Technische Universität Berlin

Martin Strunz

Design of a Cooperative Overtaking Algorithm for Platoons on Freeways

Bachelor Thesis in Computer Science

6 December 2020

Please cite as:
Martin Strunz, “Design of a Cooperative Overtaking Algorithm for Platoons on Freeways,” Bachelor Thesis (Bachelorarbeit), Institute of Telecommunication Systems, Technische Universität Berlin, Germany, December 2020.

TKN Telecommunication Networks Group

Fachgebiet Telekommunikationsnetze
Technische Universität Berlin, Germany

Einsteinufer 25 · 10587 Berlin · Germany

https://www.tkn.tu-berlin.de/

Design of a Cooperative Overtaking Algorithm for Platoons on Freeways

Bachelor Thesis in Computer Science

vorgelegt von

Martin Strunz

angefertigt in der Fachgruppe

Fachgebiet Telekommunikationsnetze

Institut für Telekommunikationsnetze
Technische Universität Berlin

Betreuer:	**Julian Heinovski**
Gutachter:	**Falko Dressler**
	Thomas Sikora
Abgabe der Arbeit:	**6. Dezember 2020**

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte fremde Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.

Berlin, den 6. Dezember 2020

Declaration

I hereby declare that the thesis submitted is my own, unaided work, completed without any unpermitted external help. Only the sources and resources listed were used. The independent and unaided completion of the thesis is affirmed by affidavit:

Berlin, 6 December 2020

(Martin Strunz)

Abstract

Platooning can improve road safety, optimize traffic flow, and raise the driver's comfort. However, when a platoon is stuck behind a slower vehicle, the total travel time is significantly increased. Overtaking as a whole platoon is a complicated maneuver and there is no communication-based algorithm for this task so far. Therefore, we design such a cooperative overtaking algorithm for platoons on freeways. The algorithm can decide, depending on the traffic situation, whether it is possible, useful, and safe to overtake. If that is the case, it executes the overtaking maneuver cooperatively without splitting up the platoon.

The algorithm requires little technical equipment that goes beyond standard platooning. In particular, we do not assume that other road users have a Vehicle-to-Vehicle (V2V) communication system installed.

We theoretically define the algorithm by the use of Finite State Machines (FSMs) and implement it into a simulation environment. Thorough testing of the algorithm within the simulation environment shows that it works not only as expected, but can also avoid or handle possibly dangerous traffic situations. We also benchmark our algorithm against a non-communication based algorithm, the best case (the platoon is not delayed), and the worst case (the platoon cannot overtake). Our results show that it competes well. We further analyze the behavior of our algorithm in a parameter study and show that the algorithm does not overreact to changes of neither scenario-specific nor algorithm-specific parameters.

Kurzfassung

Platooning kann die Verkehrssicherheit verbessern, den Verkehrsfluss optimieren und den Komfort des Fahrers erhöhen. Wenn allerdings ein langsameres Fahrzeug das Platoon aufhält, kann sich die Gesamtfahrzeit erheblich verlängern. Das Überholen als vollständiges Platoon ist jedoch ein kompliziertes Manöver, für das es bisher keinen kommunikationsbasierten Algorithmus gibt. Aus diesem Grund entwickeln wir einen entsprechenden kooperativen Überhol-Algorithmus für Platoons auf Autobahnen. Der Algorithmus kann in Abhängigkeit von der Verkehrssituation entscheiden, ob das Überholen möglich, nützlich und sicher ist. Wenn das der Fall ist, kann er das Überholmanöver kooperativ ausführen, ohne dabei das Platoon aufzulösen.

Der Algorithmus setzt nur wenig technische Ausstattung voraus, die über die bei Platooning benötigte hinausgeht. Insbesondere setzen wir nicht voraus, daß die übrigen Verkehrsteilnehmer über ein V2V-Kommunikationssystem verfügen.

Wir definieren den Algorithmus mithilfe von endlichen Automaten und implementieren ihn in einer Simulationsumgebung. Umfangreiche Tests innerhalb der Simulationsumgebung zeigen, dass der Algorithmus nicht nur die erwarteten Aufgaben erfüllt, sondern auch möglicherweise gefährliche Verkehrssituationen vermeiden bzw. bewältigen kann. Außerdem vergleichen wir unseren Algorithmus mit einem anderen, nicht kommunikationsbasierten Algorithmus sowie mit dem besten (das Platoon wird nicht von einem langsameren Fahrzeug aufgehalten) und schlechtesten Fall (das Platoon kann nicht überholen). Er zeigt dabei eine gute Performance. Zudem analysieren wir sein Verhalten in einer Parameterstudie. Dabei zeigen wir, dass der Algorithmus auch bei der Änderung von szenario- bzw. algorithmusspezifischen Parametern weiterhin gute Ergebnisse liefert.

Contents

Chapter 1

Introduction

Researchers and car manufacturers are striving to make driving more enjoyable, cost-efficient and ecological. Based on advances in the field of autonomous driving, they have developed various new concepts for road traffic. One of them is platooning, which refers to road trains that consist of multiple vehicles driving at a very short inter-vehicle distance.

While platooning has a lot of advantages like improving road safety and increasing traffic flow, a platoon can also be seen as a single long road user, which is more difficult to maneuver. Therefore, a platoon can run into traffic situations that are more complicated to handle for a platoon than for a single vehicle. This may require complex cooperative maneuvers and cooperative overtaking is one of them: an overtaking maneuver that is automatically executed “is one of the most complex maneuvers for road automation” [1, p. 1643]. It requires a variety of decisions to be made, which is even more complex when not a single car has to overtake but a whole platoon.

Cooperative overtaking is not only a complicated but necessary task, since the platoon could loose some of its major advantages when being stuck behind a slower vehicle: it cannot drive at its optimal speed and the total travel time is extended. Thus, a platoon should assess the necessity and possibility of overtaking as soon as possible, especially in a freeway environment where overtaking is rather easy at least for individual vehicles.

One possible solution to this problem would be to split up the platoon before the maneuver, forcing each former platoon member to overtake individually. After finishing the individual maneuvers, the platoon could be reestablished on the original lane. Algorithms for overtaking by individual vehicles have already been proposed [2], [3]. Since reestablishing the platoon is a complicated maneuver itself, it would be ideal if the platoon had not been split up in the first place.

While the problems of lane-keeping and lane-changing are covered widely in scientific research, the “problem of automated overtaking has attracted less attention” [1, p. 1644]. According to our research, no cooperative overtaking algorithm for platoons that is based on Vehicle-to-Vehicle (V2V) communication has been developed so far.

Therefore, we propose a V2V-based cooperative overtaking algorithm for platoons in a freeway environment that does not need to split up the platoon. Its main features are to decide whether it is necessary and safe to overtake and, if so, to execute the maneuver without splitting up the platoon. The algorithm requires little technical equipment that goes beyond standard platooning. In particular, we do not assume that other road users have a V2V communication system installed. The algorithm is based on legal requirements in Germany. However, these can be overwritten or deactivated by setting other parameter values.

It is important to distinguish the term "cooperative overtaking" from similar concepts. In the context of this thesis, cooperation is utilized between the platoon members but not between the platoon and other road users. It must therefore be differentiated from Cooperative Overtaking Assistance (COA), where the overtaking intention of a vehicle is transmitted to its surrounding vehicles, and from Collaborative Overtaking Assistance (ClOA), where vehicles actively negotiate the overtaking maneuver, making it more easy or even possible [4]. The latter occurs, for example, when the overtaker and overtaken vehicles synchronize their velocities [5].

Chapter 2

Fundamentals

Before we describe the proposed overtaking algorithm in the next chapter, we start out with a short introduction into the concept of platooning. Then we present the work on which this algorithm is based on and other concepts that are closely related to overtaking and lane change maneuvers. This chapter concludes with a description of the simulation environment in which the proposed algorithm is implemented.

2.1 Platooning

Platooning is a part of modern Intelligent Transportation Systems (ITSs) [6]. Hoef, Johansson, and Dimarogonas [6] define platooning as a group of vehicles that are not physically but electronically coupled and thereby form a road train. A key feature of platooning is the very short inter-vehicle distance that platoon members maintain. The leading vehicle can be either controlled by a human driver or an Adaptive Cruise Control (ACC). While a Cruise Control (CC) can only keep a desired speed regardless of the traffic situation, the more advanced ACC can hold a fixed headway[1] to its predecessor and thereby influence the speed [7]. The trailing vehicles could automatically adopt decelerating and accelerating impulses from the leader by ACC. But since the platoon members drive at a very short inter-vehicle distance, this is not sufficient, because the reaction time of the followers would be too long. The Cooperative Adaptive Cruise Control (CACC) tackles this problem by adding V2V communication to an ACC. The followers, therefore, do not have to wait until the leader actually brakes, but receive a message from him that all platoon members will have to brake. This enables driving at short distances and also ensures string-stable acceleration and deceleration in regular operation (but emergency braking) [8].

Jia et al. [9] list four main advantages of platooning: firstly, roads can be used in a more efficient way, as the space between platoon members is reduced compared

[1]Defined in meters or seconds.

to the necessary safety distance for human controlled vehicles. Secondly, platooning can cut CO_2 and other emissions, as fuel consumption is reduced due to a decrease in air drag by driving at a very short inter-vehicle distance. However, this is an assumption that is not fully shared among the industry. Daimler Trucks reassessed its view on platooning in 2019 and stated that “fuel savings, even in perfect platooning conditions, are less than expected”.[2] Thirdly, platooning enhances road safety and the driver’s comfort. And fourthly, it improves vehicular networking, as the positions of the platooning vehicles are relatively fixed.

As there is, from a technical point of view, only one attentive driver needed for the leading vehicle, the drivers of the trailing vehicles can be allowed to rest or do other work.[3] Therefore, in the case of truck platooning, a driver can maintain his rest times without a second driver present while the truck continues to drive. This leads to a reduction of idle times and a better usage of truck assets.

Platooning is a research object particularly related, but not limited, to trucks. The Safe Road Trains for the Environment (SARTRE) project [10], which is funded by the European Commission, and the Program on Advanced Technology for the Highway (PATH) at the University of California, Berkeley [11] are two major platooning projects.

However, platooning also bears disadvantages: the necessity for inter-vehicle communication makes it vulnerable to hacking [12]. Another aspect, and at the same time of particular relevance to this thesis, is the fact that complicated situations can occur while driving which require complex cooperative maneuvers. It is relatively easy to operate a platoon in a dedicated platooning lane on a freeway, whereas overtaking of slower vehicles requires a complex cooperative maneuver. Corresponding algorithms have to handle these situations in a most efficient and, of course, safe way.

2.2 Related Work

Although no communication-based algorithm for cooperative overtaking has been developed so far, there are already algorithms for non-cooperative overtaking, i.e., for single vehicles. It is advisable to further analyze whether they could serve as a suitable basis for a cooperative version. An overtaking maneuver consists, among others, of two lane changes, thus we will analyze such algorithms as well. As the proposed algorithm needs to assess the safety of a possible overtaking maneuver as well, the analysis of work on risk assessment in a freeway environment is likewise of relevance to this thesis.

[2] https://media.daimler.com/marsMediaSite/en/instance/ko.xhtml?oid=42188247 (access date: 05/23/20)

[3] From a legal point of view, this can be a different situation, but this is not considered further here.

2.2.1 Autonomous Overtaking

Non-cooperative autonomous overtaking algorithms come closest to meet the goal of cooperative autonomous overtaking. So they are a promising basis for such a new algorithm. We present four different approaches: trajectory planning, imitation of human behavior, deep reinforcement learning, and multi-agent approaches.

The first approach is to plan trajectories of the involved road users. Düring et al. [2] propose a modularized trajectory planning algorithm which is intended for cooperative maneuver planning including overtaking. The algorithm distinguishes influenceable and non-influenceable road users and solves a possible conflict between them in five steps: generate reachable target points for each vehicle, decide which of these are safe target points, compute safe trajectories towards these points, select the best maneuver, i.e., trajectory combination, and finally execute this maneuver. The best maneuver is determined by evaluating, e.g., collision checks of the trajectories and applying cost functions to promising trajectories. A disadvantage of the algorithm: it only achieves real-time capabilities if the number of parameters is small, thus it can not deal with a high number of safe target points. Petrov and Nashashibi [1] consider overtaking as a tracking problem of virtual trajectories and develop an according kinematic model. Their system works autonomously, i.e., it does not rely on roadway markings or V2V communication and could be used as a back-up, when V2V or Vehicle-to-Everything (V2X) communication is failing.

Gong et al. [13] developed a two-part decision making model for overtaking. Firstly, a multilevel microscopic scene model describes the actual traffic situation, which is based on measurement information. Secondly, a hierarchical state machine is used for the decision making process for overtaking. The latter divides the overtaking behavior into different states and levels, which are inspired by human driving. It also uses a radial basis function neural network to "learn and fit the person-specific driving characteristics" [13, p. 12]. A disadvantage of the model: it can not take all possible traffic scenarios into account, which leads to unpredictable behavior.

Another approach to use human behavior as a basis for autonomous overtaking is proposed by Naranjo et al. [14]. They use Global Positioning System (GPS) data, a V2X network environment, and fuzzy controllers to imitate the behavior. Their subdivision of the overtaking phases ("first lane change to the contiguous left lane, circulation in the left lane, second lane change to the right lane" [14, p. 441]) will be a basis for our own definition (see Section 3.1).

Hoel, Wolff, and Laine [15] use deep reinforcement learning for decision making not only in the context of lane changing and overtaking, but also for general purpose decision making. They trained and tested their Deep Q-Network in a simulation environment and successfully simulated an overtaking maneuver on a road with oncoming traffic. Another approach for decision making, which uses model predictive control, is presented by Nilsson and Sjöberg [16]. Their algorithm can decide in which lane the vehicle should drive during the respective time step and whether an overtaking maneuver is useful and possible, whereas the actual lane change execution is not part of the algorithm.

Multi-agent approaches are proposed by Lam and Katupitiya [17] and Groza, Iancu, and Marginean [18]. The former define agent-based discrete events, which are chained to model more complex maneuvers that can handle also hazardous situations. Their simulation results "indicate that some additional fail-safes could be implemented", but their approach "overall worked as expected" [17, p. 1157]. The latter define and formalize four different agents for different overtaking maneuvers, e.g., a *piggy agent* that follows another overtaker.

2.2.2 Lane Changing

Lane change algorithms constitute a very good basis for designing an algorithm for (cooperative) overtaking maneuvers because, as described in more detail in Section 3.1, an overtaking maneuver consists, among others, of two lane changes.

Hsu and Liu [19] design a lane change maneuver for platoons. The leader defines when and in which direction to change lanes, whereas the followers execute the maneuver themselves. This concept will be incorporated in our own cooperative overtaking algorithm. But while Hsu and Liu [19] describe the operational level including lateral control, we exclude this translation of driving decisions into specific vehicle commands like steering and acceleration.

Ulbrich and Maurer [20] propose another approach. They use the outputs of two signal processing networks as input for a Partially Observable Markov Decision Process (POMDP) decision making algorithm. The networks define whether a lane change is possible and beneficial and are intended to simplify the POMDP model. The authors define three different regions of interest around the ego vehicle to assess the lane change possibility. We adopt the idea of the regions and develop it further. In another paper [21], the authors describe in more detail the assessment whether a lane change is possible and beneficial and name the influencing factors that are important for both decisions. We adopt some of these decision criteria. In particular, they define a formula for determining how hard a vehicle approaching from behind must brake if the platoon changes to its lane. Based on this formula, we develop a more detailed one.

Samiee et al. [22] propose a lane change collision avoidance system that consists of three multi-layer controllers: decision making (possibility of a lane change), path planning (generating trajectories), and vehicle control (steering of the vehicle). They focus on the fact that a traffic situation can change during lane changing compared to the initial decision to execute the maneuver. For this reason, they explore possible threats during lane changing. Their idea of assessing the traffic situation during lane changing is incorporated in our algorithm.

Simulation of Urban MObility (SUMO) in which we test our proposed algorithm also uses a lane change model. It distinguishes different levels of lane change reasons: strategic (necessary lane changes to reach the desired destination), cooperative (helping other road users), tactical (avoiding to drive behind slower vehicles), legal (obligation to drive on the right hand side of the road), and remote controlled (changing the vehicle's behavior by the user of the simulation). Our algorithm will incorporate tactical and legal reasons for lane changing.

2.2.3 Traffic Risk Assessment

Assistants for traffic risk assessment can be used in a variety of traffic situations, including lane changing and overtaking. These systems can inform the driver about their findings and are often part of Advanced Driver Assistance Systems (ADASs). In the context of lane changing, it is of particular importance to assess the risk of vehicles approaching from behind and driving in the front. Different measurands have been proposed for this. The *deceleration to safety time* "denotes the deceleration that has to be applied to a vehicle to maintain a certain safety time (with respect to another vehicle)" [23, p. 612] . Noh and An [24] describe the risk metrics *time to collision* and *minimal safety margin*. We will use a (differently defined) *minimal safety margin* for the area in front of the platoon and the formula mentioned in Section 2.2.2 for vehicles in the rear.

Vehicle detection is required for traffic risk assessment. Different technical approaches can be used for this (e.g., cameras, radar, lidar [1], [23], [24]), but as we focus on the actual cooperative overtaking algorithm, it is of no particular relevance to this thesis which of these technologies are used. On these grounds, we do not further investigate the advantages and disadvantages of these systems.

2.3 Simulation Environment

In order to simulate the behavior of the proposed algorithm, we will use a simulation environment based on SUMO [25] and Plexe [8]. SUMO is "an open source, highly portable, microscopic and continuous road traffic simulation package designed to handle large road networks".[4] Its purpose, with respect to this thesis, is to simulate the road traffic on a freeway (both the platoon and additional vehicles). SUMO can write extensive output streams which will be very useful for our performance evaluation. It can also simulate sublanes (the SL2015 sublane model [26]), meaning more than one car can be in a lane at the same time and longitudinal position.

[4] https://sumo.dlr.de/docs/

This is of importance because thus it becomes possible to simulate two vehicles simultaneously merging from different lanes to the same target lane. This creates an unsafe situation which has to be handled by the proposed algorithm.

Plexe enables "realistic studies of platooning concepts" [8, p. 53] within SUMO by integrating "all the necessary components to study platooning ranging from controller models [, including CACC,] to maneuvers" [8, p. 53]. The Plexe API for Python provides an easy access to this functionality.[5] It does not support V2V communication, but we will need this functionality for exchanging messages between the leader and his followers for a cooperative execution of the overtaking maneuver. Therefore, we will have to develop a simple version of such a system ourselves.

Plexe already possesses an implementation of a *coordinated lane change* algorithm for platoons,[6] but it is not V2V-communication-based and platooning vehicles can use all information available in the simulation. This, however, makes the algorithm not realistic. The omnipresence of information could be understood as V2V or even V2X communication, but the majority of road users do not support such technologies today. The Plexe lane change algorithm is not activated by default. If activated, according to the Plexe API for Python,[7] the leader will check whether the platoon could gain speed by changing lanes. If that is the case, he will assert whether a lane changing maneuver is possible and, if so, execute it. We will compare our proposed algorithm to this integrated implementation within the performance evaluation in Chapter 4.

As of Plexe's version 2.1, its lane change algorithm is still experimental[8] and not yet published. In fact, the platoon did not always successfully change lanes cooperatively during our tests. In a few scenarios only the leader changed lanes, leaving his followers in the old lane. They accelerated, overtook the leader on his right hand side, and then crashed into the next car on that lane. At least at one time (see Figure 2.1), the platoon changed lanes from the right to the middle one, despite the fact that another vehicle was also changing from the left lane to the middle one. The situation was resolved by an emergency braking of the vehicle approaching from the left lane.[9] These problems occurred both with the LC2013 and the SL2015

[5]https://github.com/michele-segata/plexe-pyapi (access date: 11/11/20)

[6]A lane change algorithm is not a complete overtaking algorithm because it does not take into account, for example, how long a planned overtaking maneuver would take. Nevertheless, it can be used as a comparison, since an overtaking maneuver also consists, among others, of two lane changes.

[7]https://github.com/michele-segata/plexe-pyapi/blob/master/plexe/plexe.py#L443 (access date: 09/21/20)

[8]https://plexe.car2x.org/features/ (access date: 11/11/20)

[9]The proposed algorithm can detect such a dangerous situation and aborts the lane change maneuver by returning to the right lane.

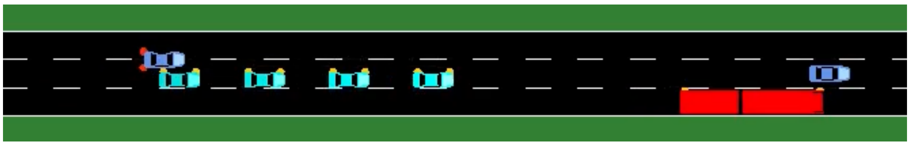


Figure 2.1 – A dangerous situation during a coordinated lane change by Plexe's integrated lane change algorithm: the platoon changes lanes to the middle lane at the same time as another vehicle changes lanes to this lane. The situation was solved by an emergency braking of the other vehicle.

lane change models of SUMO.[10] For the sake of comparability, we will only choose scenarios that did not result in such crashes.

[10] Even if the more simple LC2013 model would have worked, it is no option: SL2015 is required for the proposed algorithm. Using LC2013 for Plexe would only lead to a different behavior of the randomized traffic compared to SL2015, making the scenarios incomparable.

Chapter 3

The Proposed Cooperative Overtaking Algorithm

First of all, the tasks of the proposed overtaking algorithm have to be defined in the overall context of driving. Donges [27] distinguishes three different levels for driving tasks: the navigation tasks on the strategic level, guidance tasks on the tactical level, and stabilization tasks on the operational level. Ulbrich and Maurer [20, pp. 2063–4] apply these levels to a lane change maneuver. This idea is continued here and applied to an overtaking maneuver. On the strategic level the vehicle's total route is computed on a lane-level basis, for example by an A* path search algorithm. To keep the estimated time of arrival, overtaking maneuvers might be necessary when slower vehicles delay the platoon. But they are not incorporated in the planned route for a specific point in time.[11] On the tactical level the precomputed route is modified while driving based on the actual dynamic traffic situation, for instance by adding useful overtaking maneuvers. On the operational level the tactical decisions are translated into specific vehicle commands to control the actual driving behavior, e.g., latitudinal (steering) and longitudinal (speed) adjustments at a specified point in time. The proposed algorithm focuses on the tactical part of driving tasks.

