



smart cities



Review

A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation

Hafiz Usman Ahmed, Ying Huang and Pan Lu

Special Issue

Feature Papers for Smart Cities


Edited by
Prof. Dr. Pierluigi Siano



<https://doi.org/10.3390/smartcities4010019>

Review

A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation

Hafiz Usman Ahmed ¹, Ying Huang ^{1,*} and Pan Lu ² 
¹ Department of Civil and Environmental Engineering, North Dakota State University, Fargo, ND 58108, USA; hafiz.ahmed@ndsu.edu

² Department of Transportation, Logistics and Finance, North Dakota State University, Fargo, ND 58102, USA; pan.lu@ndsu.edu

* Correspondence: ying.huang@ndsu.edu; Tel.: +1-701-231-7651

Abstract: The platform of a microscopic traffic simulation provides an opportunity to study the driving behavior of vehicles on a roadway system. Compared to traditional conventional cars with human drivers, the car-following behaviors of autonomous vehicles (AVs) and connected autonomous vehicles (CAVs) would be quite different and hence require additional modeling efforts. This paper presents a thorough review of the literature on the car-following models used in prevalent micro-simulation tools for vehicles with both human and robot drivers. Specifically, the car-following logics such as the Wiedemann model and adaptive cruise control technology were reviewed based on the vehicle's dynamic behavior and driving environments. In addition, some of the more recent "AV-ready (autonomous vehicles ready) tools" in micro-simulation platforms are also discussed in this paper.



Citation: Ahmed, H.U.; Huang, Y.; Lu, P. A Review of Car-Following Models and Modeling Tools for Human and Autonomous-Ready Driving Behaviors in Micro-Simulation. *Smart Cities* **2021**, *4*, 314–335. <https://doi.org/10.3390/smartcities4010019>

Academic Editor: Pierluigi Siano

Received: 1 February 2021

Accepted: 26 February 2021

Published: 3 March 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/>).

Keywords: car-following models; microscopic traffic simulation; autonomous and connected vehicles; PTV VISSIM; AV-ready

1. Introduction

The transport simulation models provide opportunities for decision-makers to evaluate the potential benefits and associated risks of future vehicle technologies and infrastructure automation. The automation industry for vehicle technologies is progressively improving nowadays to provide benefits for the end-users in terms of traffic congestion reduction, safety enhancements, fuel cost savings, and reduction in driver's stress levels. In native terms, autonomous vehicles (AVs) are referred to as self-driving cars that do not require or require limited inputs from human drivers [1]. The AVs and connected autonomous vehicles (CAVs) are expected to improve the traffic flow and overall road safety issues. Many states in the USA are currently testing the autonomous concepts, innovations, and the availability of this technology for the general public in the coming years [2]. A number of public and private sectors from such as automobile industries and universities are conducting research and development work on machine-learning algorithms, artificial intelligence, sensors technology innovation, and in-vehicle equipment [2,3]. However, currently, the penetration rate of the AVs is still very low, and it will still need decades for the AVs to penetrate fully into the market [1]. With the low actual penetration rate of AVs in traffic, the use of transport simulation modeling tools to analyze the impacts of AVs on transportation networks will be a significant approach for studies related to the AVs.

In general, transport simulation models can be divided into three major categories, including macroscopic, mesoscopic, and microscopic modeling, depending upon the level of details required for network analysis. The macroscopic models are developed and operated by transport planners for managing large-scale traffic networks [4] such as motorways, freeways, corridors, urban traffic congestion [5], mass transit operations, and travel

demand models. These models consider deterministic characteristics of a traffic stream, such as speed, flow, and density [6,7]. The microscopic-simulation models, on the other hand, records the behavior of all individual cars for each simulation time-step based on the directions given by car-following, lane-changing, and gap-acceptance algorithms. It models the complex urban street network systems, interchanges, pedestrian movements, traffic signals, multi-modal systems, and proposed mitigations for existing network problems. Lastly, the mesoscopic models combine both macroscopic and microscopic simulation models, however, it provides less reliable information for individual vehicles as compare to micro-simulation. The use of mesoscopic models is typically for transport planning analysis techniques [7]. Among all the three categories of simulation modeling tools, since the micro-simulation presents the impact of each individual AVs on the road network, in the condition of low penetrations of AVs, the application of micro-simulation to analyze the behaviors of AVs would generate very useful information for their future development, which will be the focused review topic in this paper.

In micro-simulation, the driving behaviors of the AVs can be achieved either by modifying the driving model parameters of the commonly applied human drivers and/or by developing new smart logics for vehicle cooperation and communications [8,9]. No matter if it is a modified driving model from human drivers or a new smart driving logic, a car-following model is required to state the driving behavior of individual vehicles in a longitudinal direction in the simulated traffic condition of micro-simulation, which describes how the simulated vehicles interact with the preceding vehicle (leader) in the same lane [10,11].

A car-following model for either human or robot drivers can be achieved based on the driving strategies that are derived from the assumptions of a real driving phenomenon, such as keeping a “safe-gap” from the preceding vehicle, cruising at a desired velocity, or choosing the accelerations from the desired limits [12]. Different researchers have explained the car-following behavior through acceleration distributions, safety gaps, and reaction times of the subjected following-vehicle under numerous traffic conditions. This paper first presents a detailed review of traditional car-following models used in prevalent micro-simulation tools for conventional and self-driving cars. Secondly, VISSIM microsimulation software is taken into the spotlight as it shows potentials for a wide range of applications (i.e., human, and robotic driving) compared with other commercially available simulation tools in the market. The objective of this paper is to establish a comprehensive review of human and AV/CAVs car-following models and investigate the responses of microsimulation tools towards the performance of each model, which will be beneficial for users who are interested in advancing the modeling capacities of AV/CAVs.

2. Car-Following Models for Human-Driven and Autonomous Vehicles

Figure 1 shows the basic illustration of a typical car-following, lane-change, and gap-acceptance behaviors of any vehicles in a micro-simulation. For a human driver, the driving logic of a vehicle tends to illustrate how a real human driving phenomenon would be on a typical roadway in both longitudinal and lateral directions [10]. In the longitudinal direction, a vehicle with human drivers would follow preceding cars with appropriate gap acceptance, speed adoption, and desired acceleration or deceleration. In the lateral direction, the human drivers need to decide when and how to perform lane changes, merging/diverging, and overtaking [13].

For any car-following model, the micro-simulation tool uses various statistical distributions and functions for defining the maximum, minimum, desired acceleration/deceleration values, maximum, desired speed values, and other vehicular characteristics. To determine an acceptable or safe gap from the leading vehicle before applying any following models for the human drivers, a micro-simulation tool employs a user-defined value for the required critical gap or suggested gap-rejection criteria for congested and uncongested networks [6,14]. A driver needs to decide such a gap value at multiple locations while driving on a roadway such as a vehicle approaching an intersection or leaving any intersection

(especially signalized), joining a weave section, merging, diverging, and yield locations, etc., which determines the quality of the network throughput and traffic flow. With the acceptable gap determined, the car-following models can then be implemented.

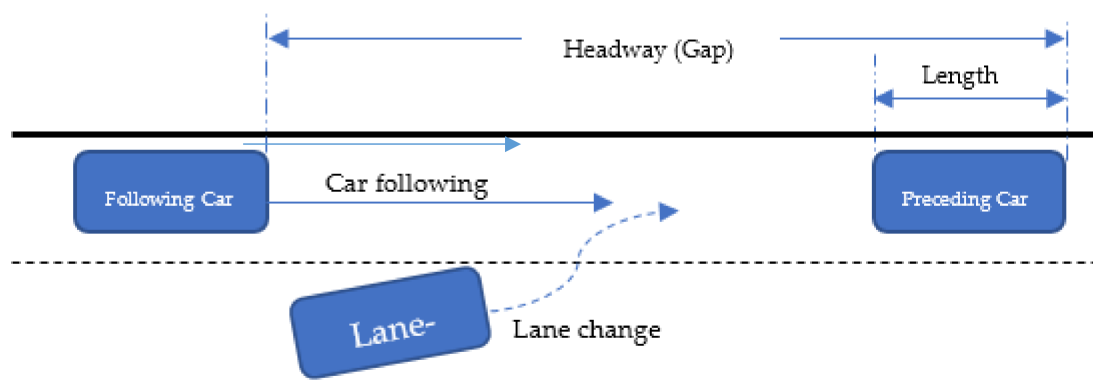


Figure 1. Typical illustration for car-following, lane-change, and gap-acceptance.

For human drivers and autonomous vehicles, there are several models that can be applied. In 2004, Olstam classified the car-following model based on utilized logic [10]. Table 1 shows a summary comparison of three classes of car-following models accordingly. Specifically, these car-following models include the General Motors (GM) model, the Safety-distance or collision avoidance model, the psychophysical model such as the Wiedemann model, the optimal velocity model (OVM), the intelligent driver model (IDM), and the adaptive cruise control (ACC) and cooperative adaptive cruise control (CACC) models, which are detailed in this section. In addition, since the lane-changing behavior is also closely related to the car-following model, a brief review was also performed on the lane-changing models for human drivers.

Table 1. Car-following models based on utilized logic [10].

Model Class	Model Logic and Assumption
Gazis-Herman-Rothery (GHR) model	"The following vehicles' acceleration is proportional to the subjected vehicle (own vehicle), the speed difference between the follower and the leader, and the space headway."
Safety-distance model	"The follower always keeps a safe distance to the vehicle ahead"
Psychophysical car-following model	"These models use threshold values." Drivers react to vehicles when the set threshold for relative velocity (speed difference between a follower and a lead vehicle) or spacing is reached.

