the gap with the preceding vehicle and the speed of the previous vehicle. With respect to the control strategy, AV is controlled with Full Velocity Difference Model (FVDM) developed by Jiang et al. (2001). Its basic form is $\dot{v}_n(t) = k_1[v_{opt}(s) - v_n(t)] +$

$k_2[v_{n-1}(t) - v_n(t)]$, where k_1, k_2 are the sensitivity parameter, and $v_{opt}(s)$ is the optimal velocity (OV) function in terms of spacing s_n. The original OV function uses a hyperbolic tangent. To be consistent with the setup of AV, a more intuitive OV function presented in Treiber and Kesting (2013) is adopted in this study, as shown in Eq. (3):

$$v_{opt}(s) = \max\left[\min\left(v_0, \frac{s_n(t) - s_0}{T}\right)\right] \tag{3}$$

where the variables have the same meanings as introduced before. The parameters are as follow: desired speed v_0=120km/h; desired time gap T=1s; k_1=2; k_2=1.

CAV is controlled base on FVDM which considers additional acceleration information of previous CAV as per Jin and Orosz (2014). Its form is as follow:

$$\dot{v}_n(t) = k_1[v_{opt}(s) - v_n(t)] + k_2[v_{n-1}(t) - v_n(t)] + k_3 * \dot{v}_i(t) \tag{4}$$

where $\dot{v}_i(t)$ is the acceleration of preceding CAV in the platoon, k_3 is the sensitive parameter, and the other variables have the same meanings as introduced before. k_3 is set as 0.5 in this study. We adopt the defacto industry protocols IEEE 802.11p for V2V communication, and the related networking parameters are given in Table 2. The function of V2V communication is implemented in Omnet++.

Table 2: IEEE 802.11p Parameter Setting

Parameter	**Value**	**Parameter**	**Value**
Channel data rate	6Mbps	Slot time	13 μs
SIFS	32 μs	AIFS	71 μs
Preamble length	32 μs	Plcp duration	8 μs
Propagation delay	2 μs	CWmin	7
Beacon frequency	0.1 s	Beacon priority	2
Beacon size	200 bytes	Transmission range R	500m
CCH interval	46ms	Sync interval	4ms

4.2.2. Simulation scenario and results

Regarding the simulation, firstly, following the setup in Sun et al. (2018), a platoon of identical conventional vehicles enters a very long single-lane straight road and quickly reaches the equilibrium state, driving at the same speed (10m/s, i.e. 36km/h) and the corresponding equilibrium gap. At t=50s, a disturbance is introduced to the first leading vehicle by forcing it to first decelerate and then accelerate for the same time duration (5s) with the same absolute acceleration value (1/-1 m/s^2). The number of vehicles in the platoon is 11 while the following 10 vehicles are controlled by IDM in SUMO. The time step is 0.05s. The time-speed diagram is shown in Fig. 9. The oscillation amplitudes of vehicles are measured by the largest speed drop during the deceleration and acceleration period (Saifuzzaman et al., 2017, Zheng et al., 2011), as the speed drop of the kth vehicle in ith cycle of oscillation is defined as $v_{drop}^{k,i} = v_{dec}^{k,i} - v_{acc}^{k,i}$ and the oscillation amplitude is the maximum speed drop of kth vehicle within infinite time. The average oscillation amplitude of the conventional vehicles based platoon is 4.121 m/s.

We then integrate AV/CAV into the platoon and consider two configurations for each scenario: the AVs/CAVs are evenly distributed in the platoon; all the AVs/CAVs are allocated in the front of the platoon. In addition, considering the uncertainty of connected environment, we further study the impact of connection loss on the conventional and CA vehicle mixed platoon oscillation.

In this case study, we replace 5 conventional vehicles with AVs/CAVs. The AVs/CAVs are controlled with the cruising speed control in Webots. More specifically, at each time step, we calculate the target speed of AV/CAV according to the models as described before and set the cruising speed of corresponding Webots vehicle as the target speed. The time-speed diagrams for different scenarios are shown in Fig. 10, and the average oscillation amplitudes of different scenarios are presented in Table 3.