Some assumptions are made in order to let the algorithm become not too complex and thereby obscure the view of its core elements. These assumptions concern the platooning vehicle, its surrounding infrastructure and the V2V system. A platooning vehicle is equipped with V2V and ACC (leader vehicle) or V2V and CACC (follower vehicle) capabilities. Otherwise platooning would be impossible. In addition, the vehicle is equipped with front and rear mid-range radar sensors that can measure the distance, speed, and position of objects with a distance of up to 160 m (front)

[11] In contrast to lane changes, which have to be planned before the start of the journey (e.g., the necessary lane change to exit a freeway).

and 80 m (rear).[12] Side cameras are responsible for detecting objects beside the vehicle on the neighboring lanes (see Figure 3.1).[13]

The vehicle can detect lanes automatically and change to a neighboring lane when told so (as described, for example, in [1], [23]). Furthermore, the platoon has already been formed in an optimal way, meaning, e.g., that the order of vehicles does not need to be changed and the desired speed does not exceed the technical capabilities of single members. Regarding the infrastructure, the proposed algorithm has to work in a freeway environment with the obligation to drive on the right hand side of the road.[14] By this means, not only oncoming traffic, but also crossroads, sublane overtaking, or pedestrians can be excluded. In addition, there is always a neighboring lane left of the platoon. This lane allows for overtaking from a legal point of view and is long enough to support an overtaking maneuver.

Details of the V2V communication system are not modeled within this thesis, e.g., a physical or data link layer will not be described. Messages can be sent during every simulation step and are either received in the next simulation step or received in one of the next simulation steps with an exponentially distributed reception probability (see Section 3.2).[15]

Some of the assumptions made are weak and could be dropped by rather small changes of the proposed algorithm. For instance, the assumption that there is always a neighboring lane can easily be dropped by assessing this with the use of cameras as proposed, for example, in [28].

The proposed algorithm will be modeled as Finite State Machines (FSMs) and the entire overtaking maneuver is discretized into short steps (which correspond to the steps in the simulation environment). In each step, every platoon member can be in only one state of the FSM. This design allows for a modular approach. If, e.g.,

[12] The stated values are based on mid-range radar sensors developed by Bosch. They fit well as one intended usage of the front version is ACC, and one of the rear version is lane change assistance. Source: https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/driver-assistance-systems/lane-change-assist/mid-range-radar-sensor-mrrrear/ (access date: 05/27/20).

[13] E.g., the autopilot of a Tesla can rely on eight surround cameras, twelve ultrasonic sensors, and a forward facing radar. Source: https://www.tesla.com/autopilot (access date: 11/20/20)

[14] This is based on German law and is also the default in the simulation environment.

[15] While the focus of this thesis is on a cooperative overtaking algorithm operating under a reliable V2V communication system, it can still function correctly when it becomes unreliable. Our assumption regarding an unreliable V2V communication system is that a message will never get lost, but can be received with a (substantial) delay. This is not a strong assumption, because a lost message could be sent again (which is not modeled here).

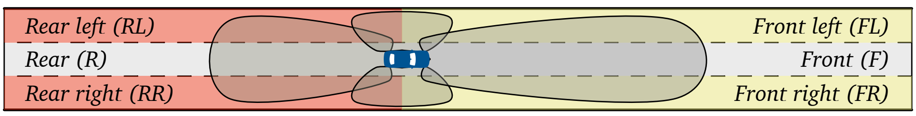


Figure 3.1 – Illustration of the platooning vehicle's sensor areas and the different areas of interest around it.

the hazard management during a lane change has to be changed, the corresponding state can simply be substituted.

To simplify the description of the algorithm, we define some terms in the following. Ulbrich and Maurer [20, p. 2064] use three regions of interest for a lane change around a vehicle: the rear left (RL), front left (FL), and front ego (FE) area. We define six areas of interest, with the front left (FL), rear left (RL), front right (FR), and rear right (RR) being the most important ones (see Figure 3.1). In addition, the *L* (*R*) area is defined as the combination of the *FL* and *RL* (*FR* and *RR*) area. Based on this, the closest vehicle in the *XY* area is called *XY* vehicle, its speed is v_{XY}. An overtaking maneuver always starts in the original lane, passing the slower vehicle will take place in the overtaking lane. The target lane of a lane change is either the original lane or the overtaking lane, depending on the direction of the lane change.

Following these introductory remarks, the proposed algorithm will be described below. This is followed by a brief description of its implementation in the simulation environment. The chapter concludes with a description of all traffic scenario test cases, which are intended to show the correct behavior and safety of the proposed algorithm.

3.1 Theoretical Design

To make the process of overtaking more tangible, it can be broken down into several phases. A typical subdivision consists of three phases: a lane change to the left, passing the vehicle to be overtaken, and a lane change back to the original lane [1], [14], [17]. In this classification, the very first phase is missing in which it is decided whether overtaking is useful, possible, and safe. Thus, this phase will be added to our own classification: (1) decision for overtaking, (2) lane change left, (3) passing the slower vehicle, and (4) lane change right (see Figure 3.2). The proposed algorithm must implement all four phases.

It is noticeable that phases 1 and 3 as well as 2 and 4 are similar. In phases 1 and 3, the decision must be made when to change lanes. It is a general property of the proposed algorithm that the leader is always responsible for making decisions. His followers only carry out the decisions and, as will be explained more in detail later, are not aware that they are participating in an overtaking maneuver. Instead, they only change lanes two times at the request of the leader. In phases 2 and 4, the

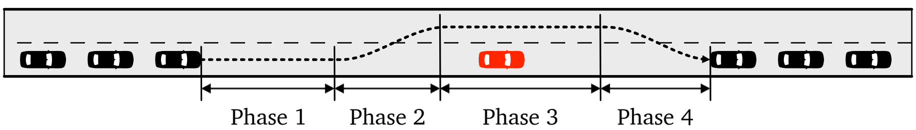


Figure 3.2 – Phases of an overtaking maneuver.

lane change will be attempted and finally carried out. This similarity of the phases will be of relevance for the algorithm's design. In the following, the overall concept of the algorithm and each state of the proposed FSMs are explained in detail.

In the **first phase**, the platoon is driving on a freeway lane and the leader is monitoring the *F* area. When he detects a vehicle in that area, he has to decide whether the platoon could and should overtake. The usefulness and possibility of overtaking will be determined by the leader himself as he has access to all relevant information (e.g., the platoon's desired speed, speed of the vehicle in front). If overtaking is useful and possible, he will change to phase two, otherwise he will stay in phase one and continue to monitor the *F* area. As this is a decision phase, the algorithm does not intervene actively in the concrete traffic situation.

In the **second phase**, the safety of the lane change has to be evaluated, and if positively conducted, the platoon has to execute the maneuver. To evaluate the safety, the leader requires additional information about the traffic situation around the platoon (e.g., the presence of other vehicles that prevent safe overtaking), which can be obtained from his followers. So the exchange of messages is necessary in this phase of the algorithm. As soon as the leader has received all relevant information, he will decide whether overtaking is safe and inform his followers accordingly. If the situation is considered unsafe, the platoon will stay in its lane and, after a backoff, the leader restarts the algorithm by going back to phase one. In the other case, the actual lane change is carried out by every member. As the platoon is driving in dynamic traffic, hazard detection is necessary [17, p. 1155] in order to be able to resolve potentially unsafe situations quickly. The proposed algorithm takes this into account, too.

When the platoon successfully changed lanes to the left, the **third phase** (the second decision phase) starts, i.e., to decide when to attempt to go back to the original lane. This is the case if there is no vehicle in the *FR* area that could and should also be overtaken. The evaluating set of rules does not differ from phase one and the leader can do this on his own again. If changing back is useful, the leader will change to phase four. Otherwise, he will stay in phase three and continue to monitor the *FR* area.

In the **fourth phase**, the safety of the second lane change has to be evaluated. This is done exactly in the same manner as in phase two, apart from the monitored area, which is now the right side. If a hazard is detected during the actual lane change, the platoon will abort the maneuver and the leader will stay in phase three. When the platoon successfully changed back to the original lane, the overtaking maneuver is completed. This concludes the proposed algorithm. The leader changes back to phase one and the algorithm starts over.

In order to explicitly describe the behavior of the algorithm, FSMs have to be designed for the overtaking maneuver from the leader's perspective and for a lane change from both the leader's and follower's perspective. These FSMs are described in detail in the following subsections.

3.1.1 Overtaking Maneuver: Leader's Perspective

The overtaking maneuver is only known to the leader, for this reason only his perspective needs to be considered. The corresponding FSM (see Figure 3.3) represents the aforementioned four phases: the *idle* and *vehicle ahead* states (phase one), the *lane change left* pseudo-state (phase two), the *passing* state (phase three), and the *lane change right* pseudo-state (phase four). The pseudo-states represent a sequence of states. They are introduced because, except for three parameters, the *lane change left* and *lane change right* maneuvers are identical. Describing them separately (see Section 3.1.2 and Section 3.1.3) makes the overtaking maneuver more comprehensible.

3.1.1.1 Idle State

This is considered as the initial state of the algorithm: the platoon is driving on a freeway lane at its desired speed and in optimal formation. The leader scans his own lane in front for vehicles with the equipped radar. When a vehicle is detected, he changes to the *vehicle ahead* state, regardless of its speed. Otherwise, he remains in the *idle* state.

3.1.1.2 Vehicle Ahead State

In this state the leader assesses whether it is useful and possible to overtake the vehicle in front. According to Ulbrich and Maurer [29, p. 991], the benefit of a lane change depends on the relative velocity gain on a neighboring lane, which in turn depends on the dynamic traffic situation and on infrastructure related information. Because of the assumptions made (e.g., the freeway is sufficiently long), infrastructure-related information is assumed to have no effect on the overall decision. For assessing the potential velocity gain in the neighboring lane on the left, the leader monitors the *FL* area for vehicles.[16] A neighboring lane is considered faster and therefore beneficial, if

$$\min(v_{\text{desired}}, v_{\text{limit}}) - v_{\text{F}} \geq v_{\Delta}, \tag{3.1}$$

where v_{desired} is the platoon's desired speed, v_{limit} is the speed limit of the neighboring lane, v_{F} is the speed of the vehicle in front, and v_{Δ} is the threshold. The minimum function describes the maximum speed the platoon could drive on the

[16]The *RL* area is of no relevance in this phase because a *RL* vehicle does not define the possible speed in the left lane. This vehicle could be close to the platoon and therefore be a safety risk when changing lanes. This is evaluated during the *lane change left* maneuver.

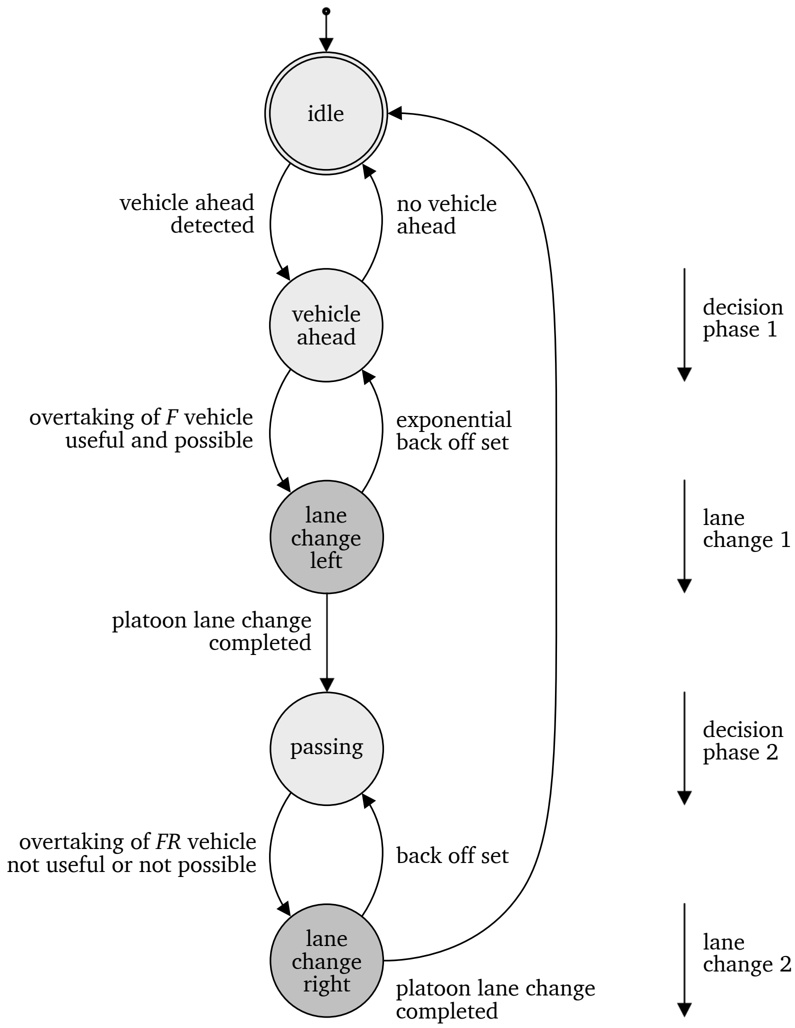


Figure 3.3 – FSM for the overtaking maneuver from the leader's perspective.

neighboring lane.[17] Therefore, we define

$$v_{\max} = \min\left(v_{\text{desired}}, v_{\text{limit}}\right), \tag{3.2}$$

which is used later on. v_{Δ} will be set to 2.7 m/s in the simulation because it is a required minimum speed difference between an overtaking and an overtaken vehicle

[17]A possibly detected vehicle in the *FL* area is not taken into account because it represents only a temporary speed restriction for the lane. If this vehicle changes to another lane, the platoon can drive with $v_{\max}$ again. Nevertheless, we will analyze the platoon's overtaking performance in Section 4.2.2, when this vehicle is taken into account as well.

according to German jurisprudence.[18] If Equation (3.1) is true, overtaking is useful. Now the leader has to assess whether overtaking is possible.

Ulbrich and Maurer [21, p. 977-978] define four influencing factors for a lane change being possible: 1) the dynamic traffic situation, 2) the infrastructure situation, 3) ability-induced skill restrictions, and 4) temporary skill restrictions. The dynamic traffic situation will be handled in the *assert maneuver area* state of the lane change maneuver (see Section 3.1.2.4). The infrastructure situation refers to determine whether the platoon is driving on a valid lane, to evaluate traffic signs and lane markings, if there is a valid neighboring lane, and whether traffic rules permit an overtaking maneuver. As infrastructure restrictions are omitted in this thesis, only traffic rules are taken into account. Ability-induced skill restrictions comprise limitations of the overtaking vehicle (e.g., the vehicle cannot handle oncoming traffic on a rural road). Skill restrictions refer to temporary reduced capabilities of the leader (e.g., a vehicle that is driving right behind the leader could limit his sensor capabilities). Ability-induced and temporary skill restrictions are not taken into account any further. Traffic rules are discussed in more detail below.

While a platoon could attempt an overtaking maneuver if its desired speed is greater than that of the vehicle to be overtaken, legal regulations restrict overtaking. Besides the already mentioned minimum speed difference between overtaking and overtaken vehicle of 2.7 m/s, the OLG Hamm ruled that an overtaking maneuver may take a maximum of 45 s.[19] It follows that an overtaking maneuver is possible if

$$t_{\text{overtaking}} \leq t_{\max}, \tag{3.3}$$

where $t_{\text{overtaking}}$ is the time needed for the maneuver and $t_{\max}$ is the threshold of 45 s. The concrete formula for $t_{\text{overtaking}}$ is derived below.

First of all, we define some necessary distances that are relevant for the following calculations (see Figure 3.4):

$$l_{\text{total}} = d_{\text{P}} + l_{\text{F}} + d_{\text{F\_safety}} + l_{\text{P}}, \tag{3.4}$$

where d_{P} is the distance to the vehicle in front,[20] l_{F} and l_{P} are the lengths of the vehicle in front and the platoon respectively, and $d_{\text{F\_safety}}$ is the required safety

[18] § 5 Abs 2 S 2 StVO: "Furthermore, overtaking is only permitted to those who drive at a significantly higher speed than the person being overtaken" (own translation). As this law does not define a concrete speed, the court ruling of the Higher Regional Court (Oberlandesgericht, OLG) Zweibrücken (case number: 1 SsRs 45/09, 1 Ss Rs 45/09, ECLI:DE:POLGZWE:2009:1116.1SSRS45.09.0A) was consulted: "A permitted overtaking at 'significantly higher speed than the one to be overtaken' is generally still present between trucks on a two-lane freeway if the difference is at least 10 km/h" (own translation).

[19] Case number: 4 Ss OWi 629/08, ECLI:DE:OLGHAM:2008:1029.4SS.OWI629.08.00.

[20] Which is at least the platoon's safety distance because the ACC will observe this distance.

distance of the vehicle in front. The latter is defined by

$$d_{\text{F\_safety}} = \max(v_{\text{F}} \cdot T_{\text{h}}, d_{\text{truck}}) \tag{3.5}$$

where T_{h} is the expected time headway of the slower vehicle and $d_{\text{truck}} = 50\,\text{m}$ is the minimum distance a truck has to obey on a German freeway.[21]

Then we compute the time needed for acceleration t_{a} from the platoon's speed v_{P} up to speed v. The acceleration is defined by

$$a = \frac{\mathrm{d}v}{\mathrm{d}t}, \tag{3.6}$$

where v is the velocity and t is the time. The platoon will start accelerating at its current speed v_{P} and stop accelerating at speed v. The duration of this acceleration is t_{a}. Integrating Equation (3.6) leads to

$$\int_{v_{\text{P}}}^{v} \mathrm{d}v = a_{\text{P}} \int_{0}^{t_{\text{a}}} \mathrm{d}t \tag{3.7}$$

$$\Rightarrow v - v_{\text{P}} = a_{\text{P}} \cdot t_{\text{a}} \tag{3.8}$$

$$\Rightarrow t_{\text{a}} = \frac{v - v_{\text{P}}}{a_{\text{P}}}, \tag{3.9}$$

where a_{P} is the platoon's constant acceleration.

Now we compute the distance the platoon drives in time t_{a}. The velocity is defined by

$$v = \frac{\mathrm{d}d}{\mathrm{d}t}, \tag{3.10}$$

[21] §4 Abs 3 StVO: "Anyone driving a truck with a gross vehicle weight over 3.5 t [...] must keep a minimum distance of 50 m from vehicles in front if the speed is more than 50 km/h" (own translation).

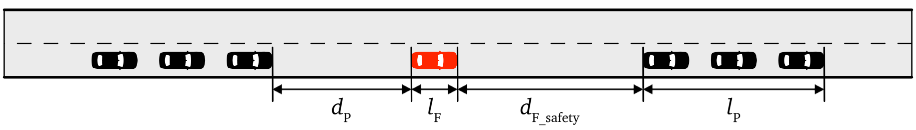


Figure 3.4 – Relevant distances for calculating the necessary time for an overtaking maneuver.

where d is the distance. The platoon will start accelerating at $t = 0$ and stop at t_a. With $v(t) = v_P + a_P \cdot t$ (see Equation (3.8)), integrating Equation (3.10) leads to

$$\int_0^{d_a} \mathrm{d}d = \int_0^{t_a} v \,\mathrm{d}t \tag{3.11}$$

$$= \int_0^{t_a} (v_P + a_P \cdot t)\,\mathrm{d}t \tag{3.12}$$

$$\Rightarrow d_a = v_P t_a + \frac{a_P}{2} t_a^2. \tag{3.13}$$

Now we assume that the platoon will accelerate during the whole overtaking maneuver ($t_a = t_{\text{overtaking}}$). The distance d_a the platoon will cover in this time is the length of the platoon, the slower vehicle, and the safety distances of both the slower vehicle and the platoon l_{total} plus the distance the slower vehicle drives in this time:

$$d_a \stackrel{!}{=} l_{\text{total}} + v_F \cdot t_{\text{overtaking}}. \tag{3.14}$$

With Equation (3.13) this leads to

$$v_P t_{\text{overtaking}} + \frac{a_P}{2} t_{\text{overtaking}}^2 = l_{\text{total}} + v_F \cdot t_{\text{overtaking}} \tag{3.15}$$

$$\Rightarrow 0 = t_{\text{overtaking}}^2 + \frac{2}{a_P}(v_P - v_F)\, t_{\text{overtaking}} - \frac{2}{a_P} l_{\text{total}} \tag{3.16}$$

$$t_{\text{overtaking}_{1,2}} = -\frac{v_P - v_F}{a_P} \pm \sqrt{\frac{(v_P - v_F)^2}{a_P^2} + \frac{2}{a_P} l_{\text{total}}}. \tag{3.17}$$

The negative case can be omitted as $t_{\text{overtaking}}$ has to be positive. With Equation (3.9), this leads to

$$\frac{v_{\text{overtaking}} - v_P}{a_P} = -\frac{v_P - v_F}{a_P} + \sqrt{\frac{(v_P - v_F)^2}{a_P^2} + \frac{2}{a_P} l_{\text{total}}} \tag{3.18}$$

$$\Rightarrow v_{\text{overtaking}} - v_P = -v_P + v_F + \sqrt{(v_P - v_F)^2 + 2a_P \cdot l_{\text{total}}} \tag{3.19}$$

$$\Rightarrow v_{\text{overtaking}} = v_F + \sqrt{(v_P - v_F)^2 + 2a_P \cdot l_{\text{total}}}. \tag{3.20}$$

This is the speed the platoon reaches when it accelerates during the whole overtaking maneuver. As the platoon would only drive as fast as v_{max} (see Equation (3.2)), this result only holds if $v_{\text{overtaking}} \leq v_{\text{max}}$. If $v_{\text{overtaking}} \leq v_{\text{max}}$, the platoon accelerates during the entire overtaking maneuver and overtaking is possible if

$$t_{\text{overtaking}} = \frac{v_{\text{overtaking}} - v_P}{a_P} \leq t_{\text{max}}. \tag{3.21}$$

If $v_{\text{overtaking}} > v_{\max}$, the platoon reaches $v_{\max}$ before the overtaking maneuver is completed. With Equation (3.13), Equation (3.9), and $v = v_{\max}$, the distance driven during acceleration is given by

$$d_a = \frac{v_P}{a_P}(v_{\max} - v_P) + \frac{1}{2a_P}(v_{\max} - v_P)^2 . \tag{3.22}$$

The distance $d_{\text{overtaking}}$ can also be defined as the sum of the distance d_a the platoon drives during acceleration to $v_{\max}$ and the distance d_c driven at $v_{\max}$ until the maneuver is completed:

$$d_{\text{overtaking}} = d_a + d_c, \tag{3.23}$$

where

$$d_c = v_{\max} \cdot t_c \tag{3.24}$$

is the distance driven at constant speed. As $d_{\text{overtaking}}$ is still the distance needed for overtaking, it can also be defined by

$$d_{\text{overtaking}} \stackrel{!}{=} v_F (t_a + t_c) + l_{\text{total}}. \tag{3.25}$$

With Equation (3.22), Equation (3.24), and Equation (3.25), we can now compute t_c (the time driven at constant speed):

$$v_F (t_a + t_c) + l_{\text{total}} = d_a + v_{\max} \cdot t_c \tag{3.26}$$

$$\Rightarrow v_F \cdot t_a + v_F \cdot t_c + l_{\text{total}} = d_a + v_{\max} \cdot t_c \tag{3.27}$$

$$\Rightarrow t_c (v_{\max} - v_F) = v_F \cdot t_a + l_{\text{total}} - d_a \tag{3.28}$$

$$\Rightarrow t_c = \frac{v_F \cdot t_a + l_{\text{total}} - d_a}{v_{\max} - v_F}. \tag{3.29}$$

With

$$t_a = \frac{v_{\max} - v_P}{a_P}, \tag{3.30}$$

$t_{\text{overtaking}}$ can be defined by

$$t_{\text{overtaking}} = t_a + t_c. \tag{3.31}$$

After inserting and simplifying, $t_{\text{overtaking}}$ can be computed by

$$t_{\text{overtaking}} = \frac{l_{\text{total}}}{v_{\max} - v_F}\left[1 + \frac{(v_{\max} - v_P)^2}{2a_P \cdot l_{\text{total}}}\right]. \tag{3.32}$$

In this case, overtaking is possible if

$$\frac{l_{\text{total}}}{v_{\max} - v_{\text{F}}} \left[1 + \frac{(v_{\max} - v_{\text{P}})^2}{2a_{\text{P}} \cdot l_{\text{total}}} \right] \leq t_{\max}. \tag{3.33}$$

The formulas so far only compute the time needed until the platoon is far enough away from the overtaken vehicle to safely change back to the original lane. Yet, the time t_{lc} to change back needs to be considered as well:[22]

$$t_{\text{lc}} = \frac{l_{\text{lane\_width}}}{v_{\text{lat}}} \tag{3.34}$$

where v_{lat} is the platoon's lateral speed and $l_{\text{lane\_width}}$ is the lane width. In conclusion, $t_{\text{overtaking}}$ can be computed by

$$t_{\text{overtaking}} = \begin{cases} \frac{1}{a_{\text{P}}} \left(v_{\text{F}} - v_{\text{P}} + \sqrt{(v_{\text{P}} - v_{\text{F}})^2 + 2a_{\text{P}} \cdot l_{\text{total}}} \right) + t_{\text{lc}} & \text{, if } v_{\text{overtaking}} \leq v_{\max} \\ \frac{l_{\text{total}}}{v_{\max} - v_{\text{F}}} \left[1 + \frac{(v_{\max} - v_{\text{P}})^2}{2a_{\text{P}} \cdot l_{\text{total}}} \right] + t_{\text{lc}} & \text{, if } v_{\text{overtaking}} > v_{\max}. \end{cases} \tag{3.35}$$

To avoid oscillation, both thresholds are raised[23] a little when the platoon is driving in the original lane or when changing to it. This ensures that a decision made and being executed is not reversed when the thresholds are just met in the decision phase.

