This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

Characterization of Transient Communication Outages into States to Enable Autonomous Fault Tolerance in Vehicle Platooning

Shahriar Hasan, Svetlana Girs, and Elisabeth Uhlemann

The benefits of platooning, e.g., fuel efficiency, road throughput enhancement, driver offload, etc., have sparked an interest in a more connected, intelligent, and sustainable transportation ecosystem. However, efficient platooning is realized through wireless communications, characterized by transient connectivity, which is caused by occasional packet losses. Being a safety-critical system of systems, a platoon must be fail-operational even during transient connectivity. Moreover, a platoon should be capable of transitioning into a fail-safe state upon encountering a hazard. To this end, we propose a strategy for classifying the transient communication outages incurred by platooning vehicles into states. Furthermore, a state machine using these states to enable safe automated platooning is proposed that also defines the transitions between the states based on the nature and levels of transient connectivity and hazards. To achieve this, a graceful degradation and upgradation method is proposed, such that the platoon can remain fail-operational by adjusting, e.g., the automated controller and/or the inter-vehicle gaps based on the current communication quality. An emergency braking strategy is also proposed to enable a fast transition into a fail-safe state, should the platoon encounter a hazard. Rigorous simulation studies show that the proposed strategies enable fault-tolerant automated platooning also during transient connectivity.

***Index Terms*—ACC, CACC, cooperative driving, connected vehicles, collision avoidance, emergency braking, fail-operational, fail-safe, fault tolerance, platoon, Plexe, SUMO, Veins, V2V.**

I. Introduction

A group of highly automated and connected vehicles forms a platoon by autonomously following a Lead Vehicle (LV) and maintaining short inter-vehicle distances by means of wireless vehicular communications and onboard sensors. Vehicle-to-Vehicle (V2V) communication is a key enabling technology in platooning, and it is tightly coupled with vehicle dynamics, control, and computing technologies [1]. However, the wireless communication quality typically varies, causing transient errors that may significantly affect the platooning operations, e.g., joining, merging, splitting, maintaining, braking, etc. The challenges get further aggravated by the short response times, which are implicit when fulfilling the requirement to sustain short inter-vehicle gaps since this directly regulates the degree to which the platooning benefits in terms of fuel efficiency can be attained [2].

Since the consequences of failures in automated platooning potentially can endanger human life and lead to damage to equipment or the environment, a platoon can be considered a safety-critical system of systems [3]. Any safety-critical system should be fault-tolerant, e.g., include fail-operational and/or fail-safe states, to mitigate the effects of failures [2]. *Fail-operational* in this context implies that a platoon should provide certain critical functionalities and remain at least as safe as it was before the temporary communication outage occurred, i.e., a nominal performance in terms of safety should always be ensured [4]. In order to facilitate the fail-operational state, the platooning vehicles should gracefully degrade their performance in terms of, e.g., fuel efficiency, during runtime in a way that is proportionally related to the level of transient communication errors [5]. Graceful degradation is of the essence here because communication errors are usually transient, and declaring one or more communication links as failed can be premature [6]. In platooning, performance degradation implies increasing the inter-vehicle gaps and/or switching to a more suitable *controller* that regulates the consensual speed and desired gap between the platooning vehicles in a different way, which is better given the currently experienced communication quality.

Another important component of fault tolerance is a *fail-safe* state, which becomes crucial in platooning applications when a platoon encounters an irrecoverable failure or a hazard. The fail-safe design principle widely used in aviation safety states that "an inspection method must easily detect a hazard or failure during runtime, and the system must sustain the hazard for an adequate time before safety is compromised" [7]. A hazard in platooning can be caused by scenarios such as the sudden appearance of animals or debris, a stalled vehicle on the highway, abrupt emergency braking by a vehicle or platoon in front, road closure due to accidents or weather conditions, etc. In these scenarios, simply steering away by changing lanes is often not an option as the visibility or road monitoring capabilities of the Following Vehicles (FVs) in the platoon typically are obstructed by the LV [8]; hence, autonomous emergency braking is of the essence here. To attain the fail-safe design principles in the context of platooning, the hazard must be detected in time, and the platoon must perform emergency braking sufficiently fast such that the stopping distance of the LV is short enough to avoid the hazard while collisions within the platoon are avoided. Consequently, the aim is to ensure a fail-safe state such that, in the event of a hazard or a failure, the platoon responds in a way that will cause minimal or no harm to other equipment, the environment, or people.

Most previous work on platooning addresses performance degradation in case of transient connectivity and emergency braking due to a hazard as two separate problems [9]–[13].

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska - Curie grant agreement No 764951.

The authors are with the School of Innovation, Design and Engineering, Mälardalen University, 722 20 Västerås, Sweden (e-mail: {shahriar.hasan, svetlana.girs, elisabeth.uhlemann}@mdu.se).

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

However, these two events are tightly coupled. For instance, when a hazard is encountered, a platoon might be in any degraded state due to previously encountered communication errors. Therefore, a platoon must be capable of performing emergency braking to reach a fail-safe state at all degraded modes and all wireless connectivity conditions. As stated previously, to perform emergency braking, the platooning vehicles are required to brake hard, avoid collisions, and the LV needs to minimize its stopping distance to circumvent the hazard that triggered the emergency braking. However, in most recent works, only collision avoidance within the platoon is regarded as emergency avoidance or considered to be fail-safe [14], [15]. Sustaining large gaps after a full stop has also been emphasized in some recent works [14], [16], [17]. However, it is not beneficial to maintain large inter-vehicle gaps when in a fail-safe state if it takes longer time to reach it or if it causes the LV to traverse longer. In [10], [13], [18], and [19], the authors focus on precisely this: minimizing the stopping distance of the platoon. However, until now, reaching the fail-safe state from any degraded but fail-operational state due to previously encountered problems with the wireless connectivity has not been considered. In addition, many recent works propose control approaches for switching communication topologies, assuming that wireless connectivity is either present or not, e.g., [20]–[23]. However, this is an oversimplification as wireless communication outages are transient, and the communication quality fluctuates due to occasional channel access delays, packet drops, fading, path loss, etc.

The main contribution of this paper is twofold: first and foremost, the classification of transient communication outages into different states enabling the *good*, *fair*, and *poor* communication thresholds. Furthermore, we propose a state machine for automated platooning that captures these various degraded communication states. The state machine also includes emergency braking as a function of the experienced communication quality levels. The level of instantaneous communication quality regulates the autonomous switching between different platooning modes in the state machine. Secondly, we conduct a literature review to analyze, categorize and assign other relevant studies to the different states of the proposed state machine. In order to enable fault tolerance in automated platooning, a *Graceful Degradation and Upgradation (GDU)* method is proposed that keeps the automated platoon fail-operational or fail-safe by continuously monitoring the presence of hazards and the current communication quality during runtime and autonomously switching between the states. Note that the platoon vehicles can change states in the state machine based on the individually experienced communication quality. This enables both that we can aim at being as fuel-efficient as the communication quality currently allows and that the state machine works for both homogeneous and heterogeneous platoon vehicles. Finally, we provide a general framework for evaluating different types of automated emergency braking strategies based on the instantaneous communication quality and the source of information needed for and available on braking. Using this framework and the state machine, the *Enhanced Synchronized Braking (ESB)* strategy is proposed as a fail-safe measure which can autonomously adjust such that it can perform emergency braking for all levels of available communication quality. The ESB strategy focuses on avoiding collisions between the platooning vehicles, minimizing the stopping distance of the LV, and transitioning the whole platoon into a fail-safe state fast by enabling as high deceleration rate as the communication quality allows.

To the best of our knowledge, our work is the first of its kind that proposes to classify transient communication outages into different levels instead of simply declaring communication as either present or absent between two vehicles and introduces the idea of heterogeneous controllers in a platoon. The fine-grained characterization of communication quality allows us to decentralize platoon control and keep the platooning vehicles fault-tolerant by assigning different controllers and/or gaps as a function of the experienced communication quality levels. Related works such as [21], [22], and [23] propose to switch between different communication topologies but not switching controllers. Further, the topology is changed based on whether the communication is absent or present between two vehicles.

We have conducted rigorous simulation studies to evaluate the state-of-the-art control algorithms that are used as the states in the GDU method, i.e., the GDU method governs the switching between these controllers based on the experienced communication quality. The evaluation of the controllers is carried out in terms of safety, fuel efficiency, string stability, and LV tracking ability. Furthermore, we evaluate the proposed GDU method under the same simulation scenarios and criteria to understand the benefits of classifying wireless connectivity into *good*, *fair*, and *poor* qualities and performing switching between different controllers and/or adjusting inter-vehicle gaps to keep a platoon fault-tolerant. In addition, the proposed GDU method and the ESB strategy are evaluated in terms of their fail-operational and fail-safe conditions under challenging scenarios, e.g., time-varying communication delays, short inter-vehicle gaps, high speed, and strong deceleration. Finally, based on the obtained simulation results, we define a set of safety contracts that captures the component behavior of the system given the input conditions such as active controller, experienced communication quality, deceleration rate, etc.

The rest of the paper is structured as follows: Section II reviews related works on platooning and details the state-of-the-art controller properties, whereas the description of the proposed state machine is presented in Section III. In Sections IV and V, the state machine is split into two main parts, i.e., platoon cruising, including fail-operational states in one part and emergency braking and fail-safe states in the other part, with the relevant studies from the literature attributed to the different states. In Section VI, the simulation scenario, traffic model, and metrics used to evaluate the proposed approaches are described. Next, the evaluation results of fail-operational and fail-safe automated platooning in the light of the proposed state machine are presented first separately in Sections VII and VIII respectively, and then together in Section IX. Based on the proposed GDU method and the evaluation results, some safety contracts are suggested in Section X that capture the operation modes of the system components. Finally, Section XII concludes the paper.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

II. Background and Related Works

This section describes the state-of-the-art works on fault tolerance in platooning and details the properties of different types of controllers for vehicle strings and automated platooning suggested to be used in different states of the state machine proposed in this paper.

Most modern vehicles are already equipped with Adaptive Cruise Control (ACC) that enables a vehicle to maintain the desired speed or if that is not possible, follow the preceding vehicle by adjusting to its relative speed and distance measured by radar or lidar sensors. However, a vehicle string in which each vehicle uses an ACC controller lacks string stability due to the engine lag, sensor detection, processing, and actuation delay propagated downstream [24]. The ability to maintain *string stability* is a property of the controller that attenuates the spacing errors as they propagate from the head to the tail of a vehicle string [25]. The efficacy of a controller that regulates a vehicle string is usually assessed by its ability to maintain string stability, use short gaps, and avoid inter-vehicle collisions during platooning. Using Cooperative Adaptive Cruise Control (CACC) or applying a so-called PLATOON controller tackles the problem of sensor detection and processing delays by adding V2V communications to an existing ACC. In contrast to most previous works, Shladover *et al.* suggest using the terms CACC and PLATOON distinctively in [24]. The authors reason that a PLATOON is a closely coupled system of systems in which the vehicles follow a Constant Distance Gap (CDG) policy, offering both lateral and longitudinal controls. On the other hand, a CACC string of vehicles relies on a Constant Time Gap (CTG) policy facilitating longitudinal control only. A *time gap* is the elapsed time from when the preceding vehicle's rear bumper traverses a reference point on the road to when the ego vehicle's front bumper traverses the same point. Following a CTG policy, the vehicles increase the inter-vehicle gaps as a function of speed, whereas with a CDG, the gaps between the vehicles are kept the same despite speed changes. To enable the CDG policy in a PLATOON of vehicles, the FVs require periodic updates from both the LV and the preceding vehicle (*leader-predecessor following strategy*). On the other hand, the vehicles in a CACC string can maintain longitudinal control upon receiving periodic updates from their respective predecessors only, i.e., *predecessor following strategy*. Note that a PLATOON following the CDG policy may enable inter-vehicle gaps as short as 5 meters [26], implying higher fuel efficiency and enhancement of road throughput. However, as all the FVs require periodic updates from the LV and the gaps are short, temporary communication outage is more severe from a safety point of view when using a PLATOON controller [24], especially for the rear vehicles in the platoon which are farthest away from the LV. In the remainder of this paper, the terms CACC and PLATOON are used distinctively to denote controllers for a string of vehicles and a platoon, respectively.

The information required for automated platooning is disseminated via V2V communications, using some type of periodic messages, e.g., Cooperative Awareness Messages (CAMs) [27] that contain necessary parameters for lateral and longitudinal control. In addition, when a hazard or an event of common interest occurs, the LV, an FV, and/or a roadside unit may broadcast event-driven messages, e.g., Decentralized Environmental Notification Messages (DENMs) [28] for the duration of the event, instructing the vehicle string to react, e.g., by performing emergency braking.

