Road Traffic Models for Validation of Autonomous Vehicle Functionalities

Lu Zhao

Road Traffic Models for Validation of Autonomous Vehicle Functionalities

Thèse de doctorat de l'Université Gustave Eiffel

École doctorale n° 532 :
MATHÉMATIQUES ET SCIENCES ET TECHNOLOGIES
DE L'INFORMATION ET DE LA COMMUNICATION (MSTIC)

Spécialité de doctorat : Mathématiques appliquées et applications des mathématiques
Unité de recherche : COSYS-GRETTIA

**Thèse présentée à l'Université Gustave Eiffel,
le 17/12/2021, par**

Lu ZHAO

Composition du Jury

Feng CHU Professeure, Université d'Evry	Présidente
Romain Billot Professeur, IMT Atlantique	Rapporteur
Tai-Yu Ma Chercheur, LISER, Luxembourg	Rapporteur
Georges Yannis Professeur, Université Nationale Technique d'Athènes	Examinateur
Maria Nadia Postorino Professeure University of Bologna	Examinatrice
Rodolphe Gelin Expert en IA, Renault	Co-Encadrante en entreprise

Encadrement de la thèse

Nadir Farhi Chercheur, UGE	Directeur de thèse
Zoi Christoforou Chercheuse, UGE, ENPC	Co-Encadrante
Nadia Haddadou Référent en connectivité, Renault	Co-Encadrante en entreprise
Yves Tourbier Expert en optimisation, Renault	Co-Encadrant en entreprise

Mathematics is the language with which God has written the universe.

– Galileo Galilei.

Acknowledgements

I am deeply grateful my thesis director Nadir FARHI, and my academic co-supervisor Zoi CHRISTOFOROU, for their guidance, support, and patience in helping me with all my work. I am also very grateful to my industrial supervisors, Nadia HADDADOU, Yves TOURBIER, Rodolphe GELIN, for their insightful advice during these three years.

I would like to express my gratitude to all the members of the jury, Romain BILLOT, Tai-Yu MA, Feng CHU, Georges YANNIS, Maria Nadia POSTORINO, Rodolphe GELIN for participating in this thesis committee and for providing their availability in my PHD thesis defense. My sincere thanks to the two reviewers, Romain BILLOT, Tai-Yu MA for taking the time to review my PhD thesis and for their very valuable comments.

I would like to sincerely thank the community of PhD students at Renault, especially Clara GANDREZ, Edouard LEURENT, Marc NABHAN, Jean MERCAT, Sonia ASSOU, Matthias COUSTE, Benoit LAUSSAT, Jia-Jun WU, Han-Wei GAO for the good times and atmosphere we had at Renault. It was a pleasure to discuss some technical issues with Edouard LEURENT and Jean MARCAT. It was very helpful for me to progress my work. I would like to thank Edouard LEURENT and Marc NABHAN for their PhD thesis. They gave me a good example to write my thesis.

I sincerely thank all my colleagues at Renault, Sebastien, Martine, Soukaina, Sopheaktra, Jean-Luc, Sylvain, Eric, Gregory, Nelson, and many others for the good times we spent together over meals and coffee breaks. I also extend my sincere thanks to all my colleagues at the GRETTIA laboratory at Gustave Eiffel University. I thank Yeltsin VALERO for discussing all my work and for helping with some collaborative projects.

Finally, and most importantly, I am deeply grateful to my parents and my boyfriend, Yan, for their unwavering emotional support and unfailing help. Thank you to my dog, Paola, for giving me companionship and for letting me walk out during the confinement of Covid.

Abstract

There is a growing interest in autonomous driving as it is expected that fully autonomous vehicles can reduce car accidents and improve overall traffic safety. However, autonomous driving is a complex process combining sensing, perception, prediction, computation, and decision. In addition, the traffic environment is dynamic and involves interactions among road users. Therefore, driving tests are essential to validate the autonomous vehicle's functionalities. Real-world driving tests seem to be a great challenge as fatal accidents cannot be prevented yet. Alternatively, performing driving tests by simulation can reduce time and cost, and avoid potentially dangerous situations. The increasing use of traffic simulation for many studies highlights the importance of a good understanding and modeling of human driving behavior.

This thesis mainly focuses on microscopic traffic modelling for human driving models, with the aim of creating, with numerical simulation, a realistic vehicular traffic, which is useful for the validation of autonomous vehicle's features. The main contributions of this thesis consist in 1) an approach of microscopic traffic modeling for car-collision generation in numerical simulation, based on real data-sets, 2) a Q-learning model for mimicking human lane-change in a highway road, and 3) a LSTM neural network model for replicating simultaneously car-following and lane-change behavior.

Car-collision generation in numerical traffic simulation. I proposed an approach of car-collision generation in numerical traffic simulation considering different car-following behaviors. After the investigation of different driver profiles in a real traffic data-set, I classified three driving profiles, where I distinguished *aggressive* and *inattentive* driver profiles from the *normal* profile. I then proposed to increase the proportion of the two 'extreme' driver profiles (aggressive and inattentive) in the whole traffic population by replacing the normal drivers, to simulate in a traffic simulator, SUMO (Simulation of Urban Mobility), and observe eventually the occurrence of car-collisions. I was able to formulate a relationship between the ratios of these two driver profiles over the entire driver population, and the number of car collisions. This analysis used part of the NGSIM 101 data-set and was validated on another part of the same data-set. I also studied the severity of the generated collisions. I found that collisions involved between an inattentive driver as the leader and an aggressive driver as the follower are the most frequent ones, while collisions between two inattentive drivers are the severest ones.

Lane change modeling using reinforcement learning. The second work in my PHD is on the lane change modeling, where a reinforcement learning model has been developed. The model aims to imitate real lane change decisions, based on the NGSIM traffic data-set. I proposed a

Q-learning model for the human lane change decisions. The model shows good performances in mimicking human decisions with up to 95% of success. Moreover, the model uses numerical traffic simulation (SUMO) to complete the unknown situations in the real data-set. We observed that 13% additional traffic conditions were created by the traffic simulation environment.

LSTM neural network for human driving behavior. In the third work of my PHD, I proposed an LSTM neural network model for car-following and lane-changing behaviors modeling on road networks. In this work, I proposed different models with different input designs and compared them. The selected model shows good performances on both predicting the longitudinal speed and the lateral position of cars. Moreover, the obtained results show that the selected model outperforms the classical IDM (Intelligent Driver Model) in the accuracy of replicating car-following behavior. The models were implemented on the NGSIM 101 and the HighD traffic data-sets.

Finally, I did some adaptation of proposed models and implemented them with different combinations in the SUMO traffic simulator. In these experiments, all private cars were controlled by our models, while other types of vehicles (trucks and motorcycles) were controlled by the default SUMO model. I compared the models and derived their performance for both the NGSIM 101 and the HighD traffic data-sets. I concluded that the AI-based models (a car-following model using neural network and a lane-changing model using Q-learning) provide better performance compared to the physical models. However, the simulation using those models was calibrated on the NGSIM data-set and was not found to be equally at representing HighD traffic. Finally, the NGSIM-calibrated IDM model is less sensitive to traffic data and can be used to simulate unknown traffic like the HighD data-set, but it requires re-calibration with the maximum speed of vehicles in the traffic.

Resumé des Travaux de Thèse

La conduite autonome soulève un intérêt croissant et l'on s'attend à ce que les véhicules entièrement autonomes puissent réduire les accidents de la route et améliorer la sécurité globale du trafic. Un véhicule autonome est un véhicule entièrement automatisé qui est capable de rouler sur la route dans toutes les conditions, sans intervention d'un conducteur. Cependant, le système de véhicule autonome est très complexe, et consiste en un système de perception, un système de décision et un système de contrôle. En outre, l'environnement routier est dynamique et implique de nombreuses interactions entre les usagers de la route. Par conséquent, les tests de conduite sont essentiels pour valider des fonctionnalités du véhicule autonome. Les tests de conduite dans le monde réel semblent être un grand défi, car des accidents mortels pourraient être provoqués par un véhicule autonome. Alternativement, la simulation numérique permet de réduire le temps et les coûts, et d'éviter les situations potentiellement dangereuses pour la validation des véhicules autonomes. L'intérêt croissant proté à l'utilisation de la simulation numérique du trafic routier souligne l'importance d'une bonne compréhension et d'une bonne modélisation du comportement de conduite humaine.

Dans cette thèse, le travail se concentre principalement sur la modélisation microscopique du trafic pour les modèles de conduite humaine, dans le but de créer des simulations numériques du trafic routier les plus réalistes possibles, à utiliser ensuite pour la validation et les tests des fonctionnalités de véhicules autonomes. Les principales contributions de cette thèse consistent en 1) une approche de modélisation microscopique du trafic pour la génération de collisions inter véhiculaires dans la simulation numérique, basée sur une base de données réelles, 2) un modèle par l'apprentissage par renforcement pour imiter le changement de voie humain sur une autoroute, et 3) un modèle de réseau neuronal, en utilisant le modèle LSTM, pour reproduire simultanément le comportement de poursuite et le comportement de changement de voies.

Génération de collisions par la simulation numérique de trafic Mon premier travail dans cette thèse est l'analyse du comportement du conducteur en ce qui concerne la loi de poursuite. En outre, dans le but de tester la réaction d'un véhicule autonome en cas d'accident dans le trafic, j'ai proposé une approche pour la génération de collisions entre véhicules. Après

avoir étudié différents profils de conducteurs dans un ensemble de données de trafic réelles, j'ai classifié trois profils de conducteurs, en distinguant les profils *agressif* et *inattentif*, du profil *normal*. J'ai ensuite proposé d'augmenter la proportion des deux profils de conducteurs (agressif et inattentif) dans la population totale du trafic, et de simuler le trafic à l'aide du simulateur SUMO (Simulation of Urban Mobility), pour observer les occurrences éventuelles de collisions de voitures. Par cette méthode, j'ai caractérisé une relation entre les ratios de ces deux profils de conducteurs sur l'ensemble de la population de conducteurs dans le trafic, et le nombre de collisions de voitures. Cette approche est étudiée sur une partie de l'ensemble de données NGSIM 101, et ensuite, validée sur une autre partie de cette base de données. J'ai également étudié la gravité des collisions générées dans la simulation. J'ai constaté que les collisions impliquées par un conducteur leader inattentif suivi par un conducteur agressif, sont les plus fréquentes; tandis que les collisions impliquées entre deux conducteurs inattentifs sont les plus graves.

Modélisation du changement de voie par apprentissage par renforcement Après avoir étudié le comportement des conducteurs en termes de poursuite (car-following), le deuxième travail de ma thèse porte sur la modélisation des changements de voie, où un modèle d'apprentissage par renforcement a été développé. Le modèle permet d'imiter les décisions réelles de changement de voie, en se basant sur l'ensemble de données de trafic NGSIM. En conséquence, le modèle montre de bonnes performances dans l'imitation des décisions humaines de changement de voie, avec jusqu'à 95 % de précision. En outre, le modèle utilise la simulation numérique du trafic (SUMO) pour compléter les situations inconnues dans la base de données réelles, où nous avons observé que 13 % des conditions de trafic supplémentaires sont créées par l'environnement de simulation du trafic.

Modèle de comportement de conduite humaine par le réseau neuronal LSTM Dans le troisième travail, j'ai proposé un modèle pour prédire le comportement de poursuite et le comportement de changement de voies en même temps, en utilisant les réseaux neuronaux LSTM. Dans ce travail, j'ai proposé différents modèles avec différentes conceptions des variables d'entrée; puis je les ai comparés. Le meilleur modèle sélectionné montre de bonnes performances pour la prédiction de la vitesse longitudinale et de la position latérale des voitures. De plus, les résultats obtenus montrent que ce modèle sélectionné est plus performant que le modèle classique IDM (Intelligent Driver Model) dans la précision de la reproduction du comportement de poursuite. J'ai d'abord implémenté les modèles proposés sur la base de l'ensemble de données NGSIM 101, ensuite sur l'ensemble de données HighD.

Enfin, j'ai implémenté les modèles proposés avec différentes combinaisons et quelques adaptations dans le simulateur de trafic SUMO, afin de simuler l'ensemble du trafic. Dans ces

implémentations, toutes les voitures sont contrôlées par nos modèles proposés, alors que les autres types de véhicules, y compris les camions et les motos, sont contrôlés par le modèle SUMO par défaut. J'ai comparé les modèles et dérivé leurs performances dans la simulation numérique du trafic, pour les deux trafics présentés respectivement dans la base de données NGSIM 101 et la base de données HighD. J'ai trouvé que les modèles basés sur l'IA (un modèle de poursuite utilisant un réseau neuronal et un modèle de changement de voie utilisant l'apprentissage par renforcement) fournissent de meilleures performances, par rapport aux modèles physiquesCependant, la simulation utilisant ces modèles proposés et calibrés avec la base de données NGSIM n'arrive pas à reproduire correctement le trafic de la base de données HighD. Le modèle IDM calibré par NGSIM est moins sensible aux données de trafic et peut être utilisé pour simuler un trafic inconnu comme les données de HighD, mais il requiert une recalibration avec la vitesse maximale des véhicules qui correspond à la condition de trafic dans le base HighD.

Contents

List of Acronyms

A

AD	Autonomous Driving
ADAS	Advanced Driver-Assistance Systems

D

DP	Dynamic Programming
DQN	Deep Q-Network

F

FCN	Fully-Connected Network

I

IDM	Intelligent Driver Model

M

MDP	Markov Decision Process
ML	Machine Learning
MLE	Maximum Likelihood Estimation

N

NN	Neural Network

R

RL Reinforcement Learning

Chapter 1

Introduction

Traffic has a significant importance in our daily life, and traffic problems attract researches' attentions, because of its complexity and the uncertainty that arises from human interaction. Researchers are interested in understanding the traffic flow dynamics and the human driving behavior for several reasons : 1) the increase in traffic demand and the limited capacity of road cause a lot of jams; 2) the construction of road infrastructure and driving rules limit human driving behavior; 3) traffic accidents occur every day, which may be caused by the inattention of human drivers; 4) finding useful methods to optimize traffic is essential to improve traffic conditions. This interest of understanding of road traffic dynamics and human driving behavior helps the improvement of road traffic modeling and numerical simulation. In addition, those research works lead to some further advances of some new applications in the future regarding to the intelligent traffic system, such as the novel driver assistance system, and some works on predicting the long-term evolution of traffic for optimizing traffic flow.

1.1 Problem statement

According to the UN, 68% of the world population is projected to live in urban areas by 2050 from 55% today (*United Nations, 2018* n.d.). Mobility needs in cities are thus expected to grow, at least proportionally to this rate, leading to an increase of car traffic. On the other hand, urban space is scarce and new road infrastructure is not always a feasible solution. The increasing imbalance between demand and supply is expected to aggravate congestion phenomena and exacerbate all relevant externalities: delays and travel time variability, user discomfort, noise and emissions, car crashes. As a result, **traffic management** remains a hot topic as it allows a better usage of the existing capacity to counter for additional demand. Also, **traffic analysis** continues to attract research attention as new data become available, new tools

become accessible (such as simulation), network optimization techniques are evolving, new technologies allow for the implementation of cutting-edge techniques.

Some researchers and policy makers believe that the arrival of **ICVs (Intelligent Connected Vehicles)** and **AVs (Autonomous Vehicles)** will provide a viable solution to urban mobility (Sjoberg et al., 2017; Jiajia Liu and Jianhao Liu, 2018; Laris, 2018). Others argue that ICVs may prove highly beneficial in terms of safety and mobility extension of certain groups (seniors, etc), but can bring some unintended effect caused by this new class of travel, such as empty runs (Pakusch et al., 2018). In any case, all analysts and stakeholders agree on the fact that the expected benefits are not to be observed in the short-term.

The car industry, in particular, needs to **validate and test** ICVs in real driving situations both in the field and on simulators; the latter mainly for the case of accidents and other extreme-case scenarios. A transitional period will first occur with increasing levels of automation where **the cohabitation of both human and non-human driving vehicles** need to be considered. **Highways** are expected to be concerned first as they provide a better controlled environment compared to city centers. The transitional phase attracts significant research interest as it is expected to be critical for the future full-scale deployment of ICVs and their characteristics: legislative, technical and operational.

The first step towards describing the critical transitional period is to **accurately describe driving behaviors**. Driving behavior includes a number of decisions made by the driver under variable driving conditions (speed choice, position on the pavement, distance to preceding vehicle, lane-changing maneuver, acceptance gap), but also some rather intrinsic attributes such as reaction time. It is known and well established that driver behaviors are strongly heterogeneous across individuals and, even, across choices of the same individual under different conditions and states. In fact, human factors that are not easily quantified and assessed, such as fatigue, have an important impact upon driving behavior. Consequently, an individual's behavior is hard to be predicted and the collective behavior of a group of drivers is predictable at a certain level of certainty and error. Most behavioral studies use average values of the choice parameters for statistical aggregation. However, a more detailed analysis would provide additional insight and contribute to the better understanding of traffic dynamics over the transitional phase.

Secondly, the aspect of the **safety outcome** should be put forward after **relevant quantitative analysis** in order to validate ICVs and increase their acceptability rate as well as the confidence of the general public. Currently, traffic data related to incidents and accidents (both pre- and post-crash) remain scarce. Collisions are rare events and naturalistic experiments or full-coverage CCTV (Closed-circuit television) data do not capture them in sufficient numbers to perform statistical analysis. As a result, crashes are poorly represented in traffic simulators also. Regarding pre-crash patterns, risky situations and traffic conflicts lead to collisions with a

specific probability that is not always data-proven. Regarding to post-crash patterns, simulators fail to reproduce the ground-truth and, in most cases, they simply remove all implicated vehicles (*SUMO safety, 2021* n.d.). Secondary accidents are thus not considered even though studies show that they represent half of total crashes. The problem is even greater when it comes to ICVs as crash data are almost non-available. In conclusion, a more sophisticated representation of crashes in traffic simulators proves to be necessary for the test and validation of ICVs.

Thirdly, nowadays many **tools and knowledge bases** become available and have not been fully exploited in the purpose of better understanding human driving and proposing high-performance solution of this cohabitation of human diving and non-human driving. Those tools include (i) new disaggregated data (especially from traffic cameras and drones), (ii) latest developments of traffic software integrating ICV functionalities, (iii) new powerful methods (such as deep learning, reinforcement learning) suitable for modelling from data, and (iv) latest research results from a constantly growing corpus of literature. As a result, we identify here an urgent need for integration and synthesis of existing tools and knowledge in order to propose a holistic approach for the representation and assessment of the cohabitation period, its possible characteristics and risks.

In response to previous remarks and observations, **the present thesis focuses on traffic modelling in the context of validation and testing of ICVs especially during the critical period of cohabitation of human and non-human driving**. The case studies target urban highways and heterogeneous human driving behaviors under variable traffic dynamic states. From a methodological standpoint, effort was made to integrate many new tools that had not been fully exploited before: real world traffic data, sophisticated modeling and simulation techniques and statistical methods.

1.2 Industrial context

The present PhD thesis was realized in an industrial context as it was partly financed by Renault, the leading car industrial stakeholder in France. This paragraph summarizes industrial challenges.

1.2.1 Autonomous Vehicles

An autonomous vehicle is a vehicle that is capable of sensing its environment and moving safely itself in the road traffic. Autonomous vehicles have been developed following a traditional robotics pipeline and there are three main modules in the autonomous vehicle system: Perception, Decision, Control. I show the architecture and an image of autonomous vehicle

in the Figure 1.1. The autonomous vehicles need to be equipped with a variety of sensors to

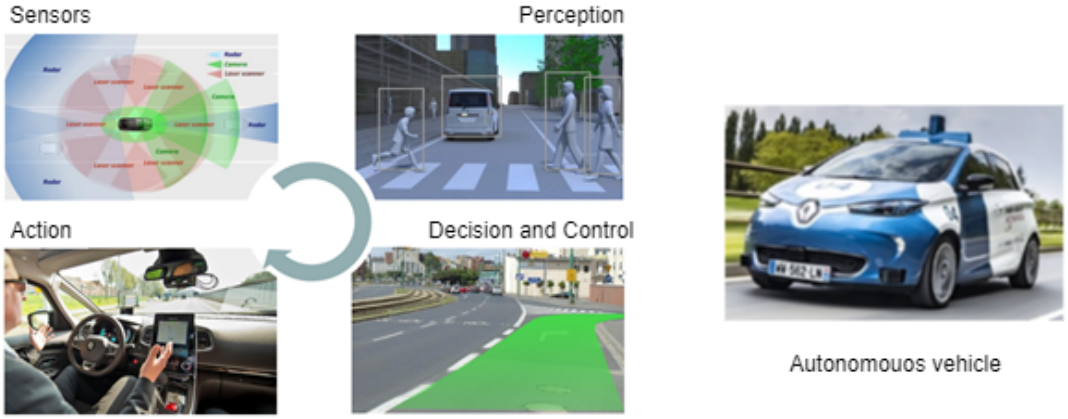


Figure 1.1 – Autonomous vehicle architecture

perceive their surroundings, such as radar, lidar, sonar, GPS and inertial measurement units. The Perception module takes sensor data as input and produces a high-level reconstruction and recognition of the scene. The Decision module then chooses the desired behavior and trajectory of the vehicle, based on the current situation Finally, the Control module manipulates forces, by the way of steering and throttle controls, to follow the desired trajectory decided by the decision module. The decision and control is the most complex part, and in the context of Autonomous Driving, the decision and control module is often implemented with a hierarchical structure whose layers work at different timescales.

- **Route Planning**: The highest level is the route planning, in which, vehicle chooses a route through the given road network. Such a route is formulated as a problem of finding a cost-minimization path on a road network.

- **Behavior Decision Making**: After choosing the optimal path from route planning, the autonomous vehicle should be able to navigate the selected route and interact with other traffic components depending on the rules of road, road conditions, and signals of infrastructure. In this layer, an appropriate behavior can be selected based on respecting every constraint. So in this layer, it takes optimal route path decided from *Route Planning*, combining the information of other agents in the road (other vehicles around, obstacles), road signs (eg. speed limit), rules of road (eg. propriety of right) as inputs.

- **Motion Planning**: Once the *Behavioral Layer* decides on which driving scenario to be performed, the decision arrive into the motion planning to get a path and trajectory considering comfortable of passenger and avoidance of collision with obstacle.

- **Vehicle Control**: In order to execute the trajectory calculated in *Motion Planning* path, the vehicle controller is used to select appropriate inputs to figure out the motion and correct

errors. This layer aims to enhance and make the system more stable. It takes path and trajectory found in before layers and puts out the steering, throttle or brake commands.

The system of autonomous vehicle control is complex, each layer carries out an appreciate decision in each time step while taking the result of layer before and other environment and estimation information as inputs. Otherwise, a good decision making depends of the capacity of estimation environment information and decision model. Every decision need to be computed based on scenario (task of behavior decision making), road conditions (highway, urban, national way), surrounding vehicles types (moto, normal car, truck) and their behavior. The final motion decision need consider about safety, comfort of passenger and capacity of the vehicle.

1.2.2 Validation of autonomous vehicles

As indicated above, the autonomous vehicle represents a highly complex system. A failure can occur during any stage of this process, which will result in a wrong behavior on the road. Nowadays, the self-driving software testing is realized by million lines of codes included in all the algorithms of perception, decision, and control for the autonomous vehicle. However, it is not enough to ensure the testing of the autonomous vehicle facing on all the various situations in the traffic (Koopman and Wagner, 2016; Koopman and Wagner, 2017; Hutchison et al., 2018). According to the authors in (Kalra and Paddock, 2016), to make a car 95% as safe as an experienced driver, autonomous vehicles would have to be driven hundreds of millions of miles and need to be failure-free. Even to make sure the autonomous vehicle is 10% or 20% safer than humans, hundreds of billions of miles autonomous driving are needed to demonstrate their reliability in terms of fatalities and injuries (Kalra and Paddock, 2016). Alternatively, due to the increase of computer simulation techniques, traffic simulations can benefit the training and testing of autonomous vehicles (Belbachir et al., 2012; Chao et al., 2020). In simulation methods of the validation of autonomous vehicle, we can differentiate three principal methods:

1. **Data-driven resimulation** : Representing the collected real driving scene directly in numerical simulators of data recorded during real driving testing to resimulate what happens in the road numerically. In this way, we can insert autonomous vehicle in the traffic to see how it works in the traffic. The resimulation of driven data allows checking the performance of each components of autonomous vehicle during driving. That is the most simple way by numerical simulation to make sure if the perception system recognizes well the traffic scene, if the decision system makes a good decision, and the controller works well. Resimulation could be the first test step for some software updating.

2. **Model-based simulation**: This approach is entirely numerical and it generates numerical traffic data. We can easily parameterize the behavior of each vehicle and obtain various traffic conditions. In this approach, the test of autonomous vehicle is expected to be divided into two types : Scenario-Based Validation and Validation in a large-scale traffic.
    - **Scenario-based Validation**: In this method, we define specific scenarios to test separately the functionalities of the autonomous vehicle. such as scenario for testing Adaptive Cruise Control (ACC), scenario for testing the Automatic Emergency Braking (AEB), scenario for testing lane changing and so on.
    - **Validation in a large-scale traffic**: Finally, we expect the simulation of traffic can achieve as a high level of reality, that the autonomous vehicle can drive in simulation like driving in the real world for a long trajectory.
3. **Virtual immersive simulator**: A person drives a real vehicle that is connected to the environment through simulator software using virtual reality glasses or a 180-degree high-resolution screen. This technique is useful in studying the ergonomics and behavior of drivers using autonomous driving modes, while the realism of the scenario remains always a challenge.

All these simulation approaches of the validation of autonomous vehicle need a reconstruction of a realist traffic in simulation environment. That encourages a hard research work on the road traffic modelling and human driving behavior modelling.

1.2.3 Some relative projects in Renault

Renault, as known as one of the biggest automobile manufacturers in the world, also has a highly interest of developing the autonomous vehicle and investing in autonomous vehicle simulation to complete tasks of the validation of autonomous vehicle, by the real-driving testing. Renault participates in some European projects on the validation of autonomous vehicle. The project PEGASUS (Pütz et al., 2017) develops a validation framework with a database containing relevant traffic scenarios. Scenarios can then be generated for a single use case by choosing certain values for all inputs, and can be launched into a Renault in-house simulation software called SCANeR Studio (Champion et al., 1999). As well, Renault collaborates with other participants in the project UDRIVE (Nes et al., 2019), which has established a rich cross-European naturalistic driving database to study the human driving behavior. This databases provides extensive, reliable insights into driving behavior in real traffic as a foundation for improving the safety and sustainability of European road traffic. In addition, Renault has a long-term strategy on launching a massive simulation more and more accessible, which is useful to complement real test-driving. However, the models in SCANeR Studio for simulating the traffic have many limitations, which avoid to have human-like vehicle movements, that is

why the models used in SCANeR Studio need to be improved and to be updated. Moreover, the simulation of several accidents in the traffic via numerical simulation is interesting for the validation of the autonomous vehicle in dealing with traffic accidents. However, for the generation of traffic with accidents, this modelling mechanism is even not completely cited in the existent traffic simulators for the research work, such as SUMO, Vissim, Aimsum. Therefore, the proposition of simulating accidents is also an objective of this thesis.

1.3 Scope and objectives

Therefore, we are interested in this thesis in microscopic traffic models, focusing on the human driving maneuvers (longitudinal and lateral decision making in driving task), related to vehicular interactions. By this kind of models, we would like to build a realistic traffic environment for testing and validating some autonomous vehicle's features in the future. There are some interesting topics for this thesis:

Car-following models Car-following models describe the longitudinal behavior of the vehicle. A car-following model is complete if it is able to describe all situations including acceleration and cruising in free traffic, following other vehicles in stationary and non-stationary situations, and approaching slow or standing vehicles, and red traffic lights. However, CF models have been well developed compared to lane changing models. It is important to well calibrate car-following models.

Lane changing models Lane changing models describe the lateral behavior of the vehicle. Considering LC behavior, the modeling efforts are not as many as in the CF behavior. This is due to the complexity of the LC behavior, which is affected by the surrounding vehicles (of the considered vehicle) and by the traffic flow environment (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007). In some reviews of lane changing models (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007; Moridpour, Sarvi, and Rose, 2010; Z. Zheng, 2014), the authors present several rule-based lane changing models, but they tend to exhibit limited performance due to the uncertainty and to the complexity of the driving environment. For this reason, Machine Learning methods, and especially Reinforcement Learning (RL) ones, provide an alternative approach, which has shown a great success in many different domains, such as robotics, video game playing, dialogue chat bot, etc.

A simultaneous model for car-following and lane changing As well known, two driving behaviors are generally distinguished in the microscopic traffic modeling: Car-Following (CF) and Lane Change (LC), which describe respectively longitudinal and lateral vehicular

movements. Researchers begin to consider the development of models which can describe the car-following as well as the lane changing behaviors, in the same time. This is important because these two behaviors are related each other.

Traffic accidents One of the major challenges of autonomous vehicles is communicating with other vehicles and accurately recognizing the patterns of human driving behavior in mixed traffic (Schwarting, Alonso-Mora, and Rus, 2018). As well known, human driving behavior is not perfect and million deaths occur each year due to traffic accidents, among that, 90% of accidents are produced because of driver's mistakes and inattention(Singh, 2015). Therefore, the process of testing autonomous vehicles in dealing with traffic accidents is very important. However, the relationship between different driver profiles and the car-collision occurrences has not been understood clearly. Meanwhile, the procedure of car-collisions generation in traffic numerical simulators is not fully integrated. By this point of view, it is interesting to identify the driver profiles in a traffic data, and understanding their relationship with accident occurrence.

To clarify, the purpose of this thesis is to provide additional insight in the test and validation of ICVs by exploiting and further developing recently developed methods and tools. The specific thesis objectives can be summarized as follows:

1. Enhancing car-collision generation in numerical traffic simulation
2. Exploring the impact of the presence of different driving profiles in the same traffic stream in terms of both car-following behavior and collision severity
3. Enhancing the realism of lane-changing behaviors using reinforcement learning
4. Enhancing the realism of simultaneous car-following and lane-changing behaviors using neural networks

The level of traffic analysis adopted was microscopic. The disaggregated traffic databases used were the NGSIM 101 dataset and the HighD dataset. The main traffic simulator was SUMO (Simulation of Urban Mobility). All the data and tools are fully described in the following chapters.

1.4 Outline and Contributions

The remainder of the manuscript is organized as follows:

Chapter 2 reviews the state of the art for the traffic modelling, from classical models to the models using artificial intelligence.

Chapter 3 presents our first research work on the analysis of driver behavior and proposition of a novel approach of vehicular-collision generation in numerical traffic simulator. This work is based on a real traffic dataset, the NGSIM dataset, and considered only car-following behavior.

Chapter 4 describes the second work in my PHD, where we proposed a lane changing model using Reinforcement Learning. This is a model with imitation of real lane change decision from the NGSIM dataset.

In Chapter 5, a long short-term memory (LSTM) neural network model is proposed to replicate simultaneously car-following and lane-changing behaviors. This work is done first using the NGSIM dataset, and then validated using the HighD dataset.

Chapter 6 illustrates the simulation results on simulating a whole traffic, where we control all the vehicles in the traffic using our proposed models. In order to understand the performance of each model, we implemented these models with some adaptation in SUMO for the case of NGSIM and HighD and analyzed data sensitivity of all proposed models.

List of publications

Publications in international conferences with proceedings

- Lu Zhao, Nadir Farhi, Zoi Christoforou, and Nadia Haddadou (2020). A data-based traffic modeling approach for inter-vehicular collision simulation. In *Proceedings of 8th Transport Research Arena TRA 2020, April 27-30, 2020, Helsinki, Finland*. TRA (used in Chapter 3)
- Lu Zhao, Nadir Farhi, Zoi Christoforou, and Nadia Haddadou (2021b). Driver Behavior Analysis and Inter-Vehicular Collision Simulation Approach. In *Proceedings of ICITS002 2021: 15. International Conference on Intelligent Transportation Systems December 27-28, 2021 in Vienna, Austria* (used in Chapter 3)
- Lu Zhao, Nadir Farhi, Zoi Christoforou, and Nadia Haddadou (2021c). Imitation of Real Lane-Change Decisions Using Reinforcement Learning. *IFAC-PapersOnLine* 54.2, pp. 203–209 (used in Chapter 4)

Publications of journals

- Lu Zhao, Nadir Farhi, Zoi Christoforou, and Nadia Haddadou (2021a). Analysis of driver behavior and inter-vehicular collision : a data-based traffic modeling and simulation approach. *Journal of advanced transportation* (used in Chapter 3)

Submission of conferences

- Lu Zhao, Nadir Farhi, Zoi Christoforou, and Nadia Haddadou (2022). Long Short-Time Memory Neural Networks for Human Driving Behavior Modelling. In *Transport Research Arena, TRA Conference – Lisbon 2022* (used in Chapter 5)

Collaborations not presented in this thesis

- Clara Gandrez, Fabrice Mantelet, Lu Zhao, Aoussat Ameziane, Francine Jeremie, and Eric Landel (2019). Modelization of human risk feeling during near-crashed situations in autonomous vehicle

Chapter 2

Literature Review

In the recent years, many contributions have been made on road traffic modeling, based on several approaches and disciplines, including physics, mathematics, machine learning, behavioral psychology, etc. That leads to a better understanding of driver behavior and traffic flow dynamics. Mathematical models can be roughly classified into macroscopic, microscopic and mesoscopic models. Macroscopic models describe the collective state of traffic in terms of space-time fields for local density, speed, and flow; while microscopic models describe traffic flow from the perspective of individual drivers and vehicles. Mesoscopic models, on the other hand, combine the strengths of microscopic and macroscopic models to simulate traffic at different levels of detail. This chapter reviews the most recent approaches and methods of road traffic modeling and simulation.

Contents

2.1 Overview of road traffic modelling and simulation

As indicated above, road traffic models can be classified by distinguishing the scale of modeling. In one of the most recent publication of the SUMO (Simulation of Urban Mobility) traffic simulator (Lopez et al., 2018), four scales have been considered:

- **Macroscopic** : presents traffic flow on the basis of continuous fluid dynamics theory, which handles the description of aggregated information such as density (number of vehicles per km), mean speed (km/h), and the vehicular flow (number of vehicles per hour). Macroscopic models describe vehicle's behavior and interactions at a low level of detail, focusing on reproducing aggregated behaviors measured with quantities such as flow density and traffic flux (Chao et al., 2020). At this level of modelling, the vehicles' behavior is homogeneous. Such models are limited for simulating street level traffic, which requires the representation of detailed interactions among individual vehicles.

- **Microscopic** : each vehicle and its dynamics are modeled individually. Microscopic models are developed for modelling heterogeneous behaviors of agents in the traffic, for diverse road typologies, and interactions among surrounding vehicles. In microscopic models, car-following models and lane changing models are two fundamental ones. They describe respectively the vehicle longitudinal and lateral movement.

- **Mesoscopic**: a mixture of macroscopic and microscopic. Mesoscopic models are an intermediate approach between macroscopic and microscopic approaches. The core idea of the mesoscopic models is to describe traffic flow dynamics in an aggregate manner while representing the behaviors of individual drivers using probability distribution functions(Chao et al., 2020). Herein, vehicle movement is mostly simulated using queue approaches and single vehicles are moved between such queues.

- **Sub-microscopic** : each vehicle and also functions inside the vehicle are explicitly simulated, for example, the engine's rotation speed in relation to the vehicle's speed or the driver's preferred gear switching actions. In such models, the vehicle dynamics are generally complex.

One of the advantages of macroscopic models is the computation and simulation time, which is very low compared to the ones needed in microscopic models. Consequently, large networks can be modeled and numerically simulated. However, the vehicle individual behavior is not considered in macroscopic models. For the validation of autonomous vehicle features, one needs to test and validate the safety aspects of the autonomous vehicle's decision making for its movements. It is obvious that the numerical simulation of the local interactions of vehicles is important in this case. Microscopic and/or sub-microscopic traffic models are then necessary. They are more accurate for the modeling of individual vehicle behavior. Sub-microscopic traffic models allow more detailed computations compared to simple microscopic simulations.

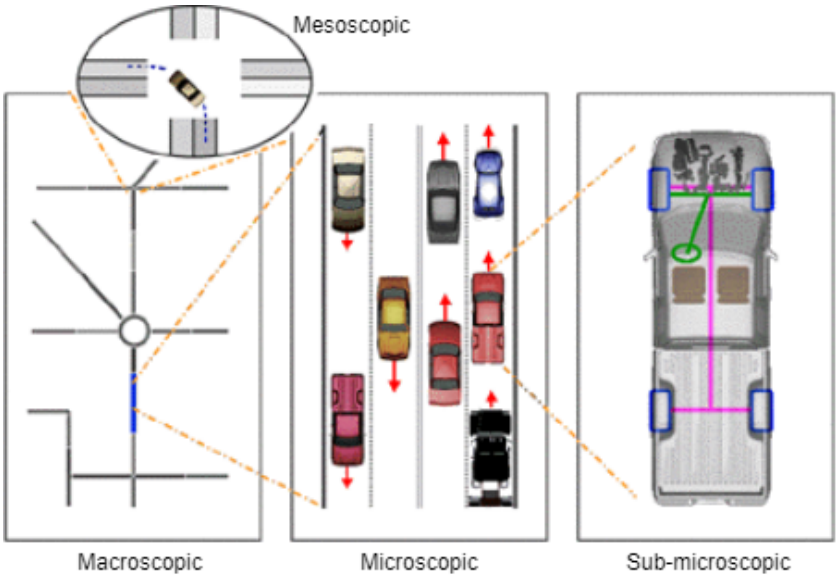


Figure 2.1 – Four different approaches for traffic modelling and simulation

Furthermore, sub-microscopic models require longer computation times, which restrains the size of the networks to be simulated.

We are interested in this thesis in microscopic traffic models, focusing on the human driving manoeuvres (longitudinal and lateral decision making in driving task), related to vehicular interactions. By this kind of models, we would like to build a realistic traffic environment for testing and validating some autonomous vehicle's features in the future.

My literature review on the traffic flow modeling is mainly based on one book and some research articles. The book (Treiber and Kesting, 2013b) offers a comprehensive and didactic account of the different aspects of vehicular traffic flow dynamics, and the calibration mechanism for the traffic flow models. In (Kotusevski and Hawick, 2009; Chao et al., 2020), the authors summarized the traffic simulation techniques including different simulation soft-wares, traffic models, traffic data-sets, with some new applications in Autonomous Vehicles. In this chapter, I will begin with a short talk on macroscopic models, then give a review on road traffic microscopic models, with more details.

2.2 Little words on macroscopic traffic modelling

Macroscopic traffic models are generally based on a flow-density (or equivalently speed-density) relationship, traditionally called the fundamental diagram. The latter is obtained from collected traffic data, and presents generally at least two traffic phases: a free flow phase where the vehicular flow increases with the car-density, and where the vehicles run freely; and a

congestion phase where the vehicular flow decreases with the car-density, and where congestion appears. Macroscopic models are generally capable to reproduce important phenomena in road traffic such as the backward propagation of congestion, as well as the shock-wave propagation. One of the most known macroscopic model is the first order one, called the model of Lighthill Whitham and Richards (LWR) (Khan and Gulliver, 2018). This model combines simply the conservation equation of traffic with a fundamental diagram curve formula. Several other models, mainly second order ones have been developed, in particular to include the acceleration variable into the traffic modeling, such as Payne–Whitham (PW) model (Grewal and Payne, 1976).

2.3 Classical microscopic traffic models

Car-Following (CF) and Lane-Changing (LC) maneuvers describe vehicular longitudinal and lateral interactions on the road respectively. They are two fundamental models for the microscopic traffic modeling and simulation. CF models describe the movements of a vehicle in response to the actions of its leading vehicle (LV), while LC models describe the choice of lane in a multi-lane roads. CF models generally compute the vehicle continuous longitudinal movement, e.g. speed or acceleration. Lane changing models are different from car-following models. The decision of LC is a discrete choice, i.e., performing a lane change, or not. Nevertheless, discrete decisions and actions can also pertain to the longitudinal dynamics; for example, when approaching a yellow traffic light, drivers have to decide whether it is safe to pass or not safe, need to stop. In a traffic simulation system, it is often essential to apply discrete choices, that involves several levels of decision:

1. *Strategic level* aims at choosing the destination, the mode, and the route.
2. *Tactical level* includes changing lanes to enter a priority road. This includes also cooperative behavior such as allowing another vehicle to merge at a point of lane closure. However, modeling the tactical level is notoriously difficult and is only attempted in the most elaborate commercial simulators (Treiber and Kesting, 2013b).
3. *Operative level*, the actual decision is made at this level; e.g. the decision of changing lane, stay on the same lane, stop for the yellow traffic light; etc.
4. *Post-decision level*, the actions executing the decision are simulated, e.g. performing the lane change or keeping to one's lane, waiting or entering a priority road, or cruising versus stopping at the traffic light.

The CF and LC models we consider here are restricted to the operative and post-decision levels.

2.3.1 Car-following models

A car-following model describes the movements of vehicles in response to the actions of their respective leading vehicles (LV). In a CF model, variables on the individual movement of each vehicle are considered: x_n (position), v_n (speed), a_n(acceleration); see Fig. 2.2, where n is the index of the considered (following) vehicle, and $n-1$ is the index of the leader one. Some relative variables are also used, such as $\Delta x = x_{n-1} - x_n$ (the inter-vehicular distance, which is the rear-to-front bumper distance of two vehicles), $\Delta v = v_{n-1} - v_n$ (the relative speed), and $\Delta a = a_{n-1} - a_n$ (the relative acceleration).

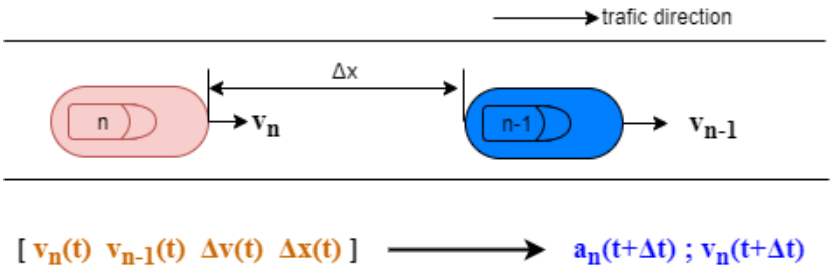


Figure 2.2 – Car-following scenario description

Car-following models are based on the principle of stimulus-response, assuming that every (following) car responds at time $t + \Delta t$ to a stimulus of its leader car at time t. The time step Δt represents the discritization in time, including implicitly the reaction time of the following car.

Among the main car following models, we distinguish First order models (formula (2.1)) from second order ones (formula (2.2)), where $\dot{x}_n(t)$ and $\ddot{x}_n(t)$ denote respectively the car speed and the car acceleration of vehicle n at time t.

$$\dot{x}_n(t + \Delta t) = f(\Delta x(t), \dot{x}_{n-1}(t)) \tag{2.1}$$

$$\ddot{x}_n(t + \Delta t) = f(\Delta x(t), \Delta v(t), \dot{x}_n(t), \dot{x}_{n-1}(t), \ldots) \tag{2.2}$$

Several interesting reviews on car-following can be found in the literature (Brackstone and Mark, 1999; Mohammad and Z. Zheng, 2014; Aghabayk, Sarvi, and Young, 2015). I briefly present here the most commonly used. The first car-following models were proposed in 1950s. Since then, a large number of models were developed; we cite the Gazis-Herman-Rothery model (Gazis, Herman, and Rothery, 1961), the optimal velocity model (Bando et al., 1995), the models proposed by Helly (Helly, 1959), Gipps (Gipps, 1981), and Wiedemann (Wiedemann, 1974), and the intelligent driver model (Treiber, Hennecke, and Helbing, 2000). Car-following models can be classified into five categories (M. Zhu, Xuesong Wang, Tarko, et al., 2018): stimulus-based, safety distance, desired measures, optimal velocity, and psycho-physical models.

Gazis-Herman-Rothery (GHR) model

The GHR model (Chandler, Herman, and Montroll, 1958) is a stimulus–response model in which drivers perform their acceleration/deceleration depending on car-speeds, relative car-speeds with respect to leading vehicles, and inter-vehicle distances.

$$a_n(t+\Delta t) = \alpha v_n(t)^{\beta} \frac{\Delta v(t)}{\Delta x(t)^{\gamma}} \tag{2.3}$$

Where, α, β, γ are parameters of the model.

Gipps model

The Gipps model (Gipps, 1981) was developed based on the required safety-distance between two consecutive vehicles to avoid car-collision. The speed of the follower vehicle is chosen to always maintain a minimum distance with the leader vehicle, in order to prevent collision, even when the leader vehicle brakes suddenly. In this model, the safe gap with the leading vehicle needs to exceed a minimum value s_0 plus a stopping distance. The worst case scenario is when the leading vehicle suddenly decelerates and stops. In this case, the following vehicle needs to brake to stop too. To guarantee safety, the final distance to leading vehicle should not become smaller than the minimum gap s_0. The safe distance can be formulated as follows :

$$s_{safe} = s_0 + v_n T + \frac{v_n^2}{2b} - \frac{v_{n-1}^2}{2b} \tag{2.4}$$

where $\frac{v_n^2}{2b}$ and $\frac{v_{n-1}^2}{2b}$ describe the necessary stopping distance for the following vehicle n and for the leading vehicle $n-1$ respectively, b is the maximum deceleration, and T is a reaction time for the following driver. In this way, the safe speed and the real speed taken in $t+\Delta t$ of vehicle can be formulated as follows :

$$v_{safe}(\Delta x, v_{n-1}) = -b\Delta t + \sqrt{b^2\Delta t^2 + v_{n-1}^2 + 2b(\Delta x - s_0)} \tag{2.5}$$

$$v(t+\Delta t) = min[v + a\Delta t, v_0, v_{safe}(\Delta x, v_{n-1})] \tag{2.6}$$

where a is the vehicle maximum acceleration, v_0 is the desired speed for the driver; see Tab. 2.1 where examples for the values of the Gipps model parameters are given.

Krauss model

The Krauss model (Krauß, 1998) is also based on a safety distance and it is car-following model by default in SUMO (Simulation of Urban Mobility) traffic simulator. As for the Gipps model, the Krauss model calculates directly the follower speed $v(t+\Delta t)$ with a delay of reaction

Table 2.1 – Parameters of the Gipps' model and typical values in different scenarios (Treiber and Kesting, 2013b)

Parameter's name	Highway	City
Desired speed v_0	120 km/h	54 km/h
reaction time T	1.1 s	1.1 s
Minimum distance s_0	3.0 m	2.0 m
Acceleration a	1.5 m/s2	1.5 m/s
Deceleration b	1.0 m/s2	1.0 m/s

time T. The speed is carried out from a desired speed with a perturbation of random human imperfection (μ). The Krauss model is an extension of the Gipps one, in which, the safe distance is taken into account to avoid car-collisions. So that the desired speed is also carried out from the comparison among: safe speed, maximal speed, and the maximal speed supposed using maximal acceleration. The safe speed is submitted to the condition of collision avoidance.

$$v(t+\Delta t) = max(0, v_{des}(t) - \mu) \tag{2.7}$$

$$v_{des}(t) = min[v + a\Delta t, v_0, v_{safe}(\Delta x, v_{n-1})] \tag{2.8}$$

$$v_{safe}(\Delta x, v_{n-1}) = v_{n-1}(t) + \frac{\Delta x(t) - v_n(t)T}{\frac{v_{n-1}(t)+v_n(t)}{2b} + T} \tag{2.9}$$

In this model, b, a are the maximal deceleration and maximal acceleration, respectively, v_0 is the desired speed of the following driver, and T is its reaction time.

Wiedemann model

The Wiedemann model (Wiedemann, 1974) and Wiedemann 99 (VISSIM, 2012) are psycho-physiological models. The Wiedemann model is the car-following model by default in Vissim simulator. It describes the psycho-physiological aspects of driving behavior in terms of four discrete driving regimes, for different modes of operation, divided into 'no reaction zone' (free-road), 'closing in', 'must decelerate', and 'car-following' by human perceptual. In each of these regimes, different acceleration functions $a(s, v, \Delta v)$ apply. The boundaries between the regimes are given by nonlinear equations of the form $f(s, v, \Delta v) = 0$ defining curved areas in three-dimensional state space $(s, v, \Delta v)$, spanned by the exogenous variables (M. Zhu, Xuesong Wang, and Y. Wang, 2018).

Full velocity difference(FVD) model

The FVD model (Jiang, Wu, and Z. Zhu, 2001) resembles to the optimal velocity model developed in (Bando et al., 1995). The acceleration function consists in a term proportional to an

optimal velocity $v^*(\Delta x_n)$ and a term that takes into account the velocity difference as a linear stimulus :

$$a(t+\Delta t) = \alpha[\ v_n^*(\Delta x(t) - v_n(t))]\ + \lambda \Delta v(t) \tag{2.10}$$

$$\lambda = \begin{cases} \lambda_0 & \text{if } \Delta x(t) \leqslant s_c \\ 0 & \text{if } \Delta x(t) > s_c \end{cases} \tag{2.11}$$

where α and λ_0 are constant sensitivity coefficients, s_c is a threshold between car-following and free driving, and $v_n^*(\Delta x(t))$ is the optimal velocity depending on the headway to the lead vehicle, which can be defined as :

$$v_n^*(\Delta x(t)) = \frac{v_0}{2}\left[\tanh(\frac{\Delta x(t)}{b} - \beta)\right] - \tanh(-\beta) \tag{2.12}$$

where v_0 is the desired speed of the following vehicle, β and b are parameters.

