Article

Longitudinal Control for Connected and Automated Vehicles in Contested Environments

Shirin Noei ^{1,*} , Mohammadreza Parvizimosaei ^{2,†} and Mohammadreza Noei ^{3,†} ¹ Center for Energy Systems Research, Tennessee Technological University, Cookeville, TN 38505, USA² Department of Computer Engineering, K. N. Toosi University of Technology, Tehran 16317-14191, Iran; rezaparvizi@email.kntu.ac.ir³ Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran 14117-13116, Iran; mohammadrezanoei@modares.ac.ir

* Correspondence: snoei@tntech.edu; Tel.: +1-931-372-6546

† These authors contributed equally to this work.

Abstract: The Society of Automotive Engineers (SAE) defines six levels of driving automation, ranging from Level 0 to Level 5. Automated driving systems perform entire dynamic driving tasks for Levels 3–5 automated vehicles. Delegating dynamic driving tasks from driver to automated driving systems can eliminate crashes attributed to driver errors. Sharing status, sharing intent, seeking agreement, or sharing prescriptive information between road users and vehicles dedicated to automated driving systems can further enhance dynamic driving task performance, safety, and traffic operations. Extensive simulation is required to reduce operating costs and achieve an acceptable risk level before testing cooperative automated driving systems in laboratory environments, test tracks, or public roads. Cooperative automated driving systems can be simulated using a vehicle dynamics simulation tool (e.g., CarMaker and CarSim) or a traffic microsimulation tool (e.g., Vissim and Aimsun). Vehicle dynamics simulation tools are mainly used for verification and validation purposes on a small scale, while traffic microsimulation tools are mainly used for verification purposes on a large scale. Vehicle dynamics simulation tools can simulate longitudinal, lateral, and vertical dynamics for only a few vehicles in each scenario (e.g., up to ten vehicles in CarMaker and up to twenty vehicles in CarSim). Conventional traffic microsimulation tools can simulate vehicle-following, lane-changing, and gap-acceptance behaviors for many vehicles in each scenario without simulating vehicle powertrain. Vehicle dynamics simulation tools are more compute-intensive but more accurate than traffic microsimulation tools. Due to software architecture or computing power limitations, simplifying assumptions underlying conventional traffic microsimulation tools may have been a necessary compromise long ago. There is, therefore, a need for a simulation tool to optimize computational complexity and accuracy to simulate many vehicles in each scenario with reasonable accuracy. This research proposes a traffic microsimulation tool that employs a simplified vehicle powertrain model and a model-based fault detection method to simulate many vehicles with reasonable accuracy at each simulation time step under noise and unknown inputs. Our traffic microsimulation tool considers driver characteristics, vehicle model, grade, pavement conditions, operating mode, vehicle-to-vehicle communication vulnerabilities, and traffic conditions to estimate longitudinal control variables with reasonable accuracy at each simulation time step for many conventional vehicles, vehicles dedicated to automated driving systems, and vehicles equipped with cooperative automated driving systems. Proposed vehicle-following model and longitudinal control functions are verified for fourteen vehicle models, operating in manual, automated, and cooperative automated modes over two driving schedules under three malicious fault magnitudes on transmitted accelerations.

Keywords: traffic microsimulation tool; cooperative automated driving systems; vehicle powertrain; safety; road capacity; contested environments

Citation: Noei, S.; Parvizimosaei, M.; Noei, M. Longitudinal Control for Connected and Automated Vehicles in Contested Environments.

Electronics **2021**, *10*, 1994. <https://doi.org/10.3390/electronics10161994>

Academic Editor: Alexey Vinel

Received: 15 June 2021

Accepted: 9 August 2021

Published: 18 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

SAE defines six levels of driving automation

- Level 0: drivers perform entire dynamic driving tasks;
- Level 1: driver assistance systems execute either longitudinal or lateral vehicle motion control subtask, and drivers perform all remaining dynamic driving tasks;
- Level 2: driver assistance systems execute both longitudinal and lateral vehicle motion control subtasks, and drivers perform all remaining dynamic driving tasks;
- Levels 3–5: automated driving systems perform entire dynamic driving tasks [1].

Dynamic driving tasks are real-time operational (e.g., longitudinal and lateral vehicle motion control) and tactical (e.g., object and event detection, recognition, classification, and response preparation) functions required to operate a vehicle. Delegating dynamic driving tasks to automated driving systems can eliminate 94% of crashes attributed to driver errors [2].

Cooperative driving automation enables cooperation among road users, intending to enhance dynamic driving task performance, safety, and traffic operations. Cooperative driving automation can prevent 439,000 to 615,000 crashes, save 987 to 1366 lives, reduce 305,000 to 418,000 maximum abbreviated injury scale 1–5 injuries, and eliminate 537,000 to 746,000 property damage only vehicles annually [3]. Vehicles equipped with cooperative automated driving systems can also follow their leaders at shorter gaps and with less variation in acceleration than vehicles dedicated to automated driving systems. SAE defines four classes of cooperative driving automation cooperation: Class A (status-sharing), Class B (intent-sharing), Class C (seeking-agreement), and Class D (prescriptive) [4]. Classes C–D cooperative driving automation cooperation can be achieved at Levels 3–5 driving automation.

Cooperative automated driving systems can be simulated using a vehicle dynamics simulation tool (e.g., CarMaker and CarSim) or a traffic microsimulation tool (e.g., Vissim and Aimsun). Vehicle dynamics simulation tools are mainly used to simulate longitudinal, lateral, and vertical dynamics on a small scale, while traffic microsimulation tools are mainly used to simulate vehicle-following, lane-changing, and gap-acceptance behaviors on a large scale.

- **Verification scale:** Vehicle dynamics simulation tools cannot simulate many vehicles in each scenario.
- **Verification resolution:** Conventional traffic microsimulation tools cannot estimate microscopic (e.g., reduction in distance gaps and time gaps) or macroscopic (e.g., increase in road capacity) benefits associated with driving automation or cooperative driving automation with reasonable accuracy;
- **Vehicle powertrain (i.e., engine, transmission, and driveline):** Conventional traffic microsimulation tools do not simulate vehicle powertrain;
- **Maximum acceleration and maximum deceleration:** Conventional traffic microsimulation tools estimate or use constant maximum accelerations and maximum decelerations. Aimsun considers maximum acceleration of 8.2 ft/s^2 and maximum deceleration of 6.6 ft/s^2 as default [5]. Vissim estimates maximum acceleration as $a_{max}(t) \approx 3.5(1 - v(t)/40)$ and maximum deceleration as $d_{max}(t) \approx 20(1 - v(t)/800)$, where a_{max} is maximum acceleration (m/s^2), v is speed (m/s), and d_{max} is maximum deceleration (m/s^2) [5]—since all units in Vissim User Manual are metric, metric units are preferred to report these regression models with full precision. However, maximum acceleration and maximum deceleration are sensitive to vehicle model, grade, pavement conditions, and traffic conditions;

- **Longitudinal control variables:** Conventional longitudinal control functions (e.g., Adaptive Cruise Control (ACC), Cooperative Adaptive Cruise Control (CACC)) rely on constant distance gaps, time gaps, and controller coefficients, potentially sacrificing safety (i.e., when short gaps are set) or reducing road capacity (i.e., when long gaps are set). Conventional traffic microsimulation tools rely on user inputs for distance gap, time gap, and longitudinal controller coefficients (e.g., proportional, integral, and derivative) to simulate vehicles in a platoon or string. However, distance gap, time gap, and longitudinal controller coefficients are sensitive to driver characteristics, vehicle model, grade, pavement conditions, operating mode, malicious fault magnitude, and traffic conditions.
- **Contested environments:** Onboard sensor measurements and transmitted messages are inherently prone to noise, natural fault, and malicious fault. Minor faults may lead to malfunction or even failure if not responded promptly. A single cyberattack can cost an average original equipment manufacturer \$1.6 billion a year, assuming one individual recall costs \$800 [6]. From 2010 to 2021, 367 cyberattacks on connected vehicles have been reported [6].

A cyberattack can exploit one user application's vulnerabilities (e.g., spoofing, data falsification, and replay attacks) or multiple user application vulnerabilities (e.g., denial-of-service attack), leading to severe consequences for vehicle and potentially its operating environment [7]. Spoofing, data falsification, replay, and denial-of-service attacks are common cyberattacks on connected vehicles [8]. Spoofing attack is when hackers steal authentication credentials or use a legitimate vehicle's identity to send unchanged or manipulated messages to other vehicles; data falsification attack is when hackers read, insert, or modify transmitted messages; replay attack is when hackers copy a message stream between two vehicles and repeat that stream to other vehicles; denial-of-service attack is when hackers prevent or interfere with target vehicles from receiving specific messages.

Conventional fault detection methods are broadly classified into model-driven and data-driven methods [9]. Model-driven methods (e.g., unknown input observer and Kalman filter) require partial plant model; data-driven methods (e.g., neural network) require measured inputs and outputs under normal and faulty conditions to derive plant model. Model-driven methods are more computationally intensive but more accurate than data-driven methods [10].

Conventional traffic microsimulation tools do not simulate contested environments. A simple strategy is to rely on onboard sensor measurements when there is a significant discrepancy between onboard sensor measurements and transmitted messages [11].

Our traffic microsimulation tool is superior to vehicle dynamics simulation tools and conventional traffic microsimulation tools because it can achieve these objectives

- **Verification scale:** simulate many vehicles in each scenario;
- **Verification resolution:** estimate microscopic and macroscopic benefits associated with driving automation and cooperative driving automation with reasonable accuracy [12,13];
- **Vehicle powertrain:** simulate vehicle powertrain;
- **Maximum acceleration and maximum deceleration:** estimate maximum acceleration and maximum deceleration with reasonable accuracy at each simulation time step, considering vehicle model, grade, pavement conditions, and traffic conditions;
- **Distance gap and time gap:** estimate minimum safe distance gap and minimum safe time gap with reasonable accuracy at each simulation time step for vehicles dedicated to automated driving systems or equipped with cooperative automated driving systems, considering vehicle model, grade, pavement conditions, operating mode, vehicle-to-vehicle communication vulnerabilities, and traffic conditions;

- **Longitudinal controller coefficients:** estimate longitudinal controller coefficients (i.e., proportional, integral, and derivative gains) with reasonable accuracy at each simulation time step for vehicles dedicated to automated driving systems, considering vehicle model, grade, pavement conditions, and traffic conditions;
- **Contested environments:** employ a reduced-order Kalman filter unknown input observer to estimate distance gap, speed, and acceleration with reasonable accuracy at each simulation time step for vehicles dedicated to automated driving systems or equipped with cooperative automated driving systems under noise (e.g., measurement noise and process noise) and unknown inputs (e.g., noise with unknown statistics, natural fault, and malicious fault).