2.1. The GM Car-Following Model

In 1961, General Motors (GM) research laboratories developed a car-following model by using the acceleration/deceleration as a stimulus [15,16]. This model calculated the driver's reaction time by estimating the acceleration/deceleration values by subtracting the leader's current speed from the speed of the following vehicle. The GM model is very simple and it established a basis for all other car-following models with varying sensitivity parameters [17]. The GM model is a linear car-following model with a constant sensitivity parameter [18] and the acceleration of the following vehicle [17] can be estimated as:

$$a_n(t) = \alpha \cdot \Delta V_n^{front}(t - \tau_n) \quad (1)$$

where, $a_n(t)$ is the acceleration applied by driver “ n ” at the time “ t ”, $\Delta V_n^{front}(t - \tau_n)$ is the leader-relative speed recorded at the time $(t - \tau_n)$, τ_n is the driver’s reaction time, and α is a constant term.

The basic GM linear model has an impractical approach because the spacing between the leading and the following vehicle does have its impact on stimulus. Later, researchers overcame some of the GM model limitations by suggesting several extensions. Herman et al.’s extension allows the following driver to accelerate if the relative speed of the leader is increased. Additionally, different sets of acceleration and deceleration parameters were proposed to make a decision [19]. Subramanian proposed an improvement that allows the driver to react faster under deceleration scenarios as compared to the acceleration cases [20]. A further extension to the driver alertness models includes the drivers’ behavior concerning the headway with the lead vehicle as either a free-flow or car-following behavior. A critical headway value is assigned to determine the state of driving behavior [21]. The GM model was further expanded to consider the non-linearity behavior in the sensitivity of relative distance between the following and leading vehicles and the sensitivity of relative speed for the subject vehicle [16]. The extended model was also known as Gazis–Herman–Rothery (GHR) model. In the GHR model, the acceleration of the following vehicle was modified [17] as:

$$a_n(t) = \alpha \cdot \frac{V_n(t)^\beta}{\Delta X_n^{front}(t - \tau_n)^\gamma} \cdot \Delta V_n^{front}(t - \tau_n) \quad (2)$$

where, $V_n(t)$ is the speed of the subject “ n ” vehicle measured at a time “ t ”, $\Delta V_n^{front}(t - \tau_n)$ is the relative speed of the leading vehicle, and $\Delta X_n^{front}(t - \tau_n)$ is the relative spacing measured at the time $(t - \tau_n)$, τ_n is the driver’s reaction time, and $\alpha > 0$, β , γ are constant parameters.

The acceleration equation in Equation (2) was expanded to incorporate traffic density as sensitivity terms and different drivers’ reaction times. However, the models based on GM-type has their own behavioral limitations [22,23], such as that the drivers react to arbitrary small changes in the stimulus, e.g., relative speed and the driver following the lead vehicle remains affected by the actions of its leader even when the distance is very large and that the follower’s response dismisses as the relative speed is zero.

2.2. The Safety-Distance or Collision Avoidance Car-Following Model

Unlike the GM family models that describe the relationship of stimulus-response type function, the safety-distance model always maintains a safe distance between the following and the lead vehicles. These collision avoidance models are based on the basic newton’s equation of motion that explains that the collision would be unavoidable if the lead vehicle behaves unpredictably [24]. The basic rule of safety-distance was quoted by Pipes in 1953 [25] as “a good rule for following another vehicle at a safe distance is to allow yourself at least the length of a car (about 15 ft) between you and the vehicle ahead for every 10 miles of hour speed at which you are traveling”.

The first general acceleration model was developed by Gipps in 1981 that records the car-following (vehicle ahead) and free-flow (no vehicle ahead) conditions. This model confirms a “no-collision” behavior between the follower and the leader if the time-gap is equal or greater than $3T/2$ (safe-headway), and the follower’s approximation about the leader’s deceleration is equal or larger than the leader’s actual deceleration value (apply limits to a driver’s braking rate) [26]. The acceleration model has two limitations that the desired speed must not surpass, and a critical safe headway should be maintained. The acceleration (v_n^{acc}) and deceleration (v_n^{decc}) of these models correspond to empirical formulations [26,27], which can be expressed as:

$$v_n^{acc}(t + T) = v_n(t) + 2.5 \cdot a_n \cdot T \left(1 - \frac{v_n(t)}{v_n^d} \right) \cdot \sqrt{0.025 + \frac{v_n(t)}{v_n^d}} \quad (3)$$

$$v_n^{decc}(t+T) = -T \cdot d_n + \sqrt{T^2 \cdot d_n^2 + d_n \left\{ 2x_{n-1}(t) - x_n(t) - (S_{n-1}) - T \cdot v_n(t) + \frac{v_{n-1}(t)^2}{d'_{n-1}} \right\}} \quad (4)$$

where T is the reaction time, “ n ”, “ $n - 1$ ” is the follower and leader, respectively, $v_n(t)$, $v_{n-1}(t)$ are the vehicle speeds of follower and leader respectively at the time “ t ”, v_n^d , a_n are the follower’s maximum desired speed and maximum acceleration, respectively, d_n , d'_{n-1} are the most severe braking that the follower wishes to undertake and his estimate of the leader’s most severe braking capability ($d_n > 0$ and $d'_{n-1} > 0$), $x_n(t)$, $x_{n-1}(t)$ are the longitudinal position of the follower and leader at the time “ t ”, S_{n-1} is the intervehicle spacing at a stop, which includes the length of the leader’s vehicle added to the follower’s desired intervehicle spacing at the stop. Based on Equations (3) and (4), the output result of these models is the speed of each vehicle at a time “ t ” corresponding to its previous time-step speed.

The Gipps’ model has gained numerous extensions, modifications, and calibration by various researchers [28–30] to meet the needs of problem-solving through the use of microsimulation tools. Most of the leading micro-simulation tools use the Gipps model [31–34] as a default driving behavior because of its simple calibration that requires the common-sense assumption about human driving behavior. One of the weaknesses of such safety-models, as explained in a review article [35] is that the model does not consider the driver’s perception. Further, a minor variation causes an end to the reaction for the following driver.

2.3. The Psychophysical Car-Following Models and the Wiedemann Model

Another approach of car-following models is the psychophysical or action point models [24]. The psychophysical models use a “perceptual threshold” which is a function of space headway and relative speed for the following vehicle pairs. The drivers react to the changes in spacing or relative velocity only when the threshold values are reached. In this way, it also records the driver’s attentiveness for small spacing and lack of the following behavior for larger spacings [17]. In 1992, Wiedemann and Reiter applied a psychophysical approach (perceptual thresholds) to a simulation system using a tool named “MISSION”. Four different threshold and regimes are defined to describe the interaction of a follower-leader vehicle pair [36], including the free-flowing (the subject vehicle is not influenced by the lead vehicle), the approaching/closing (the subject vehicle is consciously influenced by the slower-leading vehicle), the following/closing (the subject vehicle is unconsciously influenced by being in the following process), and the emergency (the headway between vehicle pair drops below the desired safety distance).

In the psychophysical car-following model developed by Wiedemann, also known as the Wiedemann model, the driving behavior of humans is considered to be naturally distributed [36], which essentially means that each driver has different driving capabilities for perception, reaction, and estimation of surrounding traffic environment, safety needs, desired speed, and aggressiveness towards maximum acceleration/deceleration values. Similarly, the same principle applies to vehicle abilities in exercising basic features such as maximum speed and maximum acceleration/deceleration. According to Wiedemann, the varying behavior of drivers can be explained by normal distributions. Hence, this car-following model uses several ranges of different random parameters for the calculation of threshold values and the driving functions [36]. Wiedemann defines different thresholds and regimes in the relative speed/space (DX/DV) diagram for the psychophysical follower-leader pair, including the desired distance (AX), the desired minimum following distance (ABX), the maximum following distance (SDX), the perception threshold (SDX), and the decreasing and increasing speed differences (CLDV, OPDV) [10,22].

The desired distance for a stationary vehicle (front-to-front) is denoted by the threshold of “AX”. This threshold contains the length of the front vehicle and the desired front-to-front distance (L), which can be defined as [10,22,36]:

$$AX = L_{n-1} + AXadd + RND1_n * AXmult \quad (5)$$

where “ n ” is for the subject vehicle and “ $n - 1$ ” is for the leading vehicle, “ $AXadd$ ” and “ $AXmult$ ” are calibration parameters as also detailed in Table 2 for typical values in simulation, and $RND1_n$ is a normally distributed parameter for desired front-to-rear distance that depends on the driver’s safety.

Table 2. Typical values for Wiedemann calibration parameters in simulation tools [10,11,36].

Parameter	Description	Value
AXadd	Additive calibration parameter	1.25 (m)
AXmult	Multiplicative calibration parameter	2.5 (m)
BXadd	Additive calibration parameter	2.0 (m)
BXmult	Multiplicative calibration parameter	1.0 (m)
EXadd	Additive calibration parameter	1.5 (m)
EXmult	Multiplicative calibration parameter	0.55 (m)
OPDVadd	Additive calibration parameter	1.5
OPDVMult	Multiplicative calibration parameter	1.5
CX	Calibration Parameter	40 *
BNULLmult	Multiplicative calibration parameter	0.1 (m/s ²)
NRND	Normally distributed random number	N (0.5, 0.15) **
RND1	Normally distributed driver number	N (0.5, 0.15) **
RND2	Normally distributed driver number	N (0.5, 0.15) **
RND4	Normally distributed driver number	N (0.5, 0.15) **
Bmax	Maximum acceleration	$3.5 - \frac{3.5}{40} * v$ ***
Bmin	Maximum deceleration	$-20 - \frac{1.5}{60} * v$ ***

* estimation from graphs [11], ** mean values have been used [36], *** estimation from graphs [36].