Since sensor systems and wireless communications, the key enabling technologies for automated platooning, are never completely error-free [29], fault tolerance mechanisms in platooning have received significant research attention. In [30], a graceful degradation algorithm is proposed that takes the inaccuracies caused by radar sensor failure in a CACC-based system as input, and a safe time headway is chosen dynamically. The results show that a vehicle maintains a longer safe distance upon detecting a radar fault. Yu *et al.* [12] are addressing performance degradation and communication interruption, but only on the string stability and fuel efficiency in a fleet of ten vehicles. Ploeg *et al.* [9] propose graceful degradation of platooning functions by transitioning from CACC to "degraded CACC" (dCACC) mode when a host vehicle experiences communication latency. The criteria for switching between CACC and dCACC based on the experienced communication delay are also provided in [9]. However, exactly how this transition and graceful degradation should be performed in case of irregular packet losses rather than a slowly varying communication delay is directed toward future investigations. Kaiser *et al.* propose a canonical approach to design degradation cascades in automated systems [31]. A state machine is also proposed to demonstrate how a degradation cascade can be used in the event of failures. Sljivo *et al.* also propose a degradation cascade capturing various failure modes in vehicle platooning and derive a set of safety contracts based on that [20]. However, in both [31] and [20], the authors do not consider transient connectivity errors of varying levels during cruising or emergency braking.

Nunen *et al.* propose fault-tolerant and fail-safe mechanisms following the V-model of the system development process in [32]. A set of safety measures is defined, based on which a platooning vehicle chooses between the safe states, e.g., fault-tolerant, fail-safe, and nominal CACC states. The state transitions are defined by brake threat number, probability of the lead vehicle braking, and communication latency duration. In the state machine proposed in this paper, transitions between states are made based on *good*, *fair*, or *poor* communication quality, together with received instructions and/or encountering road hazards. In our view, the proposed state machine provides a more holistic framework for automated platooning, taking into account transient communication errors as well as fail-operational and fail-safe measures, which this paper aims to prove.

In [33], Segata *et al.* propose a state machine in which the states are represented by different CACC and ACC controllers. The authors propose that the vehicles have multiple communication interfaces such as the IEEE 802.11p [34], Visible Light Communications (VLC), and Long-Term Evolution (LTE). The switching between different states happens due to the failure or recovery of a communication interface. In [35], Yu *et al.* study the effects of switching between PLATOON and ACC controllers on string stability, energy consumption, and carbon

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

emission. The authors introduce communication interruptions in their simulations to evaluate the controller switching. In the state machine presented in this paper, we consider the instantaneous communication quality of the vehicles, and the switching between different states happen by monitoring the communication quality, i.e., *good*, *fair*, or *poor*, with the LV and the vehicle in front during runtime. Moreover, the works in [33] and [35] do not analyze the tradeoffs between safety, fuel efficiency, string stability, and LV tracking ability during time-varying communication quality. The transient presence of required information needed from adjacent vehicles, regardless of the underlying communication interface, can be classified into states and used for safe autonomous platooning, given our framework.

Several studies in the literature address how switching between communication topologies affects platoon stability. Li *et al.* in [21] propose finite-time control protocols under fixed and switching communication topologies to achieve platoon stability and consensus. Their numerical experiment results show that spacing and velocity errors converge to zero in finite time. In [22], the authors propose a distributed model predictive control (DMPC) approach that is applicable for a platoon with switching communication topology. They conduct numerical simulations in which the platooning vehicles switch between communication topologies, e.g., Predecessor Following (PF), leader predecessor following, PF-failure, etc. Simulation results show that the position and velocity errors of the FVs approach zero asymptotically despite the communication failure in the PF-failure topology. However, string stability is not guaranteed, and tracking error exists when external disturbances are introduced in the simulations. In [23], Gao *et al.* present a distributed control approach that also considers switching communication topology, heterogeneity in vehicle dynamics, and external disturbances. Simulation results show the robustness of the proposed distributed controller in terms of distance and velocity errors. However, the possibility of communication failure is expressed as a function of distance only, and other important factors, e.g., channel access delays, bit errors, interference, etc., are not considered. Chehardoli and Homaeinezhad also study which effects the switching communication topologies have on platoon stability under time-varying communication delays [36]. The fundamental difference between the work presented in this paper and in [21]–[23] is that we consider the transient nature of wireless connectivity outages and divide communication quality into various levels, i.e., *good*, *fair*, *poor*, instead of classifying the communication as either successful or failed. In addition, in our proposed state machine, the platooning vehicles switch communication topology at the same time as they switch gap policy (CDG or CTG) as dictated by the control algorithm. The aim is to keep the platoon fail-operational in terms of safety by degrading the performance in terms of fuel efficiency and string stability when required.

As mentioned above, V2V communication is crucial for emergency braking of platooning vehicles traveling with short inter-vehicle gaps. Alkim *et al.* showed in their simulation studies that if the FVs in a platoon are V2V-enabled, they could respond to the hazard much faster [37]. However, this study does not consider the communication latency incurred by the neighboring vehicles. In our previous work [19], we showed that this communication latency must be accounted for to avoid collisions within the platoon during emergency braking in a dense vehicle scenario that induces both high levels of data and road traffic. Murthy and Masrur propose leveraging the space buffer between vehicles in a heterogeneous platoon during emergency braking on a flat road [10] or in a downhill [18]. To this end, the authors propose that if a platooning vehicle cannot brake at its assigned deceleration rate on a downhill road, it sends distress messages so that the other vehicles can adapt their deceleration rates to the one under distress. However, safe braking is not guaranteed if the number of distress messages lost is greater than a threshold. Moreover, if the number of CAMs lost exceeds a threshold, the authors propose to dissolve a platoon. In contrast, instead of dissolving the platoon, we propose to switch between controllers and/or adjust inter-vehicle gaps proportionally with the levels of communication outages in this paper. The works in [19], [10], and [18] emphasize minimizing the stopping distance of the LV in addition to avoiding collisions between the FVs to attain a fail-safe state. However, to the best of our knowledge, the transition from fail-operational states to emergency braking states leading to a fail-safe state is not considered in previous works.

The vast majority of the works in the literature either focus on degradation algorithms to maintain certain platoon functionalities even in the presence of transient errors or they focus on emergency braking strategies. However, a detailed picture of both fail-operational and fail-safe algorithms, together with their inter-dependencies under varying communication errors and delays, is missing in the literature, which is why this aspect is addressed here.

III. State Machine for Automated Platooning

In Figure 1, we propose a state machine that demonstrates how to transition between platoon forming, cruising, emergency braking, and dissolving to tackle the challenges imposed by transient communication errors of different lengths also when coupled with the requirement to enable emergency braking in case of a hazard. An initial concept of the state machine was first proposed in a technical report by the authors [38]. In this section, the states in Figure 1 are first defined, and then the state transitions are explained. We formulate three research questions based on the proposed state machine and address them in the remainder of this paper.

A. State Definitions

The state machine is divided into *Platoon forming*, *Cruising*, *Emergency Braking*, *Fail-Safe*, and *Dissolve platoon* states. In addition to the states mentioned here, there can also be other platooning states, such as platoon joining, merging, cut-in, cut-out, etc., under the same communication constraints as in Figure 1. However, since this paper focuses on the fail-operational and fail-safe states caused by hazards and communication errors, we do not consider these general cruising scenarios separately. Nevertheless, we note that a cut-in scenario,

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

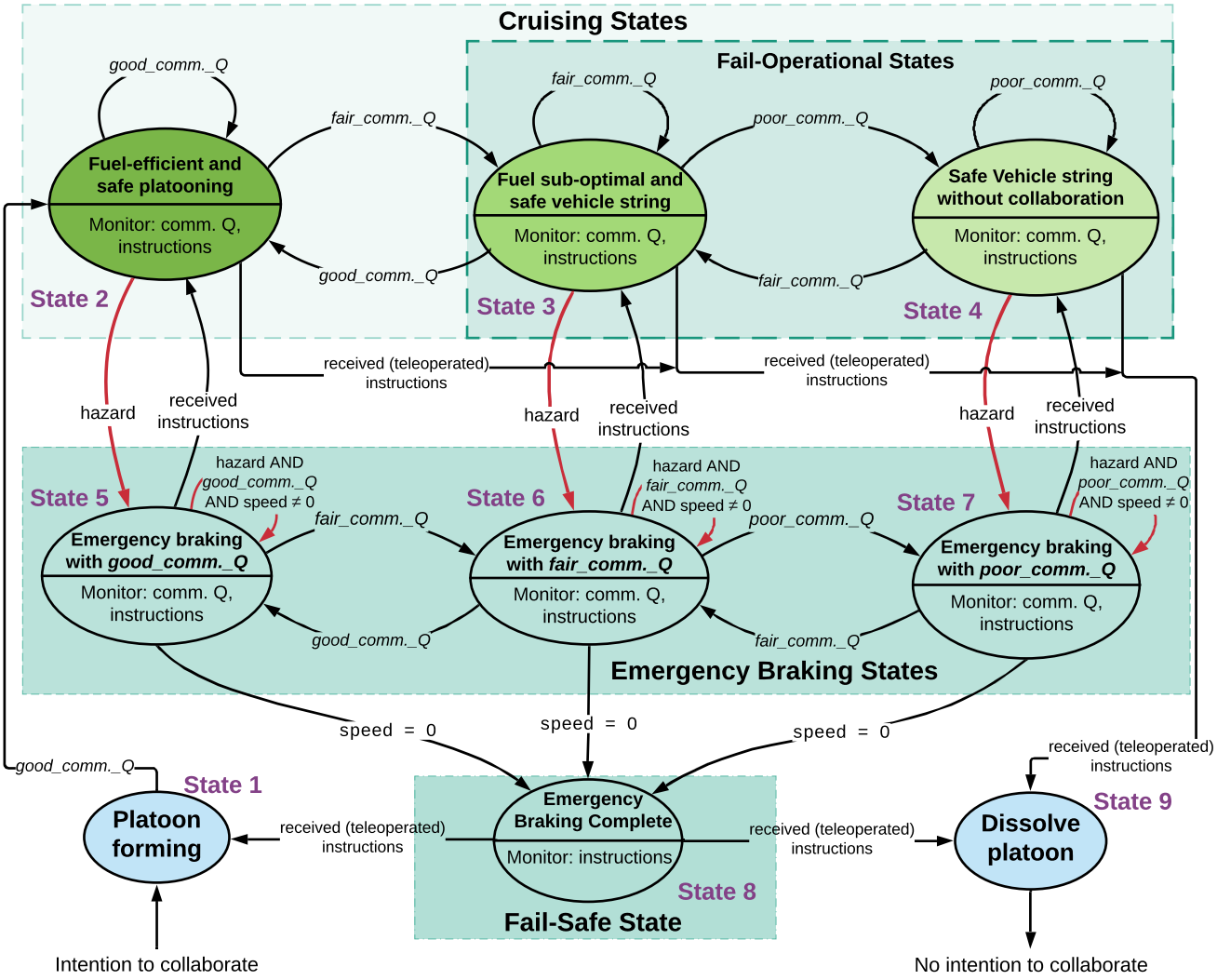


Fig. 1: State machine representing different states during platooning operation and the transitions between the states due to transient communication errors or road hazards.

e.g., a non-platooning vehicle changes lanes to place itself between the platooning vehicles, can be considered a platoon-related hazard, potentially leading to strong deceleration while cruising at high speed.

Platoon Forming: This is the starting state in which a platoon is formed when there exists an intention to collaborate. Instructions regarding platoon formation, such as route planning, platoon size, inter-vehicle distances, speed, etc., are given from, e.g., a fleet operating control center through Vehicle-to-Everything (V2X) communications.

Cruising States: In the Cruising States, the platooning vehicles cruise obeying a control law with a given speed and inter-vehicle gaps dictated by the controller and communication quality. The cruising states are subdivided into *fuel-efficient* and *fail-operational* states.

- *Fuel-Efficient State: State 2* in Figure 1 represents the fuel-efficient state. In this state, the platooning vehicles maintain short inter-vehicle gaps following the CDG policy to enable fuel efficiency by reducing the aerodynamic drag. Moreover, due to the *good* communication quality in *State 2*, string stability can be maintained while still providing the required level of safety. In addition, the communication quality and the presence of potential road hazards or external instructions are monitored.
- *Fail-Operational States: States 3* and *4* in Figure 1 comprise the fail-operational states. In the fail-operational states, a vehicle maintains platooning functionalities with *at least the same nominal performance in terms of safety* during a transient communication outage. To facilitate the nominal safety level, the vehicle can exhibit *lower than nominal performance, i.e., degraded performance in terms of e.g., string stability or fuel efficiency* for the duration of the communication outage.

  – *State 3*: In this state, performance is degraded in terms of fuel efficiency by increasing the inter-vehicle gaps and/or performing controller switching due to experiencing a deteriorated communication quality. If the controller is exchanged to CACC, the CTG policy is adopted instead of the CDG. Obviously, a sufficient level of safety and string stability is still targeted, but at the expense of fuel efficiency by changing the communication topology and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

distance policy and/or increasing the inter-vehicle gaps.
- *State 4*: Due to the *poor* communication quality in *State 4*, the vehicle string no more relies on V2V communications and collaboration. Instead, a radar or lidar-based controller such as the ACC is adopted, which requires longer inter-vehicle gaps and adoption of the CTG policy. Safety takes precedence over fuel efficiency and string stability in this state.

Emergency Braking States: A platoon transitions into one of the emergency braking states from the cruising states upon receiving instructions from the ego vehicle using local sensors, from another vehicle through V2V communications, or remotely through V2X communications to initiate an emergency braking since a hazard has been detected. It is clear that when a hazard is encountered, the communication quality can be either *good*, *fair*, or *poor*. The platooning vehicles monitor the instructions from the LV or Adjacent Vehicles (AVs) and their distance to the preceding vehicle to adjust their deceleration rates. Note that fuel efficiency and string stability are of no concern during emergency braking.