Intelligent Driver Model (IDM)

The IDM (Treiber, Hennecke, and Helbing, 2000) output is the acceleration chosen by the follwing driver :

$$a(t+\Delta t) = a\left[1 - \left(\frac{v_n}{v_0}\right)^{\delta} - \left(\frac{s^*(v_n, \Delta v)}{\Delta x}\right)^2\right], \tag{2.13}$$

where

$$s^*(v_n, \Delta v) := s_0 + \max\left(0, v_n T + \frac{v_n \Delta v}{2\sqrt{ab}}\right), \tag{2.14}$$

where T is a minimal time headway, a and b are maximal acceleration and deceleration respectively, s_0 is the minimal gap and v_0 is the desired speed which describe also the (maximal) desired speed. This model can describe vehicles' behaviors in a free road and the car-following dynamics.

On a free road, the acceleration respects a relation with the current speed : $a[1 - (\frac{v_n}{v_0})^{\delta}]$. It decreases to zero when approaching the desired speed. The parameter δ controls the reduction of the acceleration value. The decision of decelerating depends on the desired distance s^* and the actual gap. When the actual gap is close to s^*, the acceleration is nearly zero. Meanwhile, when approaching a slower vehicle, s^* increases and it decreases when the leader is faster. In addition, see Table 2.2 and Table 2.3 for the examples of IDM model parameters.

Table 2.2 – Parameters of IDM for different vehicle types (Treiber and Kesting, 2013b)

Parameter's name	Car	Truck
Desired speed $V_{desired}$	120 km/h	80 km/h
Minimal Time headway T	1.5 s	1.7 s
Minimal gap s_0	2.0 m	2.0 m
Maximal Acceleration a	0.3 m/s2	0.3 m/s
Maximal Deceleration b	3.0 m/s2	2.0 m/s
Acceleration exponent δ	4	4

Table 2.3 – Parameters of IDM in different scenarios (Treiber and Kesting, 2013b)

Parameter's name	Highway	City traffic
Desired speed $V_{desired}$	120 km/h	54 km/h
Minimal Time headway T	1.0 s	1.0 s
Minimal gap s_0	2.0 m	2.0 m
Maximal or comfortable Acceleration a	1.0 m/s2	1.0 m/s
Maximal or comfortable Deceleration b	1.5 m/s2	1.5 m/s
Acceleration exponent δ	4	4

2.3.2 Lane change models

Lane change models determine lane change choices on multi-lane roads and speed adjustments related to lane changing. Considering lane changing behavior, the modeling efforts are not as many as in the CF behavior. This is due to the complexity of the LC behavior, which is affected by the surrounding vehicles and by the traffic flow environment (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007). Fig. 2.3 illustrates surrounding vehicles in LC models.

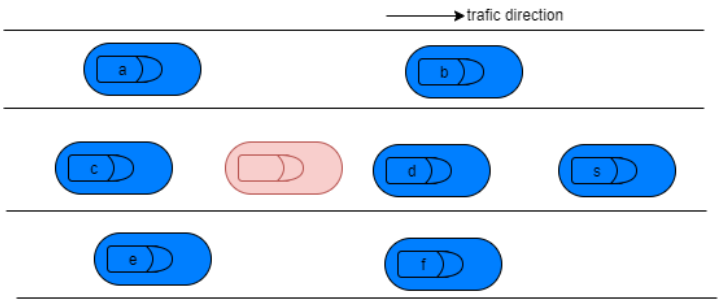


Figure 2.3 – The considered surrounding vehicles in the LC models

Lane change models are combined with car-following ones in order to have a complete microscopic traffic model. When the car-following model is parameterized to simulate aggressive drivers, the lane-changing driver profiles should be aggressive as well, without introducing new parameters. In some reviews of lane changing models (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007; Moridpour, Sarvi, and Rose, 2010; Z. Zheng, 2014), the authors propose

several rule-based lane changing models. More specifically, LC includes two components : the decision to consider a lane change and the decision to execute the lane change. Lane change decisions are often categorized into mandatory (MLC) or discretionary (DLC). MLC is performed when the driver must leave the current lane to reach the planned destination. DLC is performed in order to gain a speed advantage or a better driving environment. In (Z. Zheng, 2014), the authors mentioned that the major for LC models in the literature can be also categorized into two groups: 1) models that aim to capture the lane changing decision-making process, where the LC decision is taken only depending on the drivers motivation; 2) models that aim to quantify the impact of lane changing behavior on surrounding vehicles, where the LC decision is taken related to a predictive impact for the overall traffic agents. Furthermore, gap acceptance models are used to model the execution of lane changes and with the purpose of traffic safety. We are interested in this thesis in mostly DLC models that the decision of lane changing is motivated by a gain in speed, and in the safety of execution of lane changing, which is important. I summarize below an overview of classical LC models types; see 2.4.

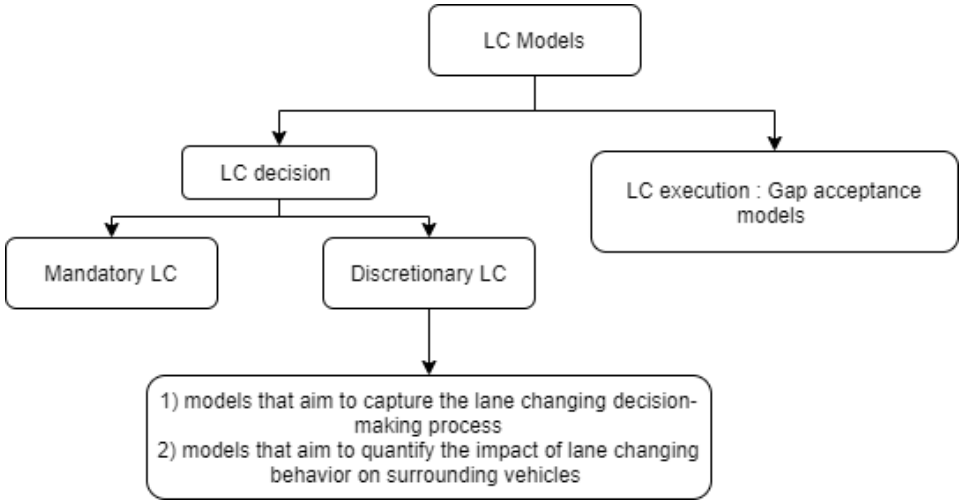


Figure 2.4 – An overview of LC models types

Gipps' LC model:

In the Gipps' LC model (Gipps, 1986), the decision of lane change is based on the following factors:

- whether it is physically possible and safe to change lanes without an unacceptable risk of collision,
- the location of permanent obstructions,
- the presence of special purpose lanes such as transit lanes,

- the driver's intended turning movement,
- the presence of heavy vehicles,
- the possibility of gaining in speed

In the Gipps' model, which is deterministic, a driver decides to maintain the desired speed or prepare for the LC maneuver, depending on the distance to the intended LC. Moreover, this model assumes that a lane changing takes place only when it is safe, and the gap is enough in the target lane. This model has a serious limitation in congested and incident-affected condition. In the model of Yousil and Hunt (Yousif and Hunt, 1995), a similar logic is used to model the decision of lane changing. Instead of a deterministic process of each rule, the model of Yousil and hunt introduces randomness, to make the model more realistic. More specifically, the probability for a driver to start a mandatory LC at a distance x_n from the downstream node (or incident, lane drop, etc.) is defined as follows :

$$p_n = \begin{cases} exp[(x_n - x_0)^2/\sigma_n^2] & \text{if } x_n > x_0 \\ 1 & \text{if } x_n \leqslant x_0 \end{cases} \tag{2.15}$$

MOBIL (Minimizing Overall Braking Induced by Lane Changes)

MOBIL model (Kesting, Treiber, and Helbing, 2007) is a discrete decision of lane change maneuver, which is used by combining with a longitudinal acceleration model. According to this model, a vehicle $n \in [1, N_v]$ decides to change lane when:

1. it is *safe* to cut-in:

$$\tilde{a}_n \geq -b_{\text{safe}};$$

2. there is an *incentive*, for the ego-vehicle and possibly its followers:

$$\underbrace{\tilde{a}_n - a_n}_{\text{the vehicle}} + p \left(\underbrace{\tilde{a}_f - a_f}_{\text{old follower}} + \underbrace{\tilde{a}_{f'} - a_{f'}}_{\text{new follower}} \right) \geq \Delta a_{\text{th}};$$

where f is the old follower in the current lane, f' is the new follower in the target lane. $\tilde{a}_x$ and a_x, $x \in \{n, f, f'\}$ are the future and current acceleration if the lane change takes place and the future acceleration $\tilde{a}_x$ is predicted by a longitudinal acceleration model (e.g. IDM). The parameter $p \in [0, 1]$ models the politeness. $p = 0$ means a very altruistic behavior, and $p \in [0, 0.5]$ is realistic behavior, while $p = 1$ means very egoistic. We notice that the safety criterion is not ignored: $p < 0$: very malicious that the driver does not respect safety criterion. δa_{th} is a threshold to respect an overall acceleration if lane change takes place. b_{safe} is the

maximum braking imposed to a vehicle for the lane changing. See Tab. 2.4 for the example of MOBIL model's parameters.

Table 2.4 – Parameters of MOBIL model (Kesting, Treiber, and Helbing, 2007; Treiber and Kesting, 2013b)

Parameter's name	value
Politeness factor p	[0,0.5]
Maximum safe deceleration b_{safe}	4m/s2
Threshold a_{th}	0.2 m/s2

By application of the right priority rule on most European highways, left lane should only be used for the purpose of overtaking, and passing in the right lane is forbidden unless the traffic is congested. In this case, the 'keep-right' directive of the lane-usage rule is implemented by a constant bias Δa_{bias} in addition to the threshold Δa_{th}. Thus, the MOBIL model is reformulate to :

- left to right : $\tilde{a}_n^{eur} - a_n + p(\tilde{a}_f - a_f) > \Delta a_{th} - \Delta a_{bias}$
- right to left : $a_n - \tilde{a}_n^{eur} + p(a_{f'} - \tilde{a}_{f'}) > \Delta a_{th} + \Delta a_{bias}$

where

$$\tilde{a}_n^{eur} = \begin{cases} min(a_n, \tilde{a}_n) & \text{if } v_n > v_{lead} > v_{crit} \\ a_n & \text{otherwise} \end{cases}$$

where $\tilde{a}_n^{eur}$ is the acceleration in the right lane, $\tilde{a}_n$ is the acceleration in the left lane, v_{lead} is the speed of the preceding vehcile in the left lane, v_n is the speed of lane changer in the right lane and v_{crit} is a critical speed below which the traffic is congested, e.g. $v_{crit} = 60km/h$. Δa_{bias} can be set to $0.3m/s^2$.

Cellular automata based LCD models

A typical CA (CA = $(\varepsilon, \Sigma, N, \delta)$ model includes four key components : the physical environment (ε), the cells' states (σ), the cells' neighborhoods (N), and local transition rules (δ), which uses integer variables to describe the dynamical properties of the system. In traffic CA models, the road is divided into cells of a certain length Δx, and the time is discretized into steps of Δt. Each road cell can be occupied by a vehicle or it can be empty. In (Barlovic et al., 1999), the author proposes a microscopic traffic model (car-following and lane changing) based on CA, in which the motion of a vehicle performs with respect to following rules:

- Collision-free acceleration : $v_i^{(1)=min(v_i, v_max, g_i)}$,
- Randomisation : with a certain probability p do $v_i^{(2)} = max(v_i^{(1)} - 1, 0)$
- Movement: $x_i = x_i + v_i^{(2)}$

Where g_i is the gap (the number of empty cells) with the preceding vehicle, x_i is the vehicle's position. If a driver wants to change lane, he has to take into consideration the gaps g_s and g_p (in Fig2.5) on the alternative lane in order to prevent crashes.

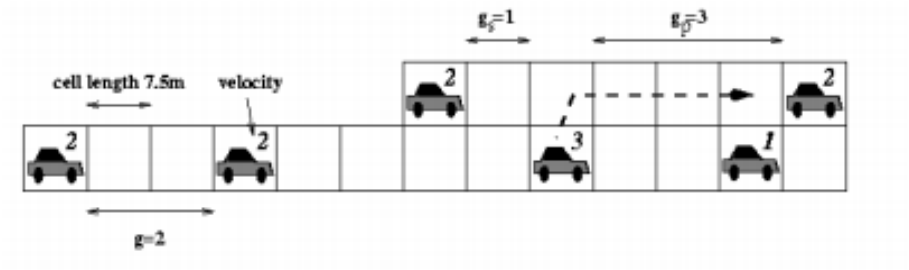


Figure 2.5 – An example of CA model

2.4 Modelling of driving behavior by artificial intelligence

Replicating vehicles' trajectories is one of the primary objectives of modeling vehicles' motion in traffic flow. We can find many classical models for CF and LC models. Although these traditional models have produced great achievements in microscopic traffic simulation, they have limitations :

- Limitation of trajectory accuracy. For example, most current car-following models are simplified, i.e., they contain only a small number of parameters (desired speed, minimal time headway, minimal safe distance...etc.). Simplification leads to suitable analytical properties and rapid simulations. However, it also renders the models limited in flexibility and accuracy, because using few parameters can hardly model the inherently complex driving process.

- Poor generalization capability. Classical models calibrated with empirical data try to emulate drivers' output (e.g., speed and spacing) by finding the model parameters that maximize the likelihood of the actions taken in the calibration dataset. As such, the calibrated car-following model cannot be generalized to traffic scenarios and drivers that were not represented in the calibration dataset.

- Absence of adaptive updating. For example, most parameters of car-following models are fixed to reflect the average driver's characteristics.

To address the above limitations, instead of handcraft manual models, for the driving behavior, machine learning provides an alternative approach based on data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

2.4.1 Deep Neural networks

During the last decades, deep learning (deep neural networks) methods, a group of specific machine learning methods aiming at building data based, haven shown excellent performances in many fields (e.g. natural language processing, image processing, finance trading strategy, etc.) In the field of microscopic road traffic flow modeling, researchers are interested in developing models which can describe and reproduce the vehicle motion in the real road traffic. The use of deep learning approaches for the microscopic modeling of road traffic flow is also becoming increasingly popular, as the physical models are limited in the face to such complex systems.

Though first proposed in the 50's (Rosenblatt, 1958), neural networks became utterly successful since only about a decade ago; thanks to i) the availability of huge amounts of data (mass data), ii) the amazing growth of the available computing power, and iii) the newly developed optimization algorithms (efficient variants of classical gradient algorithms). A neural network (NN) is an interconnected collection of artificial neurons. An artificial neuron transforms given input values into an output value by multiplying the inputs by weights and by running them through an activation function:

$$\hat{Y} = \phi(w \cdot X) \tag{2.16}$$

where $\hat{Y}$ is the output, ϕ is the activation function, w is the vector of weights, and $X = (x_1, x_2, \ldots, x_n)$ is the input vector with n input features. Examples of popular activation functions are the Sigmoid function, the Hyperbolic Tangent (tanh) and the Rectied Linear Unit (ReLU). Several Neurons can be arranged into one layer, then many layers can be arranged together and become a deep neural network. Formally, the output $\hat{Y}$ of a multi-layer neurons with an input X and with k layers, is given as follows :

$$\begin{cases} f_k(x) = \phi_k(w_k \cdot x) \\ \hat{Y} = f_k(f_{k-1}(..f_1(X))) \end{cases} \tag{2.17}$$

In order to learn the weights of a neural network in the context of supervised learning, a loss function must be defined, suitable to the problem to handle. This function should denote how close the predictions provided by the neural network are to the expected predictions called as actual labels (Y). After choosing the loss function, the goal is to globally minimize this error (loss function L). One of the existing method in the learning process to update the weights, is the Stochastic Gradient Descent (SGD).

In practice, to efficiently train a neural network, epochs and mini-batches are defined for the weights updating process. A mini-batch refers to a subset (with m individual samples) of the whole training set, to compute a partial gradient and update the weights accordingly. An

epoch refers to processing the whole training set once. A neural network training algorithm can be summarized as follows :

Algorithm 2.1: Neural network learning procedure (Stochastic Gradient Descent)

Data: input X, output Y,
Parameter :batch size = m, learning rate = α
for *number of epochs* **do**
  for *number of mini-batches* **do**
    - sample a batch x_m from the training data set
    - predict $\hat{y}_m$
    - compute the loss function $L_m = L(\hat{y}_m, y_m)$
    - update weights : $w_i \leftarrow w_i - \alpha \frac{1}{m} \sum_{j=1}^{m} \nabla L_j(w_i)$

Long short-Term Memory (LSTM) neural networks

Long short-Term Memory (LSTM) neural networks (Hochreiter and Schmidhuber, 1997) are a kind of recurrent neural networks (RNNs), which are able to learn time series with keeping a memory of previous sequences (Gers, Schmidhuber, and Cummins, 1999). The idea of RNNs is

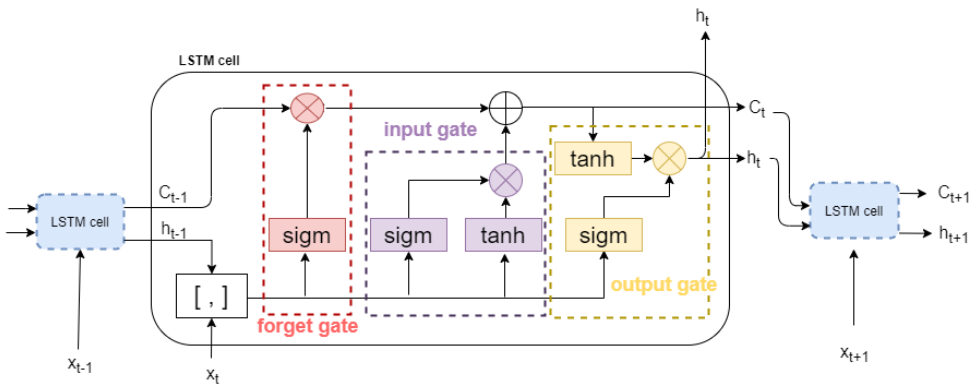


Figure 2.6 – LSTM neural network architecture (Gers, Schmidhuber, and Cummins, 1999). A LSTM cell consists of 3 gates (forget gate, input gate and output gate), where sigm denotes the sigmoid function, $tanh$ denotes the tangent function, $[,]$ denotes a concatenation.

to be able to connect previous information to the present task, where previous information has long-term effect on the present behavior. On Fig. 2.6 we have a LSTM neural network, where a sequence of LSTM cells are used to connect the input signal $(x_0, x_1, ...x_t, x_{t+1}, \ldots)$. A LSTM cell consists of 3 gates (forget gate, input gate, output gate), and each cell can give an output h_t (as shown in Fig. 2.6). The forget gate decides how much to "forget" from the previous cell

state c_{t-1}. The input gate determines the amount of new information to be stored in memory, based on new input x_t and the previous output information h_{t-1}. The output gate computes the new output by combining the previous cell state and the output of input gate. We give below the operation function of the 3 gates of a LSTM cell :

1. Forget Gate :
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
2. Input gate :
$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i])$$
$$\widetilde{c}_t = tanh(w_c[h_{t-1}, x_t] + b_c)$$
3. Output gate:
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o])$$
4. LSTM cell state :
$$c_t = f_t * c_{t-1} + i_t * \widetilde{c}_t$$
5. LSTM cell output:
$$h_t = o_t * tanh(c_t)$$

LSTM for driving modelling : Some Related works

Several studies have been performed on LSTM networks for human driving behavior modeling, and/or human driving forecasting. We cite (Xie et al., 2019), where authors developed a LC model (the LC decision prediction combining the LC execution trajectory) using neural networks. In (Xie et al., 2019), the execution of lane changing trajectory used LSTM neural network. In (Y. Zhang et al., 2018), the authors proposed a simultaneous CF and CL model using LSTM neural network, with a constraint of time headway condition to retrain the neural network when this condition is not satisfied. In (Altché and La Fortelle, 2017), the authors used a LSTM network for vehicle trajectory prediction from 1s to 10s in the future. All these works show that LSTM is essentially useful for the prediction of human driving behavior. However, different studies used different model structures, and they used also different input designs for the neural networks.

2.4.2 Reinforcement Learning (RL)

Reinforcement Learning (RL) is a mathematical approach for learning by interaction in a complex and uncertain environment, where an agent tries to maximize the total amount of a reward received from the environment. Different from the supervised learning method, RL guides

the agent to discover most rewarded actions by trial-and-error exploration in the environment, meanwhile agent's action affects the environment changes also. Many researchers have shown the great efficiency of using RL for car-driving modeling. Reinforcement learning(RL) is a general framework for decision-making under uncertainty. It provides an agent who intends to solve optimal control problems based on Markov Decision Processes(MDP).

Markov decision process

A Markov decision process (MDP) describes the sequential decision making process, denoted (S, A, P, R, γ), where S is the state space, A is the discrete action set, P denotes the transition probabilities,$R(S, A)$ is the rewards, and γ is a discount factor. The objective is to seek a policy $\pi(a|s)$ applied in every time step, that maximises the the future return (G_t), which is discounted by γ (the discount factor), with $0 \leq \gamma \leq 1$. G_t is defined as:

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k}, \tag{2.18}$$

where R_{t+k} is the reward given at time step $t + k$. The return value G_t should be a random variable along a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots)$ which indicates the accumulated rewards induced by the policy $\pi(a_t|s_t)$ and the system dynamics $P(s_{t+1}|s_t, a_t)$. The discount factor γ essentially determines how much the distant rewards impacts the agent's decision of each action. If $\gamma = 0$, the agent will be completely short-sighted and takes care only of the immediate reward produced by the action. If $\gamma = 1$, the agent will evaluate each action based on the total of all future rewards; see (Sutton and Barto, 2018) for a further comprehension of MDP and reinforcement learning.

State Value Function and Action-State Value function

The state value function, denoted $V^\pi(s)$, is defined as the expected return of the policy when starting with a given state s :

$$V^\pi(s) = E[G|s_0 = s] \tag{2.19}$$

The state-action value function, denoted $Q^\pi(s, a)$, is defined as the expected return when starting with a given state s and a given action a :

$$Q^\pi(s, a) = E[G|s_0 = s, a_0 = a] \tag{2.20}$$

State Value Function and Action-State Value function Optimality

As mentioned above, the objective of a reinforcement learning task is to obtain a policy which can achieve a great return for the agent. The policy with the greater expected return than all other policies consists in the optimal policy, denoted π^*. Therefore, we can obtain the optimal policy by maximizing either the state-value function or the the action-value function. We also define the optimal state-value function $V^*(s)$ and the optimal action-value function $Q^*(s,a)$ as follows :

$$\begin{aligned} &\forall s \in S, && V^\star(s) \triangleq Q^{\pi^\star}(s) = \max_\pi V^\pi(s); \\ &\forall (s,a) \in S \times A, && Q^\star(s,a) \triangleq Q^{\pi^\star}(s,a) = \max_\pi Q^\pi(s,a). \end{aligned}$$

Based on the Bellman principle of dynamic programming (see for example in (Sutton and Barto, 2018)), the Bellman equation on $V^*(s)$ is written as follows :

$$\begin{aligned} V^*(s) &= \max_a q_{\pi^*}(s,a) \\ &= \max_a \mathbb{E}_{\pi^*}[G_t | s_t = s, a_t = a] \\ &= \max_a \mathbb{E}_{\pi^*}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | s_t = s, a_t = a] \\ &= \max_a \mathbb{E}_{\pi^*}[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | s_t = s, a_t = a] \\ &= \max_a \mathbb{E}[R_{t+1} + \gamma V^*(s_{t+1}) | s_t = s, a_t = a] \end{aligned} \tag{2.21}$$

The Bellman equation on $Q^*(s,a)$ is written as follows :

$$Q^*(s,a) = \mathbb{E}[R_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a] \tag{2.22}$$

Model-free and Model-based approaches

Reinforcement learning algorithms can be grouped into two main families, called Model-free and Model-based. Model Based algorithms attempt to estimate $P(s_{t+1}|a_t, s_t)$ and $R(s_t, a_t)$ of the MDP, based on a history of transitions $D = \{(s_t, a_t, r_t, s_{t+1})\}$, using for example Maximum Likelihood Estimation, with a hypothesis for dynamics function :

$$\max_{\hat{P}} \prod_t \hat{P}(s_{t+1} \mid s_t, a_t) \tag{2.23}$$

and reward function :

$$\min_{\hat{R}} \sum_{t} \|R(s_t, a_t) - \hat{R}(s_t, a_t)\|^2 \tag{2.24}$$

Thus, the estimated MDP(S, A, P, R, γ) can be achieved and the optimal policy π^* can be computed by dynamic programming. In the other side, model-free algorithms aim at getting the optimal policy π^* directly, without knowing the $P(s_{t+1}|a_t, s_t)$ in the MDP.

Model-free methods

Value-based methods: Among the model-free algorithms, the Value-based methods optimizes the value function by evaluation and improvement. Then, from the optimal value function, the optimal policy is attained by selecting the action at a state with the largest optimal value.

Policy-based methods: Differently, in the Policy-based methods, the agent learns policies directly, where we define an objective function which could be the value function of the initial state s_0 for policy π_θ :

$$J(\theta) = V_{\pi_\theta}(s_0) \tag{2.25}$$

and aiming at finding : $\theta^* = \arg\max_\theta J(\theta) = \arg\max_\theta V_{\pi_\theta}(s_0)]$ It can be formulated by the gradient ascent :

$$\nabla J(\theta) = \nabla V_{\pi_\theta}(s_0)] \tag{2.26}$$

Actor-Critic: Actor-Critic methods are a combination of value and policy based methods consisting of 1) the "Critic" to estimate the value function and 2) the "Actor" to update the policy distribution in the direction suggested by the Critic (e.g. with policy gradients). In Figure 2.7, a simple Actor-Critic algorithm is shown :

We give in Table 2.5 some famous reinforcement learning algorithms :

1: **for** each step **do**
2: generate sample s, a, r, s', a' following π_θ
3: $\delta = r + \gamma Q_{\mathbf{w}}(s', a') - Q_{\mathbf{w}}(s, a)$ #TD error
4: $\mathbf{w} \leftarrow \mathbf{w} + \beta\delta\psi(s, a)$
5: $\theta \leftarrow \theta + \alpha\nabla_\theta \log \pi_\theta(s, a) Q_{\mathbf{w}}(s, a)$
6: **end for**

Figure 2.7 – Simple Actor-Critic algorithm (Zhou, 2020)

Algorithm	type
Q-learning (Watkins and Dayan, 1992)	Value-based
Sarsa (Sutton and Barto, 2018)	Value-based
Deep Double Q-learning (Van Hasselt, Guez, and Silver, 2016)	Value-based
REINFORCE (Sutton and Barto, 2018)	Policy-based.
A3C (Mnih et al., 2016)	Actor-Critic
DDPG (Lillicrap et al., 2015)	Actor-Critic
PPO (Schulman et al., 2017)	Actor-Critic

Table 2.5 – Summary of some famous reinforcement learning algorithms.

Reinforcement learning for lane-changing model: some related works

Reinforcement Learning (RL) gains increasingly great performance among other methods for modelling and automating driving behaviour. Especially for lane-changing decisions, for which previously existed models were less efficient at describing the complexity of the driver's behaviour, RL methods perform as good as rule-based models, or in some cases, even better. Some works in RL approach the decision learning problem using discrete state and/or action spaces, while other works treat use continuous spaces. Depending on this decision regarding the description of the state and actions spaces, different RL algorithms have been applied either to describe both longitudinal and lateral vehicle movement or only one of them.

In (L. Wang et al., 2019), the authors developed an ego-efficient lane-change strategy using Q-learning, and also tested the effects of the strategy on the network level through Aimsum traffic simulation platform. The possible actions considered were three included a left lane change, a right lane change, and no change of lane. The state consisted of nine variables which included the ego-vehicle's acceleration gain from a lane change, the acceleration gain of the surrounding vehicles as well as the mean acceleration difference between the middle lane and the other lanes, for a number of vehicles downstream the ego-vehicle, and its standard deviation. All variables were discretized corresponding to six predefined intervals. The reward is defined as the acceleration gain, that allows to optimize the traffic condition by maximizing the controlled vehicle speed. In the experiment, the simulated traffic environment consists of two kinds of vehicles (ego-vehicles, controlled by teh IDM model (Intelligent Driver Model) for car-following, and RL for lane changing. Normal vehicles are controlled by the IDM model for car-following and the Gipps model for lane changing. Numerical simulation showed an improvement in speed of the vehicle adopting the lane-changing strategy developed with RL. The impact on the network was also positive, as more vehicles adopted this strategy, but only while the percentage of these vehicles remained below 60%.

Another approach was followed by (P. Wang, H. Li, and Chan, 2019) where the authors used an actor-critic algorithm for the automation of lane changing decisions. Using the Deep Deterministic Policy Gradient (DDPG), they extended the problem to a continuous action space considering the yaw acceleration as the lateral control variable. The state representation included the ego-vehicle's speed, position, orientation and acceleration, the speed and position of vehicles in the target lane, the road geometry, etc. The reward function took into account the lateral dynamics, punishing large yaw rates and yaw acceleration, the maneuvering time, and the position deviation. The results from numerical simulation showed that, after sufficient training, the agent learned efficiently how to make a smooth lane-changing maneuver with a success rate of 100%.

(Ye et al., 2020) also used an actor-critic algorithm to propose a method for automating lane-changing decisions. They proposed a model for mandatory lane-changes developed using the

Proximal Policy Optimization algorithm. In this work as well the state space was continuous, but the action space was discrete. The state representation included the ego-vehicle's longitudinal speed, position and acceleration, the lateral position and speed, as well as the relative distance of the surrounding vehicles, the longitudinal speed, acceleration and lateral position. Four surrounding vehicles were taken into account : the leading and following vehicle in the current lane and the leading and following vehicle in the target lane. The action space was designed in both lateral and longitudinal direction, and included 6 possible actions. The action was a combination of deciding whether to keep or change a lane, as well as abort a lane-changing maneuver, and selecting which leading vehicle to follow longitudinally. The reward function was designed so as to incorporate comfort, safety and efficiency of the lane change. It included a part for evaluation of the lateral and longitudinal jerk, another part for evaluation of the travel time and relative distance to the target lane, and a part for the evaluation of the collision risk. The numerical simulation results showed that the agent could learn successfully the lane changing strategy with a success rate of 95% and a collision rate of 2%, which outperformed the trained rule-based agent used as a reference.

With the aim to study both longitudinal and lateral movements, (Hoel, Wolff, and Laine, 2018) proposed a model for the automation of both speed and lane-changing decisions based on Deep Q-Learning. Using this algorithm, which aims to approximate the optimal Q-function with a deep neural network, they were able to treat the state space as continuous in contrast with simple Q-learning. In this work the state consisted of 27 variables which represented the ego-vehicle's normalised speed, the existence of lanes beside it, the normalised relative speed and position of the surrounding vehicles, and a value between (-1) and 1 depending on the lane of each surrounding vehicle with respect to the ego-vehicle. The action space was discrete but included also decisions for acceleration or deceleration along with keeping or changing the lane. In (Hoel, Wolff, and Laine, 2018), the reward was positive and based on the normalized driven distance during a time step, while also penalising with (-10) in the condition of collision or driving out of the road. A small negative reward of (-1) was also assigned to lane-changes in order to limit their number. Two agents were implemented: an agent which developed with DQN a strategy only for lane-changing, and used the IDM model for car-following, and an agent trained to control both speed and lane-changing. Using the IDM and MOBIL models as a reference, the results showed similar performance with the reference model for the first, but better performance for the second, indicating that maybe longitudinal and lateral decisions should not be studied separately. Finally, the application to both a highway driving case and an overtaking case with oncoming traffic illustrated the generality of their method.

In Table 3.2, I give out the overview of these related works using reinforcement learning based algorithms for the driving modelling, which I presented above. I indicate how state, action and reward variables have been defined.

Source	State	Action	Reward
(L. Wang et al., 2019)	ego-vehicle's acceleration gain, surrounding vehicles acceleration gain, mean acceleration difference between the middle lane and the other lanes for 10 vehicles downstream and standard deviation	left lane-change, right lane-change, lane keeping	Acceleration gain
(Hoel, Wolff, and Laine, 2018)	ego-vehicle's normalised speed, existence of lanes beside it, normalised relative speed and position of the surrounding vehicles, a value between -1 and 1 depending on the lane of each surrounding vehicle with respect to the ego-vehicle	left lane-change, right lane-change, lane keeping	normalized driven distance during a time step, -10 for collision, -10 out of the road, -1 for lane-change
(P. Wang, H. Li, and Chan, 2019)	ego-vehicle's speed, position, orientation and acceleration, target lane vehicles' speed and position, road geometry	yaw acceleration	a function punishing large yaw rates, yaw acceleration, maneuvering time, and position deviation
(Ye et al., 2020)	ego-vehicle's longitudinal speed, position and acceleration, lateral position and speed, surrounding vehicles' relative distance, longitudinal speed, acceleration and lateral position. (surrounding vehicles: leading and following vehicle in the current lane and in the target lane.	a combination of deciding whether to keep or change a lane, as well as abort a lane-changing maneuver, and selecting which leading vehicle to follow longitudinally	a function for evaluation of the lateral and longitudinal jerk, the travel time and relative distance to the target lane, as well as the collision risk.

Table 2.6 – Overview of environment description in reinforcement learning based algorithms for the driving modeling.

2.5 Traffic safety and Accidents

Another topic of interest in this thesis is the car-collision occurrence modeling. We are interested in generating car-collisions by numerical simulation of the road traffic, in order to evaluate how the autonomous vehicle can deal with a traffic with car-collisions and accidents. In this field, a large body of literature focuses on the traffic safety, especially exploring the link between ambient traffic conditions and collision occurrences (Shi et al., 2018) or crash injury severity (Savolainen et al., 2011). On highways specifically, the authors of (Salem, Farhi, and Lebacque, 2013) provide evidence that crashes have a specific relationship with traffic flow characteristics (volume, density, speed). In (Christoforou, Cohen, and Karlaftis, 2011), the authors identified the impact of general geometric parameters, weather, and traffic flow on different types of crashes (rear-end, sideswipe, multiple vehicle involved) based on a French traffic and crash data-set. Other studies employed statistical and machine learning approaches, such as Multivariate Probit models (Christoforou, Cohen, and Karlaftis, 2011) or Support Vector Machine (Dong, Huang, and L. Zheng, 2015), to investigate the relationship between influencing traffic factors and crash occurrences. Another approach consists in using traffic simulation for crash modeling. In (Azevedo, Cardoso, and M. E. Ben-Akiva, 2018), the authors proposed models for three types of crashes depending on vehicle interactions and maneuvers, and estimated the parameters of the models via simulation. In (J. Wang et al., 2017), the authors studied the impact of driving violations (driving over speed limit, slow driving, and abrupt hard braking) on crash occurrence, for different traffic conditions through simulation.

2.6 Road traffic simulators

There exists a wide variety of traffic simulators based on microscopic models, multi-agent or cellular automata. However, most of the simulators are for a business purpose, and therefore provide very few design details, such as Aimsun (Casas et al., 2010a), PTV Vissim (Fellendorf and Vortisch, 2010a), Paramics (Cameron and Duncan, 1996). Some open source alternatives exist also, such as SUMO (Lopez et al., 2018), MATSim (Horni, Nagel, and Axhausen, 2016) and MITSIMLab (M. Ben-Akiva et al., 2010). In a recent publication (Luo et al., 2020), the authors present their simulator SUMMIT and compare it with some other simulators in the context of testing autonomous vehicles in terms of realworld maps integration, possibility of simulating unregulated behaviors, possibility of simulating dense traffic, and realistic visual & sensors; see Table2.7 below.

In this thesis, we used principally SUMO traffic simulator, and we realised some works also using SCANeR studio driving simulator, which is a Renault internal simulator for the autonomous vehicle.

Simulator	real-world maps	unregulated behaviors	Dense traffic	realistic visual & sensors
SimMobilityST	✓	x	✓	x
SUMO	✓	x	✓	x
TORCS	x	✓	x	✓
Apollo	x	x	x	✓
Sim4CV	x	x	x	✓
GTAV	x	x	x	✓
CARLA	x	x	x	✓
Autono ViSim	x	✓	✓	✓
Forced-based simulator	x	x	✓	✓
SUMMIT	✓	✓	✓	✓

Table 2.7 – Comparison between SUMMIT and existing driving simulators.(Luo et al., 2020)

2.6.1 SUMO traffic simulator

"Simulation of Urban MObility", or "SUMO", is an open source, microscopic, multi-modal traffic simulator (Lopez et al., 2018), including modeling of road vehicles, public transport and pedestrians. It is released since 2001 and is available for many state-of-the-art microscopic traffic models. SUMO provides traffic simulations, including synthetic road network creation, real road network import, traffic and emission computation, and simulation visualisation. SUMO allows to generate a large number of different measures. The outputs are collected into files or a socket connection following the common rules for writing files. In addition, SUMO can be enhanced with custom models, and it provides various APIs to remotely control the simulation.

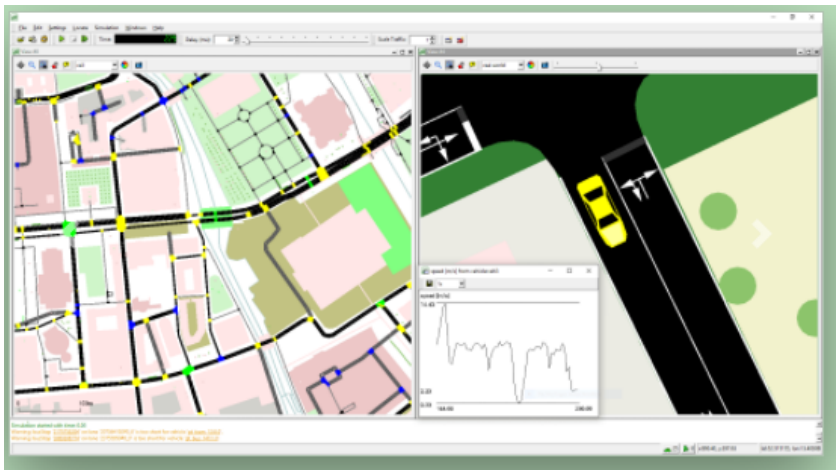


Figure 2.8 – SUMO screenshot(SUMO, 2021)

In principal, in SUMO, the microscopic driving dynamics of road vehicles are determined by the interplay of several models briefly listed below(Erdmann, 2014):

- Car-following model: determines the speed of a vehicle in relation to the vehicle ahead of it.
- Intersection model: determines the behavior of vehicles at different types of intersections in regard to right-of-way rules, gap acceptance and avoiding junction blockage.
- Lane-changing model: determines lane choice on multi-lane roads and speed adjustments related to lane changing.

2.6.2 SCANeR studio driving simulator

Renault has been developing driving simulators since several years. SCANeR studio is the driving simulator software which includes vehicle dynamics, traffic generation, visual and kinaesthetic modules for driving simulators. SCANeR studio provides a default traffic vehicle model which controls the traffic vehicle's movement in real time (Champion et al., 1999). The architecture of SCANeR studio is shown in Figure 2.9.

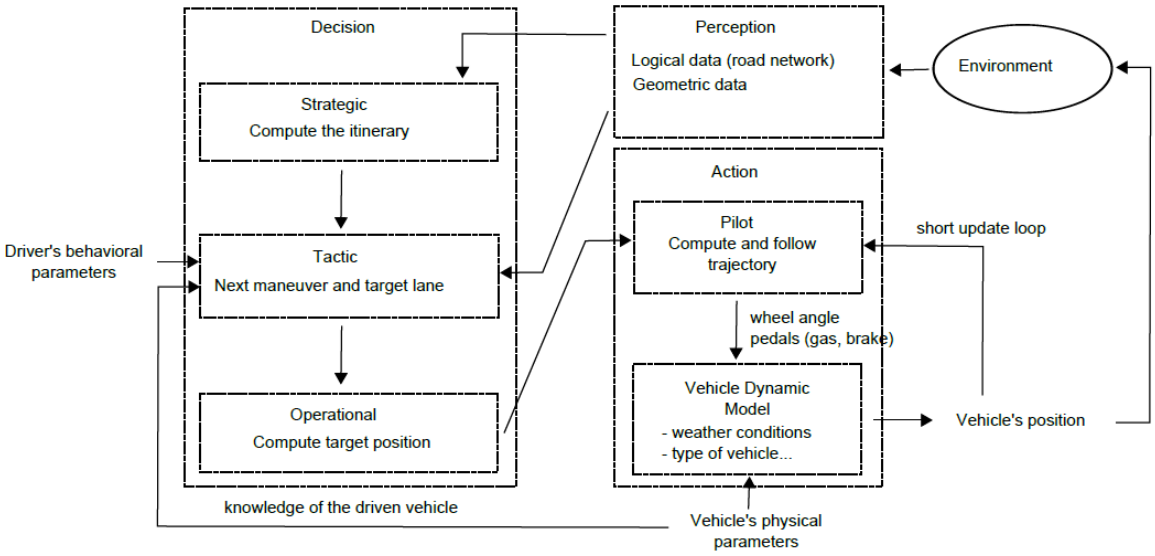


Figure 2.9 – Architecture of the SCANeR studio driving simulator.

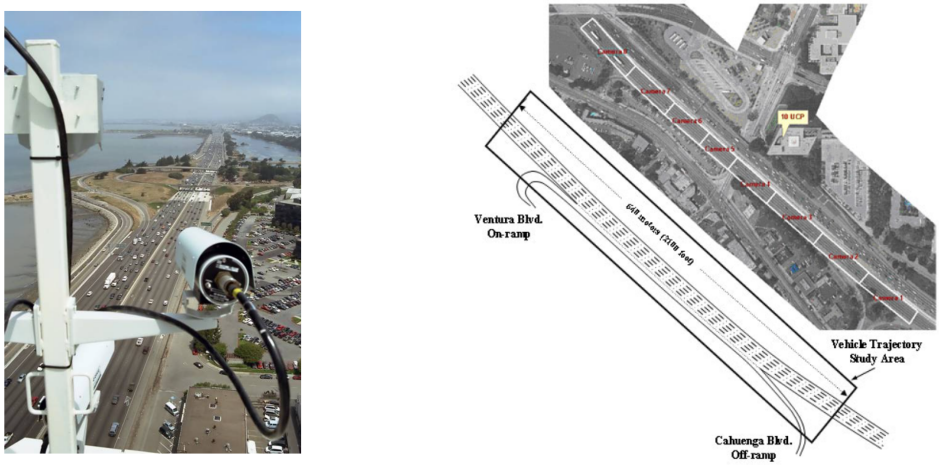
2.7 Road traffic data-sets

Regarding to traffic data collection, the traffic can be directly observed by cameras on top of a tall building or by cameras fixed in a drone. Then, using some object-tracking algorithms, the

trajectories, i.e. the positions of each vehicle over time, can be extracted from the video. By this process, vehicles within a given road section are captured in this way, a microscopic traffic trajectory data-set is achieved. These kinds of data-set can be used for studying both on the human driving behavior and the traffic flow. In this thesis, we have used two traffic trajectory data-sets : NGSIM (Alexiadis et al., 2004) and HighD (Krajewski et al., 2018). The work in the Chapter 3 and in the Chapter 4 are based on NGSIM dataset. The work in the Chapter 5 is first based on NGSIM dataset, then we have validated the model using HighD dataset. In addition, the work in the Chapter 6 is on NGSIM data-set and HighD data-set to understand and compare the sensitivity of each proposed models. I give in this section some descriptions of the NGSIM and the HighD data-sets.

2.7.1 NGSIM data-set

NGSIM 101 (Alexiadis et al., 2004) is an open data-set released by the US Federal Highway Administration (FHWA). The dataset acquisition camera and the road section map is shown in Figure 2.10. In NGSIM 101 dataset, all vehicle trajectories are provided with a rate of 10Hz,



(**a**) NGSIM traffic camera installation

(**b**) NGSIM 101 road section

Figure 2.10 – NGSIM 101 dataset acquisition camera and road section map

during the rush hour from 7:50 a.m.to 8:35 a.m, on a highway section in Los Angeles, California, covering 640 meters of length. This section of road has 5 normal lanes and one auxiliary lane connecting an on-ramp and an off-ramp. NGSIM 101 dataset is divided into three files of sub-data-sets and each sub-data-set contains 15 minutes traffic data, which cover the traffic from 7:50 a.m. to 8:05 a.m., from 8:05 a.m. to 8:15 a.m., and from 8:20 a.m. to 8:35 a.m. In total, more than 6000 individual drivers real-time trajectories are recorded, which represent a total of 800 hours vehicle-trajectories in various traffic densities. Our studies focus mainly on the

traffic in the 5 main lanes with the assumption that the driver behavior in the auxiliary lane is particular and different from the driver behavior in the main straight lanes.

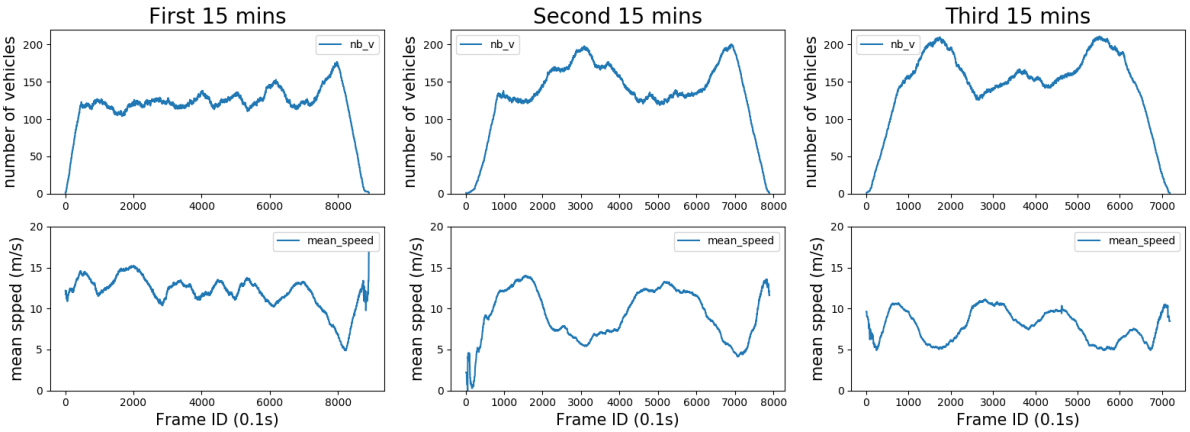


Figure 2.11 – NGSIM 101 dataset traffic flow (number of vehicles in traffic and traffic mean speed for each time frame) for three 15-minute periods respectively : from 7:50 a.m. to 8:05 a.m., from 8:05 a.m. to 8:15 a.m., and from 8:20 a.m. to 8:35 a.m..

In Figure 2.11, we present traffic volumes and speeds during each of the three time periods. It can be observed that, in the beginning of the first 15 minutes, the mean speed is between 10 and 15 m/s. After that, traffic becomes denser resulting to a sudden fall of the mean speed. In the second and third 15-minute time periods, traffic is more congested and mean speed varies between 5 and 10 m/s.

2.7.2 HighD data-set

Similarly to the NGSIM dataset, the HighD dataset (Krajewski et al., 2018) is a dataset of vehicle trajectories recorded on German highways. The vehicular traffic was recorded at six different locations and includes more than 110 500 vehicles using the camera from a drone(see. Figure 2.12). Each vehicle's trajectory, including vehicle type, size and manoeuvres, is automatically extracted. Using state-of-the-art computer vision algorithms, the positioning error is typically less than ten centimeters. Although the dataset was created for the safety validation of highly automated vehicles, it is also suitable for many other tasks such as the analysis of traffic patterns or the parameterization of driver models.The HighD dataset provides totally 60th sub-datasets with different road sections and different time periods. Differently from the NGSIM dataset, the HighD dataset has a higher resolution and its data frequency is 25Hz, while the data frequency of the NGSIM dataset is 10Hz.

