sustainability

Article

The Action Point Angle of Sight: A Traffic Generation Method for Driving Simulation, as a Small Step to Safe, Sustainable and Smart Cities

Minh Sang Pham Do [1,*], Ketoma Vix Kemanji [2], Man Dinh Vinh Nguyen [1], Tuan Anh Vu [1] and Gerrit Meixner [2]

1 Vietnamese-German Transport Research Centre, Vietnamese-German University, Ben Cat Town 75000, Vietnam
2 Usability and Interaction Technology Laboratory, Heilbronn University, 74081 Heilbronn, Germany
* Correspondence: sang.pdm@vgu.edu.vn; Tel.: +84-902-30-30-30

Abstract: Computer simulations of traffic and driving provide essential solutions to reduce risk and cost in traffic-related studies and research. Through nearly 90 years of simulation development, many research projects have attempted to improve the various aspects of realism through the use of traffic theory, cameras, eye-tracking devices, sensors, etc. However, the previous studies still present limitations, such as not being able to simulate mixed and chaotic traffic flows, as well as limited integration/interoperability with 3D driving simulators. Thus, instead of reusing previous traffic simulators, in this paper, we define relevant concepts and describe the development and testing of a novel traffic generator. First, we introduce realistic aspects to improve traffic generation, including interactive physics (i.e., interactions based on physics among the vehicles, infrastructure, and weather) and natural traffic behaviors (e.g., road user behaviors and traffic rules), allowing the self-driving vehicle behaviors to mimic human behaviors under stochastic factors such as random vehicles and speed. Second, we gain experiences from the technical deficiencies of existing systems. Third, we propose methods for traffic generation based on the action point angle of sight (APAS) formula, which adheres to these constraints and is interoperable with modern driving simulators. We also conducted quantitative evaluations in two experiments (comprising 250 trials), in order to prove that the proposed solution can effectively simulate mixed traffic flows. Moreover, the approaches presented in this study can help self-driving cars to find their way at an intersection/T-junction, as well as allowing them to steer automatically after an accident occurs. The results indicate that traffic generation algorithms based on these new traffic theories can be effectively implemented and used in modern driving simulators and multi-driving simulators, outperforming previous traffic generators based on repurposed technologies.

Keywords: mixed traffic flow simulation; traffic generator; action point angle of sight; artificial traffic algorithms; traffic flow theory; social force model

check for updates

Citation: Pham Do, M.S.; Kemanji, K.V.; Nguyen, M.D.V.; Vu, T.A.; Meixner, G. The Action Point Angle of Sight: A Traffic Generation Method for Driving Simulation, as a Small Step to Safe, Sustainable and Smart Cities. *Sustainability* **2023**, *15*, 9642. https://doi.org/10.3390/su15129642

Academic Editors: Wael Alhajyaseen and Ariane Cuenen

Received: 17 April 2023
Revised: 20 May 2023
Accepted: 29 May 2023
Published: 15 June 2023

1. Introduction

Studying traffic in real life is dangerous for the participants, takes a lot of time, and typically requires a high budget. Therefore, virtual traffic simulation has been researched as a suitable alternative to overcome some of these challenges. However, visualization in the driving simulation typically conflicts with the driver's perceived motion, causing motion sickness [1]. According to Ihemedu-Steinke et al. [2], one of the solutions to ease such simulator sickness is applying diverse traffic factors, such as non-player character pedestrians (NPCP), self-driving vehicles (SDV), and traffic structures. The first problem raised while we conducted the literature review is that there is no clear definition of the term traffic generator (TG). In this paper, a TG is a system that aims to generate realistic traffic factors for the driving simulator (DS) and multi-driving simulators (MDS), where the

DS is the entire collection of hardware components and systems used for virtual driving simulation and MDS is similar to DS, but it can let many real road users (driver: bus, car; rider: motorbike, bike; pedestrian) interact with each other in the TG. To date, some previous studies have reused a Traffic Simulator (TS) to integrate with the DS. This type of system (i.e., the TS) helps transportation experts to study and manage the traffic flow by applying various traffic theories, such as traffic flow theory [3] and three-phase traffic theory [4].

In order to gain best practices from the TSs to build suitable TG, reviewing old problems from other TSs is necessary. The goal is to learn from them, not compare our TG with them, so the following problems in this study can be fixed or still need to be solved. Traditionally, most TSs have been developed using 2D technologies [5] and researchers have attempted to convert these 2D TSs into 3D [6] when integrating with DS. Consequently, these systems present some deficiencies, such as cars flying/sliding when changing lanes (e.g., the wheels do not steer) [7,8] or crossing through other car bodies [9], suggesting flaws in the physics simulation of the TG. Alternatively, when implementing TGs using modern real-time 3D game engines, such as Unity3D and Unreal Engine, these engines provide libraries for seamless realistic physics simulation, providing virtual objects with characteristics including weight, friction, hardness, velocity, and acceleration, similar to real-world objects. According to Saidallah et al., ARCHISIM, MATSim, MITSIMLab, SUMO, and TRANSIMS still do not apply 3D techniques [5]. Furthermore, Lee et al. have criticize the limits of realistic physics in prior TSs such as MITSIMLab, TransModeler, VISSIM, and CORSIM [10]. SimTraffic, RASY, and Anylogic still do not satisfy realistic physics simulation, as expected [6,9,10]. Nevertheless, SUMO only does not simulate 3D physics until it is integrated with the Unity3D DS to solve the physics problem [11]. Recently, VISSIM, AIMSUN and other systems are now integrated with modern platforms, such as Unity or Unreal Engine. Notwithstanding, their models and theories have still remained when adopting new technologies because the TS itself is the independent software while connected with the 3D DS. For instance, SUMO is capable of setting up vehicles with pre-defined route, node, and edge files before starting the simulation [12,13]. While this approach works, it does present some shortcomings; suppose an accident or a traffic jam occurs at an intersection (possibly caused by a real driver controlling one of the virtual cars). In this case, SUMO requires time and computing resources to re-calculate the whole routing network of all other SDVs, which is an undesired outcome for an interactive real-time driving simulator [14]. Another example of the physics problem when converting the TS to the 3D simulator can be the vehicle rotation around its center instead of steering the wheels, even though the TS applies a modern framework (Unity or Unreal Engine). The reason behind this error is that the model in TS only calculates the steering time regardless of the road friction. If that model still does not change, the DS might be able to visualize the wheel of TS's vehicle steering the wheel faster to be matched with the short time from TS's model. However, the traffic researchers, the system's end users, cannot efficiently study the interaction between SDV and actual drivers in the road environment. To deal with this limitation, they must manually calculate or assume the physics and interaction sequences of vehicles and roads to find the right steering time to input back to the TS. Thus, in order to contribute new value, this study improves not only the technology but also the models for TG, which can utilize the strength of technology libraries.

Second, several diverse technical integration challenges have been reported in the state of the art when integrating TGs with DSs. These challenges were identified in the integration cases between TSs and 3D-physics DS such as S-PARAMICS (by SIAS) and UC-win/Road (by FORUM8) [15], or Aimsun and SCANeR™ [16]. For instance, in the combination of S-PARAMICS and UC-win/Road, the researchers reported several problems such as excessive configuration times, setting the number of vehicles on each road, each lane, and specific manual lane-changing cases [15]. Punzo et al. have also reported challenges related to road network consistency and real-time versus prepared scenarios [17]. Furthermore, different platforms, technologies, and new separate chronological releases have resulted in a stalemate that prevents the systems from communicating unless they are built using specific integrated software packages, as Mansoureh Jeihani et al. have reported [15]. After nearly ten years since these problems were found, the developers could find the solutions (middleware packages and API) to solve the integration challenges. Nevertheless, new technologies will cause new problems unless we can use the modern approaches. Hence, this is one of our study motivations to propose the novel TG philosophy (models, theories) that can interact with the modern platforms of the DS and MDS. Another integration study has been conducted by Liao et al. [18], in order to verify their proposed game theory-based ramp merging strategy; however, they used redundant and long ray-casts for SDVs to detect other SDVs [18,19].

A third—and arguably more critical—issue with TG for DSs is the realistic simulation of traffic factors, including the traffic environments and behaviors of the virtual artificial objects (e.g., NPCPs and SDVs). Unlike the problems of converting the TS from 2D to 3D in the first paragraph, this part mentions the systems which are built for DS. These system's outcomes, similar to our TG definition that leverage the capabilities of real-time game engines such as Unity3D and Unreal Engine have been able to more adequately satisfy the realistic physics for environments such as CARLA [20]. However, CARLA still applies the traditional waypoint [21] method to build pre-defined routes, similar to SUMO. Moreover, some research projects do not include intersections or ramps, in order to improve computational efforts [22,23]. On the other hand, the simulation of artificial vehicle and pedestrian behaviors in some studies is still characterized by limitations, such as performing car rotations around its center instead of steering the wheels when turning [24], while other solutions consider only single-lane traffic situations [10]. Furthermore, conflict prediction algorithms, such as that created by Liao et al. [18], are dependent on traffic situations such as changing lanes and ramps. This algorithm is valid only in the single-flow situation, where each lane contains a single vehicle (see lane number two depicted in Figure 1). If a lane is wide enough for more than one vehicle, the situation will be a dynamic flow (see lane number one depicted in Figure 1) or a mixed traffic flow (e.g., both cars and scooters running together in a small street with one lane). This traffic situation usually happens in Asian countries, such as Vietnam and China. Thus, this study contributes a different approach by separating the SDV functions into navigation and reaction modes.

Figure 1. Two lanes (lane 1, scooters; lane 2, cars) in the proposed TG.

Since 1800, the population growth worldwide has been 7.7 fold [25] over only 200 years. This population growth, coupled with the under-developed mobility infrastructure and regulations in developing countries, has created substantial adverse impacts on traffic [26]. For instance, in populous developing countries of the Yuxi circle (4000 km radius), such as China and Vietnam (first figure from [27]), many scooters operate in complicated mixed traffic flows, as illustrated in Figure 1. From this viewpoint, the proposed model in this paper can be used in a TG to simulate the mixed traffic flow, which is not considered in many state-of-the-art systems.

In this research work, we propose a novel TG system and its requirements (see Section 2). The idea is similar to the TS but focuses on generating realistic traffic (SDV and NPCP) around the driver in the DS and later on is in MDS. It is independent software that can be integrated into any DS or MDS, developed in a similar 3D platform. The TG supports a relatively high number of automated vehicles and virtual pedestrians, and there is no requirement for a physical vehicle or person (except the pedestrian simulator, which interacts with the DS and the TG) in real life to interact with our DS. Thus, previous approaches that use cameras to analyze the vicinity of these virtual SDVs and NPCPs would be redundant and even reduce performance in the context of this study; we discuss this further in Section 2.3. The SDV behaviors include observation and reaction. Thus, in Section 3, we propose a novel APAS traffic formula and apply it using the Unity game engine for the SDV observations, in order to allow it to react as naturally as an actual traffic participant. Section 2.1 explains why we chose the Unity framework. Concerning the routing for the SDV, we take another approach, where SDV decides to turn in specific situations, such as at an intersection or T-junction. These decisions are based on its destination, traffic rules, and random choices when it has more than one option. Further, it is based on natural human decision making, helping SDVs to choose a direction at intersections more flexibly and diversely. Furthermore, when steering at the intersection or avoiding the ahead accidents and obstacles, the researchers can also configure many traffic factors (introduced in Section 3.3.2) for the SDV to study the diverse interaction between SDV and real drivers. We examine our models using quantitative evaluation methods in Section 4. Moreover, Section 5 discusses the outcomes and demonstrates that our APAS formula and other proposed methods can contribute to the traffic and driving simulation field. Furthermore, we discuss the outlook on future work and some limitations in Section 6. Finally, Section 7 summarizes the results presented.