- *State 5* represents emergency braking originated from *state 2*, which is less challenging despite short inter-vehicle gaps due to the *good* communication quality.
- *State 6* represents emergency braking in the presence of transient communication errors. Given an appropriate braking strategy, emergency braking with *fair* communication quality can still be done safely and quite effectively as the platooning vehicles have longer inter-vehicle gaps in their originating state, *State 3*.
- *State 7* represents emergency braking with *poor* communication quality. As this state originates from *state 4*, the vehicles have even longer gaps. It should be noted that the vehicles may still be able to perform communication-assisted braking despite using a control law such as ACC since the communication is not lost permanently and instructions from the other platooning vehicles are monitored continuously.

Fail-Safe State: State 8 in Figure 1 represents the *Fail-Safe* state. The platooning vehicles have come to a complete standstill in this state by performing emergency braking. All platooning vehicles must avoid collisions to satisfy the conditions of a fail-safe state, e.g., no harm done to people, environment, or equipment. In addition, the lead vehicle is required to traverse a sufficiently short distance to avoid the hazard that caused the emergency braking [10], [13], [19]. The inter-vehicle gap at a complete standstill is of no concern in this scenario, and the communication quality does not need to be monitored. The conditions of a fail-safe state are formulated in more detail in Section V. From here, the platooning vehicles await instructions on whether to reform the platoon or dissolve it.

Dissolve Platoon: The platooning vehicles may have reached the *Dissolve platoon* state, *State 9* in Figure 1, either by doing an emergency braking using *State 5, 6*, or *7* and then transferring to *State 8* as soon as the speed is zero, or by simply having received instructions to stop collaborating when in one of the *Cruising* states, *State 2*, *3*, or *4*. The inter-vehicle distance during cruising may or may not have been retained when the platoon dissolves.

B. State Transitions

Before we describe the state transitions, it is important to note that different platooning vehicles can experience different levels of communication quality, e.g., *good*, *fair*, or *poor*, at a particular time instance. Especially, the tail vehicles in the platoon experience more packet losses compared to the vehicles in the front when communicating with the LV due to path loss and shadowing effects [19]. When a vehicle is increasing its gap to the vehicle in front, it can in turn deteriorate the communication quality experienced by the other FVs even further. Since all the platooning vehicles adjust the gap to the vehicle in front and/or perform controller switching in a distributed way, it is possible that, for instance, the second vehicle in the platoon is in *State 2*, while the last vehicle is in *State 4* due to experiencing *poor* communication quality. The state machine proposed in Figure 1 works both in cases where all the platoon members do and where they do not experience the same communication quality. However, for *States 1, 8* and *9*, the platoon acts as one entity rather than a system of collaborating autonomous systems. This is because the platoon vehicles cannot transition to *State 2* until they have agreed to collaborate, are connected by V2V communications, and have formed the platoon in *State 1*. Similarly, the fail-safe state is not reached until all platoon vehicles have stopped. Nor can the platoon be said to have dissolved until all the vehicles receive instructions to stop collaborating.

Once the decision to form a new platoon is made (*State 1*) and *good* communication quality can be established between all vehicles, the platoon can adjust the vehicle gaps according to CDG and transition to the *fuel-efficient and safe platooning* state (*State 2*).

The transitions between the *Cruising States*, i.e., *States 2, 3,* and *4* are regulated by the communication qualities perceived by each vehicle. We divide the communication quality into three levels, i.e., *good*, *fair*, and *poor*. When the communication quality is *good*, *fuel-efficient and safe* platooning is enabled, *State 2*. The inter-vehicle distances are short in *State 2*, and the communication quality is monitored periodically. If the communication quality deteriorates to *fair*, the platoon vehicle transitions into the *fuel sub-optimal, string stable and safe* state, *State 3*. From this state, the vehicle can switch back to *State 2* once the communication quality becomes *good* again (performance upgradation). However, if the communication quality further worsens to *poor*, the vehicle adopts a radar or lidar-based controller, e.g., ACC, to maintain safe but minimal platooning functionalities, *State 4* (performance degradation). In this state, the inter-vehicle gaps are further increased according to the CTG approach to ensure safety. Neither fuel efficiency nor string stability is the primary goal at this stage; instead, the vehicle monitors the V2V communication links to see if *fair* communication can be reestablished.

The transitions to the emergency braking states, *States 5, 6*, or *7* occur when a hazard is encountered. The transitions in-between *States 5, 6*, and *7* are regulated by the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

presence of a hazard and the experienced communication quality. When a hazard is detected in *State 2*, the CDG is short, and the communication quality is *good*. However, if the communication quality becomes *fair* shortly after, the platooning vehicle transitions from *State 5* to *State 6*. In such a scenario, it is required to perform emergency braking with *fair* communication quality, but the vehicle may still have a short gap which was inherited from *State 2*. However, note that in general, short gaps imply better communication quality as the path loss is lower. Hence, it is crucial that the communication-assisted automated emergency braking strategy is tailored to the communication quality experienced and the inter-vehicle gaps used, taking into account that the braking causes an increased communication load as several vehicles may broadcast hazard warnings, but also that a shorter range usually improves the communication quality. Finally, we note that if instructions are received that emergency braking is no longer needed, the platoon can switch back to *State 2*, *3*, or *4* depending on the instantaneous communication quality. From the *Emergency Braking States*, a platoon is said to have transitioned into the *Fail-Safe* state if there are no collisions and all vehicles in the platoon are at a standstill (zero speed). Clearly, inter-vehicle collisions must also be avoided if and when decelerating in *States 2, 3,* and *4* or if stopping due to dissolving the platoon in *State 9*, but in these cases, it is not necessary to minimize the stopping distance of the LV.

A platoon can transition from the *Fail-Safe* state (*State 8*) to the *Dissolve platoon* state (*State 9*) if instructions to stop collaborating have been received. Similarly, a platoon can be formed again (*State 1*) if instructions to do so are received, in which case the communication quality will be monitored again. Note that a platoon can also transition to the *Dissolve platoon* state from the *Cruising States* when receiving instructions that collaboration is no longer desired. Finally, the platooning vehicles can transition to *State 1* from the *Fail-Safe* or from the *Dissolve platoon* states, e.g., upon receiving teleoperated instructions from a control center.

Based on the discussion above, it is clear that all the platooning vehicles need to agree on the state machine, such as the one in Figure 1, and certain parameters while collaborating. For instance, once the decision of collaboration is made, the vehicles must agree on the *good*, *fair*, and *poor* communication thresholds, constant distance gaps, constant time gaps, and the factor by which the inter-vehicle gaps are to be adjusted in case of *fair* communication quality in order to form a platoon. Moreover, vehicle kinematic parameters, such as speed, position, steering angle, and acceleration, are required to be communicated. Note that the platoon does not need to be homogeneous, as, e.g., different deceleration capabilities can be handled given that different platoon vehicles can be in different states - but the prerequisites need to be known, and the instantaneous conditions are required to be communicated in order to select values of different parameters which are safe enough. In the emergency braking states, all the platooning vehicles must be aware of the nature, severity, and distance to the hazard, as well as the detection time of the hazard. In addition, if the LV detects the hazard, it must also inform the other platoon members when and how to perform the braking maneuver and the deceleration rates to be pursued. In other words, given that the weight, length, inter-vehicle gaps, and braking capacities of the individual platooning vehicles are known and agreed upon, the proposed state machine is applicable for both homogeneous and heterogeneous platoons.

The following Research Questions (RQs) can be derived based on the state machine above:

- RQ1: How can automated platooning be maintained in the presence of transient communication errors, i.e., *fail-operational*, and with the aim of using short inter-vehicle distances and high vehicle speed while assuring safety?
- RQ2: How does the mapping between the duration of transient communication outages and the *good*, *fair*, or *poor* thresholds which dictate the autonomous transition between the states impact platoon safety, fuel efficiency, string stability, and lead vehicle tracking?
- RQ3: In case of emergencies caused by road hazards, how should the platoon coordinate to perform its emergency braking maneuver to transition into a *fail-safe* state fast given different qualities experienced on the communication links and different inter-vehicle gaps?

IV. Cruising States

This section describes the cruising states, including the two fail-operational states. First, the conditions for attaining the fail-operational states are given. Then the control algorithms available in the literature are described, and appropriate control laws are attributed to the cruising states, i.e., *States 2, 3,* and *4* in Figure 1. Finally, the proposed GDU method is presented, which facilitates autonomous controller switching based on the experienced communication quality to the LV and/or the vehicle in front.

A. Fail-Operational Conditions

The most obvious way for a platoon to be fail-operational in case of transient communication errors is to increase the inter-vehicle gaps. However, the fuel efficiency constraint should still be considered if possible, and the inter-vehicle gap and communication quality must match the requirements of the control law under consideration. To this end, we define the following conditions of a fail-operational state:

1. Regardless of the control law being used, we consider the safety condition in *States 2, 3,* and *4* to be that the measured inter-vehicle distance between any two platooning vehicles is greater than 0 meters for all conditions, i.e., no collisions.
2. String stability takes precedence over fuel efficiency. This is to ensure that individual vehicles do not prioritize their own fuel efficiency over the platoon fuel efficiency. Similarly, safety takes precedence over both string stability and fuel efficiency according to the definition of a fail-operational state. This is to make sure safety to all always comes first for all vehicles.
3. The selected control law and inter-vehicle distance should be adapted to *the instantaneous communication quality of the link to the LV, the vehicle in front, and the immediate FV*. Moreover, the platoon should autonomously close the inter-vehicle gaps and adopt a more fuel-efficient controller when the communication quality improves.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

B. Control Algorithms

According to the ACC controller proposed by Ioannou and Chien in [39], the control law of the i^{th} vehicle can be given by

$$\ddot{x}_{i\_des} = -\frac{1}{T}(\dot{\varepsilon}_i + \lambda\delta_i), \tag{1}$$

$$\delta_i = x_i - x_{i-1} + l_{i-1} + T\dot{x}_i, \tag{2}$$

$$\dot{\varepsilon}_i = \dot{x}_i - \dot{x}_{i-1}, \tag{3}$$

where i and $i-1$ denote the ego and the preceding vehicles, respectively. Further, $\ddot{x}_{i\_des}$ is the desired acceleration of the i^{th} vehicle, λ is a design parameter, T is the time gap, δ_i is the spacing error, i.e., the difference between the measured distance $x_i - x_{i-1} + l_{i-1}$ to the front vehicle and the desired distance $T\dot{x}_i$. Finally, $\dot{\varepsilon}_i$ is the relative speed between the ego and front vehicles. A string of vehicles using the ACC controller requires fairly long time gaps due to the detection, processing, and actuation delay that propagates downstream [40]. For *State 4* in Figure 1, the ACC controller represented by Equation (1) can be used, as the state assumes *poor* communication quality and stipulates a long time gap.

Several CACC controllers are proposed in the literature, all of which are aiming at string stability and minimizing the inter-vehicle gaps. Santini *et al.* propose a consensus controller in which the vehicles' data to be used for computing the actuation of the ego vehicle is determined based on network characteristics during runtime [41]. The most straightforward CACC controller facilitating longitudinal control is likely the one in which the ego vehicle calculates its acceleration using the preceding vehicle's intended acceleration obtained through V2V communications, e.g., the CACC controller proposed by Ploeg *et al.* in [42]. The relative speed and distance data are obtained through the radar sensor as in ACC. The control law of the Ploeg CACC is defined as

$$\dot{u}_i = \frac{1}{T}(-u_i - k_p\delta_i + k_d(-\dot{\varepsilon}_i - T\ddot{x}_i) + u_{i-1}), \tag{4}$$

where u_{i-1} is the intended acceleration of the preceding vehicle that is communicated to the ego vehicle through V2V communications, and k_p and k_d are the controller gains. The Ploeg CACC controller exhibits better fuel efficiency and string stability than the ACC controller by minimizing the inter-vehicle gaps since the FVs can learn their predecessors' intentions even before they actuate. Moreover, the PLOEG CACC controller only requires feedback information from the preceding vehicle and relies on the CTG policy; hence, the control law represented by Equation (4) can be attributed to *State 3* in Figure 1.

Milanés *et al.* propose two control algorithms for addressing the smooth gap-closing maneuver in a cut-out scenario and the car following maneuver in a platoon [43]. Ali *et al.* propose a modification of the CTG policy in which the inter-vehicle gaps are changed based on the speed difference between the ego vehicle and the reference speed of a string of collaborating vehicles [44]. The authors also propose to switch to the classical CTG policy in case of communication loss. The strategies proposed in [43] and [44] can be used in *State 3* in Figure 1, as well as in *State 4* in case a previously agreed speed has been communicated and agreed on before the communication quality deteriorated.

Lee *et al.* propose distinctive controllers for longitudinal and lateral control; a headway controller for longitudinal control and a magnetic sensor-based controller for lateral control [45]. The PLATOON controller proposed by Rajamani also provides both lateral and longitudinal control by using the leader-predecessor following strategy and the CDG policy [25]. Due to the requirement of *good* communication quality with the LV and the immediate predecessor, the PLATOON controller proposed by Lee *et al.* in [45] and by Rajamani in [25] is suitable for *State 2* in Figure 1.