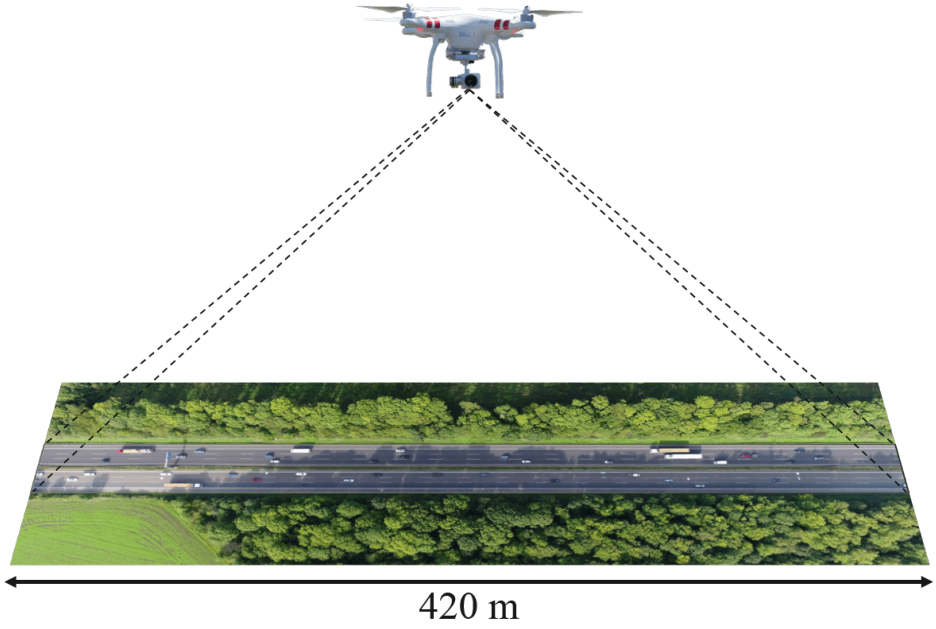


Figure 2.12 – HighD dataset recording setup includes a drone that hovers next to German highways and captures traffic from a bird's eye view on a road section with a length of about 420 m (Krajewski et al., 2018)

Chapter 3

Analysis of driver behavior and inter-vehicular collision : a data-based traffic modeling and simulation approach

The emergence of Intelligent Connected Vehicles (ICVs) is expected to contribute to resolving traffic congestion and safety problems. However, it is inevitable that ICV safety issues in mixed traffic (involving ICVs and human driven vehicles) will be a critical challenge. The numerical simulation of scenarios involving a mix different driving profiles is expected to be an important safety assessment tool to use in the process of testing and validating ICVs. Numerical simulation should be specifically important regarding extreme scenarios, including car collisions, which are rarely captured in real-world data-sets and can be generated numerically. In this work, we propose a novel approach for car-collision generation in numerical simulations based on the assumption that car-collision occurrences are mostly associated with certain specific driver profiles. Using the NGSIM 101 data-set, we identify three different driver profiles: aggressive, inattentive and normal drivers. We then replicate car-collision occurrences by varying the percentages of these three driver profiles in the simulated environment, allowing us to establish a relationship between driver profiles and car-collision occurrences. We also investigate the severity of generated car-collisions and classify them with respect to the driver profiles of the cars involved in the collisions. Our approach of replicating car-collision occurrences in numerical simulations will facilitate the testing and validation of ICVs in the future, especially regarding the testing of ICV functionalities in dealing with traffic accidents.

Contents

3.1 Motivation

With the development of the information and the telecommunication technologies, along with the rapid growth of New Energy Vehicles (NEVs), Intelligent Connected Vehicles (ICV) have become an increasingly active research topic. ICVs are expected to reshape future mobility and contribute to mitigating road traffic congestion and safety problems (Fagnant and Kockelman, 2015).

One of the major challenges of ICVs is communicating with other vehicles and accurately recognizing the patterns of human driving behavior in mixed traffic (Schwarting, Alonso-Mora, and Rus, 2018). As we know, human driving behavior is not perfect and million deaths occur each year due to traffic accidents, among that, 90% of accidents are produced because of driver's mistake and inattention(Singh, 2015). Therefore, the process of testing ICVs in dealing with traffic accident is very important. Moreover, the validation and the testing of ICVs in mixed traffic consisting of ICVs and human driven vehicles, are necessary to ensure the security of ICVs functionalities. However, performing physical tests of ICVs is not only time-consuming, but it is sometimes dangerous as well, especially in the traffic with accidents. The numerical traffic simulation is an alternative tool to test and validation autonomous vehicle. One can generate easily massive scenarios even dangerous ones with vehicular collisions, which happen rarely in the real life. It is important to understand human driver behaviors in order to observe the collision related driver profiles. However, the relationship between different driver profiles and the car-collision occurrences has not been understood clearly. Meanwhile the procedure of car-collisions generation in traffic numerical simulators is not fully integrated. We propose in this chapter an approach for generating vehicular-collision in numerical traffic simulation with the purpose of facilitating the testing and validation of ICVs in the future.

3.2 Background

It is well known that traffic numerical simulation is established by using mathematical models to reconstruct road traffic (Chao et al., 2019), and can be performed on many scales (i.e. microscopic, mesoscopic, macroscopic, etc.). In particular, for the validation of the ICVs behavioral decision system, microscopic traffic simulators are particularly interesting, as they focus on replicating the vehicles' individual behaviors in details, in order to create an entire traffic environment. Various microscopic traffic simulators have been developed; we cite SUMO (Krajzewicz et al., 2012), VISSIM (Fellendorf and Vortisch, 2010b), and AIMSUN (Casas et al., 2010b). However, the vehicle collision generation mechanism has not yet been described completely in any of those simulators.

As well known, drivers have different driving patterns. Therefore, several driver profiles can be distinguished and observed from real driven data, e.g. real vehicle trajectories. Next Generation simulation (NGSIM) published four vehicle trajectory data-sets (NGSIM (Alexiadis et al., 2004)), which have been widely used by many researchers in transportation (He, 2017). More recently, data-sets on motion trajectory of traffic objects detected by autonomous vehicle equipment were published (e.g. Waymo data (Waymo, 2019), Argoverse (argoverse, 2019), nuScenes (nuscenes, 2019), etc.). These data-sets can be used to observe the single human driver behavior (Marina Martinez et al., 2018). However the whole traffic situation is not captured in these data-sets. The driving profile identification is mostly based on clustering approaches with real driven data (Júnior et al., 2017). Many open data-sets on vehicular traffic are nowadays available and could be applied for studying and analyzing human driving behavior, but driving profile patterns are not given directly in the data-sets. The authors of (Meiring and Myburgh, 2015) reviewed various methods of classification of driving styles: 'normal', 'aggressive', 'inattentive' and so on. Aggressive drivers are often characterized by risky speeding (such as abrupt instantaneous speed change, over limit speed, excessive acceleration or deceleration) and frequent lane changing behavior (Aljaafreh, Alshabatat, and Al-Din, 2012). Inattentive drivers can be characterized by long reaction time and forced sudden lane change with an instantaneous deviation from normal behavior (Meiring and Myburgh, 2015). Some studies used clustering approaches (K.-T. Chen and H.-Y. W. Chen, 2019) to classify the different drivers profiles, but this kind of methods have the uncertainty in the result of classification, the algorithm complexity, and the need of specific data-sets, which makes the method hard to generalize.

In the other hand, vehicle collisions are rarely observed. In (Abdulhafedh, 2017), several vehicle collision related data-sets have been summarized. However, most of these data-sets provide only vehicle collision statistics, without involving car-trajectories. Nevertheless, the Strategic Highway Research Program 2-Naturalistic Driving Study (SHRP2-NDS, 2016) provides a data-set where several accident situations are captured, yet in terms of quantification of the vehicle collision causes and impacts, it seems to be unsatisfactory. Therefore, a captured vehicle collision study would be an untouched research area.

As what we can find, a large body of literature on road traffic safety explores the link between ambient traffic conditions and collision occurrences (Shi et al., 2018) and/or vehicle-collision injury severity (Savolainen et al., 2011). On highway specifically, the authors of (Salem, Farhi, and Lebacque, 2013) provide evidence that vehicle collisions have a specific relationship with traffic flow characteristics (volume, density, speed). In (Christoforou, Cohen, and Karlaftis, 2011), the authors identified the impact of general geometric parameters, weather, and traffic flow on different types of collision (rear-end, sideswipe, multiple vehicle involved) based on French traffic and vehicular statistic data. Some other studies employed statistical and machine learning approaches, such as Multivariate Probit models (Christoforou, Cohen, and

Karlaftis, 2011) or Support Vector Machine (Dong, Huang, and L. Zheng, 2015) to investigate the relationship between influencing traffic factors and vehicle collision occurrences. Several previous works consist in using traffic simulation for vehicle collision modeling. In (Azevedo, Cardoso, and M. E. Ben-Akiva, 2018), the authors proposed models for three types of vehicle collision depending on vehicle interactions and maneuvers, and estimated the parameters of the models via simulation. In (J. Wang et al., 2017), the authors studied the impact of driving violations (driving over speed limit, slow driving, and abrupt hard braking) on car-collision occurrence for different traffic conditions through simulation. Nevertheless most of the literature treats the road traffic safety, without considering driving profiles.

In this work, we propose a novel approach for car-collision (collision between cars) generation in numerical simulation by varying the percentages of different driver profiles in the traffic, aiming at establishing a relationship between driving profiles and car-collision occurrences. The profiles are extracted from the NGSIM database and integrated in the traffic simulator SUMO (Lopez et al., 2018), using the calibrated Intelligent Driver Model (IDM), which is one of the most human-like car-following model (Vasconcelos et al., 2014; M. Zhu, Xuesong Wang, Tarko, et al., 2018).

Our purposes in this work are: 1) Extracting driver profiles from a real driven data, NGSIM 101 data-set; 2) Replicating different driver profiles using microscopic traffic model after a calibration; 3) Establishing a method of reproducing realistic traffic simulation based on microscopic road traffic modelling in SUMO simulator; 4) Generating car-collisions by appropriately varying the percentages of the driver profiles; 5) Characterizing the relationship between the car-collision occurrences with the driver profiles; and 6) Observing the severity of the generated car-collisions.

3.3 Materials and Method

In this section, the concerned materials and the proposed method will be presented. The approach structure is illustrated in Figure 3.1. We study the driver profiles from car-trajectories and we propose the definition of an 'aggressive' driver profile as the one who leaves always short time-headways with respect to the leader vehicle, and an 'inattentive' driver profile as the one with particularly long reaction time, compared to the other drivers. We classify all the drivers with intermediate values of reaction time and time-headway at the 'normal' driver profile. The thresholds for the three driver profiles are derived from the NGSIM 101 data-set (Alexiadis et al., 2004). After the classification of the driver profiles, the specific driver profiles (aggressive and inattentive) are represented using calibrated IDM model (the IDM model with an extension of reaction time), and the road traffic is then simulated using SUMO ("Simulation of Urban MObility") (Lopez et al., 2018). The IDM model calibration is

performed by using a genetic algorithm, to find the optimal set of IDM parameters with an objective minimize the predefined error between real driver trajectory and the output of the IDM model. In essence, we artificially increase the percentage of drivers with 'extreme' profiles (aggressive or inattentive) and, then, count and analyse the car-collisions generated by traffic simulation. In the numerical simulation experiments, we proposed 4 different experiments.

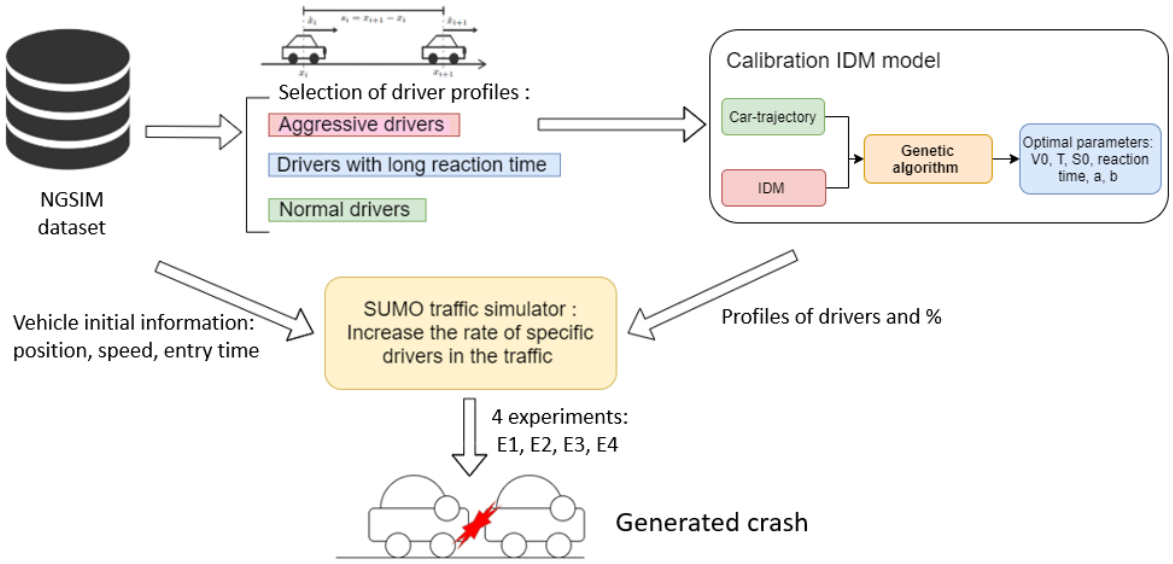


Figure 3.1 – Steps of the approach

The remaining of the presentation of this work is structured as follows: In Section 3.4, we present the classification of driver profiles from the NGSIM 101 dataset, and this part includes 1) the presentation of the NGSIM 101 dataset 2) the proposition of different driver profiles, and 3) the profiles extracted from NGSIM 101 dataset. In Section 3.5, we present SUMO traffic simulator, the choice of car-following model, IDM model, and the calibration of the IDM model for aggressive driver profiles and inattentive driver profiles, as well as for the normal driver profiles. Section 3.7 shows in details for the 4 simulation experiments for the generation of vehicular collisions. Section 3.8 presents the results obtained from numerical simulations and the investigation of the relationship between generated car collisions and driver profiles. Section 3.9 presents the validation of the approach with a different database. Section 3.10 deals with the severity of simulated car-collisions with respect to different driving profiles. Finally, Section 3.11 summarizes major findings and provides recommendations for further research.

3.4 Classification of driver profiles from the NGSIM 101 dataset

A driver profile can be defined as the average driving behavior of a given driving class (Itkonen et al., 2017). Driving behavior is related to driver baseline skills, socio-demographics (age,

gender, occupation... etc.) and current state (fatigue, distraction... etc.). From driven data, driver's characteristic can be illustrated by the speed, the acceleration, the jerk, and some other relevant traffic indicators, such as time-headway (THW) and time to collision (TTC) (Itkonen et al., 2017). THW and TTC are critical safety indicators for car-following behavior (Suweda, 2016). THW measures the time passed for reaching the leader's position while running at current speed, while TTC is usually used for judging the moment to start braking, and in the control of braking (Ayres et al., 2001). Several studies show that the distribution of THW is related to driving speed and also to traffic flow conditions. According to (WINSUM and Heino, 1996), a negative correlation between the car-speed and THW can exist. In (Maurya et al., 2016), it was shown that the speed and THW follow different distribution patterns under different traffic density levels. In this work, the classification of different driver profiles begins with the observations from the NGSIM data.

3.4.1 NGSIM 101 dataset description

A brief description of the NGSIM 101 data-set has been given in section 2.7.1. In this data-set, it provides totally 3 subsets of 15-minute traffic registration. In this work, we applied the proposed approach based on the first 15-minute data-set. The second 15-minute data-set is then used for the method validation. We note that we are interested here in understanding driving profiles in terms of the car-following behavior, without lane change or other behaviors. After a data processing of the first 15-minute part, we selected approximately 1500 car trajectories (drivers) from a total population of 1993 vehicle trajectories, to extract specific driver profiles. The selection was based on the following condition: each car must always have a leading car (downstream) whose trajectory is continuous at least over 40s. In addition, the same data processing was made for the second 15-minutes of data and resulted to the selection of 1300 car trajectories out of a total of 1495.

3.4.2 Proposition of driver profiles classification

In this work, the classification of different driver profiles begins with observing the vehicle time headway (THW) from the NGSIM data. Figure 3.2 shows the distribution of each car driver's maximum THW, mean THW, minimum THW, and standard deviation of THW, for the 1993 vehicle trajectories of the first 15 min NGSIM 101 data-set. Figure 3.2 provides evidence on the heterogeneity of human driving profiles as the mean THW ranges from near 0s to 5s and the minimum of THW ranges from near 0s to more than 3s. Based on this preliminary findings, we propose the definition of three profiles: (i) 'aggressive': shorter car time-headway, (ii) 'inattentive': longer reaction time, and (iii) 'normal' for intermediate values of reaction time and car time-headway. We notice that since the reaction time cannot be detected in the real driven data, we estimate and suppose that the reaction time is approximate to the minimum

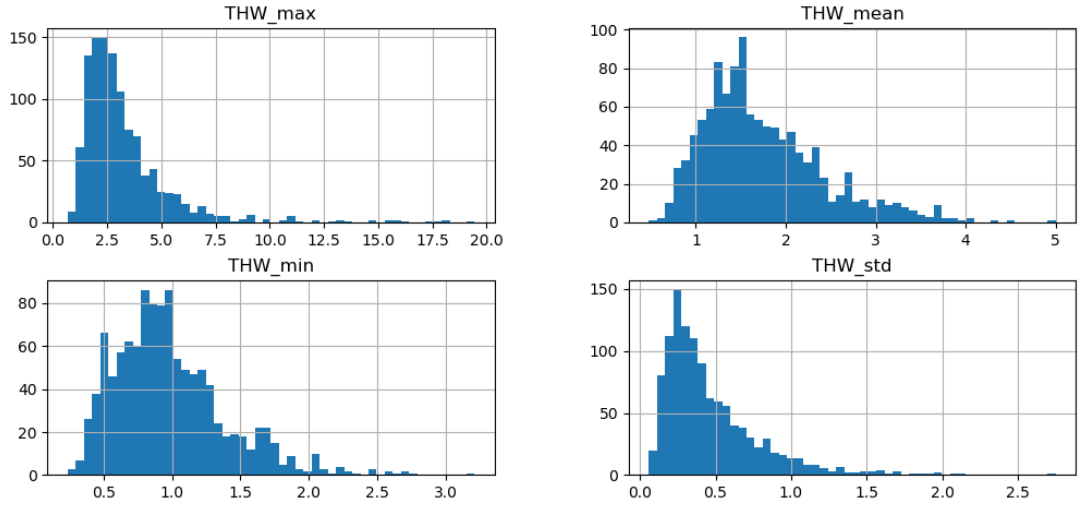


Figure 3.2 – The distribution of the maximum THW, the mean THW, the minimum THW and the standard deviation of THW the first 15 minutes data of NGSIM 101 dataset. The unit on the x-axis is seconds.

THW value, which is the minimum safe time gap that the driver estimate with the leading vehicle during his whole trajectory (longer than 400 time steps, explained in section 4.3.4). The definitions of the the two driver profiles (aggressive and inattentive) are formalized below.

1. Aggressive driver profile: A driver i is considered to be aggressive with respect to a threshold t^* on the time headway THW if

$$\overline{\mathrm{THW}}(i) := \frac{1}{T}\sum_{t}^{T} \mathrm{THW(i,t)} < \mathrm{t}^* \tag{3.1}$$

   We will explain in section 3.4.3 how we fixed the threshold t^* in our approach.

2. Inattentive driver profile (drivers with long reaction time): A driver i is considered to be inattentive (with a long reaction-time) with respect to a threshold $\widetilde{t}$ on the time headway THW if

$$\min \mathrm{THW}(i) := \min_{t} THW(i,t) > \widetilde{t} \tag{3.2}$$

   We will explain in section 3.4.3 how we fixed the threshold $\widetilde{t}$ in our approach.

3. Normal driver profile : the drivers whose profiles are not aggressive or inattentive are called *normal*. They have intermediate values of reaction time and time headway.

3.4.3 Specific driver profiles in the NGSIM data-set

In a traffic data-set with a large number of trajectories, it is always possible to distinguish 'good' drivers from 'risky' ones. Following the definition of the two 'extreme' driver profiles of section 3.4.2, we consider the following four driver profile groups in the NGSIM dataset: group 1: the 2.5% most aggressive drivers, group 2: the second 2.5% most aggressive drivers, i.e. situated between the 2.5 and 5-percentile of mean THWs, group 3: the 2.5% most inattentive drivers, and group 4: the second 2.5% most inattentive drivers (drivers ranked between 2.5% and 5% in terms of inattentiveness). The four groups are defined as follows:
Let us denote the set $\mathcal{I}_{agg}(t)$ of the drivers i whose $\overline{THW}(i)$ is less than t:

$$\mathcal{I}_{agg}(t) := \{i, \overline{THW}(i) < t\}. \tag{3.3}$$

Now, we denote by N the total number of drivers of the considered data-set, and by $N_{agg}(t)$ the cardinal (number of elements) of $\mathcal{I}_{agg}(t)$:

$$N_{agg}(t) := \text{Card}(\mathcal{I}_{agg}(t)). \tag{3.4}$$

Finally, we denote by $p_{agg}(t)$ the proportion of the number of drivers in $\mathcal{I}_{agg}(t)$ with respect to the total number of drivers (N):

$$p_{agg}(t) := N_{agg}(t)/N. \tag{3.5}$$

Let us now define t_1^* and t_2^* as follows:

$$t_1^* = p_{agg}^{-1}(0.025) := \max\{t, p_{agg}(t) \leq 0.025\}, \tag{3.6}$$
$$t_2^* = p_{agg}^{-1}(0.05) := \max\{t, p_{agg}(t) \leq 0.05\}. \tag{3.7}$$

Groups 1 and 2 are then defined as follows:

$$\text{Group 1} = \mathcal{I}_{agg}(t_1^*) = \mathcal{I}_{agg}\left(p_{agg}^{-1}(0.025)\right), \tag{3.8}$$
$$\text{Group 2} = \mathcal{I}_{agg}(t_2^*) \setminus \mathcal{I}_{agg}(t_1^*) = \mathcal{I}_{agg}\left(p_{agg}^{-1}(0.05)\right) \setminus \mathcal{I}_{agg}\left(p_{agg}^{-1}(0.025)\right). \tag{3.9}$$

The groups 3 and 4 are defined similarly, that is:

$$\mathcal{I}_{inatt}(t) := \{i, \min THW(i) > t\}, \tag{3.10}$$
$$N_{inatt}(t) := \text{Card}(\mathcal{I}_{inatt}(t)), \tag{3.11}$$
$$p_{inatt}(t) := N_{inatt}(t)/N, \tag{3.12}$$
$$t_3^* = p_{inatt}^{-1}(0.025) := \min\{t, p_{inatt}(t) \leq 0.025\}, \tag{3.13}$$

$$t_4^* = p_{inatt}^{-1}(0.05) := \min\{t, p_{inatt}(t) \leq 0.05\}, \tag{3.14}$$

$$\text{Group 3} = \mathcal{I}_{inatt}(t_3^*) = \mathcal{I}_{inatt}\left(p_{inatt}^{-1}(0.025)\right), \tag{3.15}$$

$$\text{Group 4} = \mathcal{I}_{inatt}(t_4^*) \setminus \mathcal{I}_{inatt}(t_3^*) = \mathcal{I}_{inatt}\left(p_{inatt}^{-1}(0.05)\right) \setminus \mathcal{I}_{inatt}\left(p_{inatt}^{-1}(0.025)\right). \tag{3.16}$$

Applying the definitions of groups 1, 2, 3, and 4 above, we obtain the thresholds of Table 3.1 on $\overline{THW}$ for groups 1 and 2, and on $\min THW$ for groups 3 and 4; see also Figure 3.3. In Figure 3.4a and Figure 3.4b, we give the distribution of the average THW and minimum THW values respectively for the selected aggressive drivers and inattentive drivers. We can see that the value of minimun THW for two groups of aggressive drivers' ranges from about 0.1s to 0.8s. On the other side, the average THW of inattentive drivers is far larger than that of aggressive drivers, which ranges from about 2s to 5s.

Type of drivers	Indicator	Threshold
1st aggressive (group 1)	mean THW	$\leqslant 0.82s$
2nd aggressive(group 2)	mean THW	[0.82s, 0.94s]
1st inattentive (group 3)	min THW	$\geqslant 1.84s$
2nd inattentive (group 4)	min THW	[1.656s, 1.84s]

Table 3.1 – 4 groups specific drivers and obtained threshold of profile definition

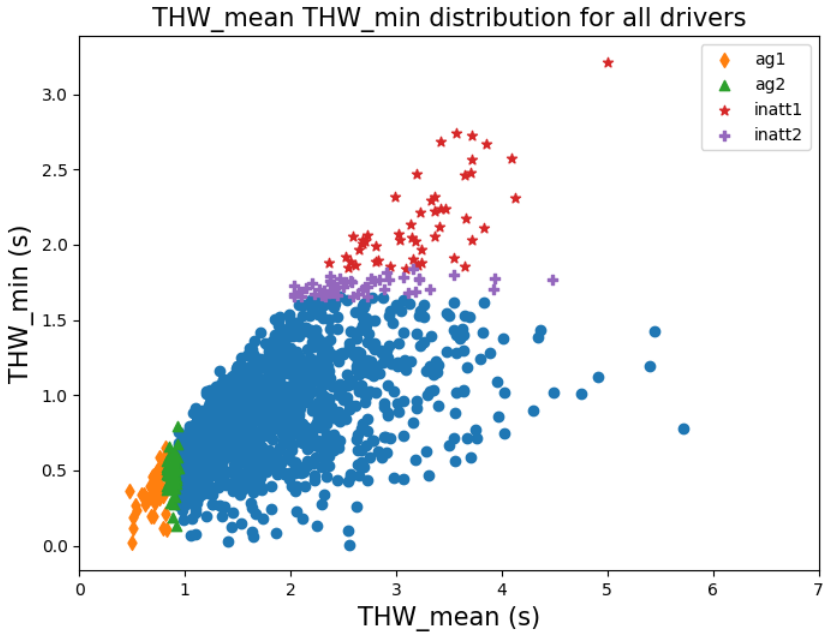
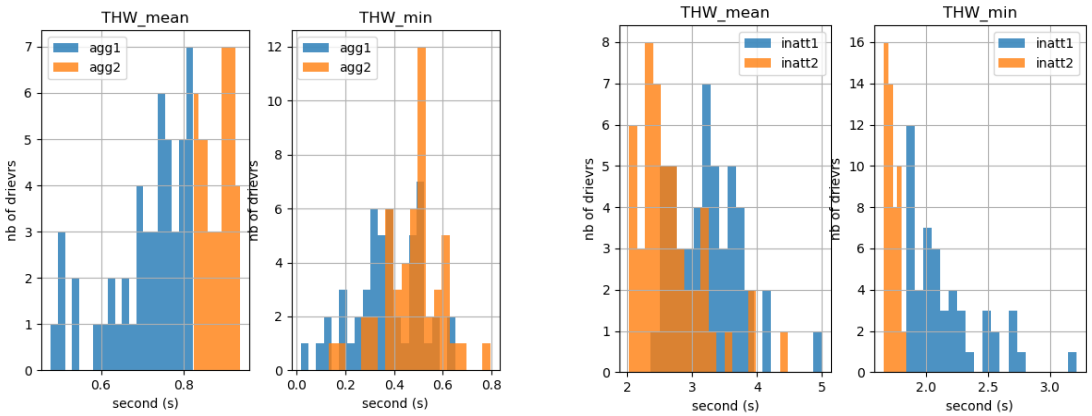


Figure 3.3 – $\overline{THW}$ (THW_mean) and $\min THW$ (THW_min) distribution of all drivers in the first 15 mins period data, group 1 in orange, group 2 in green, group 3 in red and group 4 in purple. The remainder drivers (blue points) are considered as drivers with 'normal' driving profile.



(**a**) aggressive drivers

(**b**) inattentive drivers

Figure 3.4 – $\overline{THW}$ (THW_mean) and $\min THW$ (THW_min) distribution for the selected aggressive drivers and inattentive drivers

3.5 SUMO traffic simulator and the choice of car-following model

Once we extracted the different driver profiles from NGSIM 101 data-set, we need a traffic simulator to simulate the NGSIM traffic and the microscopic traffic model for representing the driver profiles. We choose SUMO ("Simulation of Urban MObility") as the traffic numerical simulator to apply our proposition, which have beed already presented in the Section 2.6.1. As we are interested only in car-following behavior in this work, the generation of different driver profiles is performed by calibrating a car-following model. Many interesting reviews on car-following can be found in the literature (Brackstone and Mark, 1999; Mohammad and Z. Zheng, 2014; Aghabayk, Sarvi, and Young, 2015). We briefly present here some of the most commonly used. GHR model (Chandler, Herman, and Montroll, 1958) is a stimulus–response model in which drivers perform their acceleration/deceleration depending on car-speeds, relative car-speeds with respect to leading vehicles, inter-vehicle distances, and drivers' reaction time. Wiedemann (1974) car-following model is used in Vissim simulator and describes the psycho-physiological aspects of driving behavior in terms of four discrete driving regimes. This model considers different modes of operation, divided into 'no reaction zone' (free-road), 'closing in', 'must decelerate', and 'car-following' by human perceptual thresholds (Mohammad and Z. Zheng, 2014). Gipps model (Gipps, 1981) and Krauss model (Krauß, 1998) are based on safety distance. In the Gipps model, drivers update their car-speeds with respect to keep a minimum safety distance to the leader ones, in order to avoid collisions and while considering the extreme case of a leader car braking suddenly. Krauss model (Krauß, 1998) is the car-following model by default in SUMO simulator. It extends the Gipps model by modeling the imperfection of human driver with stochastic terms of car-speed. This propriety makes (or is supposed to make) the model more realistic. The IDM car-following model was first published in (Treiber, Hennecke, and Helbing, 2000) which improved the initial results produced by

the Gipps' model. The acceleration is calculated as a function of the desired speed and the desired space headway. The IDM model is suitable for both free flow and congestion phases (Mohammad and Z. Zheng, 2014). This attribute makes the model more efficient and facilitates its calibration with real driven data.

We chose here the IDM car-following model (Treiber, Hennecke, and Helbing, 2000), which is already implemented in SUMO, and it is shown by several research works that IDM is the most human-like car-following among certain compared models (Vasconcelos et al., 2014; M. Zhu, Xuesong Wang, Tarko, et al., 2018). Let us briefly recall the IDM model. The model is described by the following formulas :

$$\frac{dV}{dt} = a\left[1 - \left(\frac{V}{V_0}\right)^{\delta} - \left(\frac{s^*(V,\Delta V)}{s}\right)^2\right], \tag{3.17}$$

where

$$s^*(V,\Delta V) := s_0 + \max\left(0, VT + \frac{V\Delta V}{2\sqrt{ab}}\right), \tag{3.18}$$

with

dV/dt: vehicle acceleration

V: vehicle speed

s: inter-vehicular distance

ΔV: relative speed with leading vehicle

a: maximum acceleration

b: maximum deceleration

V_0: desired speed

δ: exponential parameter of speed, we fixed it at 4 referring to (Treiber, Hennecke, and Helbing, 2000)

s_0: minimal inter-vehicular distance with leading vehicle

T: desired time headway (desired THW)

The output of the IDM model is the acceleration dV/dt given as a function of the influencing factors (the inputs of the model): the own speed V, the inter-vehicular distance s and the relative speed ΔV with respect to the leading vehicle.

3.5.1 Reaction time in car-following models

We notice that the original IDM car-following model does not include reaction time as a parameter (Aghabayk, Sarvi, and Young, 2015). Nevertheless, in a recent development of SUMO simulator (Lopez et al., 2018), the authors improved the work on reaction time modeling. They indicated that reaction time could be introduced in the driving model (car-following and lane change models) as an additional parameter. In the work we present here, we introduced a reaction time parameter into the IDM car-following model. In SUMO, the simulation step can be set to Δt, and the driver reaction time is a time delay to make a decision both for updating acceleration or lane changing depending on the present state of traffic. This delay of decision time (reaction time) can be set as $(n * \Delta t)$, where n is an additional parameter which we determine by calibration.

3.6 Car-following model calibration method

To achieve a realistic representation of traffic behavior, the calibration with real driven data for the traffic model has to be applied. Two main types of calibration of car-following models exist: 1) Estimation of driving model parameters accounting for the physical meanings of each parameter (Lu et al., 2016), in which, parameters can be extracted directly from vehicle trajectory data; 2) Calibration of driving models, which can be constructed as an optimization problem. In the second approach, the objective function and the optimization method is selected so as to minimize the distance between the simulated vehicle trajectories by the model and real vehicle trajectories, and in order to find the optimal set of parameters. Several mathematical optimization methods and algorithms such as Newton, Gauss-Newton, gradient descent, and Levenberg-Marquardt are presented for car-following model parameters optimization in (Treiber and Kesting, 2013a). The authors of (Treiber and Kesting, 2013a) proved that genetic algorithms (GA) are also effective to solve optimization problems for car-following model calibration. Many of the recent works on car-following model calibration used GA methods (Vasconcelos et al., 2014; M. Zhu, Xuesong Wang, Tarko, et al., 2018). Their works show that GA optimisation method is efficient for car-following model calibration. In Table 3.2, an overview is given regarding to the several recent works on car-following models calibration by optimisation methods.

Articles	Car-flowing model	Optimization algorithm	Objective function	Data	Result
(Pourabdollah, 2017)	Krauss, IDM, wiedemann	GA	Normalized MSE for energy demand	Microscopic data within 200 different trips	IDM is best to represent drivers behaviors and energy demand
(Li,2016)	GHR, IDM	Nelder-Mead (NM), sequential quadratic programming (SQP), GA	Sum of square error (SSE) for position and velocity	NGSIM data	GA gives the best result for the calibration, IDM is better than GHR
(Zhu,2018)	GHR, GIPPS, IDM, FVD, wiedemann	GA	RMSE of spacing and velocity	42 Chinese drivers on urban expressway	IDM is the best and more stable
(Treiber,2014)	IDM	Levenberg -Marquardt algorithm	SSE of gaps, acceleration, and speed	NGSIM data	Calibration based on gaps error is more reliable

Table 3.2 – Overview of calibration of car-following models.

3.6.1 Pre-definitions of IDM model calibration

In our approach, the identification of drivers with 'extreme' profiles is carried out by calibrating the IDM model parameters, using real driven data. We use here the same method used in (M. Zhu, Xuesong Wang, Tarko, et al., 2018), where a genetic algorithm (GA) is applied and a measure of error is defined as the objective function. The used GA method is the default model in Matlab 2020b (R2020b, 2020) and for the algorithm description, it is presented in the book (Singiresu S. Rao, 2020). As what is presented, GA is inspired by biologic evolution carried out by mechanisms of inheritance, selection, mutation and recombination to find the optimal solution among all generations (sets of parameter groups).

In our work, the GA is implemented to find the optimal values of the IDM model parameters for each driver trajectory with the following procedure (same as what is presented in (M. Zhu, Xuesong Wang, Tarko, et al., 2018)):

1. A population of N individuals is initialized depending on the predefined parameter boundaries, each representing a set of parameters of the IDM model.

2. Computation of the fitness of each individual in the population, and the faintness is determined by a predetermined objective function (RMSPE on the vehicle position).

3. Crossover between randomly selected pairs of individuals (parents) and mutation within randomly selected individuals to produce the next generation (children)

4. Repeat steps 2 and 3 until the termination criteria are met.

The goodness of each individual (a parameter group) is evaluated by the objective function. In general, the mean square error (MSE) is widely used for calibration of car-following models. Here, another error metric is used as presented in (M. Zhu, Xuesong Wang, Tarko, et al., 2018), the root mean square percentage errors (RMSPE).

$$RMSPE(\theta) = \sqrt{\frac{1}{P}\sum_{i=0}^{P}\frac{(x_i^{simul}(\theta) - x_i^{data}(\theta))^2}{x_i^{data}(\theta)^2}}, \tag{3.19}$$

where x_i^{data} (respectively x_i^{simul}) denotes the car trajectory (positions in time) from data (respectively from simulation), and where θ represents all the parameters of the considered model. The car longitudinal position is chosen as the target variable in the objective function.

The IDM model calculates car acceleration, as described in equation (3.20), based on the drivers reaction time ($n * \Delta t$). Car's speed and position are calculated as follows in equation (3.21) and (3.22), based on the Euler Method.

$$a^{simul}(t + n\Delta t) = IDM(\theta; variables(t)) \tag{3.20}$$

$$v^{simul}(t + \Delta t) = v^{simul}(t) + a^{simul}(t) * \Delta t \tag{3.21}$$

$$x^{simul}(t + \Delta t) = x^{simul}(t) + v^{simul}(t) * \Delta t \tag{3.22}$$

In this work, the parameters which need to be calibrated are: a (maximal acceleration), b (maximal deceleration), V_0 (desired speed), s_0 (minimal inter-vehicular distance to leading vehicle) and T (desired THW), as well as the additional parameter n for the reaction time ($n * \Delta t$). In addition, the parameter δ (exponential parameter of speed) is fixed at $\delta = 4$ as proposed in (Treiber, Hennecke, and Helbing, 2000).

In the calibration process of the IDM model with the reaction time parameter for the extreme driver profiles, upper and lower bounds are needed to be set for all the parameters of IDM. For the *reaction time* and *desired THW* parameters, the bounds for these two parameters are different for the aggressive and inattentive driver profiles. For aggressive drivers, we chose the desired THW ranges from 0.1 to 4 s, while the reaction time is from 0.1 to 2 s. For inattentive drivers, we chose the desired THW ranges from 1 to 4 s while the reaction time goes from $tr_i - 0.3$ to $tr_i + 0.3$ seconds, where tr_i is the value of $\min \mathrm{THW}(i)$; see section 3.4.2 and we approximate the driver's reaction time to be around tr_i. For the other parameters, the bounds are the same for both aggressive and inattentive parameters. Thus, the desired speed (V_0) ranges from 10 to

40 m/s, s_0 ranges from 0.1 to 10 m, maximum acceleration and maximum deceleration range from 0.1 and 5 m/s^2 as what is proposed in (M. Zhu, Xuesong Wang, Tarko, et al., 2018).

For drivers with 'normal' profile, we simulate them using an average profile. Their calibration of IDM parameters is extracted directly from the data-set by their physical meanings and combined with the default values in SUMO simulator.

3.6.2 IDM model calibration result for extreme driver profiles

As mentioned above, the calibration of extreme driver profiles is considered as the optimization problem with the aim at minimizing the error between the simulated car-trajectories by IDM model and the real ones available from the data-set using a genetic algorithm. Moreover, we repeated the genetic algorithm 10 times for every driver trajectory, in order to approach the global optimum solution for the parameters of the calibration.

Firstly, we observed that the IDM model with the reaction time parameter is efficient for replicating the human driver behavior; see. A.1. The result by IDM model with the extension of reaction time is better than the original IDM model without reaction time parameter. Therefore, we use the IDM model with the reaction time parameter to present the driver profiles in numerical simulated traffic .

The results of calibration for the 4 specific groups of drivers are given in Tables A.1 - A.4 of Appendix A.2. For the calibration on the car-trajectories variable (car-positions in time), we obtained for all 4 driver profile groups an RMPSE ranging in (1% ,2%). From the calibration results, given in Tables A.1 - A.4, we notice that group 1 driver profiles have shorter reaction times and desired headway times compared to group 2 driver profiles. On the other hand, group 3 driver profiles have longer reaction times and desired headway times compared to group 4 driver profiles. Moreover, the desired THW (parameter T in formula (2.14)) of attentive drivers is much larger than the one of aggressive drivers.

3.7 Numerical simulation experiments setup

The reproduction of the NGSIM traffic data by numerical simulation consists of the creation of the road section, the selection of the microscopic traffic model (IDM car-following model, plus the default lane changing model in SUMO), and the simulation using SUMO. NETEDIT (SUMO, 2019a) tool of SUMO simulator allows to create manual map network of NGSIM, presented as a road of 640 meters with 5 straight lanes. The creation of each vehicle is provided by its longitudinal origin position, longitudinal destination position, entering lane, and entering time (in seconds). This necessary information is extracted from the car trajectory data. In

addition, during the simulation, SUMO can detect physical collisions (front and back bumper meet or overlap) (SUMO, 2019b). We can get the collision counts at the end of each simulation.

Four experiments of car-collision simulation are proposed for different combinations of the two selected groups of drivers. Each experiment is carried out using different combinations of one group of aggressive drivers and one group of inattentive drivers. Driver profiles used in simulation are described in Table 3.3, and the different groups of driver profiles have been described in Section 3.4.3.

Name of experiment	Aggressive drivers group	Inattentive drivers group
Experiment 1 (E1)	2.5% 1st aggressive drivers	2.5% 1st inattentive drivers
Experiment 2 (E2)	2.5% 2nd aggressive drivers	2.5% 1st inattentive drivers
Experiment 3 (E3)	2.5% 1st aggressive drivers	2.5% 2nd inattentive drivers
Experiment 4 (E4)	2.5% 2nd aggressive drivers	2.5% 2nd inattentive drivers

Table 3.3 – Numerical simulation experiments

As what we proposed in above, for example, for the experiment E1 (Table 3.3) which is the first experiment, we use 2.5% of first aggressive driver profiles, 2.5% of first inattentive driver profiles, and 95% of "normal drivers" as the initial scenario of the traffic. This corresponds to the original traffic condition of the first 15-minute sub-dataset in NGSIM 101 dataset. We simulate the traffic in SUMO for this first scenario, where all the driver profiles use their own calibrated parameters of the IDM model. The results are shown in Figure 3.5, where we give the mean traffic speed from the real data and the mean traffic speed presented from numerical simulations in SUMO. We can see that the traffic simulation represents well the state of traffic (car-speed) over time, compared to the real data.

We assume then the car-collisions can be generated in each simulation by increasing the number of aggressive and/or inattentive drivers in the traffic. The increasing of the percentages of extreme driver profiles is done artificially and randomly by replacing the same number of 'normal' drivers by the selected extreme driver profiles. Thus, we reset the simulation by increasing the rates of drivers associated to each extreme driver profile (2.5% in the origin data-set, 2.5%, 5%, 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45% and 50% in simulation). For each simulation, once the rates of the extreme driver profiles are fixed, all the other drivers are taken as 'normal'.

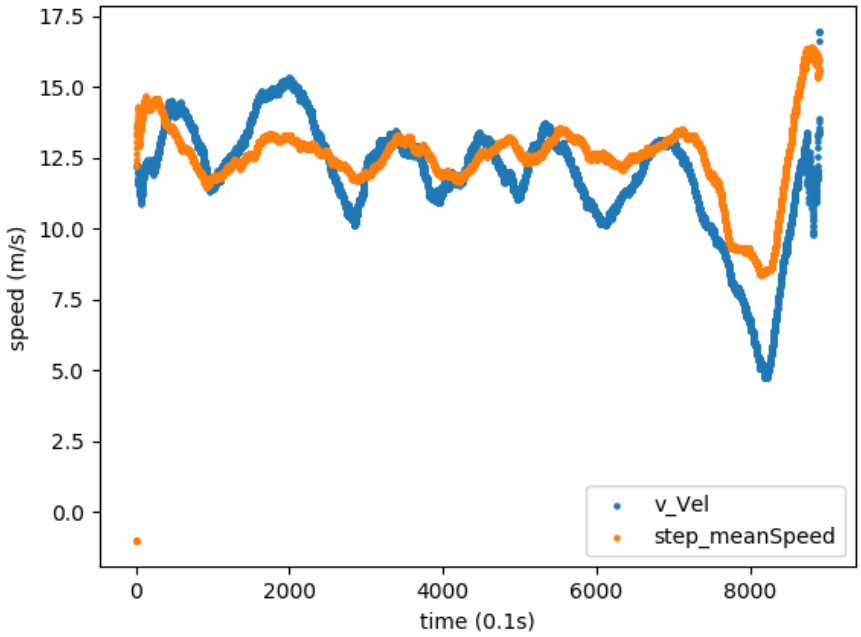


Figure 3.5 – The average traffic flow speed from real data (blue) and the simulated average car speed (orange) in SUMO of the initial traffic of the first 15 minutes sample, where 2.5% are aggressive drivers simulated using their own IDM calibrated parameters, and 2.5% are inattentive drivers simulated using their own IDM calibrated parameters. The remaining driver profiles (95%) are normal drivers in the traffic.

3.8 The results for vehicular collision generation

In this section, we give the results of the numerical simulations for the car collision generation. In subsection 3.8.1, we provide the number of collisions generated by numerical simulation based on the first 15 minutes data. The results are presented for the 4 predefined scenarios. Moreover, in order to understand the relationship between the driver profiles and the generated car-collisions, we give in subsection 3.8.2 a deeper analysis of the collisions obtained in the numerical simulation experiment 1 (E1). In subsection 3.8.3, we show that the number of simulated car-collisions can be written as a function of the rates of aggressive and inattentive driver profiles in the traffic.

After the presentation of the obtained results on the first 15-minutes sub-dataset, we give in section 3.9 the results for the validation of the proposed approach using the second 15-minute time period of the NGSIM 101 dataset.

3.8.1 Number of collisions by numerical simulation

We start with the implementation of experiment E1. We observed that the number of collisions occurred in each simulation is various even when using the same rate of specific drivers. This is due to the random distribution of the specific (aggressive and inattentive) driver profiles over all the drivers. Nevertheless, we observed that the mean number of car-collisions converges in simulation time, for each case. Thus, the presented collision counts in each case shown in the following tables are the average value over 10 simulations for each case. The number of collisions obtained by numerical simulation is given in Tables 3.4-3.7 for each experimentation (E1, E2, E3, E4) respectively.

1st aggressive drivers (horizontal) 1st inattentive drivers (vertical)	2.5%	5%	10%	15%	20%	25%	30%	35%	40%	45%	50%
2.5%	0	0	1	3	3	5	6	7.	11	11	13
5%	0	0	2	4	5	7	10	10	12	18	19
10%	4	4	7	10	12	16	21	26	29	38	43
15%	5	7	11	17	21	27	31	37	46	53	52
20%	8	11	16	18	24	31	38	46	56	64	69
25%	11	13	20	26	36	45	50	57	66	72	82
30%	12	16	22	30	39	49	60	62	75	87	92
35%	17	20	29	37	46	52	60	73	78	92	99
40%	19	22	28	35	50	55	65	76	88	96	106
45%	23	25	33	46	58	65	78	88	92	106	123
50%	25	32	42	52	62	74	87	95	105	118	124

Table 3.4 – Experiment 1: Simulated collision counts for different percentages of 1st aggressive and 1st inattentive drivers. For example, in the case that 25% for aggressive, and 25% for inattentive, the remaining 50% are normal drivers.

2nd aggressive drivers (horizontal) 1st inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0	0	1	3	4	5	7
5%_inatt	0	1	2	3	5	7	8
10%_inatt	4	5	5	9	13	16	17
20%_inatt	10	11	15	22	25	35	43
30%_inatt	12	14	17	24	32	39	52
40%_inatt	21	21	34	36	47	53	63
50%_inatt	27	29	37	48	61	69	80

Table 3.5 – Experiment 2: Simulated car-collision counts for different percentages of 2nd aggressive and 1st inattentive drivers.

1st aggressive drivers (horizontal) 2nd inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0	0	2	4	4	8	10
5%_inatt	0	1	1	6	6	9	11
10%_inatt	2	2	2	9	15	18	20
20%_inatt	5	11	10	18	28	35	56
30%_inatt	11	11	18	25	38	54	69
40%_inatt	19	16	28	45	53	78	97
50%_inatt	24	29	32	55	73	82	101

Table 3.6 – Experiment 3: Simulated car-collision counts for different percentages of 1st aggressive and 2nd inattentive drivers..

2nd aggressive drivers (horizontal) 2nd inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0	0	1	2	2	4	8
5%_inatt	0	1	2	2	4	5	6
10%_inatt	2	1	2	4	6	10	14
20%_inatt	4	5	8	14	14	16	30
30%_inatt	8	12	12	16	28	30	42
40%_inatt	16	16	22	24	28	34	52
50%_inatt	25	20	26	38	42	52	58

Table 3.7 – Experiment 4: Simulated car-collision counts for different percentages of 2nd aggressive and 2nd inattentive drivers.

In Table 3.4, the simulated collision counts are shown along with different rates of specific profiles for E1. The results indicate clearly that the number of inter-vehicular collision occurrences increases with the considered number of aggressive and/or inattentive driver profiles. The results for E2, E3, and E4 are given respectively in Tables 3.5-3.7. The number of obtained collisions is lower compared to the ones of Table 3.4. The number of collisions obtained with E4 is the lowest one compared to the other three experiments. This result shows that a different risk level is associated to each profile. The first group of aggressive driver profiles could potentially cause more collisions than the second group of aggressive driver profiles (comparison of E1 and E2 results). Similarly, the first group of inattentive driver profiles could cause more collisions than the second group of inattentive driver profiles (comparison of E3 and E4 results). Interestingly, in the four experiments, no collisions occurred when the percentage of 'extreme' drivers was small. With the increase of the rate of each specific driver profile, the number of collisions increases. However, this increase seems to be different from one experiment to an other; see Figure 3.6, where each line presents the number of collisions in the simulation of each four experiments, under the condition that aggressive drivers and inattentive drivers have

the same rate in the traffic. The Experiment 1, where we generated collisions based on varying the percentage of the most aggressive and the most inattentive drivers, can generate more collisions than the other three experiments. The Experiment 4 has probably the lest number of collisions, where we generated collisions based on varying the percentage of the second aggressive and the second inattentive drivers.