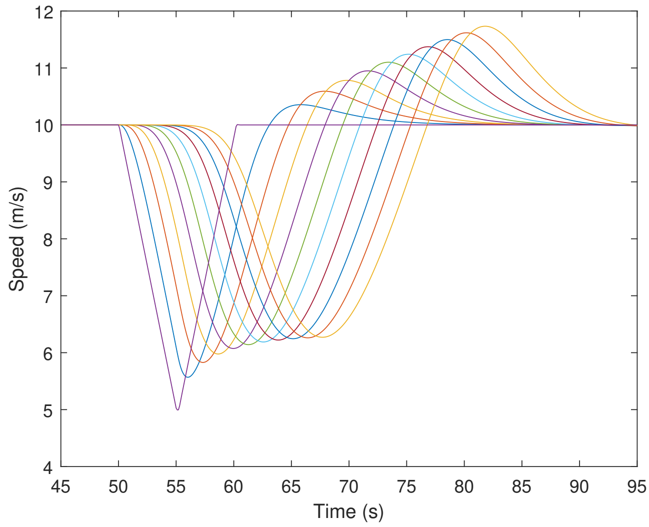


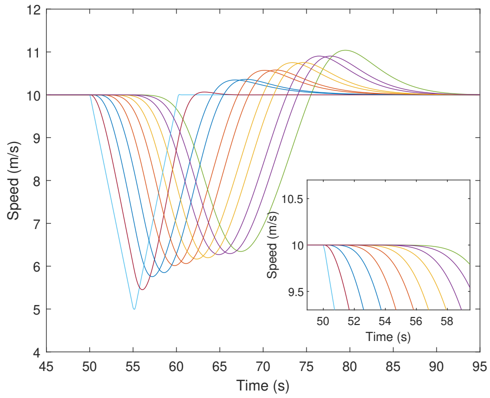
Figure 9: Time-speed diagram for conventional vehicles based platoon

Table 3: The impact of AV/CAV on platoon oscillation

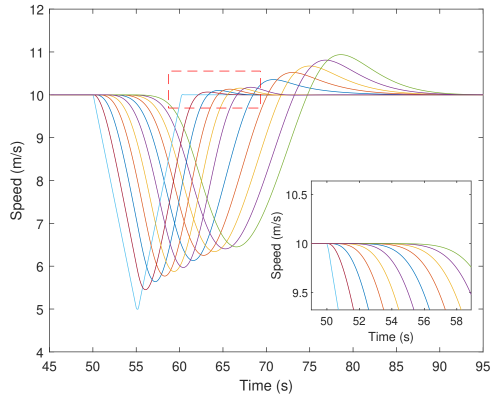
Scenario	Average oscillation amplitude	
	Even distribution	Front distribution
conventional platoon	4.121	
conventional/AV platoon	4.066	4.055
conventional/CAV platoon	3.860	3.739
conventional/CAV platoon with 50% loss	3.877	3.742

As indicated by Fig. 10 and Table 3, the incorporation of AV and CAV can both improve the traffic flow, while the connectivity between vehicles can significantly mitigate the traffic oscillation. Regarding the location of AV/CAV in the platoon, the impact of AVs' locations shows minor discrepancy, whereas putting CAVs together in the front of platoon is much better than distributing them evenly, which is consistent with the recent study of Jia et al. (2019). This indicates that the information from closer preceding vehicles is more important to CAVs. Moreover, the connection loss has negative impact on the traffic flow optimization, while the impact grows with the increase of CAVs' distance, as the negative impact of connection loss is larger when using even distribution (0.017) compared to front allocation (0.003). The results demonstrate that CAV platoon is more beneficial for the optimization of traffic.

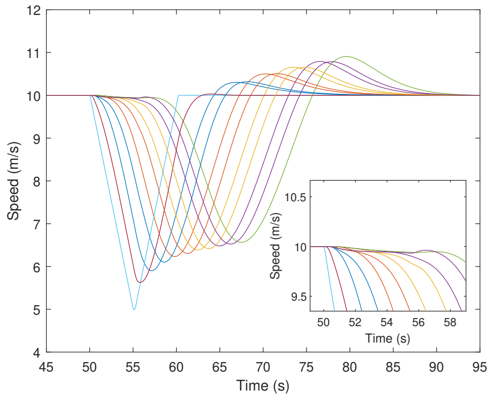
In this case study, we evaluate traffic flow optimization by our CCAD platform. Compared to other related work, the CCAD platform can provide a more realistic testing environment and more reliable evaluation. Firstly, the defacto industry protocols IEEE 802.11p is integrated into CAVs in this study, which can reflect the realistic impact of V2X communication, e.g. stochastic time delay and packet loss, on the control strategies. In contrast, most studies adopt simple assumptions for vehicular communication (e.g., fixed connection delay and connection range), which may facilitate the problem solution but could compromise the accuracy of testing results (Jin and Orosz, 2014, Sun et al., 2018). Secondly, the AV/CAVs are built on the high-fidelity physical vehicle model in Webots, and thus the control performance obtained from the simulation is more close to the reality. Previous studies (Jia et al., 2019), however, usually ignore or simplify the vehicle model, which could result in inaccurate performance evaluation. In summary, the simulation results demonstrate that the integrated platform can provide a more realistic testing environment regarding V2X communications and its potential for many other experiments. For example, by revising vehicular communication protocols in Omnet++, new message dissemination strategy may be applied in the simulation