2. Requirements

In this section, we will discuss the requirements of the TG. These are presented in three categories, representing the requirements of the TG at three different stages: Integration, Development, and Operational use. The first section describes platform requirements for the TG to integrate with DSs. The second section discusses the realistic traffic simulation requirements for the TG. The last section focuses on the required best practices for improving the performance of the TG.

2.1. Integration: Platform Requirements and Support for Developer Tools and Existing DSs

The TG is intended for use with our previously published DS; therefore, it should be compatible with our DS and other DSs based on similar technologies. As discussed in the introduction regarding state-of-the-art simulators, some mentioned TSs are incompatible with modern 3D platforms, such as Unity or Unreal Engine, used for DS development. Therefore, the TG should operate in real-time and be compatible with existing DS platforms.

The TG should be compatible with state-of-the-art DSs (see, for example, [28–30]), in order to conduct research studies on driver behaviors under various conditions, testing self-driving algorithms, and in-car Human–Machine Interface (HMI) designs.

Additionally, Unreal Engine and Unity3D are two of the most popular and commonly used platforms for 3D game development and real-time simulations. Therefore, many VR/3D DSs have been created with these two platforms. For instance, when searching on Google Scholar, the following keywords yielded many results:

- Unity driving simulator: 54,300 results.
- Unreal engine driving simulator: 16,400 results.

Each platform uses a different programming language: Unity3D uses C#, while Unreal Engine uses C++. At the current state, we decided to test the proposed formula and theories with Unity, as C# and Unity are arguably more commonly used. Therefore, the developer effort to reuse our results was minimized, and the proposed system offers easy integration with other DS systems already implemented with the platform. Nevertheless, the proposed theories in this research work can be implemented and tested with other platforms for integration with any DS.

2.2. Development: System Requirements and Constraints

Secondly, the TG should improve the HCI for the driver in the DS, which involves the interactions between real drivers and SDVs and NPCPs. The system should simulate traffic based on related traffic theories, rules, and Unity physics, instead of pre-defined vehicle (game–object) transform logic. The default transform logic uses the force pushing the car body to move forward or rotate around its center when turning, and the wheels do not interact with the road surface. For these reasons, the two categories of simulation requirements to satisfy realism constraints are physics and behaviors (see Table 1).

Table 1. TG physics and behaviors examples.

Constraint	Examples
Physics and environment	Designs: Vehicle and environment structures (e.g., roads and houses). Sounds: Accident crashes, driving noise, pedestrian voices. Motions: Vehicle responses (e.g., to specific road and weather conditions).
Behaviors	NPCPs and SDVs observe and interact with each other and with the driver, based on: + Traffic and Behavior models + Traffic rules + Human capabilities

Unity includes built-in physics libraries that synchronize appropriately with visualization and interaction. For example, when a self-driving car hits other vehicles, the system must respond to the collision, instead of allowing that car to run through the vehicle. Moreover, the physics of the environment (e.g., road friction and road designs) are essential to help us study road safety, vehicle safety, and the performance of self-driving algorithms. For example, SDVs must run based on physics (e.g., road friction and car structures) rather than transform positions, regardless of the environment (e.g., rain and dry) and context (e.g., different vehicle types and diverse driver behavior). The audio interactive physics are also essential factors. In a traffic simulation, several vehicles (perhaps hundreds) are running simultaneously and in real-time [31,32], while the driver in the DS only hears sounds from a small vicinity, such as wheels on the road and honking. If a car is 5 miles away, the driver cannot hear this vehicle's sounds. For these reasons, in this study, we develop the audio feature only in the DS, instead of for all SDVs in the TG, in order to improve the traffic generator performance.

Previous TSs have a long history of researching and improving vehicle and pedestrian behaviors using diverse traffic models; however, these models are not all appropriate for the TG of the virtual DS or MDS, especially in the mixed traffic flow context mentioned in Section 1. As mentioned above, the virtual SDVs and NPCPs should perceive the vicinity objects in a natural manner. For instance, SDV detection is less effective in heavy rain. This detection should be similar to human capabilities. Importantly, they should respond smartly, as a human would. Otherwise, the traffic environment that the driver in the DS interacts with will not be realistic. This means that the interaction between the actual driver and virtual components in the TG should be realistic in a two-way manner, instead of the user trying to evade the SDV. For example, previous DSs were non-interactive simulations that ran pre-defined/pre-recorded scenarios [14]. Later, virtual DSs aimed to test the HMI concepts for autonomous driving systems [29].

A two-step process for selecting suitable models is presented in Section 3, in order to meet these requirements:

- In Section 3.1, we classify and cluster the models into a classification scheme in the first step. Then, we collate the outcomes to choose an appropriate category.
- In Section 3.2, we indicate the details of the models in that group while testing them in Unity.

With the results from those sections, we propose our theories in Section 3.3.1 and describe their development in Unity in Section 3.3.2.

The table below provides some examples of physics and behaviors that the TG should satisfy.

2.3. Operational Use: Performance

The TG should meet specific performance requirements for real-time operations. As stated in the introduction, some previous studies (Liao et al. [18] and SUMO [12]) have used inefficient methods [19] that negatively affect performance. Therefore, in order to improve the performance of the TG, it should not use:

- Too many/long rays for vehicle detection (Raycast technique) [19];
- Complicated mesh colliders [19];
- Pre-defined routing for SDV (as explained in Section 1);
- Redundant camera and sound data for every SDV.

The camera can detect any objects if they are in the camera's field of view. This method has the advantage of detecting multiple objects. On the other hand, the downside of this solution is demanding more computational efforts and resources to render all of the objects. Another solution is the Raycast technique: using a straight ray (such as a laser) to detect a specific object when the ray hits an obstacle. Hence, in the context of detecting only the preceding vehicle, regardless of other objects (e.g., houses on the side or clouds in the sky), Raycasting will provide better performance than the camera method when used for dozens or hundreds of automated vehicles.

The TG should also meet the following traffic characteristics, which are not satisfied by preceding systems (Liao et al. [18], Lee et al. [10]):

- Stochastic factors (accidents, traffic jams, driver behaviors) require the TG to be an interactive system.
- Mixed and chaotic traffic flows allow the following SDVs to frequently turn and pass the leading SDVs in a lane or switch lanes, mimicking the many small scooters in countries such as Vietnam.
- Diverse traffic conditions (e.g., rules and environments). The traffic condition differs from the condition of a specific vehicle.

3. Methodology and Proposed Solutions

This section discusses our approach to developing the TG in three sub-sections. In the first sub-section, we explain the classification of traffic models and choose the suitable one(s). In the following sub-section, we discuss the models in the chosen category and other needed theories while conducting preliminary tests in Unity, then point out the observed problems. Finally, we propose possible solutions to these challenges in order to develop a system that satisfies the requirements presented in Section 2. Moreover, the models mentioned in this section are only to be chosen and adapted to our study or to introduce models in other branches, as systematized in Figure 1 from [33], p. 447. Due to these reasons, the comparison with other models only occurs in Section 4, not this section.

3.1. Traffic Theory Categories

To ensure that the system is based on grounded TS research, we started with a thorough analysis of traffic theories in former studies, in order to develop unique artificial traffic algorithms that satisfy realistic constraints. First, to classify the traffic theories, we decided to use the classification scheme proposed by Greenshields et al. (1934) [3] in their traffic stream model [3]. This was the first theory to model the linear relationship between velocity and spacing. Hence, this model became the foundation of traffic flow theory, which includes four classifications: the fundamental relation, macroscopic, microscopic, and mesoscopic [33]. These categories have different properties, which can be applied in a specific study context. This research focuses on traffic details, but future projects may explore the relationships between drivers and other aspects—for example, using data mining to study driver behavior from real traffic jam data in large cities. The relationships between the classifications has been systemized as a genealogical tree by van Wageningen-Kessels et al. [33], as depicted in high-resolution Figure 1 from [33], p. 447, which can be zoomed in 500 times.

Most macroscopic, mesoscopic, and microscopic models (underlying assumptions) include the fundamental relation. As per the genealogical tree defined by van Wageningen-Kessels et al. [33], the fundamental relation gathers the different approaches under the model of Greenshields et al. [3], including the Shape model (Drake et al. [34], Smulders [35], Daganzo [36]) and the three-phase model (Kerner [4]). Furthermore, they detail relationships [37], such as the Payne model [38], degenerating from the microscopic to macroscopic flow concept.

A macroscopic approach only models the continuum traffic flow (speed, flow, and density), regardless of the individual vehicle. This flow associates the vehicular flow with the fluid/gas flow, such as Kinematic wave models (Lighthill et al. [39], Daganzo et al. [36], Hoogendoorn et al. [40]). Due to these characteristics, many TSs use this approach to solve problems over a large area, such as traffic congestion, urban planning, and goods logistics.

Unlike the above class, microscopic approaches consider contrasting elements, including individual vehicular factors such as traffic behaviors (acceleration, deceleration, and lane changing), travel/delay time, and long queues, and utilize the flow, speed, and density concepts from the macroflow and fundamental relation. Prominent models in this category include the car following (CF) model [41], lane changing (LC) model [42], and the action point model [43,44].

Mesoscopic approaches comprise the last category, which combines both macroscopic and microscopic aspects. Their main advantage is improving the computational performance, due to requiring less data than microscopic and being more sophisticated than macroscopic approaches. On the other hand, its advantages also create disadvantages, such as a relative lack of fidelity compared to the microscopic category. Additionally, from the macroscopic perspective, it demands more resources and higher technical skills. Models in this category include the gas-kinetic model (Prigogine and Andrews [45], Prigogine [46]), the cluster model (Botma [47]), and the headway distribution model (Hoogendoorn and Bovy [48]).

Based on these arguments and the recommendation of the US Department of Transportation [49], microscopic models were found to be the most suitable, as their philosophy matches this study's contexts; that is, the driver and road user behaviors, as well as the interactions between the driver and other SDVs on the 3D virtual roads.

3.2. Analysis of Traffic Models in Unity

Many techniques in Unity can help the generated SDVs and NPCPs recognize each other and the driver; however, in this research, we aimed to simulate actual traffic, not gaming traffic. Thus, we adopted traffic models and their results retrieved from traffic experiments run by humans, in order to build the SDV and NPCP structures and develop their human-like behaviors. Microscopic models include two sub-categories:

- CF (15 sub-models) has three sub-classes:
  - Safe distance [50] (4 sub-models);
  - Stimulus-response [51] (8 sub-models);
  - Action point model variants [43,44].
- Cellular automata (two sub-models) [52].

Thus, we began with some models in the CF category and the LC model, the Oversaturated Freeway Flow (OFF) model [53]. Moreover, we evaluated not only the Wageningen-Kessels report [33], but also other sources such as the book by Barceló [54], Pariota [55], Hoffmann experiments [56], and Mahut [57]. In some contexts, such as pedestrian and motorcycle behaviors, we indirectly adapted ideas from external models out of the CF category and out of the microscopic category; namely, cell transmission (CT) [36] and the social force (SF) model [58].