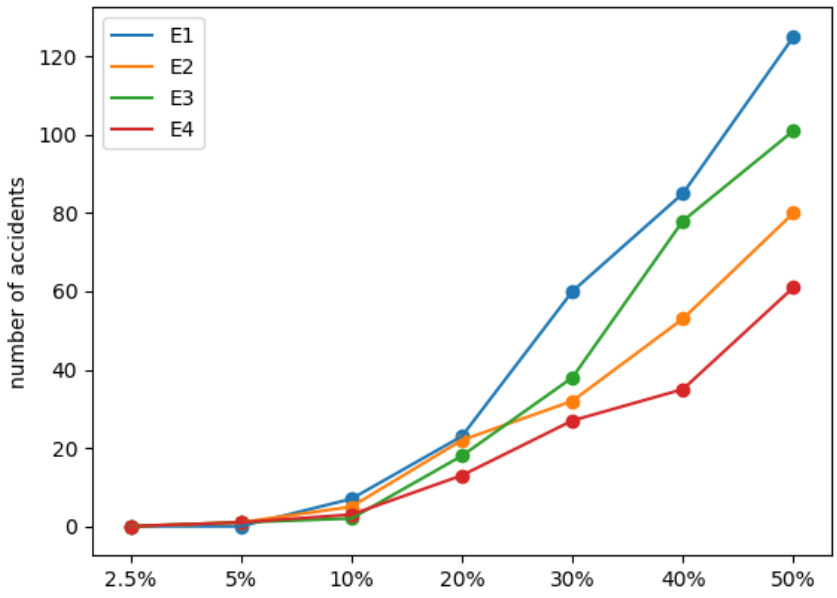


Figure 3.6 – Simulated car-collision counts for the four experiments, where aggressive and inattentive drivers have same rate (axis-x) in traffic (based on the first 15-min data). For example, 2.5% (axis-x) means each profile (aggressive, inattentive) takes 2.5% in the traffic.

3.8.2 Analysis of collisions obtained in Experiment 1 (E1)

We further analyze here the results obtained in E1 (Table 3.4) to investigate the relationship between generated car collisions and driver profiles. Figure 3.7 depicts the number of the simulated collisions as a function of the two rates of aggressive and inattentive driver profiles. More precisely, Figure 3.7 shows the contours of this function, which takes indeed the values of the result shown in Table 3.4. From Figure 3.7, it seems that the sum of the two rates of aggressive and inattentive driver profiles is important in determining the number of collisions. Indeed, we can see on Figure 3.7 that the contours resemble to decreasing straight lines with slope (-1). However, the contours are not linear, which means that the sum of the rates of aggressive and inattentive driver profiles, although important, is not the unique parameter for the determination of the number of collisions. This will be confirmed in the next section,

where, using statistical regression, we approximate the number of collisions with a non linear function of the two rates of aggressive and inattentive driver profiles.

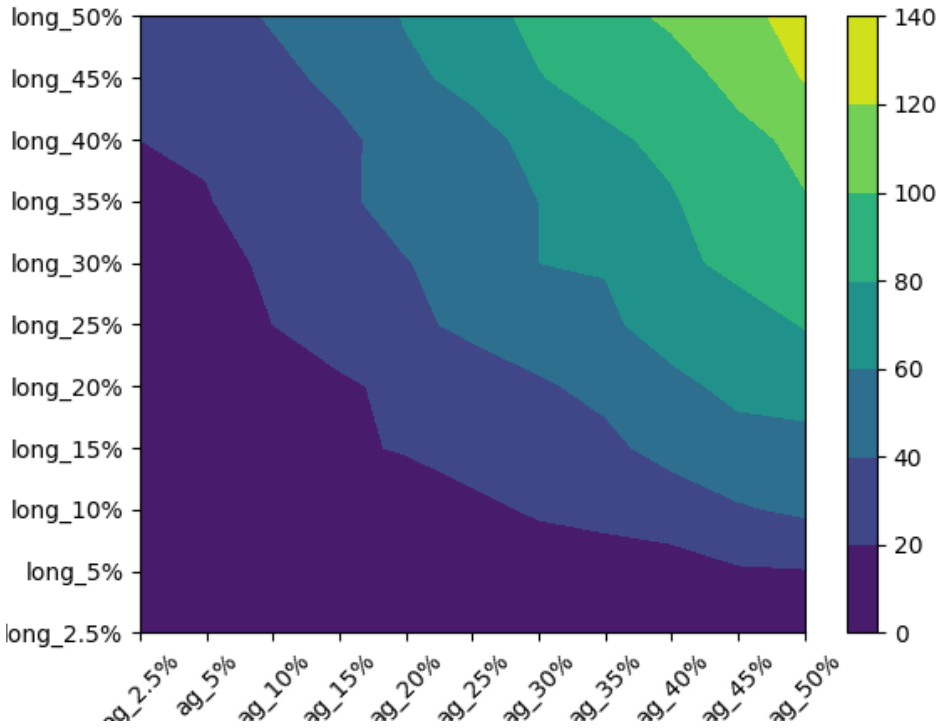


Figure 3.7 – Counter curves of the number of generated car-collisions, function of the percentages of aggressive and inattentive driver profiles if the traffic, for scenario E1 on first 15-minute data. Abbreviations "ag" and "long" mean "aggressive" and "inattentive" respectively.

As mentioned above in section 3.7, a car-collision is observed in SUMO when a following vehicle (veh2) collides the rear-end of the proceeding vehicle (veh1). We can get information on the profiles of the two vehicles involved in each collision. The distribution of rear-end collisions produced by different driver profiles (aggressive, normal, and inattentive) in percentage is shown in Figure 3.8. Abbreviations *ag*, *long*, and *nor* are used to indicate aggressive, inattentive, and normal driver profiles respectively. In Figure 3.8, we give the percentages of the number of collisions caused by each pair of driver profiles rates. Thus, collisions caused by the pair inattentive(veh1)-aggressive(veh2) of driver profiles, in this order, account for the largest ratio, which is 28.99 % of the whole number of simulated collisions.

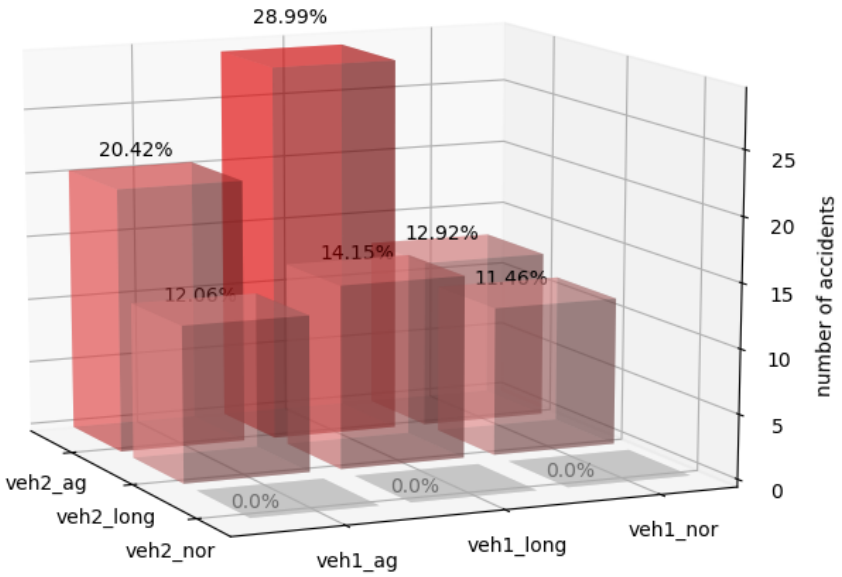


Figure 3.8 – Distribution of involved driver profile types in simulated rear-end collisions (E1 based on first 15-minute data), veh1 indicates the leading vehicle, veh2 indicates the following vehicle. The suffix ag, long, nor are the aggressive, inattentive, normal driver profiles respectively.

3.8.3 The number of collisions as a function of the rates of aggressive and inattentive driver profiles in traffic

According to the results given in Table 3.4, the relationship between the number of simulated collisions and the rates of aggressive and inattentive driver profiles is shown in Figure 3.7. In order to better understand this relationship, we use statistical regression and assume that car-collision counts (Y) are proportional to the percentage of aggressive drivers (x_1 %) and the percentage of inattentive drivers (x_2 %), which can be formulated as $Y = f(x_1, x_2)$.

We have first tried linear regression and the result is shown in section A.3.1 of Appendix A.3, which concludes that the assumption of linearity in not convincing. Indeed, we observe from Table 3.4 (red cases), for example, that the number of collisions generated with 30 % of aggressive driver profiles and 30 % of inattentive driver profiles is 60. Holding on this 60%, however, with 25 % (also respectively 20 %, 15 %, 10 %, 35 %, 40 %, 45 %, and 50 %) of aggressive driver profiles, and 35 % (also respectively 40 %, 45 %, 50 %, 25 %, 20 %, 15 %, and 10 %) of inattentive driver profiles, the number of generated collisions is less substantial, although the sum of all the pairs of considered rates of aggressive and inattentive driver profiles is 60 %. Moreover, it seems that the number of generated collisions decreases with the increasing of the absolute difference between the rates of aggressive and inattentive driver profiles. In other terms, the number of generated collisions increases with the uniformity of mixing both profiles

(aggressive and inattentive). Based on this observation, we propose to add a quadratic term of the approximation of the relationship between the number of generated collisions and the rates of aggressive and inattentive driver profiles.

$$\hat{Y} = f(x_1, x_2) = \max\left(0, \beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \beta_3 * (x_1 - x_2)^2\right), \tag{3.23}$$

where $\hat{Y}$ is the estimation of car-collision counts; x_1 % is the percentage of aggressive drivers and x_2 % is the percentage of inattentive drivers; We lower bound the function by zero, since the number of collisions cannot be negative. We obtained the following coefficients: $\beta_0 = -23.35, \beta_1 = 1.4201, \beta_2 = 1.3602$, and $\beta_3 = -1.4377$, with RMSE $= 4.864$; Let us note that another non-linear regression has been tested in section A.3.2 of Appendix A.3. However, the results are not better than the ones obtained by the regression (3.23).

We notice here that $\beta_3 < 0$. First, the fact that $\beta_3 \neq 0$ confirms the significance of the effect of the absolute difference between the two rates of aggressive and inattentive driver profiles on the number of generated collisions. Second, the fact that β_3 is negative confirms our hypotheses, that the number of generated collisions decreases with the increase of the absolute difference between the rates of aggressive and inattentive driver profiles.

In Figure 3.9, the number of collisions for each percentage of the two drivers profiles is presented, as well as the surface obtained by the non-linear regression.

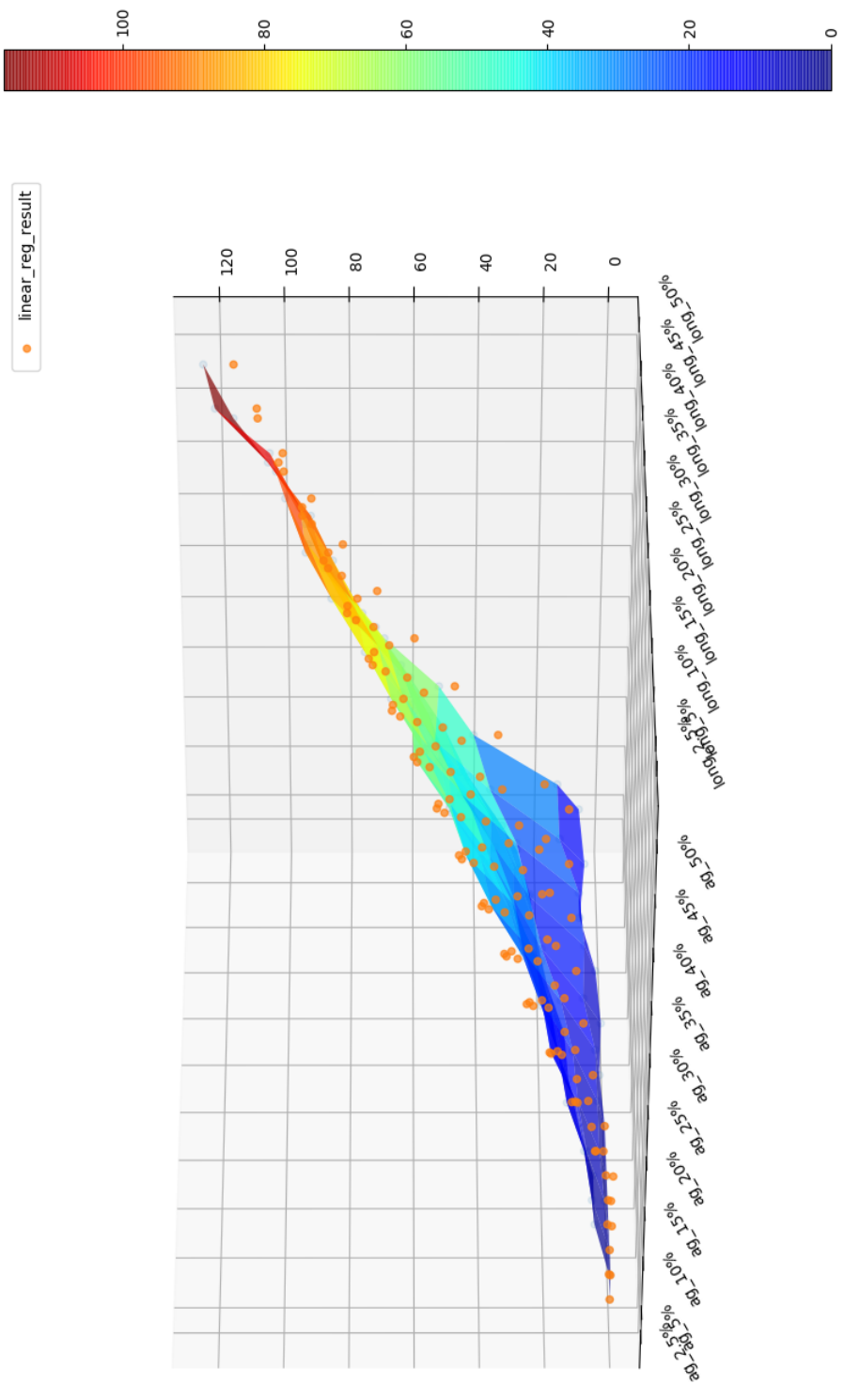


Figure 3.9 – E1: Non-linear regression of the number of simulated collisions. Abbreviations "ag" and "long" indicate "aggressive" and "inattentive" driver profiles respectively.

3.9 Validation of the approach on the second 15-minute time period of data

In this section, the same approach (summarized in Figure 3.1) for car-collision generation is carried out based on the second 15-min time period of the NGSIM 101 data-set, with the purpose of validation of the approach, where 1495 vehicles are registered in this period on the 5 normal traffic lanes. Similarly to the work using the first 15-minute data, we begin by selecting the 4 driver profiles groups by the percentage of 2.5%. As the result, the thresholds on $\overline{THW}$ (mean THW) and $\min THW$ for each driver profile are shown in Table 3.8. The thresholds obtained for the 4 groups of drivers are similar to the ones obtained based on the first 15-minute period of data (comparison of Table 3.8 and Table 3.1). In addition, the simulation of the original traffic in SUMO for this period is shown in Figure 3.10. This Figure shows that the traffic simulation is well calibrated for the second period of data, with initial percentage of aggressive drivers and inattentive drivers (2.5% for each group) in the traffic, and the limit speed conditions on the two sides of the road.

Type of drivers	Indicator	Threshold
1st aggressive (group 1)	mean THW	$\leqslant 0.86s$
2nd aggressive (group 2)	mean THW	[0.86s, 0.98s]
1st inattentive (group 3)	min THW	$\geqslant 1.99s$
2nd inattentive (group 4)	min THW	[1.74s, 1.99s]

Table 3.8 – Selected drivers and the associated threshold for the second 15-minute data.

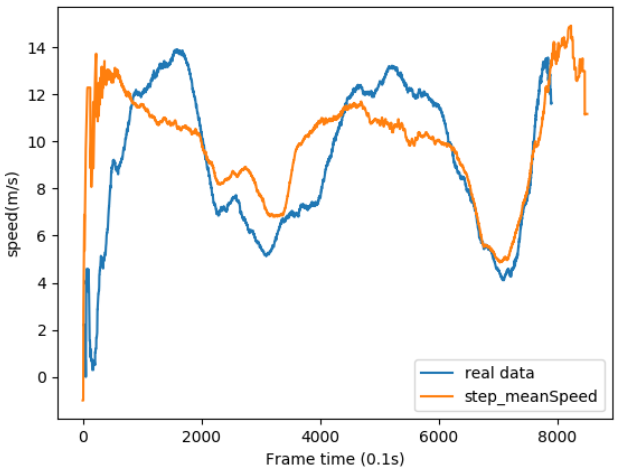


Figure 3.10 – Average traffic flow speed of real data (blue) and simulation (orange) in SUMO for the second 15-min data.

3.9.1 Number of collisions by numerical simulation based on the second 15-minute data

The simulated resulting collisions, for the 4 experiments (E1-E4), are shown in Tables 3.9-3.12. Furthermore, Figure 3.11 shows the number of generated collisions for the four experiments.

1st aggressive drivers (horizontal) 1st inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0.0	2.0	4.0	9.0	11.0	17.0	20.0
5%_inatt	3.0	5.0	7.0	12.0	15.0	22.0	26.0
10%_inatt	11.0	14.0	16.0	21.0	30.0	34.0	39.0
20%_inatt	25.0	28.0	34.0	42.0	50.0	65.0	70.0
30%_inatt	34.0	39.0	45.0	51.0	56.0	68.0	79.0
40%_inatt	47.0	52.0	57.0	70.0	79.0	91.0	105.0
50%_inatt	58.0	60.0	68.0	79.0	82.0	102.0	115.0

Table 3.9 – Experiment 1: Simulated car-collision counts by varying respectively the percentage of 1st aggressive and 1st inattentive drivers for second 15-min period.

2nd aggressive drivers (horizontal) 1st inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0.0	1.0	2.0	5.0	7.0	12.0	16.0
5%_inatt	4.0	5.0	6.0	9.0	13.0	15.0	19.0
10%_inatt	9.0	10.0	12.0	16.0	18.0	24.0	29.0
20%_inatt	22.0	25.0	26.0	32.0	37.0	41.0	44.0
30%_inatt	38.0	37.0	37.0	44.0	45.0	55.0	61.0
40%_inatt	44.0	48.0	52.0	56.0	61.0	67.0	78.0
50%_inatt	58.0	59.0	60.0	68.0	84.0	83.0	88.0

Table 3.10 – Experiment 2: Simulated car-collision counts by varying respectively the percentage of 2nd aggressive and 1st inattentive drivers over the second 15-minute period.

The result using the second 15-minute period data is close to the result using the first 15-minute period. The simulated collisions using the second 15-minute data show the same trend as the first 15-minute data, where Experiment E1 simulates the greatest number of collisions, while E4 simulates the lowest number of collisions.

1st aggressive drivers (horizontal) 2nd inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0.0	1.0	4.0	10.0	12.0	19.0	20.0
5%_inatt	2.0	3.0	5.0	10.0	11.0	19.0	24.0
10%_inatt	6.0	7.0	8.0	14.0	18.0	26.0	31.0
20%_long	16.0	18.0	18.0	25.0	30.0	37.0	43.0
30%_inatt	22.0	26.0	29.0	33.0	38.0	47.0	54.0
40%_inatt	29.0	34.0	35.0	43.0	52.0	58.0	68.0
50%_inatt	33.0	33.0	40.0	50.0	53.0	69.0	72.0

Table 3.11 – Experiment 3: Simulated car-collision counts by varying respectively the percentage of 1st aggressive and 2nd inattentive drivers over the second 15-min period.

2nd aggressive drivers (horizontal) 2nd inattentive drivers (vertical)	2.5%_ag	5%_ag	10%_ag	20%_ag	30%_ag	40%_ag	50%_ag
2.5%_inatt	0.0	1.0	2.0	7.0	9.0	10.0	15.0
5%_inatt	2.0	4.0	6.0	8.0	11.0	16.0	16.0
10%_inatt	5.0	6.0	7.0	10.0	12.0	17.0	19.0
20%_inatt	13.0	15.0	18.0	20.0	22.0	27.0	30.0
30%_inatt	20.0	22.0	23.0	26.0	31.0	35.0	38.0
40%_inatt	28.0	28.0	31.0	32.0	37.0	42.0	43.0
50%_inatt	30.0	32.0	32.0	37.0	40.0	45.0	46.0

Table 3.12 – Experiment 4: Simulated collisions number by varying respectively the percentage of the 2nd aggressive drivers and the 2nd inattentive drivers of second 15-min data.

3.9.2 The number of collisions and the rates of different driver profiles in traffic based on the second 15-minute data

The results of the number of collisions for experiment E1 based on the second 15-min data are given in Table 3.9 and Figure 3.12. Therefore, as for the first 15-min time period data, a non-linear regression is performed (Equation 3.23), where we get the following coefficients: $\beta_0 = 0, -8.154, \beta_1 = 1.580, \beta_2 = 0.8502$, and $\beta_3 = -0.7575$, with an RMSE $= 3.761$.

In Figure 3.13, the number of collisions for each percentage of two drivers profiles is presented, as well as the surface obtained by the non-linear regression by the proposition (Equation 3.23).

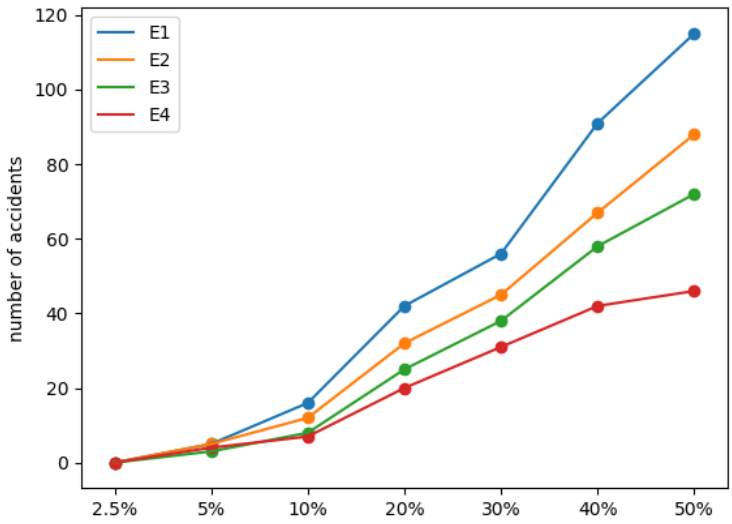


Figure 3.11 – Simulated car-collision counts for the four experiments, where aggressive and inattentive drivers have same rate (axis-x) in traffic (based on the second 15-min data). For example, 2.5% (axis-x) means each profile (aggressive, inattentive) takes 2.5% in the traffic.

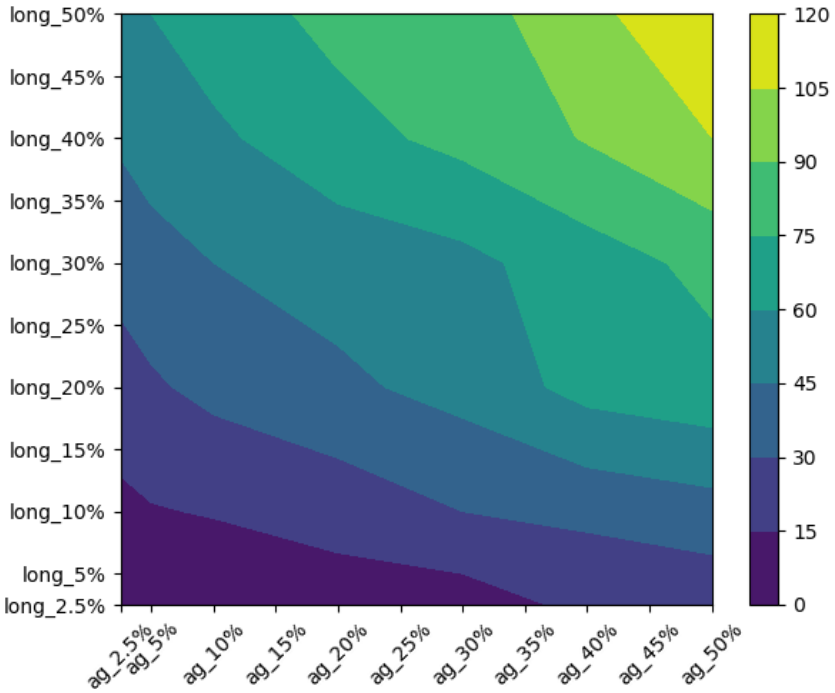


Figure 3.12 – E1: Simulated car-collision counts for different percentages of aggressive and inattentive drivers based on second 15-minute data. The suffix ag, long are the aggressive, inattentive driver profile respectively.

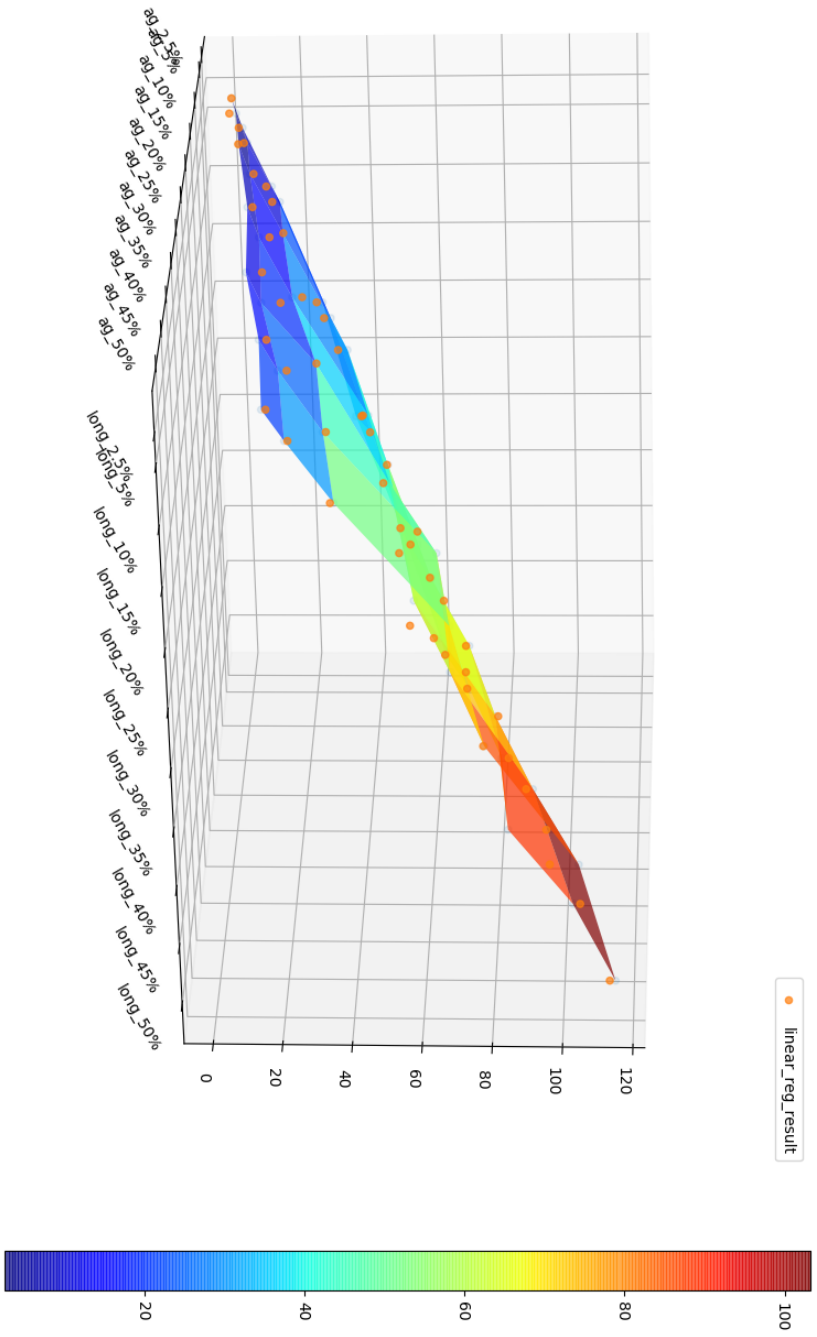


Figure 3.13 – Second 15-mints data E1: Non-linear regression of the number of simulated collisions. Abbreviations "ag" and "long" denote "aggressive" and "inattentive" driver profile respectively.

3.10 Analysis of car-collision severity

All the simulated car-collisions can be further used to explore car-collision severity. Several works have investigated the relationship of rear-end car-collision severity to car speed. In (Jurewicz et al., 2016), the authors reported that the critical impact speed was approximately 55 km/h for rear-end collisions. In addition, in (Elvik, 2013), the authors claim that the change of speed before and after the car-collision is a critical indicator for car-collision severity outcomes.

In physics, the kinetic energy (KE) of an object is the energy that it possesses due to its speed (3.24). In the case of front-rear collision of two moving cars, the kinetic energy is related to the relative speed of the front vehicle and the rear car (3.25).

$$KE = \frac{1}{2} m * v^2 \tag{3.24}$$

$$KE \propto \Delta v \tag{3.25}$$

Relative speed could thus be an important surrogate to indicate the severity of collision. In simulations with 50% drivers being aggressive and the other 50% being inattentive, 866 collisions were generated (with 5 simulations). Among all collisions, 220 (25.4%) concerned aggressive-aggressive couples of driver profiles, 408 (47.1%) concerned inattentive-aggressive, 107 (12.4%) concerned inattentive-inattentive, and 131 (15.1%) concerned aggressive-inattentive. In Figure 3.14, the distribution of relative speed for these 866 simulated collisions is shown. collisions implicating inattentive(veh1)- aggressive(veh2) drivers account for the largest rate of all simulated collisions. All generated collisions have a relative speed below 50 km/h. Furthermore, collisions are more severe when they implicate two inattentive drivers, since the mean relative speed is at 23.27 km/h, which is more critical than in the other cases. Car-collision severity is much lighter between two aggressive drivers who have an average relative speed at 15.7 km/h.

SCANeR Studio (AVSimulation, 2019)(Champion et al., 1999) is a driving simulation software, which also includes Bullet physic engine (a free and open-source software of simulation of collision detection, soft and rigid body dynamics). It provides the opportunity to simulate vehicle collisions with a vehicle physical model. For every simulated car-collision in SUMO, the involved two vehicles' trajectory can be registered. Then the car-collision scene can be replayed in SCANeR Studio. Thus, the result of a car-collision simulated in SUMO can be re-simulated in SCANeR Studio; see Figure 3.15. In Figure 3.16, the collision force for certain replayed collisions is shown. Each point corresponds to one re-simulated car-collision by SCANeR. It seems that the force is a super-linear function of the relative speed; see Figure 3.16. However, this fact needs more investigations and should be considered in our future research. The objective will

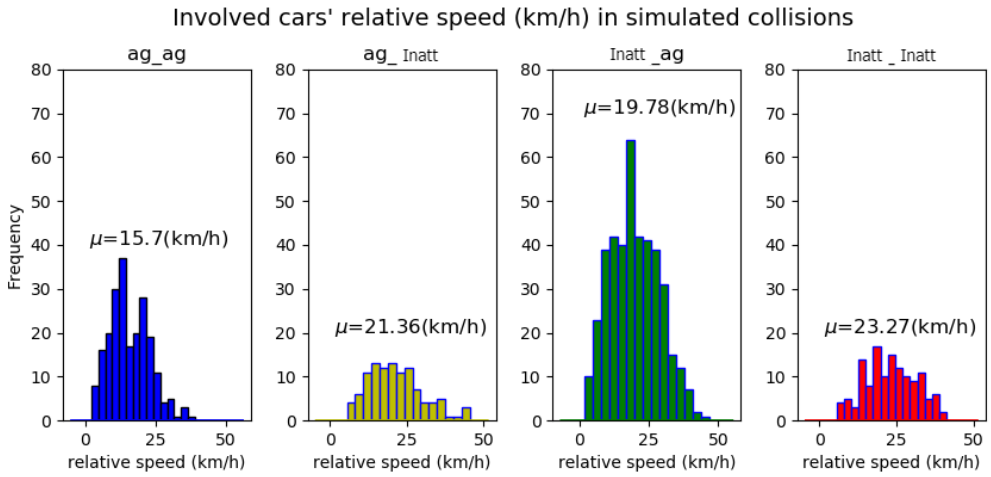


Figure 3.14 – The distribution of relative speed for simulated collisions (μ indicates the mean value for each sub-figure), Abbreviations "ag" and "inatt" denote "aggressive" and "inattentive" driver profile respectively. It shows that the most collisions are occurred between inattentive-aggressive drivers, while the most severe (dangerous) collisions are produced between inattentive-inattentive drivers.

be to couple SUMO traffic simulator with SCANeR studio (the immersive driving simulator) in order to build a high performance system to test and validate autonomous vehicles.

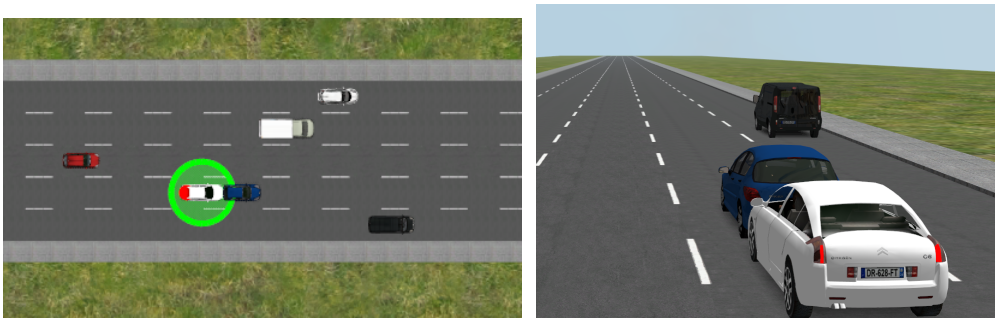


Figure 3.15 – SCANeR screenshot for a collision.

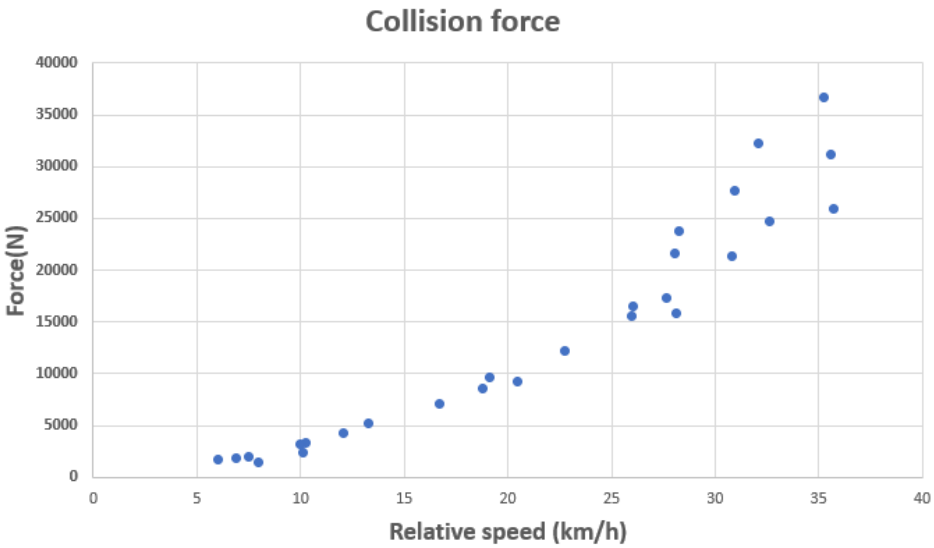


Figure 3.16 – The collision force by SCANeR as a function of the relative speed (Y-axis for collision force (Newton) and X-axis for relative speed of collision (km/h)).

3.11 Conclusion and perspectives

In this work, the main contributions consist in 1) extracting three proposed driver profiles based on real driven data-set; 2) replicating them in simulated environment; and 3) establishing a relationship between car-collision occurrences and these different driver profiles by varying their percentages in the whole traffic via virtual simulation.

Based on the NGSIM 101 data-set, two specific driver profiles which are related to car-collisions on road networks have been characterized, defined as i) aggressive drivers who keep short time-headways with their leaders, and ii) inattentive drivers with long reaction-times. All the other driver profiles are considered as "normal" with intermediate values of reaction time and time-headway. These three driver profiles have been simulated by using the *Intelligent Driver Model* (IDM) car-following model, with an extension including driver reaction time. In order to represent the real driver profiles, the IDM model is calibrated using a genetic algorithm. Finally, by increasing the percentage of the two extreme driver profiles (aggressive and inattentive) over the whole traffic in virtual traffic simulation, we investigated the effect of these driver profiles on the car-collision occurrences.

The results of the numerical simulations show that the percentages of the aggressive and inattentive driver profiles over the whole driver population are determinant in the car-collision occurrences and in the resulting severity outcomes. One of the important results we obtained in this work is the characterization of the relationship between the ratios of these two driver profiles over the whole drivers population, and the car-collision occurrences counts. We have also classified the occurred car-collisions and analyzed their severity, in particular with respect

to the relative speed between the cars which cause the collisions. Another important result of this research is that, the generated car-collisions involved between an aggressive driver as a leader and an inattentive driver as a follower occupy the most frequent collision occurrences, meanwhile the generated car-collisions involved between two inattentive drivers are the most severe ones.

The safety validation of Intelligent Connected Vehicles is essential and could be critical for their deployment. In order to complete the demonstration of the reliability of the system's safety, autonomous vehicles need to be driven for hundreds of millions of miles. However, the huge cost of such physical tests, combining with its inherent danger of testing situations where collisions can happen, makes the numerical simulation of scenarios mixing different driver profiles as an important safety assessment tool for ICVs testing. Indeed, during the deployment phase of ICVs, the recognition of drivers' profiles should be considered to avoid collision generation. As shown in our experiments, based on our approach, different traffic scenarios can be generated with different driver profiles in traffic simulation. For future research, this work is expected to highly facilitate future ICVs testing and validation for the car manufacturing industry via numerical traffic simulation.

Chapter 4

Imitation of Real Lane-Change Decisions Using Reinforcement Learning

Microscopic modeling of human driving consists generally in combining both car-following and lane-change models. While the human car-following process has been extensively developed and well modeled, the lane-change behavior is more complex to understand and still remains to be explored. Classical lane-change models are usually rule-based and handcrafted, that tend to exhibit limited performance. Machine Learning algorithms, particularly Reinforcement Learning (RL) ones, provide an alternative approach and have recently achieved high success in modeling difficult decision-making processes in many fields. We propose in this chapter a reinforcement learning based model for the human lane-change behavior, with an online calibration of real lane-change decisions, extracted from the NGSIM data-set. We use the traffic vehicular simulator SUMO ("Simulation of Urban MObility") to create a numerical simulation environment. The utilization of numerical traffic simulation allows us enriching the data-set, for training the agent to find an optimal policy for lane change. Thus, about 13% additional traffic situations, not present in the real data, are created by the traffic simulation environment. The trained agent is collision-free and human-like. It is able to reproduce well lane-change behaviors in real data and those in the additional simulated data. Indeed, our RL model can perform up to 95.37% of the real decisions observed in the data-set.

Contents

4.1 Motivation

Driving behaviour models describe the drivers manoeuvring decision in different traffic conditions, which are important for traffic researches and for microscopic traffic simulation. Car-following (CF) and lane-change (LC) models are the two fundamental models in human driving modelling. CF models have been well developed. They describe the longitudinal behavior when a given vehicle follows the leading one. Most of literature reviews focus on summarizing car-following models (Mohammad and Z. Zheng, 2014; Aghabayk, Sarvi, and Young, 2015). Considering LC behavior, the modeling efforts are not as many as in the CF behavior. This is due to the complexity of the LC behavior, which is affected by the surrounding vehicles (of the considered vehicle) and by the traffic flow environment (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007). In some reviews of lane changing models (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007; Moridpour, Sarvi, and Rose, 2010; Z. Zheng, 2014), the authors present several rule-based lane changing models, but they tend to exhibit limited performance due to the uncertainty and to the complexity of the driving environment. For this reason, Machine Learning methods, and especially Reinforcement Learning (RL) ones, provide an alternative approach, which has shown a great success in many different domains, such as robotics, video game playing, dialogue chat bot, etc.

Reinforcement Learning (RL) is a general framework for learning-based sequential decision making. With reinforcement learning, an agent can learn to solve problems where the decision process is facing previously unknown situations through trial and error. Therefore, using reinforcement learning combined with simulated traffic environment, vehicles can be trained to develop a lane changing strategy, trying an action at a situation, and observing the result of this action. Being rewarded for good actions and penalised for bad ones, they learn how to act optimally through the experience, which is quite useful considering all the different situations that might exist, and how difficult it would be to program the vehicle's actions in all these situations.

Many researchers have shown the great efficiency of using RL for car-driving modeling. In (L. Wang et al., 2019), the authors developed an ego-efficient lane-change strategy using Q-learning and also tested the effects of the strategy on the network level through Aimsum traffic simulation platform. With the aim to study both longitudinal and lateral movements, the authors of (Hoel, Wolff, and Laine, 2018) proposed a model for the automation of both speed and lane-change decisions based on Deep Q Learning. Another approach was followed by (P. Wang, H. Li, and Chan, 2019) who used an actor-critic algorithm for the automation of lane-change decisions, and the problem has been solved using the Deep Deterministic Policy Gradient. In (Ye et al., 2020), an actor-critic algorithm is also used to propose a method for automating lane-change decisions. The authors of (Ye et al., 2020) proposed a model for mandatory lane-changes, developed using the Proximal Policy Optimization algorithm. All

this relative works are using only simulation environments and propose intelligent lane-change models, but the calibration with real human driving behavior has not been mentioned in these relevant works. However, real human driving behavior needs to be considered to make the model more realistic. Even for autonomous vehicle, a human-like driving model could be more comfortable for the passengers.

4.2 Background of Imitation Human Driving Behavior

Imitation learning, also called behavior cloning or learning from demonstration, is a form of supervised learning that can learn a policy from off-line collected data $D(S^*, A^*)$, where S^* indicates the real state of real world and A^* is the corresponding action (Schaal, Ijspeert, and Billard, 2003; Ross, Gordon, and Bagnell, 2011). This method is recently used for car-driving modeling. In (Codevilla et al., 2018), the authors presented an end-to-end vision-based driving model by supervised learning using deep neural networks, which is a model to predict the driving actions from camera input. However, using only real data is limited due to the lack of situation in the collected data-set.

Indeed, RL methods allow the agent to learn on-line by taking mistakes through the reinforcement signal from the external environment. The agent then tries to reinforce its action to improve future performance. Therefore, a signal can be formulated to guide the agent to learn from real data. As presented in (Y. Zhang et al., 2018; M. Zhu, Xuesong Wang, and Y. Wang, 2018), the authors used real data to learn a policy of car-following through RL. They associate a positive reward when the action is the same as in the data-set, for a state present in the collected data. Otherwise, the action is penalised with a negative reward.

Deriving from this idea and aiming at learning from both the simulation environment and the real lane changing decisions, we propose in this chapter an approach for lane changing modeling using the Q-learning algorithm (Watkins and Dayan, 1992) with an on-line imitation of the human lane-change decision from the NGSIM 101 data-set. In this work, our major contribution is the use of Q-learning algorithm, and the calibration of the lane changing decisions extracted from the NGSIM 101 data-set. In addition, we use the traffic vehicular simulator SUMO to create a numerical simulation environment, that allows us enriching the data-set for training the agent to find an optimal policy for lane change. The results show that the policy is sufficiently trained and closed to the real human behavior. In order to understand the learned policy, we use a color map visualisation to show the distribution of lane changes as functions of several couples of states.

4.3 A model for imitation of the real lane change behavior from the NGSIM dataset

As mentioned above, we propose in this study a lane changing model based on a Q-learning algorithm with an on-line calibration with human lane changing decision. With the purpose of obtaining a human-like and collision-free lane changing strategy, we apply this study based on the NGSIM 101 data-set. We use the vehicular simulator SUMO to build a traffic dynamic environment. In this section, we first introduce in sub-section 4.3.1, the architecture of our method. We then give the definitions of the state and the action variables for the lane changing model, in sub-section 4.3.3. We describe the NGSIM data-set as well as the preparation of the lane changing decision data in sub-section 4.3.4. In sub-section 4.3.5, we give the discretization of the state variables. Finally, we give in sub-section 4.3.6, the definition we use here for the reward, with the concept of learning the real lane changing decision.

4.3.1 Method architecture

We Illustrate in Fig.4.1 the architecture of our approach of lane changing modeling using Q-learning and online calibration with real data.

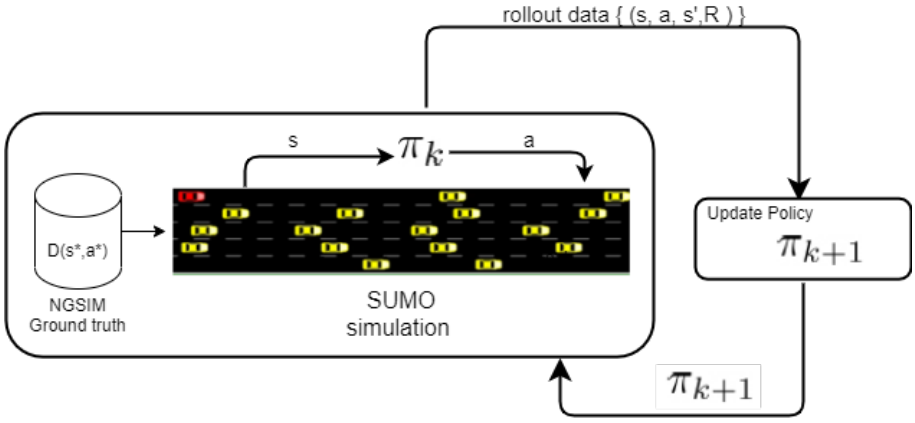


Figure 4.1 – Illustration of the lane changing model using reinforcement learning and online calibration with real data.

We use the vehicular traffic simulator SUMO ("Simulation of Urban MObility") (Lopez et al., 2018) to create a simulation environment. The controlled vehicle (the agent) exchanges with the simulated traffic environment and is trained to develop a lane changing strategy through evolving the policy by the return from the environment. In addition, real lane changing decisions extracted from the NGSIM data-set are used for online calibration of the target vehicle lane changing actions. As presented in Fig. 4.1, once a current policy π_k applies an action a

at a given state s, SUMO combining real data $D(S^*, A^*)$ returns a new state s' and a reward R. Then, with the data (s, a, s', R), the policy can be updated by the Q-learning algorithm to get a new policy (π_{k+1}). Then by iterating, the optimal policy can be found by trial-and-error with the environment. Referring to (Y. Zhang et al., 2018), where mimicking the real behavior has been studied, at a given state, which is present in the collected data, the reward could be positive when the action is the same as in the data; otherwise, the action needs to be penalised by a negative reward.

4.3.2 Q-learning

Q-learning (Watkins and Dayan, 1992) is an algorithm of RL, which is model-free and with the purpose to optimize the action-value function defined as $Q(s, a)$. This Q-value function expresses the expected return of taking action a following a policy π and being at state s:

$$\begin{aligned} Q_\pi(s_t, a_t) &= \mathbb{E}_\pi \left[G_t | s_0 = s, a_0 = a \right] \\ &= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k} | s_0 = s, a_0 = a \right], \end{aligned} \tag{4.1}$$

where $G_t, s_0, a_0, k, \gamma^k$ and R_t denote respectively ....

According to the Bellman principle, the optimal action value function $Q^*(s, a)$ is the maximum action-value function over all policies, which can be formulated as :

$$\begin{aligned} Q^*(s, a) &= \mathbb{E} \left[R + \gamma \max_{a'} Q^*(s', a') \right] \\ &= R_t + \gamma \, \mathbb{E} \left[\max_{a'} Q^*(s', a') \right]. \end{aligned} \tag{4.2}$$

In theory, the value of a state is relative to the values of its successor states. The value $Q(s, a)$ of the current state must equal the discounted $Q(s', a')$ of the expected next state, plus the immediate reward. Therefore, for the optimal $Q^*(a, s)$, the optimal policy is to select an action a', that maximizes the expected value of $Q^*(s', a')$. At the end, the optimal policy is attained by selecting at each state the action with the largest optimal Q-value :

$$\pi_*(s) = \arg\max_a Q^*(s, a) \tag{4.3}$$

Algorithm 4.1: Q-learning

initialize;
$Q(s, a)$, S, A, discount factor γ, learning rate α
Repeat (for each episode):
while *not converged* **do**
    choose a from s using policy derived from Q (e.g., ε-greedy or decaying-ε-greedy policy)
    take action a, observe R, s'
    $Q(s, a) \leftarrow Q(s, a) + \alpha[R + \gamma \max_{a'} Q(s', a') - Q(s, a)]$
    $S \leftarrow S'$
end

The Q-learning Algorithm 4.1 is an off-policy RL algorithm, in the sense that the learned policy can be different from the policy used to choose an action. The algorithm uses the Temporal Difference (TD) method, which learns by bootstrapping from current estimate, where $R + \gamma \max_{a'} Q(s', a')$ named TD target, is a target function for $Q(s, a)$ in the current estimate.