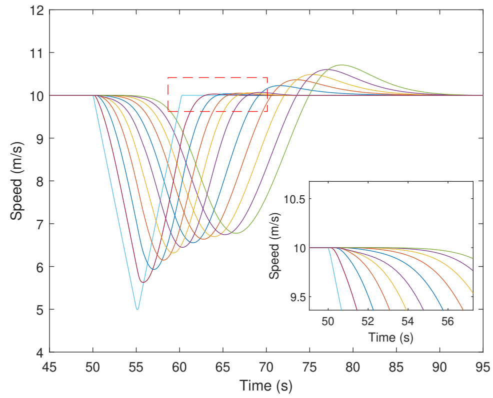
(a) Even distribution of AVs



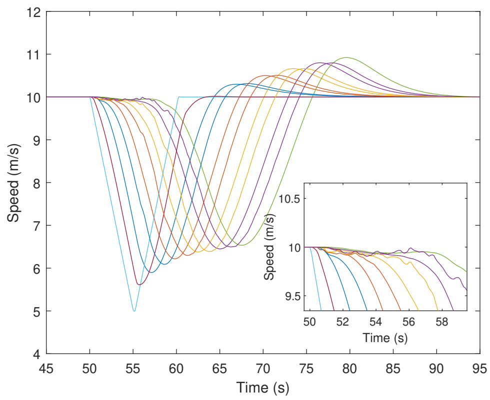
(b) Front allocation of AVs



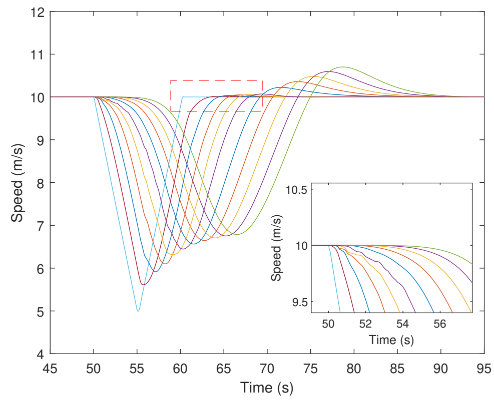
(c) Even distribution of CAVs



(d) Front allocation of CAVs



(e) Even allocation of CAVs with 50% loss



(f) Front allocation of CAVs with 50% loss

Figure 10: Time-speed diagram for conventional vehicle and AV/CAV mixed platoon

platform to improve the traffic flow stability.

4.3. Look-ahead eco-driving for AV

Eco-driving is environmentally friendly used to not only reduce fuel consumption from automobile but also mitigate CO2 emission. In eco-driving style, driver needs to simultaneously adjust the vehicle speed and accelerate the vehicle according to driving environment, determine the route decision by minimizing the fuel consumption. Some typical strategies may include new powertrain systems, cooperative driving, and traffic signal optimization (Alam et al., 2010, Osorio and Nanduri, 2015, Sciarretta and Guzzella, 2007, Wang et al., 2014). In this case study, we demonstrate how a AV adjusts the speed profile based on its on-board sensors' perception of preceding traffic conditions and explore its impact on fuel consumption.

4.3.1. Vehicle modeling and driving strategy

Table 4: AV Parameters and control coefficients

Parameter	**Value**	**Parameter**	**Value**
Vehicle prototype	BmwX5	Maximum acceleration	3.8 m/s^2
engineType	combustion	Side sensor	camera
Fuel consumption model	LDV_G_EU6	Top sensor	camera
Camera Recognition range	200 m	Rear sensor	Distance sensor
Control parameters α, β, γ_1, γ_2	1, 1, 0.1, 0.5	Switching threshold θ	0.8

The AV is modeled with the physical dynamics of BmwX5 and on-board sensors, as illustrated in Table 4. Specifically, to let the AV stay its own lane and avoid collision with the surrounding vehicle, we use a camera to recognize the road line and Rear range sensors to detect the gap to the neighbours. Moreover, an extra camera is installed on the top of the vehicle to perceive the front traffic condition, which can be realized by finding vehicles on an camera image and estimating the distance from the ego-vehicle. The speed of the preceding vehicle can also be further calculated based on the distance information difference between two adjacent time steps. To calculate the ego-vehicle's fuel consumption, we adopt the HBEFA3.1-based model which has been implemented in SUMO Krajzewicz et al. (2015):

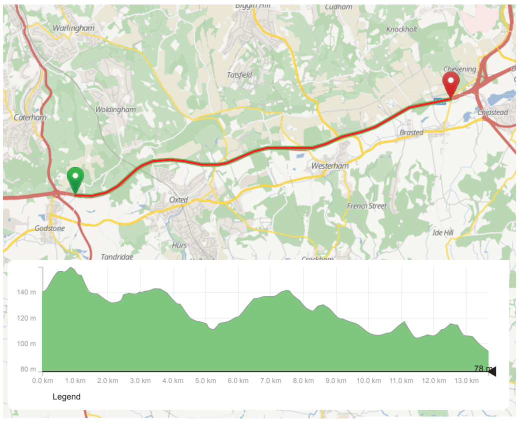
$$mu = c_0 + c_1 va + c_2 va^2 + c_3 v + c_4 v^2 + c_5 v^3 \tag{5}$$

where v and a are the speed and acceleration of vehicle, and c_0-c_5 are specified constant coefficients for certain vehicle type. It is noted that the slope information given in HBEFA was used to take the part of the missing dependency on acceleration.