In the safe distance model, we implemented the acceleration and safety constraints [54,57] of Mahut using the following formula:

$$x_f(\mathrm{t}+\tau)+\delta\Big(v_f(\mathrm{t}+\tau),\ \beta_f\Big)+\lambda\ \leq\ x_l(\mathrm{t})+\delta(v_l(\mathrm{t}),\ \beta_l). \tag{1}$$

The main concept of Equation (1) is illustrated in Figure 1 from [57], p. 2, where the result for group function A (the left group of the following vehicle, F) is smaller than that of group B (the right group of the leading vehicle, L).

Group A is a function of the following car, starting to stop from time t + τ to time T. It includes A1 + A2 + A3, where

- A1 is the position where the following vehicle (x_f) starts to brake (at the current time t and response time τ).
- A2 is the gap function δ(v, β), in which the follower must decelerate from speed v_f to the desired deceleration speed β_f.
- A3 is the minimum spacing (λ) between the vehicle's front bumpers.

Group B is a function of the following car starting to stop from time (t) to time T, including B1 + B2, where

- B1 is the position of the follower (x_l) at the current time t.
- B2 is the gap function δ(v, β), in which the preceding vehicle needs to decelerate from speed v_l to the anticipated deceleration speed β_l.

When the leading car (L) decelerates at time t, the following car (F) takes a lag time τ (the follower's response time) to observe, then brakes from time t + τ to time T (the stop moment). Car F needs a minimum spacing of λ to avoid a collision with car L.

Previous studies [59,60] identify that perception response time (PRT) starts from the object detection moment until brake pressing, which is similar to the lag time τ (from t to τ). They indicated that the PRT of motorcyclists when they respond to a stopping sign distance situation ranges from 2.12 s (normal perception) to 2.5 s (under fatigue and intoxication states) [60]. This PRT τ is a meaningful metric applied to traffic rules in many countries under many forms. For instance, New South Wales [61] applies a three-second rule and divides the three-second duration from t to τ into two same parts reaction time (time to see and decide) and response time (time to take actions such as steering or braking). Other places such as Canada and the USA [62] apply a two-second rule; meanwhile, Vietnam [63] regulates the minimum safe distance for four-speed cases (Article 12, Chapter III, Circular No. 31/2019/TT-BGTVT of 29 August 2019), as per the following Table 2:

Table 2. Safe driving distances at different speeds.

Speed (km/h)	Minimum Safe Distance (m)	PRT (τ)
V = 60	35	2.1
60 < V ≤ 80	55	2.475
80 < V ≤ 100	70	2.52
100 < V ≤ 120	100	3

Nevertheless, the leading drivers' behaviors are not only caused by themselves but also by other surrounding traffic factors such as traffic objects (traffic signals and signs) and road users (pedestrians, cyclists, etc.). Therefore, in reality, the aforementioned reaction time tends to reduce because the followers have a chance to react much sooner and with smaller decelerations when observing many initial root factors rather than a single leading car, as stated in many models. On the other hand, in hazardous situations, the driver's biological characteristics could increase the heartbeat and response duration and even make them freeze because some motorists produce more stress hormones (cortisol and adrenaline) than others [64]. Moreover, cortisol and adrenaline will narrow road users' peripheral vision [65]. Since these two durations are inversely proportional when compared to the standard traffic rules or models, this paper decides the PRT, the sum of these periods, based on the traffic rules and other experimental studies above. Therefore, in our study, we took the standard PRT (τ_0) as 2.1 s when the driving speed (v_0^{sd}) is 60 km/h. In the future, we will study more about these durations (reaction and response time) in the PRT.

From this traffic regulation, we calculate the PRT using the formula:

$$\tau_0 = x_0^{sd} * 3600/v_0^{sd}. \tag{2}$$

where

τ_0 denotes the standard PRT,
x_0^{sd} is 35 m, and
v_0^{sd} is 60 km/h.

In the case that the driver accelerates or decreases the speed under v_0^{sd}, we calculate the relative PRT ($\Delta\tau$) between the new PRT τ_1 and the standard PRT τ_0 using

$$\Delta\tau = \tau_1 - \tau_0 = x_1^{sd} * 3600/v_1^{sd} - \tau_0. \tag{3}$$

The traffic rule only gives four specific cases, as shown in Table 2, while we want to find a standard relative PRT ($\Delta\tau_0$) to calculate any new safe distance at any given new speed. As can be seen from Table 2, the relative PRT between each case is not the same; for example, compare 2.475–2.1, 2.52–2.475, and 3–2.52. If we take the rate equal to 0.375 (2.475–2.1), then the next PRTs would be 2.85 and 3.225. When the time increases, driving will be safer with a larger distance. Thus, $\Delta\tau_0$ for a 20 km difference in speed was considered suitable. Furthermore, $\Delta\tau_0$ for each 1 km difference in speed was set as 0.01875.

In order to calculate a new safe distance when the speed changes, the new formula derived from Formula (3) is:

$$x^{sd} = \left(v_1^{sd}/3600\right) * (\Delta\tau + \tau_0) = \left(v_1^{sd}/3600\right) * \left\{\Delta\tau_0 * \left(v^{sd} - v_0^{sd}\right) + \tau_0\right\}. \tag{4}$$

Next, we analyzed the action point model of the driver's reaction when they perceive the size of the preceding vehicle change. This physical event occurs when the visual angle from the follower to the leading car size changes, calculated using Equation (5) and as shown in the first figure from [55], p. 3 (it is not Figure 1 in the article) [43,55]:

$$d\theta/dt = -\mathrm{w} \times \Delta v^t/\left(\Delta x^t\right)^2. \tag{5}$$

where

$d\theta/dt$ denotes the perception threshold,
w is the vehicle width,
Δv^t is the relative speed at time t, and
Δx^t is the current spacing at time t.

In this theory, the visual angle starts from the front bumper of the follower to the rear bumper of the leader. Besides, its visual range is the leading vehicle width (from the left to the right side of that preceding automobile). Consequently, the isosceles triangle is created with the top at the middle of the following vehicle's front bumper, the bottom is the leader's width, and the altitude of the triangle is the distance between them (Δx). Michaels [43] assume that when the leading speed (v_L^t) reduces less than the following speed (v_F^t), the follower comes near ((Δx) is decreased), and the visual angle and the width (the bottom of the triangle (w)) increase proportionally. The relative speed is calculated by $v_L^t - v_F^t$. As soon as the result of these relative changes in Formula (5) approach the perception threshold, the following driver will acknowledge that the leader is slowing down.

According to Michaels [43], $d\theta/dt$ is the perception threshold, which is only non-negligible when the follower comes near, and its value is 6×10^{-4} rad/s. Nevertheless, this model is not appropriate for SDVs, as its threshold is too small and the vehicles will not perceive the leading vehicle's width easily without the help of a camera, which reduces performance (as explained in Section 2). Furthermore, it cannot understand the situation when the car crosses or changes lanes. Moreover, vehicle sizes vary (consider the sizes of trucks, buses, cars, motorbikes, and bikes). Hence, using this width detection method requires more effort to analyze all these contexts (Problem A).

Another study similar to Michaels's works is Wiedemann's model, which mentioned the action point based on the perception threshold. Nevertheless, Michael's threshold is calculated by the looking angle due to the width changes, while Wiedemann's perception threshold is based on a car-following straight line [66,67]. Further, in the context of mixed traffic flow, the vehicle's trajectories change rapidly and frequently, which means the vehicles are not on the single line, and their distance is the diagonal line (hypotenuse). Moreover, in the hazardous case caused by the abrupt change back to lane 1 of the amateur driver, after he or she intends to leave lane 1 but an aggressive follower in lane 2 prevents him/ her from changing lane, the follower in lane 1 might crash him/ her. Although the width metric in the Michael model is not suitable, the angle to calculate that width gives us high possibilities to explore and adapt in mixed traffic flow. Hence, this study considers developing the perception threshold concept upon the initial Michaels theory (1965) instead of the later Wiedemann model (1975).

Regarding the lane changing model, we analyzed the situations and locations to change lanes. According to the PG Gipps model [42] and the workflow of the SITRAS system developed by the University of New South Wales in 1995 [68], there are three types of lane changing: Essential, Desirable, and Unnecessary. Each type has various reasons that cause the driver change lanes, such as turning to another road at the intersection or meeting an accident/ obstacle. Moreover, diverse driver characteristics (e.g., demographic, mental, and physics) increase the relevant complexities. In a different approach, Yeo and Skabardonis have combined CF and LC to create the Oversaturated Freeway Flow (OFF) algorithm [53]. Regarding the turning point for lane changing, Nurul Nasuha Nor Azlan et al. [69] have divided the road into three zones.

As mentioned above, previous studies have attempted to manipulate the behaviors of SDVs depending on the traffic situation, such as changing lanes and merging ramps. This approach limits the capability of the SDV when it faces dynamic traffic flows. In other words, the traffic rules and environments will not be the same worldwide, such that the SDVs cannot react adequately, such as a Tesla self-driving car coming to a deadlock in Vietnamese traffic [70]. While developing and testing our algorithms, we figured out that the vehicle's navigation should be independent of the vehicle's reaction, using so-called navigation and reaction modes. It is essential to make the SDV algorithm more adaptable to any traffic situation. The navigation mode of the SDV is in charge of guiding the vehicle to follow traffic rules such as changing lanes and turning at an intersection under the traffic light and direction. On the other hand, the reaction mode ensures that the SDV can react with other SDVs, drivers, and NPCPs in its vicinity. For example, as observed in the Tesla SDV case [70], we assume that, if the reaction mode is unified with the navigation mode, the SDV will come to an impasse under the following stochastic and diverse factors: SDV structures, random speed, NPCPs, and actual driver behaviors based on different demographics and cultural mindset. For instance, if the SDV is running while the driver in the DS abruptly changes lanes, the SDV will collide with the driver's vehicle. This error is due to the neglect of the SDV regarding the driver's previous states before changing lanes.

This state still needs to be observed in natural circumstances, as the driver usually does in real life (Problem B).

3.3. Proposed Solutions

3.3.1. Proposed Formula and Adaptation

To solve the crucial problems (A and B) described above, we constructed a new formula for the reaction mode. We borrow and add part of the action point formula into ours, as shown in Formula (6). Furthermore, we incorporate the perception threshold (3×10^{-3} rad/s) of Hoffmann et al. [56] and the angle of sight that Huynh et al. [71] contributed when they studied the SF model, as well as the safe distance (x_{sd}) concept mentioned above, into the APAS formula:

$$[(\Delta x_m < 1 \cup \Delta x_c < 2) \cap (\left|\gamma_l - \gamma_f\right| > \Delta\gamma \rightarrow \ v_f = 0)] \cup \ \{\left(\left|\gamma_l - \gamma_f\right| < \Delta\gamma\right) \cap \ \Delta x < x_{sd} \rightarrow d\theta/dt = \left[\ \Delta v^t/(\Delta x^t)^2\right]\}. \quad (6)$$

where

Δx_m is the side-spacing between a two-wheel vehicle and another vehicle,
Δx_c is the side-spacing between a four-wheel vehicle and another vehicle,
γ_f is the follower's angle of sight [71],
γ_l is the leading vehicle's view angle,
$\Delta\gamma$ is the APAS,
v_f is the speed of the following vehicle (which is the vehicle used in this formula to interact with the leading vehicle),
Δx is the current spacing between the following vehicle and the leading vehicle,
$d\theta/dt$ is the perception threshold,
Δv^t is the relative speed at time t, and
Δx^t is the current spacing at time t.