Liu *et al.* define a control law that uses the feedback information from both the LV and the preceding vehicle [46]. In addition, they found that asynchronous actions, e.g., immediate acceleration changes upon reception of a CAM packet, lead to string instability due to the propagation of the error in the platoon's downstream direction. Hence, Liu *et al.* [46] propose that the vehicles delay their actions until all the vehicles have received the CAMs to cancel out the effect of communication latency on string stability. Fernandes and Nunes also show that delayed action can improve string stability in a leader-predecessor following strategy when the FVs in a platoon hold their actuation until having received 'anticipatory information' from the LV [47]. Both [46] and [47] are suitable for *State 2* in Figure 1.

The PLATOON controller by Rajamani in [25] facilitates inter-vehicle gaps as short as 5 m [48], and thereby higher fuel efficiency due to its reliance on the CDG policy and receiving feedback information directly from the LV in addition to the vehicle in front. Using to the PLATOON controller [25], the desired acceleration of the ego vehicle can be given by

$$\begin{aligned}\ddot{x}_{i\_des} = {} & (1-C_1)\ddot{x}_{i-1} + C_1\ddot{x}_l \\ & - \Big(2\xi - C_1\big(\xi + \sqrt{\xi^2-1}\big)\Big)\omega_n\dot{\varepsilon}_i \\ & - \big(\xi + \sqrt{\xi^2-1}\big)\omega_n C_1(V_i - V_l) - \omega_n^2\varepsilon_i,\end{aligned} \tag{5}$$

where the spacing error $\varepsilon_i = x_i - x_{i-1} + l_{i-1} + gap_{des}$, gap_{des} is the desired gap in meters, and V_l is the lead vehicle's longitudinal velocity. Further, C_1 is the weighting factor between the data from the lead vehicle and the preceding vehicle, ξ is the damping ratio, and ω_n is the controller bandwidth. The value of C_1 plays an important role in string stability. Fernandes and Nunes show that the tracking error approaches zero as the value of C_1 approaches one [47]. The authors suggest using C_1 values between 0.5 and 0.7 so that the platooning vehicles do not need to rely only on the lead vehicle's data in the cases it is not available due to transient communication errors.

In summary, the ACC [39], CACC [42], and PLATOON [25] control laws represented by Equations (1), (4), and (5) can be attributed to *States 4, 3*, and *2* in Figure 1, respectively. In the simulations carried out in this paper, we chose the controller parameters based on the research results from state-of-the-art works. For instance, the values of the parameters, e.g., $gap_{des} = 5$ m, $C_1 = 0.5$, $\xi = 1$, and $\omega_n = 0.2$ Hz, in the PLATOON controller are motivated by the arguments in [47], [49], and [26]. Ploeg *et al.* in [42] suggest suitable values for

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

the CACC controller parameters, e.g., k_p = 0.2, k_d = 0.7, T = 0.5 s. In addition, Segata shows that a string of vehicles with an ACC controller behaves safely when a time gap T = 1.2 s is maintained [26].

C. Transitioning between the Cruising States Using Degradation Cascades

So far, we have introduced the *Cruising* states that include *fuel-efficient* and *fail-operational* states and assigned some state-of-the-art controllers to different cruising states based on their suitability in terms of communication requirements. Now, we propose a Graceful Degradation and Upgradation method considering the controller requirements, such as available links and communication quality, and settings such as CTG and CDG.

1) Performance Degradation Employing Safety Contracts

Runtime monitoring of a safety critical system of systems, e.g., a platoon, is necessary to design appropriate system responses in case of transient errors; for instance, graceful performance degradation proportionally to the level of system component failure. The SafeCOP runtime monitoring architecture [50] introduces a Runtime Manager (RTM) concept that builds upon contract-based safety assurance of the components in a cooperative system. A safety contract $C =< A, G >$ of a system component can be defined as a pair of assertions in which the component behavior is guaranteed according to the Guarantee G, given that the Assumptions A are fulfilled [51]. In other words, a contract reflects the performance that is guaranteed at a particular degraded mode, given that the assumptions on the system environment are fulfilled.

In the degraded states in Figure 1, i.e., *States 3* and *4*, different levels of transient connectivity can be defined to form an ordered set of degraded operation modes, termed *degradation cascade* [20]. At various levels in the hierarchy of the degradation cascade, we can define the requirements on the controllers that the platooning vehicles should adopt and/or the extent to which the inter-vehicle distances should be increased. Based on the performance goals of the different levels of such a degradation cascade, a set of safety *contracts* can be derived. For instance, Sljivo *et al.* in [20] propose a set of safety contracts obtained from a state machine that represents a degradation cascade for different failure modes in car platooning. The authors then instantiate arguments to assure that the contracts sufficiently address the failure modes of the degradation cascade. Girs *et al.* [6] build upon the RTM concept and also define safety contracts to capture different operation modes, e.g., normal, degraded, and full-stop, in a cooperative cyber-physical system, e.g., a platoon. The definition of the safety contracts in [6] is preceded by safety analyses which describe the reasons for communication failure in a cooperative function and identify two parameters to detect the failure, i.e., Packet Delivery Ratio (PDR) and the number of consecutive packet losses. However, we have chosen not to use PDR to assess communication quality since it is an average measure that does not cover the instantaneous communication quality experienced by the platooning vehicles. The number of consecutive packet losses, however, is used as this defines the duration of an outage and relates to the *good*, *fair*, and *poor* thresholds. In our previous work [5], we proposed a set of conditions that defines performance degradation in platooning in the events of transient errors. The preliminary results demonstrate how a platooning vehicle switches between different controllers and manages to avoid collisions in dense data and road traffic scenarios. However, none of the works in [5], [6], [20] investigate the impact of packet losses on different communication quality levels that dictates the upgradation or degradation chains. Moreover, should a hazard or a permanent failure be encountered, the way emergency braking should be performed from different *Cruising States* is not addressed either.

2) Graceful Degradation and Upgradation (GDU) Method

Now, we present the proposed GDU method that aims to keep a platoon fail-operational. The GDU method builds upon the RTM concept introduced in [50] and is depicted with the help of a state machine that considers the communication quality in combination with the platoon safety requirements, Figure 2. To construct the proposed GDU method, high-level and straightforward safety requirements were first defined based on a literature review. Next, the requirements were updated and adjusted to build requirement cascades and set safety targets. Rigorous simulation studies and analyses facilitated the adjustments of the requirements. For brevity, the safety requirements are not separately elaborated in this paper. However, in Section X, we present a set of safety contracts in which the *Guarantees* G_i must fulfill the safety requirements on the component level, i.e., the *Guarantees* reflect the safety requirements. A requirement cascade that defines a hierarchy such as "System shall do X; if X cannot be done, the system shall do Y, and so on" is the basis of designing a degradation cascade [31].

As communication errors cannot be anticipated during design time, the GDU method aims to select a safe state autonomously based on the perceived communication quality during runtime. Moreover, the state should be as efficient as possible, given the occurrence of communication errors. Suppose the communication quality with the LV or the immediately preceding vehicle has changed. The GDU method then upgrades or degrades the platooning vehicle's performance by adjusting the gap to the vehicle in front and/or adopting a suitable controller with the corresponding gap policy (i.e., CDG or CTG), which is based on more or less input from the LV and/or the vehicle in front or behind, given the safety contracts.

When using the GDU method, the *connection to lead (c2l)* vehicle and *connection to front (c2f)* vehicle are monitored during runtime[1]. These connection types are further classified into *good*, *fair*, and *poor* communication qualities, which have already been discussed within the scope of the state machine in Figure 1. The GDU state machine depicted in Figure 2 is a more detailed version of the *Cruising States* presented

[1]In a platoon, the connection to the immediately following vehicle also need to be monitored for safety reasons. However, the data that needs to be exchanged using this link refers mostly to braking, and thus for brevity, we exclude this link when discussing the cruising states and return to it when discussing the emergency braking states below.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

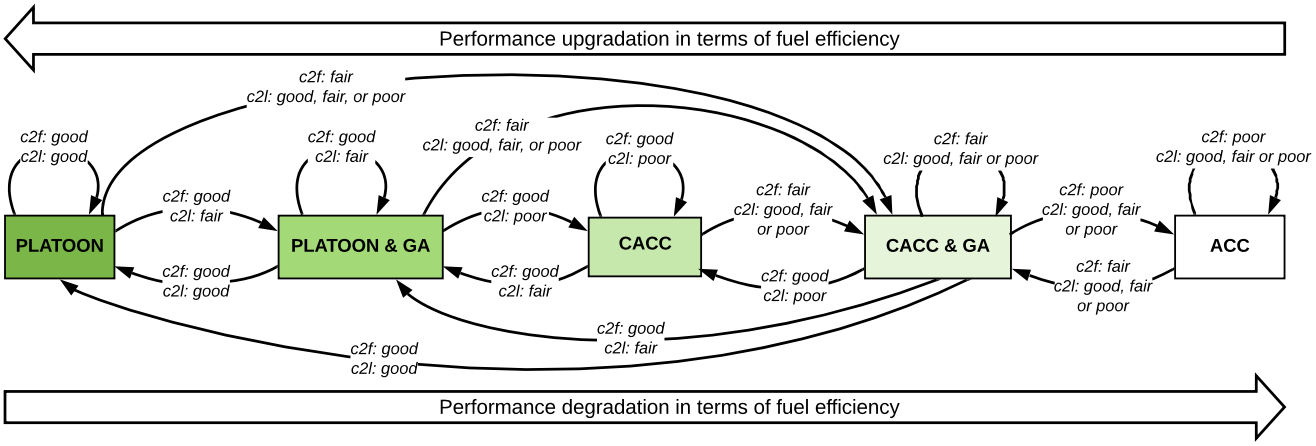


Fig. 2: Elaboration of the *Cruising States* representing the graceful degradation and upgradation method.

in Figure 1. *States 2, 3,* and *4* in Figure 1 are represented by the PLATOON, CACC, and ACC states in Figure 2. In addition, there are two intermediate states named *PLATOON & GA* and *CACC & GA*, which represent the Gap Adjustment (GA) functions of the GDU method. The aim of the GA states is to facilitate *graceful degradation*, i.e., first increasing the inter-vehicle gap slightly when a connectivity error is noticed and only switching to a less fuel-efficient controller when the specific communication requirements no longer can be fulfilled using the current controller. A slightly increased gap enables better string stability than controller switching. In addition, transitioning to the GA states before controller switching facilitates graceful acceleration or deceleration, which can, e.g., enhance passenger comfort. To this end, the GDU method presented in Figure 2 offers three types of operation modes to enable performance degradation and upgradation, i) switching between the PLATOON, CACC, and ACC controllers, which implies changing control topology and gap strategy, ii) maintaining the same controller but adjusting the gap to the vehicle in front, or iii) changing controller and adjusting the gap.

A vehicle using the PLATOON state requires *good c2f* and *good c2l* to facilitate a short inter-vehicle gap according to the CDG policy in order to enable high fuel efficiency. However, should the *c2l* deteriorate to *fair*, the vehicle can transition to the *PLATOON & GA* state by slightly increasing the distance to the vehicle in front. When delaying the change of controllers by first increasing the gap, more graceful degradation is obtained, which leads to better string stability. The rationale behind increasing the gap is due to safety, e.g., the potential risk of not being able to receive emergency braking messages from the LV in time, and thus a larger gap can maintain safety. If the *c2l* becomes *poor*, but the *c2f* remains *good*, the vehicle transitions to the CACC state in which the CTG policy is followed, and thereby the inter-vehicle gaps are much longer at high speeds than in the PLATOON state. In Figure 2, transitions to the *CACC & GA* are possible from all other states. The reason is that if the *c2f* becomes *fair*, the vehicle must increase the gap to the vehicle in front regardless of the communication quality with the LV. If the *c2f* further deteriorates and becomes *poor*, the vehicle adopts the ACC controller regardless of the quality of the *c2l*. Recall that all vehicles periodically monitor the communication quality to see if performance can be upgraded due to an improvement in communication quality. For instance, when the ACC controller is active in a vehicle and a packet is received from the vehicle in front such that the communication quality improves, the vehicle transitions to the *CACC & GA* state. In the next monitor interval, if the communication quality to the vehicle in front improves further to the *good*-threshold, the vehicle transitions to the CACC state even if no packet is yet received from the LV. Note that the states in Figure 2 represent various levels of the degradation/upgradation cascade from which a set of contracts can be derived. The current controller and the current communication quality are considered as the *assumptions* in a contract, and the degraded or upgraded operation mode is the *guarantee*. We carry out rigorous simulation studies to define the degraded modes based on the levels of communication quality, and the results are presented in Sections VII and IX. The derived safety contracts then appear in Section X since the guarantees reflect some quantitative target values obtained from the simulation studies.

The *good*, *fair*, and *poor* communication quality thresholds are defined by the number of consecutive packet losses, i.e., the duration of the transient communication outage on a specific link. The occurrence of a communication outage which makes the communication quality transition from *good* to *fair* to *poor* on a particular link will depend on a multitude of factors, some of which are analyzed further in Section VII-D. Classifying the communication quality into different levels and assigning a degraded mode based on these levels is one of the core contributions of this paper. This approach prevents aggravated degradation and facilitates the possibility of returning to the original mode (upgradation) in a short time.