2. Literature Review

Longitudinal control variables are mostly treated as constant parameters (see Tables 1 and 2) or variables estimated using empirical or simplified mechanistic models. However, maximum acceleration, maximum deceleration, minimum safe distance gap, and minimum safe time gap are sensitive to driver characteristics, vehicle model, grade, pavement conditions, operating mode, and traffic conditions.

Akçelik and Besley (2001) empirically estimated maximum acceleration and maximum deceleration based on initial speed and final speed for passenger cars, and based on initial speed, final speed, power-to-weight ratio, and grade for trucks [14]. Ahn et al. (2002) generated a lookup table to identify maximum acceleration over 17 driving schedules (see Table 3) [15]. Fang and Elefteriadou (2005) recommended a maximum acceleration and a maximum deceleration for each vehicle classification (i.e., passenger car and truck), interchange configuration (i.e., Single-Point Urban Interchange (SPUI) and diamond), and traffic microsimulation tool (i.e., Vissim, Aimsun, and CORSIM) (see Table 4) [16]. Kuriyama et al. (2010) considered aerodynamic resistance, rolling resistance, and grade resistance in calculating acceleration and deceleration for electric vehicles [17]. Maurya and Bokare (2012) generated a lookup table to identify maximum deceleration for each vehicle classification at each speed range (see Table 5) [18]. Lee et al. (2013) considered a higher maximum acceleration (13.1 ft/s^2 vs. 10.0 ft/s^2) and a lower maximum deceleration (9.8 ft/s^2 vs. 15.0 ft/s^2) for connected vehicles than the Federal Highway Administration's recommended maximum acceleration and maximum deceleration [19]. Anya et al. (2014) believed vehicle-following, lane-changing, travel time, and queue discharge had an impact on maximum acceleration and maximum deceleration [20]. Song et al. (2015) empirically estimated maximum acceleration based on speed [21]. Bokare and Maurya (2017) generated two lookup tables to identify maximum acceleration and maximum deceleration for each vehicle classification (i.e., diesel car, petrol car, and truck) at each speed range (see Table 6) [22]. Ramezani et al. (2018) generated a lookup table to identify maximum acceleration for trucks in CACC mode at each speed range (see Table 7) [23].

Shladover et al. (2010) identified that drivers maintain 2.2 s, 1.6 s, and 1.1 s time gaps for 31.1%, 18.5%, and 50.4% of their vehicle-following time in ACC mode, respectively, and drivers maintain 0.6 s, 0.7 s, 0.9 s, and 1.1 s time gaps for 57%, 24%, 7%, and 12% of their vehicle-following time in CACC mode, respectively [24]. Willigen et al. (2011) recommended a distance headway and a time headway for each platoon size (i.e., 20 and 30) and operating mode (i.e., ACC, CACC with transmitted accelerations, and CACC with estimated accelerations) (see Table 8) [25]. Horiguchi and Oguchi (2014) calculated distance gap for vehicles in CACC mode based on minimum safe distance gap, follower's speed, leader's speed, maximum acceleration, and maximum deceleration [26]. Flores et al. (2017) calculated time gap based on minimum safe time gap, desired time gap, and speed, and calculated distance gap based on actuator delay, speed, maximum deceleration, and maximum jerk [27]. Askari et al. (2017) calculated distance gap based on minimum safe distance gap, follower's speed, reaction time, leader's speed, maximum acceleration, and maximum deceleration [28]. Flores and Milanés (2018) recommended a time gap for each controller type (i.e., fractional-order proportional derivative and integer-order proportional

derivative), desired performance (i.e., ensuring loop bandwidth, phase margin, and string stability), and operating mode (i.e., ACC and CACC) (see Table 9) [29]. Chen et al. (2019) calculated time gap for vehicles in ACC and CACC modes based on jam density, free-flow speed, follower's speed, follower's acceleration, and leader's acceleration [30]. Bian et al. (2019) recommended a time headway for each platoon size (i.e., 1, 3, 10, 20, and 30) and controller type (i.e., linear, nonlinear, and nonlinear subject to communication delay) (see Table 10) [31].

Conventional traffic microsimulation tools (1) should be integrated with a vehicle dynamics simulation tool to simulate vehicle powertrain [32], (2) employ kinematics to estimate quantities associated with motion [33], (3) automatically confine accelerations and decelerations to constant (e.g., Aimsun and MITSIM) or estimated (e.g., Vissim and INTEGRATION) maximum accelerations and maximum decelerations, and (4) rely on constant distance gaps and time gaps to simulate longitudinal control for automated vehicles in a platoon or string. This research proposes a traffic microsimulation tool that can estimate maximum acceleration, maximum deceleration, minimum safe distance gap, and minimum safe time gap with reasonable accuracy at each simulation time step for conventional vehicles, vehicles dedicated to automated driving systems, and vehicle equipped with cooperative automated driving systems, considering driver characteristics (see Section 3.1), vehicle model (see Section 3.2), pavement conditions (see Section 3.2.3), grade (see Section 3.2.3), operating mode (see Section 3.5), traffic conditions (see Section 3.2.3), and vehicle-to-vehicle communication vulnerabilities (see Section 4).

Table 1. Constant distance gaps and time gaps used in literature.

Author	Distance Gap (ft)	Time Gap (s)	
		ACC	CACC
Bu et al. (2010) [34]	-	[1.1–2.2]	[0.6–1.1]
Naus et al. (2010) [35]	-	2.6 *	0.8 *
Shladover et al. (2010); Liu (2018) [24,36]	-	1.1, 1.6, 2.2	0.7, 0.9, 1.1
Ploeg et al. (2011) [37]	-	-	0.7 *
Willigen et al. (2011) [25]	15.8 *, 25.2 *, 26.0 *, 34.5 *, 39.9 *, 57.2 *	0.5 *, 0.6 *	0.2 *, 0.3 *, 0.4 *
Shladover et al. (2012) [38]	-	-	0.5
Zhao and Sun (2013) [39]	-	1.4	0.5
Horiguchi and Oguchi (2014) [26]	-	2.0, 2.0 *	-
Segata et al. (2014) [40]	16.4	0.3 *, 1.2 *	-
Milanés and Shladover (2014) [41]	-	1.1	0.6
Nikolos et al. (2015); Delis et al. (2016) [42,43]	-	1.2	1.0
Wang et al. (2017) [44]	42.7, 46.9, 68.2	-	0.4, 0.5, 0.6 *, 0.7, 0.9 *
Terruzzi et al. (2017) [45]	16.4 *	1.4 *	1.0 *
Zhou et al. (2017) [46]	16.4	0.5, 1.0, 1.5, 2.0	0.5, 1.0, 1.5, 2.0
Askari et al. (2017) [28]	9.8, 13.1	1.1	0.8
Flores and Milanés (2018) [29]	-	0.5, 0.6	0.3
Chen et al. (2019) [30]	-	[0.2–2.2]	-
Bian et al. (2019) [31]	-	-	0.0 *, 0.1 *, 0.2 *, 0.3 *, 0.4 *, 0.6 *, 0.9 *, 1.0 *, 1.1 *

* headway.

Table 2. Constant maximum accelerations and maximum decelerations used in literature.

Author	Max. Acceleration (ft/s ²)	Max. Deceleration (ft/s ²)
Akçelik and Besley (2001) [14]	8.8	10.1
Lemessi (2001) [47]	8.2	8.2
Ahn et al. (2002) [15]	3.4, 3.9, 4.8, 5.0, 5.4, 5.5, 7.3, 7.4, 7.7, 8.3, 8.5, 8.7, 9.3, 10.1	-
Rakha and Ding (2003) [48]	4.9	8.2
Wang and Liu (2005) [49]	8.2	11.5
Fang and Elefteriadou (2005) [16]	4.9, 6.9, 8.2, 9.2, 11.5, 15.1	9.8, 12.1, 15.1
Ossen et al. (2006) [50]	26.2	26.2
Kesting et al. (2007) [51]	4.9	13.1
Kesting and Treiber (2008) [52]	19.7	19.7
Kuriyama et al. (2010) [17]	8.8	9.8
Talebpoor et al. (2011) [53]	13.1	26.2
Song et al. (2012) [54]	11.5	13.1
Shladover et al. (2012) [38]	6.6	6.6
Lee and Park (2012); Lee et al. (2013) [19,55]	13.1	9.8
Maurya and Bokare (2012) [18]	-	2.4, 2.5, 2.9, 5.0, 5.1, 5.3
Treiber and Kesting (2013) [56]	1.7, 4.6, 4.8	2.1, 4.8
Anyia et al. (2014) [20]	0.7, 1.5, 8.5, 9.8, 11.2, 19.1, 22.0, 25.0	1.6, 3.7, 16.4, 19.7, 23.0, 36.7, 44.0, 51.5
Li et al. (2014) [57]	4.5	11.0
Tang et al. (2014) [58]	-	19.7
Desiraju et al. (2014); Liu et al. (2018) [59,60]	6.6	-
Horiguchi and Oguchi (2014) [26]	5.2	-
Song et al. (2015) [21]	8.8	-
Amoozadeh et al. (2015); Zhou et al. (2017) [46,61]	9.8	16.4
Bokare and Maurya (2017) [22]	2.5, 2.9, 3.1, 3.3, 6.2, 6.5, 7.3, 8.1, 9.4	2.4, 2.5, 2.9, 11.0, 13.0, 14.1, 14.2, 14.8, 16.4
Askari et al. (2017) [28]	2.6, 4.9, 8.2	6.6
Li et al. (2017) [62]	3.3	-
Ramezani et al. (2018) [23]	0.4, 0.5, 0.8, 1.3, 1.6, 1.8, 8.2	9.8

Table 3. Maximum acceleration vs. driving schedule.

Driving Schedule	Max. Acceleration (ft/s ²)
Freeway, High Speed	3.9
Freeway, LOS * A–C	5.0
Freeway, LOS * D	3.4
Freeway, LOS * E	7.7
Freeway, LOS * F & LA92	10.1
Freeway, LOS * G	5.5
Freeway Ramps & Arterial/Collectors LOS * C–D	8.3
Arterial/Collectors LOS * A–B	7.3
Arterial/Collectors LOS * E–F	8.5
Local Roadways	5.4
Non-Freeway Area-Wide Urban Travel	9.3
LA4 & Running 505	4.8
ST01	7.4
New York City Cycle	8.7

* Level of Service.