There are two angle types: the Euler angle (Eu) and the local Euler angle (local Eu); these angle types will be explained more in Figure 2 and the paragraphs in Section 3.3.2. The sight angle γ is the local Eu in the Unity game engine. This angle type is relative to the parent vehicle transform's rotation. It will be compared to the world rotation if it does not belong to another object. We name this formula the action point angle of sight (APAS). The angle X is the difference between the sight angles of the leading and following SDVs. The threshold is the angle $\Delta\gamma$ (APAS). We discuss how we apply these in Unity in Section 3.3.2, then run experiments to determine the right-angle degrees for the APAS threshold angle in Section 4. The APAS formula aims to help the SDV recognize other rear and preceding cars and motorcycles automatically and continuously. Indeed, a mixed traffic flow of motorbikes is composed of many vehicles frequently steering while driving. For this reason, APAS can help the SDVs to observe the complicated mixed traffic flow continuously, which other models such as CF have not considered, as CF was developed to simulate the single flow of the cars typical on Western roads (not the mixed flow typical in Eastern countries such as China and Vietnam).

The succeeding SDV will control its speed (v_f) depending on the APAS degree. There are two contexts in which the following vehicle reduces its speed:

- If angle X is larger than $\Delta\gamma$, the following vehicle will stop. The side-spacing is the length of the ray from the left or right side of the vehicle to other vehicles. The side-spacing in South Australia is 1 m [72], and that in the U.K. is 1.5 m [73]. Thus, we propose this distance type for a two-wheel vehicle (e.g., motorbike, scooter) to other vehicles, Δx_m, to be 1 m, while that from a four-wheel vehicle to other vehicles, Δx_c is 2 m. In the traffic calming measure, there is a technique called lane narrowing that helps drivers increase alertness and reduce speed to drive comfortably. As a result, the side distance between two cars of more than 2 m cannot be obtained. Part ($\Delta x_m < 1 \cup \Delta x_c < 2$) helps the vehicle recognize the presence of other vehicles when angle X is larger than $\Delta\gamma$.

- If angle X is smaller than $\Delta\gamma$, then if the distance between them reduces to under the safe distance (x_{sd}), the system will calculate the action point model's perception threshold value ($d\theta/dt$). The following vehicle will stop or steer when this threshold is larger than 3×10^{-3} rad/s. Part $\Delta x < x_{sd}$ helps the vehicle recognize the presence of other vehicles when angle X is smaller than $\Delta\gamma$.

This perception threshold is adapted from the action point model. It divides the relative speed by the current spacing. The relative speed is the following speed minus the leading speed (Δv^t). The current spacing is from the position of the following vehicle to that of the preceding vehicle (Δx^t). In Unity, a vehicle object can detect other's settings (e.g., speed and angle) using the Raycast or Boxcast techniques (explained further in the following section). The width parameter, w, in Formula (5) describes how the driver recognizes the distance between two vehicles. This distance is inversely proportional to the width of the leading vehicle. Thus, our concept does not need to use the width. Instead, the APAS threshold angle and safe distance (x_{sd}) are the conditions utilized to calculate the perception threshold $d\theta/dt$.

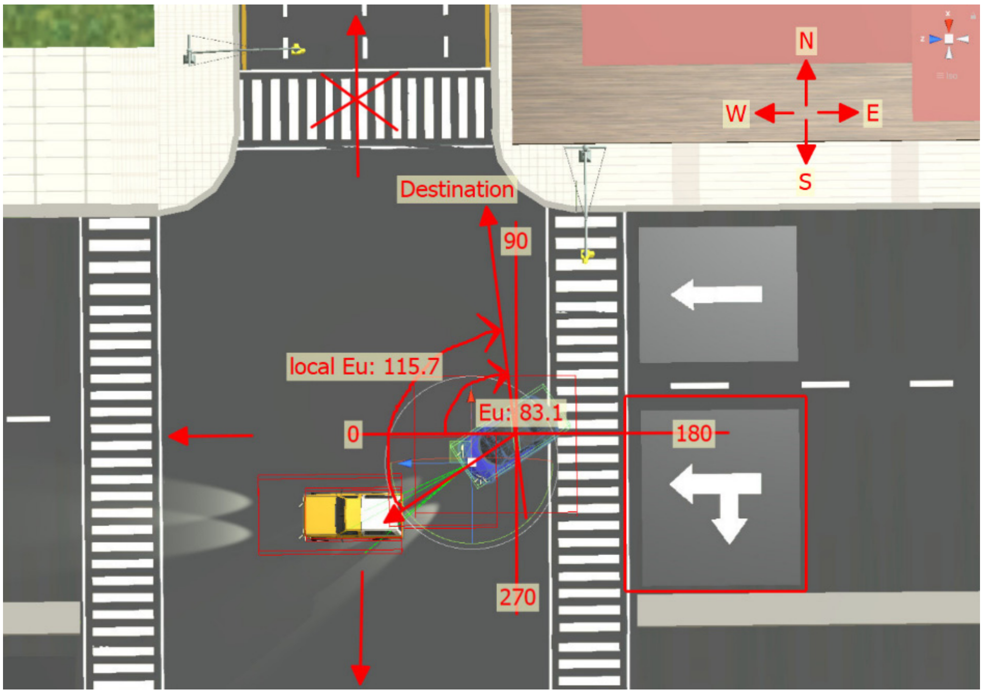

Figure 2. The Euler angle (Eu) and the local Eu in the RNM.

The cell transmission model is used for the speed reset function. With this function, a vehicle can run at a random speed when entering a new road or area [36], in order to avoid all cars running unnaturally at the same velocity.

Next, we discuss the features of the navigation modes: the lane changing mode (LCM) and the route navigation mode (RNM). The LCM adapts the turning point concept [69] while designing the lane structures in Unity. As the explanation above regarding the reaction time and response time, the driver may decide to brake or steer to other lanes depending on the situations of angle X that the APAS formula states. Hence, when connecting the LCM (navigation mode) with the APAS formula (reaction mode), if the preceding/aside vehicles are driving dynamically with a large angle X, the followers should reduce their speed to avoid accidents caused by conflicting with other road users' moving vectors. Then, in the context that other motorists or SDVs (in front of or aside) run stable flow with an angle X that is smaller than $\Delta\gamma$, the subject can decide to slow down or turn to another direction, either on the same lane or other lanes. For more details on how this study connects the LCM with the APAS model while programming in Unity3D, this paper describes these methods in Section 3.3.2. End-users can use this LCM thanks to the APAS feature of the reaction mode, which runs in the background; for example, experts can test many lane-changing scenarios with different times, road widths, and friction coefficients, as this TG builds the AI for SDVs, which can act (i.e., observe and respond) similar to ordinary drivers. Thus, road safety engineers can use our LCM feature to study lane-changing methods. Finally, the RNM makes this study different from others, by building the navigation at the intersection rather than using a manual waypoint network. For example, waypoints in Unity require the route network from SUMO [11,13,74]. This network builds whole routes at once, which are then rebuilt when the TS meets stochastic factors (accidents, traffic jams) [12] caused by the actual driver in the DS, leading to a large waste of time. In contrast, our RNM feature can help the SDV to recognize the direction from its current position to the destination through Unity coordinators. When it comes to an intersection, it will choose to steer or run straight, depending on the traffic rules and randomization. This approach can help the SDV to run on any road regardless of the prepared map, as in natural driving behavior. Moreover, if the driver hits the SDV, causing the SDV to rotate differently from the road direction, the RNM can help it to turn back smartly.

Regarding pedestrian distance, our system still applies the same spacing between NPCPs. In the future, we can adapt theories regarding spacing, such as in the social force theory [58], because we prepared the structures for our TG with modularity in mind. Additionally, the pedestrian spacing difference should be diverse, depending on the tribe, culture, area, country, and pedestrian relationships (e.g., friends and family or strangers) [75]. There are various types of distance, such as intimate space, social space, and public space [76].

3.3.2. Implementing the Novel Theories in Unity

As portrayed in Figure 3, in Unity3D, each SDV uses only two Boxcast (BC) rays and four Raycast (RC) rays (less than Liao et al., who used 40 RC rays [18]) to detect and calculate the APAS of the leading SDV. The RC uses a single laser ray, while the BC attaches a box at the head of a ray to detect a larger area.

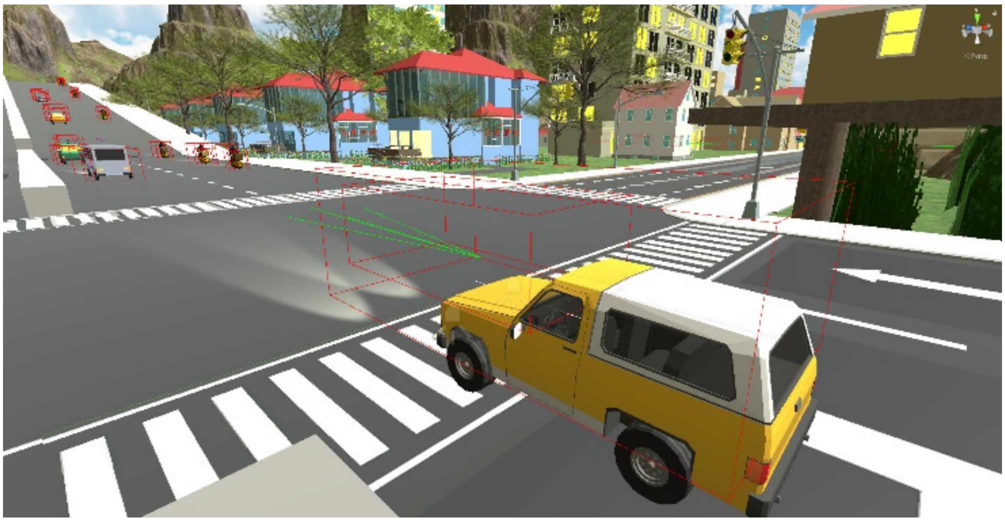

Figure 3. Efficient RC and BC rays for each SDV.

Based on recommended industry best practices [19], instead of using the complex mesh collider, we use a simple box collider around the SDV. Thus, when the SDV using the RC and BC rays detects another SDV's collision box (the invisible boundary of that vehicle), the memory required for this process is much lower. The APAS formula is implemented using two functions: the first solves for the typical car-following situation and includes two BC rays, action point logic, and the perception threshold; the second contains four RC rays (left, right) and an angle algorithm. To adopt our proposed theory ideas, in Unity3D, we program the LCM function (one of the RNM functions) by receiving the status parameters from the APAS function as the conditions to decide how the subject SDV moves such as the running (motor/brake torque), steering (speed and angle). The APAS function aims to collect data from other vehicles' characteristics (positions, vehicle size, speed, etc.) by using sensor techniques such as RC and BC. Then, the APAS will calculate the observed data and return the status results to put into the LCM function, such as a safe distance and APAS.