V. Emergency Braking States Leading to the Fail-Safe State

In this section, the conditions for attaining a fail-safe state are first defined. Next, a state machine representing a general framework for emergency braking strategies is presented. Then the state-of-the-art braking strategies are analyzed in terms of

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

the fail-safe conditions and their placement in the proposed state machine. Finally, the proposed Enhanced Synchronized Braking strategy is presented.

A. Fail-Safe Conditions

An emergency braking strategy must fulfill two criteria: first, it must enable reaching a state which satisfies the conditions of being fail-safe. Second, it must be able to autonomously adjust to the changing quality of the wireless connectivity, e.g., by satisfying the communication requirements of *States 5, 6,* and *7* in Figure 1. We define the conditions for attaining the fail-safe state as follows:

1. The actual gap at a complete standstill must be $d_{i,stop} > 0$ i.e., no collisions have occurred between the vehicles even when the platoon completely stops.
2. Further, $d_L < d_{hazard}$, where d_L is the stopping distance of the LV since the detection of the hazard and d_{hazard} is the distance from the place where the hazard was detected to the actual hazard. This condition is to ensure that the hazard that triggered the emergency braking is avoided. It should be noted that whether or not the LV is able to fulfill the condition $d_L < d_{hazard}$ at all times depends on the distance to the hazard once it occurs or is detected. This in turn depends both on the sensors of the LV and the actual location of the occurrence of the hazard.
3. Finally, the whole platoon transitions to the safe state sufficiently fast.

B. State Machine for Emergency Braking

In the *Cruising States*, depending on the quality of the connectivity, fuel efficiency, string stability, and safety have different priorities. However, safety is the only concern in the *Emergency Braking* states according to the fail-safe conditions defined above. To this end, in Figure 3, we propose a state machine that elaborates the *Emergency Braking* states in Figure 1 and serves as a general framework for different types of braking strategies.

For convenience, the states of the state machine in Figure 3 are abbreviated depending on if deceleration has been initiated or not together with the availability of information about other vehicles. In the *No deceleration, full information available (NDFI)* state, full information on braking is available from the LV and the AVs regarding, e.g., deceleration rate to be used, distance to the hazard (d_{hazard}), when to start braking, braking intention of the immediately preceding and following vehicles (AVs), etc. However, in the *No deceleration, partial information available (NDPI)* state, information from either the LV or the AVs is missing due to packet losses. Further, the *No deceleration, only onboard information available (NDOI)* state represents that information regarding the AVs is only available through sensors, and information from the LV and AVs communicated through V2V communications is missing. These three states, i.e., NDFI, NDPI, and NDOI, originate from the *Cruising States 2, 3,* and *4* of Figure 1 upon encountering a hazard. The vehicles have not started decelerating in the NDFI, NDPI, or NDOI states. The bottom three states in Figure 3 represent that deceleration has been initiated with either *full (DFI)*, *partial (DPI)*, or *only onboard information (DOI)* available.

The horizontal transitions between the states in Figure 3 depend on whether the information is available or missing from the LV and/or the AVs. Information is said to be missing when a vehicle does not receive packets from the LV and/or AVs for a period of time. The length of the period of time depends on the nature and severity of the hazard encountered and the braking strategy being pursued. The vertical transitions from the NDFI, NDPI, and NDOI states to the DFI, DPI, and DOI states, respectively, indicate the starting of the braking maneuver based on information received through V2V communications and/or onboard sensors. Moreover, suppose the vehicles receive instructions that braking is no longer needed. In that case, they can transition back to the NDFI, NDPI, or NDOI states and eventually to the *Cruising States* in Figure 1 again. Note that during emergency braking, which is event-driven, it could be the case that only one packet regarding the braking information is sufficient, unlike the *Cruising States* where periodic packets are likely required to maintain string stability which is of the essence here. However, with some braking strategies, how and when the platooning vehicles actually start emergency braking depends on whether the event-driven messages are received from the LV or from one or more adjacent vehicles.

C. State-of-the-Art Braking Strategies

Next, we analyze several state-of-the-art braking strategies in terms of the fail-safe conditions and discuss in which states they fit in the state machine presented in Figure 3.

The most obvious and straightforward emergency braking strategy is that the FVs in a platoon perform emergency braking as soon as a hazard is detected (a DENM is received from the LV or the vehicle in front) [37]; we denote this as *Normal Braking (NB)*. Performing emergency braking with a high deceleration rate as soon as a DENM is received can be problematic since, if we are in the DPI or DOI states in Figure 3, not all FVs may have received the DENM - especially since the inter-vehicle distances are longer in these states, which aids safety but worsens communication quality.

Magdici *et al.* [52] propose to increase the deceleration rate exponentially until the maximum deceleration rate is reached in a braking scenario. This control design helps to ensure that the inter-vehicle gap remains greater than a minimum safe distance at all times. However, the authors do not consider V2V communications, i.e., the strategy is designed for use with an ACC controller and the DOI state in Figure 3. Ligthart *et al.* [53] elaborate this *gradual deceleration* approach by formulating a collision avoidance controller mathematically in conjunction with a nominal CACC controller. Their simulation results demonstrate that emergency braking with gradual deceleration can avoid collisions in a two-vehicle platoon, while sudden full-deceleration cannot. The authors use a constant duration of gradual deceleration (0.2 s). As [52] is improved in [53] to be used in conjunction with the CACC controller where information from the AVs and onboard sensors is available, it is suitable for the DPI and DOI states, but not the DFI state.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

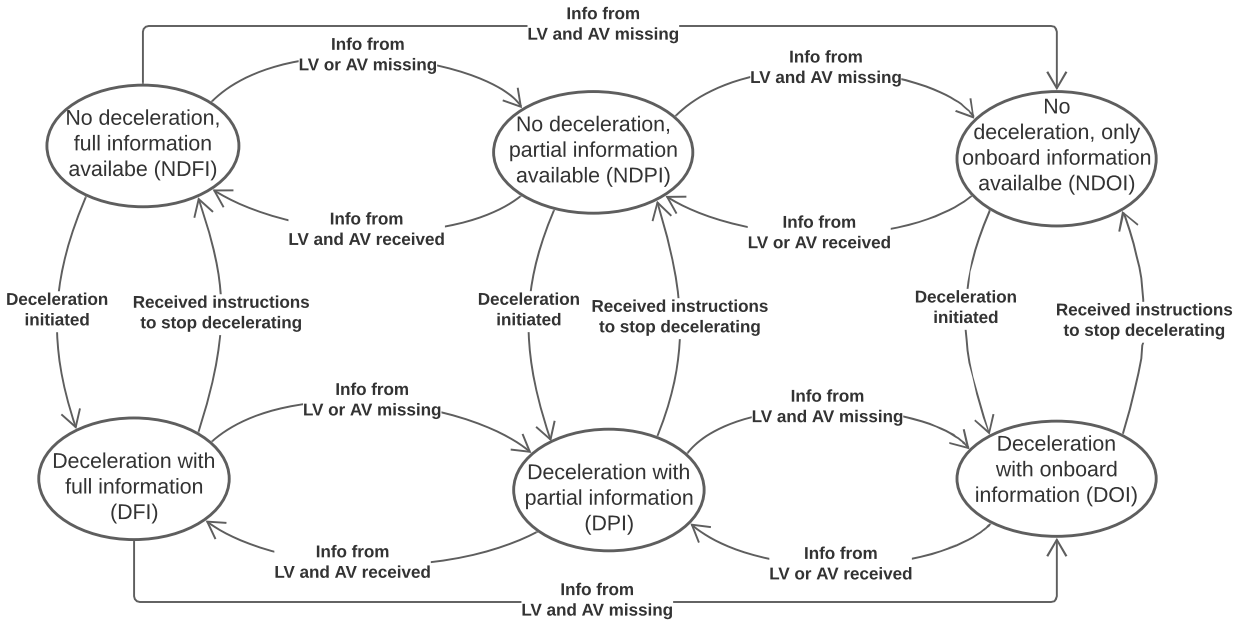


Fig. 3: Elaboration of the *Emergency Braking* states representing a general framework for different types of braking strategies.

Note that the braking strategies in [52] and [53] are evaluated only in terms of collision avoidance in a two-vehicle braking scenario. However, in a longer platoon, the tail vehicles are more prone to collisions due to higher communication latency and propagation of errors in the downstream direction. The front-most vehicles would require decelerating slower to avoid such collisions, imperiling the second and third conditions of the fail-safe state.

In [11], the authors propose a *Coordinated Emergency Brake Protocol (CEBP)* in which the last vehicle brakes first and the lead vehicle brakes last. A platooning vehicle starts braking upon receiving an acknowledgment from its immediate successor only, i.e., using a successor following strategy. Miekautsch *et al.* [13] propose to adjust the communication topology in a platoon depending on the scenarios such as cut-in or emergency braking. The authors analyze collision avoidance and stopping distance of the LV using a reverse leader-predecessor following strategy, i.e., a vehicle receives its braking instructions from the last vehicle and the immediate FV. In both [11] and [13], emergency braking is initiated by the last vehicle. However, the last vehicle in the platoon is located furthest away from the LV and therefore, has the communication link with the lowest quality due to path loss and fading effects. This implies that it will more often be in the DOI state of Figure 3 unless instructions from the LV are forwarded by its successors, in which case a higher delay is instead experienced. In addition, the propagation of the braking message from the last vehicle to the LV incurs additional delay as the braking messages are required to be relayed by all the FVs, which also includes multiple transmission attempts in case of packet losses. Therefore, the stopping distance of the LV and the whole platoon can be considerably high with the braking strategies proposed in [11] and [13]. Another reason for a possible higher stopping distance with these braking strategies is that the first DENM received from the LV is not sufficient to start the braking maneuver; additional information from one or more adjacent vehicles is required, causing further delay. Considering the communication topology, the CEBP strategy [11] can be attributed to the DPI state in Figure 3 as braking starts when a packet is received from the immediate FV, whereas the reverse leader-predecessor following strategy [13] can be placed in the DFI state. However, it is not clear if and how these braking strategies can adjust autonomously to the communication requirements of the different states, especially if not all vehicles are in the same state.

Liu *et al.* in [46] and Fernandes and Nunes in [47] show that delaying the actions of the vehicles in a platoon for a short period can help achieve synchronization, which leads to string stability in the cruising states. Murthy and Masrur also use the concept of delayed action for achieving synchronization [10], [18]; however, in the context of emergency braking rather than string stability. The authors propose that all vehicles in a platoon should wait for 20 ms before braking simultaneously. Such simultaneous braking facilitates a high deceleration rate, reducing the stopping distance of the LV and the whole platoon. However, the assumption of a 20 ms waiting time before braking is based only on controller feedback delay, but the possibility of time-varying communication delay is not considered. In [19], we evaluate the effects of delayed actions in the context of platoon emergency braking using IEEE 802.11p, which is the basis for both the US standard DSRC and the EU standard ITS-G5. As both channel quality and channel access delay are unpredictable with IEEE 802.11p [34], rather than using a fixed period, we instead propose to continuously monitor the communication latency and use the obtained average latency as the waiting period after which all the platooning vehicles should perform Synchronized Braking (SB) in the event of a hazard. In [19], we performed simulation studies to demonstrate how SB can be used in conjunction with the PLATOON, CACC, and ACC controllers to avoid

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

collisions but still enable a high deceleration rate which reduces the stopping distance of the LV. The SB strategy can be adjusted to the communication requirements of the DFI and DPI states in Figure 3 as the waiting time varies with the level of communication delay. It can also be used in the DOI state if the vehicles have been made aware of the hazard, as the waiting period based on the long-term average delay can be calculated and stored locally. However, in a dense data and road traffic scenario with high communication delay, the waiting period required in SB can lead to a long stopping distance despite the high deceleration rate it facilitates, which contradicts the second condition of a fail-safe state.

The braking strategies discussed above mainly focus on homogeneous platoons, i.e., the physical properties and dynamics of all the vehicles are the same. Emergency braking in a heterogeneous platoon has also received significant research attention. For instance, Zheng *et al.* in [14], [54] propose that the last vehicle in a platoon should brake at the highest deceleration rate, and the rate should gradually decrease in the upstream direction. The authors conduct experimental studies under the assumption that braking is performed manually by the human drivers; hence, this braking strategy does not exactly fit in the state machine in Figure 3 that assumes automated braking. However, it should also be noted that a human driver would brake differently given full, partial, or no information available about the hazard and the strategies of the other drivers. Murthy and Masrur propose *the law of the weakest*, i.e., the whole platoon should tune its maximum deceleration to the one with the weakest braking capacity [55]. As the authors use predecessor following communication topology, *the law of the weakest* strategy can be placed in the DPI or the DOI state in Figure 3. Thunberg *et al.* propose an analytical model that determines a feasible region of communication latency within which the platooning vehicles are guaranteed to perform safe braking [56]. Sidorenko *et al.* in [57] present a mathematical model to determine the minimum safe distance between two vehicles that are required to perform safe braking in a multi-brand platoon. In both [56] and [57], the authors consider leader following communication topology, i.e., partial information available; hence, these works can be attributed to the DPI and DOI states in Figure 3.