4.3.3 Definition of the state and action variables

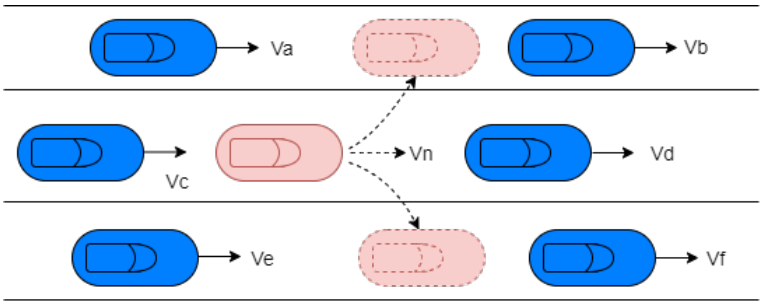


Figure 4.2 – The presentation of traffic environment for the control of the lane changing behavior of the target vehicle (in pink).

For the reinforcement learning, once we define the task that the agent needs to accomplish, the definition of the state, action and reward variables are important. The authors of (L. Wang et al., 2019; Hoel, Wolff, and Laine, 2018; P. Wang, H. Li, and Chan, 2019; Ye et al., 2020) have used different ways to represent the traffic including that of the target controlled vehicle and of its surrounding vehicles. We propose in our approach a new structure of the representation of the target vehicle (vehicle n) and of its surrounding vehicles (vehicles a, b, c, d, e, and f) as shown in Fig. 4.2. We consider here 7 state variables; see Figure 4.2 :

V_n: target vehicle speed

$\Delta V_{preceding}$: $V_n - V_d$

Dis_{ab}: inter-vehicular distance between a and b

Dis_{ef}: inter-vehicular distance between e and f

ΔV_{ab_n}: $(V_a + V_b)/2 - V_n$

ΔV_{ef_n}: $(V_e + V_f)/2 - V_n$

l_{info}: the current lane of target vehicle. 0: in the leftest lane; 1: in the rightest lane; 2 in the other intermediate lanes; 3: out of road.

In the case where the target vehicle is in the leftest (respectively, rightest) lane, then Dis_{ab} (respectively, Dis_{ef}) is set to zero, and the relative speed ΔV_{ab_n} (respectively, ΔV_{ef_n}) is set to $-V_n$.

The action is the lane change decision, which belongs to {0,1,2}: 0: do not change the lane, 1: change to the right lane and 2: change to the left lane.

4.3.4 The NGSIM 101 data-set

As we mentioned previously, a brief description of the NGSIM data-set has been given in the section 2.7.1. NGSIM 101 dataset consists of three files of sub-data-sets for 15 minutes traffic data and 1500-2000 vehicles in each 15-minute data. Therefore, we can extract all the lane changing trajectories in the three files of NGSIM 101 dataset. Indeed, we are interested in the trajectories of lane changing without a motivation of going to the off-ramp. With this objective, a pre-processing of data is done and we obtained all the available lane changing trajectories are presented in Fig.4.3.

For imitating the real lane changing behavior, the state variables at the moment that the driver decides to change lane need to be extracted. We estimate approximately the beginning of lane changing from vehicle trajectories as the moment that the driver decides to change lane. We use the same method as the one used in (Thiemann, Treiber, and Kesting, 2008) to capture the beginning of lane changing behavior and get the set of lane changing data (S^+, A^+). In the other side, other trajectories not in the lane-changing trajectories can be regarded as no change

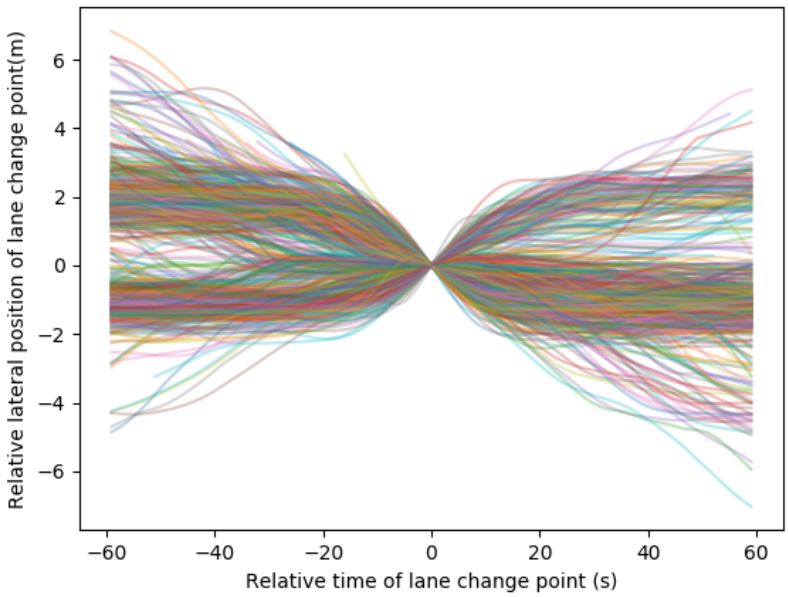


Figure 4.3 – Vehicle trajectories of lane changing in the NGSIM 101 data-set. The point (0s,0m) is the moment when vehicle changes the index of lane.

lane moments. We can get by that a set of no lane changing data (S^-, A^-). The data $D(S^*, A^*)$, where $(S^*, A^*) = (S^+, A^+) \cup (S^-, A^-)$, is prepared for learning real lane change decisions. In the collected data $D(S^*, A^*)$, we save the 7 state variables as well as the corresponding action, as defined above in section 4.3.3.

4.3.5 Discretization of the state Variables

The first 6 state variables are empirically discretized and the discretization is corresponding to the NGSIM data. We set each variable to 5 possible values, (0,1,2,3,4), to represent five sub-intervals, divided according to the quantiles 20%, 40%, 60%, and 80%; see Tab. 4.1. The five sub-intervals are $[0\%, 20\%[$, $[20\%, 40\%[$, $[40\%, 60\%[$, $[60\%, 80\%[$, $[80\%, 100\%]$.

In addition, the data $D(S^*, A^*)$ is discretized by the same methods, and after removing all duplicates in the NGSIM data, we obtain 13199 distinct couples of (s^*, a^*).

4.3.6 Learning lane change decision in NGSIM data-set

The reward estimates how good the agent performs an action in a current state. With the objective of imitating the real actions (those of the data-sets), we use the reward to define how close is the agent's behavior to the real driver behavior.

Table 4.1 – Quantile values in the NGSIM data for the first 6 state variables.

Variables	20%	40%	60%	80%
V_n(m/s)	8,43	10,75	12,51	15,08
$\Delta V_{preceding}$ (m/s)	-1.37	-0.5	0,06	0,94
Dis_{ab} (m)	6,61	17,04	24,11	33,84
Dis_{ef} (m)	10,65	19,23	26,31	36,78
ΔV_{ab} (m/s)	-5,88	-1,86	-0,432	0,84
ΔV_{ef}(m/s)	-3,49	-0,57	0,68	2,15

We denote the collected data $D(S^*, A^*)$, which contains a set of groups of (s^*, a^*). Regarding imitation learning, we look for a policy $\pi_\theta(s)$ which takes the same action a^* if $(s, a^*) \in D$. Therefore, the reward for an action a which follows the policy $\pi_\theta(s)$ and $s \in S^*, (s, a^*) \in D$ is defined as follows.

$$\text{reward } = \begin{cases} 1, & \text{if } \pi_\theta(s) \equiv a^* \\ \text{-1}, & \text{if } \pi_\theta(s) \neq a^* \end{cases} \tag{4.4}$$

However, we need to consider that the target vehicle can meet an unknown state s not present in the collected data. In this case, the action a^* taken by the closest $s^* \in S^*$ need to be applied. The distance between the SUMO simulation state and the real state in collected data is calculated using the Euclidean norm ($\|\cdot\|_2$), and the closest s^* is taken as the $\arg\min_{s^*} \|\mathbf{s}^* - \mathbf{s}\|_2$. The reward is defined as in (4.4).

In the other side, we give a punishment (negative reward) for the collision appearances and for out of road situations. For any action that causes a collision or an out-of-road situation, the reward is taken equal to $-c$, and this reward is defined as follows:

$$\text{reward } = -c \quad \text{if crash or out-of-road situation.} \tag{4.5}$$

where c is a positive constant. We set the reward in this case to $c = 50$ during the simulation experiment.

4.3.7 Simulation Experiment

We use the simulator SUMO to reproduce the NGSIM traffic after calibration. The reproduction of the NGSIM traffic data by numerical simulation consists in the creation of the road section, the selection of the microscopic traffic model (IDM car-following model, plus the default lane changing model in SUMO), and the model's parameters. We chose here the IDM car-

following model (Treiber, Hennecke, and Helbing, 2000), which is already implemented in SUMO. The IDM parameters are calibrated using the NGSIM vehicle's real trajectory with a genetic algorithm (Zhao et al., 2020). Besides that, the creation of each vehicle is provided by its longitudinal origin position, longitudinal destination position, entering lane, and entering time (in seconds). This necessary information is also extracted from NGSIM 101 data-set. All the details of the preparation of simulation of NGSIM traffic in SUMO are presented in the Section 3.7 of chapter 3.

Training setup

Using the SUMO's associated Python API, named *traffic control interface* (TraCI), we can access the vehicle sequential information in the environment, and execute the lane change decision with our model, in order to control the target vehicle lane changing dynamics. Regarding the training of our model, we choose only one target vehicle to control from its origin to its destination, in every episode of simulation. We deactivate completely the default lane changing model for the chosen target vehicle. Moreover, the target vehicle is controlled by the IDM for car-following, and by the Q-learning algorithm for lane changing. We control the target vehicle to make a decision every 1 second. For each lane change, the target vehicle can arrive the target lane in 1 second. The other vehicles are considered as traffic vehicles, and they are controlled by the IDM model for car-following and by the SUMO default lane changing model, for lane changing behavior. Model configuration for each vehicle is shown in Tab 4.2. Furthermore, The SUMO simulation frequency is set to 10Hz. The visualisation of traffic simulation in SUMO is shown in 4.4, where the red vehicle is the target one at each episode.

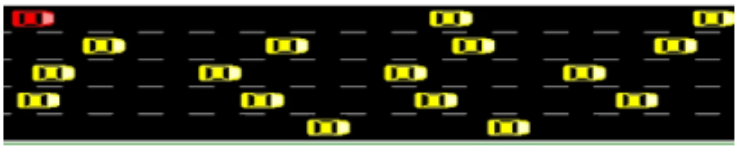
The setting up of Q-learning hyper-parameters is shown in Tab. 4.3. As we mentioned it in section 2.4.2, γ is a parameter which represents how important the future return for the immediate action. As we model here human behavior, which is not perfect to anticipate so far for the decision of an action, we take here $\gamma = 0.55$ that the impact of the decision making is approximate to zero after 3 steps (3 seconds) far away, because $0.55^3 \approx 0.16$ and $0.55^4 \approx 0.09$. Moreover, the epsilon-greedy policy provides the agent's exploration of all possible couples of (s, a). We use here a decaying-epsilon-greedy policy algorithm, where the ε is a decreasing sequence on the number of episodes (n): $\varepsilon_n = e^{-n/d}$, where d is a parameter, taken here $d = 3000$.

Table 4.2 – Model Configuration for each type vehicle

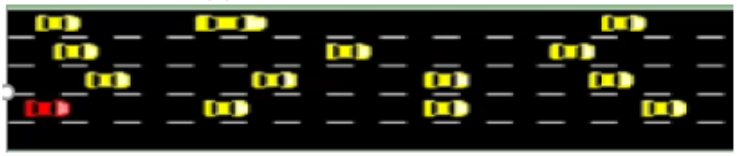
Behavior	Target vehicle	Traffic vehicles
CF model	IDM	IDM
LC decision	Q-learning	SUMO default model
Application of LC	1 second	1 second

Table 4.3 – Configuration of Hyper-parameters of Q-learning

Parameter name	Value
ε decaying parameter (d)	3000
Discount factor γ	0.55
Learning-rate α	0.05



(a) entry position for one vehicle



(b) entry position for another vehicle

Figure 4.4 – Screenshots of SUMO simulation. The red car is the target car in two different episodes

4.4 The results

4.4.1 The results for a first tested model without imitation from NGSIM Data

In the first proposition, we consider that the motivation of lane changing behavior is simply the gain in speed for the target vehicle. In this context, the reward of lane changing decision is the 'Speed gain', where the agent maximizes always the speed to take a decision of lane change. The states and actions have the same definitions as in section 4.3.3. Unlike what is presented in 4.3.6, the reward is defined as :

$$\text{reward} = \begin{cases} \text{Speed Gain } (v_{t+1} - v_t) \\ \text{- 10} \quad \text{if crash or out-of-road situation} \end{cases} \tag{4.6}$$

For this model, the results are shown in the Appendix B.2. The results of the evolution of the total reward of each episode are shown in Fig. B.1. The expectation of the total reward tends to converge after 40000 simulation episodes. Moreover, the crash and the out of road situations do not occur any more at the last episodes after about 41000; see Fig. B.2, where 1 means the presence of collisions or out-of-roads and 0 means no presence of the event. However, using this model, the number of lane changes for the vehicles in the last 200 episodes ranges from 1

to 20 times, which is shown in Figure B.3. This result seems unrealistic, since changing lane 20 times in a road of 600m is not really human-like.

4.4.2 The results for imitation of human lane change model

We have created a traffic simulation environment, based on the human driving data from the NGSIM 101 data-set, and have used the Q-learning algorithm proposed above in order to mimic the real lane changing decision of human drivers. The limitation of the real data and the lack of traffic situations prevents the supervisor learning approach to cover all driving situations. The simulation traffic environment completes the unknown situations in the collected data. In the NGSIM 101 data-set, we distinguish about 13000 different state-action couples, while by numerical simulation, the vehicles have visited more than 16000 different couples of (s, a). We choose different cars in traffic as the target one at each episode, among randomly 800 chosen cars. The result of the evolution of the total reward of each episode is shown in Fig. 4.5. The expectation of the total reward tends to converge after 12000 simulation episodes. Moreover, the crash and the out of road situations do not occur any more at the last episodes; see Fig. 4.6, where 1 means the presence of collisions or out-of-roads and 0 means no presence such events.

Figure 4.5 – Total reward evolution. Blue line is the total reward for each episode. Orange line presents the average for each 20 episodes

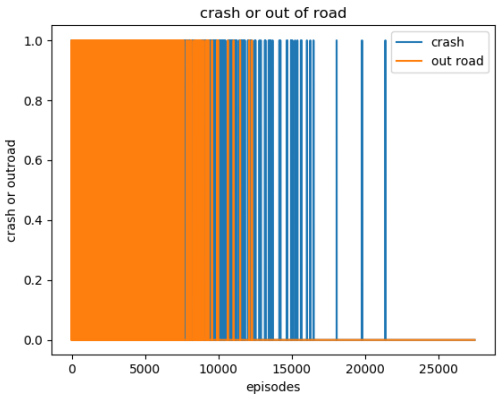


Figure 4.6 – Crash and out of road occurrence evolution. 1 means presence of the event, 0 means no presence of the event

From the last 800 episodes, the model has been already converged (Fig 4.5) and has non presence of neither collision nor out-of-road (Fig 4.6). Moreover, the distribution of the number of lane changes for the last 1000 vehicles is provided in Figure 4.7. From this result, the number of lane changes is most realistic. In order to measure the accuracy of lane change behavior imitation, see Tab. 4.4, we collected all visited states of vehicles in the last 800 episodes, and we found that 87.09% of the states are present in the real data, while 12.23% of the states have the closest sate in the collected data with a distance of 1, as defined in sub-section 4.3.6. In

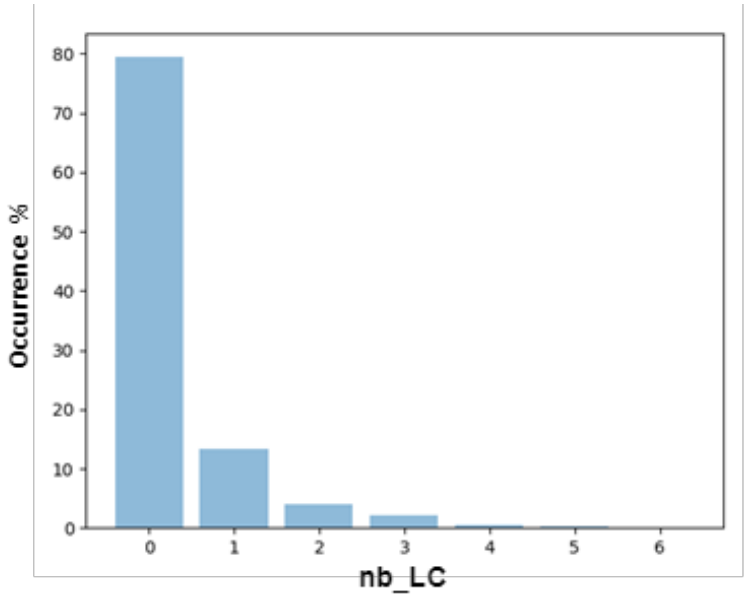


Figure 4.7 – Distribution of number of lane changes for the vehicles in the last 1000 episodes

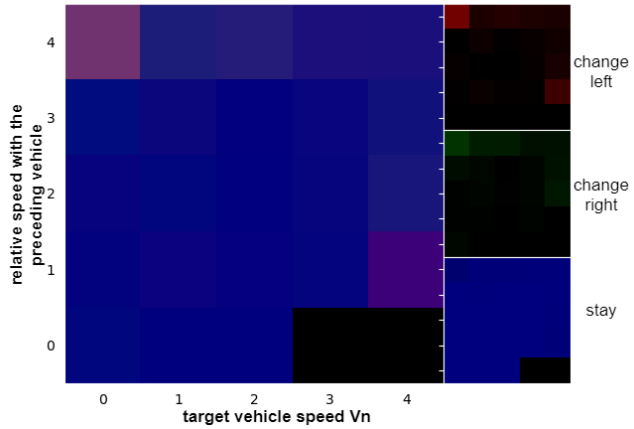
addition, 95.37% of the taken actions are present in the real data-set; see Tab. 4.4. This accuracy of imitation of real human behavior shows that our lane changing model is human-like.

Table 4.4 – Comparison of the real and simulated states and actions

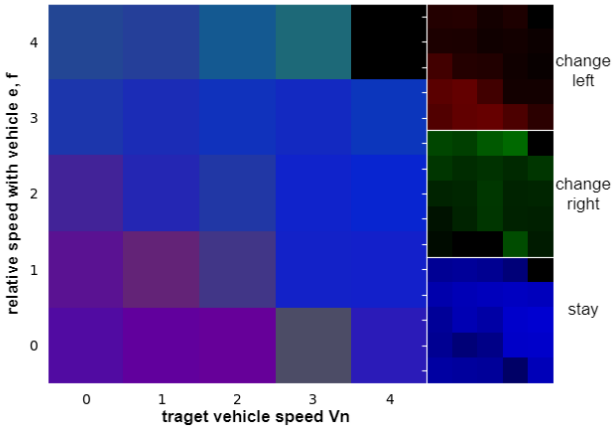
simulation states present in the data-set	87.09%
simulation states with distance of 1 to the data-set	12.23%
simulation states with distance of 2 to the data-set	0.80%
simulation actions present in the data-set	95.37%
simulation actions not present in the data-set	4.63%

In order to illustrate the learned policy, and to understand its logic, we plot it as functions of several couples of states: $a = f(Vn, \Delta V_{preceding})$, $a = f(Vn, \Delta V_{ef\_n})$, $a = f(Vn, Dis_{ef})$ and $a = f(Dis_{ab}, Dis_{ef})$ illustrated with color maps in Fig. 4.8-(A), (B), (C), and (D) respectively. We recall that we have 5 discrete values for each state variable. We use RGB color to represent the action (change left (in red), change right (in green) and stay on lane (in blue). The grey level show the probability of each action in each case. The black cases are the states never visited. In addition, we have 4 views in each figure: the three views on the right side show each action respectively for change left, change right, and stay on current lane; plus a main view with combination of all actions.

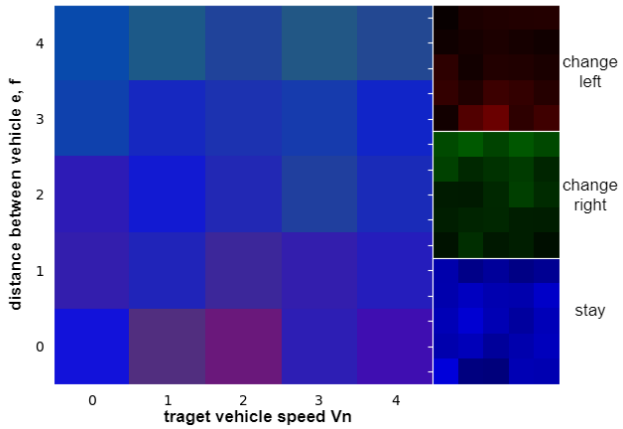
In Fig. 4.8-(A), the x-axis represents the speed of the target vehicle, while the y-axis represents the relative speed of the target vehicle, with respect to the preceding vehicle. We can see from this figure that the controlled vehicle tends to maneuver a lane change in the cases when



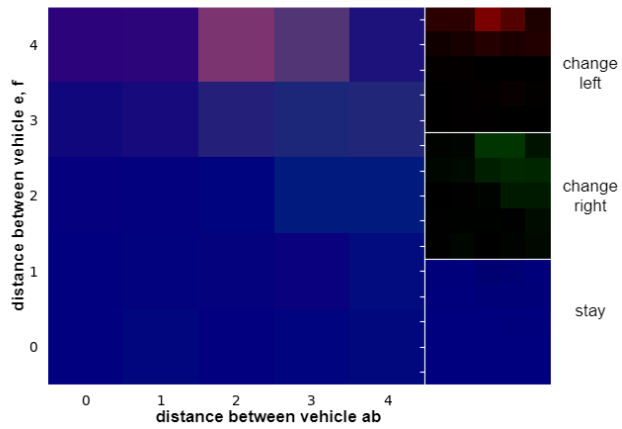
(A) $a = f(Vn, \Delta V_{preceding})$



(B) $a = f(Vn, \Delta V_{ef\_n})$



(C) $a = f(Vn, Dis_{ef})$



(D) $a = f(Dis_{ab}, Dis_{ef})$

Figure 4.8 – Color maps for the learned policy. red: change left, green: change right, blue: stay in the current lane.

the preceding vehicle obstructs it, i. e. when the controlled vehicle's speed is small and the relative speed with the preceding vehicle ($\Delta V_{preceding}$) is big (see cases (0,4), (1,4), (2,4) in Fig. 4.8-(A). Lane changes are less observed for high speeds of the controlled vehicle.

From Fig. 4.8-(B), we observe that the agent tends to change to the left lane for low values of ΔV_{ef_n}, and to the right lane for high values of ΔV_{ef_n}. In other terms, the agent decides to go to the lane with a high relative speed, i. e. where the controlled vehicle's speed will increase significantly. Moreover, it seems that there is a curve $\Delta V_{ef\_n} = 3 - V_n$, on Fig. 4.8-(B), under which the agent tends to change to the left lane, and over which it tends to stay on its current lane. We do not yet fully understood this observation. On Fig. 4.8-(C), we can see that the vehicle tends to change right when the inter-vehicular distance with respect to vehicles e and f, is sufficiently large. On Fig. 4.8-(D), it is obvious that lane changes are possible only when at least one of the inter-vehicular distances on the left and right lanes is sufficiently large. All these observations of learned strategy of lane changing are interesting to understand and to validate this lane changing model using RL.

4.5 Conclusion

In this work, we proposed a vehicular lane change model using Q-learning with on-line calibration on the real human lane change decision data. It is a data-based model, which aims to mimic the real lane change decisions taken by human drivers. We applied this study based on the NGSIM 101 data-set and used the vehicular simulator SUMO to complete the data and to cover all the state and action variable spaces. A big number of episodes have been considered for the training, during each of them the controlled vehicle (the agent) executes our Q-learning algorithm to take decisions on its lane change, where the other vehicles run under traffic models included in SUMO (car-following and lane-change). Moreover, our Q-learning algorithm permits to complete the unobserved situations, and to propose actions for the states not present in real data. Thus, about 13% additional traffic situations, not present in the real data, are created by the traffic simulation environment.

We have shown that with the approach proposed in this article we can mimic and reproduce real lane change decisions observed in traffic data, up to a rate of 95%. Finally, we have illustrated the obtained optimal policy by means of color maps, and discuss preliminary conclusions on the determinant states and thresholds on states in the decision process of lane changing. For future research, we aim at understanding more about this lane change model, and trying to test this model in other road traffic conditions.

Chapter 5

Long Short-Time Memory Neural Networks for Human Driving Behavior Modelling

In this chapter, a long short-term memory (LSTM) neural network model is proposed to replicate simultaneously car-following and lane-changing behaviors in road networks. By combining two kinds of LSTM layers and three input designs of the neural network, six variants of the LSTM model have been created. These models were trained and tested on the NGSIM 101 dataset, and the results were evaluated in terms of longitudinal speed and lateral position respectively. Then, we compared the LSTM model with a classical car-following model (the Intelligent Driving Model (IDM)) in the part of speed decision. In addition, the LSTM model is compared with a model using classical neural networks. After the comparison, the LSTM model demonstrates higher accuracy than the physical model IDM in terms of car-following behavior and displays better performance with regard to both car-following and lane-changing behavior compared to the classical neural network model.

Contents

5.1 Introduction

In the last decades, deep learning methods, a specific set of machine learning methods designed to build models based on sample data, have shown excellent performance in many fields (e.g. natural language processing, image processing and financial trading strategies). In the field of microscopic road traffic, where researchers are interested in modeling individual vehicle movements on road networks, the use of deep learning is also becoming increasingly popular.

As well known, two driving behaviors are generally distinguished in the microscopic traffic modeling: Car-Following (CF) and Lane Change (LC), which describe respectively longitudinal and lateral vehicular movements. In the literature review, one of the first car-following models is the GHR model (Chandler, Herman, and Montroll, 1958). Among the most famous models, we cite the Wiedemann model (Mohammad and Z. Zheng, 2014), Gipps model (Gipps, 1981), Krauss model (Krauß, 1998) and Intelligent driver model(IDM) (Treiber, Hennecke, and Helbing, 2000). Some reviews of car-following models can be found in (Brackstone and Mark, 1999; Mohammad and Z. Zheng, 2014; Aghabayk, Sarvi, and Young, 2015). In addition, in order to model the imperfect human driver, the authors of (Treiber, Kesting, and Helbing, 2006; Lindorfer, Mecklenbraeuker, and Ostermayer, 2018) carried out some extensions based on car-following models, to describe mathematically for the reaction time, the anticipation, and estimation errors of human drivers. For lane change models, we cite (Toledo, Koutsopoulos, and M. Ben-Akiva, 2007; Moridpour, Sarvi, and Rose, 2010; Z. Zheng, 2014) as reviews, authors have shown that the lane change decision is usually taken based on several sub-models of the driver's lane change motivation (e.g. mandatory LC and discretionary LC; or free, cooperative, and forced LC). We can observe that, compared to CF models, which are well developed, LC models are more complex and rule-based with many hand-crafted parameters. LC models generally show limited performance and are hard to calibrate, and to catch with human lane-changing behavior (Z. Zheng, 2014).

On the other hand, a number of data-driven CF and LC models using machine learning methods have been developed (Q. Wang, Z. Li, and L. Li, 2014; Xie et al., 2019; Xiao Wang et al., 2017). It has been demonstrated that, comparing with physical models, the machine learning based ones, including deep learning methods, provide a better performance for the accuracy of trajectories, and for the replication of traffic flow characteristics.

Long short-Term Memory (LSTM) neural networks (Hochreiter and Schmidhuber, 1997) are recurrent neural networks (RNNs), able to learn time series with keeping a memory of previous sequences (Gers, Schmidhuber, and Cummins, 1999). Several studies have been applied using LSTM for human driving behavior modeling or human driving forecasting, such as in (Xie et al., 2019), where the authors developed a LC model (the LC decision prediction combining the LC execution trajectory). In the latter model (Xie et al., 2019), the execution of the lane change decisions is formulated based on a LSTM neural network. In (Y. Zhang et al.,

2018), the authors proposed a simultaneous CF and CL model using a LSTM neural network, and with a constraint on the car time headway, which permits to retrain the neural network in the case where the time headway is not satisfied. Moreover, in (Altché and La Fortelle, 2017), the authors proposed a LSTM model for vehicle trajectory prediction from 1s to 10s in the future. All these works show that LSTM modeling is essentially useful for the prediction of human driving behavior. However, different studies used different model structures, with different input designs for the neural networks.

In this chapter, we propose a LSTM model to learn human driving CF and LC behavior simultaneously. Our model is based on the relevant works in (Y. Zhang et al., 2018; Altché and La Fortelle, 2017), where the sequential decision for longitudinal (speed and acceleration) and lateral (position or lane number) movements are taken simultaneously. We propose two different designs for the model architecture, using neural networks with LSTM layer, and three designs for the input variables. By studying different combinations of the model architectures and the input designs, we aim at discovering the best combination of architecture and input design. The models are trained and tested on the NGSIM 101 dataset (Alexiadis et al., 2004). The model performance is evaluated in terms of prediction accuracy on longitudinal speed and lateral position. The obtained result is compared with a physical car-following model, the Intelligent Driver Model (IDM) (Treiber, Hennecke, and Helbing, 2000), and with another model using classical neural networks.

5.2 Problem statement

In the state-of-the-art, we can find that Car-following and Lane Change behaviors are usually considered and modelled independently. However, in the real world, human drivers can hardly split the driving task into two independently tasks (Y. Zhang et al., 2018). We propose in this work to model simultaneously CF and LC decisions using a LSTM neural network. The experiment is applied on the NGSIM 101 dataset. Our model is based on an existent work presented in (Altché and La Fortelle, 2017).

The prediction of the driving decision making (e.g. speed, position, etc.) can be considered as a regression problem by LSTM architecture, where the vehicle's present behavior is related to current traffic conditions. The latter conditions consist of all the information of surrounding vehicles (presented by their own trajectory), and also of the vehicles' own previous behavior in the passed time. We denote by $\mathcal{X}$ the space of observable features, and by $\mathcal{O}$ the motion prediction outputs for the target vehicle. With this assumption, for LSTM neural networks, the input is a sequence of history observations ($X = (x_1, x_2, ...x_k)$). $x \in \mathcal{X}$ denotes the information on the target vehicle history as well as the history information of its surrounding vehicles at each time step. The variable k is the length of time series in the input. The output of the neural

network is the prediction of the target vehicle motion $\hat{Y} \in \mathcal{O}$. We aim at using supervised learning approach and train a regression function f such that the predicted outputs $\hat{Y} = f(X)$ match the actual values (real data) as closely as possible.

As in (Altché and La Fortelle, 2017), we use here the Keras deep learning framework, where an extended LSTM model has been implemented. The algorithm is described in (Gers, Schmidhuber, and Cummins, 1999). The models are trained and tested on the NGSIM 101 dataset (Alexiadis et al., 2004), which is an open highway traffic dataset containing about 6000 vehicle trajectories. In this work, we evaluate totally two different model architectures combining three different input designs. Among the different combinations, we aim to discover the most efficient model.

In the following sections, we will present the NGSIM 101 dataset (section 4.3.4). Then, we will describe the model structure, including the models description, the proposition of the different designs for input variables, and the definition of output variables (section 5.4). In order to show the result with comparison with a physical traffic model (Intelligent driver model (IDM) (Treiber, Hennecke, and Helbing, 2000)), we will demonstrate the calibration method of the IDM model in the section 5.5. At the end, we will illustrate the obtained results in section 5.6.

5.3 NGSIM 101 data-set and Data preparation

As we mentioned previously, a brief description of the NGSIM data-set has been given in the section 2.7.1.

In our study, we focused only on car trajectories in the five main lanes. In addition, we are only interested in discretionary lane changes, which are independent on entering and leaving the road. The authors of (Altché and La Fortelle, 2017) observed that the data is noised for the vehicle's position, velocity, and acceleration signals. Therefore, in this work, we firstly pre-processed the data-set using the same method as in (Altché and La Fortelle, 2017), where we apply a first order Savitzky-Golay filter (Savitzky and Golay, 1964) to smooth each vehicle longitudinal and lateral position, and then we recompute their corresponding velocities and accelerations.

5.4 Model structure

The model we propose in this chapter is based on an existent work presented in Altché and La Fortelle, 2017, where a LSTM neural network is used to predict the future car trajectories from 1s to 10s. We investigate here different neural network architectures with different input designs,

for both longitudinal and lateral vehicle movements. We use the neural network structure in Altché and La Fortelle, 2017 as the reference network, as shown in Fig. 5.1. Furthermore, we propose a second architecture using two layers of LSTM, instead of one layer; see Fig. 5.2.

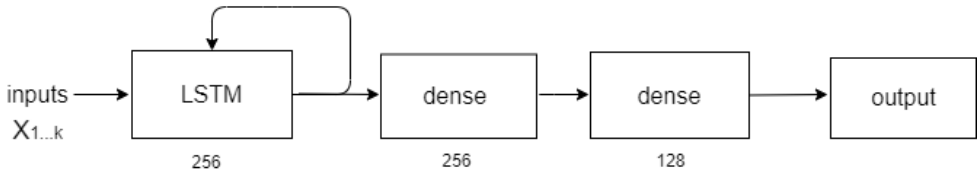


Figure 5.1 – M1: The reference architecture (Altché and La Fortelle, 2017) uses a first layer of 256 LSTM cells, followed by two dense (fully connected) hidden layers of 256 and 128 neurons respectively and a final dense output layer containing as many neurons as the number of outputs.

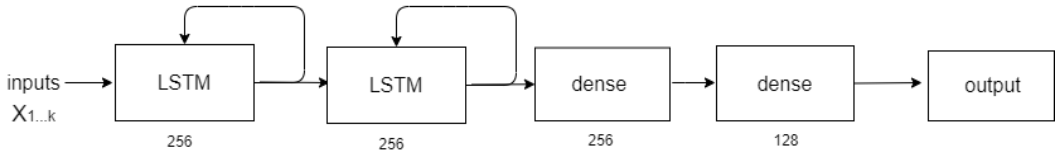


Figure 5.2 – M2: Neural network with two layers of LSTM

In addition, in order to compare with the models with LSTM, we proposed use another two structures with only fully connected neural networks (see. Fig. 5.3 and Fig. 5.4).

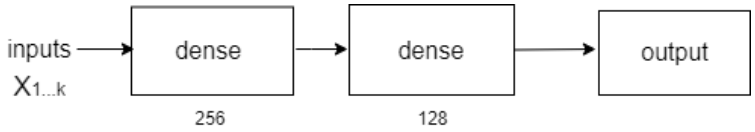


Figure 5.3 – M_{NN}: Neural network without LSTM

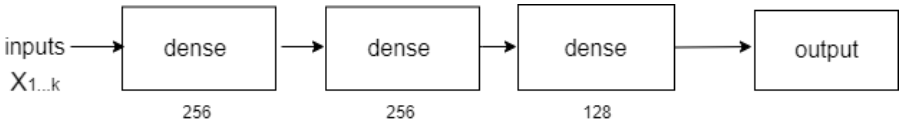


Figure 5.4 – M_{NN2}: Neural network without LSTM

5.4.1 Input variables

Several previous works have been performed regarding the use of deep neural networks to model human driving behavior (Altché and La Fortelle, 2017; X. Zhang et al., 2019). Different designs of the input variables of the models have been proposed. In (Altché and La Fortelle,

2017), the authors have varied the selected surrounding vehicles, where they tested two scenarios: one with vehicles (a, b ,c, d, e, f) and the other one with vehicles (a, b, c, d, e, f, s); see Fig. 5.5. They found that the results without vehicle s are better than the ones with vehicle s. Based on this result, we decided to consider in this work only (a,b,c,d,e,f) as surrounding vehicles.

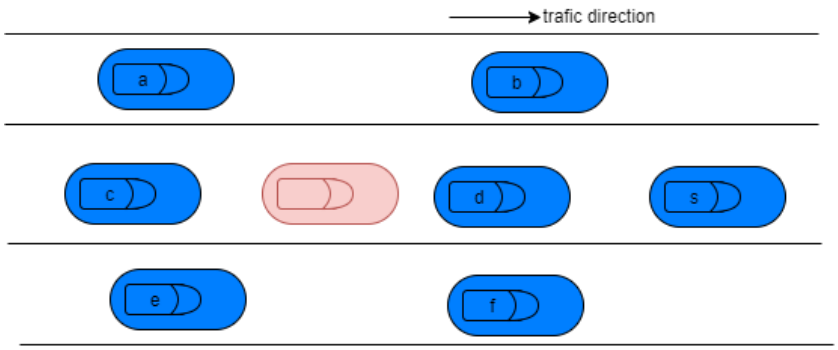


Figure 5.5 – Considered vehicles in neural network inputs, the red vehicle is the target vehicle to predict its longitudinal speed and lateral position

For the input variables, the authors of (X. Zhang et al., 2019) considered only one input variable which is the *vehicle positions* of the surrounding vehicles. In order to better understand the effect of different input variables on the human driving behavior using LSTM neural network, we consider in this article different variables. Moreover, we propose three different input designs, see Tab. 6.2. The considered input variables are defined as follows.

1. Input design 1: in this design, we apply all possible input variables for the vehicle movements, including its longitudinal position, lateral position, speed, acceleration and vehicle size. For the surrounding vehicles, we consider position, speed, acceleration, as well as the relative distance and speed with respect to the target vehicle. We notice that the variable time-to-collision (TTC) has also been considered as an input variable in (Altché and La Fortelle, 2017). which we do not consider here, because we think that this information is redundant with the two variables of relative speed and distance.

   The following input variables are considered for the target vehicle (red vehicle in Fig. 5.5):

   - longitudinal position x_{target} and lateral position y_{target}
   - longitudinal speed v_{target} and longitudinal acceleration a_{target}
   - vehicle length l_{target} and vehicle width w_{target}

   The following input variables are considered for the surrounding vehicles ((a,b,c,d,e,f) in Fig. 5.5):

   - longitudinal position x_i and lateral position y_i

- longitudinal speed v_i and longitudinal acceleration a_i
- vehicle length l_i and vehicle width w_i
- relative speed $\Delta v_i = v_{target} - v_i$
- longitudinal distance $s_{long\_i} = x_{target} - x_i$
- lateral distance $s_{lat\_i} = y_{target} - y_i$

For the input design 1, we have a total of 60 input variables of the model.

2. Input design 2: in the second input design, we are interested in finding the performance influence by using different inputs. Comparing with the first design, we consider here that the relative variables with respect to the target vehicle are more important than the absolute ones. The acceleration is ignored in this design to avoid redundancy. In addition, we defined a new variable for noticing the presence of surrounding vehicle i, named as presence p_i (0 for non presence, and 1 for presence). If vehicle i is not present, all the relevant value $(l_i, w_i, \Delta v_i, s_{long\_i}, s_{lat\_i})$ for this vehicle are set to 0. For the input design 2, we have a total of 41 input variables of the model.

3. Input design 3: In this design we take all the input variables of design 1 and design 2. Similarly, if a vehicle i is not present, then all the relevant value $(x_i, y_i, v_i, a_i l_i, w_i, \Delta v_i$ $, s_{long\_i}, s_{lat\_i})$ for this vehicle are set to zero. For the input design 3, we have a total of 66 input variables of the model.

	x_{target}	y_{target}	v_{target}	a_{target}	l_{target}	w_{target}	x_i	y_i	v_i	a_i	l_i	w_i	Δv_i	s_{long_i}	s_{lat_i}	p_i
input design 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
input design 2	✓	✓	✓		✓	✓					✓	✓	✓	✓	✓	✓
input design 3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 5.1 – input designs

5.4.2 Output variables

As described in the sections before, our problem is set to learn the decision of human driving behavior using LSTM neural networks by prediction of car-following and lane changing behavior, with the condition of keeping a memory of vehicle previous action information and the previous motion information, as well as the current motion information of surrounding vehicles. We choose to use the longitudinal speed and lateral position as output, because for the longitudinal positions, the values can become quite large compared to the lateral positions. As mentioned in section 5.4.1, for the inputs, we use k_{hist} sequences information in the history as inputs. Therefore, we define the two outputs of the model to control the vehicle in real

time, which are $v_{target}(k_{hist}+1)$ and $x_{target}(k_{hist}+1)$. The index k_{hist} goes up to 50 in our experiments.

5.4.3 Training and Test experiments

The NGSIM 101 dataset is separated into 3 sub-datasets, and each sub-dataset corresponds to 15-minute traffic data. We train the neural network using the first 15 minutes data, and test separately with the second and third 15-min sub-datasets. In practice, to train the neural network, epochs and mini-batches are defined for the weights updating process. A mini-batch refers to a subset (m individual samples) of the whole training set, to compute a partial gradient and update the weights accordingly. An epoch refers to processing the whole training set once. We used the Adam algorithm (Kingma and Ba, 2014), which is a stochastic gradient algorithm and has been implemented in Keras framework. For the loss function of neural network, we used the Mean Square Error (MSE). The learning rate is another parameter, which determines how big a step is taken toward the descent direction to the minimum of loss function when updating the weights in neural networks. Therefore, regarding hyper-parameters for training the neural networks, the configuration is shown in Table 5.2.

Parameter name	Value
batch size	128
number of epochs	20
learning rate	0.01

Table 5.2 – Configuration of some Hyper-parameters

5.5 Intelligent driver model (IDM)

We propose to compare our LSTM model with an existing physical car-following model. In this purpose, we choose the Intelligent driver model (IDM) (Treiber, Hennecke, and Helbing, 2000). The IDM model is a classical car-following model used in various microscopic traffic simulators, such as SUMO (Krajzewicz et al., 2012), VISSIM (Fellendorf and Vortisch, 2010b), and AIMSUN (Casas et al., 2010b). Authors in (M. Zhu, Xuesong Wang, Tarko, et al., 2018; Pourabdollah et al., 2017) found that the IDM model is one of the most human-like car-following models. There are several methods for calibrating microscopic models. We use in this paper the same method as in (M. Zhu, Xuesong Wang, Tarko, et al., 2018), where a genetic algorithm is applied for each driver data to find the optimal IDM parameters. The objective function used to calibrate the IDM model is the root mean square error (RMSE) of the vehicle speed. The

IDM model is written as in 3.18, and the parameters considered to be calibrated are pre-defined in 3.6.1.

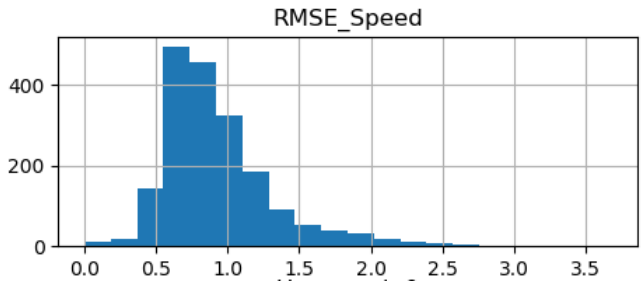


Figure 5.6 – RMSE distribution on the car-speed variable from the calibration of the IDM model. The RMSE value is presented for each driver data in the first 15 minutes of the NGSIM 101 data-set.

As the result, we give in Figure 5.6 the distribution of the RMSE on the car-speed variable, obtained from the calibration of the IDM model. The RMSE value is presented by every driver data in the first 15-minutes of the NGSIM 101 data-set (about 1900 vehicles). We notice that from Figure 5.6, the RMSE of calibrated IDM model for each driver ranges (mostly) from about 0.5 to 1.5 (m/s).

5.6 Results

We present in this section the results of our proposed LSTM model by comparison with the IDM model for speed decision, and with the model of classical neural networks, for both longitudinal speed and lateral position.

Firstly, Fig. 5.7 gives the loss evolution of our proposed LSTM models in the training phase, for different input designs. The loss value presented here is the default Keras mean error (MSE) between label and prediction. In the case of having two outputs, the error is the average MSE of the two outputs. Fig. 5.7 shows the training loss during 20 epochs. We can see that for all the different models, the loss decreases and converges. We can also infer from Fig. 5.7 that the proposed LSTM models with input design 2 give the best results in terms of the MSE evolution in training phase.

In order to compare the results of all models, we use here a metric denoted by '*Mean driver RMSE*' which calculates the average error between the real data and the predicted value from the outputs of each model, for every driver in the data-set. It is defined as follows :

$$Mean\ driver\ RMSE = \frac{1}{D}\sum_{i=1}^{D}\sqrt{\frac{1}{n_i}\sum_{t=1}^{n_i}(y_t - \hat{y}_t)^2} \tag{5.1}$$

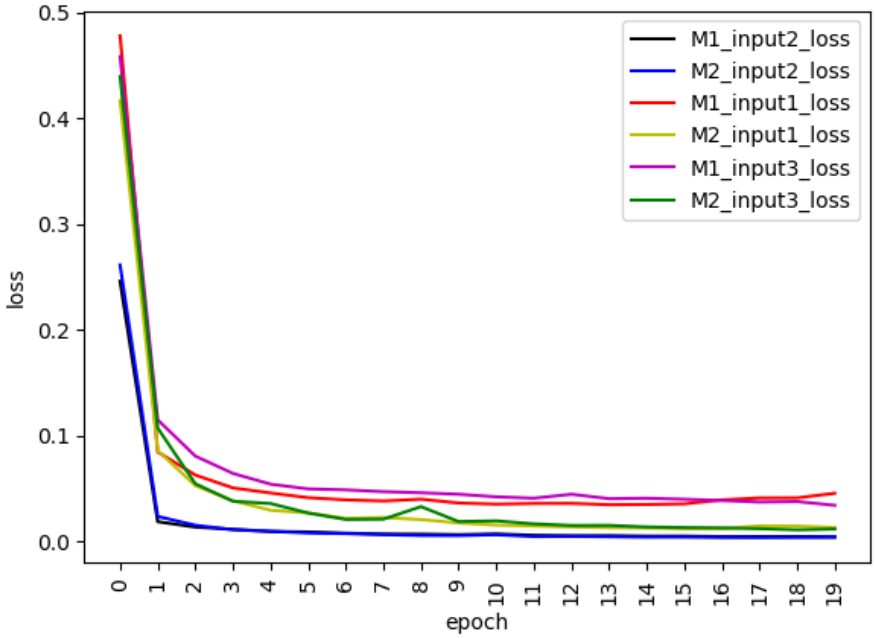


Figure 5.7 – The evolution of loss during the training for different models using different input designs.

where D is the number of drivers in each sub-dataset; n_i is the total number of sequences of the vehicle trajectory, in every sub-dataset i; y_t is the value of the considered variable in the real data; and $\hat{y}_t$ is the output of the model for the considered variable (car-position, car-speed, etc.). Here, the output of the IDM model is the longitudinal speed, while for LSTM model, the outputs are the longitudinal speed and the lateral position. The '*Mean driver RMSE*' on the car-speed variable for the IDM model is 0.9263 (see the RMSE distribution in Fig. 5.6). This error is far from the accuracy provided by the LSTM models, as it is shown below.

Mean-driver-RMSE	Train		Test1		Test2	
	Speed (m/s)	lateral Position (m)	Speed (m/s)	lateral Position (m)	Speed (m/s)	lateral Position (m)
M1_input1	0.2819	0.1308	0.27	0.14	0.294	0.1389
M2_input1	0.1524	0.1608	0.1767	0.167	0.1964	0.1709
M1_input2	**0.056**	**0.024**	**0.05**	**0.024**	**0.051**	**0.023**
M2_input2	0.0686	0.0848	0.0688	0.0752	0.0695	0.0669
M1_input3	0.4188	0.1185	0.4356	0.1353	0.4155	0.1494
M2_input3	0.2465	0.1331	0.3248	0.1305	0.3021	0.1396
M_{NN}_input2	0.2495	0.1266	0.2565	0.1229	0.2749	0.1209
M_{NN2}_input2	0.1423	0.1338	0.1389	0.16415	0.1503	0.1802

Table 5.3 – Mean RMSE for each different model using different input design and for training, test1, test2 respectively.

The '*Mean driver RMSE*' for the different models in terms of longitudinal velocity and lateral position is given in Table 5.3. These results in Table 5.3 show that M1_design_2 provides a best performance among all presented models, thus, the accuracy both in longitudinal velocity and lateral position is the best, as well as for training and tests. Models using input design 2 have

better performance than models using other input designs. In addition, models using LSTM are better than models using only classical neural networks (comparing M1_input2, M2_input2 and M_{NN}_input2, M_{NN2}_input2 in Table 5.3). In Fig. 5.8, we show some trajectory prediction results for the LSTM model (model 1 with input design 2: M1input2) during the training phase. We show the longitudinal velocity and lateral position predictions for two drivers. In Fig. 5.9, we show some trajectory prediction results of the LSTM model (model 1 with input design 2: M1input2) in the test phase for the other two drivers. Fig. 5.8 and Fig. 5.9 show the high accuracy achieved by the M1input2 model.