To calculate the angle that the SDV needs to turn at the intersection in our RNM functions, we use the local EU in the Unity game engine, as it provides the rotation of the compass system inside a vehicle, as depicted in Figure 2; that is, its angle relative to the parent vehicle transform's rotation. The blue sports-car's destination is north, but the traffic rule does not allow it, so it chooses a random option (go straight or turn left). The scenario also shows how the blue car reacts when it runs into the wrong lane, then turns to the left and meets the yellow car that runs quickly from behind to ahead in a parallel manner. With the help of the reaction mode, the blue automobile automatically recognizes the jeep and stops until the jeep passes, allowing enough space (2 m) for it to complete the turn. This test turns off the traffic rule of lane changing, in order to prove that the blue car has an effective reaction mode to deal with navigation mode defects, instead of causing unnecessary accidents. To satisfy the realistic constraints, our system allows the researcher to configure the following functions:

- Infrastructure (similar to Table 3): friction, size, slope, curve, rotation, network.
- SDV:
  - Structure: vehicle weight, size, type, wheel radius, wheel physics.
  - Speed: max/min/acceleration speed.
  - Safety factors: ray detection length and angle, safe distance, APAS.
  - Steering: change lane speed, turning speed at the intersection, turning angle.
  - Traffic rule: some simple rules with traffic signs.
- Light:
  - SDV: user can control front light angle (up/down), intensity and range, no back signal light.
  - Streetlight: automatically follows the time.
  - Sun: synchronize with time.
  - Traffic light: setup manually.
- Weather: only visualize rain and fog with sounds, but the physics for road and vehicle have to be input by the researcher.
- NPCP: pedestrians can jump, walk, run, and follow traffic light rule.

Table 3. Test variables and parameters for Experiment 1.

Name	Type	Value
Accident-rate	metric	The total number SDVs in each trial/total SDV collisions in each trial
Speed	parameter	Random (km/h)
Safe distance	parameter	According to speed
SDV sizes	parameter	Random: car, jeep, bus, scooter, etc.
SDV types	parameter	Random when spawning
Number of SDVs	parameter	Random: 90–150 SDVs
Each trial's duration	parameter	3 min
Angle	parameter	0–90

Table 3. *Cont.*

Name	Type	Value
Traffic infrastructure	parameter	Roads, intersections, T-junctions, two bridges, one mountain road, one roundabout, traffic lights and some simple traffic signs (allow to turn right/left when red, reduce speed, speed limits, dangerous turning situations)
SDV collisions in each trial	variable	Discrete, not accumulated
Environment	parameter	The weather is normal dry, clear visibility

4. Evaluation and Results

The motivation for these experiments is achieved by the physics experiment through testing how a vehicle loses control when brake or steer is applied abruptly after running too fast. The APAS theory is a novel study angle different from other prior concepts such as car following, action point and safe distance models. Further, the APAS itself is the math formula. Thus, this section aims to calibrate our perspective by conducting only the appropriated quantitative methods with parameters and variables from the APAS formula.

In this section, we present two experiments:

Experiment 1 was conducted to test and find the appropriate angle for the APAS formula. Thus, we compare the accident rates in trials using different angles.

Experiment 2 was conducted to prove the validity of the dynamic traffic flow arguments of the TG (presented in Section 3) by looking at the accident rate. In this test, we compared the accident rate between APAS and the previous traffic theories: the action point and car following models. For this experiment, we used the same environmental conditions as in Experiment 1 to conduct new trials for previous models and reused 50 APAS trials from Experiment 1.

These experiments are operated in our system on the Unity3D platform. Therefore, the TG has not only the SDVs to interact bidirectionally with the DS but also the interfaces for researchers to input the parameters in Table 3 above. Due to the object-oriented interactions, end-users will choose the object and then setup the configuration data of it. For example, if users study how a car brakes or slides on the raining streets, they can change the friction setting from zero to other numbers, as shown in Figure 4. Researchers can change other settings such as vehicle speed and steering angles when clicking the SDV or change the trial duration when interacting with the Management object.

4.1. Experiment 1: Testing APAS

This section describes our evaluation of the APAS formula discussed in the previous sections and presents the evaluation results. In particular, we measured the accident-rate metric while setting different angle parameters in each trial. Other random parameters are presented in Table 3. Navigation at intersections was based on traffic rules. Considering the stochastic number of SDVs, the total SDV collisions were divided by the total number of SDVs spawned in each trial to calculate the accident rate. The experimental results were not affected by differences in the total number of cars, because the total number of cars is directly proportional to the total vehicle collisions. Thus, only different angles were used to decide the output of these trials.

Figure 4. Road friction in the proposed TG.

In each test case, we included appropriate traffic rules and some of the above traffic infrastructures to test the APAS concept. The streets allowed the SDV to switch the flow dynamically to overtake the preceding vehicles, similar to lane number one of the scooters, as depicted in Figure 1. For example, when a motorcyclist is driving while looking for a house address on the other side of the street (reverse direction traffic flow), their behavior may be steering the wheels abruptly right after either turning on the signal light or not [77]. The reasons behind this mistake could be the following:

- The stress of missing the chance to turn near the destination could make the driver ignore traffic safety.
- Psychological inertia [78]: in the driver's mindset, the priority of turning when seeing the address is higher than traffic safety (observing the road and turning on the signal a few seconds before turning). Thus, due to inertia, they choose to turn automatically.

Other reasons for turn signal neglect in mixed traffic (e.g., in countries such Vietnam) are talking with passengers, sleepiness, and advertisement distraction [79]. Thus, we also performed tests in scenarios without signal lights, in order to investigate how the SDVs use the APAS formula to react. In the test case, the automated vehicles could detect the NPCPs when they ran, walked, and jumped over obstacles. The test was conducted for angles from 0° to 90°, based on the car following and action point concepts, which we adapted together using the APAS function in Unity. After this 90-degree angle, the SDVs will engage in the opposite flow instead.

As explained above, physics and behaviors are two main factors that affect traffic and also the reasons to cause the accidents. Inside the behavior scope, traffic psychology and demographic are factors that can make accident causes to become complicated. For instance, in addition to the formal traffic regulations, informal rules can make road users misunderstand and cause accidents [80]. From this fact, the trials will exclude the actual driver and set the artificial reaction and perception of the SDV based on the theories and data from previous studies and traffic rules such as PRT, safe distance, perception threshold and so forth. Moreover, the traffic environment will be based on the standard setting of the Unity3D framework's physics libraries to prevent real-time weather-road physics changes. Therefore, the experiment can avoid the domino effects, which can jam the accident rate metric. The reasons for choosing the accident-rate metric were:

- The experiment scenario has diverse and random traffic factors, as mentioned above. Nevertheless, we supervise the test cases by developing the automatic safety mechanisms (except the APAS we want to test). Thus, whatever the random speed or vehicle size, the safe distance and PRT will change, respectively, similar to the result of 2/4 or 4/8 that are still 0.5.
- The goal of this formula is traffic safety for an autonomous vehicle in mixed and complicated traffic flows.

After running 20 trials at each angle, cleaning the data before analyzing was necessary because some trials accumulated accidents instead of separate accidents. This means that many accidents caused by only one SDV led to the accident rate of different SDVs becoming wrong.

Figure 5 shows the pattern of accidents for each angle from the cleaned data. The accident rate at an angle of 0 degrees began at 3%, before it plummeting to 1.95% when the angle was 10 degrees. Then, the rate fluctuated within 0.03% with angles between 10° and 30°. From an angle of 40° to 70°, the accident percentage presented a continuously slight increase. Finally, the rate from angles of 70° to 90° showed an upward trend.

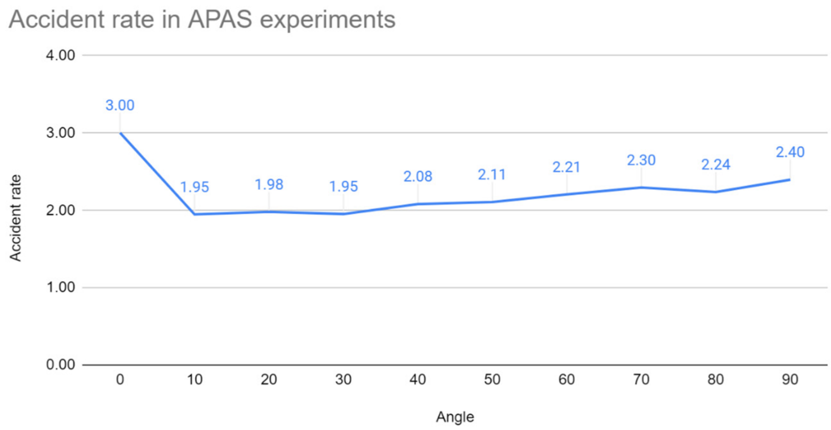

Figure 5. Experiment 1: Testing APASs.

The reason for this high-rate kickoff was that many SDVs reduce their speed for every slight angle difference between them. As shown in Figure 6, under the 10-degree angle, the SDV still drives in its current lane, instead of changing to the next lane. Hence, the traffic flow is slow, leading to many traffic jams exceeding the road capacity. The numerous trials showed that an angle from 10° to 30° was the best range for the APAS. The larger angles resulted in higher accident rates, due to the late detection of SDVs by other nearby vehicles.

After a vehicle changes lane or switches flow in one lane, the states are important. After an angle of 40°, the vehicle tended to climb to the sidewalk before returning to the road up to an angle of 60°, when the SDV finds it hard to turn back when it turns at the intersection. Furthermore, the trials showed us four types that caused separate accidents (Figure 7) and a root cause of the accumulated accidents (Figure 8). After one of the four accident types, the following SDVs may hit the leading SDV until it rotates angle A to the satisfied angle B, as illustrated in Figure 8. The thick white linear arrows indicate the possible driving directions, while the red dashed arrow shows how the vehicle rotates from the current to other directions. For example, the APAS formula-set angle condition is 70; when follower 1 collides with leader 1 at an angle of 40°, leader 1 will move to another lane and continue to be hit by other followers (2, 3, etc.), until its angle is 70°.

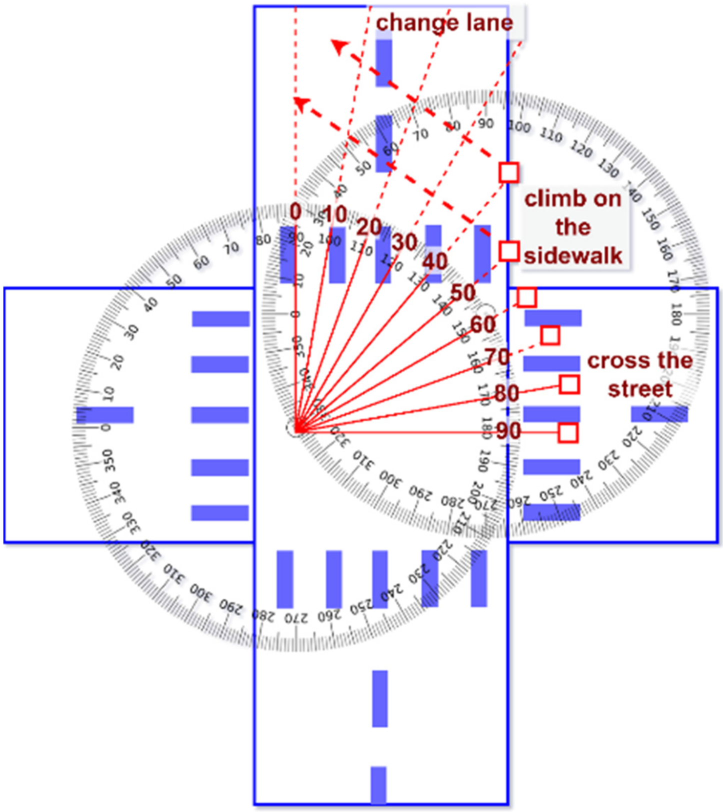

Figure 6. The results for each steering angle group.