If different platoon members are in different *Cruising States* when a hazard occurs, they may learn about the hazard at different times, either through V2V communications or through distance sensors when communication is not sufficient. A good emergency braking strategy should take this into account by continuously adjusting the deceleration rates to the state of the ego vehicle and to the reported states of the other platooning vehicles. Considering the benefits and drawbacks of the different emergency braking strategies outlined above together with the criteria that an emergency braking strategy must enable, i.e., fulfilling the fail-safe conditions in Section V-A and adjusting to the instantaneous communication quality as outlined in Figure 3, we propose an improvement of the previously suggested SB strategy which is presented below.

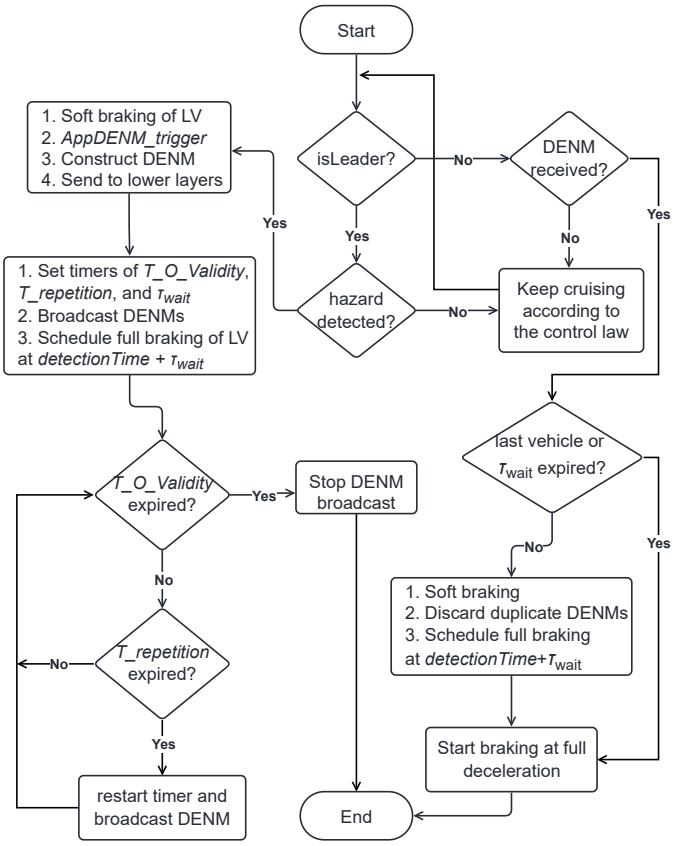


Fig. 4: Enhanced Synchronized Braking strategy.

D. Enhanced Synchronized Braking to Attain the Fail-Safe State

In this paper, we propose the Enhanced Synchronized Braking (ESB) strategy to further improve the SB strategy proposed in [19]. With SB, all vehicles wait a predefined period of time (τ_{wait}) before braking all at once (synchronized) at a much higher deceleration rate than what would be possible given the short inter-vehicle distances. This enables using a higher deceleration rate which in most cases leads to a reduced stopping distance of the LV. However, our research shows that τ_{wait} can be considerably higher in dense data and road traffic scenarios which instead can increase the stopping distance of the LV, imperiling the second condition of the fail-safe state, i.e., $d_L < d_{hazard}$. To circumvent this, using the ESB strategy, all platooning vehicles, except the last one, instead perform soft-braking immediately upon receiving a DENM. Once the agreed waiting time τ_{wait} has passed, full deceleration is then performed synchronously. Unlike with SB, the last vehicle in the platoon does not wait until τ_{wait} has passed before acting. It performs braking at a full deceleration as soon as it receives a DENM. The ESB strategy is represented by a flowchart in Figure 4.

With the ESB strategy, the LV starts soft-braking as soon as it detects a hazard and broadcasts DENMs. The DENMs are constructed according to the specifications of the ETSI DEN basic service [28]. Among other data, the DENMs contain τ_{wait} and *detectionTime* that specify the waiting time before full deceleration and the event detection time, respectively. Moreover, upon detecting a hazard, the LV starts the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

T_O_Validity and *T_Repetition* timers which signify the validity duration of the DENMs and the DENM repetition interval. The LV broadcasts DENMs at an interval of *T_Repetition* until the *T_O_Validity* timer expires. However, the LV can reset the *T_O_Validity* timer in case it detects the absence of hazards or learns that the hazard duration has increased. According to the ETSI DEN basic service [28], the *T_O_Validity* timer is set to 600 s from the event *detectionTime* by default. The LV starts full-deceleration when its τ_{wait} timer expires. The FVs learn about the τ_{wait} time from the LV upon receiving a DENM. Although different FVs can receive the DENMs at different times, the synchronization of the full-deceleration action is performed using the *detectionTime* timer (event detection time). Here, note that we assume that the clocks of the platooning vehicles are synchronized. Each vehicle, except the last vehicle, starts full-deceleration at $detectionTime + \tau_{wait}$, given that they received any DENM successfully. The soft-braking is not synchronized, i.e., the vehicles start soft-braking immediately upon reception of a DENM. However, during asynchronous soft-braking, the following vehicle using ESB has more time to react to the predecessor's speed change compared to when using SB due to slow deceleration. As a result, the following vehicle can start soft-braking using the radar sensor even if it has not yet received a DENM. Compared to an immediate full-deceleration, the same vehicle would not have enough time to react using the radar sensor only unless the inter-vehicle distance is sufficiently large [19].

In the context of the state machine proposed in Figure 3, the vehicles using the ESB strategy stay in the NDFI, NDPI, or NDOI states until braking information is received via V2V communications and/or sensors. Next, the vehicles transition to the DFI, DPI, or DOI states, which include both the soft-braking and full deceleration. Different vehicles can be in different states during soft-braking, e.g., in DFI, DPI, or DOI. However, switching to full deceleration requires the vehicles to be in either DFI or DPI. If no DENM has been received, a vehicle will simply adjust its distance to the vehicle in front. We note that as the inter-vehicle distances reduce during soft-braking, the communication quality improves, as does the likelihood of receiving a DENM. We also note that as the *detectionTime* (event detection time) is included in the DENM, a vehicle that receives a DENM will know exactly when to start braking hard. With proper selection of the deceleration rates, the ESB strategy is, therefore, possible to use in all three states NDFI, NDPI, and NDOI, in contrast to the SB strategy which is problematic if some vehicles are in the NDOI state.

It should be noted that the soft braking proposed in this paper is different from the gradual deceleration proposed in [52] and [53] as a very low deceleration rate, e.g., –2, –3 ms^{-2} is maintained during the waiting period using ESB, whereas with gradual deceleration, the deceleration rate increases exponentially until the maximum deceleration is reached. This has several benefits, as we will see in the simulations conducted for performance evaluation of the ESB strategy in Section VIII. Note that in the simulations, we consider homogeneous braking capabilities of the vehicles under the assumption that all the platooning vehicles tune their deceleration rates to the vehicle with the weakest braking capacity as proposed by Murthy and Masrur in [55]. Also, external disturbances, e.g., wind drag force, rolling resistance, variation in road slope, or vehicle mass, are not considered in the evaluation.

VI. Simulation Scenario, Settings, and Evaluation Criteria

In this section, the evaluation metrics to analyze the *Cruising*, *Emergency Braking*, and *Fail-Safe* states are first defined. After that, we describe the simulation settings and traffic scenarios.

A. Evaluation Criteria

The following criteria are used to evaluate the platoon performance in terms of fuel efficiency, string stability, and safety:

- *Minimum inter-vehicle distance* $d_{i,min}$: The minimum gap between any pair of vehicles while cruising or after the platoon completely stops is greater than zero, i.e., $d_{i,min} > 0$ m. We assume that the maintained gap at a complete standstill is irrelevant as long as collision is avoided (from a fail-safe point of view). This is to evaluate the first condition in both the fail-safe and the fail-operational states.
- *Stopping distance of the LV* (d_L): The distance traversed by the lead vehicle from the time it detects a hazard until it comes to a complete standstill. This is to evaluate if the state fulfills the second condition of being fail-safe, $d_L < d_{hazard}$.
- *Total time to stop* (t_{total}): The total time required by the whole platoon to come to a complete standstill. This metric assesses the third condition of the fail-safe state.
- *Inter-vehicle distance during cruising*: The inter-vehicle distance measured between any pair of vehicles while cruising is less than a threshold to enable fuel efficiency. The inter-vehicle distance should always be greater than zero to ensure safety, see above. To be string stable, a controller should attenuate the spacing variations from the head to the tail of a platoon.
- *Speed profiles*: The speed profiles of the platooning vehicles can be used to evaluate string stability and fuel efficiency by analyzing the variation of speed and tracking error with respect to the LV.

The following criteria are used to evaluate the communication quality:

- *good*, *fair*, and *poor* - thresholds: This metric defines the number of packet losses that should be attributed to the *good*, *fair*, and *poor* communication thresholds to control the state switching in Figures 1 and 2 during cruising of a platoon. Hence, they are also needed to evaluate the third condition of being fail-operational. These thresholds can also be expressed in terms of the duration of temporary communication outages. For instance, *poor = 4* implies that a vehicle did not receive any CAM for the last 400 ms, given that the CAM update frequency is 10 Hz.

B. Simulation Settings and Traffic Model

To facilitate the evaluation of the states and the transitions between them in Figure 1, we have extended the Plexe

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

TABLE I: Configuration parameters for simulation analysis.

	Parameter	Value
communication	PHY/MAC model	IEEE 802.11p/IEEE 1609.4
	Path loss model	Free space ($\alpha = 2$)
	Fading model	Nakagami-m (m = 1.86)
	Tx Power	100 mW
	Packet size	200 B
	Bit rate	6 $Mbps$
	Sensitivity	-94 dBm
	Thermal noise	-95 dBm
	Frequency	5.89 GHz
	Bit rate (non-platooning vehicles)	3 $Mbps$
	DENM, CAM frequency	10 Hz
	CAM frequency (non-platooning vehicles)	50 Hz
mobility	Leader speed	100 kmh^{-1}
	Platoon size	7
	Non-platooning vehicles	400
	No. of platoons	1
	Leader oscillation frequency	0.2 Hz
	Oscillation amplitude	10 $kmph$
	Total no. of lanes	4
	Vehicle density	95 vehicles/km
	Platooning vehicles insert time	1 s
	Non-platooning vehicles insert time	50 s
	Simulation time limit	$80 \sim 100$ s
controller	Controllers	PLATOON, CACC, ACC
	C_1	0.5
	ω_n	0.2 Hz
	ξ	1
	k_p	0.2
	k_d	0.7
	T	$0.2 \sim 1.2$ s
	gap_{des}	5 m
RTM	rmEnabled	$true$
	rmMonitorInterval	0.1 s
	constantSpacingFactor	0.25
	timeGapFactor	0.25
	poor	$3, 4, 5, 6$
	fair	$1, 2, 3, 4$
CEB	ESBEnabled	$true$
	brakeAtTime	70 s
	ACC, CACC τ_{wait}	1.12 s
	PLATOON τ_{wait}	0.433 s
	softDecelerationRate	$-2, -3$ ms^{-2}
	fullDecelerationRate	-8 ms^{-2}

simulator [49]. Plexe is an OMNeT++-based simulator that is built on top of Veins [58], which is a VANET simulator. In addition, Plexe extends the road traffic simulator SUMO [59] to provide realistic traffic models, vehicle dynamics, and controller implementations, e.g., PLATOON [25], CACC [42], ACC [39]. A SUMO vehicle in the Plexe simulator has a corresponding node in OMNeT++, and they communicate through TraCI interface [60], a Transmission Control Protocol (TCP) based client/server interface. As an extension of the Plexe simulation framework, we have developed two separate modules named Runtime Manager (RTM) and Cooperative Emergency Braking (CEB). The RTM module is responsible for performing the switching between the states in the GDU method (see Figure 2) based on experienced communication quality, whereas the selected emergency braking strategies, e.g., Normal Braking (NB), SB, ESB, are implemented in the CEB module. These two modules can be activated together or separately to evaluate emergency braking strategies without activating the GDU method or vice versa. This helps us compare and contrast the braking strategies independently of the GDU method as well as together with it.

A platoon of seven vehicles is simulated (the LV and last vehicle indices are V_0 and V_6, respectively). The platooning vehicles are inserted into the simulation at 1 second, and they reach the desired CDG or CTG 50 s into the simulation time. Further, 400 non-platooning vehicles are inserted in three additional left lanes to generate a challenging road and data traffic scenario, inducing high communication delays required for evaluating the robustness of the proposed GDU method and the ESB strategy. The non-platooning vehicles are injected 50 s into the simulation time with different initial positions and 50 meters inter-vehicle distances so that the platoon is in the interference range of the maximum number of neighboring vehicles; this is to avoid edge effects on the simulation results. In order to consider high-speed vehicles, all the simulations carried out in this paper use 100 kmh^{-1} speed for both the platooning and non-platooning vehicles. The channel models used to account for the path loss and fading effects are the free space path loss model with $\alpha = 2$ and the Nakagami-m fading model with $m = 1.86$, respectively. Cheng *et al.* in [61] report that fading due to increasing vehicle separations can be modeled by a Nakagami distribution, and the free space model with path loss exponent $\alpha = 2$ can be used to represent the line of sight propagation of signals in a freeway scenario [16]. The values of the parameters α and m are chosen to represent an outdoor freeway environment such as the one considered in this paper. The IEEE 802.11p and IEEE 1609.4 models that the Plexe simulator inherits from the Veins simulator simulate the PHY and MAC layers. The parameters such as transmit power, sensitivity, thermal noise, frequency band, etc., follow the IEEE 802.11p standard specifications [62]. As discussed before, the values of the controller parameters are taken from the literature, e.g., [26], [47], [49]. Table I summarizes the simulation parameters used in this research.