If overtaking is useful and possible, the leader will change to the *lane change left* pseudo-state, otherwise he stays in the *vehicle ahead* state. The *lane change left* states are described in Section 3.1.2. If the vehicle in front is out of range (e.g., by leaving the freeway or accelerating), the leader will change back to the *idle* state.

3.1.1.3 Passing State

Just before entering this state, the platoon successfully changed lanes from the original one to the overtaking lane by passing through the *lane change left* states described in Section 3.1.2 or aborted a *lane change right* maneuver (see Section 3.1.2.10). Now the platoon is driving in the overtaking lane and the leader has to decide when to start changing back. This is considered useful if no vehicle in the *FR* area is detected that should be overtaken. As the algorithm will identify the slower vehicle as a vehicle that should (still) be overtaken, this assessment can start right after changing

[22] The time needed to change to the overtaking lane is omitted due to two reasons: first, this would only be necessary if the platoon reaches its safety distance to the slower vehicle before being in the overtaking lane, which is not always the case. Second, the overtaking threshold is already raised a bit (see Footnote 23).

[23] This means that the speed threshold of 2.7 m/s is increased and $t_{\text{overtaking}}$ is decreased, making overtaking a little bit more difficult.

lanes and before passing the slower vehicle. Therefore, the leader scans the *FR* area for vehicles. If a vehicle is detected, he measures its speed using his radar and decides with the same set of rules already defined in the *vehicle ahead* state whether it is useful and possible to overtake it.[24] When at least one of the equations

$$\min(v_{\text{desired}}, v_{\text{limit}}) - v_{\text{FR}} < v_{\Delta} \tag{3.36}$$

or

$$t_{\text{overtaking}} > t_{\text{max}} \tag{3.37}$$

is true, overtaking the *FR* vehicle is not useful. While the formulas are based on the *vehicle ahead* state, the distance d_{P} from the platoon to the slower vehicle, which is used in Equation (3.4), is replaced by $d_{\text{P\_new}}$: d_{P} is subtracted by the distance the platoon will drive longitudinally when changing lanes to the right and subtracted by the distance the platoon wants to drive in time t_{stay} in the original lane before attempting to overtake again (see Figure 3.5), which leads to $d_{\text{P\_new}}$:

$$d_{\text{P\_new}} = d_{\text{P}} - d_{\text{adjust}}, \tag{3.38}$$

where d_{adjust} is defined by

$$d_{\text{adjust}} = \left(\frac{l_{\text{lane\_width}}}{v_{\text{lat}}} + t_{\text{stay}} \right) (v_{\text{P}} - v_{\text{FR}}). \tag{3.39}$$

If $d_{\text{P\_new}}$ becomes negative, the platoon will stay in the overtaking lane. This adjustment of d_{P} avoids that the platoon will change lanes and shortly after completing or even during executing the maneuver will abort and change back to the overtaking lane.

If overtaking is not useful or possible anymore or when no vehicle is detected in the *FR* area, the leader will change to the *lane change right* pseudo-state (see Section 3.1.2).

[24] The formulas are the same, but the area of speed measurement is changing due to the fact that the platoon is now in the overtaking lane: v_{F} becomes v_{FR}.

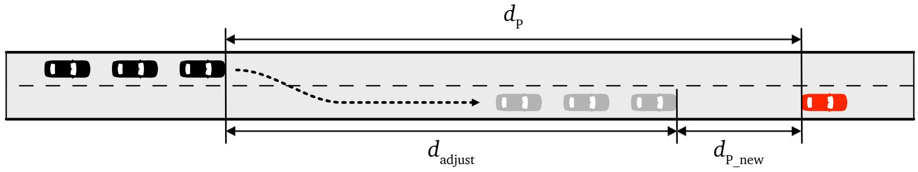


Figure 3.5 – The platoon is virtually moved forward to ensure that it will not overtake again immediately after changing back to the original lane.

3.1.2 Lane Change Maneuver: Leader's Perspective

The aforementioned *lane change left* and *lane change right* pseudo-states of the overtaking FSM are, in fact, a sequence of states and represent the lane change FSM of the leader. A lane change to the left is basically the same maneuver as one to the right, they only differ in three parameters: the area to be monitored in the front (*FL* vs. *FR* area) and in the rear (*RL* vs. *RR* area) and whether the reason for overtaking still exists vs. a new reason for overtaking appeared.[25] The following description of the FSM (see Figure 3.6) is based on a *lane change left* maneuver; the *lane change right* maneuver specifics are explained in the affected states.

From the leader's perspective, both lane changes are part of the overtaking maneuver (see Section 3.1.1) which means that there is no explicit start state. The first state of the lane change maneuver (*assert α and β areas*) can be reached either from the *vehicle ahead* or the *passing* state of the overtaking maneuver. This means that the entry condition for starting a lane change is the decision to do so made by the leader. During a lane change maneuver the leader first assesses whether his own *FL* and *RL* areas (or *FR* and *RR* areas, depending on the direction of the lane change) are free. If they are occupied, a lane change will not be safe and the leader aborts the maneuver. If they are free, he sends a request to his followers that they have to assess their own *FL* and *RL* (or *FR* and *RR*) areas. The followers send their information back to the leader, who can now evaluate the safety of the whole platoon's maneuver area. Again, if the area is occupied, a lane change is considered unsafe and the maneuver will be aborted. If the area is free, he orders his followers to change lanes. Then the actual lane change is executed by telling the vehicle's operational level to move in the appropriate direction. As already mentioned in the introductory section of this chapter, hazard management is necessary in this phase. If another vehicle is suddenly detected in the platoon's maneuver area (e.g., by merging into the same target lane), the lane change must be aborted to avoid a collision. If the slower vehicle becomes faster or is out of range, the maneuver can be aborted as well. When the leader has successfully reached the center of the target lane, he awaits a message from each of his followers that they also completed the lane change. After that, the platoon's lane change is completed, the leader informs his followers about this, and changes to the *passing* or *idle* state of the overtaking FSM, depending on the direction of the completed lane change.

3.1.2.1 Assert α and β Areas

This is the first state of the lane change FSM and can be entered either from the *vehicle ahead* or the *passing* state of the overtaking FSM. First of all, the leader assesses whether it is safe to overtake as a single car. Two areas on the neighboring

[25] For details see Section 3.1.2.6.

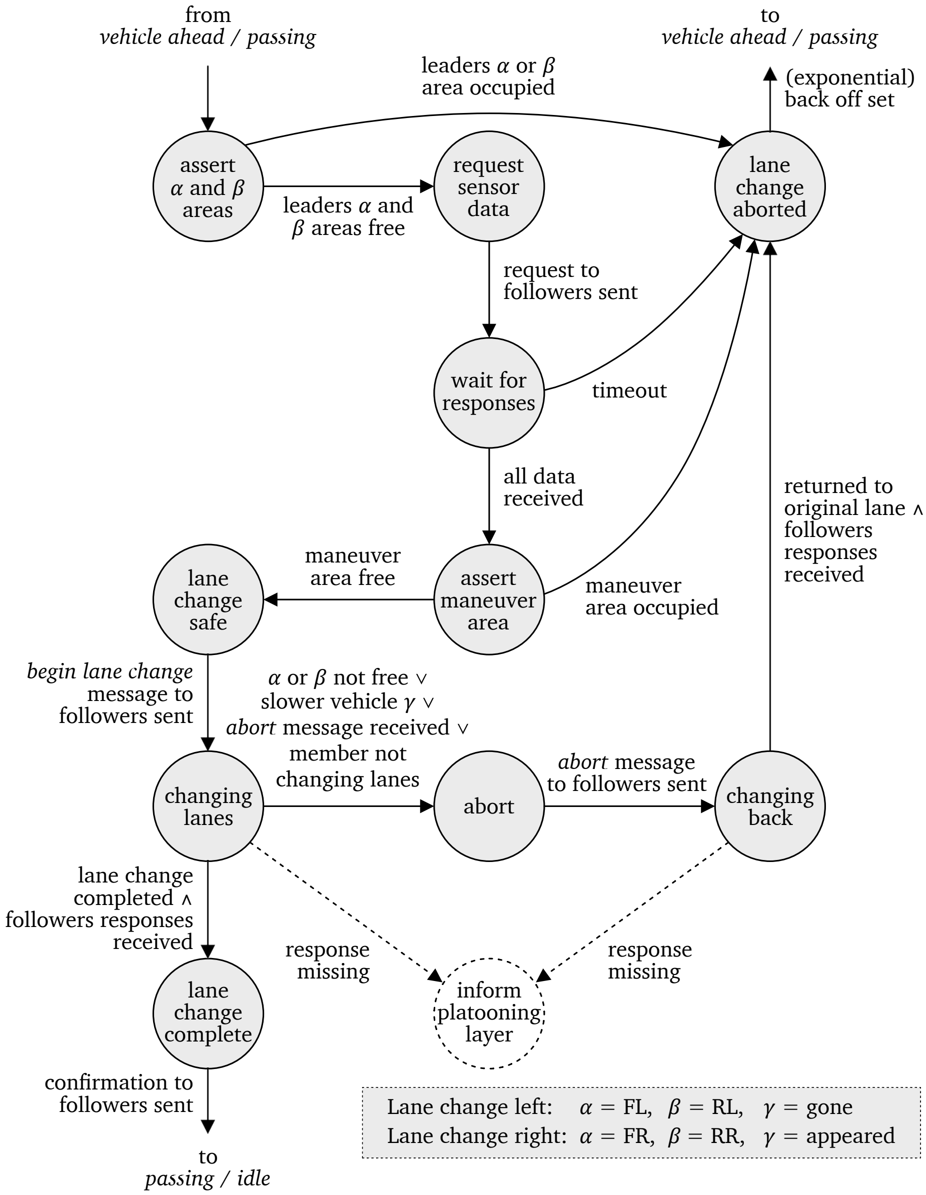


Figure 3.6 – FSM for the lane change maneuver from the leader's perspective.

lane have to be evaluated whether they are free: the *FL* and *RL* areas.[26] The closest vehicle in the *FL* area has to be at least the safety distance away for being evaluated as free. Therefore, the leader will measure this distance. To prevent oscillation, the safety distance is multiplied by a factor > 1 when the platoon is in a decision phase.

The more critical area than *FL* is *RL* [20, p. 2064]. An approaching vehicle should not be forced to brake harder.[27] Schubert, Schulze, and Wanielik [23, p. 612] name different possible threat measures: e.g., "distance, time to predicted collision, [...] deceleration to safety time" and propose a formula for the latter one. Ulbrich and Maurer [21, p. 977] propose a different formula to compute the necessary constant (negative) acceleration a of the closest vehicle in the *RL* area to avoid collisions. Although their formula includes reaction time and time gap of the *RL* vehicle (which sum up to the time headway), only the time gap is taken into account after the rear vehicle slowed down but not the reaction time. So this vehicle violates his time headway after braking. This is a valid goal when being in an emergency situation. But a standard lane change should not result in a situation where a vehicle in the *RL* area does not observe the safety distance (time headway) anymore. Instead, the safety distance should be maintained at all times. This is why we, based on the idea of Ulbrich and Maurer [21], propose a new formula to evaluate the *RL* area. Three cases need to be distinguished:

1. the *RL* vehicle is faster than the platoon ($v_{\mathrm{RL}} > v_{\mathrm{P}}$) and may be forced to brake ($a < 0$)[28] or

2. the *RL* vehicle is at most as fast as the platoon ($v_{\mathrm{RL}} \leq v_{\mathrm{P}}$) and may be forced to brake or keep his speed ($a \leq 0$) or

3. no *RL* vehicle is detected.

In all other cases, the *RL* area is considered occupied. We will begin with the first case.

A minimum distance $d_{\min}$ has to be computed which allows for the *RL* vehicle to decelerate from v_{RL} to v_{P}. After adjusting to the platoon's speed in the time frame t_{a}, the distance d_{RL} between the *RL* vehicle and the platoon has to be at least the safety gap d_{safety}. In addition, a reaction time T_{r} and a desired time gap T_{g} of the *RL* vehicle's driver is taken into account.

The distance $d_{\min}$ can now be defined as the distance d_{react} the *RL* vehicle drives during its driver's reaction time, added by the distance d_{a} needed for deceleration,

[26] For a *lane change right* maneuver *FR* replaces *FL* and *RR* replaces *RL*.

[27] One might argue that other vehicles should not be hindered at all. But Krauß [30, p. 82] states that this behavior would lead to a situation in which the right lane is congested and the left one allows for almost free flow. This is unrealistic behavior and therefore not considered further.

[28] If $a \geq 0$, the necessary distance would be infinite as the *RL* vehicle is faster than the platoon and assumed not to brake. The vehicle cannot adapt to the platoon's speed in finite time.

added by the safety gap d_{safety} after deceleration, subtracted by the distance d_{P} the platoon drives during deceleration and during the reaction time of the *RL* vehicle's driver (see Figure 3.7):

$$d_{\text{min}} = d_{\text{react}} + d_{\text{a}} + d_{\text{safety}} - d_{\text{P}}. \tag{3.40}$$

The distance d_{react} the *RL* vehicle is driving during its driver's reaction time T_{r} and the safety gap d_{safety} are defined by

$$d_{\text{react}} = v_{\text{RL}} T_{\text{r}} \tag{3.41}$$

$$d_{\text{safety}} = v_{\text{P}} \left(T_{\text{r}} + T_{g} \right). \tag{3.42}$$

In this case, v_{P} instead of v_{RL} is used to compute d_{safety} because the *RL* vehicle has to obey the safety gap when it has adapted the platoon's speed v_{P}.

The time frame t_{a} needed for deceleration can be computed as follows: as the initial speed of the *RL* vehicle is v_{RL} and its final speed is v_{P}, integrating the definition of the acceleration (see Equation (3.6)) leads to

$$\int_{v_{\text{RL}}}^{v_{\text{P}}} \mathrm{d}v = a \int_{0}^{t_{\text{a}}} \mathrm{d}t \tag{3.43}$$

$$\Rightarrow v_{\text{P}} - v_{\text{RL}} = a t_{\text{a}} \tag{3.44}$$

$$\Rightarrow t_{\text{a}} = \frac{v_{\text{P}} - v_{\text{RL}}}{a} \tag{3.45}$$

where a is the allowed (negative) acceleration of the *RL* vehicle.

Now the distance d_{a} the *RL* vehicle needs for decelerating can be computed as follows. Integrating the definition of speed (see Equation (3.10)) leads to

$$\int_{0}^{d_{\text{a}}} \mathrm{d}d = \int_{0}^{t_{\text{a}}} v \mathrm{d}t. \tag{3.46}$$

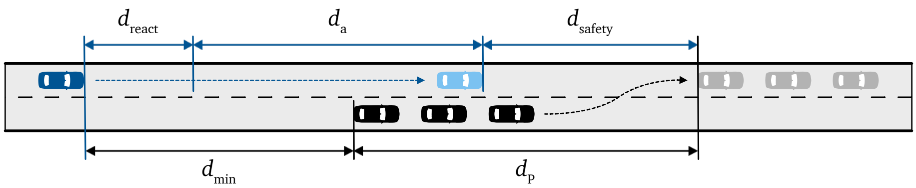


Figure 3.7 – Derivation of d_{min}: the approaching *RL* vehicle drives d_{react} during its driver's reaction time, drives d_{a} while decelerating to the speed of the platoon, and then still obeys its safety distance d_{safety}. In this time, the platoon drives d_{P}, which gives the *RL* vehicle more space for deceleration, and is therefore subtracted. The result is d_{min}.

With

$$v(t) = at + v_{\mathrm{RL}} \tag{3.47}$$

and Equation (3.45), Equation (3.46) leads to

$$\begin{aligned}
d_{\mathrm{a}} &= \int_0^{t_{\mathrm{a}}} at\,\mathrm{d}t + \int_0^{t_{\mathrm{a}}} v_{\mathrm{RL}}\,\mathrm{d}t && (3.48)\\
&= \left[\frac{a}{2}t^2\right]_0^{t_{\mathrm{a}}} + \left[v_{\mathrm{RL}}t\right]_0^{t_{\mathrm{a}}} && (3.49)\\
&= \frac{a}{2}t_{\mathrm{a}}^2 + v_{\mathrm{RL}}t_{\mathrm{a}} && (3.50)\\
&= \frac{a}{2}\frac{(v_{\mathrm{P}} - v_{\mathrm{RL}})^2}{a^2} + v_{\mathrm{RL}}\frac{(v_{\mathrm{P}} - v_{\mathrm{RL}})}{a} && (3.51)\\
&= \frac{1}{2a}(v_{\mathrm{P}} - v_{\mathrm{RL}})^2 + \frac{v_{\mathrm{RL}}}{a}(v_{\mathrm{P}} - v_{\mathrm{RL}}) && (3.52)\\
&= \frac{v_{\mathrm{P}}^2 - v_{\mathrm{RL}}^2}{2a}. && (3.53)
\end{aligned}$$

The distance d_{P} the platoon drives while the driver of the *RL* vehicle has to react and while the *RL* vehicle is braking[29] is defined with Equation (3.45) by

$$\begin{aligned}
d_{\mathrm{P}} &= v_{\mathrm{P}}(T_{\mathrm{r}} + t_{\mathrm{a}}) && (3.54)\\
&= v_{\mathrm{P}}T_{\mathrm{r}} + v_{\mathrm{P}}\frac{(v_{\mathrm{P}} - v_{\mathrm{RL}})}{a}. && (3.55)
\end{aligned}$$

Now the minimum distance d_{min} as defined in Equation (3.40) can be computed by

$$\begin{aligned}
d_{\mathrm{min}} &= d_{\mathrm{a}} - d_{\mathrm{P}} + v_{\mathrm{RL}}T_{\mathrm{r}} + v_{\mathrm{P}}\left(T_{\mathrm{r}} + T_{\mathrm{g}}\right) && (3.56)\\
&= \frac{v_{\mathrm{P}}^2 - v_{\mathrm{RL}}^2}{2a} - v_{\mathrm{P}}\frac{(v_{\mathrm{P}} - v_{\mathrm{RL}})}{a} + v_{\mathrm{RL}}T_{\mathrm{r}} + v_{\mathrm{P}}T_{\mathrm{g}} && (3.57)\\
&= \frac{1}{2a}\left(v_{\mathrm{P}}^2 - v_{\mathrm{RL}}^2 - 2v_{\mathrm{P}}^2 + 2v_{\mathrm{P}}v_{\mathrm{RL}}\right) + v_{\mathrm{RL}}T_{\mathrm{r}} + v_{\mathrm{P}}T_{\mathrm{g}} && (3.58)\\
&= -\frac{1}{2a}\left(v_{\mathrm{P}}^2 - 2v_{\mathrm{P}}v_{\mathrm{RL}} + v_{\mathrm{RL}}^2\right) + v_{\mathrm{RL}}T_{\mathrm{r}} + v_{\mathrm{P}}T_{\mathrm{g}} && (3.59)\\
&= -\frac{1}{2a}(v_{\mathrm{P}} - v_{\mathrm{RL}})^2 + v_{\mathrm{RL}}T_{\mathrm{r}} + v_{\mathrm{P}}T_{\mathrm{g}} && (3.60)
\end{aligned}$$

In the second case, the *RL* vehicle is at most as fast as the platoon ($v_{\mathrm{RL}} \leq v_{\mathrm{P}}$) and may be forced to brake or to keep his speed ($a \leq 0$). Due to the speed of the *RL* vehicle, the *RL* area is considered free if this vehicle keeps at least the safety distance.