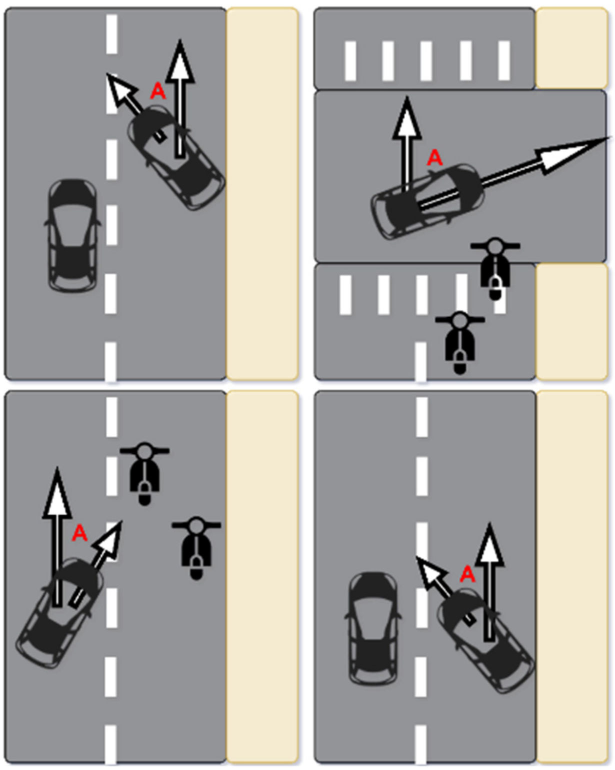

Figure 7. The APAS A in the accident cases.

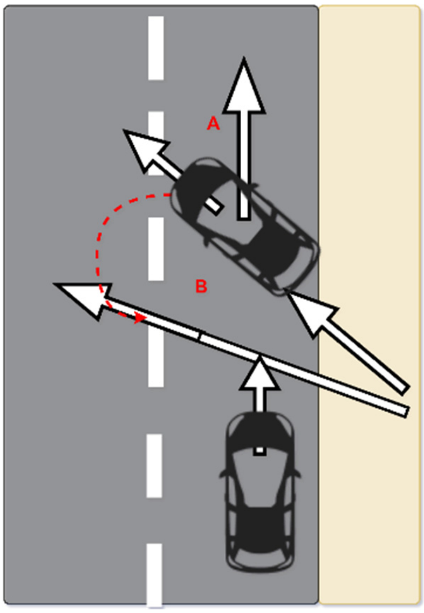

Figure 8. The APAS A in the accident cases and angle B after colliding.

While observing cases with an angle range of approximately 10–30 degrees, the APAS formula cannot solve 100% of accidents, as the SDV will go back to the street from the sidewalk at the intersection when the traffic light is red, while other vehicles are already behind. We describe this situation as "Vehicle-Pavement near the red traffic light." The reason behind this is blind-spot detection. This can be resolved, but will require more than four RC rays, leading to reduced performance of the TG. Thus, this solution comprises a trade-off of an approximate 2% accident rate against performance. This 2% accident rate is considered acceptable in the current state, as the TG applies many stochastic factors and physics simulations which are not considered in other systems.

4.2. Experiment 2: Comparing the APAS Algorithm with the Action Point and Car following Methods in Unity

To measure the accident rate of the action point model and the car following model (AP-CF), we simply deactivated the APAS function. This experiment aimed to show that, without the APAS model, the accident rates will be higher as the vehicle direction is only forward in the CF model, as portrayed in Figures 6–8. When the SDVs run in a complicated mixed traffic flow, including cars, motorbikes, and buses, they will inevitably hit each other. As mentioned above, other studies have addressed this problem by applying techniques including game theory models and lane changing; however, these approaches only solve part of the problem: each lane has a single flow (i.e., the lane size is only enough for one vehicle). The other scenario, where there is a lane with many small motorbikes or mixed-size vehicles (two and four wheels) running dynamically, as in some Asian countries, has not been considered in previous studies and still remains largely unaddressed.

The test case for this evaluation included 50 trials from the APAS concept and 50 from the AP-CF concepts. These trials contained separate and accumulated accidents, as we aimed to test the worst cases from APAS and AP-CF. However, we reused the first five trials of each APAS from the Experiment 1, in order to save time.

As shown in Figure 9, the CF accident rate was significantly higher than that of the APAS model, by over 2.4 fold.

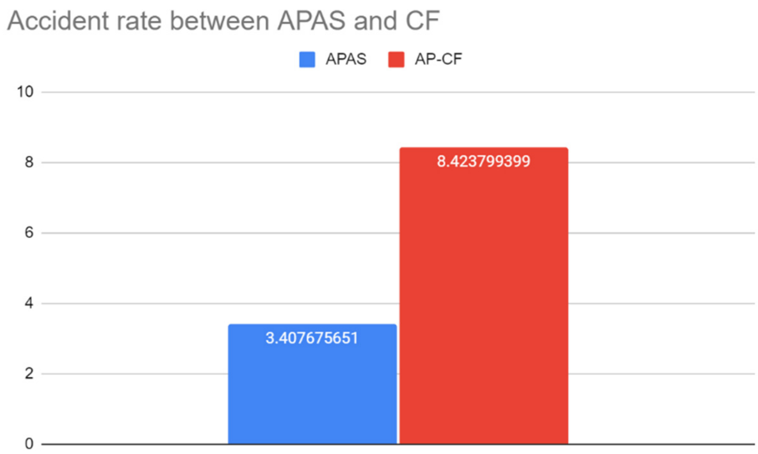

Figure 9. Experiment 2: Comparing the APAS algorithm with the action point-car following methods in Unity.

Among previous studies concerning the safety of mixed traffic flow, Chenhao Dong et al. [81] have compared the conflict rate of mixed traffic flow before and after applying a physical isolation divider. Their results indicated that a mixed traffic flow is more dangerous than a single flow, as depicted in Figure 5a from [81]. The largest difference between the two lines in this chart is 2.159 fold (0.95/0.44), when the road occupancy is 0.7.

Therefore, this result is satisfactory, when comparing the accident rate (2.4) with that of Chenhao Dong et al. [81] (2.159) as, without an APAS solution (which is similar to a physical isolation divider), the risk is higher.

5. Discussion

As presented in the introduction, there are four problems that motivated this study: those of TS vs. use of TG for DSs, challenges associated with integrating TGs with DSs, the need for the realistic simulation (physics and behaviors) of traffic, and the simulation of mixed traffic flow in over-populated/under-developed/under-regulated traffic.

The TG concepts that this paper proposes to focus on for improving the virtual traffic in the DS were explained in Section 2. We proved that this framework is essential by pointing out the following weaknesses of the TS:

- Developing methods: After conducting a literature review, we discussed examples of other TS limitations (see Sections 1 and 2) to design the novel TG framework. We chose the Unity platform, as it is one the more commonly used 3D/VR simulation technologies. Thus, our TG can be integrated with other DS systems. While developing this TG, we acknowledged the problems inherent to preceding solutions, as indicated in Section 2.
- Integration: Due to the homogeneous Unity platform, this TG can be seamlessly integrated with the DS.
- Simulation: The table below compares the differences between the technique presented in this study and those in other studies, based on three criteria: Vehicle routing, reaction, and detection.

Table 4 provides a qualitative comparison, as each system has different characteristics and components. Hence, the quantitative comparisons would not be valid and unnecessary. Another way to conduct quantification is by developing alternative solutions in parallel; however, this is only possible for the child functions, not the core functions. For example, all other sub-components and related functions must follow the core Vehicle Movement function, which is based on either waypoint or route navigation (our solution). Otherwise, time will be wasted in building whole new different systems. Nevertheless, in research, nothing is impossible; we can find a solution if we spend more time, budget, and effort to analyze such as we can conduct the quantitative comparison by comparing our TG and other systems with the same real-life scenarios (data, participants and environments).

Table 4. Qualitative methods and results of other studies and this study.

Criteria	Methods	Results and Remarks
Vehicle routing with stochastic factors (accidents, traffic jams, driver behaviors)	Other studies (CARLA, SUMO): Pre-defined route network waypoint	Results: Requires more computation time. Reasons: Each SDV/all SDVs must recalculate the whole network when the real driver in DS causes accidents
	This study (APAS): Routes navigation	Results: Requires less computation time. Reasons: Each SDV only calculates the direction at the intersection.
Vehicle detection with better technical solutions in Unity	Other studies (Liao et al. [18], Olstam [22]): Use many long sensor rays Detect complicated object colliders	Results: System performance reduced Reasons: These technical methods demand higher system resources.
	This study (APAS): Uses less and shorter sensor rays Detect simple object collider	Results: System simulates well with many SDVs Reasons: The functions demand relatively fewer system resources.
Vehicle reaction with: + Dynamic traffic flow + Diverse traffic situations	Other studies (Liao et al. [18], Olstam [22]): Combine behaviors (observation and reaction) with the traffic conditions	Results: High accident rate or SDV falls into deadlock [60]. Reasons: The solution is not generic and not suitable for mixed and chaotic traffic flows.
	This study (APAS): Splits the algorithms into the traffic rules component and reaction functions (running, turning)	Results: Relatively lower accident rate; indeterministic free movement of vehicles in the mixed traffic flow. Reasons: The SDVs observe and react in real time, regardless of any traffic condition.

Moreover, the most important contribution of this study is our proposed formula (APAS) and its applications to satisfy the realism and other quality attributes of the TG. As stated in Section 2, this project aims to provide satisfactory physics in the simulation. Thus, when implementing the proposed Formula 6 (APAS) in Section 3.3.2, our system allows the user to edit various parameters related to this formula. For instance, the Raycast length (in Unity) relates to the side-spacing (Δx_m, Δx_c) and current spacing (Δx), which can be used to study the vision of SDVs in different weather conditions such as rain, foggy, and clear sky. To deal with road friction, the TG allows researchers to change some physical inputs, as portrayed in Figure 4. The system will update these changes to the vehicle movements, such as steering (γ_f) and running (v_f), at every simulation time frame (0.02–0.333 s). Furthermore, by inheriting the cell transmission concept, the TG splits the road into small parts, which researchers may wish to use to prepare scenarios for a specific road segment, in order to study a hazardous situation or accident investigation scene. Therefore, Formula (6) is practical and can be applied by other researchers in their studies. Regarding the Unity3D framework, it has many advantages to satisfy the requirements in

Section 2—for example, it supports multi-platforms such as web, computer and mobile that provide us with the portability attribute to conduct the studies. Further, it is free of charge when we, researchers, do not intend to make annual revenue cross its policy threshold of over 100 thousand dollars. Moreover, Unity can be considered as the easy-to-use platform that develops solutions with less effort in C# when compared to other technologies, typically Unreal Engine. Thus, Unity3D has a broader user community.

Nevertheless, somebody will argue that other platforms such as Unreal Engine can provide the same benefits, even with better graphic quality. From the perspective of the developer, we totally agree with these arguments. On the other hand, this study's main objectives are to provide the APAS model and other recommendations (in Section 2) to the wide research community. When putting these objectives on the table, we decided to trade off the highest graphic quality with the easy-to-use strength. However, currently, the moderate visual quality in Unity already satisfies our standards which are mentioned in Section 2. Furthermore, when putting more effort into designing and configuring (such as setting complicated light systems, building high-poly objects in external tools (Blender, Revit and so forth), and writing custom shaders), you can achieve better visual quality such as Unreal engine. In summary, while Unreal Engine is targeted to end-users who prefer setting up graphics easily, Unity, our initial target platform, is oriented to anyone who favors the easy implementation of the solutions and can tolerate high effort in graphic preparation.