The desired minimum following distance at low-speed differences is denoted by the threshold of “ABX”. At higher speeds, the driver would undervalue the safe distance and drive closely unlike at slower speed with safe gaps, which can be estimated as:

$$ABX = AX + BX \quad (6)$$

$$BX = (BXadd + BXmult * RND1_n) * \sqrt{v} \quad (7)$$

in which $BXadd$ and $BXmult$ are calibration parameters with typical values shown in Table 2. The speed v is defined as:

$$v = \begin{cases} v_{n-1} & \text{for } v_n > v_{n-1} \\ v_n & \text{for } v_n \leq v_{n-1} \end{cases} \quad (8)$$

The maximum following distance is denoted by the threshold of “SDX”. At this threshold, the driver realizes that the following distance is growing. Therefore, an application of acceleration is required to maintain the ideal headway. The threshold distance ranges between 1.5 and 2.5 times the minimum following distance (“ABX”), which can be estimated as:

$$SDX = AX + EX * BX \quad (9)$$

$$EX = EXadd + EXmult * (NRND - RND2_n) \quad (10)$$

where EX_{add} and EX_{mult} are calibration parameters with typical values shown in Table 2, the “NRND” is an independent random parameter and the $RND2_n$ is a normally distributed driver-dependent parameter that represents the driver’s estimation ability.

The perception threshold for approaching a point at long distances is denoted by “SDV”. The driver reaches to appoint where he or she met with a slow-moving vehicle, hence the driver will react by decreasing the speed of his vehicle to maintain a gap higher than the minimum desired distance (“ABX”) as:

$$SDV = \left(\frac{\Delta x - L_{n-1} - AX}{CX} \right)^2 \quad (11)$$

$$CX = CX_{const} * (CX_{add} + CX_{mult} * (RND1_n + RND2_n)) \quad (12)$$

where “CXconst”, “CXadd”, and “CXmult” are calibration parameters with typical values shown in Table 2, and the Wiedemann model defines the range for “CX” as between 25 and 75.

The “CLDV” is a threshold that recognizes decreasing speed differences. This is similar to the behavior of the approaching point (SDV). In this threshold, the driver develops a perception of small speed differences, in short, decreasing distances. In VISSIM, “CLDV” is ignored and is assumed to be equal to SDV [11]. The “OPDV” is a threshold that recognizes increasing speed differences. This threshold defines a point where the driver observes that he or she is moving at a slower speed than the leader as:

$$OPDV = CLDV * (-OPDV_{add} - OPDV_{mult} * NRND) \quad (13)$$

in which “OPDVadd” and “OPDVmult” are calibration parameters with typical values also shown in Table 2.

Using all the thresholds as mentioned above, the driver of the following-vehicle falls in the four different regimes that decide the driving behavior, i.e., the value of acceleration in the longitudinal direction [36]. For the following behavior, the thresholds “ABX”, “SDX”, “SDV”, and “OPDV” influences the driver’s following behavior. Wiedemann defines that in the following regime if the vehicle passes “SDV” or “ABX” thresholds, it is assigned a deceleration rate $-B_{null}$. The reason is that the vehicle should slow down to prevent an accident. Whereas, if the vehicle passes “SDX” or “OPDV” regime, it is assigned with an acceleration rate $+B_{null}$ [10,36]. The B_{null} can be estimated as:

$$B_{null} = BNULL_{mult} * (RND4_n + NRND) \quad (14)$$

where $BNULL_{mult}$ is a calibration parameter with typical values shown in Table 2 and the $RND4_n$ is a normally distributed driver-dependent parameter that represents the driver’s capability to govern acceleration. For free-driving behavior, the driver is not influenced by the surrounding traffic environment and drives at the maximum acceleration in order to attain the desired speed [10,36]. The maximum acceleration for a passenger car B_{max} is given as:

$$B_{max} = BMAX_{mult} * (v_{MAX} - V * FaktorV) \quad (15)$$

$$FaktorV = \frac{v_{MAX}}{v_{DES} + Faktormult * (v_{MAX} - v_{DES})} \quad (16)$$

in which, v_{MAX} and v_{DES} are the maximum and desired speed, respectively, and $FaktorV$ is a calibration parameter.

For the closing-in (approaching) behavior, the driver is in the “SDV” threshold and realizes that he or she is approaching a slower vehicle. A deceleration is required to avoid the collision [10,36] as:

$$B_n = \frac{1}{2} * \frac{(\Delta v)^2}{ABX - (\Delta x - L_{n-1})} + B_{n-1} \quad (17)$$

where B_{n-1} is the acceleration of the leader. For the breaking behavior (emergency), the following driver experience a sudden decrease in the front-to-rear distance as compare to “ABX” because of the dangerous situation created by the leading vehicle. In such a scenario, the following driver applies a maximum deceleration B_{MIN} to avoid vehicle crashes, which can be defined as:

$$B_n = \frac{1}{2} * \frac{(\Delta v)^2}{AX - (\Delta x - L_{n-1})} + B_{n-1} + B_{MIN} * \frac{ABX - (\Delta x - L_{n-1})}{BX} \quad (18)$$

$$B_{MIN} = -BMINadd - BMINmult * RND3_n = BMINmult * v_n \quad (19)$$

where $BMINadd$ and $BMINmult$ are calibration parameters with typical values shown in Table 2 and $RND3_n$ is a normally distributed random driver parameter.

2.4. The Optimal Velocity Model (OVM) Car-Following Model

Traffic flow observes instabilities which usually creates congestion in highways and freeways. A dynamic model is required to model this behavior of instability. The optimal velocity model (OVM) describes the dynamical formulation of traffic congestions using the equation of motion for each vehicle. In this traffic flow model, the stimulus is a function of the following headway and all the drivers possess constant sensitivities. Three assumptions are made by Bando [37] for the OVM model, including (1) each driver responds to a stimulus from the preceding vehicle and has a legal velocity of V , (2) each driver controls the acceleration/deceleration observing the motion of the preceding vehicle to maintain a safe velocity, and (3) no response-lag time is considered. The OVM dynamical acceleration of the following vehicle can be obtained as [37]:

$$\ddot{x}_n(t) = a \{ V(\Delta x_n) - \dot{x}_n(t) \} \quad (20)$$

where $\Delta x_n = x_{n+1} - x_n$, is the headway of vehicle n (following) and $n + 1$ (preceding), t is the time, n is the total number of vehicles, a is the driver’s sensitivity constant (independent of the vehicle), x_n is the position for the n th vehicle with respect to time (t) and V is optimal velocity of a vehicle n . The calculated optimal velocity $V(\Delta x_n)$ of a vehicle (n) depends on the following distance of the preceding vehicle number ($n + 1$).

Based on Equation (20), it can be seen that the velocity (V) of a vehicle increases as the headways (Δx_n) gets larger but does not surpass the maximum allowable velocity. Similarly, the velocity reduces to minimal value as the headways get narrower to prevent the crash. After the numerical simulations of the OVM model and by considering the stable and unstable cases [37], the optimal velocity function $V(\Delta x_n)$ can be defined as:

$$V(\Delta x_n) = \tanh(\Delta x - 2) + \tanh(2) \quad (21)$$

Using the above equation, the driver manages the acceleration and braking behavior and would never pass the preceding vehicle in any case. This model ensures that traffic flow creates congestion phenomena instead of accidents. A noticeable weakness of the OVM outlined through a research paper [35] is that the model develops idealistically large acceleration in some conditions.

The Intelligent Driver Model (IDM) Car-Following Model

A car-following model for mixed traffic conditions is named the intelligent driver model (IDM). This model describes the acceleration as a function of the gap $S_\alpha(t)$, the speed $v_\alpha(t)$, and the speed difference $\Delta v_\alpha(t)$ between vehicle “ α ” and the vehicle in front [38] using the following expressions:

$$\frac{d}{dt}v_\alpha(t) = \alpha \cdot \left(1 - \left(\frac{v_\alpha}{v_0} \right)^\delta - \left(\frac{S^* v_\alpha \cdot \Delta v_\alpha}{S_\alpha} \right)^2 \right) \quad (22)$$

where “ δ ” is the acceleration component and the desired gap “ S^* ”, which can be given as:

$$S^*(v_\alpha, \Delta v_\alpha) = S_o + S_1 \cdot \sqrt{\frac{v_\alpha}{v_0}} + T \cdot v_\alpha + \frac{v_\alpha \cdot \Delta v_\alpha}{2\sqrt{a \cdot b}} \quad (23)$$

in which a is the maximum acceleration, b is the maximum deceleration, S_o is the stopping distance, T is the minimum time headway, and v_0 is the free speed. The above expression has been simplified in [39] and described as:

$$\frac{d}{dt}v_\alpha = a \left(1 - \left(\frac{v_\alpha}{\text{desired velocity}} \right)^\delta - \left(\frac{\text{minimum desired gap}}{\text{gap}} \right)^2 \right) \quad (24)$$

$$S^*(v_\alpha, \Delta v_\alpha) = S_o + S_1 \cdot \sqrt{\frac{v_\alpha}{\text{desired velocity}}} + T \cdot v_\alpha + \frac{v_\alpha \cdot \Delta v_\alpha}{2\sqrt{a \cdot b}} \quad (25)$$

As shown above, the IDM model considers factored maximum acceleration and minimum headway to achieve the desired velocity and to maintain a minimum gap with the lead vehicle, respectively. Researchers have formulated the potential following behavior of autonomous vehicles, advanced driver assistance features such as adaptive cruise control (ACC), and cooperative adaptive cruise control (CACC) [40] using the IDM algorithms as a basis of model construction.

2.5. The Adaptive Cruise Control (ACC) and Cooperative Adaptive Cruise Control (CACC) Models

In the ACC system, the following vehicle records the vehicular characteristics of the preceding vehicle, such as headway speed, acceleration, and automatically adjusts its speed and controls or maintains a safe gap [41]. The ACC system provides convenience to the following driver by reducing the stresses and fatigues for continuously matching the vehicle’s speed with the preceding vehicle [9]. The following vehicle in the ACC system evaluates the leading car in the same lane through cameras, laser, or radar sensors [9,42,43]. If the desired gap between the follower and the leader is fading, such as the preceding vehicle decelerates or travel at a low speed, the following vehicle will adjust its speed to maintain a safe desired gap. Similarly, if the preceding vehicle accelerates or moves to another lane, the following vehicle likewise accelerates in the presence of clear space ahead.