[29] d_{min} could be decreased a little if it is taken into account that the platoon maybe accelerates when changing lanes. This case is not considered here, however, as additional case distinctions would have to be made: does the RL vehicle reach the platoon's speed in the acceleration phase or only when the platoon has already reached its maximum speed? Or: is the platoon constantly accelerating during the entire overtaking process? In addition, the benefit achieved is (in most cases) small.

Thus, d_{min} can be computed by

$$\begin{aligned} d_{\text{min}} &= d_{\text{safety}} && (3.61) \\ &= v_{\text{RL}} \left(T_{\text{r}} + T_{\text{g}} \right). && (3.62) \end{aligned}$$

In the third case, when no *RL* vehicle is detected, the *RL* area is always free. In all other cases the *RL* area is considered occupied. In conclusion, d_{min} can be computed by

$$d_{\text{min}} = \begin{cases} -\frac{1}{2a} \left(v_{\text{P}} - v_{\text{RL}} \right)^2 + v_{\text{RL}} T_{\text{r}} + v_{\text{P}} T_{\text{g}} & \text{, if } v_{\text{RL}} > v_{\text{P}} \wedge a < 0 \\ v_{\text{RL}} \left(T_{\text{r}} + T_{\text{g}} \right) & \text{, if } v_{\text{RL}} \leq v_{\text{P}} \wedge a \leq 0 \\ \infty & \text{other.} \end{cases} \tag{3.63}$$

To avoid oscillation, d_{min} is multiplied by a factor > 1. This ensures that the platoon will not abort the maneuver due to a vehicle that is exactly d_{min} away in the decision phase and a little closer than that during the lane change maneuver.

To compute d_{min}, the allowed negative acceleration a needs to be defined. Ulbrich and Maurer [21, p. 977] state that an "averagely altruistic driver" accepts to brake with $a = -1\,\text{m/s}^2$ when another vehicle moves in front of him. So this value is chosen for a *lane change left* maneuver. As a lane change to the right should not hinder a vehicle in the *RR* area at all, $a = 0\,\text{m/s}^2$ is selected for the *lane change right* maneuver. The reaction time T_{r} of the *RL* vehicle's driver is set to 1.0 s and the desired time gap T_{g} is set to 0.8 s [21, p. 977] for a total time headway of 1.8 s.

If the *FL* and *RL* areas are considered free, i.e., $d_{\text{FL}} \geq d_{\text{P\_safety}}$ and $d_{\text{RL}} \geq d_{\text{min}}$, the leader evaluates his own lane change maneuver as safe. Therefore, he changes to the *request sensor data* state. If at least one of the areas is occupied by another vehicle, a lane change is not safe at the present time. Thus, the leader will change to the *lane change aborted* state.

When being in the overtaking lane, the computation of d_{min} remains the same but the truck minimum gap of $d_{\text{truck}} = 50\,\text{m}$ is also taken into account (see Section 3.1.1.2). This means, the actual minimum gap for a *lane change right* maneuver is computed by

$$\max\left(d_{\text{min}}, d_{\text{truck}}\right). \tag{3.64}$$

3.1.2.2 Request Sensor Data

In this state the leader's own maneuver area is free, but this does not necessarily has to apply to his followers. As the leader needs this information, he sends a *request sensor data* message to all of his followers. This is the first time the followers get involved in the lane change maneuver. The leader also sets a timer in such a manner that long response times or message delivery failure can be noticed. Then he changes to the *wait for responses* state.

3.1.2.3 Wait for Responses

While waiting for the follower's *response sensor data* messages, two things can happen. On the one hand, the timer that was set in the *request sensor data* state can expire. In this case, the leader did not get a response from all of his followers due to an unreliable V2V communication system. He can not assess the safety of the lane change[30] and aborts the maneuver by changing to the *lane change aborted* state. And on the other hand, the leader received a valid *response sensor data* message from all of his followers. In this case, he got the required data to be able to assess the safety of the platoon's maneuver area and therefore changes to the *assert maneuver area* state.

3.1.2.4 Assert Maneuver Area

Each received *response sensor data* message contains a *true* if the follower's maneuver area is free or a *false* if that is not the case. The leader connects all answers with a logical AND. If the result is *true*, the whole platoon's maneuver area is considered free and a lane change is safe. The leader therefore changes to the *lane change safe* state. If the maneuver area of at least one follower is occupied, the platoon's maneuver area is occupied and a lane change is unsafe. In this case, the leader changes to the *lane change aborted* state.

3.1.2.5 Lane Change Safe

After the leader has evaluated a lane change as safe[31], the actual lane change of the platoon can start. The leader sends a *begin lane change* message to his followers in order to synchronize the maneuver and changes to the *changing lane* state.

3.1.2.6 Changing Lanes

This is the phase where the operational level of the car turns the steering wheel in the appropriate direction.[32] The leader moves, guided by his lane change assistant, to

[30] It could still be possible to evaluate the safety even when some followers have not returned their sensor data. If, e.g., follower number three did not respond, the data of followers two and four could be sufficient when the maneuver area is completely covered with their data. This depends, for example, on the traffic situation and the sensor capabilities. This is not taken into account any further.

[31] In the time between the leader's assessment of safety and that of the followers, there is a possibility that the leader's safety is no longer given. This is, however, not a problem, as the safety of the lane change is continuously assessed during its execution (see Section 3.1.2.6). If a vehicle were to interfere with the leader in the meantime, the maneuver would be stopped immediately.

[32] This is just a metaphor for explanation. In a real autonomous vehicle the steering wheel might not actually turn when the vehicle is changing lanes.

the adjacent lane. During the lane change, he monitors the maneuver area for former undetected vehicles and the *FR* area for slower vehicles. If there is no interference, he reaches the center of the target lane which concludes the actual lane change. If that is the case and if he received a *lane change complete* message from all of his followers (see Section 3.1.3.5), he changes to the *lane change complete* state.

But as the algorithm intervenes with the traffic situation, unforeseen disruptions can occur and make hazard management necessary. A former undetected vehicle could, e.g., due to a sensor malfunctioning or because this vehicle is changing from another lane to the platoon's target lane, appear in the maneuver area. If the platoon would continue to change lanes, it could crash into this vehicle. The necessity to abort the maneuver is evaluated with Equation (3.63). But before the actual lane change started, a was set to $-1\,\mathrm{m/s^2}$. Ulbrich and Maurer [21, p. 977] propose a higher deceleration when the lane change is already in progress: a is now set to $-3.5\,\mathrm{m/s^2}$.

In addition, the leader could receive an *abort* message from one of his followers (see Section 3.1.3.6). Moreover, the reason for changing lanes could disappear, meaning the slower vehicle accelerates or gets out of range (e.g., by exiting the freeway). In this case, overtaking is not necessary (or even possible) anymore. If any of these cases occur, changing lanes is either not safe or not necessary anymore, and the leader aborts the maneuver by returning to the original lane. Therefore, he changes to the *abort* state.

The reason for a *lane change right* maneuver differs from the aforementioned *lane change left* maneuver (see Section 3.1.1.3). In this case, the leader does not evaluate whether the slower vehicle accelerates or gets out of range, but monitors the *FR* area for appearing slower vehicles. If a former undetected vehicle is present in this area and, in addition, it is useful and possible to overtake (according to the same set of rules defined in Section 3.1.1.2), he will abort the lane change and return to the overtaking lane by changing to the *abort* state.

If the V2V communication system is unreliable, messages can be received with a substantial delay. This could lead to a situation where the leader[33] started a lane change maneuver, but at least one follower did not receive the *begin lane change* message and does not change lanes. Therefore, the leader monitors his direct follower whether he has approximately the same lateral position as himself. If this is not the case, the leader assumes a problem, aborts the maneuver, and changes to the *abort* state.

If the leader does not receive a *lane change complete* message at all from at least one of his followers, a general communication problem has occurred. As platooning itself is endangered, the leader changes to the *inform platooning layer* state.

[33] And maybe some of his followers.

3.1.2.7 Lane Change Complete

In this state, the platoon's lane change is completed: the platoon is driving in the center of the target lane. The leader informs his followers about this by sending them a *lane change complete* message.

If a *lane change left* maneuver was executed, the platoon is now driving in the overtaking lane. This means that the leader will change to the *passing* state of the overtaking FSM. In the other case, the platoon returned to the original lane and the overtaking maneuver is completed: the proposed algorithm was fully executed. This could be a final state of the FSM, but the platoon is still driving on the freeway and overtaking another vehicle could become useful. Thus, the leader changes back to the *idle* state of the overtaking FSM and the proposed algorithm starts over.

3.1.2.8 Abort

When the leader enters this state, something unexpected has happened during the actual lane change and the platoon has to abort the maneuver. The leader needs to inform his followers about this for which reason he sends an *abort* message to them and changes to the *changing back* state.

3.1.2.9 Changing Back

In this phase, the lane change that has already started needs to be reversed due to an unsafe traffic situation or an obsolete lane change decision. The leader's steering wheel is turned in the opposite direction and he returns to the original lane (during a *lane change left* maneuver) or the overtaking lane (during a *lane change right* maneuver). Returning to the old lane is considered safe because the leader has not yet completed the originally planned lane change and it is assumed that no other vehicle is already blocking his lane.

The platoon successfully aborted the lane change maneuver when the leader returned to the old lane's center and received an *abort complete* message from all of his followers (see Section 3.1.3.8). Then the leader changes to the *lane change aborted* state.

3.1.2.10 Lane Change Aborted

This state is always reached when a criterion for canceling the lane change is met and the potentially dangerous traffic situation has been transferred back to a safe state. Now the leader could immediately attempt another lane change, but as traffic situations normally do not change on a per-millisecond basis, a timer is set until a new attempt is started. If the leader is in a *lane change left* maneuver, an exponential timer (with an upper limit) is set because the traffic situation might not change soon (e.g., due to a high traffic density) and the platoon should not block the communication channel unnecessarily often. If he is in a *lane change right* maneuver, a fixed timer is set because the platoon should leave the overtaking lane as soon as possible due to the obligation to drive in the right lane. After the timer expired, the leader changes back to the *vehicle ahead* overtaking state (*lane change left* maneuver) or the *passing* overtaking state (*lane change right* maneuver).

3.1.2.11 Inform Platooning Layer

This state will be reached when the V2V communication system could not deliver a *lane change complete* message at all. While delayed message delivery is handled by the proposed algorithm, the loss of messages endangers the platooning functionality itself because it relies on communication. Therefore, the leader informs the platooning functionality layer to handle the problem in a more general way.[34] But this is not part of the proposed algorithm and not further discussed here.

3.1.3 Lane Change Maneuver: Follower's Perspective

As mentioned before, followers do not know about the overtaking maneuver taking place. They are either in the *idle* state when driving in a lane behind the leader or perform a lane change maneuver.

During a lane change, each follower has to evaluate whether his own maneuver area is free, send this information to the leader, and wait for his decision. If the followers do not receive a decision, the lane change is aborted. When they receive a positive decision, each follower changes lanes in the given direction and goes back to the *idle* state to be ready for another lane change. If something unexpected happens during the lane change, the follower will abort it and return to the old lane.

In contrast to the leader's lane change FSM, the follower does not decide about lane changing and an abort of the maneuver is always communicated to all members via the leader. If an abort necessity is detected by a follower, he will send this information to the leader and not to every follower. The leader then informs all members about the abort.[35]

The lane change FSM is shown in Figure 3.8. All states are described below and are based on a *lane change left* maneuver. Differences to the *lane change right* maneuver are mentioned accordingly.

[34] For example by splitting up the platoon and establishing larger safety margins.

[35] This procedure, which leaves the leader in charge, is not a threat to the other followers. They receive the *abort* message from the leader in a later time step than they would if the message would have been directly sent to them. But if a dangerous situation arises for one of these followers, he would also directly abort the maneuver. A vehicle in danger never waits for the leader's decision.

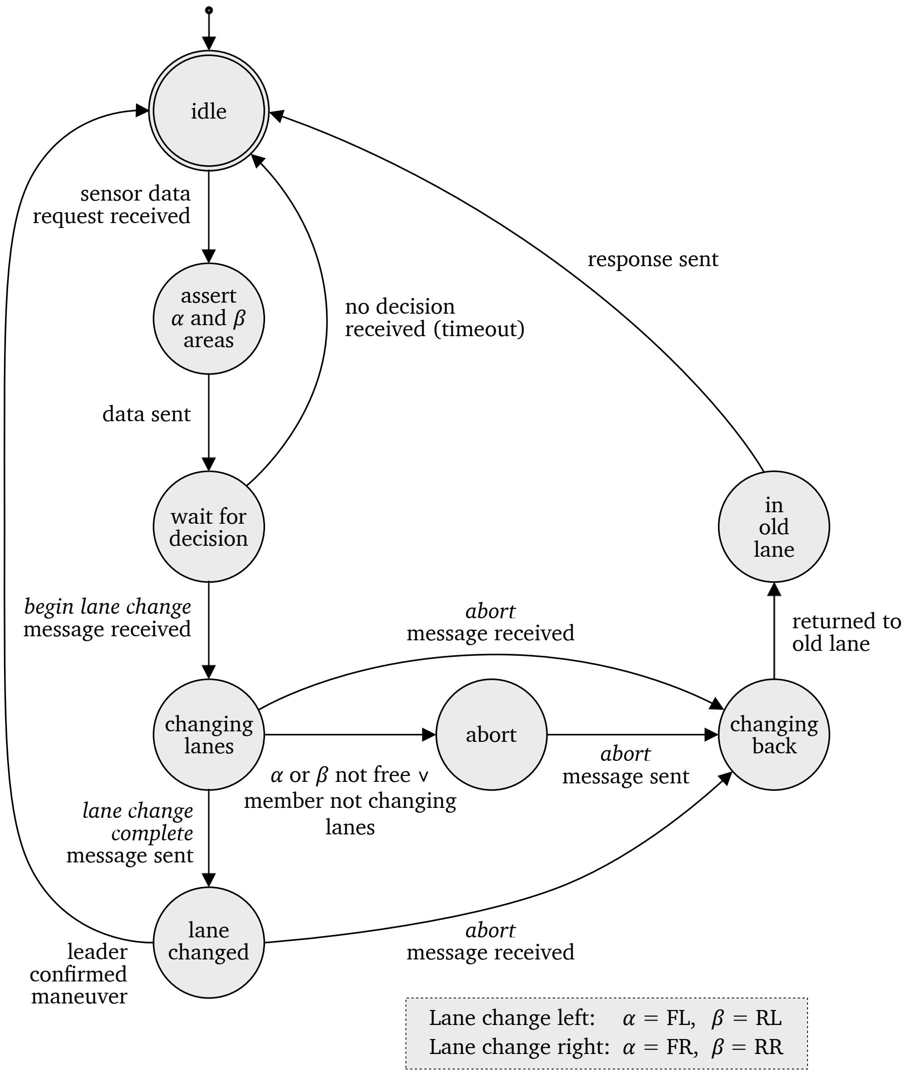


Figure 3.8 – FSM for the lane change maneuver from the follower's perspective.

3.1.3.1 Idle

While in the *idle* state, a follower is driving in a lane behind the platoon leader. When he receives a *request sensor data* message from the leader (see Section 3.1.2.2), he changes to the *assert α and β areas* state.

3.1.3.2 Assert α and β Areas

The follower was asked to assert his own maneuver area. He will do that in this state in accordance to the same set of rules as the leader did (see Section 3.1.2.1). Then he sends a *response sensor data* message back to the leader, containing *true* if his own maneuver area is free and *false* otherwise. Then he sets a timer and changes to the *wait for decision* state.

3.1.3.3 Wait for Decision

In this state, the follower is waiting for the leader's decision whether a lane change is safe or not. In the first case, the follower receives a *begin lane change* message from the leader (see Section 3.1.2.5), including in which direction to change, and changes to the *changing lane* state. In the second case, no message will be sent from the leader and the timer set in the *assert α and β areas* state expires. Therefore, the follower changes back to the *idle* state.

3.1.3.4 Changing Lanes

In this state, the operational level of the vehicle is turning the steering wheel in the direction given by the *begin lane change* message to execute the actual lane change. The state is the same as the leader's *changing lane* state (see Section 3.1.2.6), just the handling of an *abort* message received is different and the monitoring of a slower vehicle is not applicable. If the follower receives an *abort* message from the leader, the lane change became unsafe for another platoon member. The follower will directly change to the *changing back* state. If the α or β area is not free anymore, which is evaluated in the same manner as in the corresponding leader state, the lane change became unsafe for the follower. As a result, he changes to the *abort* state like the leader would do as well.

When he successfully changed lanes, meaning that he is driving in the center of the target lane, he will send the leader a *lane change complete* message and change to the *lane changed* state.

If the V2V communication system is unreliable, a follower monitors the lateral position of his direct front and rear[36] members. If at least one of these positions

[36] If applicable.

differ more than threshold d_{lat}, he will assume a problem, abort the maneuver, and change to the *abort* state.

3.1.3.5 Lane Changed

In this state, the follower is driving in the target lane. The maneuver from the platoon's point of view is not completed yet, because another member could still be in the *changing lanes* state and send an *abort* message. Since the leader decides when the follower can go back to the *idle* state, the follower now waits for either an *abort* or a *lane change complete* message from the leader. If he receives an *abort* message, he will change to the *abort* state and if he receives a *lane change complete* message, he will change to the *idle* state to be ready for another lane change. This concludes the follower's lane change algorithm.

3.1.3.6 Abort

The follower detected an unsafe traffic situation and will now send an *abort* message to the leader to inform him about it. Then he changes to the *changing back* state.

3.1.3.7 Changing Back

During this state, the follower returns to the original lane, which is handled exactly as in the leader's *changing back* state (see Section 3.1.2.9). When the follower has returned to the center of the old lane, he changes to the *in old lane* state.

3.1.3.8 In Old Lane

The follower successfully returned to the old lane and informs the leader about this by sending an *abort complete* message to him. Then he returns to the *idle* state.

This concludes the description of the FSMs states. The overtaking algorithm is fully defined theoretically which leads to the next step: the implementation.

3.2 Implementation

Before the proposed algorithm can be tested and benchmarked, it has to be implemented in order to be used within the simulation environment. Firstly, the main concepts of the implementation are described below. Secondly, the elementary communication system is introduced.

The implementation of the proposed algorithm follows an object-oriented approach. The class `PlatoonVehicle` provides basis functionality which both the leader and his followers will need to execute the proposed algorithm (e.g., a method for asserting the α area). The leader is an instance of the class `PlatoonLeader`, which is a subclass of `PlatoonVehicle`. It provides additional functionality that is necessary for the leader but not the followers (e.g., compute $t_{\text{overtaking}}$), and it also handles the leader's overtaking and lane change FSMs. A platoon follower, on the other hand, is an instance of the class `PlatoonFollower`, which is also a subclass of `PlatoonVehicle`. It handles the follower's lane change FSM. The whole platoon is an instance of the class `Platoon`, which holds one instance of `PlatoonLeader` and at least one instance of `PlatoonFollower`. This class also ensures that every platoon member carries out one step of the overtaking algorithm in every simulation step.

For easy configuration of the algorithm, all relevant parameters (e.g., the overtaking thresholds) are defined in a separate file. To have a closer look on what is happening during the simulation, we built *SimuWatch*, an additional pop-up window that provides more information about the platoon members (see Figure 3.9). In addition, extensive (debugging) information can be printed. This allows for an easy understanding in which state of the FSMs each member currently is and what happens when being in this state.

The elementary communication system we developed simulates simple V2V communication. As already stated in Section 2.3, we did not intend to model a detailed communication model with realistic physical behavior, but a rather simple "send and receive" one. Therefore, only `send()` and `receive()` functions are required.

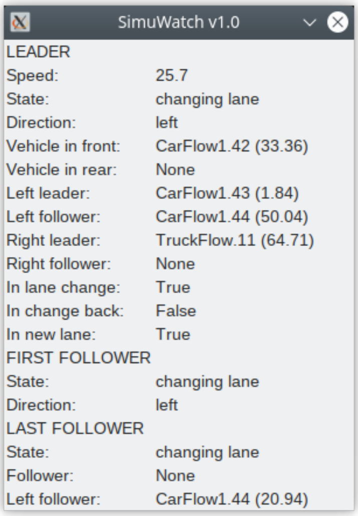


Figure 3.9 – Screenshot of our simulation information window SimuWatch for debugging purposes. It displays information about the leader and his first and last follower, e.g., the state they are currently in, which vehicles are in proximity to them, and in what direction the next lane change will be executed.

`send()` accepts a message of a certain type (e.g., *response sensor data*) and optional data (e.g., the result of an α and β area assertion by a follower). Each sent message is added to a queue and can be received by `receive()` in either the next step of the simulation or after an exponentially distributed delay.

With the proposed algorithm being implemented, we can now evaluate its safety by thorough testing.

3.3 Testing

It has to be shown that the algorithm's execution results in a successful overtaking maneuver when overtaking is possible and that it does not attempt to overtake when overtaking is unsafe. Two different kinds of tests shall ensure this: verification and validation tests. While verification tests are supposed to show that the FSMs are implemented correctly, validation tests are conducted to show that the algorithm meets the intended application: safe overtaking of one or more slower vehicles when possible.

For the verification tests every single transition of the FSMs is tested for correct behavior with a valid and an invalid input. All test cases are based on pytest[37] and executed without the simulation environment. For example, the leader's overtaking maneuver (see Figure 3.3) starts in the *idle* state. There are two options: the leader either detects a vehicle in front or none. In the first case, the leader has to change to the *vehicle ahead* state (because of the *vehicle ahead detected* transition), in the second case, he has to stay in the *idle* state. This is tested with two inputs: a detected vehicle or no detected vehicle and the corresponding expected states *vehicle ahead* or *idle*. All transitions of the FSMs are tested this way. Since the verification tests are not very complex but numerous, a more detailed description of every test is not given here. Since all tests are passed, we assume that the implementation works as expected.

The validation tests on the other hand simulate traffic scenarios which could lead to dangerous situations. We selected a variety of test cases in such a manner that a larger number of typical traffic scenarios are covered. While a verification test only tests a single transition in the FSM, a validation test will test a longer path within the FSM. For a test case considered being successful, all of the following conditions have to be met:

- no collisions of vehicles occurred within the simulation,
- the order of the platoon members is still the same after the test,
- all platoon members drove the same distance,[38]

[37] https://pytest.org (version used: 6.0.2)
[38] We use pytest.approx() with a relative tolerance of 1e-6.