In this paper, we first conduct simulation studies with the RTM module based on the suggested parameter values in the literature. Then the efficacy of the proposed GDU method is evaluated by considering even shorter time gaps than what is suggested in the literature. Moreover, rigorous simulations have been performed with various *fair* and *poor* thresholds to understand their effects on the fail-operational and emergency braking states in platooning. Moreover, the simulations have been carried out for various CTGs used by the ACC and CACC controllers and CDGs used by the PLATOON controller. Table II presents a conversion table that shows CTGs in meters for various speeds. During emergency braking, the speed and the deceleration rate play crucial roles in collision avoidance and stopping distance. In the simulations performed in this paper, we consider a high speed (100 kmh^{-1}) and a strong deceleration rate (–8 ms^{-2}) to test the braking strategies in a challenging scenario. To this end, we simulate two scenarios, denoted sinusoidal scenario and braking scenario:

- *Sinusoidal scenario*: The LV oscillates at a frequency of 0.2 Hz with an amplitude of 10 kmh^{-1} for 100 s, and the FVs try to follow the LV according to the control law. The

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

TABLE II: *Inter-vehicle gaps* in meters for various CTGs and speeds. This is applicable for ACC and CACC controllers that rely on CTG policy.

CTG (s) / speed	0.2	0.3	0.4	0.5	1.0	1.2
60 kmh^{-1}	5.33	7.0	8.66	10.32	18.62	21.94
80 kmh^{-1}	6.44	8.66	10.88	13.1	24.2	28.64
100 kmh^{-1}	7.55	10.33	13.11	15.89	29.79	35.35

purpose of oscillating is to introduce periodic acceleration and deceleration on the LV motion to evaluate how well the FVs can track the leader under such disturbances [41].

- *Braking scenario*: The LV initiates emergency braking upon detecting an imaginary road hazard 70 s into the simulation time.

In the subsequent sections, the efficacy of the proposed GDU method and the ESB strategy in maintaining fail-operational and fail-safe states are first evaluated independently using the RTM and CEB modules; then, they are evaluated together.

VII. Evaluation of the Cruising States

This section begins with the evaluation of the PLATOON, CACC, and ACC controllers in terms of fuel efficiency, string stability, and Lead Vehicle (LV) tracking ability. Next, the simulation results related to the cruising of a platoon or vehicle string without activating the RTM module are presented. Then the GDU method proposed in this paper is analyzed using the RTM module. The aim is to understand the efficiency of the GDU method in maintaining fail-operational states for various *fair* and *poor* communication thresholds. Moreover, the evaluation results regarding the *fair* and *poor* communication thresholds that dictate the transition between the cruising states are presented in this section as well. In our previous work [5], we showed how the RTM governs the switching between different controllers to avoid collisions for some selected *fair* and *poor* thresholds.

A. Evaluation of Fuel Efficiency and String Stability

Recall from Section III that fuel efficiency, string stability, and safety are the primary goals of *State 2* in Figure 1, that string stability and safety are the main focus in *State 3*, and finally, that safety is the key concern in *State 4*. Fuel efficiency is evaluated under the assumption that the controller facilitating the shortest longitudinal gap enables the highest fuel efficiency. This is motivated by the fact that the longitudinal gap between the vehicles is one of the major influencing factors on fuel efficiency [63]. Therefore, shorter gaps enable higher fuel efficiency due to the reduction of aerodynamic drag. In addition, when the FVs in a platoon experience tracking error with respect to the LV due to its speed variation, the resultant uneven inter-vehicle gaps may affect the overall fuel efficiency in the platoon. To this end, we examine the speed profiles of the vehicles in a sinusoidal scenario to evaluate the string stability, fuel efficiency, and LV tracking ability of the FVs.

Figure 5 shows the speed profiles of the vehicles using the *sinusoidal scenario* with inter-vehicle distances obtained from

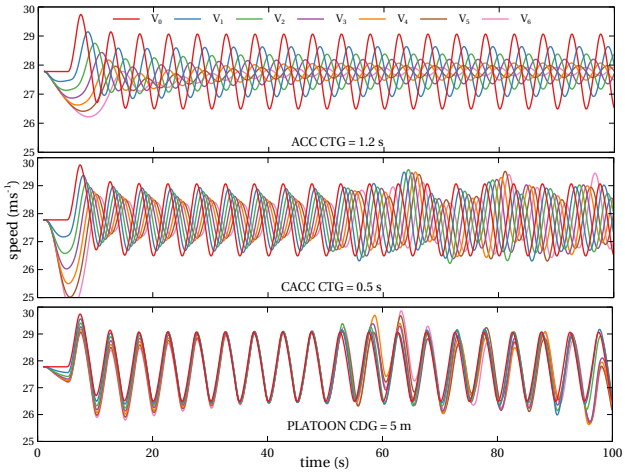


Fig. 5: The speed profiles of the vehicles in a sinusoidal scenario with ACC, CACC, and PLATOON controllers and no RTM; oscillation frequency = 0.2 Hz, oscillation amplitude = 10 km^{-1}, ACC CTG = 1.2 s (35.35 m at 100 kmh^{-1}), CACC CTG = 0.5 s (15.89 m at 100 kmh^{-1}), PLATOON CDG = 5 m, speed = 100 kmh^{-1}.

the literature, i.e., ACC CTG = 1.2 s (35.35 m at 100 kmh^{-1}), CACC CTG = 0.5 s (15.89 m at 100 kmh^{-1}), and PLATOON CDG = 5 m. The results of one representative simulation run are presented for brevity. Note that the RTM module is not activated here because we are interested in the performance of the different control algorithms. The speed profiles with the ACC controller show that the FVs can attenuate the speed variations of the LV, i.e., the platoon exhibits string stability when a 1.2 s time gap is maintained. However, for shorter time gaps than 1.2 s, the ACC controller does not demonstrate string stability [25], [26]. Although the ACC controller exhibits string stability with a 1.2 s time gap, the ability of the FVs to track the LV diminishes in the downstream direction of the vehicle string. Moreover, the last vehicle has at least one complete cycle phase lag compared to the LV due to the amplification of the sensor detection, processing, and actuation delays from the head to the tail of the vehicle string. Therefore, the ACC controller exhibits string stability when the gap is 35.35 m at 100 kmh^{-1} but demonstrates less fuel efficiency and LV tracking ability. This situation is somewhat alleviated when the CACC controller is used as V2V communication is then added to the ACC controller. During the first 50 s, there is no interference from the non-platooning vehicles. As a result, the vehicles exhibit string stable behavior for the first 50 s. However, the speed error is amplified downstream when non-platooning vehicles start generating interference (during the period 50-100 s), causing high communication delays. Despite this, it is still better than ACC in terms of LV tracking and phase lag. The vehicles exhibit highly string stable behavior with the PLATOON controller when there is no interference from the non-platooning vehicles, as can be seen by the speed of the vehicles for the first 50 s with the PLATOON controller. However, also in this case, the string stability of the rear vehicles eventually diminishes due to long channel access delays and packet drops induced by the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

data traffic of the neighboring vehicles (50-100 s). A closer look at the speed profiles between 50–100 s reveals that the tail vehicles in the platoon experience more tracking error and string instability with respect to the LV compared to the vehicles in the head of the platoon. Such situations are more hazardous with the PLATOON controller than with the CACC controller because we use a considerably shorter inter-vehicle gap with the PLATOON controller (5 m), which means less time to react in case of speed changes. In such a scenario, the GDU method, if in use, would instruct the tail vehicles to increase the gap to the vehicle in front or switch to the CACC or ACC controller based on the experienced communication quality with the LV and the preceding vehicle. However, the front vehicles which experience *good* or *fair* communication quality with the LV would use the PLATOON controller to facilitate better fuel efficiency, string stability, and LV tracking. The idea of the GDU method is that all the platooning vehicles do not need to adopt a less fuel-efficient and less string-stable controller when only the last one or two vehicles experience *poor* communication quality.

Based on the simulations, we can conclude that the PLATOON controller is more fuel-efficient, string stable, and exhibits better LV tracking ability than the CACC and ACC controllers. However, the PLATOON controller has high requirements on the attainable communication quality in order to maintain sufficient safety. The communication quality with the LV is essential, especially if braking should be necessary, as the inter-vehicle gaps are small. Next, we evaluate the safety aspects of the different controllers, which is the top priority in all the states in Figure 1.

B. Evaluation of Safety without State Switching

In this part, we focus on evaluating the safety of the PLATOON, CACC, and ACC controllers by examining their inter-vehicle distances in the sinusoidal scenario to see if the benefits in terms of fuel efficiency from Figure 5 are obtained at the expense of safety.

Figure 6a presents the platooning vehicles' distance profiles using the PLATOON controller with 5 m CDG and no RTM in play, following the *sinusoidal scenario*. Five simulation runs with different seeds are shown for the same scenario. In three out of the five runs, the last vehicle in the platoon undergoes collisions: in runs no. 1, 2, and 4, the collisions happen at 80, 70, and 96 s of the simulation time. The main reason for the collisions is the communication delays due to packet drops and channel access delays caused by the many neighboring vehicles used in the simulation setting. Moreover, the last vehicle experiences the highest delay due to path loss and fading effects as it is the farthest away from the LV. As the vehicles are using the PLATOON controller and the weighting factor C_1 is 0.5 (see Equation 5), the platoon's following vehicles require CAMs from the LV to continue platooning. An average of 100 simulation runs shows that the last vehicle experiences a 432.97 ms delay in this scenario (these results are not presented here for brevity). In this case, the logical thing would be to increase the gap and use a predecessor following strategy (C_1 = 0) like the CACC controller. For instance, the vehicles do not collide under the same network load when the CACC controller is used with a 0.5 s CTG, i.e., 15.89 m at 100 kmh^{-1} (see Figure 6b). This is precisely what the GDU method does; it monitors the LV's and the front vehicle's communication quality and chooses an appropriate controller or gap adjustment during runtime. Our simulations also show that there are no collisions when using the ACC controller when longer CTGs are used, e.g., 1.2 s. In [26] and [25], the authors also show that a vehicle string can avoid collisions during cruising with 1.2 s CTG using the ACC controller (35.35 m gap at 100 kmh^{-1} speed). In order to provide an acceptable trade-off between fuel efficiency and safety, it is necessary to allow switching between different controllers, given the instantaneous communication quality. Moreover, it is important to allow different vehicles to be in different states based on the information at hand. The PLATOON controller is sufficiently safe as long as updated data from the LV is available, but this may not be the case for the last vehicle in the platoon in a dense data traffic scenario. Still, this should not prevent vehicles located closer to the LV from selecting a fuel-efficient controller.

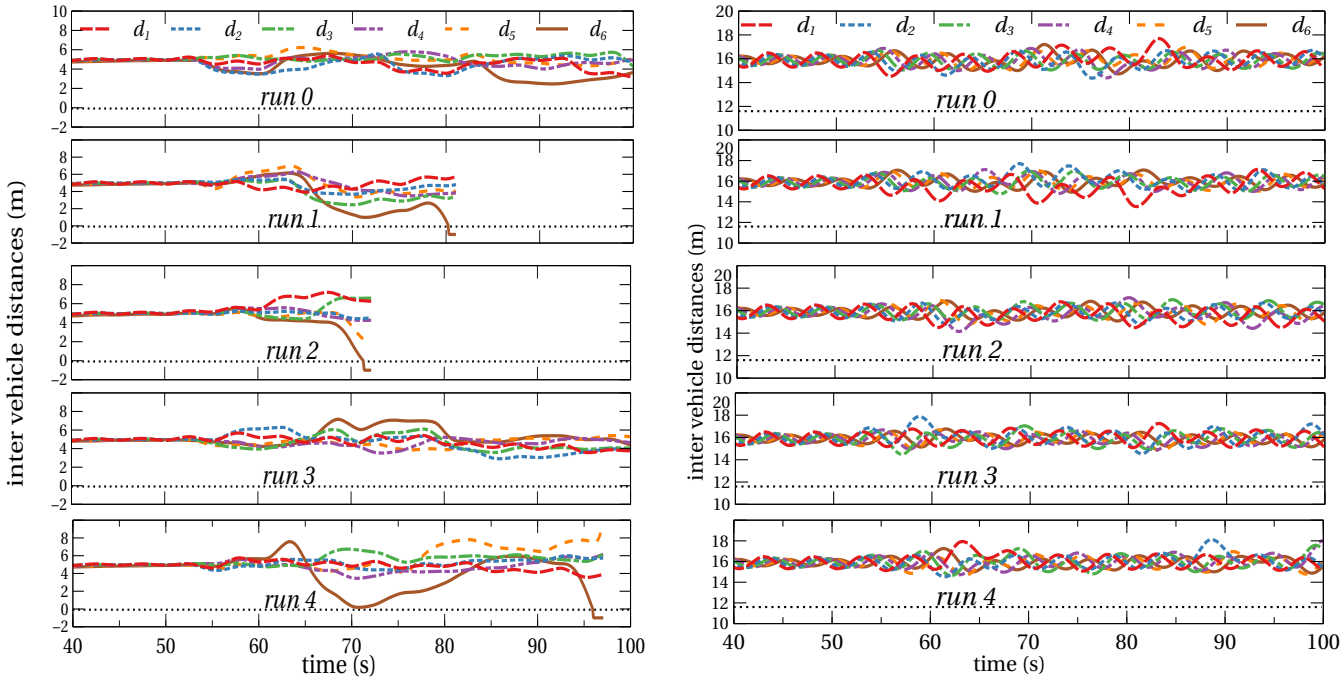
C. Evaluation of Performance when Allowing Autonomous Switching between States

The same scenario as in Figures 5 and 6 is now simulated with the RTM module (implying that autonomous switching between the PLATOON, CACC, and ACC controllers can be made) for various combinations of *fair* and *poor* thresholds, see Figure 7. This section thereby addresses ***RQ1***. More specifically, we chose 13 combinations of (*fair*, *poor*) thresholds taking the Cartesian product of sets A = {1, 2, 3, 4} and B = {3, 4, 5, 6} such that $A \times B = \{(fair, poor) \mid fair \in A \wedge poor \in B \wedge fair < poor\}$. Recall from Table I that the CAM frequency is 10 Hz; hence, the (*fair*, *poor*) thresholds, e.g., (2, 5) can be translated as temporary communication outages for 200 and 500 ms, respectively.