Table 4. Maximum acceleration and maximum deceleration vs. traffic microsimulation tool, vehicle classification, and interchange configuration.

Specification	Max. Acceleration (ft/s ²)	Max. Deceleration (ft/s ²)
Passenger Car, SPUI, Vissim	11.5	-
Passenger Car, Diamond, Vissim & Aimsun	6.9	-
Truck, SPUI, Vissim	8.2	-
Truck, Diamond, Vissim	4.9	-
Passenger Car, SPUI, Aimsun	9.2	-
SPUI, CORSIM	15.1	9.8
Diamond, CORSIM	6.9	15.1

Table 5. Maximum deceleration vs. vehicle classification.

Vehicle Classification	Speed Range (ft/s)	Max. Deceleration (ft/s ²)
Passenger Car	[83.8–85.7)	5.0
Passenger Car	[85.7–87.5)	5.1
Passenger Car	[87.5–91.1]	5.3
Truck	[18.2–27.3)	2.4
Truck	[27.3–36.5)	2.5
Truck	[36.5–54.7]	2.9

Table 6. Maximum acceleration and maximum deceleration vs. vehicle classification.

Vehicle Classification	Speed Range (ft/s)	Max. Acceleration (ft/s ²)	Max. Deceleration (ft/s ²)
Diesel Car	[62.0–69.3)	6.2	-
Diesel Car	[69.3–76.6)	7.3	-
Diesel Car	[76.6–83.8)	6.5	-
Diesel Car	[83.8–85.7)	-	14.1
Diesel Car	[85.7–87.5)	-	14.2
Diesel Car	[87.5–89.3)	-	16.4
Diesel Car	[89.3–91.1]	-	14.8
Petrol Car	[55.6–65.6)	-	11.0
Petrol Car	[65.6–75.6)	-	13.0
Petrol Car	[72.9–76.6)	7.3	-
Petrol Car	[75.6–82.9]	-	14.2
Petrol Car	[76.6–80.2)	8.1	-
Petrol Car	[80.2–83.8]	9.4	-
Truck	[18.2–27.3)	2.5	2.4
Truck	[27.3–36.5)	3.3	2.5
Truck	[36.5–45.6)	3.1	2.9
Truck	[45.6–54.7]	2.9	2.9

Table 7. Maximum acceleration.

Speed Range (ft/s)	Max. Acceleration (ft/s ²)
[0–14.7)	1.8
[29.3–44.0)	1.3
[44.0–58.7)	0.8
[58.7–73.3)	0.5
Above 73.3	0.4

Table 8. Distance headway and time headway vs. platoon size and operating mode.

Specification	Distance Headway (ft)	Time Headway (s)
20, ACC	34.5	0.5
20, CACC with Transmitted Accelerations	26.0	0.4
20, CACC with Estimated Accelerations	15.8	0.2
30, ACC	57.2	0.6
30, CACC with Transmitted Accelerations	39.9	0.4
30, CACC with Estimated Accelerations	25.2	0.3

Table 9. Time gap vs. controller type, desired performance, and operating mode.

Specification	Time Gap (s)
Fractional-Order Proportional Derivative, ACC	0.5
Integer-Order Proportional Derivative, Loop Bandwidth and Phase Margin, ACC	0.6
Integer-Order Proportional Derivative, Loop Bandwidth and String Stability, ACC	0.5
CACC	0.3

Table 10. Time headway vs. platoon size and controller type.

Specification	Time Headway (s)
1, Linear	0.3, 0.4, 0.6
3, Linear	0.1, 0.2
1, Nonlinear	0.6
3, Nonlinear	0.2
10, Nonlinear	0.1
20 & 30, Nonlinear	0.0
1, Nonlinear Subject to Communication Delay	0.9, 1.0, 1.1
3, Nonlinear Subject to Communication Delay	0.6, 0.7, 0.9, 1.1

3. Proposed Traffic Microsimulation Tool

Our traffic microsimulation tool enables users to customize driver (see Section 3.1), vehicle (see Section 3.2), road (see Section 3.3), cyberattack (see Section 3.4), and operating mode (see Section 3.5) modules separately. Our traffic microsimulation tool contains ten driver types (conservative to aggressive), fourteen vehicle models (i.e., ten passenger car models and four truck configurations), two driving schedules (i.e., US06 and Cycle D), three malicious fault magnitudes (i.e., malicious increases of 1, 3, and 5 ft/s² in transmitted accelerations), and three operating modes (i.e., cooperative automated, automated, and manual) as default to simulate many vehicles with reasonable accuracy at each simulation time step under noise and unknown inputs. Vehicles in manual mode require driver, vehicle, and road modules; vehicles in automated mode require vehicle and road modules; vehicles in cooperative automated mode require vehicle, road, and cyberattack modules to be implemented (see Figure 1).

Figure 1. Proposed traffic microsimulation tool.

3.1. Driver Module

Ten driver types are considered as default (based on an assumed value in CORSIM—a traffic microsimulation tool): type 1 is a conservative driver; type 10 is an aggressive

driver. Each driver type is associated with a speed multiplier, an acceleration multiplier, a deceleration multiplier, and a percentage included in traffic which follows a normal distribution as default [63].

3.2. Vehicle Module

Fourteen vehicle models are included as default (assumed vehicles in SwashSim—a traffic microsimulation tool): 2006 Honda Civic Si, 2008 Chevy Impala, 1998 Buick Century, 2004 Chevy Tahoe, 2002 Chevy Silverado, 1998 Chevy S10 Blazer, 2011 Ford F150, 2009 Honda Civic, 2005 Mazda 6, and 2004 Pontiac Grand Am, single-unit truck with PACCAR PX-7 engine, intermediate semi-trailer with PACCAR MX-13 engine, interstate semi-trailer with PACCAR MX-13 engine, and double semi-trailer with PACCAR MX-13 engine. Each vehicle is associated with a torque map, a drag coefficient, a width, a height, a weight, a wheelbase length, a wheel radius, a differential gear ratio, a drive axle slippage, a drivetrain efficiency, a transmission gear ratio, shift up speeds, shift down speeds, and a percentage included in traffic which follows a normal distribution as default [63]. Vehicle module contains vehicle generation, reference speed profiles, and vehicle dynamics submodules: Vehicle dispatching model in Section 3.2.1 is intended to generate many vehicles at an assumed entrance under steady-state conditions; platoon leaders are assumed to follow US06 and Cycle D driving schedules (see Section 3.2.2); maximum acceleration, maximum deceleration, minimum safe distance gap, and minimum safe time gap are estimated based on vehicle dynamics (see Section 3.2.3).

3.2.1. Vehicle Generation

Entry headways follow shifted negative-exponential distribution

$$f(h) = \begin{cases} \lambda e^{-\lambda(h-h_{min})}, & h \geq h_{min} \\ 0, & h < h_{min} \end{cases} \quad (1)$$

where f is probability density function, h is entry headway (s/veh), h_{min} is minimum entry headway (s/veh), λ is distribution parameter (veh/s) calculated as $1/(\bar{h} - h_{min})$, $\bar{h}$ is average entry headway (s/veh) calculated as $3600/q$, and q is flow rate (veh/h).

3.2.2. Reference Speed Profiles

US06 and Cycle D driving schedules are used as reference speed profiles. US06 driving schedule is designed to test passenger cars, representing an 8-mile route with average speed of 70.4 ft/s, maximum speed of 117.8 ft/s, maximum acceleration of 12.3 ft/s², maximum deceleration of 10.1 ft/s², and 600 s duration. Cycle D driving schedule is designed to test trucks, representing a 5.6-mile route with average speed of 27.6 ft/s, maximum speed of 85.1 ft/s, maximum acceleration of 6.4 ft/s², maximum deceleration of 6.8 ft/s², and 1060 s duration.

3.2.3. Vehicle Dynamics

Conventional longitudinal control functions control accelerations and decelerations using throttle and brake inputs to maintain a constant distance gap in a platoon (e.g., truck platooning) or a constant time gap in a string (e.g., ACC and CACC). Commanded accelerations and decelerations are automatically confined to maximum accelerations and maximum decelerations specific to vehicle model, grade, pavement conditions, and traffic conditions. Longitudinal controller coefficients can be tuned to achieve desired performance. Conventional traffic microsimulation tools require user inputs for maximum acceleration, maximum deceleration, distance gap, time gap, and longitudinal controller coefficients to simulate vehicles in a platoon or string.

Our traffic microsimulation tool follows these steps at each simulation time step to simulate vehicles in a platoon or string: (1) estimating maximum acceleration and maximum deceleration for each vehicle, considering vehicle model, grade, pavement

conditions, and traffic conditions, (2) estimating minimum safe distance gap and minimum safe time gap for each vehicle dedicated to automated driving systems or equipped with cooperative automated driving systems, considering vehicle model, grade, pavement conditions, operating mode, vehicle-to-vehicle communication vulnerabilities, and traffic conditions, (3) checking preset distance gaps and preset time gaps with minimum safe distance gaps and minimum safe time gaps, (4) estimating accelerations and decelerations, considering operating mode, and (5) confining accelerations and decelerations to maximum accelerations and maximum decelerations.