- actual platoon minimum inter-vehicle distance ≥ CACC constant gap,
- minimum speed of the platoon ≥ 0.95 × speed of the slowest vehicle,[39]
- maximum speed of the platoon ≤ 1.05 × desired speed,
- all test case-specific conditions that are described below for the individual test cases.

All test cases described below meet these requirements. Each test case is based on pytest, but is run, in contrast to the verification tests, within the simulation environment.

3.3.1 Simulation Set-up

As described in Section 2.3, SUMO[40] is used to simulate the freeway environment. This freeway consists of three lanes in one direction and has a total length of 50 km, enabling the platoon to drive a sufficiently long distance. The lanes are numbered, starting with lane one for the rightmost one. The speed limit of the lanes is set to 37.3 m/s. Each step and action step of the simulation is 0.01 s long to allow for the frequent communication required by platooning. A list of the main simulation parameters can be found in Table 3.1. The parameters of the proposed algorithm and the selected values are listed in Table 3.2.[41]

The simulated vehicles are based on different vehicle classes that come with SUMO.[42] For our traffic simulations, three vehicle types are derived from these classes:

[39] The factor is a buffer, as the platoon can slightly exceed or underrun the desired speed when accelerating and braking are handled by the ACC.

[40] At least version 1.7.0 is required for the proposed algorithm to work, because the necessary function `getFollower()` was introduced with this version.

[41] All parameters are varied in the parameter study in Section 4.3.

[42] https://sumo.dlr.de/docs/Vehicle_Type_Parameter_Defaults.html (access date: 06/11/20)

Parameter	Value
Step length	0.01 s
Action step length	0.01 s
Lateral resolution	0.8 m
Freeway length	50 km
Lane speed limit	37.3 m/s
Lane width	3.2 m
Number of lanes (one direction)	3
Number of sublanes	4

Table 3.1 – The most important parameters of SUMO and the simulated freeway that are set for our simulations.

Parameter	Value
Maximum overtaking time t_{max}	<45.0 s
Minimum overtaking speed delta v_{Δ}	2.7 m/s
Allowed deceleration for *RL* vehicle (before overtaking) a	$-1.0\,\text{m/s}^2$
Allowed deceleration for *RL* vehicle (during overtaking) a	$-3.5\,\text{m/s}^2$
Allowed deceleration for *RR* vehicle a	$0.0\,\text{m/s}^2$
Maximum distance to *F* vehicle when changing to the left	160 m
Time to stay in original lane after changing back t_{stay}	10.0 s
Maximum allowed lateral offset between members d_{lat}	0.4 m
Minimum exponential backoff	0.32 s
Maximum exponential backoff	2.56 s
Timer for follower and returning to original lane	0.20 s

Table 3.2 – The parameters of the proposed overtaking algorithm that are set for our simulations.

passenger cars, trucks, and platooning cars. The latter are basically passenger cars but require additional platooning-specific parameters. Most of the parameters listed in Table 3.3 are self-explanatory. Because it is necessary to simulate correct lane change behavior, the SL2015 lane change model [26] was chosen. The driver's desired minimum time headway consists of two parts: his reaction time (1.0 s) and his desired time gap (0.8 s) [21, p. 977].

Parameter	Value
SUMO vehicle class	passenger
Acceleration ability	$2.9\,\text{m/s}^2$
Deceleration ability	$7.5\,\text{m/s}^2$
Desired velocity	44.4 m/s
Multiplier for lane speed limits (speed factor)	1.2
Deviation of the speed factor	0.1
Vehicle length	4.7 m
Lane change model	SL2015 [26]
Car follow model	Krauss
Driver imperfection σ	0.5
Driver's desired minimum time headway τ	1.8 s
Minimum gap	2.5 m

Table 3.3 – The main parameters of the passenger car type that is used in the traffic scenarios.

The truck type shares most of the parameters of the passenger car type except for its physical abilities like maximum acceleration. The main parameters for the truck class that differ from the passenger cars are listed in Table 3.4.

A platoon is built of the passenger car type and consists of four cars. The leader is equipped with ACC and the followers with CACC. Platooning vehicles share the same parameters as the passenger car type except for those listed in Table 3.5.

3.3.2 Basic Test Cases

The first type of test cases consists of elementary tests designed to show the basic functionality of the algorithm. Only the platoon and at most one slower vehicle are present during the simulations. In addition to the tests already mentioned in Section 3.3.1, each test case is checked for the criteria listed in Table 3.6. We developed the following basic test cases:

- **Best case (Basic-1):** The platoon is not delayed at all and can drive at its desired speed. See Figure 3.10a.

Parameter	Value
SUMO vehicle class	trailer
Acceleration ability	1.1 m/s^2
Deceleration ability	4.0 m/s^2
Desired velocity	22.2 m/s
Vehicle length	16.5 m

Table 3.4 – The main parameters of the truck type that differ from the passenger car type (see Table 3.3).

Parameter	Value
SUMO car follow model	CC
Plexe car follow model (leader / follower)	ACC / CACC [31]
Desired velocity	27.8 m/s
ACC headway time	1.0 s
Minimum gap after leader	0.0 m
CACC constant gap d_d	5.0 m
Damping ratio ξ	1.0
Bandwidth of the controller ω_n	0.2 Hz
Leader/front vehicle weighting factor C_1	0.5

Table 3.5 – The additional parameters for vehicles that are platoon members.

Test case	Visited lanes	Vehicles overtaken	Visited states
Basic-1, Basic-2	0	-	*idle*
Basic-3	0, 1, 0	slower vehicle	see description

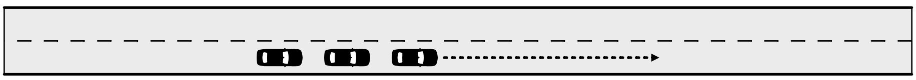
Table 3.6 – Individual checks for the basic test cases.

- **Worst case (Basic-2):** The platoon approaches a slower vehicle, adapts its speed, and does not overtake it. This means the proposed algorithm is not executed. See Figure 3.10b.
- **Plain overtaking (Basic-3):** The platoon approaches a slower vehicle and overtakes it according to the proposed algorithm. The overtaking maneuver is not disturbed by another vehicle. Every state on the shortest path possible through the FSM has to be visited at least once by each platoon member. See Figure 3.10c

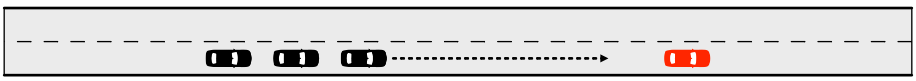
3.3.3 Decision Phase Test Cases

In the decision phase, the platoon decides whether to execute a lane change maneuver. A wrong decision can lead to unsafe situations including collisions with other vehicles, which is why the decision phases of the proposed algorithm are tested by the following test cases. All test cases have in common that the platoon is approaching a slower vehicle (which is not mentioned again in the descriptions of the test cases). Each test case is also checked for the criteria listed in Table 3.7.

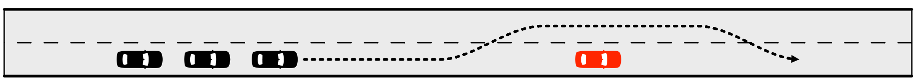
- **Approaching car (A-1):** A car approaches on the left neighboring lane from behind. The leader's maneuver area is free, but it is not free for all followers as



(a) Best case (Basic-1)



(b) Worst case (Basic-2)



(c) Plain overtaking (Basic-3)

Figure 3.10 – Basic test cases of the proposed overtaking algorithm: (a) the platoon is not delayed at all, (b) the platoon is delayed by a slower vehicle and does not overtake, (c) the platoon overtakes a slower vehicle and is not disturbed by another vehicle.

Test case	Visited lanes	Vehicles overtaken	Important visited states
A-1, A-2	0, 1, 0	slower vehicle	*lane change aborted*
A-3, A-4	0	-	see description
A-5	0, 1, 0	slower vehicle	see description
A-6	0, 1, 0	merging vehicle	-
A-7, A-8	0, 1, 0	slower vehicle, merging vehicle	-
A-9, A-10	0, 1, 0	slower vehicle	-

Table 3.7 – Individual checks for the decision phase test cases.

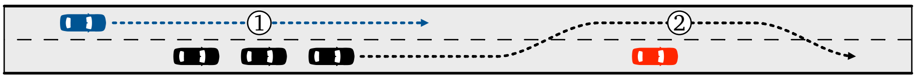
the approaching car is blocking the *RL* area of some followers. The overtaking maneuver is postponed until the approaching car has passed the platoon. Then the platoon overtakes the slower vehicle. See Figure 3.11a.

- **Neighboring car (A-2):** The situation is the same as in A-1, but the approaching car is already left of the platoon when the slower vehicle is detected. In this case, the leader can detect the approaching car himself and does not have to ask his followers for their sensor data while this car is detectable. This means that during the *lane change left* maneuver the maneuver area is always free for every follower when asked by the leader (because he will ask them only when the approaching car is out of range). See Figure 3.11b.

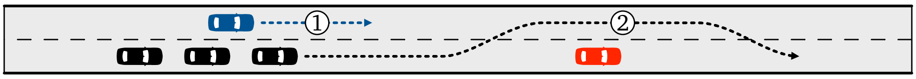
- **Low speed delta (A-3):** The speed delta of the overtaking lane and the slower vehicle is less than the threshold for overtaking. So overtaking is not useful (and not possible). Therefore, the leader should only visit the *idle* and *vehicle ahead* states. The followers should always be in the *idle* state. See Figure 3.11c.

- **Long overtaking time (A-4):** The situation is the same as in A-3 but the speed delta between left and right lane is slightly bigger than the threshold for overtaking. Although overtaking is useful, it is still impossible due to a long overtaking time. See Figure 3.11c.

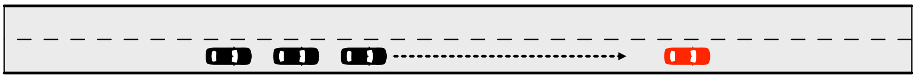
- **Unreliable V2V communication system (A-5):** When the leader asks his followers for their sensor data during the *lane change left* and *lane change right* maneuvers, message delivery can be delayed with a probability > 0 due to an unreliable communication channel. The same applies to the responses of the followers that contain the sensor data. When message delivery is delayed for a longer time, the leader aborts the lane change and goes back to the *vehicle ahead* or the *passing* state, respectively. After some retransmissions, all messages can be delivered and the overtaking maneuver will be completed. See Figure 3.11d.



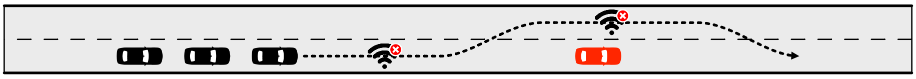
(a) Approaching car (A-1)



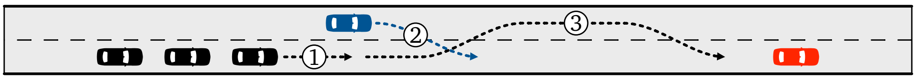
(b) Neighboring car (A-2)



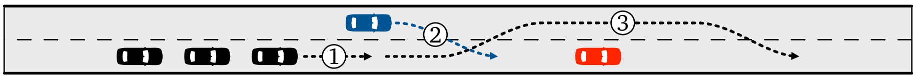
(c) Low speed delta (A-3); long overtaking time (A-4)



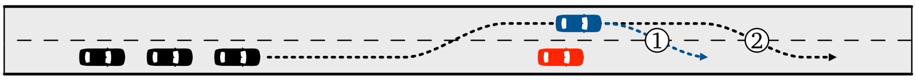
(d) Unreliable communication system (A-5)



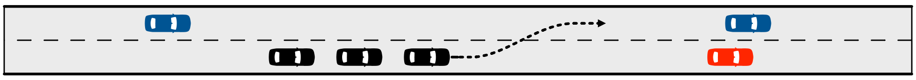
(e) New slow vehicle; overtaking one vehicle (A-6)



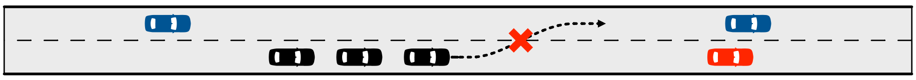
(f) New slow vehicle; overtaking both vehicles (A-7)



(g) Overtaken, then overtaking (A-8)



(h) Gap is just big enough (A-9)



(i) Gap is almost big enough (A-10)

Figure 3.11 – Decision phase test cases of the proposed overtaking algorithm (numbers in the illustrations describe the sequence of events): (a) a faster car approaches from behind, then it is detected by at least one of the followers (but not by the leader), and postpones the overtaking maneuver, (b) the approaching car is already detectable by the leader when he detects the slower vehicle, (c) the speed delta of both lanes is too small or overtaking would take too long to be allowed, (d) message delivery delays postpone the overtaking maneuver, (e) a merging vehicle becomes the new slower vehicle and is overtaken, (f) a merging vehicle and the slower vehicle are overtaken, (g) a vehicle overtakes the slower vehicle, merges into the right lane, and becomes almost as slow as the slower vehicle; the platoon overtakes both vehicles, (h) the platoon fits in between the two neighboring cars, (i) the platoon does not fit in between the two neighboring cars.

From the leader's perspective, this means that the *timeout* transition between the *wait for decision* state and the *lane change aborted* state was triggered during the *lane change left* and the *lane change right* maneuver. And the followers change at least one time from the *wait for decision* state to the *idle* state during both lane change maneuvers.

- **New slow vehicle; overtaking one vehicle (A-6):** The vehicle in front is only slightly slower than the platoon, hence it is not overtaken by the platoon. Another vehicle moves between the platoon and the slower vehicle and gets so slow that overtaking is possible. The platoon overtakes the merged vehicle and changes back to the original lane behind the slightly slower vehicle. See Figure 3.11e.

- **New slow vehicle; overtaking both vehicles (A-7):** The situation is the same as in A-6, but the platoon also overtakes the slightly slower vehicle as it is too close to change back to the original lane behind it. See Figure 3.11f.

- **Overtaken, then overtaking (A-8):** The platoon is overtaken by a faster vehicle. This vehicle overtakes also the slower vehicle in front of the platoon and merges into the right lane. After that, the merged vehicle becomes almost as slow as the slower vehicle in front of the platoon. The platoon therefore overtakes both vehicles. See Figure 3.11g.

- **Gap is just big enough (A-9):** Two vehicles on the overtaking lane are approaching the platoon. While the first neighboring vehicle is so close to the platoon that it prevents immediate overtaking, the second one is so far away from the first one that the platoon will fit in between them (according to the formulas used by the proposed algorithm) and the lane change is carried out. See Figure 3.11h.

- **Gap is almost big enough (A-10):** The situation is the same as in A-9, but the second neighbor is too close to the first one for the platoon to fit in. Therefore, the platoon remains in the original lane until both vehicles have passed. See Figure 3.11i.

3.3.4 Lane Change Phase Test Cases

In this phase, the algorithm actively controls the behavior of the platoon and could therefore produce a dangerous traffic situation. The test cases in this section shall show that the proposed algorithm can either avoid or handle such potentially dangerous situations. We defined two different groups of test cases. The first group is related to the already detected slower vehicle or a newly detected one during a lane change of the platoon. Such situations are not unsafe per se, but handling these in an appropriate way can help to optimize the traffic flow. The second group of test cases refers to a disrupting vehicle that appears during a lane change (e.g., by merging from one lane to the lane of interest).[43] This vehicle would have prevented overtaking, if it were already detected in the prior decision phase. This is an unsafe situation because the safety gap is violated if the platoon continues the overtaking maneuver. The proposed algorithm must bring such situations to a safe final state. In addition to these two groups, we finally tested the algorithm when the V2V communication system is unreliable and messages are delivered with a delay during lane changing.

Each traffic scenario consists of at least one slower vehicle that is driving in front of the platoon. When a potentially unsafe situation arises, the simulation is run until the proposed algorithm has brought the situation to a safe state. The simulation is terminated after that. This means that, e.g., when a former undetected vehicle leads to the abortion of a *lane change left* maneuver, the situation is regarded as safe when the platoon has returned to its original lane. After that, the platoon could still overtake the slower vehicle in front, but this is not simulated as this would be a standard overtaking situation which is already tested in test case Basic-1.

We defined the following test cases which are in connection with the already or newly detected slower vehicle. Each test case is also checked for the criteria listed in Table 3.8.

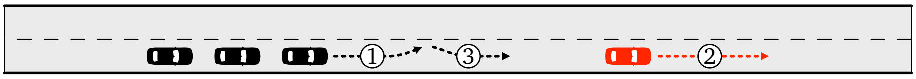
- **Slower vehicle gone soon (B-1):** The slower vehicle in front of the platoon is out of range during the *lane change left* maneuver (e.g., by exiting the freeway) and the platoon is still mainly on the original lane. The platoon aborts the maneuver and returns to the center of the lane. See Figure 3.12a.

- **Slower vehicle gone late (B-2):** The situation is the same as in B-1, but the slower vehicle gets out of range when the platoon is already mainly on

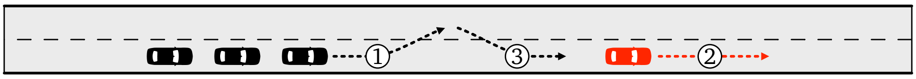
[43]Another reason for a suddenly appearing vehicle could be a temporary malfunction of a platoon member's sensors that accidentally did not detect the vehicle.

Test case	Visited lanes	Vehicles overtaken	Important visited states
B-1, B-3	0	-	*changing back*
B-2, B-4	0, 1, 0	-	*changing back*
B-5	0, 1, 0	slower vehicle new slower vehicle	*changing back*
B-6	0, 1, 0, 1, 0	slower vehicle new slower vehicle	*changing back*

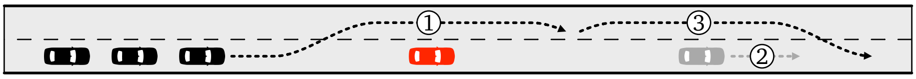
Table 3.8 – Individual checks for the lane change phase test cases which are in connection with the (new) slower vehicle.



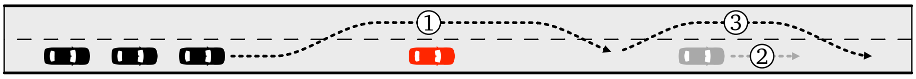
(a) Slower vehicle out of range soon (B-1); slower vehicle accelerating soon (B-3)



(b) Slower vehicle out of range late (B-2); slower vehicle accelerating late (B-4)



(c) Slower vehicle appears soon (B-5)



(d) Slower vehicle appears late (B-6)

Figure 3.12 – Lane change phase test cases of the proposed overtaking algorithm which are in connection with the (new) slower vehicle (numbers in the illustrations describe the sequence of events): (a) the platoon started the *lane change left* maneuver, but is still in the original lane when the slower vehicle is out of range or accelerates, (b) same situation as before, but the platoon is already in the left lane (but did not complete the *lane change left* maneuver yet), (c) when changing back to the original lane, a slower vehicle is suddenly detected on this lane; the platoon is still in the left lane, (d) same situation as before, but the platoon is already in the original lane.

the neighboring lane (the lane change maneuver is almost completed). The platoon aborts the maneuver and returns to the original lane. See Figure 3.12b.

- **Slower vehicle accelerating soon (B-3):** After the platoon started the *lane change left* maneuver, the slower vehicle accelerates and is no longer a slower vehicle. But in contrast to B-1 it is still detectable. The platoon aborts the maneuver as overtaking is not necessary anymore and returns to the center of the lane. See Figure 3.12a.

- **Slower vehicle accelerating late (B-4):** The situation is the same as in B-3, but the slower vehicle only accelerates when the platoon is already in the overtaking lane. The platoon returns to the original lane. See Figure 3.12b.

- **Slower vehicle appears soon (B-5):** After overtaking a slower vehicle and being in the *lane change right* maneuver, the platoon detects a former undetected vehicle in the *FR* area that is more than the safety distance away (meaning the vehicle does not represent a hazard). The platoon is still in the overtaking lane. The newly detected vehicle is slower than the platoon and overtaking is useful and possible. The platoon aborts the *lane change right* maneuver, goes back to the *passing* state, and overtakes this vehicle as well. See Figure 3.12c.

- **Slower vehicle appears late (B-6):** The situation is the same as in B-5, but the new slower vehicle appears only when the platoon is already mostly in the original lane. See Figure 3.12d.

The following test cases are in connection with a disrupting vehicle that is generating a potentially unsafe situation. Each test case is also checked for the criteria listed in Table 3.9.

- **Disrupting RL vehicle above threshold (B-7):** The platoon changes lanes to the left when a former undetected vehicle, which is driving with a higher speed than the platoon, is detected in the *RL* area. This disrupting vehicle has to decelerate more than the threshold ($-3.5\,\mathrm{m}/s^2$) for a *lane change left* maneuver considered as safe. The disrupting vehicle is not detectable by the leader but by the last follower who sends an *abort* message to the leader. The platoon aborts the *lane change left* maneuver and returns to the center of the lane. See Figure 3.13a.

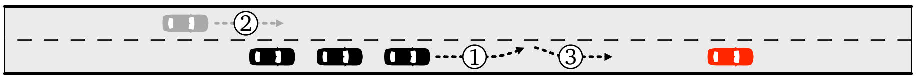
- **Disrupting RL vehicle below threshold (B-8):** The situation is the same as in B-7, but the former undetected vehicle does not have to brake that hard. Although the vehicle needs to brake harder than allowed in the decision phase (threshold $-1.0\,\mathrm{m/s^2}$), it still does not have to break so hard that the *lane change left* maneuver needs to be aborted (threshold $-3.5\,\mathrm{m/s^2}$). Thus, the platoon continues to overtake (each member must not visit the *changing back* state). See Figure 3.13b.

- **Disrupting FL vehicle (B-9):** The situation is the same as in B-7, but the former undetected vehicle appears in the *FL* area and is slower than the platoon. In contrast to B-7, the leader does not have to rely on his followers' sensors as he can detect the vehicle himself and therefore directly send an *abort* message to his followers. See Figure 3.13c.