The CACC technology is an extension of ACC, which can communicate with multiple vehicles through effective V2V communication systems [44] and therefore forming a vehicle platoon. The first vehicle in a platoon is a leader that transmits a message about the recommended cruising speed (and often lane-assignment) to the following vehicles. The followers record the signal and adjust their speed and gap according to the recommended values without the contribution of the driver [43]. The CACC system provides traffic stability and improvements to the throughput by increasing the network capacity up to a lane drop [9]. The concept of CACC is in the development stage, and therefore, no consensus or definition is fixed at this stage. The implementation of CACC logic under the simulation environment has shown smooth, safe, and efficient traffic operations from different research work.

Van Arem defines ACC/CACC using automated longitudinal control combined with intervehicle communications [8,9]. Each objected vehicle in the traffic stream is occupied by a leading vehicle and the following vehicle, i.e., the fourth vehicle, follows the third vehicle but leads the fifth vehicle in the platoon. The acceleration model for ACC/CACC is provided as [9,45]:

$$a_c = k_a * a_p + k_v * (v_p - v_f) + k_s * (s - v * t_d) \quad (26)$$

$$a = \max[a_{min}, \min(a_c, a_{max})] \quad (27)$$

where, a_c is the control acceleration with the liner function, a is the acceleration in the next step of the objective vehicle, a_p is the acceleration of the preceding vehicle, v_p is the speed of the preceding vehicle, v_f is the speed of the following vehicle, a_{min} is the minimum allowed acceleration (-3 m/s^2), a_{max} is the maximum allowed acceleration (2 m/s^2), and k_a, k_v, k_s are constant factors ($k_a = 0$; for ACC).

2.6. Lane-Change Models

The extent of possibility or prohibition for vehicles departing from the current lane on a multi-lane road network is defined by a lane-change model in a micro-simulation tool [6]. The lane-change logic is followed by the gap acceptance models because when the driver finds a safe gap (critical gap), the lane-change rule triggers immediately in order to reach the desired destination. In 1986, Gipps introduced a famous lane-change model for urban driving and maneuvering conditions that covered the impacts of traffic signals, obstructions, and the presence of heavy vehicles in a mixed traffic stream. The Gipps models focus on the risk of the possible collision between vehicle to vehicle, vehicle to obstructions, and other logical driving behaviors. The driver's decisions for a lane-change are based on a "set of rules" [46] as shown in Table 4 and categorized into two classes such as mandatory lane-change (or necessary lane-change) and discretionary lane-change (or free lane-change) [47] as seen in Table 3.

Table 3. Lane change classification in micro-simulation tools [47].

Mandatory Lane-Change	Discretionary Lane-Change
The current lane is blocked for a specific vehicle type.	The driver overtakes another slow-moving or heavy vehicle.
The current lane will drop by merging into another lane.	A driver yields its way to provide a safe gap for another merging vehicle.
The driver must change a lane to reach the desired destination.	

Table 4. Set of rules for Gipps lane-change model [48].

Situation	Decision
Is it possible to change a lane safely without meeting the risk of a collision?	The driver looks for a safe gap and speed of vehicles (front and back) in the target lane.
Where are the permanent locations of obstructions and how to avoid them?	The driver avoids blocked lanes or dropping lanes.
Are there any dedicated lanes for transit and high occupancy vehicles (HOV)?	The driver does not consider transit or HOV lanes to avoid being trapped.
When to consider a lane-change for the intended turning movements?	The driver inspects the distance from the intended turning and makes a necessary lane change. It becomes critical when the threshold is crossed and it is not safe for the driver to take a turn without a possible collision in the adjacent lanes.
The presence of heavy vehicles in urban traffic?	Heavy vehicles drive at slower speeds in mixed traffic. Hence, the driver maintains a substantial gap from heavy vehicles to ease the lane-change procedure and also to avoid being trapped behind it.
Is it useful to change lanes for a potential gain in the average speed?	This driver gains potential speed by either lane changing or overtaking a slow-moving vehicle.

Another well-known lane-change model was proposed by Sparmanns in 1978 that categorized the behavior of lane-change as slower-to-faster and faster-to-slower lanes depending upon the desire and need of the driver. A driver would move to a faster lane if the front vehicle in the actual lane is moving at a slower speed (as compared to the subject vehicle) or there is any other obstruction in the roadway. The potential obstruction is a function of speed differences between the current speed of the lead vehicle and the desired speed of the subject vehicle. Whereas, a driver would move to a slower lane if the following faster vehicle in the actual lane needs to overtake the subjected slow-moving vehicle or a general desire of the driver to keep the car on the right-side [49]. It is important to have safety as a priority element before the initiation of the lane-changing process, such as estimating optimum distances and speed differences of the preceding and the following vehicle on the particular lanes.

3. Micro-Simulation Modeling Tools for Human-Driven and Autonomous Vehicles

A number of traffic micro-simulation packages are available commercially in the market that simulate various real-world network configurations, problems, and solutions. Each computer simulation software package employs different car-following behaviors, lane-change, and gap-acceptance models, as discussed in the previous section. The four most popular simulation tools are the Advanced Interactive Microscopic Simulator for Urban and Non-urban Network (AIMSUN), the CORridor SIMulation (CORSIM), the PARAMICS, and the Verkehr In Städten –SIMulationsmodell (VISSIM) [6,10,50].

The AIMSUN uses the Gipps' safety distance models and developed by Traffic Simulation System, Barcelona, Spain. This tool is potentially famous for modeling traffic dynamic assignment, incident management, and ITS applications such as ramp metering and vehicle guidance systems [31]. The AIMSUN is mostly used in Europe, but now a wide-use is observed in the U.S. which can simulate urban streets, freeways, interchanges, and roundabouts [50]. Additionally, one of the reasons for the selection of this tool by modeling practitioners is its strong 3-D animation capabilities. Recently, the AIMSUN has introduced capabilities for modeling autonomous and connected vehicles.

The CORSIM software was developed in 1988 by the US Federal Highway Administration (FHWA) by combining several previous micro-simulation tools such as the NETSIM, which can simulate urban traffic streams whereas and the FRESIM (FREEway SIMulation), which can simulate complex freeway networks. The car-following logic of the NETSIM is that the lead vehicle moves to a new position (i.e., coordinate) as the simulation time proceeds with one time-step. The following vehicle is then moved to a new location in a way that reads the lead vehicle, such as if the lead vehicle decelerates at the maximum deceleration limit, the following vehicle will stop at a speed of zero to prevent the vehicle crash with the lead vehicle [51]. The FRESIM uses the Pitt car-following model that can be considered as a stimulus-response model. The logic for this model works idealistically the same as the GM-type model such that the following vehicle always maintains a safe space-headway [52]. The CORSIM has the most application in the US because of its reliability for driving behaviors and vehicle performance models [50].

The PARAMICS tool was developed by the UK Department for Transportation (DfT) in 1990. This software uses the Fritzsche car-following model that is an acceleration model based on psychophysical logic [53]. The Fritzsche model records human perception in five different regimes, including the free following, the following 1, the following, the closing in, and the danger. Each regime has a specific threshold value in the relative speed/space (DX/DV) diagram for the psychophysical follower-leader pair [10]. For example, the driver will only react to the vehicle if the change in speed falls within the specified threshold.

The PTV VISSIM is one of the most predominant micro-simulation tools developed by a German company PTV Vision (Planung Transport Verkehr) in 1992. The VISSIM uses a psychophysical car-following model which was first developed by Wiedemann in 1974 [22] and further enhanced in 1992 by Wiedemann and Reiter [36], which is now called Wiedemann 99 car-following model. This simulation tool is capable of performing complex

network and capacity analysis, including signalized junctions, transit operations, passenger commute, corridor modeling, multimodal systems, active traffic management, emission modeling, connected vehicles, and facility operations such as airport and terminals [11]. In VISSIM, the Wiedemann model decides how the following vehicle behaves based on the distance and difference of speed to the vehicle ahead. The model used numerous random numbers, statistical distributions, and stochastic variations that ensure each car has different behavior in each time-step [54]. Similarly, the lane-change algorithm works on the rule-based model as explained earlier [48,49]. This software package allows the user to adjust the parameters of lane-change, gap-acceptance, and car-following models. Specifically, the primary benefits of using the VISSIM micro-simulation tool can be summarized as (1) built-in features for connected autonomous vehicles, (2) effective evaluation techniques for capacity and safety, (3) allowing the user to calibrate the car-following model parameters, (4) application of external driver models (e.g., replacing internal default car-following model with a user-defined car-following algorithm), (5) use of a psychophysical car-following model as it is based on the assumptions and perceptions of the driver behavior, (6) robustness of the graphic user-interface, and (7) effective evaluation techniques for capacity and safety. Thus, the VISSIM simulation tools show potentials for wide application compared with other commercially available simulation tools in the market and the following of this section focuses on the review of car-following models in the VISSIM software for human drivers and AVs.

3.1. Modeling Human Drivers Using VISSIM

VISSIM also provides two Wiedemann car-following models for different application conditions, including [11] the Wiedemann 74 and 99. The Wiedemann 74 model can be used for urban traffic and merging areas. The minimum desired safety distance at low-speed differences is calculated using the same equation of the ABX threshold as discussed in Section 2.3. The Wiedemann 99 can be used for freeway with no merging areas. The model parameters affect the saturation flow rate of vehicles in simulation. The behavior of autonomous modeling is possible by using the Wiedemann 99 model as it contains at least nine parameters that assist the users in conducting sensitivity tests and also for trial and error of multiple parametric combinations. A detailed description of each parameter is presented in Table 5.