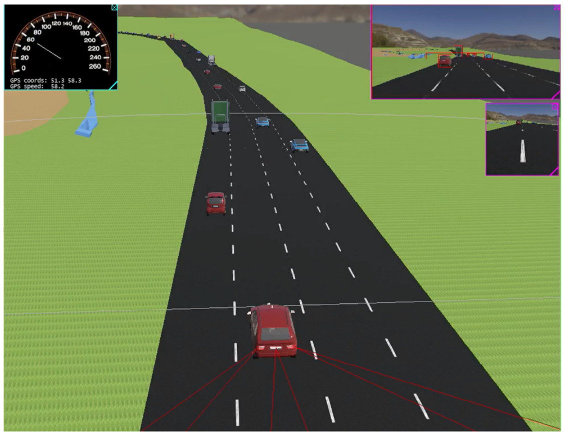
Two types of vehicle speed control methods can be used in Webots: using cruising speed which targets a reference velocity but with a constant acceleration determined by the time0to100 value in the PROTO file, and using the throttle which controls the torque of the vehicle. Since this usecase only considers the impact of downstream traffic flow speed, as the initial demonstration of eco-driving, we simply adopt the cruising speed control. For the AV driving strategy, we consider an intuitive Look-ahead control in this paper which utilizes the leading vehicle states with the preceding traffic situation:

$$u(t) = \begin{cases} \gamma_1[\alpha(d_s(t) - d_e(t)) + \beta(v_p(t) - v_e(t))] + \gamma_2(v_a(t) - v_e(t)) & \text{if } d_e(t) > \theta \cdot d_s(t) \\ \gamma_1[\alpha(d_s(t) - d_e(t)) + \beta(v_p(t) - v_e(t))] & \text{if } d_e(t) \leq \theta \cdot d_s(t) \end{cases} \tag{6}$$

where d_s and d_e are the desired gap and real-time gap between AV and its leading vehicle, respectively, v_e is the AV speed, v_p is the speed of the leading vehicle, v_a is the average traffic speed of within the camera perception range. α, β, γ_1, and γ_2 are the control coefficients, and θ defines the switching threshold of control strategy, which indicates only the preceding vehicle's information is considered in the case of smaller gap. All coefficients adopted in this simulation are summarized in Table 4.



(a) The UK highway segment in simulation



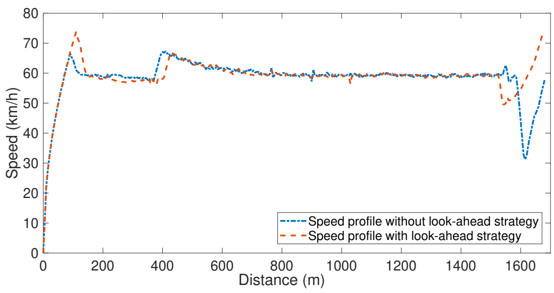
(b) Screenshot of eco-driving

Figure 11: Eco-driving simulation scenario

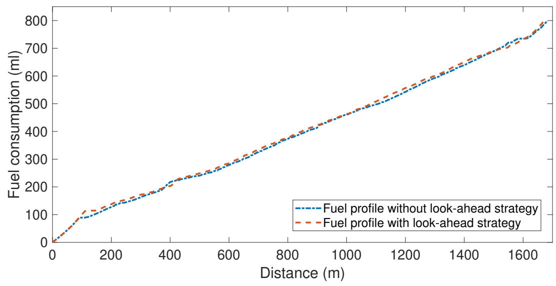
4.3.2. Simulation scenarios and results

We select one segment of the UK M25 motorway integrated with the corresponding elevation information as the target route of AV, as shown in Fig. 11(a). By taking advantage of the CCAD simulation platform, we generate different traffic scenarios to testify the look-ahead eco-driving strategy and explore the impacts of different traffic conditions on the AV. We also compare the look-ahead strategy with the simple CF strategy (i.e. only consider the leading vehicle states) to evaluate the benefit of the preceding traffic awareness.

To build up a more realistic testing environment, we create traffic demand based on the historical loop data collected on M25 highway segment. In detail, traffic demand is firstly generated by DFROUTER, and then is implemented in SUMO with IDM car-following model. Vehicles recorded in real data sets are categorized into two types, passenger cars and freight cars, and the vehicle's parameters such as vehicle speed and acceleration are set with the default values. We then adjust the sensitive parameter time gap of IDM model to minimize the deviation between the real traffic data and simulation traffic data. The simulation profiles of traffic flow and traffic speed can reflect the real traffic situation in most cases especially in stable traffic situation. Because the real dataset did not include congested traffic situation, to mimic such a scenario, we proportionally increase the loop data (i.e. traffic flow) and then create traffic demand using SUMO.