6. Limitations and Future Works

Nevertheless, there were some limitations to this study. Firstly, the "Vehicle-Pavement near the red traffic light" described in Section 4 remained a persistent problem. We trade-off a small percentage of accident rates against performance, which is an issue that will be addressed in future work.

Further, this TG aims to test the APAS formula in mixed and chaotic traffic flows and, so, few traffic rules were applied. For example, as mentioned above, the SDVs or the driver can steer and change lanes without turning on their signal, in order to test how well the artificial vehicles react. Thus, the current low accident rate is only valid in the context of testing mixed traffic flow with few traffic rules. In the experiment, the weather was normally dry with clear vision, and it is necessary to conduct tests under other environmental conditions, such as rain, snow, or fog. Finally, the test cases only let the SDVs interact with each other; no real driver was involved as a participant controlling one of the virtual vehicles. As real driver behaviors are diverse across demographics and psychologies, it is necessary to involve a real driver in future studies/tests, in order to investigate the performance of the TG across demographics.

Another limitation is applying only a qualitative comparison in Table 4, due to the various challenges mentioned above. This study especially wants to calibrate the APAS formula by conducting the appropriate quantitative approaches (experiments, charts) for a math formula similar to calculating the Pythagorean, Integral formula. In the next studies, we will conduct other quantitative methods to test our methods with real drivers in the DS and MDS [80]. In the future, we intend to conduct an experiment for APAS using real drivers, in order to analyze the interactions between the real driver (behaviors, demographics) and SDVs and NPCPs carefully, also taking into account more traffic rules and weather conditions. This TG is also useful for testing the interactions between different road users when integrated with different simulators (e.g., a car DS, a bicycle simulator, an e-scooter simulator, and a motorcycle simulator).

7. Conclusions

A detailed study and investigation of state-of-the-art and existing solutions revealed that many previous systems have encountered four key problems: the use of TS for DSs, challenges related to integrating TGs with DSs, relatively less realistic simulation (physics and behaviors) of traffic, and lack of mixed traffic flow simulation. Therefore, in this research work, we developed a model for, designed, and implemented a TG system to address these challenges.

There are three main contributions of this research. First, we developed a traffic generator (TG) for our published driving simulator (DS) and other DSs developed on a similar platform. Second, we proposed the TG framework to improve the realistic concepts (physics and behaviors) and efficiency. We also introduced novel approaches to indicate the TG framework and best practices that other researchers can adopt when developing their own TG. For instance, the proposed model structures the behavior of SDVs into two separate modes: A reaction mode (RM) and a navigation mode (NM). We presented a route navigation mode to replace the obsolete waypoint method in the navigation mode, and implemented the system using high-performance techniques (e.g., fewer RC rays, simplifying the collision detection). Finally, we proposed a new traffic formula (APAS) while studying how to actualize previous traffic theories in the development of a 3D TG. As some previous theories are not suitable to adopt to this novel TG framework and some new technologies, while developing the TG, the APAS formula not only inherited aspects from but also improved upon prior traffic studies. Therefore, other researchers can utilize these results for any traffic simulator or related system. The 250 trials with a total time of 750 min proved that the proposed formula is successful in simulating mixed and chaotic traffic flows while satisfying realistic physics and behaviors.

Overall, this study's goal is to introduce the old and new defects in the prior studies and systems as the best practices to propose the novel TG framework that other researchers can use. We want to avoid comparing our system with others because technologies are evolving daily. Thus, many mentioned problems in this paper could have been fixed or will be solved soon, making the comparison invalid.

This study is the first step for us to reach the big vision of building smart cities because after we finish the traffic generator core for the driving simulator, we will move on to connect this TG to multiple driving and riding simulators (car, bus, truck, motorbike, scooter, bike, tram and train), and pedestrian simulators. Moreover, our solutions can help researchers from other fields such as road design and construction, vehicle engineering, insurance companies, lawyers, and government (politicians, police) work in the same systems with transport experts and software engineers such as our current team. Everyone in society from around the world can work together on the enormous system, the so-called traffic generatorverse. There are many applications that this virtual universe can offer, such as developing self-driving vehicles, enhancing driving skills and laws, and predicting imminent accidents and.

Author Contributions: Conceptualization, M.S.P.D.; formal analysis, M.S.P.D.; funding acquisition, G.M.; methodology, M.S.P.D.; software, M.S.P.D.; supervision, T.A.V. and G.M.; writing—original draft, M.S.P.D.; writing—review and editing, K.V.K., M.D.V.N., T.A.V. and G.M. All authors have read and agreed to the published version of the manuscript.

Funding: The APC was financed by Heilbronn University through funds of the project HHN INVEST-PRO^3. The project HHN INVEST-PRO^3 is funded by the DAAD as part of the "UAS.International" program (funding line B) with grants from the Federal Ministry of Education and Research in Germany.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

Abbreviations

TG	Traffic generator
DS	Driving simulator
MDS	Multi-driving simulators
TS	Traffic simulator
SDV	Self-driving vehicle
NPCP	Non-player character pedestrian
RM	Reaction mode
NM	Navigation mode
RNM	Route navigation mode
LCM	Lane changing mode
APAS	Action point angle of sight
RC	Raycast
BC	Boxcast
CF	Car following model
LC	Lane changing model
PRT	Perception response time

References

1. Reason, J.T. *Motion Sickness*; Academic Press: Cambridge, MA, USA, 1975.
2. Ihemedu-Steinke, Q.C.; Rangelova, S.; Weber, M.; Erbach, R.; Meixner, G.; Marsden, N. Simulation sickness related to virtual reality driving simulation. In Proceedings of the 9th International Conference on Virtual, Augmented and Mixed Reality, VAMR 2017, Vancouver, BC, Canada, 9–14 July 2017; pp. 521–532.
3. Greenshields, B.D.; Bibbins, J.R.; Channing, W.S.; Miller, H.H. A study of traffic capacity. *Highw. Res. Board Proc.* **1935**, *14*, 448–477.
4. Kerner, B.S. Three-phase traffic theory and highway capacity. *Phys. A Stat. Mech. Its Appl.* **2004**, *333*, 379–440. [CrossRef]
5. Saidallah, M.; El Fergougui, A.; Elalaoui, A.E. A Comparative Study of Urban Road Traffic Simulators. *MATEC Web Conf.* **2016**, *81*, 05002. [CrossRef]
6. Corporation, C. Cubic | Transportation Systems, Trafficware Synchro 11 Features Webinar. 2020. Available online: https://youtu.be/aRFE9saslBw?list=PLndbN19Yb-nm62ShdaY{-}{-}Zjhtv3shfNAh&t=137 (accessed on 23 September 2021).
7. SeanAtLSA. Rail Simulation in TransModeler: Demo—3D View. 2009. Available online: https://youtu.be/iezgXbNzdjI (accessed on 23 September 2021).
8. Software, Caliper Corporation Mapping Software. TransModeler 3D. 2021. Available online: https://youtu.be/f4KblZwrup4?t=18 (accessed on 7 December 2022).
9. Anylogic. Roundabout. 2018. Available online: https://cloud.anylogic.com/model/b00c4c74-9f65-4fdf-9c27-0212db1e0bf6?mode=SETTINGS (accessed on 23 September 2021).
10. Lee, S.; Kim, J.; Song, H.; Kim, S. A 3-dimensional real-time traffic simulator considering the interaction among autonomous and human-driven vehicles. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014; pp. 1917–1918.
11. Multirotor. SUMO Traffic Simulation in Unity3D (3D Simulation). 2017. Available online: https://www.youtube.com/watch?v=a52U6CQQRcw (accessed on 23 September 2021).
12. Tchamna, R. Sumo (Simulation of Urban Mobility) Tutorial Part 1: Manually Creation of Network. 2018. Available online: https://youtu.be/9MyIABer_NY (accessed on 23 September 2021).
13. Mertin, S. Available online: https://github.com/SvenMertin/SUMO3d/search?q=ROUTE (accessed on 28 September 2021).
14. Bouchner, P. Interactive Driving Simulators—History, Design and their Utilization in area of HMI Research. *Int. J. Syst. Appl. Eng. Dev.* **2016**, *10*, 179–188.
15. Jeihani, M.; NarooieNezhad, S.; Bakhsh Kelarestaghi, K. Integration of a driving simulator and a traffic simulator case study: Exploring drivers' behavior in response to variable message signs. *IATSS Res.* **2017**, *41*, 164–171. [CrossRef]
16. That, T.N.; Casas, J. An integrated framework combining a traffic simulator and a driving simulator. *Procedia Soc. Behav. Sci.* **2011**, *20*, 648–655. [CrossRef]
17. Punzo, V.; Ciuffo, B. Integration of Driving and Traffic Simulation: Issues and First Solutions. *IEEE Trans. Intell. Transp. Syst.* **2011**, *12*, 354–363. [CrossRef]
18. Liao, X.; Zhao, X.; Wu, G.; Barth, M.J.; Wang, Z.; Han, K.; Tiwari, P. A Game Theory Based Ramp Merging Strategy for Connected and Automated Vehicles in the Mixed Traffic: A Unity-SUMO Integrated Platform. *arXiv* **2021**, arXiv:abs/2101.11237.
19. Unity. Physics Best Practices. Available online: https://learn.unity.com/tutorial/physics-best-practices#5c7f8528edbc2a002053b5b4 (accessed on 11 January 2021).
20. Dosovitskiy, A.; Ros, G.; Codevilla, F.; López, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017.