Table 5. VISSIM Wiedemann 99 car-following parameters [11].

Variable	Description	Default Value
Thresholds for Safety Distance (Δx)	CC0 Standstill distance: desired minimum distance between the two standing vehicles (lead and following) at $v = 0$ [mph]. The value for this parameter is fixed.	4.92 ft (1.5 m)
	CC1 Headway time (Gap): Desired time in seconds between the two vehicles (lead and following). The higher the value, the more cautious the driver is. Thus, at a given speed v [mph], the safety distance dx_safe is computed to: $dx_safe = CC0 + CC1 \times v$. The safety distance is defined in the model as the minimum distance a driver will keep while following another car. In the case of high volumes, this distance highly influences the capacity of the network.	0.90 s
	CC2 Following variation: this is an added value to a safe distance. Restricts the longitudinal oscillation (distance difference) as a driver moves closer to the car ahead. Hence if this value is set to 30 ft, the following distance will be dx_safe and $dx_safe + 30$ ft. The default value is 13.12 ft which results in a quite stable following process	13.12 ft (4.0 m)
	CC3 Threshold for entering “Following” state: time in seconds before a vehicle starts to decelerate in order to reach the required safety distance (negative). Hence, it defines the number of seconds a driver needed to decelerate earlier reaching the safety distance (dx_safe).	−8.00 s

Table 5. Cont.

Variable		Description	Default Value
Thresholds for Speed (Δv)	CC4	Negative “Following” Threshold: defines a negative speed variation between the following process. This parameter controls the speed differences of Wiedemann’s “following-state”. A smaller value generates a more sensitive driving behavior to the acceleration or deceleration of the preceding vehicle. The default value models a potentially tight following condition for the following drivers.	−0.35 ft/s
	CC5	Positive “Following Threshold”: defines a positive speed variation between the following process. This parameter controls the speed differences of Wiedemann’s “following-state”. A smaller value generates a more sensitive driving behavior to the acceleration or deceleration of the preceding vehicle. The default value models a potentially tight following condition for the following drivers.	0.35 ft/s
	CC6	Speed dependency of oscillation: influence of distance on speed oscillation. If set to 0, the speed oscillation is independent of the distance. Whereas larger values, lead to a greater speed oscillation with increasing distances.	11.44
Acceleration Rates	CC7	Oscillation Acceleration: minimum acceleration/deceleration during the following process.	0.82 ft/s ²
	CC8	Standstill acceleration: desired acceleration when starting from a standstill (limited by maximum acceleration defined within acceleration curves).	11.48 ft/s ²
	CC9	Acceleration at 50 mph (80 km/h): desired acceleration at 50 mph (limited by maximum acceleration defined within acceleration curves)	4.92 ft/s ²

The VISSIM lane-change model is based on Sparmann’s model as discussed in Section 2.6 [55]. Additionally, there are two types of lane-changing procedures available in the software package as the necessary lane-change (or mandatory lane change) and the free-lane change (or discretionary lane-change). In the necessary lane-change, the driver must change its lane to reach the next connector of a route. VISSIM controls this behavior by a parameter called “maximum acceptable deceleration” for lane changing vehicle (own) and the vehicle which will be its follower in the target lane (trailing vehicle) after the lane-change is completed. The free lane-change method is initiated when the vehicle demands to obtain the benefits of speed increment while maintaining the desired safety distance in the target lane. This safety distance is computed by the speed of the lane-changing vehicle (own) and the trailing vehicle in the target lane [11]. Table 6 outlines some of the important parameters that control the lane-changing logic of vehicles.

Table 6. Lane-change logic parameters [11].

Parameters	Explanation
General behavior	Defines the behavior for overtaking by two methods such as (1) free lane selection in which vehicles can overtake on each lane, (2) slow-lane rule, which allows overtaking on freeways.
Necessary lane change (route)	This parameter is subdivided into two columns for its own (subject) vehicle and a trailing (target) vehicle. It consists of three sub-components. The deceleration thresholds for the own and trailing vehicle are defined to adjust the aggressiveness of the necessary lane-change [55]. The “maximum deceleration” (upper bound value) and “acceptable deceleration” (lower bound value) are the limits of deceleration value while performing a lane change. The “−1 m/s ² per distance” is the reduction rate that defines the pace at which the maximum deceleration will reduce with the increasing distance from the emergency stop distance.

Table 6. Cont.

Parameters	Explanation
Waiting time before diffusion	When a vehicle requires to make a mandatory lane change in a congested road network to follow its desired route (when no acceptable lane-change gap is available), it comes to a halt (emergency stop), and it waits until a suitable gap for lane-change is available. VISSIM defines a threshold for the maximum amount of time a vehicle can wait at the emergency stop before making a lane-change. If the vehicle is not successful in this defined period, it is removed from the network (disappear from the simulation). VISSIM will generate a warning message (about the removal of a vehicle) when the simulation period is completed. In this way a network would not choke because of certain vehicle which comes to a halt.
Minimum clearance	The minimum distance that must be available between the lead and preceding vehicles after a lane-change.
Safety distance reduction factor	A drop in the safety distances for vehicles involved in lane-change maneuvers. Smaller the value, more aggressive lane-change.
Maximum deceleration for cooperative breaking	Defines the maximum deceleration which the trailing vehicle would accept to help the maneuver of lane-changing vehicles.
Cooperative lane-change	Through the use of this option, the trailing vehicle in the target lane would move to another side of a lane and providing room for lane change-vehicle.

3.2. Modeling of Autonomous Vehicles in VISSIM

The behavior of autonomous and connected driving is predictable and deterministic, unlike the stochastic behavior of human drivers. The existing car-following models (e.g., Wiedemann psychophysical model) and other driving behavior parameters (e.g., lane-change, gap-acceptance, acceleration/deceleration distributions, etc.) in the traffic simulation software need to be modified or replaced with external driving models for autonomous cars. This section discusses the available tools in VISSIM software (until version 2020) and associated external APIs (application programming interface) for the modeling of autonomous driving, platoon formation, connected driving, advanced driver assistance options, and recommendations for Wiedemann model driving behavior parameters (e.g., CC0 to CC9 as presented previously in Table 5). To date, many car manufacturers and other companies have already built partial autonomous vehicles, i.e., SAE (Society of Automotive Engineers) automation levels or working towards the production of fully autonomous driving. The companies are restrictive in revealing the information of car-following algorithms. The concept of autonomous driving is also in the development stage and this process varies for different manufacturers. Therefore, the simulation experiments can aid in modeling the expected driving behavior of autonomous vehicles and can inspect the associated risk before the diving logics are ready for implementation on the roads.

3.2.1. Parameters for AVs and CAVs Using Wiedemann Car-Following Model in VISSIM

Many studies have used different parametric values for the autonomous vehicle's simulation using Wiedemann's-99 car-following model. A recent study by Karlsruhe Institute of Technology-Institute for Transport Studies and PTV Group investigated the following behavior of autonomous vehicles using the Wiedemann car-following model. The study uses the data collected through the CoEXist project for three test vehicles, out of which two are real-world autonomous vehicles that drive autonomously on public roads under normal traffic conditions [56]. The leading vehicle is manually driven, whereas the following two cars are in the autonomous driving mode. The autonomous cars are equipped with two longitudinal control communications, such as CACC (autonomous car communicates with the leading vehicle) and degraded CACC (dCACC—autonomous car does not communicate with the leading vehicle). Hence the longitudinal behavior of two autonomous cars is analyzed to propose the adjustment in Wiedemann's model for autonomous driving behaviors, as shown in Table 7. It is also important to mention that

most of the urban driving would have active dACC mode because the leader is sometimes lost at a traffic light or a manual vehicle cut in the platoon formations [57].

Table 7. Parameters for AV simulation using Wiedmann 99 model [57].

Parameter	Autonomous CACC (Communication with the Leader)	Autonomous dCACC (No Communication with Leader)
CC0	4 (m)	6 (m)
CC1	0.3, 0.6, 1.0 (s)	1.0 (s)
CC2	0 (m)	0 (m)
CC3	−40 (s)	−40 (s)
CC4	0 (m/s)	0 (m/s)
CC5	0 (m/s)	0 (m/s)
CC6	0	0
CC7	0.25 (m/s ²)	0.25 (m/s ²)
CC8	3.5 (m/s ²)	3.5 (m/s ²)
CC9	1.5 (m/s ²)	1.5 (m/s ²)

A study by Atkins suggested model parameters for autonomous cars for different SAE levels (i.e., Level 2 to Level 4), as shown in Table 8. The level 3 AVs are subdivided into four categories depending upon the aggressiveness of the driving parameters. The parameters such as “minimum clearance (front/rear)” and the “safety distance reduction factor” are defined in Section 3.1. The other two user-defined merging behavior parameters are the minimum time-gap to measure the minimum time required by a vehicle on the mainline to achieve minimum headway at its present speed, and the minimum clearance or headway to measure the minimum headway required by a vehicle to merge into the mainline [58]. In another study, Peter et al. investigated the impacts of autonomous and conventional vehicles on the traffic flow for different market penetrations using Wiedemann-74 (for urban roads) and Wiedemann-99 (for freeway) car-following models using slightly different parameters as described in Table 8 [59].

Table 8. Parameter variation for Wiedemann model and VISSIM user-defined attributes [58].