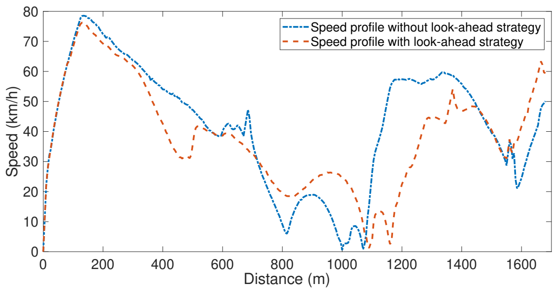
(a) Speed comparison



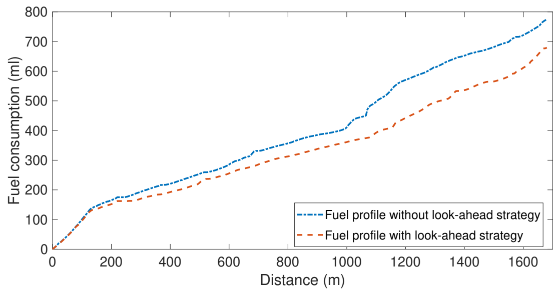
(b) Fuel consumption comparison

Figure 12: Eco-driving strategy in stable traffic scenarios

We first test the AV's performance in the stable traffic scenario where traffic flow from the real data set is set around 800 veh/lane/hour and the average traffic speed is about 90 km/h. It is observed from Fig. 12 that both the look-ahead strategy and the traditional CF strategy show the similar performance, which indicates the preceding traffic awareness does not bring significant benefit in a stable traffic scenario in terms of fuel consumption.



(a) Speed comparison



(b) Fuel consumption comparison

Figure 13: Eco-driving strategy in congested traffic scenarios

We then test the AV's performance in a heavy traffic road segment (about 1700 meters) where traffic flow (composed of ***abstract*** conventional vehicles) is set around 1500 veh/lane/hour based on the real data set and the average traffic speed drops below 20m/s. Especially, traffic jam may happen on the AV driving lane due to the mandatory lane changing for some vehicles. It is observed from Fig. 13 that the look-ahead strategy outperforms the traditional CF strategy in terms of fuel consumption. In addition, the travel time is reduced from 298 seconds to 244 seconds by adopting the look-ahead strategy. The results indicate the preceding traffic awareness can help a better eco-driving particularly in a heavy traffic scenario.

In addition to adopting the high-fidelity vehicle model in Webots, this case study demonstrates the advantage of our platform in creating larges-scale background traffic flow, by which the eco-driving strategy can be validated in a more complicated and realistic traffic scenario. In particular, based on the platform, we can further explore how the interaction between the AV and surrounding conventional vehicles affects the energy efficiency, which has not been fully studied in the literature.

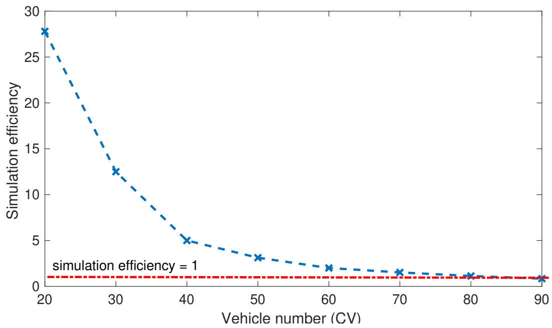
4.4. Simulation performance evaluation

In this part, we evaluate the simulation performance of our solution, including individual component (Webots, Omnet++, and SUMO) and the integrated platform. The experiments are conducted on a PC with an Intel (R) Xeon(R) E-2146G CPU @ 3.50GHz, 32GB RAM, and Quadro P620 GPU.

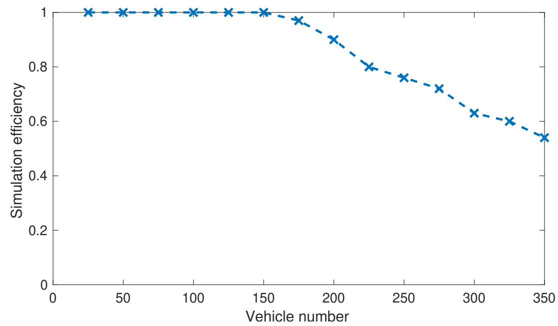
It is noted that many general metrics such as capacity, communication latency, and computation efficiency could be used to reflect the simulation performance. To simplify the performance evaluation, we consider the unique metric of ***simulation efficiency*** which is defined as the ratio of simulation runtime to physical time. Simulation efficiency essentially reflects the system capacity (e.g. communication, 3D rendering, computation, etc.) of the proposed simulation platform under the given computing resources. Normally, we expect the value of simulation efficiency is equal to or greater than 1, which indicates the time saving by utilizing simulator.

For the individual component evaluation, since Omnet++ and SUMO are physically installed in the same computer as the server, we evaluate the simulation efficiency by integrating Omnet++ with SUMO together. Specifically, we consider the heavy communication load test of CV beaconing, i.e. all CVs broadcast beacons at intervals of 0.1 seconds to the neighbors within the communication range which can be used to estimate the V2X communication capacity of the simulation platform. We change the number of testing vehicles from 50-100 vehicles, and explore its impact on simulation efficiency. Fig. 14(a) shows that with the increase of the number of the CVs, the simulation efficiency drops dramatically. The maximum number of CVs supported in Omnet++ is around 85 at the simulation efficiency equal to 1.