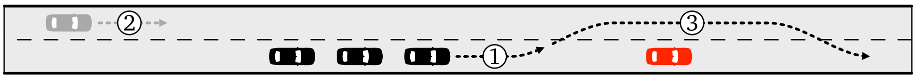
- **Disrupting vehicle beside (B-10):** The situation is the same as in B-7, but the former undetected vehicle appears next to the platoon. In contrast to B-7

Test case	Visited lanes	Vehicles overtaken	Important visited states
B-7, B-9, B-10	0	-	*changing back*
B-8	0, 1, 0	slower vehicle	-
B-11, B-12, B-13	0, 1	slower vehicle	*changing back*

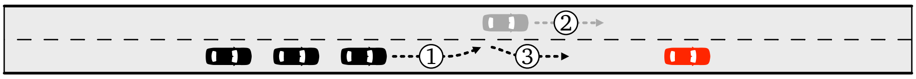
Table 3.9 – Individual checks for the lane change phase test cases which are in connection with a disrupting vehicle.



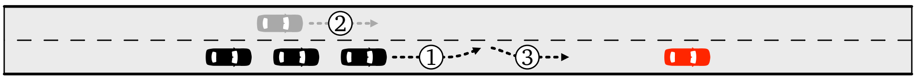
(a) Disrupting RL vehicle above threshold (B-7)



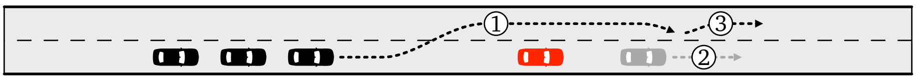
(b) Disrupting RL vehicle below threshold (B-8)



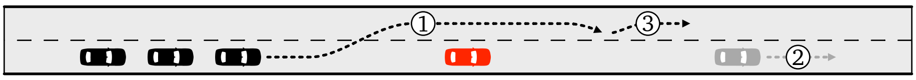
(c) Disrupting FL vehicle (B-9)



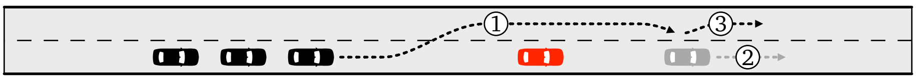
(d) Disrupting vehicle beside (B-10)



(e) Disrupting RR vehicle above threshold (B-11)



(f) Disrupting FR vehicle (B-12)



(g) Disrupting vehicle beside (B-13)

Figure 3.13 – Lane change phase test cases of the proposed overtaking algorithm which are in connection with a disrupting vehicle, that is not detected until the platoon is already changing lanes (numbers in the illustrations describe the sequence of events): (a) a disrupting vehicle appears in the *RL* area, (b) a disrupting vehicle appears in the *RL* area, but he does not have to brake that hard so the overtaking maneuver can be continued, (c) a disrupting vehicle appears in the *FL* area, (d) a disrupting vehicle appears next to the platoon, (e) a disrupting vehicle appears in the *RR* area, (f) a disrupting vehicle appears in the *FR* area, (g) a disrupting vehicle appears next to the platoon.

and B-9, the disrupting vehicle is detected by at least two members in the middle of the platoon. See Figure 3.13d.

- **Disrupting RR vehicle above threshold (B-11):** A former undetected vehicle appears in the *RR* area after the platoon passed the slower vehicle and started the *lane change right* maneuver. The platoon aborts the lane change and returns to the center of the overtaking lane. Then the situation is considered safe and therefore the simulation is terminated. See Figure 3.13e.

- **Disrupting FR vehicle (B-12):** The situation is the same as in B-11, but the former undetected vehicle appears in the *FR* area of the platoon and is detected by the leader. See Figure 3.13f.

- **Disrupting vehicle beside (B-13):** The situation is the same as in B-11, but the former undetected vehicle appears next to the platoon and is detected by at least two members in the middle of the platoon. See Figure 3.13g.

Finally, we tested the safety of the proposed algorithm under an unreliable V2V communication system.

- **Unreliable V2V communication system (B-14):** When overtaking is safe, the leader sends a message to his followers to change lanes. If this message is received with a substantial delay by at least one follower, this follower will not change lanes even though the other members do. They have to detect this by continuously monitoring the lateral position of the member in the front and in the rear. If the lateral offset is bigger than d_{lat}, they have to abort the maneuver by returning to the original lane (during a *lane change left* maneuver) or the overtaking lane (during a *lane change right* maneuver). See Figure 3.14.

As we have shown that the proposed algorithm resolves the potentially dangerous traffic situations, we will assess its performance in the next chapter.

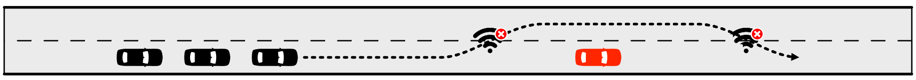


Figure 3.14 – Lane change phase test case B-14: overtaking under an unreliable communication system.

Chapter 4

Performance Evaluation

For our performance evaluation, we use traffic scenarios again. This time, however, they do not reflect a specific traffic situation (see Section 3.3), but are generated randomly.

The first subsection of this chapter describes the scenario set-up. Then the proposed algorithm is benchmarked against three other approaches: the best and worst case and the *auto lane change* algorithm of Plexe (see Section 2.3). Subsequently, we will analyze the impact of the variation of major scenario parameters on the performance of the proposed algorithm. Parameters that relate to the scenario (e.g., traffic density) and those that are related to the proposed algorithm (e.g., overtaking thresholds) are varied. This chapter concludes with a summary of the results and their interpretation.

4.1 Scenario Set-up

The main difference to the test cases presented in Chapter 3 is the randomly generated traffic flow. This is to ensure that the simulation comes close to typical road traffic on freeways. The Federal Highway Research Institute (Bundesanstalt für Straßenwesen, BASt) maintains automatic counting stations on freeways and federal highways in Germany. Their network currently comprises 1,914 counting stations of which 1,124 are on federal freeways.[44] Based on their latest data from 2018[45], we calculate the average traffic for three different traffic densities: low, medium, and high. Based on the reported average hourly traffic during daytime (6 a.m. to 10 p.m.), we determine the corresponding hourly traffic for cars and trucks for 3 lanes. The medium traffic density is computed on the basis of the average of all stations on freeways. The

[44] https://www.bast.de/BASt_2017/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/zaehl_-node.html (access date: 07/10/20)

[45] https://www.bast.de/BASt_2017/DE/Verkehrstechnik/Fachthemen/v2-verkehrszaehlung/Aktuell/zaehl_aktuell_node.html (access date: 07/10/20)

low (high) traffic density is computed on the basis of the average of the 20 % of all stations with the lowest (highest) traffic density, respectively. Our results are shown in Figure 4.1.

In each simulation run, the randomized traffic spawns at the left edge of the freeway (see Figure 4.2). It is 11 km long, consists of three lanes, and has a speed limit of 33.3 m/s for cars and 22.2 m/s for trucks.[46] Because the spawn frequency can be high, the passenger cars spawn in two lanes (lanes two and three, with one being the most right one). In order to take the normally higher speed on the third lane into account, the faster passenger cars spawn on this lane. The trucks enter the simulation in lane one. Because SUMO normally inserts the vehicles at a fixed time interval, an equally distributed random time offset is added to the planned spawn time.

The platoon, which consists of four passenger cars, enters the simulation after 180 s and drives for 10 km. This allows not only for a warm-up phase of the randomized traffic, but also ensures that the platoon cannot reach the first inserted vehicle before it reaches the 10 km goal. It enters the simulation at the beginning of the freeway and in lane one, too. A parallel inserted truck that would hinder the insertion of the platoon would be removed from the simulation.

The vehicle types and parameters are the same as in Section 3.3.1. However, the speed factor of cars and trucks is now sampled from a normal distribution (see Table 4.1).

[46] However, some vehicles will drive faster because of their speed factor being larger than 1.0.

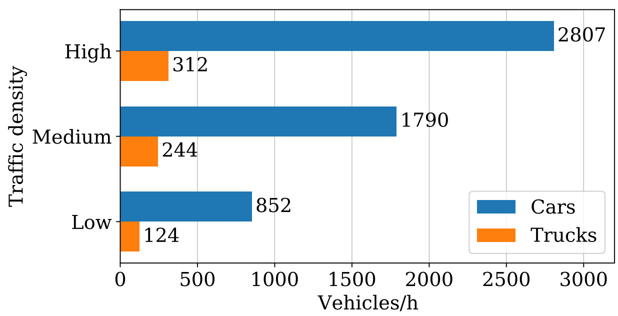


Figure 4.1 – Average hourly traffic during daytime (6 a.m. to 10 p.m.) in one direction on German freeways with three lanes (own calculations).

Figure 4.2 – The left edge of the freeway where the randomized traffic spawns: trucks in the right lane, slower cars in the middle lane and faster cars in the left lane.

Spawn lane	1 (Truck)	2 (Passenger car)	3 (Passenger car)
Mean	1.0	1.0	1.0
Deviation	0.2	0.2	0.2
Min value	0.875	0.75	1.0
Max value	1.25	1.0	1.25
Resulting speed	19.4 - 27.8 m/s	25.0 - 33.3 m/s	33.3 - 41.7 m/s

Table 4.1 – Parameters of the normal distribution the speed factor is sampled from and the resulting speed intervals for randomized trucks and cars.

4.2 Benchmarking

In this chapter, the algorithm is supposed to show its performance. Since there is no other communication-based overtaking algorithm for platoons so far, it is compared with the following three different cases:

1. **Best case:** the platoon is not delayed at all, because there is no slower vehicle in front. The platoon drives at its desired speed and cannot arrive faster at its destination than in this configuration.
2. **Worst case:** the platoon does not overtake a slower vehicle in front, because no overtaking algorithm is present. The platoon should not arrive later at its destination than in this configuration.
3. **Plexe:** the platoon is equipped with the non-communication-based lane change algorithm of Plexe as described in Section 2.3.

While the first and second case represent upper and lower limits regarding the overall performance of the proposed algorithm, the third one will be the real opponent. Each case is tested with multiple simulation runs with different random seeds.

First of all, we present one single traffic scenario with medium traffic density and discuss the major differences between the aforementioned cases. We then present the aggregated results for all runs.

4.2.1 Single Scenario

For the single scenario evaluation, we have chosen a scenario where the proposed algorithm achieves a higher average speed than the Plexe lane change algorithm, because in most cases in the performance evaluation the latter is slightly faster than our proposed algorithm. By this means, it can be examined which factors could lead to a better performance of the proposed algorithm.

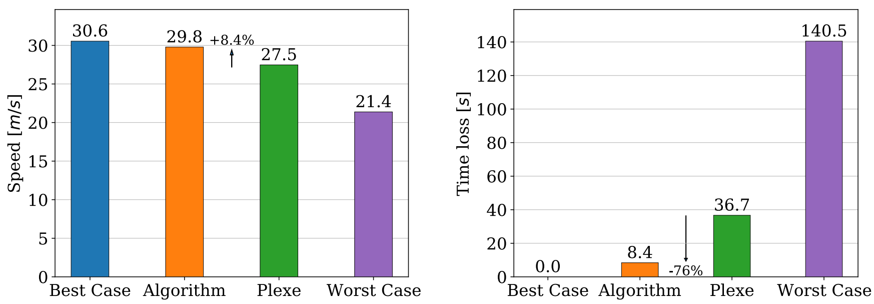
One main goal of overtaking is the reduction of total travel time. Therefore, the average speed and time loss[47] of the platoon are analyzed. By examining the overtaking behavior of the algorithms, we will detect how this speed advantage was achieved. The selected metrics for this are speed, acceleration, lateral lane position of the platoon, and the total distance traveled.

The overall results of the simulation are shown in Figure 4.3. The platoon can drive at 30.6 m/s in the best case, which is its desired speed (see Figure 4.3a). Being delayed in the worst case, its average speed is reduced to 21.4 m/s (−30.1 %) due to the slower traffic in front and the lack of opportunity to overtake. Under the control of the Plexe lane change algorithm, the platoon reaches an average speed of 27.5 m/s (−10.1 % compared to the best case). The proposed algorithm is almost as fast as the best case (29.8 m/s), making it 8.4 % faster than the Plexe algorithm.

The results for the time loss are accordingly (see Figure 4.3b). While the platoon only looses 8.4 s when controlled by the proposed algorithm during the 10 km drive, the Plexe algorithm leads to a higher time loss of 36.7 s, which is about one quarter of the worst case (140.5 s).

The reasons for these results are presented in Figure 4.4, which shows the platoon's speed, lane position, acceleration, and distance traveled over time. While the platoon does not change its speed in the best case (and arrives first), the platoon has a slower vehicle in front shortly after the start in the other cases. If not overtaking this vehicle, the platoon would need to decelerate to about 20.5 m/s (as it does in the worst case) (see Figure 4.4a). The proposed algorithm in contrast successfully begins

[47] Time loss is defined as the time that is being lost due to not driving at the desired speed. The metrics are recorded by the output capabilities of SUMO.



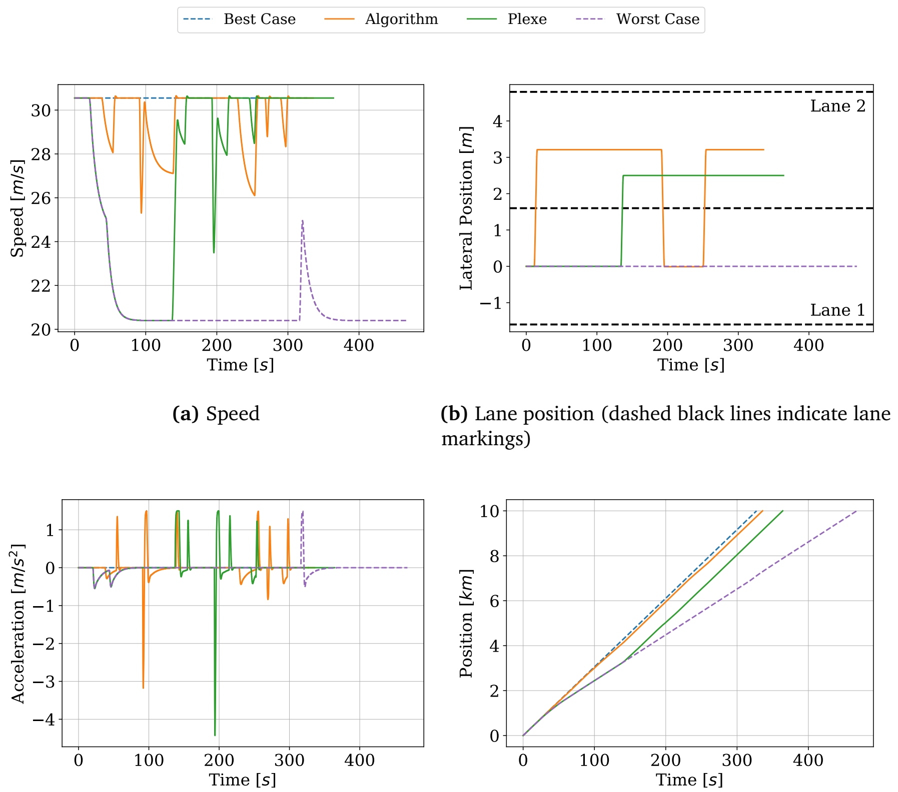
(a) Average speed

(b) Time loss

Figure 4.3 – The results of a (non-representative) single simulation run with medium traffic and a platoon of length four: the platoon's (a) average speed and (b) time loss.

an overtaking maneuver shortly after the slower vehicle is detected (see Figure 4.4b) and therefore does not need to decelerate much (see Figure 4.4a). The corresponding traffic situation is shown in Figure 4.5. The Plexe algorithm changes lanes much later and therefore has to drive longer at a lower speed. In this case, it may have become a victim of his own omniscience about the simulation and wanted to let other vehicles pass first.[48] For the sake of completeness, the speed peak in the worst case after 300 s should be explained: the platoon drives behind a vehicle, which in turn drives behind a slower vehicle. The vehicle in front overtakes the slower

[48]It should be mentioned that the proposed algorithm does not interfere with either the vehicle behind the platoon or the next one approaching in the middle lane. Both vehicles do not have to brake.



(a) Speed

(b) Lane position (dashed black lines indicate lane markings)

(c) Acceleration

(d) Distance traveled

Figure 4.4 – Detailed results of a single simulation run with medium traffic and a platoon of length four: the platoon's (a) speed, (b) lane position (dashed black lines indicate lane markings), (c) acceleration, and (d) distance traveled.

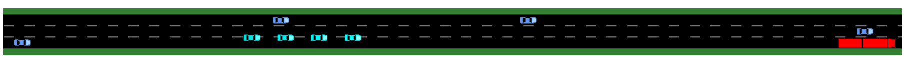


Figure 4.5 – The proposed algorithm successfully executes a lane change to the left.

vehicle after 300 s and the platoon can catch up with the other slower vehicle by accelerating.

After the algorithms have made different decisions, the traffic situations can no longer be properly compared, because an earlier decision also influences the behavior of the other vehicles. However, it is noticeable that the Plexe algorithm only changes lanes once and remains there. Although the middle lane is in most cases faster than the right one, this behavior does not allow the algorithm to catch up (see Figure 4.4d).[49]

Although the platoon had to brake harder one time (see Figure 4.4c) when controlled by the proposed algorithm, the phases at low speed are shorter than with the Plexe algorithm. This, combined with the early lane change, results in better overall performance in terms of total travel time.

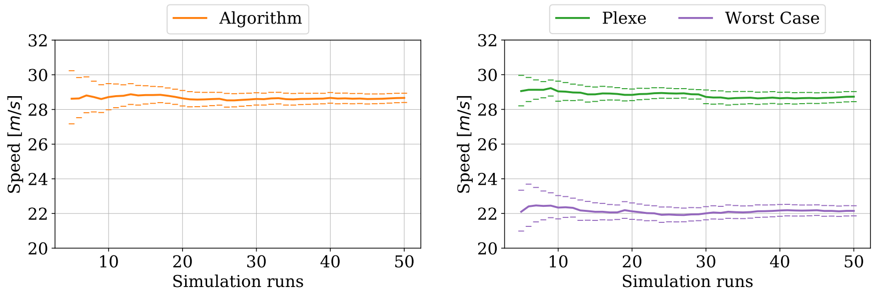
4.2.2 Multiple Scenarios

Analyzing a single scenario allows insights into what can lead to a good or bad performance of the algorithms. But it is also important to examine what the average performance is. To achieve this, we analyzed 40 randomized scenario runs per case and computed the average speed and lateral lane position[50] of the platoon. The average speed corresponds to what an onboard unit would report after the 40 simulation runs, meaning all speeds in all time steps of all 40 runs are added and divided by the number of time steps.[51] Additionally, we report the average time until the first lane change and the average number of lane change maneuvers. The use of more than 40 simulation runs does not lead to significantly different results. The 0.95 confidence interval regarding the average speed of 40 runs of the proposed algorithm is $[28.36\,\mathrm{m/s}, 28.98\,\mathrm{m/s}]$ with a width of 0.62 m/s. The interval becomes only 0.08 m/s smaller when 50 runs are taken into account (see Figure 4.6).

The results of the 40 runs (*benchmarking scenario*) are shown in Figure 4.7. On average, the best case (30.6 m/s) is 37.8 % faster than the worst case (see Figure 4.7a). Both the proposed algorithm and the Plexe lane change algorithm are 2.0 m/s slower than the best case (−6.5 %) and 28.8 % faster than the worst case. This is a good result for the proposed algorithm, especially because the proposed algorithm changes lanes later for the first time than Plexe's algorithm (see Figure 4.7c), which could be a disadvantage. But it changes lanes more often (see Figure 4.7d), meaning a later lane change could be compensated by more lane changes.

In addition, the algorithms are much closer to the best case than the worst case. The main reason for this is the duration of driving in the middle lane. The average lateral position during all 40 simulation runs is shown in Figure 4.7b. In the worst

[49] We will further analyze this in the next section.
[50] The latter gives insights in which lane the platoon tends to drive.
[51] So this is not the average of 40 average speeds.



(a) Proposed algorithm

(b) Plexe algorithm and worst case

Figure 4.6 – The average speed depending on the number of simulation runs (solid line) and the corresponding 0.95 confidence intervals (short horizontal lines) for (a) the proposed algorithm and (b) the Plexe algorithm and the worst case.

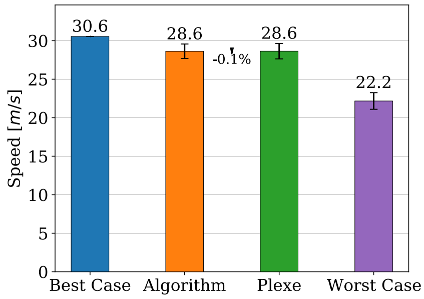
and best case, the lateral position is zero (which is the center of lane one), because the platoon never leaves this lane. When controlled by one of the two algorithms, the lateral position should be bigger than zero. The closer the value gets to 3.2 m (the center of lane two), the more the platoon drives in that lane. The platoon's average lateral position under both algorithms is already in lane two (1.84 m for the proposed algorithm and 1.75 m for Plexe).[52] Therefore, the platoon is driving more in the middle, and probably faster, lane. We will now further analyze this assumption.

To do so, we manipulated the proposed algorithm in a way that the platoon will determine the speed of the overtaking lane not only by the minimum of the desired speed and the speed limit as in Equation (3.1), but also by the speed of a vehicle detected in that lane:

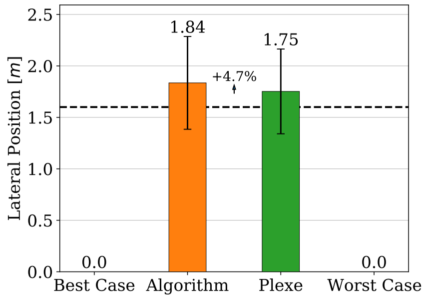
$$\min(v_{\text{desired}}, v_{\text{FL}}, v_{\text{limit}}) - v_{\text{F}} \geq v_{\Delta}. \tag{4.1}$$

The platoon will not overtake if a vehicle in the overtaking lane makes the lane slower than the threshold. One could argue that this is correct behavior, because without the *FL* vehicle taken into account the overtaking time threshold $t_{\text{overtaking}}$ could be violated due to that vehicle. But this results, e.g., in the actually unrealistic behavior that the platoon does not follow a vehicle which itself is changing to the overtaking lane (*piggyback overtaking* [18]). Instead, it waits for the vehicle to either become fast enough or to complete the overtaking maneuver. When driven by the manipulated algorithm, the platoon should be more in lane one and its average speed

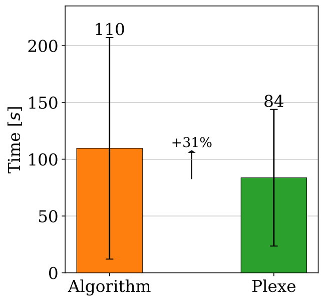
[52] Actually, the value for the Plexe platoon should be a little higher. Plexe directs the platoon not towards the center of the middle lane (as the proposed algorithm does), but only to its edge (see Figure 4.4b). As a result, the platoon's lateral position for lane two is lower.