21. CARLA. Waypoints. Available online: https://carla.readthedocs.io/en/latest/core_map/#waypoints (accessed on 11 January 2021).
22. Olstam, J. Traffic Generation for The VTI Driving Simulator. In Proceedings of the Driving Simulator Conference-North America DSC-NA 2003, Dearborn, MI, USA, 8–11 October 2014.
23. Abdelgawad, K.; Henning, S.; Biemelt, P.; Gausemeier, S.; Trächtler, A. Advanced Traffic Simulation Framework for Networked Driving Simulators. *IFAC-PapersOnLine* **2016**, *49*, 101–108. [CrossRef]
24. Goldwire. RASY: 3D Real-Time Traffic Simulator. 2014. Available online: https://www.youtube.com/watch?v=_TvEc-qdiPs (accessed on 23 September 2021).
25. Our World In Data. World Population Growth. Available online: https://ourworldindata.org/world-population-growth (accessed on 23 September 2021).
26. Sun, L.-L.; Liu, D.; Chen, T.; He, M.-T. Road traffic safety: An analysis of the cross-effects of economic, road and population factors. *Chin. J. Traumatol.* **2019**, *22*, 290–295. [CrossRef]
27. Rae, A. The Yuxi Circle: The World's Most Densely Populated Area. Available online: https://www.visualcapitalist.com/cp/the-yuxi-circle-the-worlds-most-densely-populated-area/ (accessed on 19 July 2022).
28. Häcker, C.C.B.; Meixner, G. KoFFI—Improving Trust by having the Car as a Cooperative Partner. In Proceedings of the 9th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications, Oldenbourg, Germany, 24–27 September 2017.
29. Lê, H.; Pham, T.L.; Meixner, G. A Concept For A Virtual Reality Driving Simulation In Combination With A Real Car. In Proceedings of the 9th International Conference on Automotive User Interfaces and Interactive Vehicular Applications Adjunct, Oldenburg, Germany, 24–27 September 2017; pp. 77–82.
30. Martinez-Maradiaga, D.; Raci, E.; Lüönd, M.; Meixner, G. Control of a Three Degree-Of-Freedom Motion Platform with a Real Vehicle. *IFAC-PapersOnLine* **2019**, *52*, 7–12. [CrossRef]
31. The Knights of Unity. Wrong Import Settings are Killing Your Unity Game [Part 2]. Available online: https://blog.theknightsofunity.com/wrong-import-settings-killing-unity-game-part-2/ (accessed on 27 September 2021).
32. Unity. Audio Profiler Module. Available online: https://docs.unity3d.com/Manual/ProfilerAudio.html (accessed on 27 September 2021).
33. van Wageningen-Kessels, F.; van Lint, H.; Vuik, K.; Hoogendoorn, S. Genealogy of traffic flow models. *EURO J. Transp. Logist.* **2015**, *4*, 445–473. [CrossRef]
34. Drake, J.; Schofer, J.L.; May, A.D. *A Statistical Analysis of Speed-Density Hypotheses*; Highway Research Board: McLean, VA, USA, 1965.
35. Smulders, S. Control of freeway traffic flow by variable speed signs. *Transp. Res. Part B Methodol.* **1990**, *24*, 111–132. [CrossRef]
36. Daganzo, C.F. The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. *Transp. Res. Part B Methodol.* **1994**, *28*, 269–287. [CrossRef]
37. Hoogendoorn, S.; Bovy, P. State-of-the-art of vehicular traffic flow modeling. *J. Syst. Cont. Eng.* **2001**, *215*, 283–303. [CrossRef]
38. Payne, H.J. Freflo: A macroscopic simulation model of freeway traffic. In *Transportation Research Record*; Transportation Research Board: Washington, DC, USA, 1979.
39. Lighthill, M.J.; Whitham, G.B. On kinematic waves II. A theory of traffic flow on long crowded roads. *Proc. R. Soc. London. Ser. A. Math. Phys. Sci.* **1955**, *229*, 317–345. [CrossRef]
40. Hoogendoorn, S.P.; van Lint, H.; Knoop, V. *Dynamic First-Order Modeling of Phase-Transition Probabilities*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 85–92.
41. Gerlough, D.L.; Huber, M.J. *Traffic Flow Theory: A Monograph*; Transportation Research Board: Washington, DC, USA, 1975.
42. Gipps, P.G. A model for the structure of lane-changing decisions. *Transp. Res. Part B Methodol.* **1986**, *20*, 403–414. [CrossRef]
43. Michaels, R.M. Perceptual factors in car following. In Proceedings of the Second International Symposium on the Theory of Traffic Flow; OECD: Paris, France, 1963; pp. 44–59.
44. Wiedemann, R. *Simulation des Strassenverkehrsflusses*; Instituts für Verkehrswesen der Universität Karlsruhe: Karlsruhe, Germany, 1974.
45. Prigogine, I.; Andrews, F.C. A Boltzmann-Like Approach for Traffic Flow. *Oper. Res.* **1960**, *8*, 789–797. [CrossRef]
46. Prigogine, I. *A Boltzmann-like Approach to the Statistical Theory of Traffic Flow, in Theory of Traffic Flow*; Theory of Traffic flow; Elsevier: Amsterdam, The Netherlands, 1961.
47. Botma, H. *State of the Art Report" Traffic Flow Models" Part I: Introduction*; Institute for Road Safety Research SWOV: Leidschendam, The Netherlands, 1977.
48. Hoogendoorn, S.P.; Bovy, P.H. New estimation technique for vehicle-type-specific headway distributions. *Transp. Res. Rec.* **1998**, *1646*, 18–28. [CrossRef]
49. United States Department of Transportation. Traffic Analysis Toolbox Volume XII: Work Zone Traffic Analysis—Applications and Decision Framework. Available online: https://ops.fhwa.dot.gov/publications/fhwahop12009/sec3.htm (accessed on 25 September 2021).
50. Pipes, L.A. An operational analysis of traffic dynamics. *J. Appl. Phys.* **1953**, *24*, 274–281. [CrossRef]
51. Chandler, R.E.; Herman, R.; Montroll, E.W. Traffic dynamics: Studies in car following. *Oper. Res.* **1958**, *6*, 165–184. [CrossRef]

52. Cremer, M.; Ludwig, J. A fast simulation model for traffic flow on the basis of Boolean operations. *Math. Comput. Simul.* **1986**, *28*, 297–303. [CrossRef]
53. Yeo, H.; Skabardonis, A.; Halkias, J.; Colyar, J.; Alexiadis, V. Oversaturated freeway flow algorithm for use in next generation simulation. *Transp. Res. Rec.* **2008**, *2088*, 68–79. [CrossRef]
54. Barceló, J. *Fundamentals of Traffic Simulation*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 145.
55. Pariota, L.; Bifulco, G.N. Experimental evidence supporting simpler Action Point paradigms for car-following. *Transp. Res. Part F Traffic Psychol. Behav.* **2015**, *35*, 1–15. [CrossRef]
56. Hoffmann, E.R.; Mortimer, R.G. Scaling of relative velocity between vehicles. *Accid. Anal. Prev.* **1996**, *28*, 415–421. [CrossRef]
57. Mahut, M. *Behavioural Car Following Models*; Université de Montréal: Montréal, QC, Canada, 1999.
58. Helbing, D.; Molnar, P. Social force model for pedestrian dynamics. *Phys. Rev. E* **1995**, *51*, 4282. [CrossRef]
59. Fambro, D.b.; Koppa, R.J.; Picha, D.L.; Fitzpatrick, K. Driver Perception–Brake Response in Stopping Sight Distance Situations. *Transp. Res. Rec.* **1998**, *1628*, 1–7. [CrossRef]
60. Davoodi, S.R.; Hamid, H.; Pazhouhanfar, M.; Muttart, J.W. Motorcyclist perception response time in stopping sight distance situations. *Saf. Sci.* **2012**, *50*, 371–377. [CrossRef]
61. NSW. Safe Stopping Distance. Available online: https://www.nsw.gov.au/driving-boating-and-transport/roads-safety-and-rules/safe-driving/safe-stopping-distance#:~:text=As%20a%20general%20rule%2C%20when,be%20at%20least%20three%20seconds (accessed on 22 July 2022).
62. The Two Second Rule Explained: What Is the Importance of Safe Following Distance? Available online: https://kodsiengineering.com/two-second-rule/ (accessed on 22 July 2022).
63. (Vietnam), M.o.t. Circular Speed Limits and Safe Driving Distances for Motor Vehicles and Heavy-Duty Vehicles on Public Roads. Available online: https://thuvienphapluat.vn/van-ban/Giao-thong-Van-tai/Thong-tu-31-2019-TT-BGTVT-quy-dinh-ve-toc-do-va-khoang-cach-an-toan-cua-xe-co-gioi-423005.aspx (accessed on 21 July 2022).
64. Kozlowska, K.; Walker, P.; McLean, L.; Carrive, P. Fear and the Defense Cascade: Clinical Implications and Management. *Harv. Rev. Psychiatry* **2015**, *23*, 263–287. [CrossRef]
65. Hall, D. Driving Anxiety—Who Is in Danger? Available online: https://www.brake.org.uk/how-we-help/raising-awareness/our-current-projects/news-and-blogs/driving-anxiety-who-is-in-danger (accessed on 21 July 2022).
66. Higgs, B.; Abbas, M.M.; Medina, A. Analysis of the Wiedemann Car–Following Model over Different Speeds Using Naturalistic Data. *Procedia RSS Conf.* **2011**, 1–22. Available online: https://onlinepubs.trb.org/onlinepubs/conferences/2011/RSS/3/Higgs,B.pdf (accessed on 11 February 2022).
67. Balich, B.E. Wiedemann's Car-Following Model. Available online: https://www.linkedin.com/pulse/wiedemann-car-following-model-borhan-eddin-balich-mem-eit/ (accessed on 3 November 2022).
68. Hidas, P. Modelling lane changing and merging in microscopic traffic simulation. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 351–371. [CrossRef]
69. Azlan, N.N.N.; Rohani, M.M. Overview of application of traffic simulation model. *MATEC Web Conf.* **2018**, *150*, 03006. [CrossRef]
70. Machine, H. Tesla AutoPilot on Model X Became Disable in Vietnam Street. 2021. Available online: https://youtu.be/j3n1xBvYkP0 (accessed on 26 March 2021).
71. Huynh, D.N.; Boltze, M.; Vu, A.T. Modelling mixed traffic flow at signalized intersectionusing social force model. *J. East. Asia Soc. Transp. Stud.* **2013**, *10*, 1734–1749.
72. The Hazard Perception Test Keeping a Safe Distance to the Side. Available online: https://mylicence.sa.gov.au/the-hazard-perception-test/safe-distance-to-the-side (accessed on 3 November 2022).
73. Birmingham Council. The Law When Overtaking. Available online: https://www.birmingham.gov.uk/info/50122/cycling_in_birmingham/1396/driving_around_cyclists/3#:~:text=1.5m%20is%20a%20minimum,be%20needed%20to%20overtake%20safely (accessed on 3 November 2022).
74. Li, P.; Kusari, A.; Leblanc, D. A Novel Traffic Simulation Framework for Testing Autonomous Vehicles Using SUMO and CARLA. *arXiv* **2021**, arXiv:2110.07111.
75. Wang, P. Understanding social-force model in psychological principles of collective behavior. *arXiv* **2016**, arXiv:1605.05146.
76. Sorokowska, A.; Sorokowski, P.; Hilpert, P.; Cantarero, K.; Frackowiak, T.; Ahmadi, K.; Alghraibeh, A.; Aryeetey, R.; Bertoni, A.; Bettache, K.; et al. Preferred Interpersonal Distances: A Global Comparison. *J. Cross-Cult. Psychol.* **2017**, *48*, 002202211769803. [CrossRef]
77. Production, T.T. KINH NGHIỆM lái xe AN TOÀN—Cho những bạn lần đầu đi XE MÁY. Available online: https://www.youtube.com/watch?v=fhDigpN9KuU (accessed on 3 November 2022).
78. Kowalick, J. Human Functions: The Source of Psychological Inertia. Available online: https://the-trizjournal.com/human-functions-source-psychological-inertia/ (accessed on 3 November 2022).
79. Nguyen-Phuoc, D.Q.; Oviedo-Trespalacios, O.; Vo, N.D.Q.; Le, Q.T.; Su, D.N.; Cao, M.X. Prevalence and factors associated with turn signal neglect-related crashes among motorcyclists and car drivers in Vietnam. *Transp. Res. Part F Traffic Psychol. Behav.* **2020**, *73*, 38–49. [CrossRef]

80. Oeltze, K.; Schießl, C. Benefits and challenges of multi-driver simulator studies. *IET Intell. Transp. Syst.* **2015**, *9*, 618–625. [CrossRef]
81. Dong, C.; Ma, R.; Yin, Y.; Shi, B.; Zhang, W.; Zhang, Y. Traffic Conflict Analysis of Motor Vehicles and Nonmotor Vehicles Based on Improved Cellular Automata. *Math. Probl. Eng.* **2020**, *2020*, 2529816. [CrossRef]