Three significant forces against vehicle motion are aerodynamic resistance, rolling resistance, and grade resistance. Aerodynamic resistance can be calculated as

$$R_a[k] \triangleq \frac{\rho}{2} C_D A_f v^2[k], \quad (2)$$

where R_a is aerodynamic resistance (lb), ρ is air density (slugs/ft³), C_D is drag coefficient (unitless), A_f is vehicle frontal area (ft²) calculated as vehicle width (ft) $\times$ vehicle height (ft), v is speed (ft/s), and $[k]$ denotes simulation time step. Rolling resistance can be estimated as

$$R_{rl}[k] \approx f_{rl}[k] W, \quad (3)$$

where R_{rl} is rolling resistance (lb), f_{rl} is rolling resistance coefficient (unitless) estimated as $0.01(1 + v[k]/147)$ for vehicles operating on paved surfaces [64], and W is vehicle weight (lb). Grade resistance can be calculated as

$$R_g \triangleq W \sin \theta, \quad (4)$$

where R_g is grade resistance (lb), and θ is grade (unitless). Tractive effort available to overcome resistance and to provide acceleration can be calculated as $F[k] = \min(F_{max}[k], F_e[k])$, where F is available tractive effort (lb), F_{max} is maximum tractive effort (lb), and F_e is engine-generated tractive effort (lb). Maximum tractive effort can be calculated as

$$F_{max}[k] \triangleq \begin{cases} \mu W \frac{l_r \cos \theta + h f_{rl}[k]}{L + \mu h}, & \text{front-wheel-drive} \\ \mu W \frac{l_f \cos \theta - h f_{rl}[k]}{L - \mu h}, & \text{rear-wheel-drive} \\ \mu W \cos \theta, & \text{all-wheel-drive} \end{cases} \quad (5)$$

where μ is road adhesion coefficient (unitless), l_r is distance from rear axle to gravity center (ft), h is vehicle height (ft), L is wheelbase length (ft), and l_f is distance from front axle to gravity center (ft). Engine speed can be calculated as

$$n_e[k] \triangleq \frac{v[k] \epsilon_0[k]}{2\pi r(1-i)}, \quad (6)$$

where n_e is engine speed (revs/s), ϵ_0 is overall gear reduction ratio (unitless), calculated as transmission gear ratio (unitless), selected based on vehicle speed $\times$ differential gear ratio (unitless), r is wheel radius (ft), and i is drive axle slippage (unitless). Engine power can be calculated as

$$hp_e[k] \triangleq \frac{2\pi M_e[k] n_e[k]}{550}, \quad (7)$$

where hp_e is engine power (hp), and M_e is torque (ft-lb). Engine-generated tractive effort can be calculated as

$$F_e[k] \triangleq \frac{M_e[k] \epsilon_0[k] \eta_d}{r}, \quad (8)$$

where η_d is drivetrain efficiency (unitless). Maximum braking force can be calculated as

$$B_{max}[k] \triangleq \begin{cases} \eta_b \mu W \frac{l_r \cos \theta + h f_{rl}[k]}{L - \eta_b \mu h}, & \text{front-wheel-drive} \\ \eta_b \mu W \frac{l_f \cos \theta - h f_{rl}[k]}{L + \eta_b \mu h}, & \text{rear-wheel-drive} \\ \eta_b \mu W \cos \theta, & \text{all-wheel-drive} \end{cases} \quad (9)$$

where B_{max} is maximum braking force (lb), and η_b is braking efficiency (unitless). Maximum acceleration can be estimated as

$$a_{max}[k] \approx \frac{F[k] - R_a[k] - R_{rl}[k] - R_g}{m \gamma_m[k]}, \quad (10)$$

where a_{max} is maximum acceleration (ft/s^2), and γ_m is mass factor (unitless) estimated as $1.04 + 0.0025 \epsilon_0^2[k]$ [64], accounting for rotational inertia during acceleration. Maximum deceleration can be estimated as [63]

$$d_{max}[k] \approx \frac{B_{max}[k] + R_a[k] + R_{rl}[k] + R_g}{m \gamma_b}, \quad (11)$$

where d_{max} is maximum deceleration (ft/s^2), and γ_b is mass factor (unitless), accounting for rotational inertia during deceleration. Minimum safe distance gap can be estimated as [63]

$$S_{min}[k] \approx (\tau_s^{i+1} + \tau_c^{i+1}) v_{i+1}[k] + S_{stop}^{i+1}[k] - S_{stop}^i[k], \quad (12)$$

where S_{min} is minimum safe distance gap (ft), τ_s is sensing delay (s), τ_c is communication delay (s), subscript/superscript $i + 1$ denotes follower, subscript/superscript i denotes leader, and S_{stop} is minimum stopping distance (ft) estimated as

$$S_{stop}[k] \approx \frac{m \gamma_b}{\rho C_D A_f} \ln \left(1 - \frac{R_a[k]}{B_{max}[k] + R_a[k] + R_{rl}[k] + R_g} \right). \quad (13)$$

Minimum safe time gap can be estimated as [63]

$$T_{min}[k] \approx \tau_s^{i+1} + \tau_c^{i+1} + \tau_{lag}^{i+1}[k] - \tau_{lag}^i[k], \quad (14)$$

where T_{min} is minimum safe time gap (s), and τ_{lag} is lag in tracking desired deceleration (s) estimated as $v[k]/d_{max}[k]$.

Assumption 1. Vehicles have constant speeds during sensing delay and communication delay.

Remark 1. Proposed longitudinal dynamics has been previously validated for 53,000 lb and 80,000 lb interstate semi-trailers against an industry-standard simulation tool (i.e., TruckSim) [65–67].

3.3. Road Module

Any desired freeway segment with a single lane can be simulated. Each freeway segment is associated with a grade, a road adhesion coefficient, and a free-flow speed.

3.4. Cyberattack Module

Three malicious fault magnitudes are assumed as default: 1, 3, and 5 ft/s^2 malicious increase in transmitted accelerations. Each malicious fault magnitude is associated with a percentage injected on traffic which follows a normal distribution as default.

3.5. Operating Mode Module

Three operating modes are considered as default: manual, automated, and cooperative automated. Each operating mode is associated with a percentage included in traffic which follows a normal distribution as default. This section proposes a vehicle-following model for vehicles in manual mode and longitudinal control functions for vehicles in automated and cooperative automated modes.

3.5.1. Manual Mode

Levels 1 and 2 automated vehicles are assumed to have a vehicle-following model similar to the Improved Intelligent Driver Model (IIDM)

$$a_{i+1}[k] \triangleq \begin{cases} n \times a_{max}^{i+1}[k] C_s[k] C_v[k], & S[k] \geq S_{min}[k] \\ -q \times d_{max}^{i+1}[k], & \text{else} \end{cases} \quad (15)$$

where n is acceleration multiplier (unitless), C_s is distance gap coefficient (unitless) calculated as $1 - (S_{min}[k]/S[k])^\alpha$, S is distance gap (ft) calculated as $x_i[k] - x_{i+1}[k] - L_i$, x is front bumper position (ft), C_v is speed coefficient (unitless) calculated as $1 - (m \times v_i[k]/FFS)^\beta$, m is speed multiplier (unitless), FFS is free-flow speed (ft/s), q is deceleration multiplier (unitless), and α and β are calibration parameters (unitless). IIDM has fewer calibration parameters and demonstrates a more stable performance than Wiedemann model (i.e., vehicle-following model used in Vissim) [68].

Assumption 2. *There are three significant components underpinning a traffic microsimulation tool (i.e., vehicle-following, lane-changing, and gap-acceptance models). This research mainly focuses on vehicle-following models, assuming vehicles drive in a single lane, and there is no lane-change maneuver (i.e., lane-changing and gap-acceptance models are not required). However, a lane-change maneuver can temporarily affect vehicle-following behaviors (e.g., drivers speed up or slow down to align with acceptable gaps in target lanes; drivers temporarily adopt shorter gaps after a lane-change maneuver; drivers temporarily adopt shorter gaps after a vehicle merges in front).*

3.5.2. Automated Mode

When (1) a vehicle dedicated to automated driving systems approaches a vehicle, or (2) a vehicle equipped with cooperative automated driving systems approaches a vehicle not equipped with cooperative automated driving systems, a longitudinal control function similar to ACC is activated [69]

$$a_{i+1}[k] \triangleq \max \left(\min \left(K_{p,a}[k] e_x[k] + K_{d,a}[k] e_v[k], a_{max}^{i+1}[k] \right), -d_{max}^{i+1}[k] \right), \quad (16)$$

where $K_{p,a}$ is proportional gain in automated mode (s^{-2}), e_x is distance gap error (ft) calculated as $S_{des}[k] - S[k]$, S_{des} is desired distance gap (ft) calculated as $\max(T_{set}, T_{min}[k - 1])v_{i+1}[k - 1]$, T_{set} is preset time gap (s), $K_{d,a}$ is derivative gain in automated mode (s^{-1}), and e_v is speed error (ft/s) calculated as $v_i[k] - v_{i+1}[k]$. When no leader is detected, a longitudinal control function similar to cruise control is activated

$$a_{i+1}[k] \triangleq \max \left(\min \left(K_{p,cr}[k] (FFS - v_{i+1}[k]), a_{max}^{i+1}[k] \right), -d_{max}^{i+1}[k] \right), \quad (17)$$

where $K_{p,cr}$ is proportional gain in cruise mode (s^{-1}). $K_{p,a}$ and $K_{d,a}$ should satisfy (18) to maximize road capacity without compromising safety [63]

$$\left\{ \begin{array}{l} \left[\frac{2\pi\tau_{lag}^{i+1}}{\sqrt{4K_{p,a}(t)\tau_{lag}^{i+1} - (K_{d,a}(t) + 1)^2}} - \frac{\pi}{2} \cdot \sqrt{\frac{\tau_{lag}^{i+1}}{K_{p,a}(t)}} \right] \times \\ \left[\frac{\pi(K_{d,a}(t) + 1)}{\sqrt{4K_{p,a}(t)\tau_{lag}^{i+1} - (K_{d,a}(t) + 1)^2}} - v_i(t) \right] \leq S_{min}(t), \quad \text{during acceleration} \\ K_{d,a}(t) \leq \frac{T_{min}(t)}{8\tau_{lag}^{i+1}} - 1. \quad \text{during deceleration} \end{array} \right. \quad (18)$$

3.5.3. Cooperative Automated Mode

When a vehicle equipped with cooperative automated driving systems approaches another vehicle equipped with cooperative automated driving systems, a longitudinal control function similar to CACC is activated [69]

$$a_{i+1}[k] \triangleq \max \left(\min \left(K_{p,c}[k]e_v[k] + K_{i,c}[k]e_x[k] + K_{d,c}[k]a_i[k], a_{max}^{i+1}[k] \right), -d_{max}^{i+1}[k] \right), \quad (19)$$

where $K_{p,c}$ is proportional gain in cooperative automated mode (s^{-1}), $K_{i,c}$ is integral gain in cooperative automated mode (s^{-2}), and $K_{d,c}$ is derivative gain in cooperative automated mode (unitless).

Remark 2. All driver characteristics and vehicle powertrain information used in this research are derived from <https://www.automobile-catalog.com/> [63,70,71].

Assumption 3. Class B cooperative driving automation cooperation is utilized.

Assumption 4. x_i and v_i are prone to measurement noise and process noise, and a_i is prone to measurement noise, process noise, natural fault, and malicious fault (i.e., x_i and v_i are known state subvectors, and a_i is an unknown state subvector).