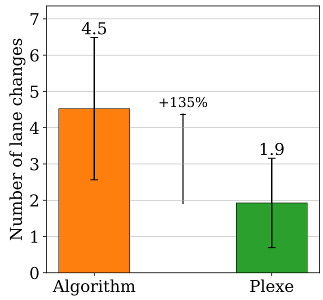
(a) Average speed with standard deviation



(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)



(c) Average time until the first lane change with standard deviation

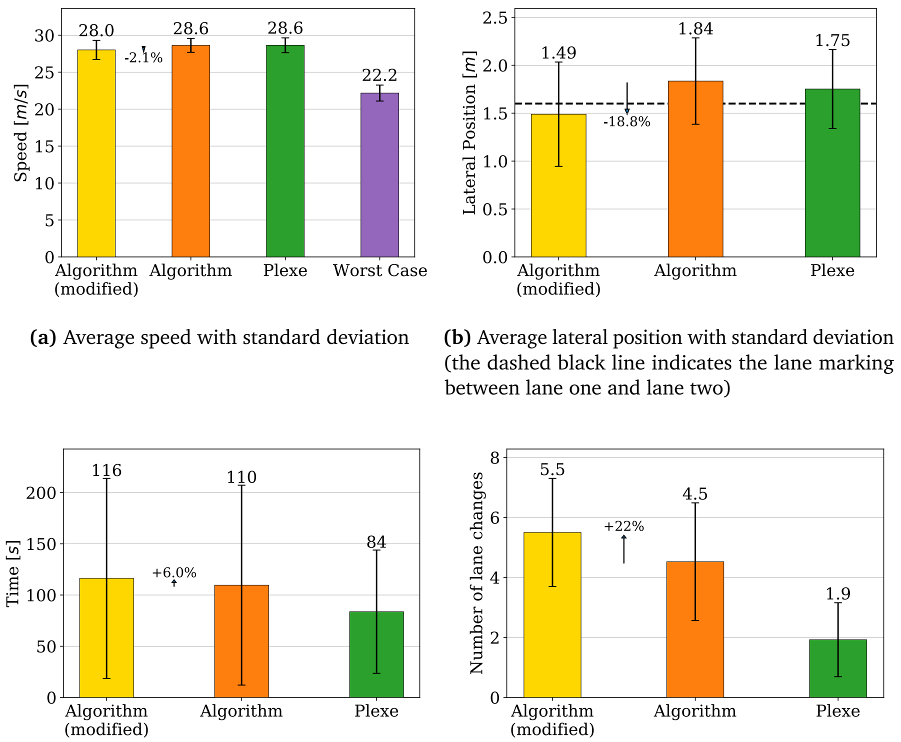


(d) Average number of lane changes with standard deviation

Figure 4.7 – The results of 40 simulation runs with medium traffic and a platoon of length four (*benchmarking scenario*): the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

should decrease. The results are shown in Figure 4.8. As expected, the platoon is now driving mainly in lane one (lateral position 1.49 m, see Figure 4.8b). This results in a reduction of the average speed to 28.0 m/s (−2.1 %, see Figure 4.8a). The platoon overtakes later (6.0 %, see Figure 4.8c) and executes one more lane change (see Figure 4.8d). On average, the platoon becomes slower as when controlled by the Plexe algorithm.

To support these findings, we have analyzed whether there is a correlation between the lateral position and the achieved average speed. We have run 650 simulations of the proposed algorithm (with four members in medium random traffic density). The results (see Figure 4.9) show that there is a moderate correlation (correlation coefficient of 0.58). The correlation is more obvious below a lateral position of 1.6 m (which is the divider between lane one and lane two). Both the



(a) Average speed with standard deviation

(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)

(c) Average time until the first lane change with standard deviation

(d) Average number of lane changes with standard deviation

Figure 4.8 – The simulation results if the proposed algorithm determines the overtaking lane's speed also by taking a vehicle in that lane into account: the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

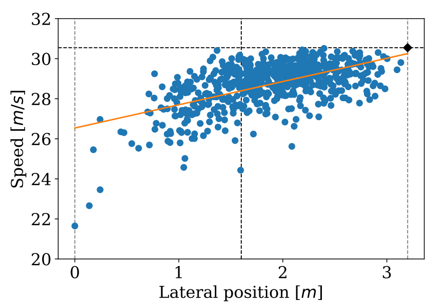


Figure 4.9 – Correlation between the platoon's lateral position and its average speed for 650 simulation runs. The line is a linear function and the black diamond in the upper right corner indicates the platoon's limits due to its desired speed of 30.6 m/s and the center of lane two at 3.2 m.

center of lane two at 3.2 m and the platoon's desired speed of 30.6 m/s represent limits for the platoon (indicated by the black diamond in the upper right corner of the figure). The closer the platoon gets towards its desired speed, the more difficult it becomes to achieve a further increase in the average speed.

In addition, we have analyzed whether an earlier lane change results in a higher average speed. But our 650 simulation runs only show a weak correlation (correlation coefficient of −0.46).

In conclusion, the middle lane is faster on average than the right one. A vehicle that drives more in the middle lane can therefore be faster than a vehicle in the right lane. However, changing lanes early does not guarantee a high average speed.

4.3 Parameter Studies

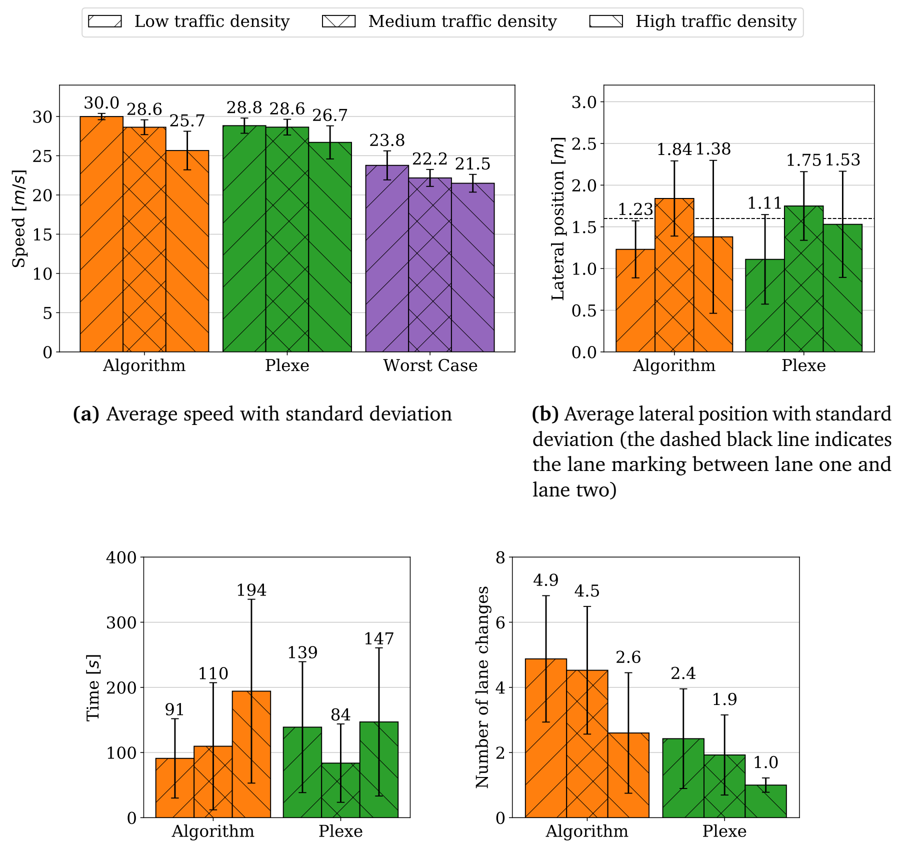
In a next step, we will analyze the influence of certain parameters on the algorithms. Two different kinds of parameters can be distinguished: first, parameters that influence the scenario (e.g., traffic density), and therefore, all approaches to handle slower traffic, and second, parameters that only have an effect on the proposed algorithm (e.g., the overtaking thresholds v_Δ and $t_{\text{overtaking}}$). We will start with the first ones. The main metric will be the platoon's average speed, because a short total travel time is a typical goal of road users. In addition, the average lateral position of the platoon will be reported as well as the time until the first lane change occurs and how many lane changes are executed during the 10 km drive of the platoon. The parameters are the same as in the previous section, with the exception of the parameter mentioned to be varied in each case. In particular, we will again evaluate 40 simulation runs with randomized (medium) traffic.

4.3.1 Scenario-Specific Parameters

The following parameters are of relevance to the scenario and therefore influence the performance of both algorithms and the worst case. The best case remains the same and is not further considered.

4.3.1.1 Traffic Density

In the last section, the traffic density was set to medium. Now we want to modify this parameter and set it to a low or high density as defined in Figure 4.1. The results are shown in Figure 4.10. While the proposed algorithm can increase its average speed to 30.0 m/s (4.9 %) at low traffic density, the Plexe lane change algorithm's average speed remains almost unchanged (see Figure 4.10a). Both algorithms keep the platoon mostly in lane one (see Figure 4.10b), which was expected as fewer traffic



(a) Average speed with standard deviation

(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)

(c) Average time until the first lane change with standard deviation

(d) Average number of lane changes with standard deviation

Figure 4.10 – The results when the traffic density is varied from low to high: the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

results in lesser slower vehicles and therefore a lesser need for overtaking. And if the platoon overtakes, it can sooner return to the original lane. But Plexe's algorithm changes lanes later for the first time compared to the proposed algorithm (139 s vs. 91 s, see Figure 4.10c) and drives less in the middle lane (1.11 m vs. 1.23 m, see Figure 4.10b), which results in the lower average speed.

While the proposed algorithm outperforms Plexe's lane change algorithm in low traffic density, it is the other way round when the platoon drives in high traffic. The average speed is reduced under both algorithms because it becomes more difficult to find a gap in the overtaking lane, which makes overtaking more difficult. But while the proposed algorithm looses 2.9 m/s (−10.1 %), Plexe only looses 1.9 m/s (−6.6 %, see Figure 4.10a). The main reason for this is the fact that Plexe executes only one lane change on average (see Figure 4.10d). This means that the platoon does on average not return to the original lane during its 10 km drive, which results in a higher lateral position of 1.53 m compared to the proposed algorithm (1.38 m, −9.8 %, see Figure 4.10b). Therefore, the platoon drives longer in the faster lane and has a higher average speed. The proposed algorithm, on the other hand, does not only change lanes later for the first time than Plexe (194 s vs. 147 s, see Figure 4.10c), but also executes still 2.6 lane changes on average in high traffic density. This results in a lower average speed compared to Plexe, because the platoon will most likely be stuck behind another slow vehicle again after returning to the original lane.

In addition, the proposed algorithm does not change lanes at all in four out of 40 simulation runs,[53] which explains the high standard deviation of the lateral position and the late average first lane change after 194 s (see Figure 4.10c). Plexe does not change lanes in only one simulation run and performs better accordingly.

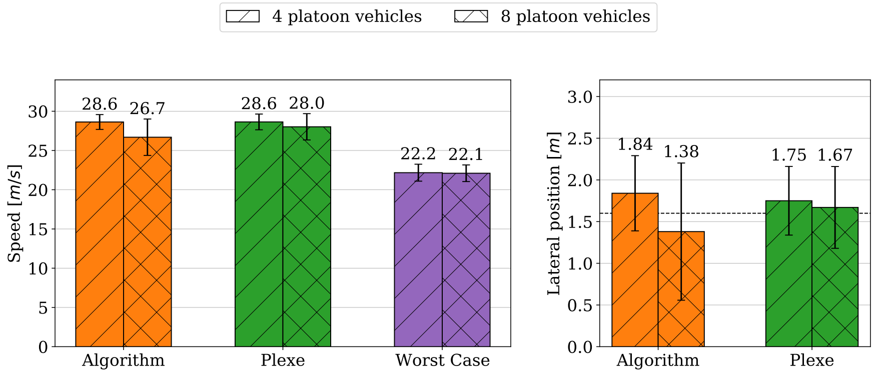
4.3.1.2 Platoon Length

The length of the platoon can also have an impact on the algorithm's performance. This is because it will be more difficult for a long platoon to find a suitable gap in the overtaking lane. For the execution of the proposed algorithm, however, the length of the platoon is of no relevance: all followers assess their maneuver area simultaneously. Only the utilization of the V2V communication system is increased.[54] To evaluate the performance of the proposed algorithm, we added four more vehicles to the platoon, eight vehicles in total. The results are shown in Figure 4.11: the average speed is reduced for both algorithms. But while the Plexe platoon's speed is reduced only by 0.6 m/s (−2.1 %), the effect is bigger for the proposed algorithm's platoon (see Figure 4.11a). Its average speed is reduced by 1.9 m/s (−6.6 %) and the standard deviation has increased.

It is noticeable that for the proposed algorithm the results are practically the same, compared to the high traffic density scenario (with exception of the average speed). Further analysis of the overtaking behavior shows that the reasons are the same as if the traffic density was increased (increasing the traffic density or the platoon length both make it harder to find a suitable gap in the overtaking lane). Under the proposed algorithm the platoon drives less in the second lane (average lateral position 1.38 m, −17.4 % compared to Plexe) (see Figure 4.11b) and again does not change lanes in four out of 40 simulation runs (Plexe: one time). But the loss of speed is not as big as in the high traffic density scenario although the average lateral position is the same. The reason for this is that the platoon can, when successfully changed to the overtaking lane, drive faster as there is less traffic compared to the dense traffic scenario.

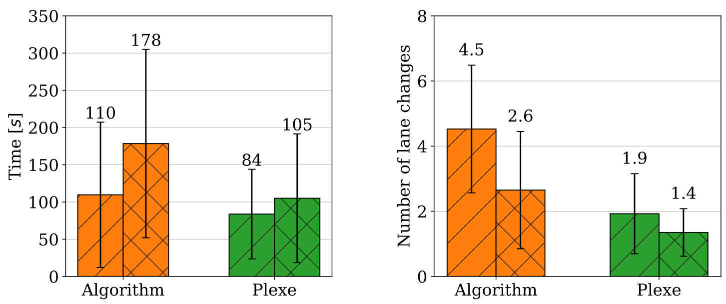
[53] Not changing lanes at all corresponds to the worst case.

[54] Which could become a problem if the utilization is already high, but this is not considered here.



(a) Average speed with standard deviation

(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)



(c) Average time until the first lane change with standard deviation

(d) Average number of lane changes with standard deviation

Figure 4.11 – The results when the number of platoon members is varied: the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

Plexe on the other hand changes lanes for the first time after 105 s (21 s more than with 4 vehicles) and therefore finds a gap more easily. The performance of Plexe hardly drops with an increased number of platoon members. Yet, it must be mentioned that Plexe on average did not complete a single overtaking maneuver during the 10 km drive (see Figure 4.11d).

4.3.1.3 Platooning Controller

Plexe offers different platooning controllers and we normally use the CACC [31] for the followers. But we also tested the Ploeg controller [32] with parameters set according to Table 4.2. The results are very similar to those with the CACC controller for both the proposed algorithm and Plexe and are therefore not shown here.

Parameter	Value
Proportional gain	0.2
Derivative gain	0.7
Time headway	0.16 s

Table 4.2 – Parameters of the Ploeg controller [32].

4.3.2 Proposed Algorithm-Specific Parameters

We will now evaluate the change of parameters that are specific to the proposed algorithm. The order of the parameters under consideration is based on overtaking maneuver conditions.

4.3.2.1 Backoff

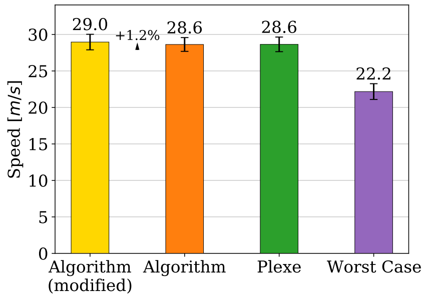
The fact that Plexe finds a gap in the overtaking lane more easily could be a result of a too long exponential backoff that is set when the leader waits before attempting a new lane change (*lane change aborted* state) and the followers wait for a lane change decision of the leader (*wait for decision* state). Therefore, we eliminated the backoff. This leads to an increased usage of the V2V communication system because, even if overtaking is not possible due to other traffic for a longer time, messages for assessing the maneuver area are sent frequently between the platoon members.

Our simulation results did not support the assumption that the backoff has an influence on the ability to find a gap in the overtaking lane because the average speed remains the same even when the backoff phase is switched off.

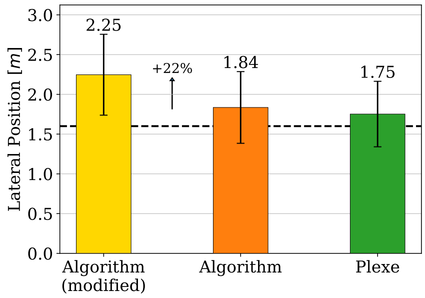
4.3.2.2 Overtaking Thresholds

When assessing whether overtaking is useful and possible, two thresholds have to be taken into account due to legal requirements in Germany: the time needed for overtaking and the possible speed in the overtaking lane. These thresholds are derived from court rulings, but it might still be interesting to see the impact of disregarding these thresholds on the performance of the proposed algorithm. Therefore, we eliminated the time restriction $t_{\text{overtaking}}$ and set the minimum speed delta v_{Δ} to 0.1 m/s. This results in an increased average speed of 29.0 m/s, which is a plus of 1.2 % (see Figure 4.12a) compared to both the unmodified algorithm and Plexe's lane change algorithm.[55] Due to the unrestricted overtaking behavior, the platoon now drives significantly more in the second lane. The average lateral position increases to 2.25 m (22 %, see Figure 4.12b). This does not only result

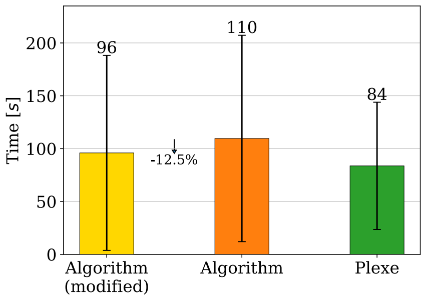
[55] Because Plexe's lane change algorithm does not consider such thresholds to the best of our knowledge, this comparison would be "fairer" than the one presented in Section 4.2.



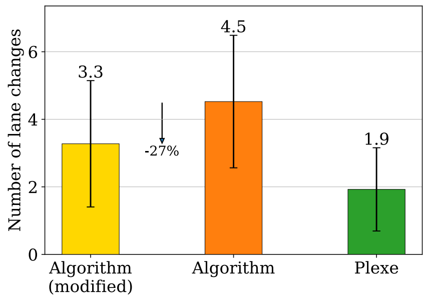
(a) Average speed with standard deviation



(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)



(c) Average time until the first lane change with standard deviation



(d) Average number of lane changes with standard deviation

Figure 4.12 – The results when the overtaking thresholds $t_{\text{overtaking}}$ and v_{Δ} are neglected: the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

in a shorter time until the first lane change (−12.5 %, see Figure 4.12c), but also in a reduction of lane changes (−1.2, see Figure 4.12d). The reason for this is that overtaking became easier and thus earlier possible. And when driving in the overtaking lane, there is less need to return to the original lane because overtaking a vehicle riding only a little slower than oneself is allowed.

We also tested the proposed algorithm's performance with the minimum speed delta v_{Δ} reduced from 2.7 m/s to 1.35 m/s and the maximum overtaking time $t_{\text{overtaking}}$ increased from 45 s to 90 s. This results in an average speed exactly between the performance with standard thresholds and without thresholds: 28.8 m/s.

4.3.2.3 Minimum Distance to the RL Vehicle

When asserting whether the maneuver area is free, each platoon member computes a minimum distance to the *RL* vehicle (see Equation (3.63)). One parameter of this formula defines how hard this vehicle should have to brake at most when the platoon moves in its lane. The standard value was chosen to reflect the behavior of a typical driver. Now we set this value to $-7.5\,\mathrm{m/s^2}$, which is the maximum possible deceleration of a passenger car in the simulation.[56] Our results do not show any significant performance changes: the time until the first lane change is somewhat reduced, but it has no noticeable impact.[57] It is relatively rare that a situation arises in which this reduced parameter comes into play, which is the reason why there is no significant change in performance.

4.3.2.4 When to Change to the Left

The proposed algorithm changes lanes to the left as soon as the slower vehicle is detected and the maneuver area is free. This ensures that the platoon does not miss a gap in the overtaking lane. But it could change lanes later to continue driving as long as possible on the original lane, minimizing potentially superfluous lane changes. Therefore, we modified the algorithm in a way that the platoon changes to the overtaking lane when the distance to the slower vehicle is at most three times the safety distance.[58] The results in Figure 4.13 only show a slight reduction of the average speed ($-0.3\,\mathrm{m/s}$, see Figure 4.13a) and the average lateral position (see Figure 4.13b). The average time until the first lane change did slightly increase accordingly (see Figure 4.13c). These effects are rather small, meaning it is not that important to change lanes as soon as the slower vehicle is detected. If optimizing the traffic flow is a main goal (instead of the platoon's total travel time), the platoon can overtake later without the performance deteriorating significantly.