Capability Levels	CC0 (m)	CC1 (s)	CC7 (m/s ²)	CC8 (m/s ²)	CC9 (m/s ²)	Min. Clearance or Headway (front/rear) (m)	Safety Distance Reduction Factor	User-Defined Minimum Time-Gap (s)	User-Defined Minimum Clearance or Headway (m)	
Level 2	1.5	0.9	0.25	3.5	1.5	0.5	60%	3	5	
Level 3	Cautious	2.5	1.8	0.1	3.2	1.2	0.8	90%	3.6	6.5
	Normal cautious	2	1.2	0.2	3.4	1.4	0.6	70%	3.2	5.5
	Normal assertive	1	0.8	0.3	3.6	1.6	0.4	50%	2.8	4.5
	Assertive	0.5	0.6	0.4	3.8	1.8	0.2	30%	2.4	3.5
Level 4 *	0.5	0.6	0.4	3.8	1.8	0.2	30%	2.4	3.5	

* Level 4 vehicles are subject to use a fixed desired speed distribution. Rest all the categories uses the VISSIM default desired speed distribution according to a link type.

3.2.2. CoEXist and VISSIM—Automation-Ready Framework

The CoEXist (<https://www.h2020-coexist.eu/> (accessed on 1 March 2021)) is a funded project by European Commission that prepares the concerned authorities for a transition phase during which both the automated vehicles (i.e., AVs and CAVs) and conventional vehicles will co-exist on the roadways. The basic effort for this project is to bridge a gap between emerging AV technology, transport planning, infrastructure development, and

enabling city authorities to effectively deploy AVs using the best practices [60]. The CoEXist project has simulated real examples of AVs in four European cities such as; Milton Keynes-UK, Stuttgart-Germany, Gothenburg-Sweden, and Helmond-Netherland. The outcomes of this exercise provide certain resources to the authorities for the potential impacts of automated technology and market penetration of AVs and CAVs.

The AV-ready (autonomous vehicles ready) framework develops specialized tools for microsimulation and macroscopic modeling of autonomous and connected cars. To date, VISSIM 2020 is the latest version released by the PTV Group, which contains all the updated AV related tools that are developed under the CoEXist study. However, PTV Group has initially provided autonomous features starting from VISSIM 9, VISSIM 10, VISSIM 11, and now VISSIM 2020 [11]. The following section introduces some of the autonomous features in VISSIM 2020. In the VISSIM 2020, there are some AV-related features and behavior parameters such as communication and cooperation, functions, and distributions. The communication and cooperation between vehicles (conventional or autonomous) are improved by introducing two options under the lane-change tab for driving behavior. The first option is “cooperation lane-change” which facilitates the process of a lane change in such a way that the trailing vehicle in the target lane would move to another side of a lane and providing room for lane change-vehicle as discussed in Section 2.6. The second option is “advanced merging,” where a lane-changing vehicle would initiate the process earlier so that no disruption of the traffic occurs and therefore the capacity of the network increases [11]. The vehicle cooperation can also be modeled by choosing smaller headways between vehicles through CC0, CC1, CC2, and CC6 parameters [61].

As discussed earlier, it is expected that autonomous cars will have deterministic behavior, unlike the random behavior of human drivers. Hence the spread of values for individual vehicles can be reduced for several functions such as desired acceleration, desired deceleration, maximum acceleration, and maximum deceleration [61]. Similarly, the distribution function such as the desired speed is an important factor for link capacity and travel times. In VISSIM, this desired speed value is assigned to each vehicle, and the vehicle desires to match this speed throughout the simulation network. However, due to the varying speed of other surrounding vehicles, human behavior towards accepting the risks, and congestions in the network; a large spread of desired speeds can be observed. The automated cars are expected to have a lower spread as they will obey the speed limit regularly. The conventional vehicle as a spread of desired speed values ranging from 88 km/h (54.7 mph) to 130 km/h (80.7 mph) whereas the automated vehicles have a minimal spread of values ranging from 99km/h (61.5 mph) to 100km/h (62.1 mph).

To facilitate autonomous driving, the CoEXist project has developed multiple additional tools to simulate the expected behavior of future autonomous and connected cars, which introduces three AV-ready VISSIM driving logics under driving behaviors such as cautious, normal, and aggressive including [11]:

(1) The AV cautious (CoEXist): the vehicle maneuvers in a safe manner at all times and maintains larger headways. This logic is conservative if compared with default urban (motorized) driving logic. Refer back to Table 7, which explains the values of a conventional human driver;

(2) AV normal (CoEXist): this logic behaves similarly to a default urban(motorized) driving behavior. Additionally, the vehicle can sense the vehicle’s distance to other vehicles in the range and their real-time speed;

(3) AV aggressive (CoEXist): the vehicle is “all-knowing” of the entire traffic situation, equipped with traffic situation prognosis, and maintains a minimum gap to other vehicles. This aggressive logic leads to cooperative driving.

Each above driving logic uses different parameters for Wiedemann 99 car-following model, lane-change, and following behaviors, as shown in Table 9. Nevertheless, this section has summarized most of the available internal autonomous features in VISSIM 2020 (available at the time of writing this review paper). Since this topic is getting lots of attraction from professionals in the industry, university researchers, the general public,

and others, it is expected that the PTV Group will provide more AV-ready options in the upcoming software releases and service packs.

Table 9. Driving behavior parameters for three different AV-ready logics.

Driving Parameters	CoEXist AV-Ready Driving Behaviors *		
	AV Cautious (CoEXist)	AV Normal (CoEXist)	AV Aggressive (CoEXist)
Following behavior			
Max look ahead distance (m)	250.0	250.0	300.0
Number of interaction objects	2	2	10
Number of interaction vehicles	1	1	8
Car-following (Wiedemann 99)			
CC0 (m)	1.5	1.5	1.0
CC1 (s)	1.5	0.9	0.6
CC3 (s)	−10.0	−8.0	−6.0
CC8 (m/s ²)	3.0	3.5	4.0
CC9 (m/s ²)	1.20	1.5	2.0
Lane-change			
Necessary lane change—max deceleration (m/s ²) for own vehicle	−3.5	−4.0	−4.0
Necessary lane change—max deceleration (m/s ²) for trailing vehicle	−2.5	−3.0	−4.0
Necessary lane change—1 m/s ² per distance (m)	80	100	100
Necessary lane change—accepted deceleration (m/s ²) for own vehicle	−1.0	−1.0	−1.0
Necessary lane change—accepted deceleration (m/s ²) for trailing vehicle	−1.0	−1.0	−1.5
Safety distance reduction factor	1.0	0.6	0.75
Max deceleration for cooperative braking (m/s ²)	−2.5	−3.0	−6.0
Cooperative merging	No	Yes	Yes
Autonomous driving			
Enforce absolute braking distance	Yes	No	No

* Note: The values for AV-ready driving behaviors are copied from VISSIM 2020 software.

3.3. VISSIM DLL-APIs and COM Interface for New Future AV Car-Following Model Development

Modifying the default car-following model parameters such as headway, safety distance, acceleration functions, or speed distributions is one approach to model the anticipated behavior of AVs and CAVs. However, it uses the same logic as by other conventional vehicles, i.e., Wiedemann Model 74 or 99. Further, VISSIM contains about 190 parameters for car-following, lane-change, gap-acceptance, and others which makes it very difficult to calibrate the model [62]. Another approach to model AVs and CAVs in the VISSIM software is through the use of external scripting (such as COM-Interface) and VISSIM DLL-APIs (application programming interfaces). The APIs are an add-on module which allows user to integrate external applications into the VISSIM [11] and thus enhancing the user's experience. It includes DriverModel-DLL, DrivingSimulator-DLL, EmissionModel-DLL, and SignalControl-DLL [11]. The advanced features of vehicle communication and

infrastructure automation, i.e., V2V (vehicle-to-vehicle) or V2I (vehicle-to-infrastructure) are modeled through VISSIM APIs by various researchers [45,54,61,63].

The Microsoft COM (component object model) provides communication, connection, and access to different applications. The data and functions contained in the VISSIM application can be accessed externally through the COM-Interface using different programming languages such as VBA, Python, C++, C#, MATLAB, and JAVA [64]. It can access each vehicle's attribute (e.g., current position, acceleration, orientation on the lane, vehicle's receiving traffic light status, etc.) during a simulation run and can also edit some of the parameters (e.g., arrival pattern of vehicles in the network, create or remove vehicles, etc.). As per the requirement of the user to run the scripts for a specified amount of time, VISSIM "event-based" COM scripts only activate during the user-defined simulation time value, or if required, it stays active throughout the simulation period. The software package contains a detailed instruction document of COM-Interface explaining the procedures for accessing the objects and attributes. COM-Interface assist in modeling the behavior of autonomous and connected vehicles such as platooning, shorter and steady headways, and the vehicle adjusting the speed to arrive at an intersection at the status of the green traffic light. The VISSIM package contains some AVs-related COM scripts example files for users.

This API module enables the user to completely replace the internal-car following model (i.e., Wiedemann 74 or 99) and lane-change models with a user-defined algorithm for selected or all vehicle types. The external driver model (EDM) can control the longitudinal (acceleration and deceleration), lateral (lane-change) behavior of the vehicles. The user-defined car-following logic code is programmed using C++ language, which outputs a DriverModel.cpp file. The CPP file can be compiled into a Dynamic Link Library (.DLL) file using the Microsoft Visual Studio software. The VISSIM package contains an example of a DLL file and a Visual Studio Project (.vcproj) that contains an example code structure. The DLL file is loaded through the "external driver model" dialog box window under the "vehicle type" tabs.

During a simulation run, VISSIM calls the DLL code for each vehicle using the EDM for each simulation time-step and decides the behavior of the vehicle. The current state of the affected vehicle and surrounding vehicles is delivered to DLL, and in return, the user-defined algorithm computes the longitudinal behavior, i.e., acceleration/deceleration values, and lateral behavior, i.e., lane-change values. Finally, the DLL file passes the computed values back to VISSIM in the current simulation-time step [65], as shown in Figure 2 [45]. The DriverModel-DLL is a potential tool for modeling the autonomous and connected behavior of vehicles such as cruise controls, platooning, and getting real-time traffic data of surrounding vehicles, e.g., headway, current speed, current acceleration, and current lane.