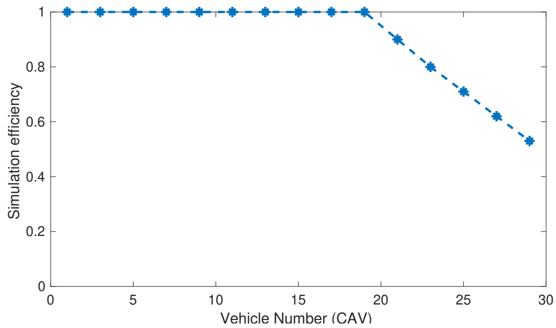
To evaluate the visualization performance (e.g. 3D rendering) of Webots, we consider a simple scenario displayed in Webots that a large number of ***abstract*** vehicles (3D shape) created by SUMO run on a 3D road. To this end, all vehicles' state information should be periodically synchronized between Webots and



(a) V2X Communication capacity (Omnet++)



(b) 3D Visualization performance (Webots)



(c) The whole system performance

Figure 14: Simulation efficiency evaluation for CCAD platform

SUMO. We let Webots work in the Real-Time running mode, i.e. simulation speed should be close to the speed of real-world phenomena. This is because we expect the simulation platform to work as a typical driving simulator which requires simulation running in physical time. We set the Webots's time-step with 50ms, and then change the number of testing vehicles. In Fig. 14(b), we can observe that the Webots can support up to 200 abstract vehicles running in physical time. Therefore, under the given PC environment, Webots can support a complicated traffic scenario with large number of vehicles.

Finally, we evaluate the simulation efficiency of the whole integrated platform by implementing case study 1 with the different number of CAVs, which requires simultaneous information synchronization among SUMO, Omnet++, and Webots. Each CAV has to implement its control algorithm independently. The time-step for each component is set as: SUMO @ 100ms, Omnet++ @ 100ms, and Webots @ 50ms. The simulation results in Fig. 14(c) indicate that the simulation platform can support about 20 fully controlled CAVs in the same simulation scenario.

Based on the preliminary analysis of the simulation performance, we can see that the capacity bottleneck of the platform is the heavy communication load among three simulators (i.e., SUMO, Omnet++, and Webots) which is implemented via TraCI API. In addition, it is noted that in the current version, the core (microscopic) simulation of SUMO can only run on a single core. To fully utilize the powerful multi-core capacity of computer resource, the work to support multi-node parallelization of SUMO is still ongoing, according to the latest release from SUMO official website.

5. Conclusion

With the upcoming massive applications of CAVs in practice, a full evaluation of their impacts on transportation is becoming an urgent request from not only manufactures but also policy-makers. In this paper, we design an integrated simulation platform for conventional, connected and automated driving from the cyber-physical system perspective, in which the core components of V2X communication, traffic networks, and autonomous/conventional vehicle model are functionally combined. Case studies demonstrate the capability and scalability of the proposed simulation platform in CCAD evaluation from the individual level to the large-scale network level. This platform can be served as an ideal testbed not only for research verification in the area of intelligent transport, but also for policy makers or authorities to plan for optimal deployment of CAVs in the transport systems and identify appropriate transport management strategies for a smooth transition from lower vehicle automation to higher traffic automation.

The current work synchronizes information among three simulators (i.e., SUMO, Omnet++, and Webots) via TraCI API, which, however, may bring heavy communication loads especially in the case of massive CAVs among CCAD. In the future, the usage of the Libsumo (DLR, 2020) functionality will be implemented to reduce the need of socket communication with TraCI. In addition, data-driven driving models will be further developed in our future work.

Acknowledgments

The authors would like to thank the University of Queensland for partially sponsoring this study through Dr Zuduo Zheng's startup package.

References

References

Alam, A.A., Gattami, A., Johansson, K.H., 2010. An experimental study on the fuel reduction potential of heavy duty vehicle platooning. 13th International IEEE Conference on Intelligent Transportation Systems , 306–311.

Ali, Y., Sharma, A., Haque, M.M., Zheng, Z., Saifuzzaman, M., 2020. The impact of the connected environment on driving behavior and safety: A driving simulator study. Accident Analysis & Prevention 144, 105643.

Baltodano, S., Sibi, S., Martelaro, N., Gowda, N., Ju, W., 2015. The rrads platform: a real road autonomous driving simulator, in: Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications, pp. 281–288.

Barceló, J., Casas, J., 2005. Dynamic network simulation with aimsun, in: Simulation approaches in transportation analysis, pp. 57–98.

Barceló, J., et al., 2010. Fundamentals of traffic simulation. volume 145. Springer.

Behrisch, M., Bieker, L., Erdmann, J., Krajzewicz, D., 2011. Sumo–simulation of urban mobility: an overview, in: Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation.

Bhadani, R.K., Sprinkle, J., Bunting, M., 2018. The cat vehicle testbed: A simulator with hardware in the loop for autonomous vehicle applications. arXiv preprint arXiv:1804.04347 .

Chen, D., Ahn, S., Laval, J., Zheng, Z., 2014. On the periodicity of traffic oscillations and capacity drop: the role of driver characteristics. Transportation research part B: methodological 59, 117–136.

Cottignies, A., Daley, M., Newton, E., Parker, H., 2017. rfpro & sumo: The road to a complete real-time simulation of urban environments for dil, adas and autonomous testing. SUMO 2017–Towards Simulation for Autonomous Mobility 31, 62.