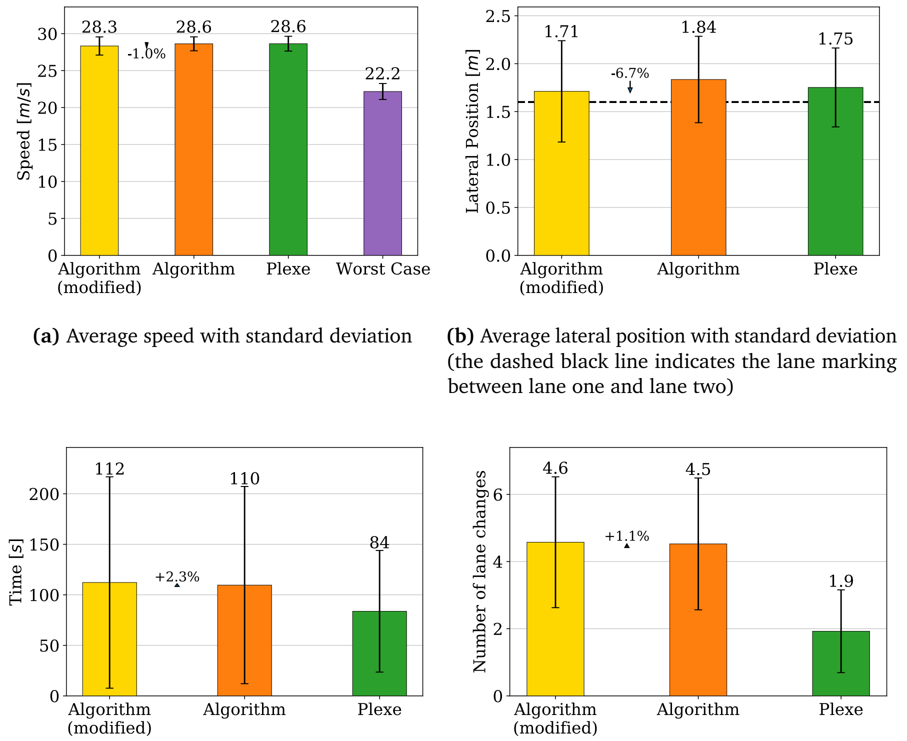
4.3.2.5 When to Change Back

The decision to move back to the original lane depends on how long the platoon wants to drive there (parameter t_{stay}) after the lane change without overtaking becoming possible and useful again. We have doubled t_{stay} from 10 s to 20 s, but it does not change the performance of the proposed algorithm. The average number of lane changes is somewhat reduced (from 4.5 to 4.3) because the platoon can

[56] The corresponding value for the *RR* vehicle remains unchanged because an overtaken vehicle should never have to brake when the overtaker returns to the original lane.

[57] Even if it would have improved the performance, requiring another vehicle to brake with maximum deceleration would not be acceptable.

[58] A shorter distance is not advisable because the ACC brakes early in order to smoothly adapt to the slower vehicle's speed. Therefore, it would take a rather long time to reach, e.g., a distance of two times the safety distance.



(a) Average speed with standard deviation

(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)

(c) Average time until the first lane change with standard deviation

(d) Average number of lane changes with standard deviation

Figure 4.13 – The results when the platoon changes lanes to the left when the slower vehicle is at most three times the safety distance away: the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

sometimes overtake another vehicle instead of returning to the original lane. But this behavior does not have an impact on the average speed.

We also set the time to 30 s and the traffic density to high in order to evaluate whether less returning to the original lane would result in a better performance in high traffic density. Indeed, the average lateral position increased from 1.38 m in the high traffic density scenario (see Section 4.3.1.1) to 1.45 m in this scenario, but the average speed remains unchanged. The performance in high traffic density is not improved.

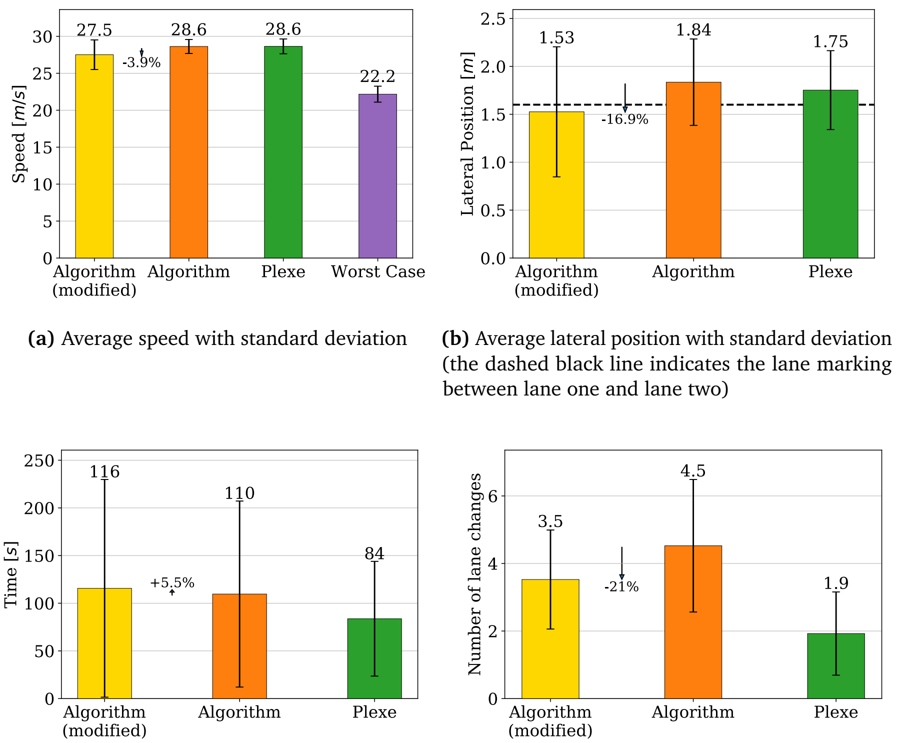
4.3.2.6 Unreliable Communication System

Finally, we analyzed the performance of the proposed algorithm if the V2V communication system is unreliable. Then more time is needed to successfully inform all followers to change lanes and the opportunity to change lanes originally detected by the leader could have passed by then. This should reduce the algorithm's performance.

The probability when a sent message will be received is exponentially distributed. We chose the mean $\lambda = 5$ steps.[59] The results of the 40 simulation runs with medium traffic density and a platoon of length four are shown in Figure 4.14.[60] The lateral position of the platoon is reduced by $-16.9\,\%$ (see Figure 4.14b). As a result, the

[59] 20 steps are the standard timer in the *wait for responses* state.

[60] The performance of the Plexe lane change algorithm remains unchanged because it does not use a V2V communication system.



(a) Average speed with standard deviation

(b) Average lateral position with standard deviation (the dashed black line indicates the lane marking between lane one and lane two)

(c) Average time until the first lane change with standard deviation

(d) Average number of lane changes with standard deviation

Figure 4.14 – The results when the V2V communication system is unreliable: the platoon's (a) average speed, (b) average lateral position (the dashed black line indicates the lane marking between lane one and lane two), (c) average time until the first lane change, and (d) the average number of lane changes. Vertical lines indicate the standard deviation.

average speed is also reduced (−3.9 %, see Figure 4.14a), which was expected due to the delay in message transmission. The time until the first lane change is now longer (5.5 %, see Figure 4.14c) for the same reasons and the average number of lane changes is reduced by −21 % (see Figure 4.14d).

4.4 Discussion

The performance of both algorithms lies between the worst and best case, which was to be expected. Yet, it is much closer to the best case for both algorithms. Overtaking is, in terms of total travel time, significantly better than staying behind slower vehicles.

A reliable performance comparison of the two algorithms turns out to be difficult for two reasons. Firstly, the Plexe algorithm is still experimental. On the one hand, this has led to collisions with other vehicles from time to time and, on the other hand, its overall performance can probably be further improved.[61] Secondly, the Plexe algorithm is not a communication-based overtaking algorithm. It can use all information available from the simulation environment which in reality could be understood as V2V communication. However, this assumes that this technology is available to all road users. But this is currently not the case. The proposed algorithm, in contrast, only requires V2V communication capabilities between the platoon members. A functionality that already exists anyways, since platooning would normally not be possible otherwise. Surrounding traffic is not required to have any form of communication capabilities.

With that in mind, the algorithms can still be compared: the proposed algorithm performs as good as Plexe's algorithm in the benchmarking scenario. It can perform even better under certain circumstances (as in the low traffic density scenario), but in general the performance of Plexe's algorithm is better.[62] The parameter study has shown that the performance of both algorithms is relatively robust. Changes in the average speed occur in most scenarios, but in no case they are particularly strong.

Generally speaking, the proposed algorithm changes lanes more often than Plexe's lane change algorithm. In the benchmarking scenario, Plexe only changes lanes 1.9 times on average (the proposed algorithm: 4.5 times), meaning it does not complete a full overtaking maneuver during the 10 km drive. This is not necessarily a disadvantage of Plexe's algorithm. Lane changes, despite all safety checks, are still a risky maneuver [33].[63] And if they are executed frequently, the risk of accidents increases. Plexe achieves the same average speed as the proposed algorithm with less lane changes. This can mainly be explained by the fact that the Plexe lane

[61] Which also applies to the proposed algorithm.
[62] But only a little and this could be due to its omniscient knowledge.
[63] At least if executed in a traffic environment that is not fully automated.

change algorithm keeps the platoon in the (normally faster) middle lane longer than the proposed algorithm, and does not return to the right lane even if it could – and should, according to the obligation to drive on the right hand side.[64]

The proposed algorithm, in contrast, tries to comply with this obligation by returning to the right lane as soon as necessary. But this behavior bears a disadvantage: it happened that the platoon changed back to the right lane and shortly afterwards a slower vehicle was in front of it again. However, it was not possible to overtake immediately as the traffic in the neighboring lane did not allow for this. For this reason, the platoon was obliged to decelerate and loose time, a situation the Plexe algorithm tends to avoid. If the platoon would have stayed in the middle lane, it could still drive at this lane's (higher) speed.

Plexe also changes lanes sooner for the first time. This was particularly observed in the scenarios with high traffic density and a longer platoon. In both cases, it becomes more difficult to find a gap in the overtaking lane. Further improvements of the proposed algorithm should focus on this. Nevertheless, the proposed algorithm not only proved that it can handle typical dangerous traffic situations, but can also compete against an omniscient lane change algorithm. It serves therefore as a very good basis for future improvements.

[64] We evaluated this by monitoring a couple of simulation runs in the SUMO Graphical User Interface (GUI). The platoon did not return to the right lane from time to time, even when it was free for a longer period.

Chapter 5

Conclusion

In this thesis, we propose a cooperative overtaking algorithm for platoons on freeways and theoretically define it by the use of FSMs. Based on the actual traffic situation, the algorithm decides whether overtaking a slower vehicle in front of the platoon is possible, useful, and safe. If that is the case, it carries out the overtaking maneuver cooperatively and reacts appropriately to unsafe situations by returning the platoon to a safe state. The algorithm has only a few technical requirements for the platooning vehicles that go beyond what is necessary for standard platooning. In particular, no communication with other road users is required. In addition, all platoon members work simultaneously, so that the system scales well.[65]

We implemented the algorithm in a simulation environment and successfully tested its behavior in defined, potentially dangerous traffic scenarios. We also benchmarked the algorithm in randomly generated traffic and compared its performance with the worst case (not overtaking at all), the best case (no slower traffic in front of the platoon), and Plexe's already integrated lane change algorithm that implies omniscient vehicles. The results are promising: compared to the omniscient cooperative lane change algorithm from Plexe, our algorithm can achieve the same average speed in the average traffic dense scenario.

Despite the very good results, there is room for improvement. One of the main goals of the proposed algorithm is to overtake cooperatively without splitting up the platoon, but in some cases this could be useful behavior. Further research could analyze under which conditions this would be advisable. The proposed algorithm can be adapted to this behavior as the necessary data for such a decision are already available to the leader.

The incorporation of V2V communication with the surrounding road users and especially with the slower vehicle could simplify an overtaking maneuver. This would

[65]The leader is a single point of failure, though. But this is a general problem of platooning. A way to avoid this is proposed, for example, in [34].

not only ensure that the slower vehicle does not get any faster during the overtaking maneuver (which was mentioned in Chapter 1 as ClOA). It could also inform the platoon of additional slow vehicles in front or other hazards that could endanger the platoon during overtaking. Another step in this direction could also include V2X communication, e.g., with traffic signs or other kinds of road-side infrastructure.

If an overtaking maneuver would take too much time and is therefore not possible, some driving tactics could be incorporated in order to decrease the time needed. The platoon could, e.g., back off a little from the slower vehicle, accelerate again, and use this additional speed to reduce the overtaking time.

Another interesting improvement of the proposed algorithm is the modification towards a *driving in the best lane* algorithm. While being in the passing state it could be useful to overtake a car in this lane as well. The algorithm could therefore unite the *vehicle ahead* and the *passing* states of the overtaking FSM and monitor the *F* as well as the *FR* area simultaneously. Based on this, it could decide whether to stay in the current lane, to change to the left (and pass the *F* vehicle), or to change back to the right. This extension of the proposed algorithm would lead to an even more flexible algorithm.

List of Abbreviations

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance System
BASt	Federal Highway Research Institute (Bundesanstalt für Straßenwesen)
CACC	Cooperative Adaptive Cruise Control
CC	Cruise Control
ClOA	Collaborative Overtaking Assistance
COA	Cooperative Overtaking Assistance
FSM	Finite State Machine
GPS	Global Positioning System
GUI	Graphical User Interface
ITS	Intelligent Transportation System
OLG	Higher Regional Court (Oberlandesgericht)
PATH	Program on Advanced Technology for the Highway
POMDP	Partially Observable Markov Decision Process
SARTRE	Safe Road Trains for the Environment
SUMO	Simulation of Urban MObility
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything

List of Figures

List of Tables

Bibliography

[1] P. Petrov and F. Nashashibi, “Modeling and Nonlinear Adaptive Control for Autonomous Vehicle Overtaking,” *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 15, no. 4, pp. 1643–1656, Aug. 2014. DOI: 10.1109/TITS.2014.2303995.

[2] M. Düring, K. Franke, R. Balaghiasefi, M. Gonter, M. Belkner, and K. Lemmer, “Adaptive Cooperative Maneuver Planning Algorithm for Conflict Resolution in Diverse Traffic Situations,” in *International Conference on Connected Vehicles and Expo (ICCVE 2014)*, Vienna, Austria: IEEE, Nov. 2014, pp. 242–249. DOI: 10.1109/ICCVE.2014.7297550.

[3] A. Marefat, R. Aboki, A. Jalooli, E. Shaghaghi, M. R. Jabbarpour, and R. M. Noor, “An adaptive overtaking maneuver assistant system using VANET,” in *IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob 2014)*, Bali, Indonesia: IEEE, Aug. 2014, pp. 316–321. DOI: 10.1109/APWiMob.2014.6920314.

[4] O. Sawade and I. Radusch, “Survey and Classification of Cooperative Automated Driver Assistance Systems,” in *82nd IEEE Vehicular Technology Conference (VTC 2015-Fall)*, Boston, MA: IEEE, Sep. 2015, pp. 1–5. DOI: 10.1109/VTCFall.2015.7391161.

[5] A. Lombard, F. Perronnet, A. Abbas-Turki, and A. E. Moudni, “On the cooperative automatic lane change: Speed synchronization and automatic courtesy,” in *IEEE/ACM Conference on Design, Automation and Test in Europe (DATE 2017)*, Lausanne, Switzerland: IEEE, Mar. 2017, pp. 1655–1658. DOI: 10.23919/DATE.2017.7927259.

[6] S. v. d. Hoef, K. H. Johansson, and D. V. Dimarogonas, “Coordinating Truck Platooning by Clustering Pairwise Fuel-Optimal Plans,” in *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*, Las Palmas de Gran Canaria, Spain: IEEE, Sep. 2015, pp. 408–415. DOI: 10.1109/ITSC.2015.75.

[7] C. Lei, E. M. van Eenennaam, W. Klein Wolterink, J. Ploeg, G. Karagiannis, and G. Heijenk, "Evaluation of CACC string stability using SUMO, Simulink, and OMNeT++," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, pp. 1–12, Mar. 2012. DOI: 10.1186/1687-1499-2012-116.

[8] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *6th IEEE Vehicular Networking Conference (VNC 2014)*, Paderborn, Germany: IEEE, Dec. 2014, pp. 53–60. DOI: 10.1109/VNC.2014.7013309.

[9] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A Survey on Platoon-Based Vehicular Cyber-Physical Systems," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 263–284, First quarter 2016. DOI: 10.1109/COMST.2015.2410831.

[10] T. Robinson, E. Chan, and E. Coelingh, "Operating Platoons On Public Motorways: An Introduction To The SARTRE Platooning Programme," in *17th World Congress on Intelligent Transport Systems (ITS 2010)*, Busan, South Korea, Oct. 2010.

[11] S. E. Shladover, C. A. Desoer, J. K. Hedrick, M. Tomizuka, J. Walrand, W.-B. Zhang, D. H. McMahon, H. Peng, S. Sheikholeslam, and N. McKeown, "Automated Vehicle Control Developments in the PATH Program," *IEEE Transactions on Vehicular Technology (TVT)*, vol. 40, no. 1, pp. 114–130, Feb. 1991. DOI: 10.1109/25.69979.

[12] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular Platooning in an Adversarial Environment," in *10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS 2015)*, Singapore, Singapore: ACM, Apr. 2015, pp. 167–178. DOI: 10.1145/2714576.2714619.

[13] J. Gong, Y. Xu, C. Lu, and G. Xiong, "Decision-making model of overtaking behavior for automated driving on freeways," in *IEEE International Conference on Vehicular Electronics and Safety (ICVES 2016)*, Beijing, China: IEEE, Jul. 2016, pp. 7–12. DOI: 10.1109/ICVES.2016.7548162.

[14] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Lane-Change Fuzzy Control in Autonomous Vehicles for the Overtaking Maneuver," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 9, no. 3, pp. 438–450, Sep. 2008. DOI: 10.1109/TITS.2008.922880.

[15] C.-J. Hoel, K. Wolff, and L. Laine, "Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning," in *21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018)*, Maui, HI: IEEE, Nov. 2018, pp. 2148–2155. DOI: 10.1109/ITSC.2018.8569568.

[16] J. Nilsson and J. Sjöberg, "Strategic decision making for automated driving on two-lane, one way roads using model predictive control," in *IEEE Intelligent Vehicles Symposium (IV 2013)*, Gold Coast, Australia: IEEE, Jun. 2013, pp. 1253–1258. DOI: 10.1109/IVS.2013.6629638.

[17] S. Lam and J. Katupitiya, "Cooperative Autonomous Platoon Maneuvers on Highways," in *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2013)*, Wollogong, Australia: IEEE, Jul. 2013, pp. 1152–1157. DOI: 10.1109/AIM.2013.6584249.

[18] A. Groza, B. Iancu, and A. Marginean, "A multi-agent approach towards cooperative overtaking in vehicular networks," in *4th International Conference on Web Intelligence, Mining and Semantics (WIMS 2014)*, Thessaloníki, Greece: ACM, Jun. 2014, pp. 1–6. DOI: 10.1145/2611040.2611096.

[19] H. C.-H. Hsu and A. Liu, "Kinematic Design for Platoon-Lane-Change Maneuvers," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 9, no. 1, pp. 185–190, Mar. 2008. DOI: 10.1109/TITS.2007.908721.

[20] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands: IEEE, Oct. 2013, pp. 2063–2067. DOI: 10.1109/ITSC.2013.6728533.

[21] ——, "Situation Assessment in Tactical Lane Change Behavior Planning for Automated Vehicles," in *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*, Las Palmas de Gran Canaria, Spain: IEEE, Sep. 2015, pp. 975–981. DOI: 10.1109/ITSC.2015.163.

[22] S. Samiee, S. Azadi, R. Kazemi, and A. Eichberger, "Towards a Decision-Making Algorithm for Automatic Lane Change Manoeuvre Considering Traffic Dynamics," *Promet - Traffic&Transportation*, vol. 28, no. 2, pp. 91–103, Apr. 2016. DOI: 10.7307/ptt.v28i2.1811.

[23] R. Schubert, K. Schulze, and G. Wanielik, "Situation Assessment for Automatic Lane-Change Maneuvers," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 11, no. 3, pp. 607–616, Sep. 2010. DOI: 10.1109/TITS.2010.2049353.

[24] S. Noh and K. An, "Risk assessment for automatic lane change maneuvers on highways," in *IEEE International Conference on Robotics and Automation (ICRA 2017)*, Singapore, Singapore: IEEE, May 2017, pp. 247–254. DOI: 10.1109/ICRA.2017.7989031.

[25] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic Traffic Simulation using SUMO," in *21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018)*, Maui, HI: IEEE, Nov. 2018, pp. 2575–2582. DOI: 10.1109/ITSC.2018.8569938.

[26] M. Semrau and J. Erdmann, "Simulation framework for testing ADAS in Chinese traffic situations," in *SUMO User Conference 2016 (SUMO 2016)*, Berlin, Germany: DLR, May 2016, pp. 103–115.

[27] E. Donges, "A Conceptual Framework for Active Safety in Road Traffic," *Taylor & Francis Vehicle System Dynamics*, vol. 32, no. 2-3, pp. 113–128, 1999. DOI: 10.1076/vesd.32.2.113.2089.

[28] Z. Kim, "Robust Lane Detection and Tracking in Challenging Scenarios," *IEEE Transactions on Intelligent Transportation Systems (TITS)*, vol. 9, no. 1, pp. 16–26, Mar. 2008. DOI: 10.1109/TITS.2007.908582.

[29] S. Ulbrich and M. Maurer, "Towards Tactical Lane Change Behavior Planning for Automated Vehicles," in *18th IEEE International Conference on Intelligent Transportation Systems (ITSC 2015)*, Las Palmas de Gran Canaria, Spain: IEEE, Sep. 2015, pp. 989–995. DOI: 10.1109/ITSC.2015.165.

[30] S. Krauß, "Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics," PhD Thesis, Mathematical Institute, Köln, Germany, Apr. 1998.

[31] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of Integrated Longitudinal and Lateral Control for the Operation of Automated Vehicles in Platoons," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 4, pp. 695–708, Jul. 2000. DOI: 10.1109/87.852914.

[32] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and Experimental Evaluation of Cooperative Adaptive Cruise Control," in *14th IEEE International Conference on Intelligent Transportation Systems (ITSC 2011)*, Washington, D.C.: IEEE, Oct. 2011, pp. 260–265. DOI: 10.1109/ITSC.2011.6082981.

[33] H. Woo, Y. Ji, Y. Tamura, Y. Kuroda, T. Sugano, Y. Yamamoto, A. Yamashita, and H. Asama, "Advanced Adaptive Cruise Control Based on Collision Risk Assessment," in *21st IEEE International Conference on Intelligent Transportation Systems (ITSC 2018)*, Maui, HI: IEEE, Nov. 2018, pp. 939–944. DOI: 10.1109/ITSC.2018.8569759.

[34] V. Deeying, K. Kawila, K. N. Nakorn, and K. Rojviboonchai, “A study of vehicular desynchronization for platooning application,” in *17th International Conference on Communication Technology (ICCT 2017)*, Chengdu, China: IEEE, Oct. 2017, pp. 1482–1486. DOI: 10.1109/ICCT.2017.8359878.