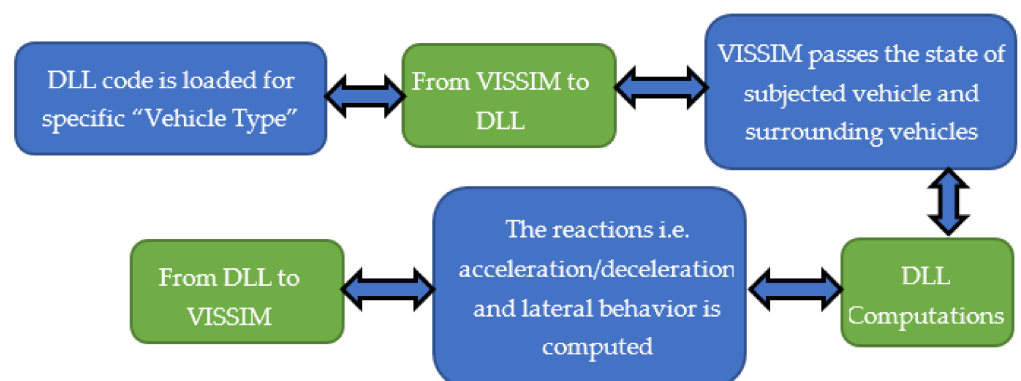


Figure 2. VISSIM and External DriverModel-DLL working procedure.

Zhao et al. developed an external driver model using ACC/CACC framework through VISSIM API, i.e., DriverModel-DLL [45]. The following steps explain the procedure of implementing the ACC/CACC DLL files into the VISSIM including:

- (1) Development of a VISSIM simulation network (e.g., links, connectors, etc.).
- (2) Writing code in C++ for ACC and CACC.
- (3) Compiling the code into a DLL file.
- (4) The DriverModel-DLL is loaded into VISSIM for a specific vehicle type.
- (5) VISSIM reads the DLL logic for each time step.
- (6) Using the “Set” function, the driver behavior is sent from VISSIM to DLL for every time-step.
- (7) Using the “Get” function, the update driver behavior is sent from DLL to VISSIM.
- (8) Using the “Execute” function, VISSIM moves the driver (i.e., DLL affected vehicle)

The driving simulator interface connects the VISSIM software to a real-world driving simulator (DS) hardware (e.g., containing steering and wheel pedals) or any other external vehicle simulation program which runs using its own algorithms (for a single vehicle or multiple vehicles). For example, a DS controls a specific vehicle type in a network while the rest of the vehicles are using the VISSIM default car-following and lane-change behaviors. The information is exchanged between the VISSIM and a DS for each simulation time-step. VISSIM provides the information for the surrounding traffic (e.g., position, acceleration/deceleration, speed, etc.) to the DS, and in return, the DS provides back the current position and orientation of the simulator vehicles [66]. The DS provides an opportunity for the users to model AVs and CAVs using the data from the real-world autonomous cars and built-in sensors.

4. Conclusions

This paper presents a review of the car-following models used in prevalent micro-simulation tools with the modeling capabilities of human and autonomous driving behaviors. The conclusions of the review of this paper can be summarized as below:

- Autonomous and connected cars would observe a longer transition period during which the robotic and the conventional human drivers will share the roadways, thus creating mixed traffic, until achieving complete market penetration. The micro-simulation tool provides an opportunity for the user and policymakers to investigate the anticipated impacts of mixed traffic by either modifying the internal car following model, i.e., driving behavior parameters or implement an external smart logic for AVs and CAVs.
- The car-following models for human drivers can be classified into three categories including the GM-type models such as Gazis–Herman–Rothery (GHR) model, the safety-distance such as the Gipps model, and the psychophysical car-following such as Wiedemann models. The GM model provides a basis for all other car-following models. For AVs and CAVs, dynamic car-following model, i.e., OVM, and a mixed condition car-following model, i.e., IDM, the ACC/CACC models can be used. It is noticeable that researchers have implemented the cruise control approach for modeling the AVs and CAVs.
- Among various simulation tools, the VISSIM simulation tool is a popular tool that uses a psychophysical car-following model, i.e., Wiedemann 74 and Wiedemann 99. A user can model the expected behavior of AVs/CAVs by modifying the Wiedemann model parameters and other driving behaviors. In addition, the dedicated features for autonomous driving were also provided in the recent version of VISSIM, such as “AV-ready” through which a user can effectively model autonomous behaviors such as platooning, communication between vehicles and infrastructures, and others. The DLL-APIs and COM interface is an additional tool that can completely replace the internal car-following model with a user-defined future AV-car following model. The COM interface, on the other hand, enables the users to perform operations that are not possible in a typical GUI of the software.

The developments in the field of driving behaviors for future autonomous and connected cars are in the evolution stage. The micro-simulation tools can provide a potential platform for researchers to investigate and implement safe and effective driving logics for

AVs and CAVs. Interests also exist in using the simulation tools to investigate the impacts of autonomous and connected vehicles on traffic flow and safety.

Author Contributions: Conceptualization, H.U.A. and Y.H.; methodology, H.U.A. and Y.H.; formal analysis, H.U.A.; investigation, H.U.A.; data curation, H.U.A.; writing—original draft preparation, H.U.A.; writing—review and editing, Y.H.; supervision, Y.H. and P.L.; project administration, Y.H. and P.L.; funding acquisition, Y.H. and P.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by U.S. Department of Transportation under the grant number of No. 69A3551747108 through MPC project No. 547.

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

References

1. Litman, T. *Autonomous Vehicle Implementation Predictions*; Victoria Transport Policy Institute: Victoria, BC, Canada, 2017.
2. Bierstedt, J.; Gooze, A.; Gray, C.; Peterman, J.; Raykin, L.; Walters, J. *Effects of Next-Generation Vehicles on Travel Demand and Highway Capacity*; FP Think Working Group: Washington, DC, USA, 2014; Volume 8, pp. 10–11.
3. Silberg, G.; Wallace, R.; Matuszak, G.; Plessers, J.; Brower, C.; Subramanian, D. *Self-Driving Cars: The Next Revolution*; White Paper; KPMG LLP & Center of Automotive Research: Ann Arbor, MI, USA, 2012; Volume 9, pp. 132–146.
4. Kotsialos, A.; Papageorgiou, M.; Diakaki, C.; Pavlis, Y.; Middelham, F. Traffic flow modeling of large-scale motorway networks using the macroscopic modeling tool METANET. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 282–292. [\[CrossRef\]](#)
5. Lu, X.-Y.; Qiu, T.Z.; Varaiya, P.; Horowitz, R.; Shladover, S.E. Combining variable speed limits with ramp metering for freeway traffic control. In Proceedings of the 2010 American Control Conference, Baltimore, MD, USA, 30 June–2 July 2010; pp. 2266–2271.
6. Lochrane, T. A New Multidimensional Psycho-Physical Framework for Modeling Car-Following in a Freeway Work Zone. Ph.D. Thesis, University of Central Florida, Orlando, FL, USA, 2014. (In German).
7. U.S. Department of Transportation Federal Highway Administration. *Types of Traffic Analysis Tools*; US DOT: Washington, DC, USA, 2020.
8. Van Arem, B.; Tampere, C.M.; Malone, K. Modelling traffic flows with intelligent cars and intelligent roads. In Proceedings of the IEEE IV2003 Intelligent Vehicles Symposium, Proceedings (Cat. No. 03TH8683), Columbus, OH, USA, 9–11 June 2003; pp. 456–461.
9. Van Arem, B.; Van Driel, C.J.; Visser, R. The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Trans. Intell. Transp. Syst.* **2006**, *7*, 429–436. [\[CrossRef\]](#)
10. Olstam, J.J.; Tapani, A. *Comparison of Car-Following Models*; Swedish National Road and Transport Research Institute: Linköping, Sweden, 2004; Volume 960.
11. PTV Group. *PTV VISSIM 2020 User Manual*; PTV Group: Karlsruhe, Germany, 2019.
12. Treiber, M.; Kesting, A. Traffic flow dynamics. In *Traffic Flow Dynamics: Data, Models and Simulation*; Springer: Berlin/Heidelberg, Germany, 2013.
13. Hoogendoorn, S.P.; Hoogendoorn, R. Generic calibration framework for joint estimation of car-following models by using microscopic data. *Transp. Res. Rec.* **2010**, *2188*, 37–45. [\[CrossRef\]](#)
14. Viti, F.; Wolput, B.; Tampere, C.M.; Vandervelden, P. Dynamic modeling of VISSIM’s critical gap parameter at unsignalized intersections. *Transp. Res. Rec.* **2013**, *2395*, 12–20. [\[CrossRef\]](#)
15. Toledo, T.; Koutsopoulos, H.N.; Ben-Akiva, M. Integrated driving behavior modeling. *Transp. Res. Part C Emerg. Technol.* **2007**, *15*, 96–112. [\[CrossRef\]](#)
16. Gazis, D.C.; Herman, R.; Rothery, R.W. Nonlinear follow-the-leader models of traffic flow. *Oper. Res.* **1961**, *9*, 545–567. [\[CrossRef\]](#)
17. Koutsopoulos, H.N.; Farah, H. Latent class model for car following behavior. *Transp. Res. Part B Methodol.* **2012**, *46*, 563–578. [\[CrossRef\]](#)
18. Chandler, R.E.; Herman, R.; Montroll, E.W. Traffic dynamics: Studies in car following. *Oper. Res.* **1958**, *6*, 165–184. [\[CrossRef\]](#)
19. Herman, R. Car-following and steady state flow. In Proceedings of the Theory of Traffic Flow Symposium Proceedings, Warren, MI, USA, 13 February 1959; Elsevier: New York, NY, USA, 1961; pp. 1–13.
20. Subramanian, H. Estimation of Car-Following Models. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1996.
21. Ahmed, K.I. Modeling Drivers’ Acceleration and Lane Changing Behavior. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1999.
22. Wiedemann, R. *Simulation des Straßenverkehrsflusses in Schriftenreihe des Instituts für Verkehrswesen der Universität Karlsruhe*; Universität Karlsruhe: Karlsruhe, Germany, 1974.
23. Leutzbach, W. *Introduction to the Theory of Traffic Flow*; Springer: Berlin, Germany, 1988; Volume 47.
24. Brackstone, M.; McDonald, M. Car-following: A historical review. *Transp. Res. Part F Traffic Psychol. Behav.* **1999**, *2*, 181–196. [\[CrossRef\]](#)
25. Pipes, L.A. An operational analysis of traffic dynamics. *J. Appl. Phys.* **1953**, *24*, 274–281. [\[CrossRef\]](#)