4. State and Unknown Input Estimation

Consider a state-space model in which unknown inputs can be modeled as an additive term

$$x[k+1] = Ax[k] + Bu[k] + Dd[k] + \xi[k], \quad (20)$$

$$z[k] = Cx[k] + \theta[k], \quad (21)$$

where $x \in \mathbb{R}^n$ is state vector, $A \in \mathbb{R}^{n \times n}$ is state matrix, $B \in \mathbb{R}^{n \times p}$ is input matrix, $u \in \mathbb{R}^p$ is input vector, $D \in \mathbb{R}^{n \times q}$ is unknown input matrix, $d \in \mathbb{R}^q$ is unknown input vector, $\xi \in \mathbb{R}^n$ is process noise, $z \in \mathbb{R}^m$ is measurement vector, $C \in \mathbb{R}^{m \times n}$ is measurement matrix, and $\theta \in \mathbb{R}^m$ is measurement noise.

Assumption 5. D is full column rank.

Assumption 6. $\text{rank } CD = \text{rank } D$.

Assumption 7. ξ is white noise: $\mathbb{E}(\xi[k]) = 0_n$, $\mathbb{E}(\xi[k]\xi^T[k]) \triangleq \Xi[k]$, $\Xi \in \mathbb{R}^{n \times n}$ is process noise covariance matrix, $\mathbb{E}(\xi[k]\xi^T[j]) = 0_{n \times n} \forall k, j \geq 0, k \neq j$.

Assumption 8. θ is white noise: $\mathbb{E}(\theta[k]) = 0_m$, $\mathbb{E}(\theta[k]\theta^T[k]) \triangleq \Theta[k]$, $\Theta \in \mathbb{R}^{m \times m}$ is measurement noise covariance matrix, and $\mathbb{E}(\theta[k]\theta^T[j]) = 0_{m \times m} \forall k, j \geq 0, k \neq j$.

Assumption 9. $\mathbb{E}(\xi[k]x^T[0]) = 0_{n \times n}$, and $\mathbb{E}(\theta[k]x^T[0]) = 0_{m \times n} \forall k \geq 0$.

Assumption 10. $\hat{x}[0] = \mathbb{E}(x[0])$ is known, where $\hat{x} \in \mathbb{R}^n$ is state estimation.

Remark 3. D is full column rank $\rightarrow [N D]^{-1}$ exists, where $N \in \mathbb{R}^{n \times (n-q)}$ [72].

Let us define $\bar{x}[k] \triangleq [N D]^{-1} x[k]$, $\bar{A} \triangleq [N D]^{-1} A [N D]$, $\bar{B} \triangleq [N D]^{-1} B$, $\bar{D} \triangleq [N D]^{-1} D$, and $\bar{C} \triangleq C [N D]$, where $\bar{x} \in \mathbb{R}^n$, $\bar{A} \in \mathbb{R}^{n \times n}$, $\bar{B} \in \mathbb{R}^{n \times p}$, $\bar{D} \in \mathbb{R}^{n \times q}$, and $\bar{C} \in \mathbb{R}^{m \times n}$.

Remark 4. $\text{rank } CD = \text{rank } D \rightarrow U \triangleq [CD Q]^{-1}$ exists, where $Q \in \mathbb{R}^{m \times (m-q)}$ [72].

Let us define $\bar{x}[k] \triangleq [\bar{x}_1^T \bar{x}_2^T]^T$, $\bar{A} \triangleq \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}$, $\bar{B} \triangleq [\bar{B}_1^T \bar{B}_2^T]^T$, $U^{-1} \triangleq [U_1^T U_2^T]^T$, where $\bar{x}_1 \in \mathbb{R}^{n-q}$ is known state subvector, $\bar{x}_2 \in \mathbb{R}^q$ unknown state subvector, $\bar{A}_{11} \in \mathbb{R}^{(n-q) \times (n-q)}$, $\bar{A}_{12} \in \mathbb{R}^{(n-q) \times q}$, $\bar{A}_{21} \in \mathbb{R}^{q \times (n-q)}$, $\bar{A}_{22} \in \mathbb{R}^{q \times q}$, $\bar{B}_1 \in \mathbb{R}^{(n-q) \times p}$, $\bar{B}_2 \in \mathbb{R}^{q \times p}$, $U_1 \in \mathbb{R}^{q \times m}$, and $U_2 \in \mathbb{R}^{(m-q) \times m}$.

Assumption 11. $\text{rank} \begin{bmatrix} sI_{n-q} - \bar{A}_{11} & -\bar{A}_{12} \\ CN & CD \end{bmatrix} = n \quad \forall s \in \mathbb{C}, \text{Re}(s) \geq 0$.

Let us define $\tilde{A}_1 \triangleq \bar{A}_{11} - \bar{A}_{12} U_1 C N$, $E_1 \triangleq \bar{A}_{12} U_1$, $\tilde{C}_1 \triangleq U_2 C N$, and $\bar{z}[k] \triangleq U_2 z[k]$, where $\tilde{A}_1 \in \mathbb{R}^{(n-q) \times (n-q)}$, $E_1 \in \mathbb{R}^{(n-q) \times m}$, $\tilde{C}_1 \in \mathbb{R}^{(m-q) \times (n-q)}$, and $\bar{z} \in \mathbb{R}^{m-q}$.

Remark 5. D is full column rank, $\text{rank } CD = \text{rank } D$, and $\text{rank} \begin{bmatrix} sI_{n-q} - \bar{A}_{11} & -\bar{A}_{12} \\ CN & CD \end{bmatrix} = n \quad \forall s \in \mathbb{C}, \text{Re}(s) \geq 0 \rightarrow \{\tilde{A}_1, \tilde{C}_1\}$ is observable [72].

State vector can be decoupled into known and unknown state subvectors. Known state subvector can be estimated as

$$\hat{x}_1[k+1] \triangleq (\tilde{A}_1 - L[k] \tilde{C}_1) \hat{x}_1[k] + \bar{B}_1 u[k] + L^*[k] z[k], \quad (22)$$

where $\hat{x}_1 \in \mathbb{R}^{n-q}$ is known state estimator ($\hat{x}_1[k] \rightarrow \bar{x}_1[k]$ as $k \rightarrow \infty$), $L^*[k] \triangleq L[k] U_2 + E_1$, $L^* \in \mathbb{R}^{(n-q) \times m}$, and $L \in \mathbb{R}^{(n-q) \times (m-q)}$ is Kalman gain, calculated as

$$L[k] \triangleq \tilde{A}_1 \Sigma[k] \tilde{C}_1^T \left(\tilde{C}_1 \Sigma[k] \tilde{C}_1^T + \Theta[k] \right)^{-1}, \quad (23)$$

where $\Sigma \in \mathbb{R}^{(n-q) \times (n-q)}$ can be recursively calculated as

$$\Sigma[k+1] = \tilde{A}_1 [\Sigma[k] - \Sigma[k] \tilde{C}_1^T (\tilde{C}_1 \Sigma[k] \tilde{C}_1^T + \Theta[k])^{-1} \tilde{C}_1 \Sigma[k]] \tilde{A}_1^T + D \Xi[k] D^T. \quad (24)$$

Unknown state subvector can be estimated as

$$\hat{x}_2[k] \triangleq U_1 z[k] - U_1 C N \hat{x}_1[k], \quad (25)$$

where $\hat{x}_2 \in \mathbb{R}^q$ is unknown state estimator ($\hat{x}_2[k] \rightarrow \bar{x}_2[k]$ as $k \rightarrow \infty$). Unknown input vector can be estimated as

$$\hat{d}[k] \triangleq U_1 z[k+1] + G_{d,1}[k] \hat{x}_1 + G_{d,2}[k] z[k] + G_{d,3} u[k], \quad (26)$$

where $\hat{d} \in \mathbb{R}^q$ is unknown input estimator, $G_{d,1} \in \mathbb{R}^{q \times (n-q)}$, $G_{d,2} \in \mathbb{R}^{q \times m}$, and $G_{d,3} \in \mathbb{R}^{q \times p}$. A controller can be further designed based on $\hat{d}$, z , and x_{des} , where $x_{des} \in \mathbb{R}^n$ is our desired state vector [73–77].

5. Test Scenario

Let us consider a traffic with ten driver types and fourteen vehicle models operating in manual, automated, and cooperative automated modes over US06 and Cycle D driving schedules with given conditions in Table 11 under malicious increases of 1 ft/s², 3 ft/s², and 5 ft/s² in transmitted accelerations, where

$$x[k] \triangleq \left[S^T[k] \ v_{i+1}^T[k] \ a_{i+1}^T[k] \ v_i^T[k] \ a_i^T[k] \right]^T, \quad (27)$$

$$u[k] \triangleq \left[K_p e_v^T[k] \ K_i e_p^T[k] \ K_d a_i^T[k] \ v_{ref}^T[k] \ a_{ref}^T[k] \right]^T, \quad (28)$$

$$A := \begin{bmatrix} 1 & -\Delta t & 0 & \Delta t & 0 \\ 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \text{ and } B := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (29)$$

Table 11. Input parameters.

Parameter	Value	Unit	Parameter	Value	Unit
Δt	0.1	s	$S_{des}[1]$	5	ft
ρ	0.002377 *	slug/ft ³	α	2	-
θ	0	-	β	4	-
Drivetrain Type	Front-Wheel-Drive	-	$K_{p,cr}[1]$	1	s ⁻¹
μ	1 **	-	$K_{p,a}[1]$	-1	s ⁻²
l_r	$L/2$	-	$K_{d,a}[1]$	1	s ⁻¹
η_b	0.95	-	$K_{p,c}[1]$	1	s ⁻¹
γ_b	1.04	-	$K_{i,c}[1]$	-1	s ⁻²
τ_s	1.0, 0.6, 0.0	s	$K_{d,c}[1]$	1	-
τ_c	0.0, 0.1	s	FFS	110	ft/s
Driver Type	5 #	-	h_{min}	2	s
$x[1]$	100 #, 0 ##	ft	q	1800	veh/h
$v[1]$	0 ##	ft/s	Warm-Up Period	900	s
$a[1]$	0 ##	ft/s ²	Replications	20	-
T_{set}	1.1, 0.6	s			

* for 0 ft altitude, 59° F temperature, and 14.7 lb/in² pressure, ** for good and dry pavement, # leader, ## follower.