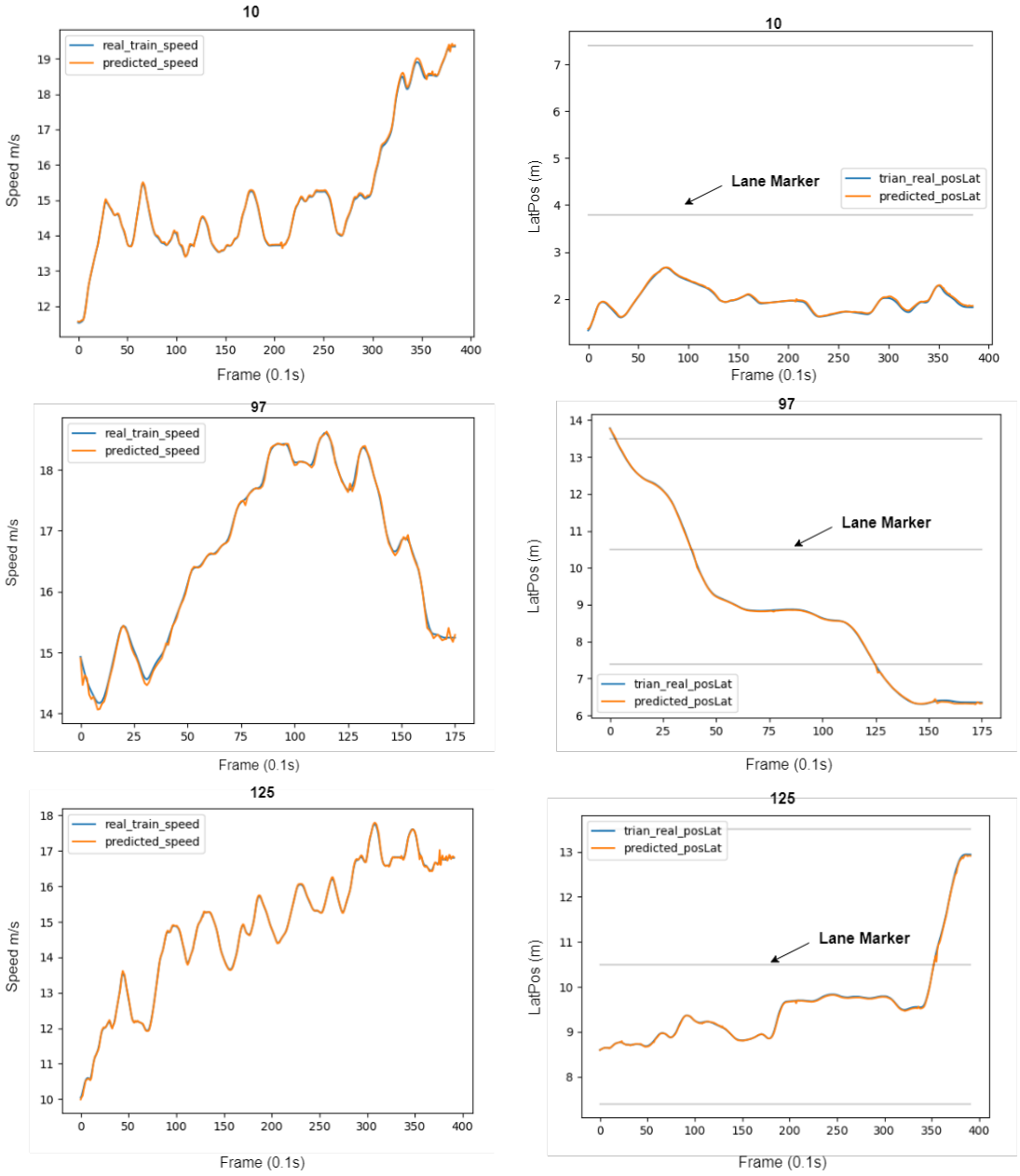


Figure 5.8 – Result of M1_input2 model for the speed and lateral position respectively for the drivers 10, 97, 125 in training data-set.

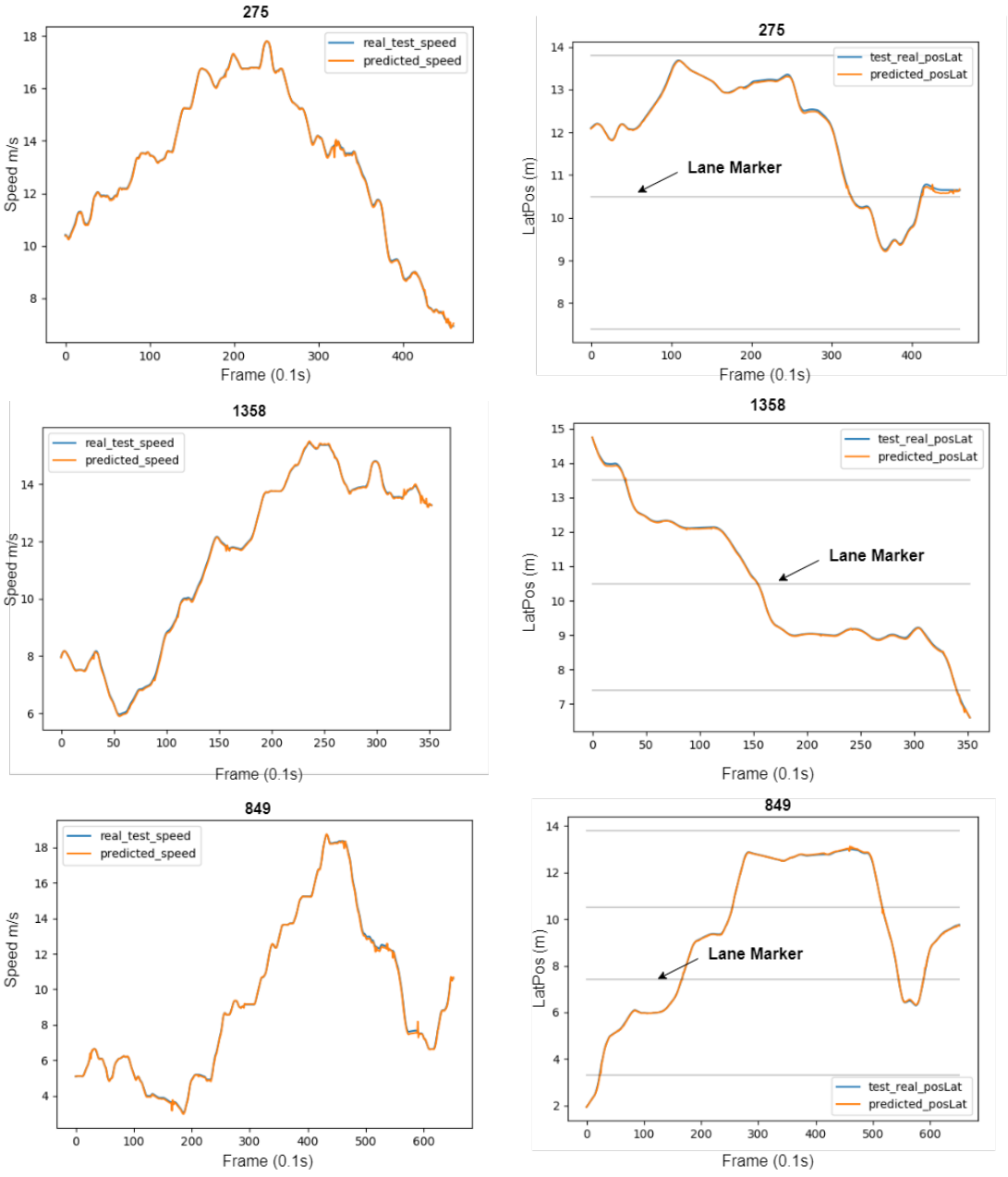


Figure 5.9 – Result of M1_input2 model for the speed and lateral position respectively for the drivers 275, 1358 in the test data-set and the driver 849 in test2 data-set.

5.7 Model validation using HighD dataset

As we presented above, the result of different models based on NGSIM 101 dataset show the M2_input2 has a best performance among all the tested models. In this section, I will present the result of all proposed models using another dataset, HighD (Krajewski et al., 2018).

5.7.1 HighD dataset

As what I presented in the previous chapter, in the Section 2.7.2, a brief presentation of HighD dataset is given. As the same application using NGSIM 101 dataset, we choose a part of HighD dataset and separated it into one part for training, one part for first test, and one part for the second test. HighD dataset is larger than NGSIM 101 dataset. It contains 60 files of registration available in the whole HighD dataset and about 2000 vehicle-trajectories in each file. The six recorded roads are all straight roads with two directions and each direction has two or three lanes.

In the experiment for the validation of the proposed models by HighD dataset, among 60 files availble in HighD dataset, we chose randomly 10 files within about 20000 vehicle trajectories as the training dataset, 5 different files within about 10000 vehicle trajectories as the dataset in test1 and another 5 files with about 10000 vehicle trajectories as the dataset in test2. First, we obtained the result of the evolution of loss value in the training phase (see. Figure 5.10). As the same, the loss value presented here is the default Keras mean error (MSE) between real data and prediction. In our case, the outputs contains the prediction of speed and the prediction of lateral position, the loss value is the average MSE of these two outputs. From the Figure 5.10, it shows the similar result that we trained the model using NGSIM dataset, where the models using second input design (M1_input2 and M2_input2) have better performance than other models using other type of input designs. Furthermore, they are better in terms of results in training result, Test1 and also Test2 (see. Table 5.4).

Mean-driver-RMSE	Train		Test1		Test2	
	Speed (m/s)	lateral Position (m)	Speed (m/s)	lateral Position (m)	Speed (m/s)	lateral Position (m)
M1_input1	0.071583	0.044184	0.07242	0.043247	0.06659	0.046756
M2_input1	0.037969	0.037736	0.037002	0.039664	0.040165	0.042509
M1_input2	**0.010865**	**0.016171**	**0.010465**	**0.016323**	**0.011431**	**0.015859**
M2_input2	0.033533	0.019365	0.031219	0.019708	0.036982	0.020328
M1_input3	0.209486	0.038016	0.189182	0.034586	0.187857	0.035655
M2_input3	0.034134	0.021585	0.032075	0.022203	0.035832	0.021433

Table 5.4 – HIGHD dataset : Mean RMSE for each different model using different input design and for training, test1, test2 respectively.

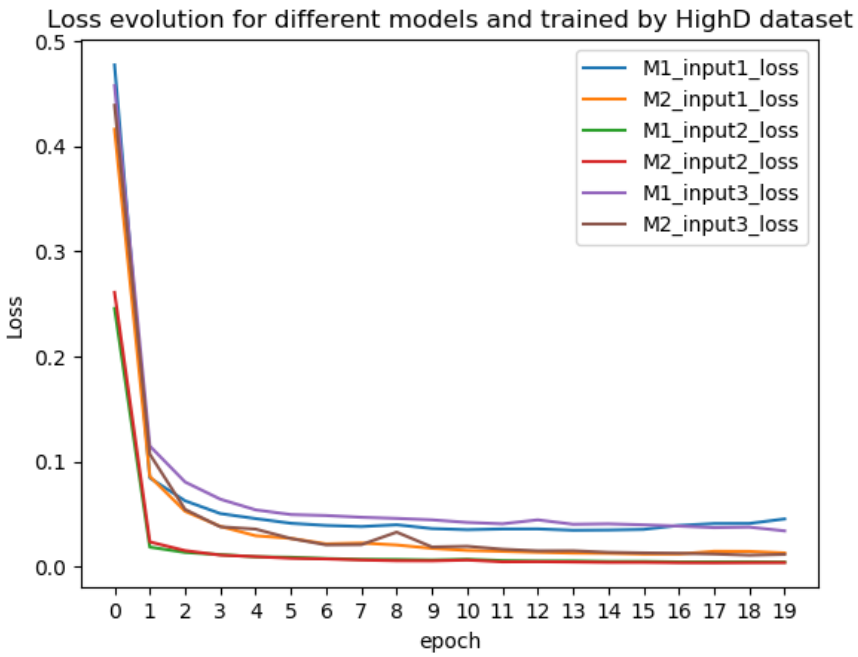


Figure 5.10 – The evolution of loss during the training for different models using different input designs by HighD dataset.

5.8 Conclusion

We have presented in this chapter an LSTM neural network model for car-following and lane-changing on road networks. The experiments have been applied on the NGSIM 101 data-set for the training and tests. Furthermore, we applied the same experiment on another data-set HighD data-set, the result is similar to the result of that on NGISM 101 data-set. The obtained results show that the LSTM neural network model has outstanding performance in terms of accuracy for longitudinal speed and lateral position variables. In comparison with the physical microscopic traffic model IDM (Intelligent Driving Model), the LSTM model has significant improved performance on the car-following part. This can be explained by the fact that physical traffic models have generally a small number of parameters, rather than neural network models. In addition, we have compared the LSTM model with a classical neural network model, and concluded that the model using LSTM has better performance than the classical neural network model.Our perspectives on this research direction are: 1) implementation the model on traffic simulators, such as SUMO (Simulation of Urban Mobility) to simulate a more realistic road traffic, and 2) extension of the model, in particular for considering several driver profiles.

Chapter 6

Highway Traffic simulation in SUMO using proposed models

We present in this chapter the implementation in SUMO of the car-following and lane-change models proposed and adapted in the preceding chapters. The objective is to simulate the whole highway traffic, where we control all the cars in the traffic by our proposed models. Meanwhile, the other types of vehicles, consisting of trucks and motorcycles, are controlled by the default models in SUMO. Different combinations of car-following models with lane changing models are tested and analyzed, aiming at understanding the performance of each model combination. Moreover, we used these different combinations of models to simulate the NGSIM traffic and the HighD traffic data-sets, in order to analyze the sensibility of all the proposed models on these two data-sets.

In this chapter, section 6.1 presents the analysis of the data we worked on, including the NGSIM data-set and the HighD data-set, with the purpose of understanding these two traffic data-sets, and therefore derive the main resemblances and differences. Section 6.2 gives the details on the implementation of the models in SUMO. Section 6.4 and section 6.5 give out respectively the results of simulation for the NGSIM traffic and the HighD traffic.

Contents

6.1 Microscopic traffic datasets : NGSIM and HighD

The traffic analysis began with some statistics of traffic (macroscopic indicators), including the traffic average speed, the car-density as well as the speed-density diagram (fundamental diagram).

NGSIM 101 dataset

As seen in the previous chapters (Chapter 3, 4, and 5), the works are mainly based on the NGSIM dataset. In Figure 6.1, a visualisation of the vehicle positions is shown.

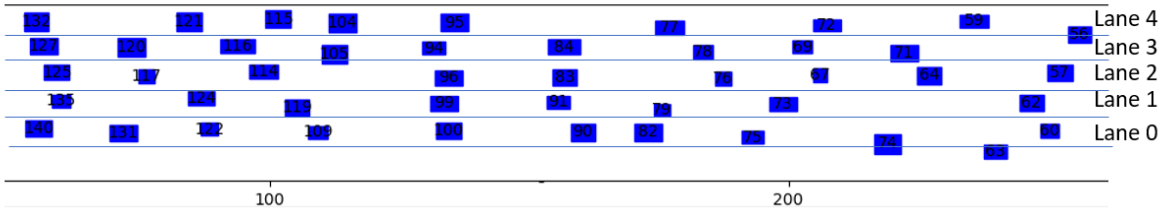


Figure 6.1 – Screenshot of the visualisation of NGSIM 101 traffic, the figure axis-x show the distance in meter (eg. 100m, 200m) to the leftest point of the road section, and the index from 0 to 4 indicating each lane from right to left.

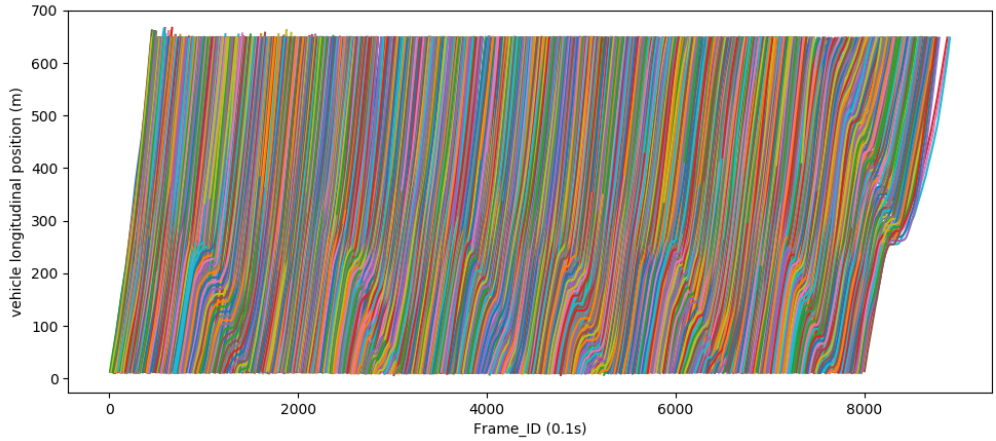
Macroscopic traffic characteristics overview Traffic macroscopic analysis is significant to show the traffic conditions. To analyze the NGSIM 101 dataset, we show firstly the vehicle longitudinal position trajectory in Figure 6.2, for the first 15-minute, second 15-minute and third 15-minute periods in Figure 6.2a, Figure 6.2b, and Figure 6.2c respectively. The vehicle speed trajectories are shown in Figure 6.3, and also for the first 15-minute, second 15-minute and third 15-minute periods, respectively in Figure 6.3a, Figure 6.3b, and Figure 6.3c.

A congestion phenomenon can be illustrated by a shock of position deferential in the vehicle trajectories (a large reducing of the slope in vehicle position trajectory, see Figure 6.2). Similarly, a congestion can also be shown by a shock-wave in the car-speed trajectories, where a shock of speed shows the local minimum of travelling speed in consecutive vehicle speed trajectories (see Figure 6.3). We can observe the traffic of NGSIM has some congestion, which sometimes starts from the middle of the road section (Figure 6.2a). From Figure 6.3, we can observe that the vehicle-speed goes up and down.

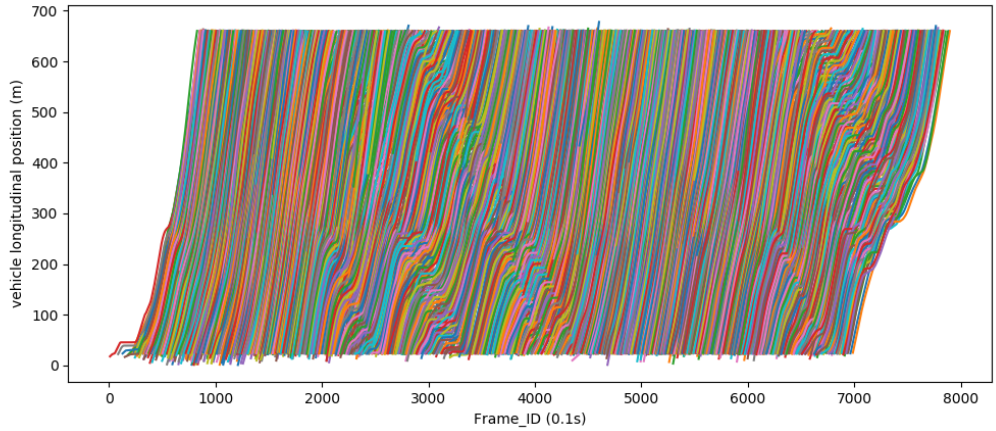
To understand the traffic on each lane, we do some investigation for the first 15 minute data. In Figure 6.4a, we show the mean traffic speed of each lane for the first minutes data. We can see that the average traffic speed evolves similarly in each lane, where the speed oscillates up

and down. In particular, we can observe that in the lane 4, which is the leftest lane and the highest speeding lane in general, the change of speed is the most one while the speed is slower than in the other lanes. Concerning the vehicle density, which is another macroscopic traffic indicator, it can be defined as the number of vehicles in each lane (here, the length of NGSIM road section is fixed (640m)). In Figure 6.4b, we show the evolution of the vehicle-density in each lane. From this figure, we can observe that the number of vehicles in the lane 4 is the biggest one, which has an effect on the traffic speed on lane 4, the slowest one (see Figure 6.4a).

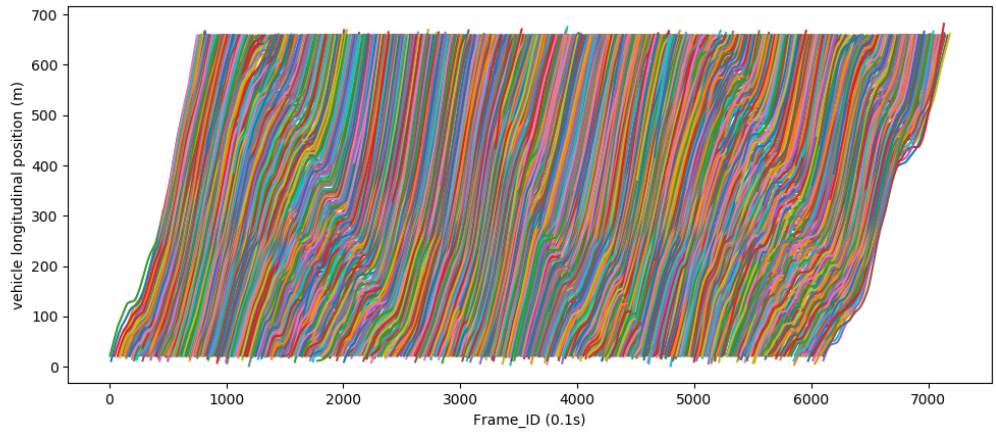
After observing the traffic speed and traffic density separately, the traffic fundamental diagram (speed-density diagram) can be displayed. For each lane in the first 15-minute period data of the NGSIM 101 dataset, the speed-density diagram is shown in Figure 6.5, while the speed-density diagram for the 3 periods of the NGSIM 101 dataset is shown in Figure 6.6. We can see that there exists some similarity on the traffic speed-density diagrams between the lanes (Figure 6.5), and also some similarity of the traffic speed-density diagrams for the 3 periods in the NGSIM dataset (Figure 6.6). We can see that the traffic is congested: the vehicle speed decreases with the increasing of the traffic density.



(**a**) First 15-minute period NGSIM 101 data : The vehicle longitudinal position (m) by Frame time (0.1s)

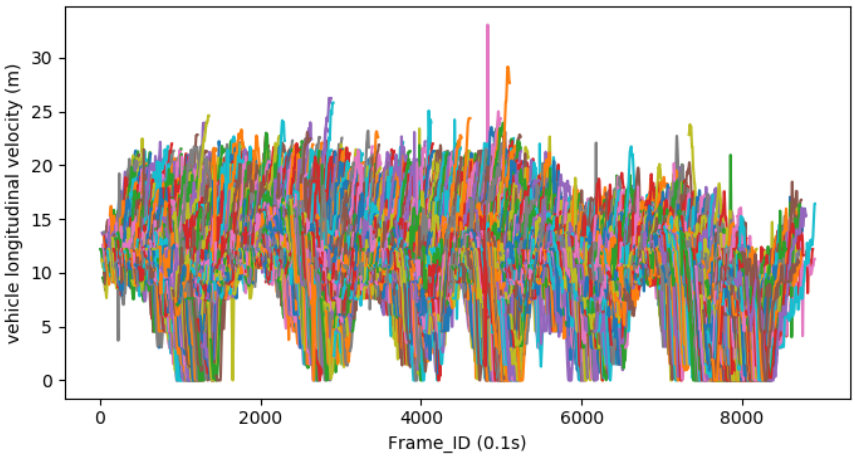


(**b**) Second 15-minute period NGSIM 101 data : the vehicle longitudinal position (m) by Frame time (0.1s)

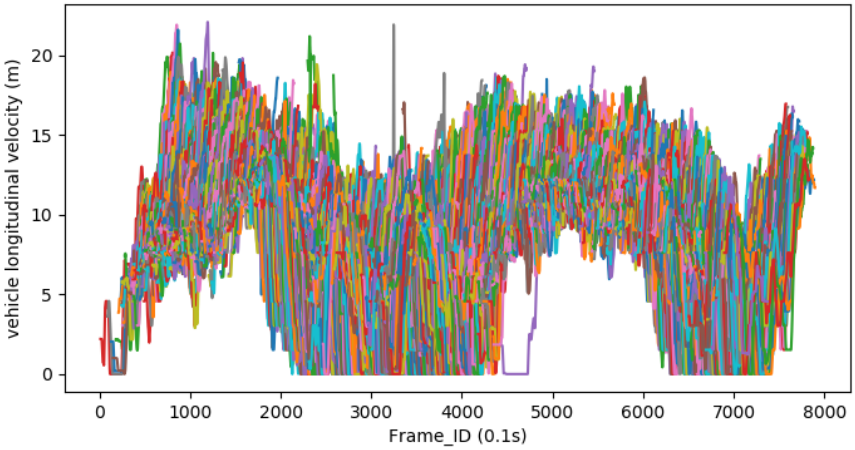


(**c**) Third 15-minute period NGSIM 101 data : the vehicle longitudinal position (m) by Frame time (0.1s)

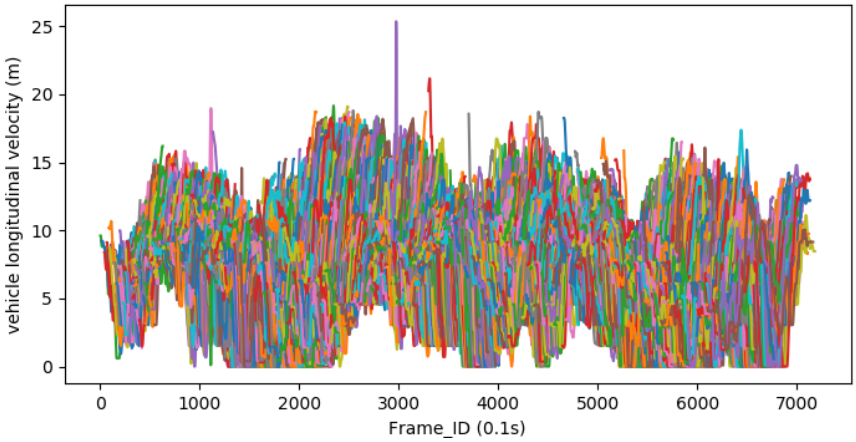
Figure 6.2 – NGSIM 101 dataset: Vehicle trajectories (longitudinal position in meter) by time (Fram ID in 0.1s)



(**a**) First 15-minute period NGSIM 101 data : The vehicle longitudinal velocity (m/s) by Frame time (0.1s)

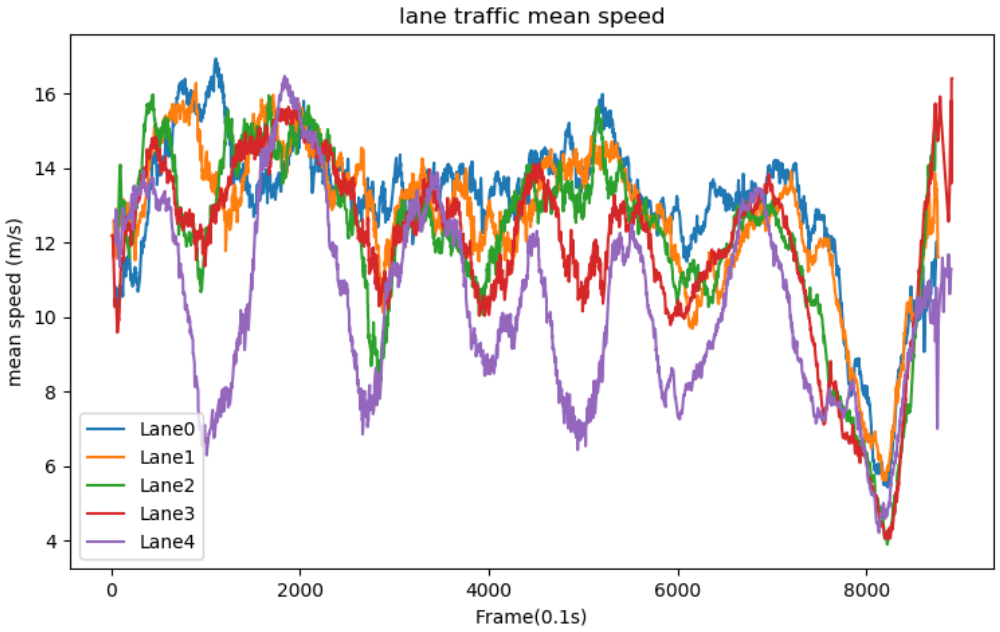


(**b**) Second 15-minute period NGSIM 101 data : the vehicle longitudinal velocity (m/s) by Frame time (0.1s)

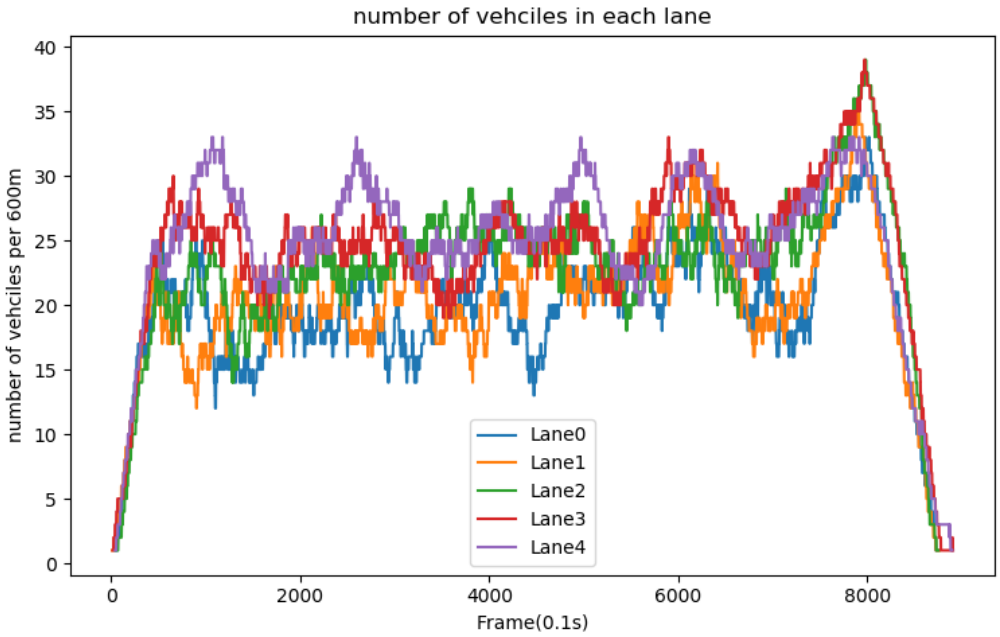


(**c**) Third 15-minute period NGSIM 101 data : the vehicle longitudinal velocity (m/s) by Frame time (0.1s)

Figure 6.3 – NGSIM 101 dataset: Vehicle longitudinal velocities (m/s) by time (Fram ID in 0.1s)



(**a**) The traffic mean speed in each lane evolution in each frame (0.1s)



(**b**) The number of vehicles in each lane evolution in each frame (0.1s)

Figure 6.4 – First 15-minute period in NGSIM 101 dataset: Traffic speed evolution and traffic density evolution in each lane

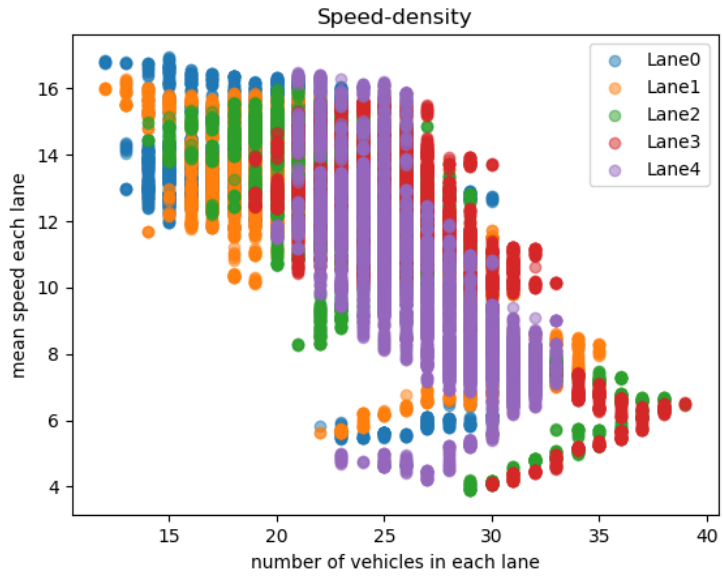


Figure 6.5 – First 15-minute period in NGSIM 101 dataset: The speed-density diagram in each lane

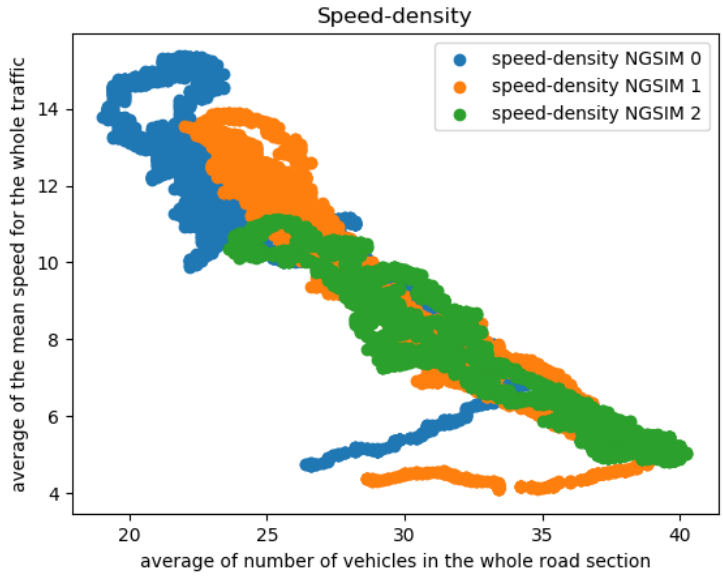


Figure 6.6 – NGSIM 101 dataset: The speed-density diagram of the whole traffic for the periods of first (NGSIM 0 in the figure), second (NGSIM 1 in the figure), third (NGSIM 3 in the figure) 15-nimute data in NGSIM 101 dataset.

The HighD dataset

We choose randomly the 13th sub-dataset to compare with the NGSIM dataset. In this sub-dataset, the traffic is captured on a bi-direction highway road section of length 420m. There are 3 lanes in each direction. We selected all the drivers in one direction (direction id = 2 in the data), and there are about 14 minutes registration of traffic consisting of 1317 vehicles' trajectories. There are about 10s of trajectory for each vehicle (see in Figure 6.7 where a screenshot for the visualisation of this sub-dataset is given).

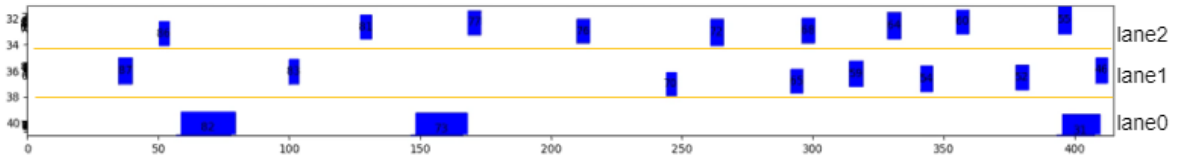
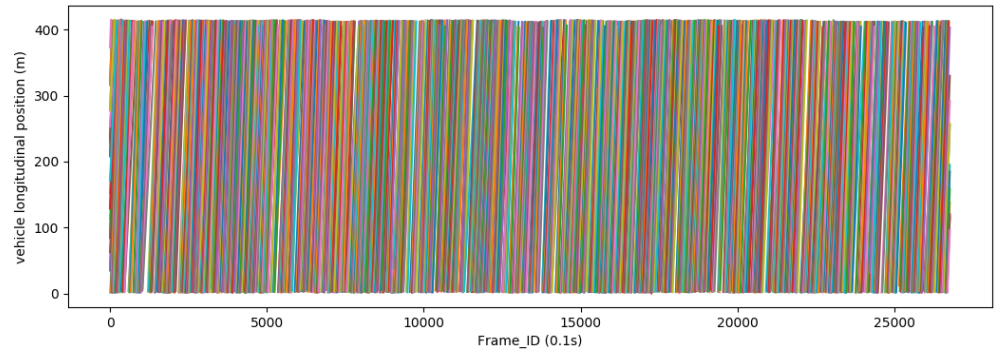
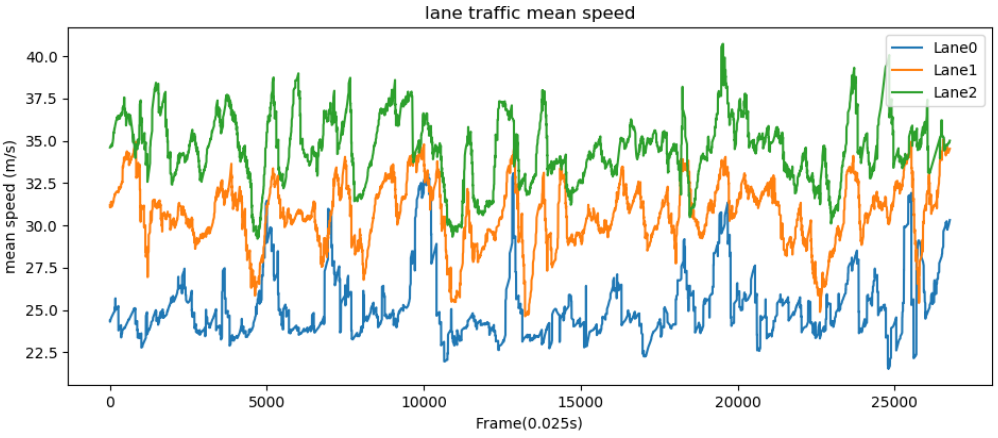


Figure 6.7 – The visualisation of vehicle position of the 13th sub-dataset HighD dataset, the figure axis-x show the distance in meter (eg. 100m, 200m, 300m) to the leftest point of the road section, and the index from 0 to 2 indicating each lane from right to left.

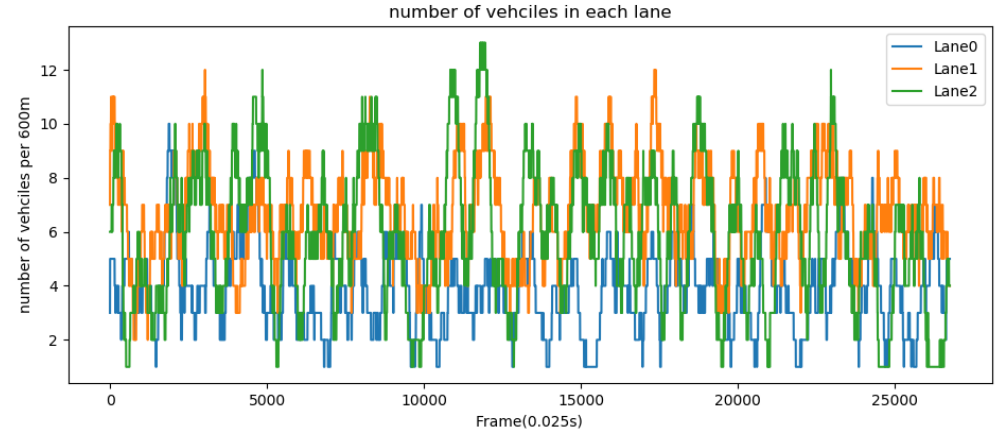
In Figure 6.8a, we show the vehicle position trajectory in the HighD dataset, and we can find the vehicle trajectory is smooth without any shocks, in contrary to the NGSIM dataset (Figure 6.2). In Figure 6.8b, we show the mean traffic speed on each lane, where we can see that it evolves similarly in each lane (at a high level), where the speed does not oscillate as in the NGSIM traffic. We can also observe that on lane 2, which is the leftest one, the average speed is the highest one among the three lanes. In Figure 6.8c, we show the evolution of the vehicle-density diagram for each lane, and from this figure, we can observe that the number of vehicles is almost the same in average on the three lanes, and it ranges from 2 to about 12 in 420 meter. This vehicular density is much lower than that of the NGSIM dataset, see Figure 6.4b).



(**a**) The vehicle longitudinal position(m) by the Frame ID (0.025s) for the 13th sub-dataset in HighD dataset



(**b**) The traffic mean speed in each lane evolution in each frame (0.025s)



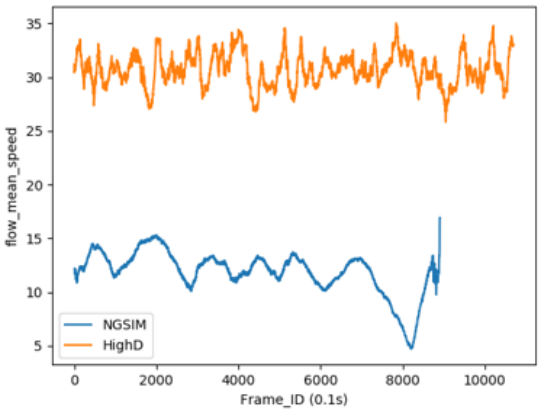
(**c**) The number of vehicles in each lane evolution in each frame (0.025s)

Figure 6.8 – The 13th sub-dataset in HighD dataset: Traffic speed evolution and traffic density evolution in each lane

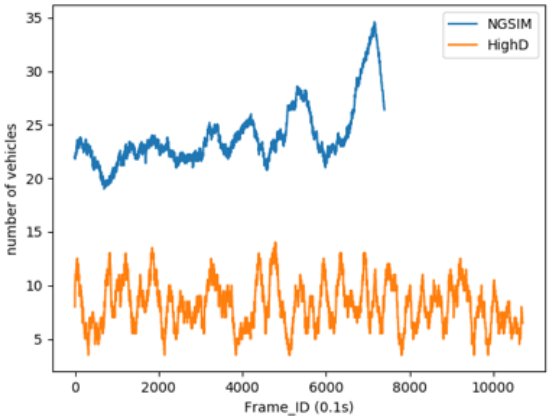
6.1.1 Comparison of NGISM dataset and HighD dataset

To show the difference between the NGSIM 101 and the HighD datasets, we give in Figure 6.9a, Figure 6.9b and Figure 6.9c, a comparison between the NGSIM 101 first 15 minute period data and the 13th sub-dataset of HighD dataset, in terms of traffic mean speed, traffic density, and the speed-density diagram. The HighD traffic is not as dense as the one in the NGSIM 101 dataset; see Figure 6.9b). This can also be seen in Figure 6.9a and Figure 6.9c.

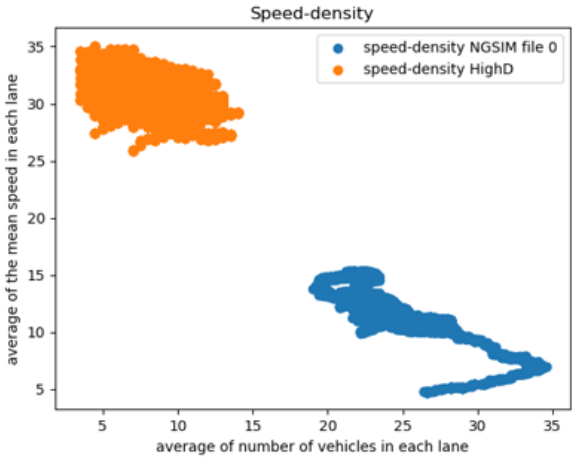
In the following sections, we simulate in SUMO, the two traffics of NGSIM and HighD with different combinations of car-following and lane-change models. In section 6.2, we first give the adaptation and the implementation of our proposed models used in Chapter 3, Chapter 4, and Chapter 5. Then, in section 6.4 and section 6.5, we give the results of simulation of the first 15 minutes period traffic in the NGSIM dataset and the traffic of 13th sub-dataset in the HighD datset.



(**a**) The comparison of NGSIM traffic mean speed and HighD traffic mean speed



(**b**) The comparison of NGSIM traffic density and HighD traffic density



(**c**) The comparison of NGSIM traffic speed-density diagram and HighD traffic speed-density diagram

Figure 6.9 – Three subfigures

Figure 6.10 – The comparison of NGSIM 101 dataset and HighD dataset : the first 15 minute period data in NGISM 101 dataset and the 13th sub-dataset in HighD dataset

6.2 Implementation of the models in SUMO

As well known, car-following and lane changing models are the two fundamental models in microscopic traffic. In this thesis, we used the car-following model IDM (intelligence driver model) in Chapter 3, which is a well known car-following model, and is already available in SUMO. In Chapter 5, we proposed a human-like driving model using LSTM neural network, simultaneously for car-following and lane-changing behavior control. The latter model controls the vehicle movement by predicting the decision on speed and lateral position, based on the vehicle own movements and on the states of its surrounding vehicles in the historic steps. As shown above, the neural network based model can be more efficient in predicting the human driving behavior, than the classical physical microscopic model, e.g. the intelligent driver model (IDM). Indeed, the neural network based model performs a good accuracy for the real vehicle trajectory data. In Chapter 4, we have shown that classical physical lane change models have limited performance, and the reinforcement learning provides an alternative method for lane changing modelling. Q-learning algorithms have been used with online calibration on human lane change, based on the NGSIM data. The results have also shown a good accuracy for human lane change imitation.

In this section, we study the use of different combinations of car-following models and lane changing models, including 1) the calibrated IDM model, 2) an adapted neural network based car-following model, 3) the SUMO default lane changing model and 4) our proposed reinforcement learning based lane changing model. We simulate the NGSIM and HighD traffics, and compare different combinations.

6.2.1 Calibration of the IDM model

The IDM model needs to be calibrated to present the real human behavior. For the car-following model calibration, we have presented previously in Section 3.6 the method of calibration for the selected extreme profiles (aggressive and inattentive), but with the purpose of generation of car collisions (in numerical simulation) by increasing the number of extreme drivers in the traffic (see. Chapter 3). In this section, with the objective of generalizing the calibration work, and in order to reuse the calibrated model for other use cases, we propose the use of the NGSIM data to calibrate the IDM model and to obtain a set of IDM parameters. This set of parameters is the average value of parameters calibrated through all drivers trajectories registered in the first 15 minute period NGSIM 101 dataset. We use here the same method as in the one in section 3.6 for calibrating all drivers presented in NGSIM data-set, and we choose the parameters' boundaries with an assumption to be suitable for each driver. Let us recall that the calibration is performed using a genetic algorithm to find the optimal parameters of the IDM model for each diver in the NGISM data-set, with the purpose of minimising a proposed

error in position. The error is defined as the root mean square percentage errors (RMSPE) :

$$RMSPE(\theta) = \sqrt{\frac{1}{P}\sum_{i=0}^{P}\frac{(x_i^{simul}(\theta) - x_i^{data}(\theta))^2}{x_i^{data}(\theta)^2}}, \tag{6.1}$$

The boundary of each parameter (upper and lower bounds) are given as an input to the genetic algorithm. In Table 6.1 we show the chosen boundary of each parameter of the IDM model, and we calculate the result for errors between the real data and the output of the calibrated IDM model, including RMSE for speed, and RMPSE for the position. We show in Table 6.1 the mean value, the standard deviation, the median, and the value of the quantile at 25% and 75% for each parameter and each error value. Table 6.1 shows the performance of IDM model and the calibration by the genetic algorithm. We have the RMSE on the car-speed at 1.004m/s for about 1900 drivers, and the RMSE on the car-position at about 9.1m, with only 4.4% of RMPSE. The performance of the IDM model calibration can be also observed in Figure 6.12, where we show the result of calibration of the IDM model for 4 selected drivers. We give in each sub-figure the driver's real speed and the speed computed by the calibrated IDM model.

Parameters and error	Bounds	Mean	std	median	25%	75%
V0 (m/s)	[10, 40]	25.07	8.70	22.33	18.54	30.10
T (s)	[0.1, 4]	1.53	0.75	1.52	1.06	1.94
S0 (m)	[0.1, 10]	2.53	3.50	0.10	0.1	4.48
Reaction time (s)	[0.1, 0.5]	0.2	0.12	0.13	0.12	0.14
a (m/s2)	[0.1, 6]	4.6	2.1	6.0	3.16	6.0
b (m/s2)	[0.1, 6]	5.6	1.27	6.0	6.0	6.0
RMSE speed (m/s)		1.004	0.37	0.93	0.75	1.18
RMSE position (m)		9.152	19.69	5.67	3.76	8.92
RMPSE position (%)		4.4	28.0	2.41	1.6	3.6

Table 6.1 – Calibration results for all drivers of the first 15-minute period data in NGSIM dataset.

From Figure 6.11, we see that the parameters $V0$ and T are well distributed in their respective intervals. However, the parameters a and b are rather close to the upper limit 6 m/s^2. This result corresponds to what is found in (M. Zhu, Xuesong Wang, Tarko, et al., 2018). We use the column of the mean value in Table 6.1 as the calibrated parameters of IDM to simulate the traffic in SUMO.