26. Gipps, P.G. Behavioral car-following model for computer simulation. *Transport. Res.* **1981**, *15*, 105–111. [\[CrossRef\]](#)
27. Vasconcelos, L.; Neto, L.; Santos, S.; Silva, A.B.; Seco, Á. Calibration of the Gipps car-following model using trajectory data. *Transp. Res. Procedia* **2014**, *3*, 952–961. [\[CrossRef\]](#)
28. Rakha, H.; Wang, W. Procedure for calibrating Gipps car-following model. *Transp. Res. Rec.* **2009**, *2124*, 113–124. [\[CrossRef\]](#)
29. Punzo, V.; Tripodi, A. Steady-state solutions and multiclass calibration of Gipps microscopic traffic flow model. *Transp. Res. Rec.* **2007**, *1999*, 104–114. [\[CrossRef\]](#)
30. Wilson, R.E. An analysis of Gipps's car-following model of highway traffic. *IMA J. Appl. Math.* **2001**, *66*, 509–537. [\[CrossRef\]](#)
31. Barceló, J.; Casas, J. Dynamic network simulation with AIMSUN. In *Simulation Approaches in Transportation Analysis*; Springer: Boston, MA, USA, 2005; pp. 57–98.
32. Liu, R. Traffic simulation with DRACULA. In *Fundamentals of Traffic Simulation*; Springer: New York, NY, USA, 2010; pp. 295–322.
33. *SISTM: A Motorway Simulation Model*; Leaflet, LF2061; Transportation Research Laboratory: Berks, UK, 1993.
34. Silcock, J. *SIGSIM Version 1.0 Users Guide*; User Guide University of London Centre for Transport Studies: London, UK, 1993.
35. Lazar, H.; Rhoulami, K.; Rahmani, D. A Review Analysis of Optimal Velocity Models. *Period. Polytech. Transp. Eng.* **2016**, *44*, 123–131. [\[CrossRef\]](#)
36. Wiedemann, R.; Reiter, U. *Microscopic Traffic Simulation: The Simulation System MISSION, Background and Actual State*; Project ICARUS (V1052) Final Report; CEC: Brussels, Belgium, 1992; Volume 2, pp. 1–53.
37. Bando, M.; Hasebe, K.; Nakayama, A.; Shibata, A.; Sugiyama, Y. Dynamical model of traffic congestion and numerical simulation. *Phys. Rev. E* **1995**, *51*, 1035. [\[CrossRef\]](#)
38. Treiber, M.; Hennecke, A.; Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **2000**, *62*, 1805. [\[CrossRef\]](#)
39. Raju, N.; Arkatkar, S.; Joshi, G. Evaluating performance of selected vehicle following models using trajectory data under mixed traffic conditions. *J. Intell. Transp. Syst.* **2019**, *24*, 617–634. [\[CrossRef\]](#)
40. 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. [\[CrossRef\]](#)
41. Nowakowski, C.; Shladover, S.E.; Cody, D.; Bu, F.; O'Connell, J.; Spring, J.; Dickey, S.; Nelson, D. *Cooperative Adaptive Cruise Control: Testing Drivers' Choices of Following Distances*; University of California, Berkeley: Berkeley, CA, USA, 2011; ISSN 1055-1425.
42. Hunter, M.P.; Guin, A.; Rodgers, M.O.; Huang, Z.; Greenwood, A.T. *Cooperative Vehicle-Highway Automation (CVHA) Technology: Simulation of Benefits and Operational Issues*; United States Federal Highway Administration: Washington, DC, USA, 2017.
43. Kockelman, K.; Avery, P.; Bansal, P.; Boyles, S.; Bujanovic, P.; Choudhary, T.; Clements, L.; Domnenko, G.; Fagnan, D.; Helsel, J. *Implications of Connected and Automated Vehicles on the Safety and Operations of Roadway Networks*; Final Report; Texas Department of Transportation: Austin, TX, USA, 2016.
44. Şentürk, M.; Uygan, I.M.C.; Güvenc, L. Mixed cooperative adaptive cruise control for light commercial vehicles. In Proceedings of the 2010 IEEE International Conference on Systems, Man and Cybernetics, Istanbul, Turkey, 10–13 October 2010; pp. 1506–1511.
45. 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. [\[CrossRef\]](#)
46. Hidas, P. Modelling lane changing and merging in microscopic traffic simulation. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 351–371. [\[CrossRef\]](#)
47. Hwang, S.Y.; Park, C.H. Modeling of the gap acceptance behavior at a merging section of urban freeway. In Proceedings of the Eastern Asia Society for Transportation Studies, Tokyo, Japan, 26 October 2005; Eastern Asia Society for Transportation (EASTS): Tokyo, Japan, 2005; p. e1656.
48. Gipps, P.G. A model for the structure of lane-changing decisions. *Transp. Res. Part B Methodol.* **1986**, *20*, 403–414. [\[CrossRef\]](#)
49. Sparmann, U. *Spurwechselvorgänge auf Zweispurigen BAB-Richtungsfahrbahnen*; Bundesminister für Verkehr, Abt. Straßenbau: Berlin, Germany, 1978.
50. Jones, S.L.; Sullivan, A.J.; Cheekoti, N.; Anderson, M.D.; Malave, D. *Traffic Simulation Software Comparison Study*; UTCA Report 02217; University Transportation Center for Alabama: Tuscaloosa, AL, USA, 2004.
51. Zhang, Y. Scalability of Car-Following and Lane-Changing Models in Microscopic Traffic Simulation Systems. Master's Thesis, Louisiana State University, Baton Rouge, LA, USA, 2004.
52. Soria, I.; Elefteriadou, L.; Kondyli, A. Assessment of car-following models by driver type and under different traffic, weather conditions using data from an instrumented vehicle. *Simul. Model. Pract. Theory* **2014**, *40*, 208–220. [\[CrossRef\]](#)
53. Fritzsche, H.-T. A model for traffic simulation. *Traffic Eng. Control* **1994**, *35*, 317–321.
54. Fyfe, M.R. Safety Evaluation of Connected Vehicle Applications Using Micro-Simulation. Master's Thesis, University of British Columbia, Vancouver, BC, Canada, 2016.
55. Fransson, E. Driving Behavior Modeling and Evaluation of Merging Control Strategies—A Microscopic Simulation Study on Sirat Expressway. Master's Thesis, Linköping University, Linköping, Sweden, 2018.
56. Sukennik, P.; Zeidler, V.; Etten, J.V. *CoEXist Deliverable D 2.6 Technical Report on Data Collection and Validation*; Linköping University: Linköping, Sweden, 2018.
57. Zeidler, V.; Buck, H.S.; Kautzsch, L.; Vortisch, P.; Weyland, C.M. Simulation of Autonomous Vehicles Based on Wiedemann's Car Following Model in PTV Vissim. In Proceedings of the 2019 98th Annual Meeting of the Transportation Research Board (TRB), Washington, DC, USA, 13–17 January 2019.

-
58. *Research on the Impacts of Connected and Autonomous Vehicles (CAVs) on Traffic Flow, Summary Report, Department for Transport*; ATKINS: Epsom, UK, 2016.
 59. Sukennik, P.; Lohmiller, J.; Schlaich, J. Simulation-Based Forecasting the Impacts of Autonomous Driving. In Proceedings of the International Symposium of Transport Simulation (ISTS'18) and the International Workshop on Traffic Data Collection and its Standardization (IWTDCS'18), Ehime, Japan, 3 August 2018.
 60. Rupperecht, S.; Backhaus, W.; Gyergyay, B.; Gomari, S. CoEXist Deliverable D1. 1 Automation Ready Framework. 2018. Available online: <https://www.h2020-coexist.eu/automation-ready-framework-2/> (accessed on 1 February 2021).
 61. Sukennik, P.; Micro-Simulation Guide for Automated Vehicles—Final. Deliverable 2.11 of the CoEXist Project. 2020. Available online: <https://www.h2020-coexist.eu/wp-content/uploads/2020/04/D2.11-Guide-for-the-simulation-of-AVs-with-microscopic-modelling-tool-Final.pdf> (accessed on 1 February 2021).
 62. Essa, M.; Sayed, T. Simulated traffic conflicts: Do they accurately represent field-measured conflicts? *Transp. Res. Rec.* **2015**, *2514*, 48–57. [[CrossRef](#)]
 63. Papadoulis, A.; Quddus, M.; Imprialou, M. Evaluating the safety impact of connected and autonomous vehicles on motorways. *Accid. Anal. Prev.* **2019**, *124*, 12–22. [[CrossRef](#)]
 64. PTV Group. *PTV VISSIM 2020 Introduction to the COM API*; PTV Group: Karlsruhe, Germany, 2019.
 65. PTV Group. *PTV VISSIM 2020 Driver Model DLL—Interface Documentation*; PTV Group: Karlsruhe, Germany, 2019.
 66. PTV Group. *PTV VISSIM 2020 Driver Model DLL—Driving Simulator Interface Documentation*; PTV Group: Karlsruhe, Germany, 2019.