6. Results

Distance gap, speed, and acceleration profiles are shown in Figures 2–4, arranged from shortest to longest time elapsed till crash occurs (showed as dashed lines)—2011 Ford F150 (7.5 s), 2004 Pontiac Grand Am (7.5 s), 2006 Honda Civic Si (7.6 s), 2009 Honda Civic (8.1 s), 2005 Mazda 6 (8.1 s), 2008 Chevy Impala (8.5 s), 2002 Chevy Silverado (8.5 s), 2004 Chevy Tahoe (9.0 s), 1998 Buick Century (9.3 s), 1998 Chevy S10 Blazer (9.3 s), intermediate semi-trailer (42.5 s), single-unit truck (42.8 s), interstate semi-trailer (44.8 s), and double semi-trailer (45.2 s). Results show that (1) proposed state and unknown input estimation model can be used to design a safe cooperative automated longitudinal control function under measurement noise, process noise, natural fault, and malicious fault; (2) vehicles over Cycle D driving schedule are more sensitive to fault magnitude than vehicles over US06 driving schedule (see Table 15), since vehicles over Cycle D driving schedule have lower average speeds and, therefore, maintain shorter time gaps than vehicles over US06 driving schedule; (3) passenger cars are more sensitive to fault magnitude than trucks, particularly at lower magnitude faults (see Table 15), since passenger cars maintain shorter time gaps than trucks; (4) errors in distance gap, speed, and acceleration are proportional to fault magnitude (see Tables 12–14); (5) errors in distance gap, speed, and acceleration are not sensitive to driving schedule; (6) distance gap is most sensitive state; (7) acceleration is least sensitive state; (8) adding 3.4 ft to estimated distance gaps, deducting 2.6 ft/s from estimated speeds, or deducting 0.8 ft/s² from estimated accelerations can mitigate impacts

of up to malicious increase of 5 ft/s^2 in transmitted accelerations (as a hypothesis) (see Tables 12–14), (9) higher magnitude faults lead to earlier crashes (see Table 15).

(a) 2011 Ford F150.

(b) 2004 Pontiac Grand Am.

(c) 2006 Honda Civic Si.

(d) 2009 Honda Civic.

(e) 2005 Mazda 6.

(f) 2008 Chevy Impala.

Figure 2. Cont.

(g) 2002 Chevy Silverado.

(h) 2004 Chevy Tahoe.

(i) 1998 Buick Century.

(j) 1998 Chevy S10 Blazer.

(k) 1998 Buick Century.

(l) 1998 Chevy S10 Blazer.

Figure 2. Cont.

(m) 2002 Chevy Silverado.

(n) 2004 Chevy Tahoe.

Figure 2. Distance gap profiles over US06 driving schedule under malicious increases in transmitted accelerations.

(a) 2011 Ford F150.

(b) 2004 Pontiac Grand Am.

(c) 2006 Honda Civic Si.

(d) 2009 Honda Civic.

Figure 3. Cont.

(e) 2005 Mazda 6.

(f) 2008 Chevy Impala.

(g) 2002 Chevy Silverado.

(h) 2004 Chevy Tahoe.

(i) 1998 Buick Century.

(j) 1998 Chevy S10 Blazer.

Figure 3. Cont.

(k) 1998 Buick Century.

(l) 1998 Chevy S10 Blazer.

(m) 2002 Chevy Silverado.

(n) 2004 Chevy Tahoe.

Figure 3. Speed gap profiles over US06 driving schedule under malicious increases in transmitted accelerations.

(a) 2011 Ford F150.

(b) 2004 Pontiac Grand Am.

Figure 4. Cont.

(c) 2006 Honda Civic Si.

(d) 2009 Honda Civic.

(e) 2005 Mazda 6.

(f) 2008 Chevy Impala.

(g) 2002 Chevy Silverado.

(h) 2004 Chevy Tahoe.

Figure 4. Cont.

(i) 1998 Buick Century.

(j) 1998 Chevy S10 Blazer.

(k) 1998 Buick Century.

(l) 1998 Chevy S10 Blazer.

(m) 2002 Chevy Silverado.

(n) 2004 Chevy Tahoe.

Figure 4. Acceleration gap profiles over US06 driving schedule under malicious increases in transmitted accelerations.

Table 12. Distance gap errors * (ft) before crash occurs **.

Malicious Fault (ft/s ²)	US06			Cycle D		
	1	3	5	1	3	5
2011 Ford F150	0.8	2.2	3.3	0.8	2.2	3.3
2004 Pontiac Grand Am	0.8	2.2	3.4	0.8	2.2	3.3
2006 Honda Civic Si	0.8	2.2	3.3	0.8	2.2	3.3
2009 Honda Civic	0.8	2.2	3.3	0.8	2.2	3.3
2005 Mazda 6	0.8	2.2	3.3	0.8	2.2	3.3
2008 Chevy Impala	0.8	2.2	3.2	0.8	2.2	3.2
2002 Chevy Silverado	0.8	2.2	3.2	0.8	2.2	3.2
2004 Chevy Tahoe	0.8	2.1	3.1	0.8	2.1	3.1
1998 Buick Century	0.9	2.2	3.3	0.8	2.1	3.2
1998 Chevy S10 Blazer	0.9	2.2	3.3	0.8	2.1	3.1
Intermediate Semi-Trailer	0.9	2.2	3.2	0.8	2.1	3.1
Single-Unit Truck	1.0	2.1	3.1	0.8	2.0	3.0
Interstate Semi-Trailer	0.9	2.2	3.2	0.8	2.1	3.0
Double Semi-Trailer	1.0	2.2	3.2	0.8	2.0	3.1

* calculated as $S_{i+1}^{normal}[k-1] - S_{i+1}^{faulty}[k-1]$, where S^{normal} is distance gap in normal conditions, and S^{faulty} is distance gap in faulty conditions, ** in absence of our proposed state and unknown input estimation model.

Table 13. Speed errors * (ft/s) before crash occurs **.

Malicious Fault (ft/s ²)	US06			Cycle D		
	1	3	5	1	3	5
2011 Ford F150	−0.2	−1.1	−2.1	−0.3	−1.1	−2.1
2004 Pontiac Grand Am	−0.2	−1.0	−1.9	−0.2	−1.0	−2.0
2006 Honda Civic Si	−0.2	−1.0	−2.0	−0.3	−1.0	−2.0
2009 Honda Civic	−0.2	−1.1	−2.1	−0.3	−1.1	−2.1
2005 Mazda 6	−0.2	−1.0	−2.0	−0.3	−1.1	−2.0
2008 Chevy Impala	−0.2	−1.1	−2.1	−0.3	−1.1	−2.1
2002 Chevy Silverado	−0.2	−1.1	−2.2	−0.3	−1.1	−2.2
2004 Chevy Tahoe	−0.2	−1.2	−2.4	−0.3	−1.3	−2.4
1998 Buick Century	−0.2	−1.1	−2.1	−0.3	−1.1	−2.2
1998 Chevy S10 Blazer	−0.2	−1.1	−2.2	−0.3	−1.2	−2.3
Intermediate Semi-Trailer	−0.1	−1.1	−2.2	−0.3	−1.2	−2.2
Single-Unit Truck	0.1	−0.8	−2.5	−0.3	−1.3	−2.6
Interstate Semi-Trailer	0.0	−1.1	−2.2	−0.3	−1.2	−2.3
Double Semi-Trailer	0.0	−1.1	−2.2	−0.3	−1.2	−2.3

* calculated as $v_{i+1}^{normal}[k-1] - v_{i+1}^{faulty}[k-1]$, where v^{normal} is speed in normal conditions, and v^{faulty} is speed in faulty conditions, ** in absence of our proposed state and unknown input estimation model.

Table 14. Acceleration errors * (ft/s²) before crash occurs **.

Malicious Fault (ft/s ²)	US06			Cycle D		
	1	3	5	1	3	5
2011 Ford F150	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
2004 Pontiac Grand Am	−0.1	−0.3	−0.5	−0.1	−0.3	−0.5
2006 Honda Civic Si	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
2009 Honda Civic	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
2005 Mazda 6	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
2008 Chevy Impala	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
2002 Chevy Silverado	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
2004 Chevy Tahoe	−0.1	−0.5	−0.8	−0.1	−0.5	−0.8
1998 Buick Century	−0.1	−0.4	−0.6	−0.1	−0.4	−0.6
1998 Chevy S10 Blazer	−0.1	−0.4	−0.7	−0.1	−0.4	−0.6
Intermediate Semi-Trailer	0.0	−0.4	−0.6	−0.1	−0.4	−0.5
Single-Unit Truck	−0.1	−0.5	−0.8	−0.1	−0.5	−0.8
Interstate Semi-Trailer	0.0	−0.4	−0.6	−0.1	−0.4	−0.5
Double Semi-Trailer	−0.1	−0.1	−0.6	−0.1	−0.4	−0.6

* calculated as $a_{i+1}^{normal}[k-1] - a_{i+1}^{faulty}[k-1]$, where a^{normal} is acceleration in normal conditions, and a^{faulty} is acceleration in faulty conditions, ** in absence of our proposed state and unknown input estimation model.

Table 15. Seconds elapsed till test till crash occurs *.

Malicious Fault (ft/s ²)	US06			Cycle D		
	1	3	5	1	3	5
2011 Ford F150	7.5	6.9	6.6	7.4	6.9	6.6
2004 Pontiac Grand Am	7.5	6.8	6.6	7.4	6.8	6.5
2006 Honda Civic Si	7.6	7.0	6.7	7.5	7.0	6.7
2009 Honda Civic	8.1	7.4	7.1	7.9	7.4	7.1
2005 Mazda 6	8.1	7.4	7.1	7.9	7.3	7.1
2008 Chevy Impala	8.5	7.7	7.4	8.2	7.7	7.4
2002 Chevy Silverado	8.5	7.7	7.4	8.2	7.7	7.4
2004 Chevy Tahoe	9.0	8.2	7.9	8.6	8.1	7.9
1998 Buick Century	9.3	8.3	8.0	8.7	8.2	7.9
1998 Chevy S10 Blazer	9.3	8.3	8.0	8.7	8.2	7.9
Intermediate Semi-Trailer	42.5	9.1	8.7	9.4	8.9	8.6
Single-Unit Truck	42.8	9.1	8.8	9.4	8.9	8.7
Interstate Semi-Trailer	44.8	9.3	9.0	9.6	9.1	8.8
Double Semi-Trailer	45.2	9.4	9.0	9.7	9.1	8.9

* in absence of our proposed controller.

Levels 1 and 2 automated vehicles are assumed to maintain minimum safe distance gap in a string; vehicles dedicated to automated driving systems and vehicles equipped with cooperative automated driving systems are assumed to maintain minimum safe time gap in a string; vehicles are assumed to maintain minimum safe distance gap in a platoon at each simulation time step to maximize road capacity without compromising safety or string stability. Therefore, increasing demand up to road capacity would not impact outputs (e.g., distance gap, time gap, speed, and acceleration) significantly. Demands exceeding road capacity will spill back behind entrance.