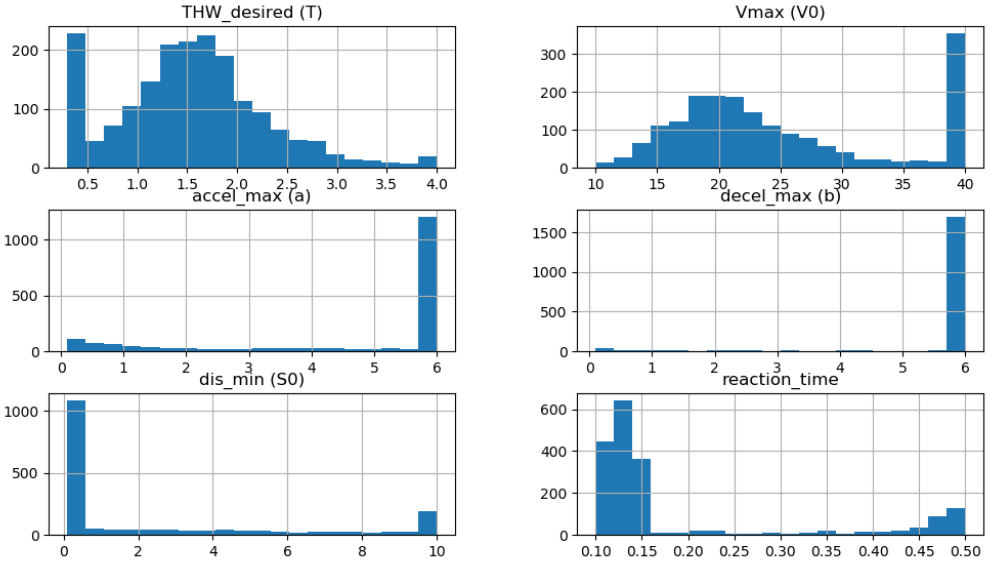


Figure 6.11 – The distribution of the calibrated IDM model parameters for all drivers in the first 15 minute period data in NGSIM 101 dataset

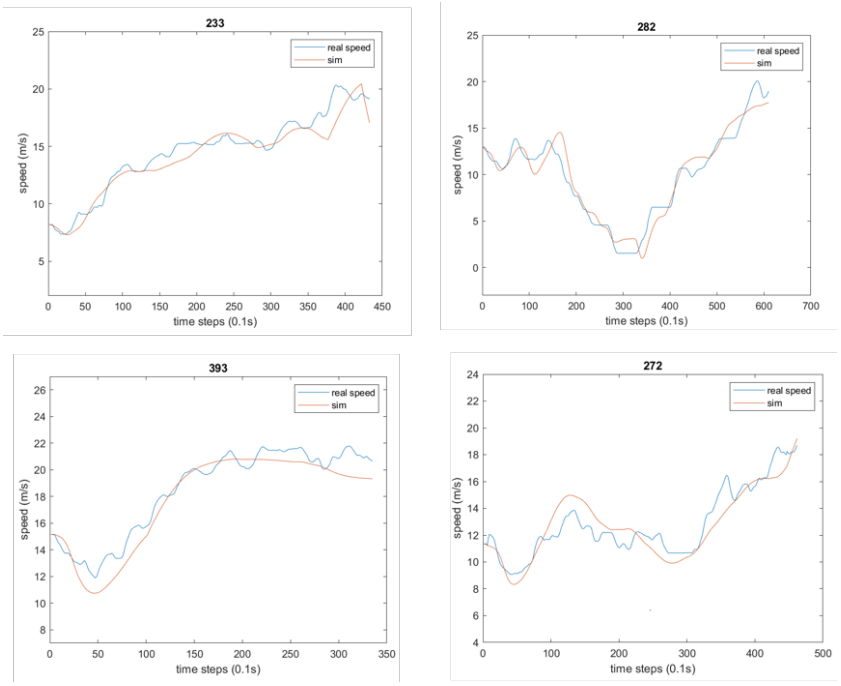


Figure 6.12 – Some result examples for the IDM model calibration, each sub-figure shows the real speed in NGSIM and the predicted speed using calibrated IDM model

6.2.2 Car-following model using Neural network

The model that we proposed in our research work in Chapter 5 is a model using LSTM neural network, where we compared three different input designs as shown in Table 6.2. Since this model has a lot of inputs, We propose a simplification of the model in order to implement it in SUMO and to simulate road traffic in SUMO.

	x_{target}	y_{target}	v_{target}	a_{target}	l_{target}	w_{target}	x_i	y_i	v_i	a_i	l_i	w_i	Δv_i	s_{long_i}	s_{lat_i}	p_i
input design 1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
input design 2	✓	✓	✓		✓	✓					✓	✓	✓	✓	✓	✓
input design 3	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 6.2 – input designs

In order to implement a neural network car-following model in SUMO to simulate the NGSIM and the HighD traffics, I propose to use a simpler car-following model with a classical neural network trained by the NGSIM data. The model uses only the features shown in Table 6.3. In addition, the target vehicle position is a variable depending on the traffic map, where the traffic road length is not constant. We don not use here the target absolute variables, but relative values with the surrounding vehicles, which are more representative. Moreover, for the surrounding vehicles, we considered only the leading vehicle and the following vehicle of the target vehicle, which are the most two important participants for the car-flowing speed controlling.

All the used features of target vehicles :

- longitudinal position $x_{target}(t)$
- longitudinal speed $v_{target}(t)$
- vehicle length l_{target} and vehicle width w_{target}

All the features of the leading vehicle and following vehicle are :

- relative speed $\Delta v_i(t) = v_{target}(t) - v_i(t)$
- longitudinal distance $s_{long\_i}(t) = x_{target}(t) - x_i(t)$
- vehicle length l_i and vehicle width w_i
- presence p_i : 0 for non presence, 1 for presence
  if vehicle v_i is no present, all the relevant value $(l_i, w_i, \Delta v_i, s_{long\_i}, s_{lat\_i})$ for this vehicle set to be 0.

In this input design, we have a total of 13 features as inputs of the neural network.

	x_{target}	y_{target}	v_{target}	a_{target}	l_{target}	w_{target}	x_i	y_i	v_i	a_i	l_i	w_i	Δv_i	s_{long_i}	s_{lat_i}	p_i
input design2 SUMO			✓		✓	✓					✓	✓	✓	✓		✓

Table 6.3 – Input design2 in SUMO implementation

Neural Network model

The neural network model is changed. We reduce the input features, and take only one output of the model, which is the longitudinal speed. In this context, we supposed using a classical fully connected neural networks (**mathew2012neural**), without the LSTM architecture; see Fig. 6.13. The neural network model for car-following is shown in Fig.6.14.

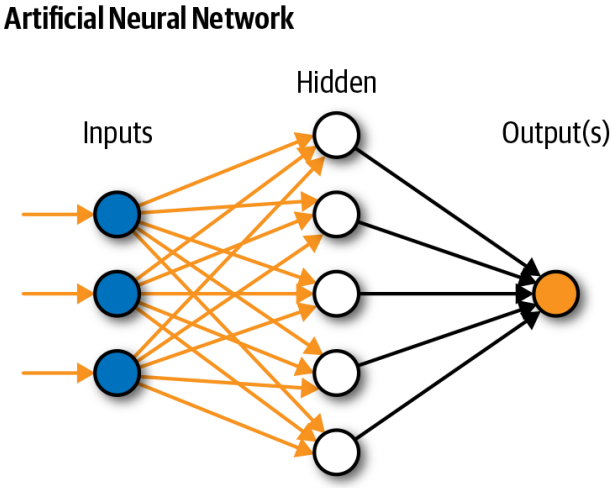


Figure 6.13 – Example of a fully connected artificial neural network

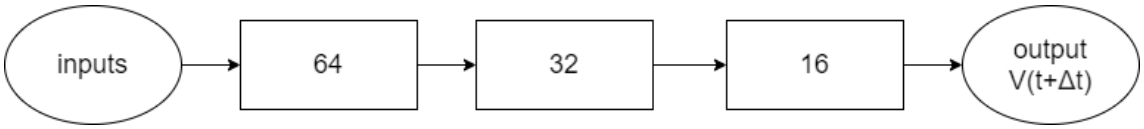


Figure 6.14 – Example of a fully connected artificial neural network

We give in Table 6.4 the RMSE for all models trained and tested using the NGSIM data-set. The car-following model proposed for SUMO implementation has a similar accuracy as the best model we obtained in Chapter 5. Therefore, we think that this car-following model using a fully connected neural network can replicate the human car-following driving behavior.

Mean-driver-RMSE	Train		Test1		Test2	
	Speed (m/s)	lateral Position (m)	Speed (m/s)	lateral Position (m)	Speed (m/s)	lateral Position (m)
M1_input1	0.2819	0.1308	0.27	0.14	0.294	0.1389
M2_input1	0.1524	0.1608	0.1767	0.167	0.1964	0.1709
M1_input2	**0.056**	**0.024**	**0.05**	**0.024**	**0.051**	**0.023**
M2_input2	0.0686	0.0848	0.0688	0.0752	0.0695	0.0669
M1_input3	0.4188	0.1185	0.4356	0.1353	0.4155	0.1494
M2_input3	0.2465	0.1331	0.3248	0.1305	0.3021	0.1396
M_{NN}_input2	0.2495	0.1266	0.2565	0.1229	0.2749	0.1209
M_{NN2}_input2	0.1423	0.1338	0.1389	0.16415	0.1503	0.1802
CF by classical NN for SUMO	0.09		0.0760		0.0764	

Table 6.4 – Mean RMSE for the different models with different input designs, and for training, test1 and test2 respectively.

Implementation of the neural network CF model

In SUMO, the car-following model is controlled by a parameter called 'speed mode', and this parameter is presented by a 5-bit value. The given integer is a bitset (bit0 is the least significant bit) with the following fields:

- bit0: Regard safe speed
- bit1: Regard maximum acceleration
- bit2: Regard maximum deceleration
- bit3: Regard right of way at intersections
- bit4: Brake hard to avoid passing a red light

We set the speed mode at 1 for controlled vehicle, in order guaranty safety (avoid car-collisions). For all the other situations, the speed is controlled by our neural network car-following model, proposed for the implementation in SUMO.

6.2.3 Lane change model using Reinforcement Learning

The proposed lane change model is the the model we presented in Chapter 4. This model is also calibrated using the NGSIM 101 data-set.

Implementation of the lane changing model based on reinforcement learning in SUMO

The lane change model discriminates four reasons to change lanes:

- strategic (change lanes to continue the route)
- cooperative (change in order to allow others to change)
- speed gain (the other lane allows for faster driving)

- obligation to drive on the right

This strategy of lane change control has a parameter named 'lane change mode' and to disable all autonomous changing but still handle safety checks in the simulation, either one of the modes 256 (collision avoidance) or 512 (collision avoidance and safety-gap enforcement) may be used. We set the lane change mode at 256 to disable all lane change strategy of SUMO, but we need to check for collision avoidance when our lane change model demands a change lane. To be noticed, the lane change order by our lane change model can be prevented by SUMO if safety is not guaranteed.

Controlling one car in the traffic

The implementation began with testing in the NGSIM scenario for controlling only one car using the neural network based car-following model and the lane changing model using reinforcement learning. The result in simulation for the vehicle speed control is presented in Figure 6.15. It seems that the vehicle is well controlled by our proposed model, and has a different speed policy comparing to the sumo default model. We show in Table. 6.16a the

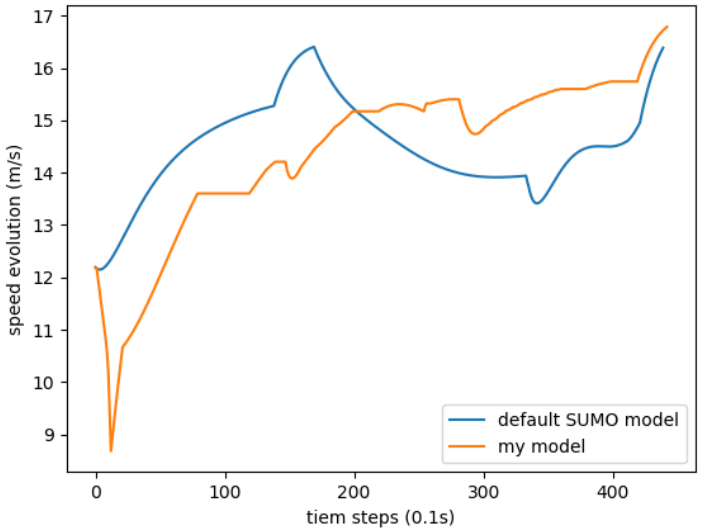


Figure 6.15 – Example of a controlled vehicle speed in SUMO

results for the lane change using SUMO LC default model, and in Table. 6.16b the results by our proposed model. The controlled car changes only once its lane by using the default model in SUMO, meanwhile, it changes 4 times using our lane change model with reinforcement learning.

Index	change_dir	hange_followerGa	ge_followerSecun	ange_followerSpe	change_from	change_id	:hange_leaderGap	nge_leaderSecure	hange_leaderSpee	ange_origLeaderG	e_origLeaderSecu	nge_origLeaderSp	change_pos	change_reason	change_speed	change_time	change_to	change_type
50	-1	27.97	19.55	16.01	route101_1	280	12.70	9.90	16.65	22.07	13.85	14.00	505.43	keepRight	13.42	96	route101_0	280@280

(**a**) Example of a controlled vehicle lane change in SUMO using default model

Index	change_dir	hange_followerGa	ge_followerSecun	ange_followerSpe	change_from	change_id	:hange_leaderGap	nge_leaderSecure	hange_leaderSpee	ange_origLeaderG	e_origLeaderSecu	nge_origLeaderSp	change_pos	change_reason	change_speed	change_time	change_to	change_type
38	-1	24.55	15.82	14.83	route101_1	280	9.79	7.25	18.78	9.79	7.25	18.78	208.07	traci\urgent	13.98	77.4	route101_0	280@280
43	1	15.10	11.14	13.53	route101_0	280	25.71	13.07	16.47	63.73	13.83	16.00	292.24	traci\urgent	15.17	83.1	route101_1	280@280
49	-1	25.92	17.47	16.10	route101_1	280	68.33	13.16	16.60	29.07	15.24	15.33	347.16	traci\urgent	15.23	86.7	route101_0	280@280
51	1	19.49	16.62	15.85	route101_0	280	25.23	18.24	13.68	46.77	19.07	13.18	409.79	traci\urgent	14.79	90.8	route101_1	280@280

(**b**) Example of a controlled vehicle lane change in SUMO using my model

Figure 6.16 – Example of a controlled vehicle lane change in SUMO

6.3 Different combinations of models for Highway traffic simulation in SUMO

Once we tested the control of one vehicle in simulation using the proposed CF and LC models, we begin the simulation of the whole traffic (NGSIM and HighD) aiming at testing all the combinations given in Table 6.5.

	Car-following models			Lane changing models	
Models in SUMO	IDM default	IDM calibrated with NGSIM	CF by NN	LC default SUMO	LC by RL
Sumo_default	✓			✓	
Model1		✓		✓	
Model2			✓		✓
Model3		✓			✓
Model4			✓	✓	

Table 6.5 – A summary of all the tested models in simulation

6.4 Results of the Simulation of NGSIM traffic

In Figure 6.17, we show a screenshot for the visualisation of NGSIM traffic in SUMO. We present the results in two parts consisting of traffic average speed evaluation and number of lane changes evaluation.

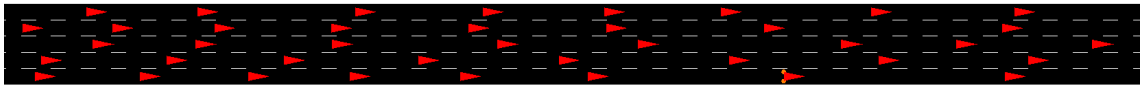


Figure 6.17 – Screenshot for the simulation of NGSIM traffic

Traffic Speed evaluation

The comparison of the average traffic speed between real data and simulated traffic using different combination of models is shown in Figure 6.18. We give the RMSE between the NGSIM data and the simulated traffics using different combinations on traffic average speed, respectively for sumo_default model, model1, model2, model3 and model4 in Table.6.6. We observe that the model2, which uses CF model by neural network and LC model using reinforcement learning

has the best performance among all the tested models. The traffic speed-density diagram of the simulated traffic obtained from SUMO is shown in Figure 6.19. We show the speed-density diagram produced by each simulated model. We observe that the Model2, which uses the CF by neural network and LC by reinforcement learning has the best performance; see Figure 6.19. We then focus on the two best models by the criteria shown in Table 6.6, which are the model2 and the model1. We show their diagram of speed-density in Figure 6.20. Therefore, by the result shown in Table 6.6 and the result shown in Figure 6.20, we observe that model2 has the best performance for simulating the NGSIM traffic compared with the real data.

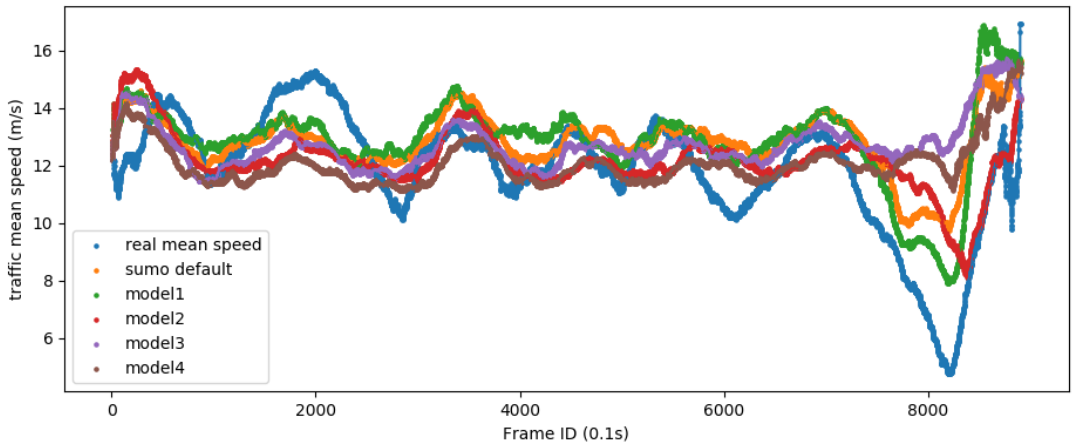


Figure 6.18 – NGSIM simulations : Traffic average speed simulation for all cars controlled by different models, respectively for sumo_default model, model1, model2, model3 and model4 (see Table. 6.5 for the presentation of all models)

RMSE	sumo_default	model1	model2	model3	model4
m/s	1.4323	1.3759	**1.2670**	1.5810	1.5424

Table 6.6 – NGSIM simulations : RMSE between traffic average speed in simulation for all cars controlled respectively by sumo_default model, model1, model2, model3 and model4 (see Table. 6.5 for the presentation of all models), with real NGSIM traffic average speed

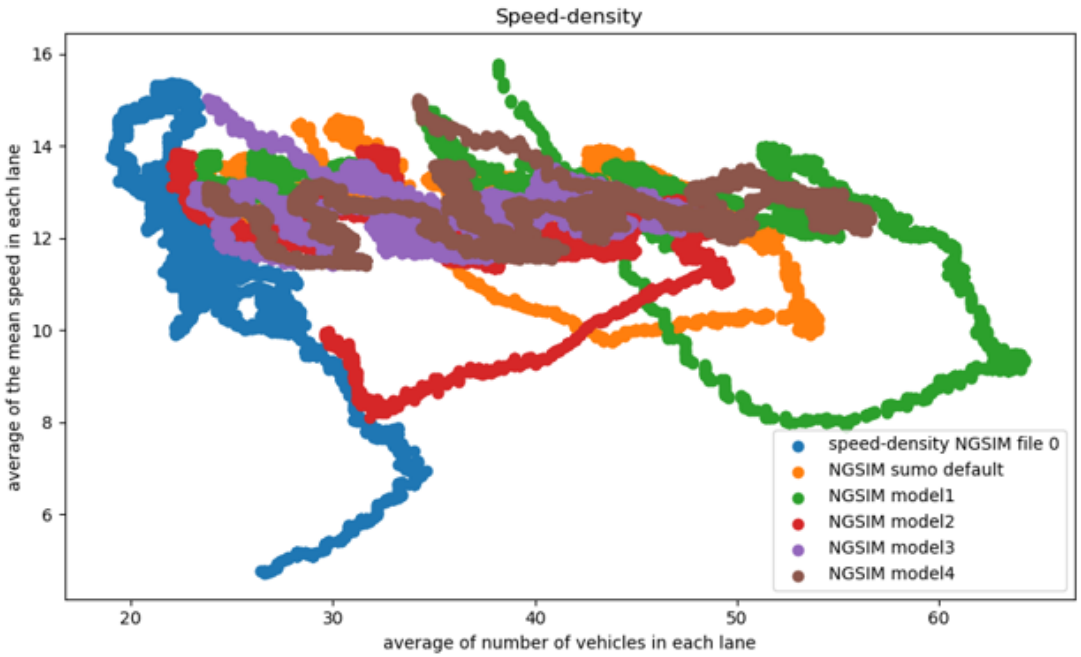


Figure 6.19 – NGSIM simulations : The display of speed-density diagram for different models and the real NGSIM data

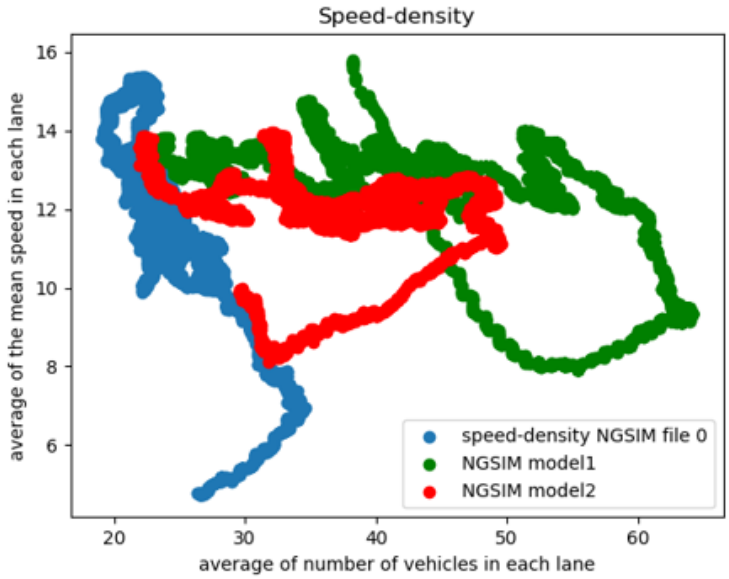


Figure 6.20 – The display of speed-density diagram the real NGSIM data, and the result simulated by the model1 and model2, which are the best models regarding Table 6.6.

Vehicles Lane-changing number evaluation

The lane change distribution of each cars is shown in Figure 6.21. Comparing to the real distribution of number of lane changes in the NGSIM data-set, the distribution of lane changing number produced by all models in simulation is similar to that of the real data. However, we notice that the result of simulation for the lane changing are obtained under the SUMO collision avoidance check (when the LC is done by our model). Therefore, even the lane changing model using reinforcement learning is shown as a collision-free in Chapter 5. In order to exted the model for controlling all the cars' LC in the traffic instead of controlling only one car, we need to introduce safety and car-collision avoidance in the lane changing modeling.

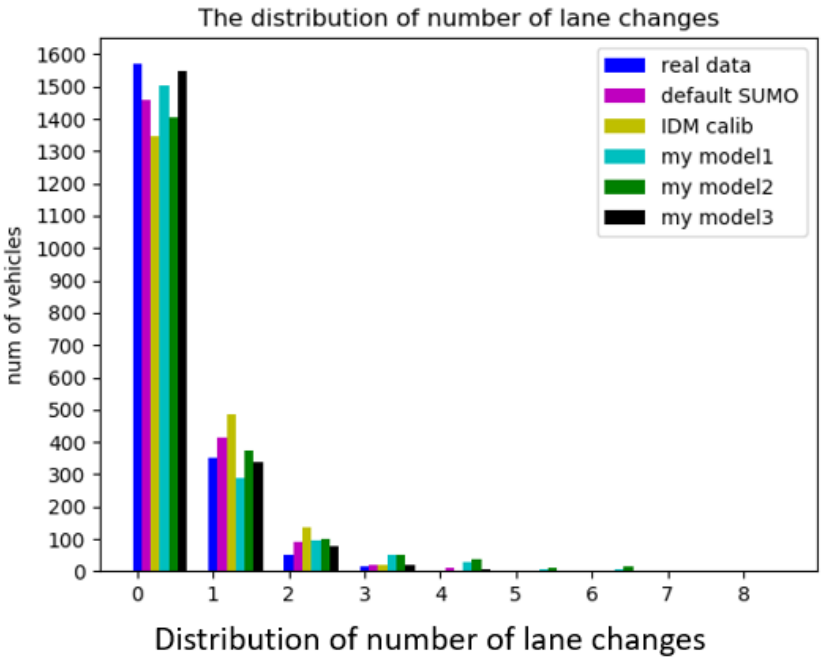


Figure 6.21 – NGSIM simulations : Vehicles' lane changes number distribution for all models, including the NGSIM dataset

6.5 Results of the Simulation of the HighD traffic

In the previous section, we have shows the simulation results for the NGSIM traffic. In this section, we present the results of simulations for the HighD traffic. In the HighD data-set, we choose the scenario of the 13th sub-dataset among from a total of 60 sub-datasets. The 13th sub-dataset contains traffic data on a 3-lane road map on the highway with 1217 vehicles in the traffic during 17 minutes. Similarly to the evaluation of simulations for the NGSIM traffic, we are also interested in the presentation of traffic speed evaluation and lane changing evaluation for the simulations of HighD traffic. In Figure 6.22, we show a screenshot of the HighD simulation in SUMO. As already indicated in section 6.1, the NGSIM and the HighD traffics are different in terms of traffic density speed. We reuse the proposed models studied for the NGSIM data-set (Table6.5), to simulate the HighD traffic and to observe the performances of each model.

Figure 6.22 – Screenshot for the simulation of HighD traffic

Traffic Speed evaluation

For the traffic speed evaluation, we evaluate the simulation using different combinations of CF and LC models; see Table 6.5). In Figure 6.23, we show the results of HighD traffic simulations regarding the average traffic speed. We firstly tested the IDM model calibrated with the NGSIM data-set. The parameter maximum speed (V0 in Table 6.1) of IDM needs to be re-calibrated to simulate the HighD traffic. The average (on drivers) of the maximum speed in the NGSIM traffic is 25.07m/s (see V0 in Table 6.1), while for the HighD traffic, we rather take 40m/s for the maximum speed.

We show in Figure 6.24 the result of the HighD traffic simulations, for different models. for the SUMO default model with the maximum speed 40m/s, the model1 using calibrated IDM model with the maximum speed 40m/s and the SUMO default lane change model, the model2, the model3 and the model4 presented in Table 6.5. In Table 6.7, we compute the RMSE error between the simulation result and the real data of the HighD dataset, in terms of traffic average speed. We observe that the model1 using the calibrated IDM model with the maximum speed at 40m/s and the SUMO default lane change model is the best one; while the model4 using the CF by neural network and default SUMO LC model is the second best model for simulating the HighD traffic. The model2 and model3 give a traffic speed very different from the real data.

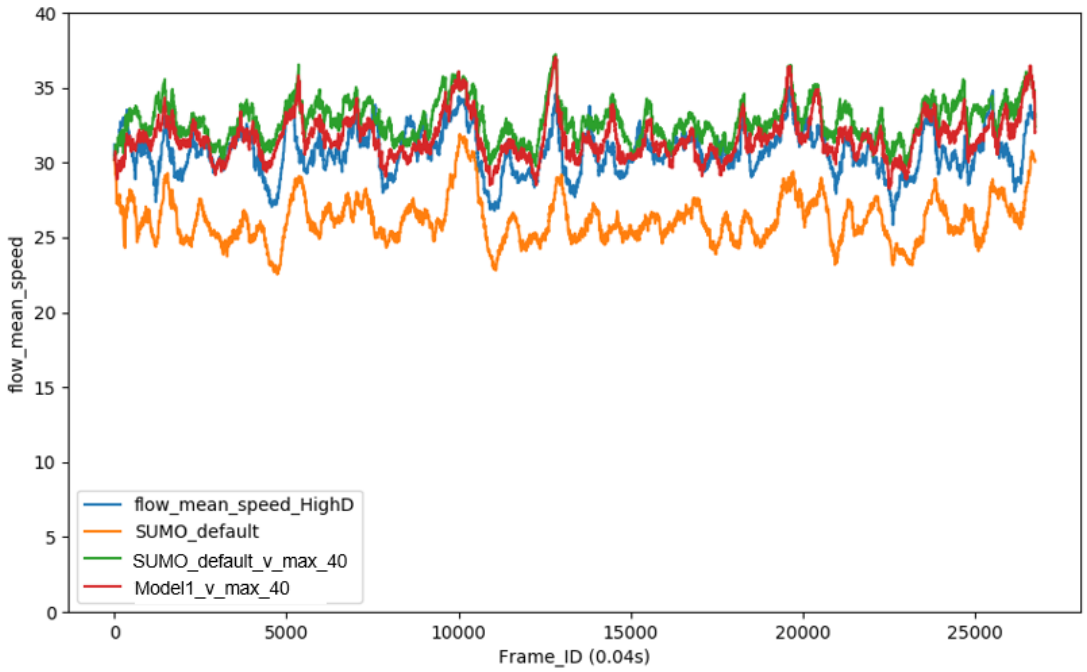


Figure 6.23 – HighD simulations: Simulate HighD using IDM model with parameters calibrated by NGSIM

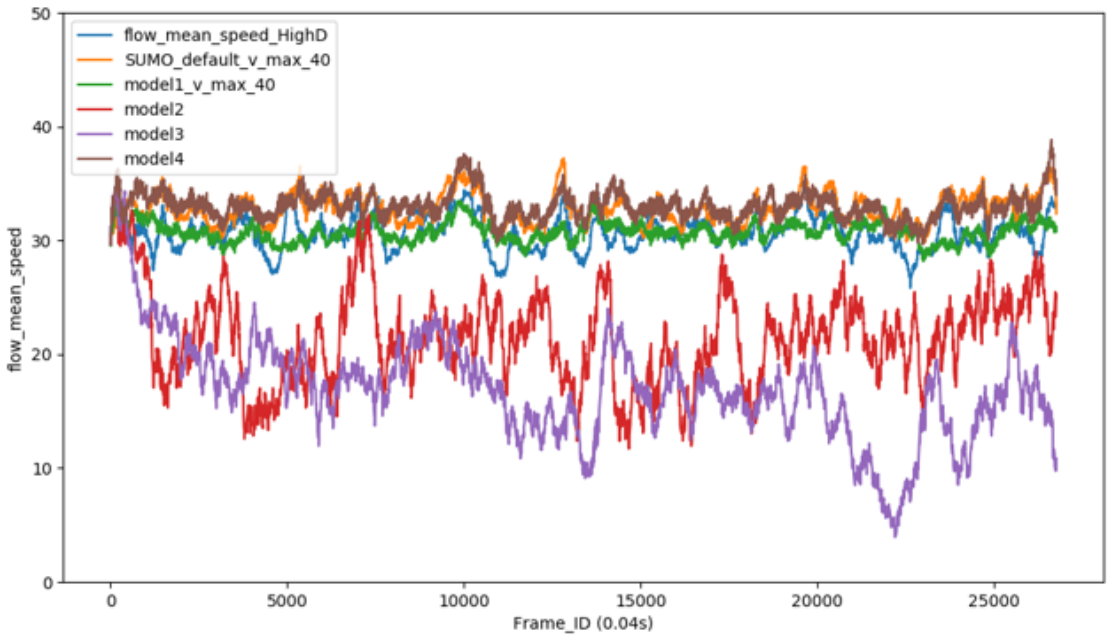


Figure 6.24 – HighD simulations: Simulate HighD using all proposed models

The LC behavior in the HighD dataset should be very different from the behavior in the NGSim dataset, that is why Model2 and Model3 both have the worst performance (both of them have the CL by reinforcement learning, and this model is trained using NGSIM dataset). For the model4, it uses the neural network CF trained by the NGSIM dataset, with good performances for simulating the HighD traffic in terms of the speed-density; that is what we can see by the speed-density diagram of Figure 6.25 and Figure 6.26.

RMSE	sumo_default	Sumo_default_v_max40	model1_v_max40	model2	model3	model4
m/s	4.7	2.44	**1.1465**	9,2501	13,7630	**2,3572**

Table 6.7 – HighD simulations : RMSE of traffic average speed in simulation for all cars controlled repectively by all the proposed models

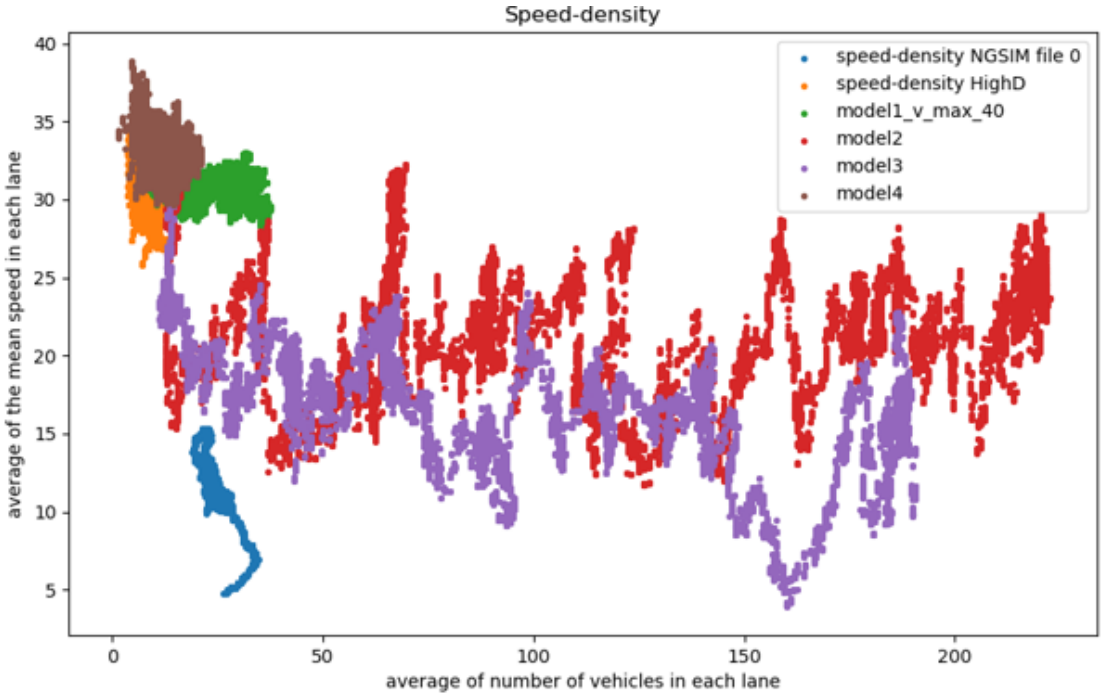


Figure 6.25 – HighD simulations : The display of speed-density diagram for different models and the real HighD data

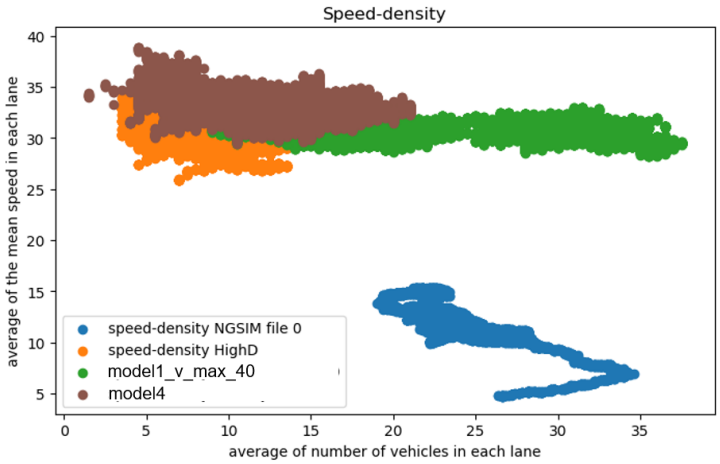


Figure 6.26 – The display of speed-density diagram the real HighD data, and the result simulated by the model1 with the re-calibration of max speed and the model4, which are the two best models regarding to table6.7.

Vehicle Lane-changing evaluation

For the number of lane changes, we show its distribution for each car in Figure 6.27, for different models and for the real data. From this figure, we find that the model2 and mode3, which are the models using LC by reinforcement learning, have a bad performance for replicating the real driving behavior in terms of the number of lane changes. Other models, SUMO default model with the maximum speed at 40m/s and the model1 using calibrated IDM model based on NGSIM dataset with the maximum speed at 40m/s, and the model4 has a similar performance on lane changing, where all the simulated vehicles have zero lane change during the simulation. In the real data, several vehicles has one lane change. The problem of lane changing behavior modeling needs to be further worked on, since the LC model using reinforcement learning trained on the NGSIM dataset cannot simulate a very different traffic as the one of the HighD data-set (in terms of lane changing behavior).

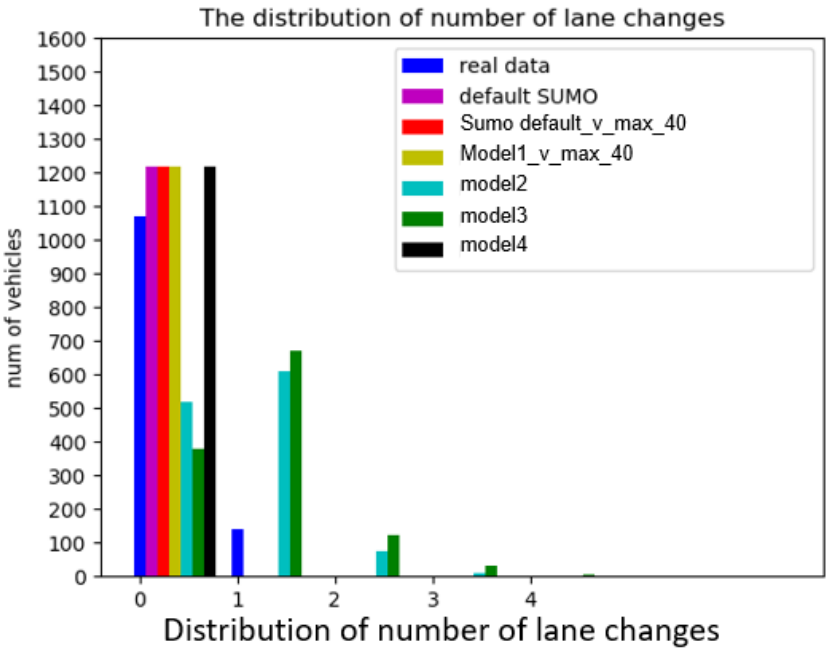


Figure 6.27 – HighD simulations : Vehicles' lane changes number distribution for all models, including the HighD dataset

6.6 Conclusion

The NGSIM traffic is very heavy, while the HighD one is very light. The neural network and reinforcement learning models trained only on the NGSIM data-set are not suitable for simulating traffic as different as the HighD one, from the NGSIM one. The IDM model, which uses physical parameters, is easy to extend with human factors (such as the reaction time) and

it can be used to generate car-collisions. The NGSIM-calibrated IDM model is less sensitive to data, and can be used to simulate other traffics like the HighD one, but it requires a re-calibration with the maximum vehicle speed.

Chapter 7

Conclusion and Perspectives

Driving behavior models describe the driver's maneuvering decisions under different traffic conditions which are crucial for microscopic traffic simulation systems and important in several research areas such as traffic safety, capacity analysis, and assignment studies. On the other hand, there is a growing interest in autonomous driving, which is a rather challenging topic because it involves the automated operation of vehicles in many different situations. We expect that fully autonomous vehicles can reduce car accidents and improve overall traffic safety. However, the complex system of autonomous vehicles combining the perception, decision, the control system and the traffic environment is a dynamically changing environment which involves many interactions between road users. Driving tests are essential to validate and to ensure the functionalities of the vehicle. Driving tests in the real world are a big challenge because fatal accidents involving autonomous vehicles may happen during the test. Digital simulation can reduce the cost and time and avoid potentially dangerous situations for validation of Advanced Driver-Assistance Systems (ADAS) and autonomous driving. All those increasing uses of traffic simulation for many studies highlight the importance of a good understanding and modelling of human driving behavior.

7.1 Conclusion on our contributions

In this thesis, the works are focused principally on human driving models, consisting of car-following and lane change models. We have also proposed a model for generation of car-collisions in numerical simulation, for the validation of autonomous vehicle features in dealing with traffic accidents.

Classical models in microscopic traffic modelling started with concepts of physics based on the classical Newtonian mechanics, and then augmented by behavioral aspects and driving rules. This kind of physical models have already been completely integrated in many traffic

simulators, such as SUMO, Aimsun, Vissim, etc. We have firstly investigated the classical models in Chapter 3, aiming at proposing a novel approach of car-collision generation method in numerical simulation. We have derived and distinguished different driver profiles, including *aggressive*, *inattentive* and *normal* drivers, based on the real traffic from the NGSIM data-set. Within this first work, we proposed a calibration mechanism for a chosen car-following model (IDM model), using a genetic algorithm. The objective of the calibration of the IDM model is for representing the different real driver profiles in numerical simulation. We then proposed to increase the percentage of 'extreme' profiles (aggressive and inattentive) in numerical simulation, in order to generate car-collisions. This work has been experimented based on the first period 15-minute data of the NGSIM 101 data-set. We have then validated our proposed approach using the second 15-minute period data. One of the important results we obtained in this work is the characterization of the relationship between the ratios of these two driver profiles over the whole drivers population, and the car-collision occurrences counts. We have also classified the occurred car-collisions and analyzed their severity, in particular with respect to the relative speed between the cars which cause the collisions. Furthermore, we have investigated the severity of the generated collisions, and find that the generated car-collisions involved between an aggressive driver as a leader and an inattentive driver as a follower occupy the most frequent collision occurrences, meanwhile the generated car-collisions involved between two inattentive drivers are the most severe ones.

In the other chapters of this thesis, we focused on artificial intelligence algorithms which have already demonstrated their efficiency in data-based modelling, during the last years. These algorithms provide alternative methods for modelling human driving behaviors. With this objective, we investigated and applied some machine learning algorithms for modelling human driving behavior, including a model for lane change using Reinforcement Learning (in Chapter 4) and a model combining car-following and lane changing behavior prediction using Long Short-Term Memory (LSTM) (in Chapter 5).

In Chapter 4, we proposed a Q-learning model for the modelling of human lane change decisions. The model demonstrates good performances in mimicking the human lane changing decisions (up to 95%). Moreover, the model uses numerical simulation to complete the data with unknown situations. We observed a percentage of 13% of additional traffic situations created by the traffic simulation environment. Training in a simulated traffic environment allows the improvement of the model performance. In Chapter 5, we presented a LSTM neural network model combining car-following and lane-changing on road networks. In this work, we proposed several models with different input designs to investigate the best one. The derived model has shown a good performance on both predicting the longitudinal speed and the lateral position. Moreover, the obtained results show that this model has a better performance than the classical physical model IDM in terms of the accuracy of replicating the car-following behavior. We experimented firstly the proposed models on the NGSIM 101 data-set, and then on the

HighD 101 data-set. We have observed that the use of relative variables between vehicles as inputs (input design 2) gives the best result compared to the other input designs. All our tested models have significant performances in the accuracy of the prediction of longitudinal speed and lateral position simultaneously.

At last, in Chapter 6, we implemented the proposed models with different combinations and some adaptations in SUMO, for simulating the whole traffic, where all the private cars are controlled by our models, while other kinds of vehicles including trucks and motorcycles are controlled by the default model in SUMO. The objective in this part was to compare the models and see the performance of traffic simulation in both the NGSIM 101 and the HighD data-sets. Traffic simulation with our proposed car-following model using neural networks, and with our lane changing model using reinforcement learning gives the best results in simulating the traffic of the NGSIM dataset. The models using artificial intelligence provide a better performance, even in the simulation of all the cars in the traffic for the NGSIM dataset. However, as indicated in section 6.1, there exists large differences between the NGSIM data-set and the HighD data-set. The simulation using the proposed models calibrated with the NGSIM dataset is not so good for replicating the HighD traffic. The NGSIM-calibrated IDM model is less sensitive to data, and it can be used to simulate unknown traffic like the HighD one. However, it requires a re-calibration with the maximum vehicle speed in traffic.

7.2 Perspectives

Modelling and simulation of realistic traffic is an essentially difficult problem. In this thesis, we investigated and proposed some microscopic traffic models, calibrated and validated with the NGSIM 101 and the HighD data-sets, and tested in numerical simulation with SUMO. Observation and analysis of real traffic data being the first work for the traffic modelling, the acquisition of big amounts of traffic data with different traffic conditions and scenarios, is to be considered in order to improve the efficiency of the models. It is then important to also improve the process of traffic data collection, under different traffic conditions, different networks, different driver profiles, etc. We note that the NGSIM traffic data have been collected with NGVIDEO (NGSIM, 2020), a customized software application developed for extracting the vehicle trajectory data from the video. The techniques of extracting vehicle trajectories from video are available by some methods using neural network models, consisting of camera calibration methods, and multi-objects tracking methods. We have tested some algorithms for automatic extraction of vehicle trajectories from videos, using a combination of an object-detection algorithm, Yolo5 (Jocher, 2020) and an object-tracking algorithm, DeepSort (Wojke, Bewley, and Paulus, 2017), with the code referenced by Mikel Brostrom (Brostrom, 2020). This application of extracting vehicle trajectory from video is presented in the Appendix C. Using this kind of methods, we can succeed extract more traffic data providing vehicle trajectories

data from video in order to enrich the traffic data-set with different traffic conditions, such as fluid dense, congested, full congested.

Furthermore, one of the important questions for the developed models is: are they able to reproduce a traffic which is very different from the one used for their calibration ? (The NGSIM 101 data-set includes congested traffic where many driver interactions are present, while, the HighD data-set is recorded from a less congested and lighter traffic, where the interactions between drivers are not very frequent). Actually, in Chapter 6 we have seen that, for physical models like the IDM model, the response to the question can be positive, even though some macroscopic calibration with the new data-set is sometimes needed (calibration of the maximum speed in our case). We have also seen that machine learning models are in general more dependent on the data than the physical ones, which makes the reproduction of other traffics more difficult. The perspectives in this direction is to develop mixed models (i.e. physical and data-based) able to benefit from the advantages of both physical and data-based models, without suffering from their disadvantages. The question is then to find the good mix !

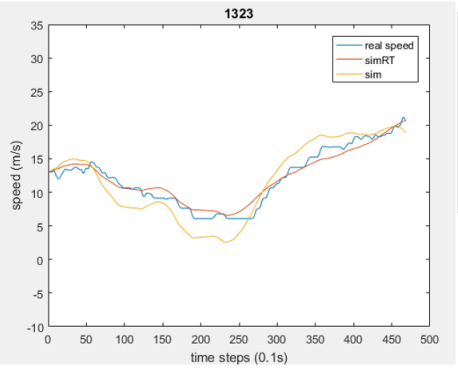
Let us now give some perspectives for the developed models. First, regarding our proposed car-collision generation method, we define three different driver profiles. Then the car-collision generation method is applied artificially by increasing the proportion of aggressive and inattentive drivers in traffic. This work is only an experiment based on the NGSIM 101 data-set. An interesting work would be to test the method with another data-set (the HighD one for example), derive different driver profiles, and try to generate car-collisions by numerical simulations. For our proposed lane change model using Q-learning and online calibration of the NGSIM data-set, we have obtained some good performances. However, as well known, with the development of reinforcement learning, deep reinforcement learning, allowing continuous approximations would further enhance model performance, and it provides the possibility to consider more variables in the input of the model. Some new deep reinforcement learning algorithms have been developed recently, such as the Deep Double Q-learning (Van Hasselt, Guez, and Silver, 2016), A3C (Mnih et al., 2016), DDPG (P. Wang, H. Li, and Chan, 2019) and PPO (Ye et al., 2020). It is then interesting to test such new algorithms for modelling lane change for vehicular traffic. In the other side, for the model using LSTM neural network, we found that the inputs using relative variables between the target vehicle and its surroundings vehicles can improve the model prediction accuracy of target vehicle's speed and lateral position. It would be interesting to consider these variables in the model of lane changing using reinforcement learning. Finally, for the LSTM model, it performs well in replicating real human driving behavior, but the structure of the model, which requires consideration of 50 sequences in the vehicle history trajectory, makes the model difficult to implement in a traffic simulator, which also poses a challenge in terms of computation time. A simpler model structure with similar performance needs to be considered in the future.

Appendix A

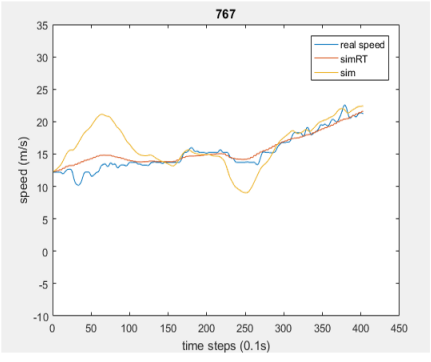
Complements on Chapter 3

A.1 Comparison of original IDM model and the IDM model with extension of reaction time

In Figure A.1, the result of the speed from calibrated IDM with a modelled reaction time and the speed from dataset for two drivers is shown. In Figure A.2, the result of the longitudinal position simulated by the calibrated IDM (before adding reaction time parameter and after adding reaction time parameter) are given. By these two figures, we can find that the calibrated IDM with the reaction time parameter represents the real human driving behaviour more efficiently.