DLR, 2020. https://sumo.dlr.de/wiki/libsumo.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V., 2017. Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938 .

Dupuy, S., Egges, A., Legendre, V., Nugues, P., 2001. Generating a 3d simulation of a car accident from a written description in natural language: The carsim system, in: Proceedings of the workshop on Temporal and spatial information processing-Volume 13, p. 1.

Fagnant, D.J., Kockelman, K., 2015. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice 77, 167–181.

Fellendorf, M., Vortisch, P., 2010. Microscopic traffic flow simulator vissim, in: Fundamentals of traffic simulation, pp. 63–93.

Google, 2015. https://www.theguardian.com/technology/2015/jul/17/crash-self-driving-google-car-injures-three.

Hussain, R., Zeadally, S., 2018. Autonomous cars: Research results, issues, and future challenges. IEEE Communications Surveys & Tutorials 21, 1275–1313.

Issariyakul, T., Hossain, E., 2009. Introduction to network simulator 2 (ns2), in: Introduction to network simulator NS2, pp. 1–18.

Jia, D., Lu, K., Wang, J., Zhang, X., Shen, X., 2016. A survey on platoon-based vehicular cyber-physical systems. IEEE Communications Surveys and Tutorials 18, 263–284.

Jia, D., Ngoduy, D., 2016. Platoon based Cooperative Driving Model with Consideration of Realistic Inter-vehicle Communication. Transportation Research Part C 68, 245–264.

Jia, D., Ngoduy, D., Vu, H., 2019. A multiclass microscopic model for heterogeneous platoon with vehicle-to-vehicle communication. Transportmetrica B: Transport Dynamics 7, 448–472.

Jiang, R., Wu, Q., Zhu, Z., 2001. Full velocity difference model for a car-following theory. Physical Review E 64, 017101.
Jin, I.G., Orosz, G., 2014. Dynamics of connected vehicle systems with delayed acceleration feedback. Transportation Research Part C: Emerging Technologies 46, 46–64.
Kahneman, D., Tversky, A., 1979. Prospect theory: An analysis of decision under risk. Econometrica 47, 363–391.
Kaths, J., Schott, B., Chucholowski, F., 2019. Co-simulation of the virtual vehicle in virtual traffic considering tactical driver decisions. EPiC Series in Computing 62, 21–28.
Koenig, N., Howard, A., 2004. Design and use paradigms for gazebo, an open-source multi-robot simulator, in: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566), IEEE. pp. 2149–2154.
Krajzewicz, D., Behrisch, M., Wagner, P., Luz, R., Krumnow, M., 2015. Second generation of pollutant emission models for sumo, in: Modeling mobility with open data. Springer, pp. 203–221.
Krajzewicz, D., Bieker, L., Härri, J., Blokpoel, R., 2012. Simulation of V2X Applications with the iTETRIS System. Procedia - Social and Behavioral Sciences 48, 1482–1492.
Krauß, S., Wagner, P., Gawron, C., 1997. Metastable states in a microscopic model of traffic flow. Physical Review E 55, 5597.
Kuefler, A., Morton, J., Wheeler, T., Kochenderfer, M., 2017. Imitating driver behavior with generative adversarial networks, in: 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 204–211.
Menhour, L., d'Andréa Novel, B., Fliess, M., Gruyer, D., Mounier, H., 2017. An efficient model-free setting for longitudinal and lateral vehicle control: Validation through the interconnected pro-sivic/rtmaps prototyping platform. IEEE Transactions on Intelligent Transportation Systems 19, 461–475.
Michel, O., 2004. Cyberbotics ltd. webots$^{\mathrm{TM}}$: professional mobile robot simulation. International Journal of Advanced Robotic Systems 1, 5.
Misener, J.A., Shladover, S.E., 2006. Path investigations in vehicle-roadside cooperation and safety: A foundation for safety and vehicle-infrastructure integration research, in: 2006 IEEE Intelligent Transportation Systems Conference, pp. 9–16.
Orosz, G., Wilson, R.E., Stépán, G., 2010. Traffic jams: dynamics and control.
Osorio, C., Nanduri, K., 2015. Energy-Efficient Urban Traffic Management: A Microscopic Simulation-Based Approach. Transportation Science 49, 637–651.
Piorkowski, M., Raya, M., Lugo, A.L., Papadimitratos, P., Grossglauser, M., Hubaux, J.P., 2008. Trans: realistic joint traffic and network simulator for vanets. ACM SIGMOBILE mobile computing and communications review 12, 31–33.
Quiter, C., Ernst, M., 2018. deepdrive/deepdrive: 2.0. URL: `https://doi.org/10.5281/zenodo.1248998`, doi:`10.5281/zenodo.1248998`.
Riley, G.F., Henderson, T.R., 2010. The ns-3 network simulator, in: Modeling and tools for network simulation. Springer, pp. 15–34.
Rosique, F., Navarro, P.J., Fernández, C., Padilla, A., 2019. A systematic review of perception system and simulators for autonomous vehicles research. Sensors 19, 648.
Saifuzzaman, M., Zheng, Z., Haque, M.M., Washington, S., 2017. Understanding the mechanism of traffic hysteresis and traffic oscillations through the change in task difficulty level. Transportation Research Part B: Methodological 105, 523–538.
Sciarretta, A., Guzzella, L., 2007. Control of hybrid electric vehicles. IEEE Control Systems Magazine 27, 60–70.
Segata, M., Joerer, S., Bloessl, B., Sommer, C., Dressler, F., Cigno, R.L., 2014. Plexe: A platooning extension for veins, in: 2014 IEEE Vehicular Networking Conference (VNC), pp. 53–60.
Shah, S., Dey, D., Lovett, C., Kapoor, A., 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles, in: Field and service robotics, pp. 621–635.
Sharma, A., Zheng, Z., Bhaskar, A., 2019a. Is more always better? the impact of vehicular trajectory completeness on car-following model calibration and validation. Transportation research part B: methodological 120, 49–75.
Sharma, A., Zheng, Z., Bhaskar, A., Haque, M.M., 2019b. Modelling car-following behaviour of connected vehicles with a focus on driver compliance. Transportation Research Part B: Methodological 126, 256–279.
Smith, M., Barton, M., Bass, M., Branschofsky, M., McClellan, G., Stuve, D., Tansley, R., Walker, J.H., 2003. Dspace: An open source dynamic digital repository .
Smith, R., et al., 2005. Open dynamics engine .
Sommer, C., German, R., Dressler, F., 2010. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. Mobile Computing, IEEE Transactions on 10, 3–15.
Sommer, C., Tonguz, O.K., Dressler, F., 2011. Traffic information systems: efficient message dissemination via adaptive beaconing. IEEE Communications Magazine 49, 173–179.
Sovani, S., 2017. Simulation accelerates development of autonomous driving. ATZ worldwide 119, 24–29.
Stahlmann, R., Festag, A., Tomatis, A., Radusch, I., Fischer, F., 2011. Starting european field tests for car-2-x communication: the drive c2x framework, in: 18th ITS World Congress and Exhibition, p. 12.
Sun, J., Ma, Z., Li, T., Niu, D., 2015. Development and application of an integrated traffic simulation and multi-driving simulators. Simulation Modelling Practice and Theory 59, 1–17.
Sun, J., Zheng, Z., Sun, J., 2018. Stability analysis methods and their applicability to car-following models in conventional and connected environments. Transportation research part B: methodological 109, 212–237.
Tang, T., Li, J., Huang, H., Yang, X., 2014. A car-following model with real-time road conditions and numerical tests. Measurement 48, 63–76.
Tideman, M., Van Noort, M., 2013. A simulation tool suite for developing connected vehicle systems, in: 2013 IEEE Intelligent Vehicles Symposium (IV), pp. 713–718.
Treiber, M., Hennecke, A., Helbing, D., 2000. Congested traffic states in empirical observations and microscopic simulations. Physical review E 62, 1805.
Treiber, M., Kesting, A., 2013. Traffic flow dynamics. Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag

Berlin Heidelberg .
Treiber, M., Kesting, A., 2018. The intelligent driver model with stochasticity-new insights into traffic flow oscillations. Transportation Research Part B: Methodological 117, 613–623.
Treiber, M., Kesting, A., Helbing, D., 2006. Delays, inaccuracies and anticipation in microscopic traffic models. Physica A: Statistical Mechanics and its Applications 360, 71–88.
Varga, A., 2010. Omnet++, in: Modeling and tools for network simulation, pp. 35–59.
Wang, M., Daamen, W., Hoogendoorn, S.P., van Arem, B., 2014. Rolling horizon control framework for driver assistance systems. Part II: Cooperative sensing and cooperative control. Transportation Research Part C: Emerging Technologies 40, 290–311.
Wilson, R.E., Ward, J.A., 2011. Car-following models: fifty years of linear stability analysis–a mathematical perspective. Transportation Planning and Technology 34, 3–18.
Wittenburg, J., 2007. Dynamics of multibody systems. Springer Science & Business Media.
Wu, C., Kreidieh, A., Parvate, K., Vinitsky, E., Bayen, A.M., 2017. Flow: Architecture and benchmarking for reinforcement learning in traffic control. arXiv preprint arXiv:1710.05465 .
Zhao, J., Wu, H., Chang, C., 2019. Virtual traffic simulator for connected and automated vehicles. Technical Report. SAE Technical Paper.
Zheng, Z., 2014. Recent developments and research needs in modeling lane changing. Transportation Research Part B 60, 16–32.
Zheng, Z., Ahn, S., Chen, D., Laval, J., 2011. Freeway traffic oscillations: microscopic analysis of formations and propagations using wavelet transform. Transportation research part B: methodological 45, 1378–1388.
Zheng, Z., Ahn, S., Monsere, C.M., 2010. Impact of traffic oscillations on freeway crash occurrences. Accident Analysis & Prevention 42, 626–636.
Zhu, W.X., Zhang, H., 2018. Analysis of mixed traffic flow with human-driving and autonomous cars based on car-following model. Physica A: Statistical Mechanics and its Applications 496, 274–285.