7. Discussion

Existing simulation tools may overestimate safety and road capacity improvements associated with cooperative driving automation due to not considering vehicle model and vehicle-to-vehicle communication vulnerabilities on a large scale. This research modifies a vehicle-following model for conventional vehicles, a longitudinal control function for vehicles dedicated to automated driving systems, and a longitudinal control function for vehicles equipped with cooperative automated driving systems, considering vehicle model and vehicle-to-vehicle communication vulnerabilities to maximize road capacity without compromising safety or string stability. Our proposed traffic microsimulation tool can be used to verify automated driving systems and cooperative automated driving systems in contested environments.

Drivers are assumed to drive in a single lane, and there is no lane-change maneuver, while a lane-change maneuver can temporarily affect vehicle-following behaviors. Future work can model other significant components underpinning a traffic microsimulation tool (i.e., lane-changing and gap acceptance)

- model motivation for mandatory, active, and discretionary lane-change maneuvers;
- model mandatory, active, and discretionary lane-change gap acceptance;
- model before lane-change, after lane-change, receiving, and yielding vehicle-following for each facility type (e.g., on-ramp and off-ramp);
- model lateral control for autonomous vehicles;
- model string operations (e.g., maximum platoon size, inter-platoon time gap, and cut-in and cut-out maneuvers).

Microscopic measures (e.g., distance headway and time headway) can be aggregated to macroscopic measures (e.g., density and flow) as $k \triangleq 1/s$ and $q = 3600 \times \bar{h}$, where k is density (veh/ft), and $\bar{h}$ is average distance headway (ft/veh). Future work can estimate macroscopic benefits associated with cooperative driving automation (e.g., increase in lane capacity) under various market penetration for autonomous and connected autonomous vehicles. Table 16 recommends potential improvements to our proposed longitudinal controller.

Table 16. Recommended control designs.

Cyberattack	Future Work	Description
Formulation	Fault and Delay	Most common cyberattacks can be modeled as fault (e.g., data falsification and spoofing attacks) or delay (e.g., denial-of-service attack).
Detection	Kalman Filter & Neural Network	Conventional fault-resilient longitudinal controllers are model-driven or data-driven, but not combined, potentially sacrificing accuracy or simulation speed.
Compensation	Adaptive Controller	Estimated distance gaps can be increased in proportion to cyberattack magnitude.

Author Contributions: Conceptualization, S.N., M.P. and M.N.; methodology, S.N.; software, M.P. and M.N.; validation, S.N.; formal analysis, M.P.; investigation, M.N.; resources, S.N.; data curation, S.N.; writing—original draft preparation, S.N.; writing—review and editing, M.P. and M.N.; visualization, M.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was sponsored by a contract from the Southeastern Transportation Research, Innovation, Development and Education Center (STRIDE), a Regional University Transportation Center sponsored by a grant from the U.S. Department of Transportation’s University Transportation Centers Program. The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation’s University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. On-Road Automated Driving Committee. *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*; Technical Report J3016; Society of Automotive Engineers: Warrendale, PA, USA, 2021.
2. National Center for Statistics and Analysis. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey (Traffic Safety Facts)*; Technical Report DOT HS 812 506; National Highway Traffic Safety Administration: Washington, DC, USA, 2018.
3. National Highway Traffic Safety Administration. *Federal Motor Vehicle Safety Standards; V2V Communications*; Technical Report NHTSA-2016-0126; US Department of Transportation: Washington, DC, USA, 2017.
4. On-Road Automated Driving Committee. *Taxonomy and Definitions for Terms Related to Cooperative Driving Automation for On-Road Motor Vehicles*; Technical Report J3216; Society of Automotive Engineers: Warrendale, PA, USA, 2021.
5. Olstam, J.J.; Tapani, A. *Comparison of Car-Following Models*; Volume 960 A; Swedish National Road and Transport Research Institute: Linköping, Sweden, 2004.
6. Upstream Security Ltd. *Global Automotive Cybersecurity Report*; Technical Report; Upstream Security Ltd.: Detroit, MI, USA, 2020.
7. Amoozadeh, M.; Raghuramu, A.; Chuah, C.N.; Ghosal, D.; Zhang, H.M.; Rowe, J.; Levitt, K. Security Vulnerabilities of Connected Vehicle Streams and Their Impact on Cooperative Driving. *IEEE Commun. Mag.* **2015**, *53*, 126–132. [[CrossRef](#)]
8. Sargolzaei, A.; Abbaspour, A.; Al Faruque, M.A.; Eddin, A.S.; Yen, K. Security Challenges of Networked Control Systems. In *Sustainable Interdependent Networks*; Springer: Cham, Switzerland, 2018; pp. 77–95.
9. Mouzakitis, A. Classification of Fault Diagnosis Methods for Control Systems. *Meas. Control* **2013**, *46*, 303–308. [[CrossRef](#)]
10. Abbaspour, A.; Mokhtari, S.; Sargolzaei, A.; Yen, K. A survey on active fault-tolerant control systems. *Electronics* **2020**, *9*, 1513. [[CrossRef](#)]
11. Nowakowski, C.; Shladover, S.E.; Lu, X.Y.; Thompson, D.; Kailas, A. *Cooperative Adaptive Cruise Control (CACC) for Truck Platooning: Operational Concept Alternatives*; Technical Report DTFH61-13-H-00012 Task 1.2; UC Berkeley: Berkeley, CA, USA, 2015.
12. Zambrano-Martinez, J.L.; Calafate, C.T.; Soler, D.; Cano, J.C. Towards realistic urban traffic experiments using DFRouter: Heuristic, validation and extensions. *Sensors* **2017**, *17*, 2921. [[CrossRef](#)]

13. Krajzewicz, D.; Hertkorn, G.; Rössel, C.; Wagner, P. SUMO (Simulation of Urban MObility)—An open-source traffic simulation. In Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM), Sharjah, United Arab Emirates, 28–30 October 2002; pp. 183–187.
14. Akçelik, R.; Besley, M. Acceleration and deceleration models. In Proceedings of the 23rd Conference of Australian Institutes of Transport Research (CAITR), Melbourne, Australia, 10–12 December 2001; p. 12.
15. Ahn, K.; Rakha, H.; Trani, A.; Van Aerde, M. Estimating vehicle fuel consumption and emissions based on instantaneous speed and acceleration levels. *J. Transp. Eng.* **2002**, *128*, 182–190. [\[CrossRef\]](#)
16. Fang, F.C.; Elefteriadou, L. Some guidelines for selecting microsimulation models for interchange traffic operational analysis. *J. Transp. Eng.* **2005**, *131*, 535–543. [\[CrossRef\]](#)
17. Kuriyama, M.; Yamamoto, S.; Miyatake, M. Theoretical study on eco-driving technique for an electric vehicle with dynamic programming. In Proceedings of the International Conference on Electrical Machines and Systems (ICEMS), Incheon, Korea, 10–13 October 2010; pp. 2026–2030.
18. Maurya, A.K.; Bokare, P.S. Study of deceleration behavior of different vehicle types. *Int. J. Traffic Transp. Eng.* **2012**, *2*, 253–270. [\[CrossRef\]](#)
19. Lee, J.; Park, B.B.; Malakorn, K.; So, J.J. Sustainability assessments of cooperative vehicle intersection control at an urban corridor. *Transp. Res. Part C: Emerg. Technol.* **2013**, *32*, 193–206. [\[CrossRef\]](#)
20. Anya, A.R.; Roupail, N.M.; Frey, H.C.; Schroeder, B. Application of AIMSUN microsimulation model to estimate emissions on signalized arterial corridors. *Transp. Res. Rec.* **2014**, *2428*, 75–86. [\[CrossRef\]](#)
21. Song, G.; Yu, L.; Geng, Z. Optimization of Wiedemann and Fritzsche car-following models for emission estimation. *Transp. Res. Part D Transp. Environ.* **2015**, *34*, 318–329. [\[CrossRef\]](#)
22. Bokare, P.S.; Maurya, A.K. Acceleration-deceleration behaviour of various vehicle types. *Transp. Res. Procedia* **2017**, *25*, 4733–4749. [\[CrossRef\]](#)
23. Ramezani, H.; Shladover, S.E.; Lu, X.Y.; Altan, O.D. Micro-simulation of truck platooning with cooperative adaptive cruise control: Model development and a case study. *Transp. Res. Rec.* **2018**, *2672*, 55–65. [\[CrossRef\]](#)
24. Shladover, S.E.; Nowakowski, C.; O’Connell, J.; Cody, D. Cooperative adaptive cruise control: Driver selection of car-following gaps. In Proceedings of the 17th ITS World Congress, Pusan, Korea, 25–29 October 2010; p. 8.
25. van Willigen, W.H.; Schut, M.C.; Kester, L.J. Approximating safe spacing policies for adaptive cruise control strategies. In Proceedings of the Vehicular Networking Conference (VNC), Amsterdam, The Netherlands, 14–16 November 2011; pp. 9–16.
26. Horiguchi, R.; Oguchi, T. A study on car following models simulating various adaptive cruise control behaviors. *Int. J. Intell. Transp. Syst. Res.* **2014**, *12*, 127–134. [\[CrossRef\]](#)
27. Flores, C.; Milanés, V.; Nashashibi, F. A time gap-based spacing policy for full-range car-following. In Proceedings of the 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–6.
28. Askari, A.; Farias, D.A.; Kurzhanskiy, A.A.; Varaiya, P. Effect of adaptive and cooperative adaptive cruise control on throughput of signalized arterials. In Proceedings of the Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 1287–1292.
29. Flores, C.; Milanés, V. Fractional-order-based ACC/CACC algorithm for improving string stability. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 381–393. [\[CrossRef\]](#)
30. Chen, J.; Zhou, Y.; Liang, H. Effects of ACC and CACC vehicles on traffic flow based on an improved variable time headway spacing strategy. *Intell. Transp. Syst.* **2019**, *13*, 1365–1373. [\[CrossRef\]](#)
31. Bian, Y.; Zheng, Y.; Ren, W.; Li, S.E.; Wang, J.; Li, K. Reducing time headway for platooning of connected vehicles via V2V communication. *Transp. Res. Part C: Emerg. Technol.* **2019**, *102*, 87–105. [\[CrossRef\]](#)
32. PTV. Virtual Testing of Autonomous Vehicles with PTV Vissim. Available online: <https://www.ptvgroup.com/en/solutions/products/ptv-vissim/areas-of-application/autonomous-vehicles-and-new-mobility/> (accessed on 16 August 2021).
33. Aimsun. Aimsun Unveils New Platform for Simulating a Driverless Future. Available online: <https://www.aimsun.com/aimsun-auto-launch/> (accessed on 1 August 2020).
34. Bu, F.; Tan, H.S.; Huang, J. Design and field testing of a cooperative adaptive cruise control system. In Proceedings of the American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 4616–4621.
35. Naus, G.J.; Vugts, R.P.; Ploeg, J.; van De Molengraft, M.J.; Steinbuch, M. String-stable CACC design and experimental validation: A frequency-domain approach. *IEEE Trans. Veh. Technol.* **2010**, *59*, 4268–4279. [\[CrossRef\]](#)
36. Liu, H. *Using Cooperative Adaptive Cruise Control (CACC) to Form High-Performance Vehicle Streams. Microscopic Traffic Modeling*; Technical Report DTFH61-13-H-00013; UC Berkeley: Berkeley, CA, USA, 2018.
37. Ploeg, J.; Scheepers, B.T.; Van Nunen, E.; Van de Wouw, N.; Nijmeijer, H. Design and experimental evaluation of cooperative adaptive cruise control. In Proceedings of the 14th International Conference on Intelligent Transportation Systems (ITSC), Washington, DC, USA, 5–7 October 2011; pp. 260–265.
38. Shladover, S.E.; Su, D.; Lu, X.Y. Impacts of cooperative adaptive cruise control on freeway traffic flow. *Transp. Res. Rec.* **2012**, *2324*, 63–70. [\[CrossRef\]](#)
39. Zhao, L.; Sun, J. Simulation framework for vehicle platooning and car-following behaviors under connected-vehicle environment. *Procedia-Soc. Behav. Sci.* **2013**, *96*, 914–924. [\[CrossRef\]](#)