ID	1323
T (s)	2.2
$\underline{s}0$ (m)	9.9
$\underline{v}0$ (m/s)	40
$n*\Delta t$	0.2
RMSE	0.95



ID	767
T (s)	0.3
$\underline{s}0$ (m)	4.6
$\underline{v}0$ (m/s)	40
$n*\Delta t$	0.1
RMSE	1.0

Figure A.1 – The comparison of original IDM model and the IDM model with the extension of reaction time, the calibration result for speed replication. The speed from calibrated IDM without reaction time (yellow), the speed from calibrated IDM with modelled reaction time (red) and real speed from dataset (bleu) for driver 1323 and driver 767. The tables show the optimal parameters of IDM with reaction time by genetic algorithm for driver 1323 and driver 767.

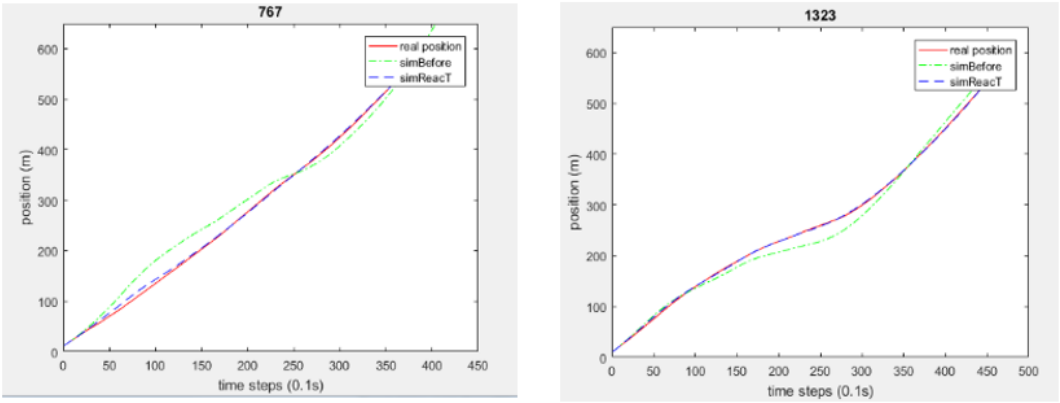


Figure A.2 – The comparison of original IDM model and the IDM model with the extension of reaction time, the calibration result for longitudinal position replication. The calibration of longitudinal position result for two drivers (id 767, id 1323) and their value parameters for IDM with the modelled reaction time. Three curves are given: the simulated position by calibrated IDM without reaction time (simBefore), the simulated position by calibrated IDM with reaction time (simReact) and real data (real position).

A.2 Calibration Result for specific drivers

The results of calibration for 4 specific groups of driver in the 15-minute data of NGSIM 101 dataset are given as follows: in Table A.1 for drivers of group 1, in Table A.2 for drivers of group 2, in Table A.3 for drivers of group 3, and in Table A.4 for drivers of group 4. These tables show the mean, the median, the standard deviation, the 25% quantile, and 75% quantile, of each parameter and for each group drivers. The tables give also the bounds for the genetic algorithm.

Parameters and error	Bounds	Mean	std	median	25%	75%
V0 (m/s)	[10, 40]	26.25	7.69	23.03	20.29	32.01
T (s)	[0.1, 4]	**1.08**	0.12	1.03	1.0	1.09
S0 (m)	[0.1, 10]	0.74	1.1	0.1	0.1	1.1
Reaction time (s)	[0.1, 2]	**0.45**	0.2	0.5	0.3	0.6
a (m/s2)	[0.1, 5]	4.49	0.95	5.0	4.52	5.0
b (m/s2)	[0.1, 5]	4.92	0.28	5.0	5.0	5.0
RMPSE position (%)		2.34	0.9	2.14	1.78	2.92

Table A.1 – Calibration results for the driver profiles of group 1 (2.5% 1st aggressive drivers).

Parameters and error	Bounds	Mean	std	median	25%	75%
V0 (m/s)	[10, 40]	25.95	8.94	22.22	19.22	36.42
T (s)	[0.1, 4]	**1.21**	0.21	1.16	1.02	1.34
S0 (m)	[0.1, 10]	0.39	0.67	0.1	0.1	0.33
Reaction time (s)	[0.1, 2]	**0.55**	0.16	0.5	0.5	0.68
a (m/s2)	[0.1, 5]	4.69	0.92	5.0	5.0	5.0
b (m/s2)	[0.1, 5]	4.87	0.61	5.0	5.0	5.0
RMPSE position (%)		1.03	2.09	1.55	2.7	1.03

Table A.2 – Calibration results for the driver profiles of group 2 (2.5% 2nd aggressive drivers).

Parameters and error	Bounds	Mean	std	median	25%	75%
V0(m/s)	[10, 40]	22.33	8.81	19.8	15.75	24.16
T (s)	[1, 4]	**2.38**	0.71	2.43	1.8	2.83
S0 (m)	[0.1, 10]	3.8	3.84	1.94	0.1	7.56
Reaction time (s)	$[tr_i - 0.3, tr_i + 0.3\ ]$	**2.16**	0.61	2.2	2.0	2.38
a (m/s2)	[0.1, 5]	3.86	1.46	4.7	2.94	5.0
b (m/s2)	[0.1, 5]	4.86	0.67	5.0	5.0	5.0
RMPSE position (%)		1.72	0.99	1.41	1.13	2.11

Table A.3 – Calibration results for the driver profiles of group 3 (2.5% 1st inattentive drivers).

Parameters and error	Bounds	Mean	std	median	25%	75%
V0 (m/s)	[10, 40]	25.18	9.38	21.94	18.02	32.89
T (s)	[1, 4]	**1.98**	0.58	2.05	1.48	2.47
S0 (m)	[0.1, 10]	3.65	3.73	2.02	0.1	7.83
Reaction time (s)	$[\ tr_i - 0.3, tr_i + 0.3]$	**1.73**	0.49	1.8	1.72	1.9
a (m/s2)	[0.1, 5]	3.99	1.64	5.0	3.59	5.0
b (m/s2)	[0.1, 5]	4.85	0.6	5.0	5.0	5.0
RMPSE position (%)		1.68	0.9	1.43	1.15	1.89

Table A.4 – Calibration results for the driver profiles of group 4 (2.5% 2nd inattentive drivers).

A.3 Regression of the number of simulated car-collisions for first 15 minutes data

A.3.1 Linear regression

In order to find the relationship between simulates crashes number and the rate of specific profile drivers for the result shown in Table 3.4, We obtain: $\hat{Y} = f(x_1, x_2) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2$, with $\beta_0 = -29.5569$, $\beta_1 = 1.4068$ and $\beta_2 = 1.3468$. The obtained root mean square error RMSE $= 9.7838$. In Figure A.3, the number of crashes for each percentage of two drivers profiles is presented, as well as the plane obtained by regression. This result allows us to estimate the proportion between aggressive and inattentive driver profiles generating the same number of

crashes:

$$\beta_2/\beta_1 = 13/14 \approx 1.$$

Although the linear regression is satisfying with RMSE = 9.7838, and $R^2 = 0.9045$, we can see on Figure A.3 that we need still improve the approximation.

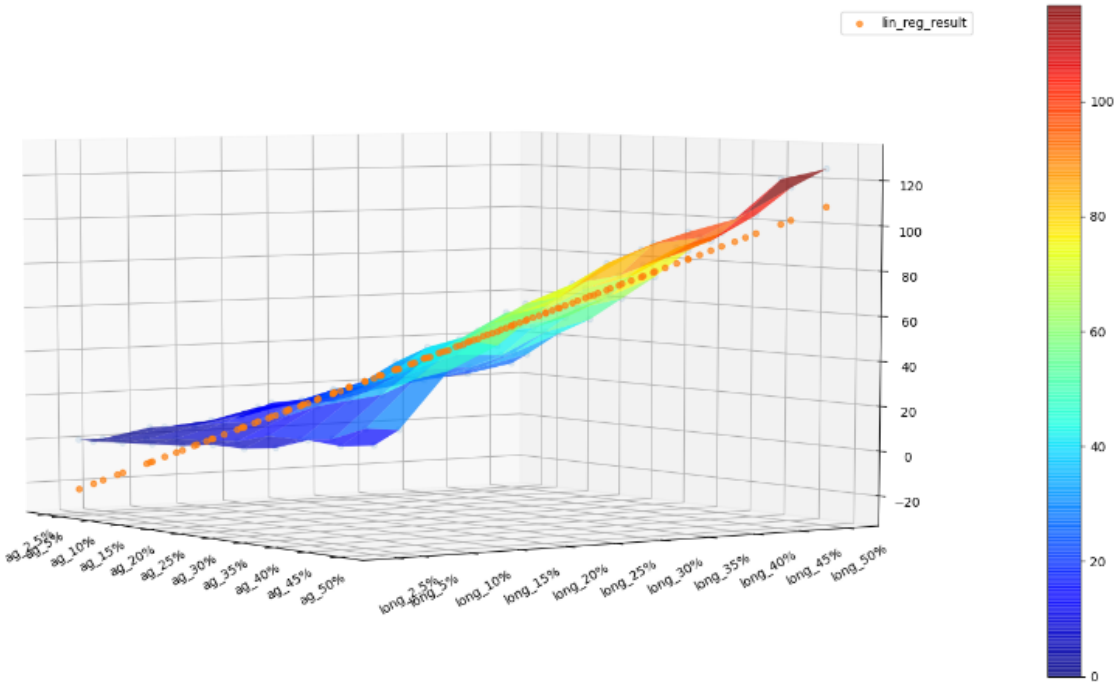


Figure A.3 – E1: Linear regression of number of simulated crashes, The suffix ag, long are the aggressive, inattentive driver profile respectively

A.3.2 First test of non-linear regression based on the first 15-minute data

In this subsection, we suppose the number of generated collisions based on the first 15-minute data has a non-linear relation with the percentage of each driver profile, where the formula is as below :

$$\hat{Y} = f(x_1, x_2) = \beta_0 + \beta_1 * (x_1 + x_2) + \beta_2 * (x_1 - x_2)^2 \tag{A.1}$$

The result of coefficients :

$$\hat{Y} = f(x_1, x_2) = -23.3532 + 1.3901 * (x_1 + x_2) - 1.4377 * (x_1 - x_2)^2 \tag{A.2}$$

where the coefficient of determination $R^2 = 0.9683$. RMSE (root mean square error)$= 5.630$

A.4 Regression result of the number of simulated car-collisions for second 15 minutes data

As the same, we investigated the regression function for the generated collision counts based on the second 15-minute data. By the same way as that for the first 15-minute data, we test firstly the linear regression, and then, a polynomial regression is tested, consisting of two different functions.

Linear regression :

We obtained : $\hat{Y} = f(x_1, x_2) = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2$, with $\beta_0 = -10.5262, \beta_1 = 1.537$ and $\beta_2 = 0.807$. where RMSE (root mean square error) = 6.16

Polynomial regression:

By the proposition in the Equation A.1, we find for the generated collision number, it respects to :

$$\hat{Y} = f(x_1, x_2) = -8.1540 + 1.2 * (x_1 + x_2) - 0.7 * (x_1 - x_2)^2 \tag{A.3}$$

where RMSE (root mean square error)$= 9.478$.

Appendix B

Complements on Chapter 4

B.1 Introduction

In this appendix, all other implementations of lane change models using reinforcement learning are presented.

B.2 Initial model using Q learning for lane change decision modelling

In the first proposition, we consider that the motivation of lane changing behavior usually is the speeding, that the driver wants to speed up. In this context, the reward of lane changing decision is the 'Speed gain', where the agent maximizes always the speed to take a decision of lane change. As presented in the section 4.3.3, the states and actions are used the same definitions. Unlike what is presented in 4.3.6, the reward is defined as :

$$\text{reward} = \begin{cases} \text{Speed Gain } (v_{t+1} - v_t) \\ \text{- 10} \quad \text{if crash or out-of-road situation} \end{cases} \tag{B.1}$$

The result of the evolution of the total reward of each episode is shown in Fig. B.1. The expectation of the total reward tends to converge after 40000 simulation episodes. Moreover, the crash and the out of road situations do not occur any more at the last episodes after about 41000; see Fig. B.2, where 1 means the presence of the collision or out-of-road and 0 means no presence of the event. However, using this model, the number of lane changes for the vehicles in the last 200 episodes ranges to once to 20 times, which is shown in Figure B.3. This result seems unrealistic that human driver can not change lane 20 times in a road section of 600m.

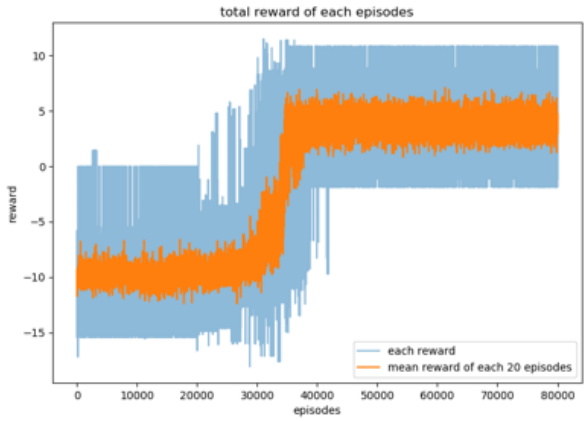


Figure B.1 – Total reward evolution. Blue line is the total reward for each episode. Orange line presents the average for each 20 episodes

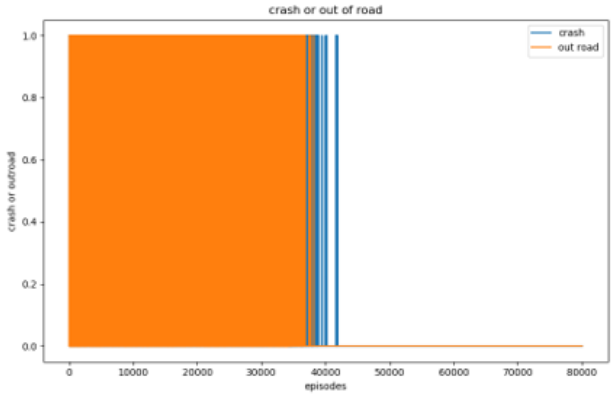


Figure B.2 – Crash and out of road occurrence evolution. 1 means presence of the event, 0 means no presence of the event

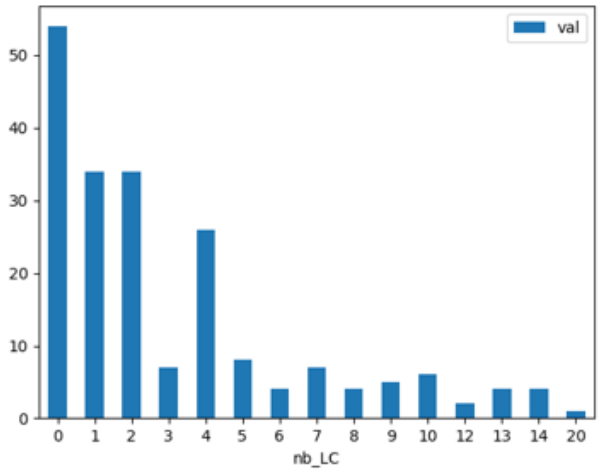


Figure B.3 – Distribution of number of lane changes for the vehicles in the last 200 episodes

Appendix C

Automated counting of vehicles in the traffic and extraction of vehicle trajectories : vehicle detection and vehicle-trajectory tracking from video

Historically, traffic data has been sourced from traffic cameras and sensors, as well as on-board sensors on vehicles. In section 2.7, we have presented the data-sets which we worked on for my PHD works, NGSIM and HighD. This kind of data-sets are very important for human driving behavior research, which contain vehicle trajectories, and are usually extracted from videos by object-tracking methods. The methods of object-tracking are widely studied in the computer vision domain, and have many advances using deep learning methods in the recent years. Here, we proposed and tested a pipeline for automated extraction of vehicle trajectories from videos, using a combination of an object-detection algorithm, Yolo5 (Jocher, 2020) and an object-tracking algorithm, DeepSort (Wojke, Bewley, and Paulus, 2017), with the code referenced by Mikel Brostrom (Brostrom, 2020). Furthermore, once we extract the objects trajectories in the video (in each image frame), the camera calibration is necessary to find the matrix, to convert the image coordinates to real world coordinates. We used an open-source software for the camera calibration, named as T-calibration, provided by an European project funded by Horizon 2020 (Lund University, 2018). As a result, we made a complete application for counting automatically different type of vehicles, including cars, trucks, motorcycles, bicycles in the traffic and extraction of vehicle trajectories from recorded traffic videos. This application can be used for the acquisition of a large-scale traffic data-set with different traffic conditions and for further research on human driving behavior. The pipeline has four steps:

1. Collect traffic video

2. Customization of detection zone and vehicle counting line
3. Yolo5 + DeepSort for multi-object tracking
4. Camera calibration to find the matrix for converting the coordinates in the video to real world coordinates

C.1 Collection of traffic video

We are interested in the mixed traffic with a bicycle path, in which we can find cars, trucks motorcycles, bicycles and pedestrians. The selected location for collect of traffic video is near the Pont de Bir Hakeim in Paris for a section of Voie Georges Pompidou along the Seine River. In Figure C.1, we show the location of camera detection zone in Google Map. The camera real view is presented in Figure C.2. The validated detection road section is about 120m in length, containing a one-direction normal vehicle lane and bicycle lanes for two directions. The duration of the video we have token is 13 minutes during the rush hour at one afternoon in the week. The video frequency is 25Hz.

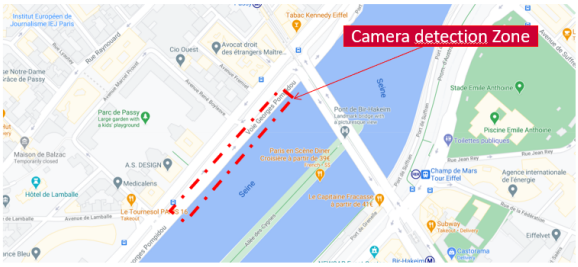


Figure C.1 – Camera Detection Zone

Figure C.2 – Camera real view

C.2 Customization of detection zone and vehicle counting line

In the application we made, the customization of detection zone and vehicle counting line is implemented to facilitate the utilisation of the application. We need to choose a detection zone in order to ignore the objects in the other zone and the counting lines can be drawn by the user to choose the counting area. Here we give an example in Figure C.3 for the counting line and in Figure C.4 for the detection zone.

Figure C.3 – Detection lines for the count of different type of vehicles

Figure C.4 – Detection Zone for extraction vehicle trajectories

C.3 Yolo5 + DeepSort for multi-object tracking

In multi-object tracking (MOT) problem, tracking-by-detection is a popular method where objects are firstly detected in each frame and represented as bounding boxes, then by the result of object detection, tracking is the assignment of the same objects from the previous frame to the current frame. In other words, we would like to detect all objects in the video, and during the video, one object need have only one identification from its first appearance to its disappearance in the video.

The object detection algorithms using deep learning can be classified into two groups, classification based algorithms and regression-based algorithms. You only look once (YOLO) is a state-of-the-art, real-time object detection algorithm and one of regression-based algorithms for predicting the classes and bounding boxes for the entire image at once. YOLO has already been evolved from YOLO (Redmon, Divvala, et al., 2016), YOLOV2 (Redmon and Farhadi, 2017), YOLOV3 (Redmon and Farhadi, 2018), YOLOV4 (Bochkovskiy, C.-Y. Wang, and Liao, 2020), to the most recent YOLOV5 (Jocher, 2020), which is the most performant and fastest version.

Simple Online and Realtime Tracking (SORT) (Bewley et al., 2016) is an efficient approach to multiple object tracking with a focus on simple, effective algorithms. In SORT, a Kalman filter is used to firstly give the estimation and prediction of objects in the current frame by knowing the previous positions. This prediction information can be used for the correction of the position given by the detection algorithm. Then, in assigning new detections in the current frame to existing targets observed in previous frames, each target's bounding box geometry is estimated by predicting its new location in the current frame. The assignment cost matrix is then computed as the intersection-over-union (IOU) distance between each detection and all predicted bounding boxes from the existing targets. The assignment is solved optimally using the Hungarian algorithm. The DeepSort (Wojke, Bewley, and Paulus, 2017) is an improved version of SORT including appearance information extracted by deep neural networks. This deep appearance descriptor is offline pre-trained on a large-scale person re-identification data-

set. During the tracking on real time, the deep appearance information is also an input for Hungarian algorithm to optimise the objects assignment.

For this step, we used the code referenced by Mikel Brostrom (Brostrom, 2020). The YOLOV5 has been trained using a large object detection data-set, COCO data-set (Lin et al., 2014), in which, all objects we want to detect in the traffic are included. The deep appearance descriptor in DeepSort is pre-trained using MOT challenge dataset (Leal-Taixé et al., 2015). We observed that this appearance descriptor pre-trained on person data-set is effective for other type of objects we want to detect. We show the result of YOLOV5 + DeepSort for the multi-object tracking in our collected video in Figure C.5.

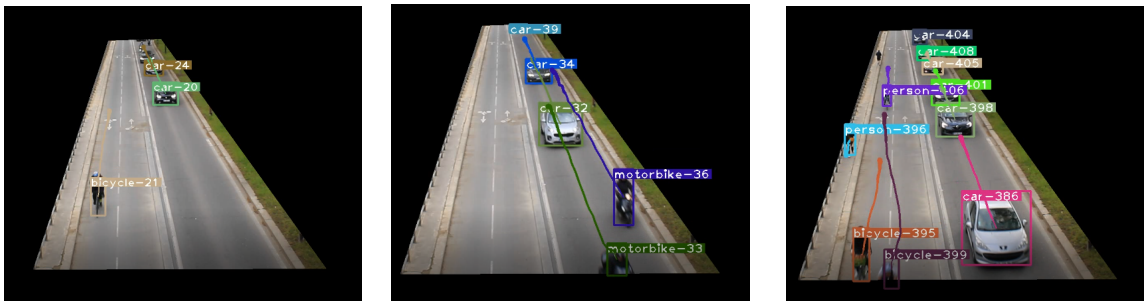


Figure C.5 – Result of using YOLOV5 + DeepSort

C.4 Camera calibration to find the matrix for converting the coordinates in the video to real world coordinates

Once we apply the YOLOV5 + DeepSort for the tracking for whole of our collected video, the vehicle trajectories in the image can be obtained by the position of the center of the bounding box in each frame. However, to convert the objects coordinates in the image to real world image, the calibration of the camera needs to be applied to find a matrix for the transformation of image coordinates to real world coordinates. We used an open-source software for the camera calibration, named as T-calibration, provided by an European project funded by Horizon 2020 (Lund University, 2018). By this application, we can get automatically the calibration result of camera by giving the reference points in the camera view image and the orthographic image of the detection zone. We show the reference points in the camera image in Figure C.6, and the referent points in the orthographic image in Figure C.7.

Figure C.6 – The reference points in the camera view image. The reference points in this image are indicated in the orthographic image

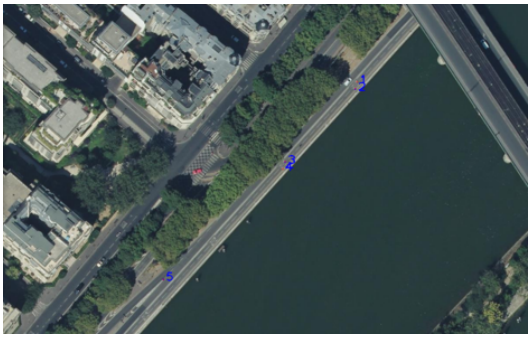


Figure C.7 – The orthographic image of the road section from Google Earth. The reference points in the camera view image are indicated in this image

C.5 Preliminary Results

We will give some preliminary results by our collected video. As a result, we extracted the vehicle trajectories from video and provided a .sqlite format table including all the vehicle positions by time. The vehicle trajectories are obtained in Figure C.8. In Figure C.9, we show the vehicle speed distribution for different types of vehicles.

Figure C.8 – Vehicle trajectories extraction presented in the orthographic Google Earth image

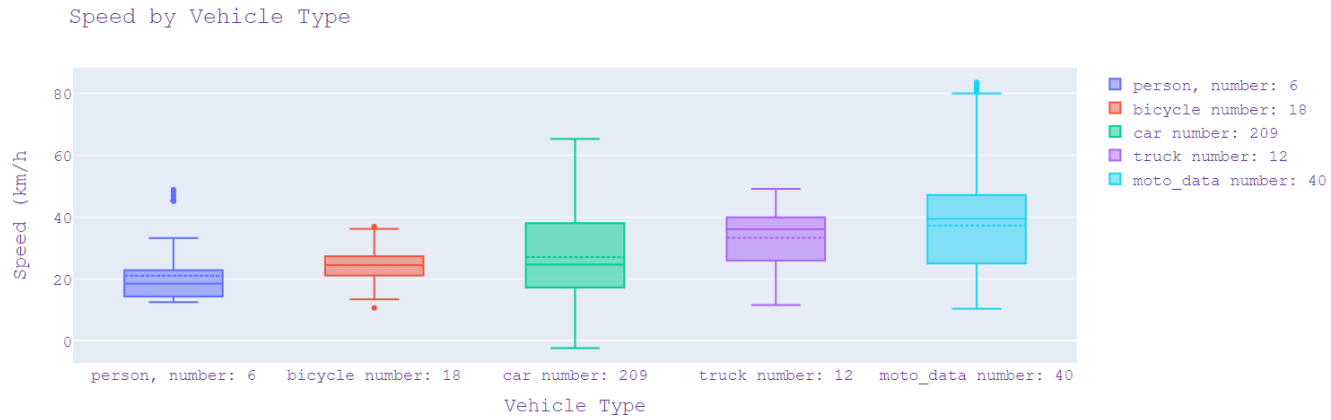


Figure C.9 – Different vehicle speed distribution

C.6 Perspectives

This work is tested and implemented with another PHD student, Yeltsin VALERO, working on Microscopic modelling of Personal Displacement Equipment (PDE) under mixed traffic and shared space conditions, with the objective to prepare the data-set for further using in the projects for studying the human driving behavior in mixed traffic.

For this project, one challenge for measuring the performance of our method is the lack of ground-truth data. In future work, we will evaluate the learned models using human annotations as ground-truth. In the other hand, change the view of camera is important for better detection. We can use the view like in the HighD and NGSIM data-set. That will facilitate the camera calibration task and reduce the calibration error to make the transformation of vehicle coordinates in image to real world coordinates.

List of Figures

List of Algorithms

List of Tables

List of References

Abdulhafedh, Azad (2017). Road traffic crash data: an overview on sources, problems, and collection methods. *Journal of transportation technologies* 7.2, pp. 206–219.

Aghabayk, Kayvan, Majid Sarvi, and William Young (2015). A state-of-the-art review of car-following models with particular considerations of heavy vehicles. *Transport reviews* 35.1, pp. 82–105.

Alexiadis, Vassili, James Colyar, John Halkias, Rob Hranac, and Gene McHale (2004). The next generation simulation program. *Institute of Transportation Engineers. ITE Journal* 74.8, p. 22.

Aljaafreh, Ahmad, Nabeel Alshabatat, and Munaf S Najim Al-Din (2012). Driving style recognition using fuzzy logic. In *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*. IEEE, pp. 460–463.

Altché, Florent and Arnaud de La Fortelle (2017). An LSTM network for highway trajectory prediction. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 353–359.

argoverse (2019). *Argoverse*. URL: https://www.argoverse.org/tasks.html.

AVSimulation (2019). *SCANeR Studio driving simulator*. URL: https://www.avsimulation.com/scanerstudio/.

Ayres, TJ, L Li, D Schleuning, and D Young (2001). Preferred time-headway of highway drivers. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*. IEEE, pp. 826–829.

Azevedo, Carlos Lima, João L Cardoso, and Moshe E Ben-Akiva (2018). Probabilistic safety analysis using traffic microscopic simulation. *arXiv preprint arXiv:1810.04776*.

Bando, Masako, Katsuya Hasebe, Akihiro Nakayama, Akihiro Shibata, and Yuki Sugiyama (1995). Dynamical model of traffic congestion and numerical simulation. *Physical review E* 51.2, p. 1035.

Barlovic, R, J Esser, K Froese, W Knospe, L Neubert, M Schreckenberg, et al. (1999). Online traffic simulation with cellular automata. In *Traffic and Mobility*. Springer, pp. 117–134.

Belbachir, Assia, Jean-Christophe Smal, Jean-Marc Blosseville, and Dominique Gruyer (2012). Simulation-driven validation of advanced driving-assistance systems. *Procedia-Social and Behavioral Sciences* 48, pp. 1205–1214.

Ben-Akiva, Moshe, Haris N Koutsopoulos, Tomer Toledo, Qi Yang, Charisma F Choudhury, Constantinos Antoniou, et al. (2010). Traffic simulation with MITSIMLab. In *Fundamentals of traffic simulation*. Springer, pp. 233–268.

Bewley, Alex, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft (2016). Simple online and realtime tracking. In *2016 IEEE international conference on image processing (ICIP)*. IEEE, pp. 3464–3468.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Brackstone and Mike Mark McDonald (1999). Car-following: a historical review. *Transportation Research Part F: Traffic Psychology and Behaviour* 2.4, pp. 181–196.

Brostrom, Mikel (2020). *Yolov5,DeepSort with PyTorch*. https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch.

Cameron, Gordon DB and Gordon ID Duncan (1996). PARAMICS—Parallel microscopic simulation of road traffic. *The Journal of Supercomputing* 10.1, pp. 25–53.

Casas, Jordi, Jaime L Ferrer, David Garcia, Josep Perarnau, and Alex Torday (2010a). Traffic simulation with aimsun. In *Fundamentals of traffic simulation*. Springer, pp. 173–232.

— (2010b). Traffic simulation with aimsun. In *Fundamentals of traffic simulation*. Springer, pp. 173–232.

Champion, Alexis, René Mandiau, Christophe Kolski, Alexandre Heidet, and Andras Kemeny (1999). Traffic generation with the SCANeR© II simulator: towards a multi-agent architecture. In *Driving Simulation Conference*, pp. 311–324.

Chandler, Robert E, Robert Herman, and Elliott W Montroll (1958). Traffic dynamics: studies in car following. *Operations research* 6.2, pp. 165–184.

Chao, Qianwen, Huikun Bi, Weizi Li, Tianlu Mao, Zhaoqi Wang, Ming C Lin, et al. (2019). A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. In *Computer Graphics Forum*. Wiley Online Library.

— (2020). A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. In *Computer Graphics Forum*. Vol. 39. 1. Wiley Online Library, pp. 287–308.

Chen, Kuan-Ting and Huei-Yen Winnie Chen (2019). Driving style clustering using naturalistic driving data. *Transportation research record* 2673.6, pp. 176–188.

Christoforou, Zoi, Simon Cohen, and Matthew G Karlaftis (2011). Identifying crash type propensity using real-time traffic data on freeways. *Journal of Safety research* 42.1, pp. 43–50.

Codevilla, Felipe, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy (2018). End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–9.

Dong, Ni, Helai Huang, and Liang Zheng (2015). Support vector machine in crash prediction at the level of traffic analysis zones: assessing the spatial proximity effects. *Accident Analysis & Prevention* 82, pp. 192–198.

Elvik, Rune (2013). A re-parameterisation of the Power Model of the relationship between the speed of traffic and the number of accidents and accident victims. *Accident Analysis & Prevention* 50, pp. 854–860.

Erdmann, Jakob (2014). Lane-changing model in SUMO. *Proceedings of the SUMO2014 modeling mobility with open data* 24, pp. 77–88.

Fagnant, Daniel J and Kara Kockelman (2015). Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice* 77, pp. 167–181.

Fellendorf, Martin and Peter Vortisch (2010a). Microscopic traffic flow simulator VISSIM. In *Fundamentals of traffic simulation*. Springer, pp. 63–93.

— (2010b). Microscopic traffic flow simulator VISSIM, pp. 63–93.

Gandrez, Clara, Fabrice Mantelet, Lu Zhao, Aoussat Ameziane, Francine Jeremie, and Eric Landel (2019). Modelization of human risk feeling during near-crashed situations in autonomous vehicle.

Gazis, Denos C, Robert Herman, and Richard W Rothery (1961). Nonlinear follow-the-leader models of traffic flow. *Operations research* 9.4, pp. 545–567.

Gers, Felix A, Jürgen Schmidhuber, and Fred Cummins (1999). Learning to forget: Continual prediction with LSTM.

Gipps, Peter G (1981). A behavioural car-following model for computer simulation. *Transportation Research Part B: Methodological* 15.2, pp. 105–111.

— (1986). A model for the structure of lane-changing decisions. *Transportation Research Part B: Methodological* 20.5, pp. 403–414.

Grewal, Mohinder S and Harold J Payne (1976). Identification of parameters in a freeway traffic model. *IEEE Transactions on systems, man, and cybernetics* 3, pp. 176–185.

He, Zhengbing (2017). Research based on high-fidelity NGSIM vehicle trajectory datasets: A review. *Research Gate*, pp. 1–33.

Helly, Walter (1959). Simulation of bottlenecks in single-lane traffic flow.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). Long short-term memory. *Neural computation* 9.8, pp. 1735–1780.

Hoel, C., K. Wolff, and L. Laine (2018). Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2148–2155.

Horni, Andreas, Kai Nagel, and Kay W Axhausen (2016). *The multi-agent transport simulation MATSim*. Ubiquity Press.

Hutchison, Casidhe, Milda Zizyte, Patrick E Lanigan, David Guttendorf, Michael Wagner, Claire Le Goues, et al. (2018). Robustness testing of autonomy software. In *2018 IEEE/ACM 40th International Conference on Software Engineering: Software Engineering in Practice Track (ICSE-SEIP)*. IEEE, pp. 276–285.

Itkonen, Teemu H, Jami Pekkanen, Otto Lappi, Iisakki Kosonen, Tapio Luttinen, and Heikki Summala (2017). Trade-off between jerk and time headway as an indicator of driving style. *PloS one* 12.10.

Jiang, Rui, Qingsong Wu, and Zuojin Zhu (2001). Full velocity difference model for a car-following theory. *Physical Review E* 64.1, p. 017101.

Jocher, Glenn (Oct. 2020). *ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements*. https://github.com/ultralytics/yolov5. Version v3.1.

Júnior, Jair Ferreira, Eduardo Carvalho, Bruno V Ferreira, Cleidson de Souza, Yoshihiko Suhara, Alex Pentland, et al. (2017). Driver behavior profiling: An investigation with different smartphone sensors and machine learning. *PLoS one* 12.4, e0174959.

Jurewicz, Chris, Amir Sobhani, Jeremy Woolley, Jeff Dutschke, and Bruce Corben (2016). Exploration of vehicle impact speed–injury severity relationships for application in safer road design. *Transportation research procedia* 14, pp. 4247–4256.

Kalra, Nidhi and Susan M Paddock (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice* 94, pp. 182–193.

Kesting, Arne, Martin Treiber, and Dirk Helbing (2007). General lane-changing model MOBIL for car-following models. *Transportation Research Record* 1999.1, pp. 86–94.

Khan, Zawar H and T Aaron Gulliver (2018). A macroscopic traffic model for traffic flow harmonization. *European Transport Research Review* 10.2, pp. 1–12.

Kingma, Diederik P and Jimmy Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koopman, Philip and Michael Wagner (2016). Challenges in autonomous vehicle testing and validation. *SAE International Journal of Transportation Safety* 4.1, pp. 15–24.

— (2017). Autonomous vehicle safety: An interdisciplinary challenge. *IEEE Intelligent Transportation Systems Magazine* 9.1, pp. 90–96.

Kotusevski, G and KA Hawick (2009). A review of traffic simulation software.

Krajewski, Robert, Julian Bock, Laurent Kloeker, and Lutz Eckstein (2018). The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125.

Krajzewicz, Daniel, Jakob Erdmann, Michael Behrisch, and Laura Bieker (2012). Recent development and applications of SUMO-Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements* 5.3&4.

Krauß, Stefan (1998). Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics. PhD thesis. Dt. Zentrum für Luft-und Raumfahrt eV, Abt. Unternehmensorganisation und …

Laris, Michael (2018). Waymo launches nation's first commercial self-driving taxi service in Arizona. *The Washington Post* 5.

Leal-Taixé, Laura, Anton Milan, Ian Reid, Stefan Roth, and Konrad Schindler (2015). Motchallenge 2015: Towards a benchmark for multi-target tracking. *arXiv preprint arXiv:1504.01942*.

Lillicrap, Timothy P, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, et al. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, et al. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*. Springer, pp. 740–755.

Lindorfer, Manuel, Christoph F Mecklenbraeuker, and Gerald Ostermayer (2018). Modeling the imperfect driver: Incorporating human factors in a microscopic traffic model. *IEEE Transactions on Intelligent Transportation Systems* 19.9.

Liu, Jiajia and Jianhao Liu (2018). Intelligent and connected vehicles: Current situation, future directions, and challenges. *IEEE Communications Standards Magazine* 2.3, pp. 59–65.

Lopez, Pablo Alvarez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, et al. (2018). Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 2575–2582.

Lu, Zhengyang, Ting Fu, Liping Fu, Sajad Shiravi, and Chaozhe Jiang (2016). A video-based approach to calibrating car-following parameters in VISSIM for urban traffic. *International journal of transportation science and technology* 5.1, pp. 1–9.

Lund University, Sweden (2018). *T-Analyst - semi-automated tool for traffic conflict analysis*. https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5c2c9951c&appId=PPGMS.

Luo, Yuanfu, Panpan Cai, Yiyuan Lee, and David Hsu (2020). Simulating Autonomous Driving in Massive Mixed Urban Traffic. *arXiv preprint arXiv:2011.05767*.

Marina Martinez, C., M. Heucke, F. Wang, B. Gao, and D. Cao (Mar. 2018). Driving Style Recognition for Intelligent Vehicle Control and Advanced Driver Assistance: A Survey. *IEEE Transactions on Intelligent Transportation Systems* 19.3, pp. 666–676.

Maurya, Akhilesh Kumar, Sanhita Das, Shreya Dey, and Suresh Nama (2016). Study on speed and time-headway distributions on two-lane bidirectional road in heterogeneous traffic condition. *Transportation Research Procedia* 17, pp. 428–437.

Meiring, Gys Albertus Marthinus and Hermanus Carel Myburgh (2015). A review of intelligent driving style analysis systems and related artificial intelligence algorithms. *Sensors* 15.12, pp. 30653–30682.

Mnih, Volodymyr, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, et al. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PMLR, pp. 1928–1937.

Mohammad, Saifuzzaman and Zuduo Zheng (2014). Incorporating human-factors in car-following models: a review of recent developments and research needs. *Transportation research part C: emerging technologies* 48, pp. 379–403.

Moridpour, Sara, Majid Sarvi, and Geoff Rose (2010). Lane changing models: a critical review. *Transportation letters* 2.3, pp. 157–173.

Nes, Nicole van, Jonas Bärgman, Michiel Christoph, and Ingrid van Schagen (2019). The potential of naturalistic driving for in-depth understanding of driver behavior: UDRIVE results and beyond. *Safety Science* 119, pp. 11–20.

NGSIM (2020). *Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data*. https://catalog.data.gov/pl/dataset/next-generation-simulation-ngsim-vehicle-trajectories-and-supporting-data.

nuscenes (2019). *Nuscenes*. URL: https://www.nuscenes.org/.

Pakusch, Christina, Gunnar Stevens, Alexander Boden, and Paul Bossauer (2018). Unintended effects of autonomous driving: A study on mobility preferences in the future. *Sustainability* 10.7, p. 2404.

Pourabdollah, Mitra, Eric Bjärkvik, Florian Fürer, Björn Lindenberg, and Klaas Burgdorf (2017). Calibration and evaluation of car following models using real-world driving data. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 1–6.

Pütz, Andreas, Adrian Zlocki, Julian Bock, and Lutz Eckstein (2017). System validation of highly automated vehicles with a database of relevant traffic scenarios. *situations* 1, E5.

R2020b, Matlab (2020). *Optimization Toolbox*. URL: https://fr.mathworks.com/help/gads/genetic-algorithm-options.html (visited on 03/31/2021).

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788.

Redmon, Joseph and Ali Farhadi (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271.

— (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Rosenblatt, Frank (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review* 65.6, p. 386.

Ross, Stéphane, Geoffrey Gordon, and Drew Bagnell (2011). A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635.

Salem, Habib Haj, Nadir Farhi, and Jean Patrick Lebacque (2013). Risk index model building: application for field ramp metering safety evaluation on urban motorway traffic in Paris. *Cybernetics and Information Technologies* 13.4, pp. 33–41.

Savitzky, Abraham and Marcel JE Golay (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry* 36.8, pp. 1627–1639.

Savolainen, Peter T, Fred L Mannering, Dominique Lord, and Mohammed A Quddus (2011). The statistical analysis of highway crash-injury severities: a review and assessment of methodological alternatives. *Accident Analysis & Prevention* 43.5, pp. 1666–1676.

Schaal, Stefan, Auke Ijspeert, and Aude Billard (2003). Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences* 358.1431, pp. 537–547.

Schulman, John, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schwarting, Wilko, Javier Alonso-Mora, and Daniela Rus (2018). Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*.

Shi, X, YD Wong, MZF Li, and C Chai (2018). Key risk indicators for accident assessment conditioned on pre-crash vehicle trajectory. *Accident Analysis & Prevention* 117, pp. 346–356.

SHRP2-NDS (2016). *The Strategic Highway Research Program 2—Naturalistic Driving Study @ONLINE*. URL: https://insight.shrp2nds.us/query/index.

Singh, Santokh (2015). *Critical reasons for crashes investigated in the national motor vehicle crash causation survey*. Tech. rep.

Singiresu S. Rao, University of Miami (2020). *Engineering Optimization: Theory and Practice, 5th edition*. John Wiley Sons, Inc., 2020.

Sjoberg, Katrin, Peter Andres, Teodor Buburuzan, and Achim Brakemeier (2017). Cooperative intelligent transport systems in Europe: Current deployment status and outlook. *IEEE Vehicular Technology Magazine* 12.2, pp. 89–97.

SUMO (2019a). *Netedit documentation*. URL: https://sumo.dlr.de/docs/NETEDIT.html (visited on 02/04/2020).

— (2019b). *simulation of collision in SUMO*. URL: https://sumo.dlr.de/docs/Simulation/Safety.html (visited on 12/09/2019).

— (2021). *SUMO website*. URL: https://www.eclipse.org/sumo/ (visited on 04/26/2021).

SUMO safety, 2021 (n.d.). https://sumo.dlr.de/docs/Simulation/Safety.html. Accessed: 2021-10-04.

Sutton, Richard S. and Andrew G. Barto (2018). *Reinforcement Learning: An Introduction*. Cambridge, Massachusetts: MIT Press.

Suweda, I Wayan (2016). Time headway analysis to determine the road capacity. *Jurnal Spektran* 4.2.

Thiemann, Christian, Martin Treiber, and Arne Kesting (2008). Estimating acceleration and lane-changing dynamics from next generation simulation trajectory data. *Transportation Research Record* 2088.1, pp. 90–101.

Toledo, Tomer, Haris N Koutsopoulos, and Moshe Ben-Akiva (2007). Integrated driving behavior modeling. *Transportation Research Part C: Emerging Technologies* 15.2, pp. 96–112.

Treiber, Martin, Ansgar Hennecke, and Dirk Helbing (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E* 62.2, p. 1805.

Treiber, Martin and Arne Kesting (2013a). Microscopic calibration and validation of car-following models–a systematic approach. *Procedia-Social and Behavioral Sciences* 80, pp. 922–939.

— (2013b). Traffic flow dynamics. *Traffic Flow Dynamics: Data, Models and Simulation, Springer-Verlag Berlin Heidelberg*.

Treiber, Martin, Arne Kesting, and Dirk Helbing (2006). Delays, inaccuracies and anticipation in microscopic traffic models. *Physica A: Statistical Mechanics and its Applications* 360.1, pp. 71–88.

United Nations, 2018 (n.d.). https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html. Accessed: 2021-10-04.

Van Hasselt, Hado, Arthur Guez, and David Silver (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1.

Vasconcelos, Luıs, Luıs Neto, Sılvia Santos, Ana Bastos Silva, and Álvaro Seco (2014). Calibration of the Gipps car-following model using trajectory data. *Transportation research procedia* 3, pp. 952–961.

Wang, Junhua, Yumeng Kong, Ting Fu, and Joshua Stipancic (2017). The impact of vehicle moving violations and freeway traffic flow on crash risk: An application of plugin development for microsimulation. *PLoS one* 12.9, e0184564.

Wang, L., F. Ye, Y. Wang, J. Guo, I. Papamichail, M. Papageorgiou, et al. (2019). A Q-learning Foresighted Approach to Ego-efficient Lane Changes of Connected and Automated Vehicles on Freeways. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 1385–1392.

Wang, P., H. Li, and C. Chan (2019). Continuous Control for Automated Lane Change Behavior Based on Deep Deterministic Policy Gradient Algorithm. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1454–1460.

Wang, Qi, Zhiheng Li, and Li Li (2014). Investigation of discretionary lane-change characteristics using next-generation simulation data sets. *Journal of Intelligent Transportation Systems* 18.3, pp. 246–253.

Wang, Xiao, Rui Jiang, Li Li, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang (2017). Capturing car-following behaviors by deep learning. *IEEE Transactions on Intelligent Transportation Systems* 19.3, pp. 910–920.

Watkins, Christopher JCH and Peter Dayan (1992). Q-learning. *Machine learning* 8.3-4, pp. 279–292.

Waymo (2019). *Waymo Open Dataset*. URL: https://waymo.com/open/.

Wiedemann, Rainer (1974). Simulation des Strassenverkehrsflusses.

WINSUM, WIM VAN and Adriaan Heino (1996). Choice of time-headway in car-following and the role of time-to-collision information in braking. *Ergonomics* 39.4, pp. 579–592.

Wojke, Nicolai, Alex Bewley, and Dietrich Paulus (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*. IEEE, pp. 3645–3649.

Xie, Dong-Fan, Zhe-Zhe Fang, Bin Jia, and Zhengbing He (2019). A data-driven lane-changing model based on deep learning. *Transportation research part C: emerging technologies* 106, pp. 41–60.

Ye, Fei, Xuxin Cheng, Pin Wang, Ching-Yao Chan, and Jiucai Zhang (2020). Automated Lane Change Strategy using Proximal Policy Optimization-based Deep Reinforcement Learning. *arXiv preprint* arXiv:2002.02667.

Yousif, Saad and John Hunt (1995). Modelling lane utilisation on British dual-carriageway roads: effects on lane-changing. *Traffic engineering & control* 36.12, pp. 680–687.

Zhang, Xiaohui, Jie Sun, Xiao Qi, and Jian Sun (2019). Simultaneous modeling of car-following and lane-changing behaviors using deep learning. *Transportation research part C: emerging technologies* 104, pp. 287–304.

Zhang, Yi, Ping Sun, Yuhan Yin, Lin Lin, and Xuesong Wang (2018). Human-like autonomous vehicle speed control by deep reinforcement learning with double Q-learning. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, pp. 1251–1256.

Zhao, Lu, Nadir Farhi, Zoi Christoforou, and Nadia Haddadou (2020). A data-based traffic modeling approach for inter-vehicular collision simulation. In *Proceedings of 8th Transport Research Arena TRA 2020, April 27-30, 2020, Helsinki, Finland*. TRA.

— (2021a). Analysis of driver behavior and inter-vehicular collision : a data-based traffic modeling and simulation approach. *Journal of advanced transportation*.

— (2021b). Driver Behavior Analysis and Inter-Vehicular Collision Simulation Approach. In *Proceedings of ICITS002 2021: 15. International Conference on Intelligent Transportation Systems December 27-28, 2021 in Vienna, Austria*.

— (2021c). Imitation of Real Lane-Change Decisions Using Reinforcement Learning. *IFAC-PapersOnLine* 54.2, pp. 203–209.

— (2022). Long Short-Time Memory Neural Networks for Human Driving Behavior Modelling. In *Transport Research Arena, TRA Conference – Lisbon 2022*.

Zheng, Zuduo (2014). Recent developments and research needs in modeling lane changing. *Transportation research part B: methodological* 60, pp. 16–32.

Zhou, Bolei (Apr. 2020). *Lecture slides in Policy Optimization*.

Zhu, Meixin, Xuesong Wang, Andrew Tarko, et al. (2018). Modeling car-following behavior on urban expressways in Shanghai: A naturalistic driving study. *Transportation research part C: emerging technologies* 93, pp. 425–445.

Zhu, Meixin, Xuesong Wang, and Yinhai Wang (2018). Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies* 97, pp. 348–368.