40. Segata, M.; Joerer, S.; Bloessl, B.; Sommer, C.; Dressler, F.; Cigno, R.L. Plexe: A platooning extension for Veins. In Proceedings of the Vehicular Networking Conference (VNC), Paderborn, Germany, 3–5 December 2014; pp. 53–60.
41. Milanés, V.; Shladover, S.E. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transp. Res. Part C Emerg. Technol.* **2014**, *48*, 285–300. [\[CrossRef\]](#)
42. Nikolos, I.K.; Delis, A.I.; Papageorgiou, M. Macroscopic modelling and simulation of ACC and CACC traffic. In Proceedings of the 18th International Conference on Intelligent Transportation Systems (ITSC), Gran Canaria, Spain, 15–18 September 2015; pp. 2129–2134.
43. Delis, A.I.; Nikolos, I.K.; Papageorgiou, M. Simulation of the penetration rate effects of ACC and CACC on macroscopic traffic dynamics. In Proceedings of the 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 336–341.
44. Wang, Z.; Wu, G.; Barth, M.J. Developing a distributed consensus-based cooperative adaptive cruise control system for heterogeneous vehicles with predecessor following topology. *J. Adv. Transp.* **2017**, *2017*, 1–16. [\[CrossRef\]](#)
45. Terruzzi, L.; Colombo, R.; Segata, M. On the effects of cooperative platooning on traffic shock waves. In Proceedings of the Vehicular Networking Conference (VNC), Turin, Italy, 27–29 November 2017; pp. 37–38.
46. Zhou, Y.; Ahn, S.; Chitturi, M.; Noyce, D.A. Rolling horizon stochastic optimal control strategy for ACC and CACC under uncertainty. *Transp. Res. Part C Emerg. Technol.* **2017**, *83*, 61–76. [\[CrossRef\]](#)
47. Lemessi, M. An slx-based microsimulation model for a two-lane road section. In Proceedings of the Winter Simulation Conference (Cat. No. 01CH37304), Arlington, VA, USA, 9–12 December 2001; pp. 1064–1071.
48. Rakha, H.; Ding, Y. Impact of stops on vehicle fuel consumption and emissions. *J. Transp. Eng.* **2003**, *129*, 23–32. [\[CrossRef\]](#)
49. Wang, X.Y.; Liu, J.S. Research of the lane utilization with microsimulation. In Proceedings of the International Conference on Machine Learning and Cybernetics (ICMLC), Guangzhou, China, 18–21 August 2005; pp. 2681–2687.
50. Ossen, S.; Hoogendoorn, S.P.; Gorte, B.G. Interdriver differences in car-following: A vehicle trajectory-based study. *Transp. Res. Rec.* **2006**, *1965*, 121–129. [\[CrossRef\]](#)
51. Kesting, A.; Treiber, M.; Helbing, D. General lane-changing model MOBIL for car-following models. *Transp. Res. Rec.* **2007**, *1999*, 86–94. [\[CrossRef\]](#)
52. Kesting, A.; Treiber, M. Calibrating car-following models by using trajectory data: Methodological study. *Transp. Res. Rec.* **2008**, *2088*, 148–156. [\[CrossRef\]](#)
53. Talebpour, A.; Mahmassani, H.S.; Hamdar, S.H. Multiregime sequential risk-taking model of car-following behavior: Specification, calibration, and sensitivity analysis. *Transp. Res. Rec.* **2011**, *2260*, 60–66. [\[CrossRef\]](#)
54. Song, G.; Yu, L.; Zhang, Y. Applicability of traffic microsimulation models in vehicle emissions estimates: Case study of VISSIM. *Transp. Res. Rec.* **2012**, *2270*, 132–141. [\[CrossRef\]](#)
55. Lee, J.; Park, B. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 81–90. [\[CrossRef\]](#)
56. Treiber, M.; Kesting, A. Microscopic calibration and validation of car-following models—a systematic approach. *Procedia-Soc. Behav. Sci.* **2013**, *80*, 922–939. [\[CrossRef\]](#)
57. Li, Z.; Elefteriadou, L.; Ranka, S. Signal control optimization for automated vehicles at isolated signalized intersections. *Transp. Res. Part C Emerg. Technol.* **2014**, *49*, 1–18. [\[CrossRef\]](#)
58. Tang, T.Q.; He, J.; Yang, S.C.; Shang, H.Y. A car-following model accounting for the driver’s attribution. *Phys. A Stat. Mech. Its Appl.* **2014**, *413*, 583–591. [\[CrossRef\]](#)
59. Desiraju, D.; Chantem, T.; Heaslip, K. Minimizing the disruption of traffic flow of automated vehicles during lane changes. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 1249–1258. [\[CrossRef\]](#)
60. Liu, H.; Kan, X.D.; Shladover, S.E.; Lu, X.Y.; Ferlis, R.E. Modeling impacts of cooperative adaptive cruise control on mixed traffic flow in multi-lane freeway facilities. *Transp. Res. Part C Emerg. Technol.* **2018**, *95*, 261–279. [\[CrossRef\]](#)
61. Amoozadeh, M.; Deng, H.; Chuah, C.N.; Zhang, H.M.; Ghosal, D. Platoon management with cooperative adaptive cruise control enabled by VANET. *Veh. Commun.* **2015**, *2*, 110–123. [\[CrossRef\]](#)
62. Li, Y.; Wang, H.; Wang, W.; Xing, L.; Liu, S.; Wei, X. Evaluation of the impacts of cooperative adaptive cruise control on reducing rear-end collision risks on freeways. *Accid. Anal. Prev.* **2017**, *98*, 87–95. [\[CrossRef\]](#)
63. Noei, S.; Zhao, X.; Crane, C.D. Longitudinal control of vehicles in traffic microsimulation. *arXiv* **2020**, arXiv:2003.07861.
64. Mannering, F.L.; Washburn, S.S. *Principles of Highway Engineering and Traffic Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2020.
65. Ozkul, S. *Advanced Vehicle Dynamics Modeling Approach in Traffic Microsimulation with Emphasis on Commercial Truck Performance and On-Board-Diagnostics Data*; University of Florida: Gainesville, FL, USA, 2014.
66. Washburn, S.S.; Ozkul, S. *Heavy Vehicle Effects on Florida Freeways and Multilane Highways*; Technical Report BDK-77 977-15; Florida Department of Transportation: Tallahassee, FL, USA, 2013.
67. Washburn, S.; Noei, S. *Interchange Design to Accommodate Ramp Metering System*; Technical Report BDV-31 977-92; Florida Department of Transportation: Tallahassee, FL, USA, 2020.
68. Zhu, M.; Wang, X.; Tarko, A. Modeling car-following behavior on urban expressways in Shanghai: A naturalistic driving study. *Transp. Res. Part C Emerg. Technol.* **2018**, *93*, 425–445. [\[CrossRef\]](#)

69. Noei, S.; Santana, H.; Sargolzaei, A.; Noei, M. Reducing traffic congestion using geo-fence technology: Application for emergency car. In Proceedings of the 1st International Workshop on Emerging Multimedia Applications and Services for Smart Cities (EMASC), Orlando, FL, USA, 7 November 2014; pp. 15–20.
70. Washburn, S.S. Drivers. Available online: <https://swashsim.miraheze.org/wiki/Drivers> (accessed on 16 August 2021).
71. Washburn, S.S. Vehicles. Available online: <https://swashsim.miraheze.org/wiki/Vehicles> (accessed on 16 August 2021).
72. Khalghani, M.R.; Khushalani-Solanki, S.; Solanki, J.; Sargolzaei, A. Cyber disruption detection in linear power systems. In Proceedings of the North American Power Symposium (NAPS), Morgantown, WV, USA, 17–19 September 2017; pp. 1–6.
73. Mokhtari, S.; Abbaspour, A.; Yen, K.K.; Sargolzaei, A. A machine learning approach for anomaly detection in industrial control systems based on measurement data. *Electronics* **2021**, *10*, 407. [[CrossRef](#)]
74. Sargolzaei, A.; Crane, C.; Abbaspour, A.; Noei, S. A machine learning approach for fault detection in vehicular cyber-physical systems. In Proceedings of the 15th International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; pp. 636–640.
75. Sargolzaei, A.; Yazdani, K.; Abbaspour, A.; Crane III, C.D.; Dixon, W.E. Detection and mitigation of false data injection attacks in networked control systems. *IEEE Trans. Ind. Inform.* **2019**, *16*, 4281–4292. [[CrossRef](#)]
76. Noei, M.; Abadeh, M.S. A genetic asexual reproduction optimization algorithm for imputing missing values. In Proceedings of the 9th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 24–25 October 2019; pp. 214–218.
77. Parvizimosaed, M.; Noei, M.; Yalpanian, M.; Bahrami, J. A containerized integrated fast IoT platform for low energy power management. In Proceedings of the 7th International Conference on Web Research (ICWR), Tehran, Iran, 19–20 May 2021; pp. 318–322.