The RTM uses ACC CTG = 1.2 s which corresponds to 35.35 meters at 100 kmh^{-1}, CACC CTG = 0.5 s which is 15.89 meters at 100 kmh^{-1}, and PLATOON CDG = 5 m and switches in-between based on the communication quality. Moreover, in the Gap Adjustment (GA) states, the gaps are increased or decreased by 25% of the original gaps. For brevity, the speed and distance profiles of all the combinations of *fair* and *poor* thresholds are not presented here. Our simulations show that the RTM can successfully help the platoon avoid collisions during cruising for all 13 combinations. Let us first look at the inter-vehicle distance profiles in Figure 7. It is evident that the collision cases shown in Figure 6a with a PLATOON controller are avoided here when GDU is applied. The reason is that when the vehicles experience temporary communication outages for the duration dictated by the selected *fair* or *poor* thresholds, they adjust the gaps or switch to the CACC or ACC controller based on the rules defined in the state machine in Figure 2. Moreover, recall that we may also keep the same controller but increase the inter-vehicle distance, i.e., the intermediate states with GA, as proposed in Figure 2. These gap adjustments can be made

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958



(a) *distance profiles with PLATOON controller, 5 m gap.*

(b) *distance profiles with CACC controller, 15.89 m gap.*

Fig. 6: *Inter-vehicle distances* in meters *during cruising* in a sinusoidal scenario with PLATOON and CACC controllers and no RTM in play; oscillation frequency = 0.2 Hz, oscillation amplitude = 10 kmh^{-1}, (a) PLATOON CDG = 5 m, (b) CACC CTG = 0.5 s, i.e., 15.89 m at 100 kmh^{-1}, speed = 100 kmh^{-1}.

with higher or lower granularity to maintain better string stability and/or fuel efficiency. The size of the GA can also be adjusted depending on the selected update rate of the communicated packets (the CAM rate), as this affects the *fair* and *poor* threshold values. Note that the CAM update rate can change with, e.g., the mobility parameters of the vehicles or when the Decentralized Congestion Control (DCC) mechanism [64] instructs the vehicles to update the CAM frequency. For an ego vehicle to be able to detect the packet losses with its predecessor and the LV, the platooning vehicles should include their currently used packet update rates in the CAMs. The simulation results suggest that the RTM is very robust *in terms of collision avoidance* in the fail-operational states for all choices of *fair* and *poor* threshold values. This is because the RTM decentralizes the platoon such that even when exposed to transient errors, it ascertains the appropriate control law for the individual platooning vehicles based on their respective communication qualities. However, to attain good fuel efficiency and string stability, the choice of *fair*, *poor* thresholds matters.

We can see that even if the selected inter-vehicle distances for the different controllers and the thresholds used for deciding when to change states are not optimized, the platoon vehicles still manage to attain better fuel efficiency without compromising safety when allowing autonomous switching between states using the state machine we propose. The PLATOON controller is tractable to use as the speed profiles and inter-vehicle distances enable efficiency, but unfortunately, it is not sufficiently safe during transient communication outages. The ACC controller, in turn, is not fuel-efficient and thereby does not add many benefits except driver offload despite adding complexity. Using the GDU and allowing the communication quality to be classified with better granularity as opposed to the traditional way of declaring it as working or failed provides significant gains. However, it should be noted that too frequent changes to the vehicle speed are also not fuel-efficient, even if the inter-vehicle gaps are small. Hence, next, we attempt to determine to what extent the selected levels of the *fair* and *poor* thresholds affect the performance.

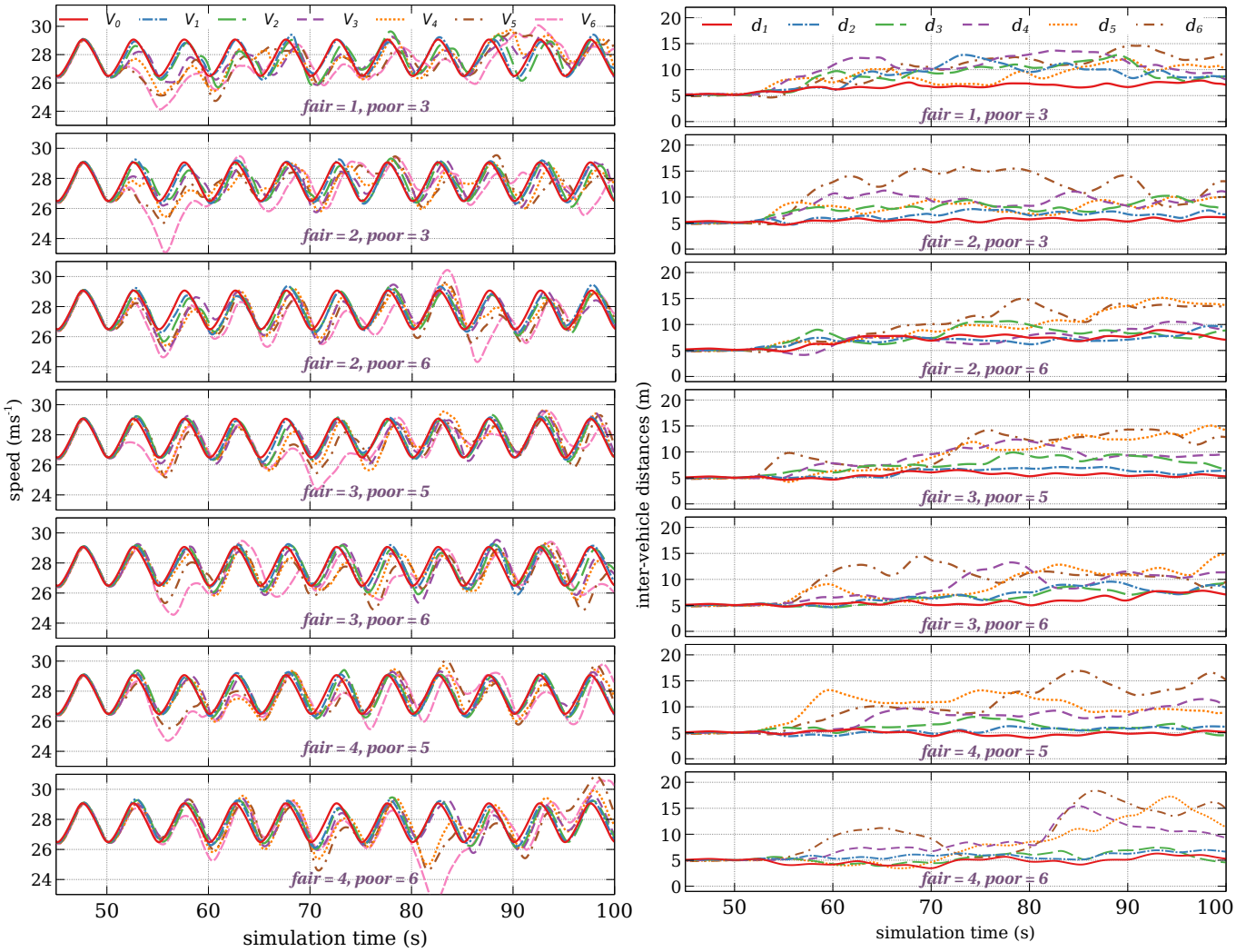
D. Impacts of Fair, Poor Thresholds on String Stability and Fuel Efficiency

The same scenario as in Figure 7 is used to analyze the impacts of *fair*, *poor*-thresholds. This section thereby addresses *RQ2*.

The speed profiles in Figure 7 demonstrate that the tail vehicles, i.e., V_5, and V_6, undergo frequent state switching when the (*fair*, *poor*) thresholds are (1, 3) and (2, 3). When the *fair* threshold is small, e.g., outage for 100 or 200 ms, which is likely to happen rather frequently, the vehicles increase the gap to their respective front vehicles more frequently while in the PLATOON and CACC states. Also, due to small *poor* thresholds, e.g., outage for 300 ms, the vehicles switch between PLATOON and CACC controllers more frequently. It should be noted that too frequent state switching causes the inter-vehicle gaps to change frequently, which is less fuel efficient and less string stable, e.g., the rear vehicles' gaps toggle between 10–15 m in Figure 7. We can prevent too frequent state switching by increasing the (*fair*, *poor*) thresholds. For instance, the platooning vehicles exhibit better

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958



(a) Speed profiles for various fair and poor thresholds

(b) Corresponding inter-vehicle distance profiles

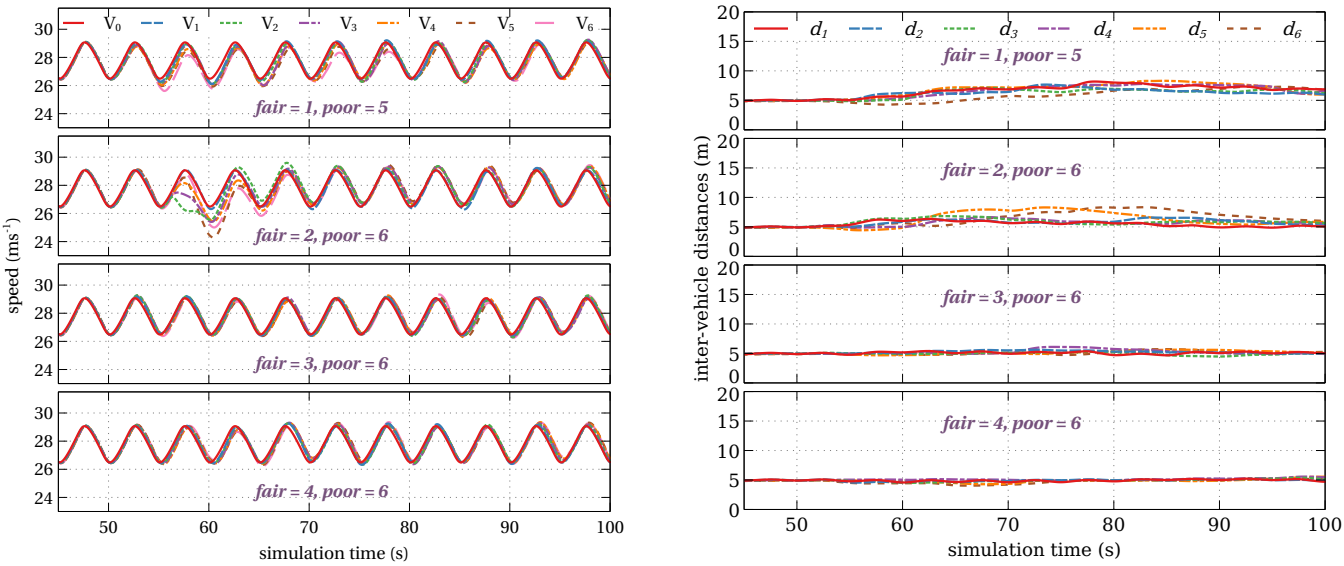
Fig. 7: Speed (ms^{-1}) and inter-vehicle distance (m) profiles during cruising in a *sinusoidal scenario* for various *fair*, *poor* thresholds using the GDU method; oscillation frequency = 0.2 Hz, oscillation amplitude = 10 km^{-1}, ACC CTG = 1.2 s, CACC CTG = 0.5 s, PLATOON CDG = 5 m, speed = 100 kmh^{-1}.

string stability and LV tracking for the thresholds (2, 6), (3, 5), (3, 6), (4, 5), and (4,6); see the speed profiles in Figure 7. The corresponding distance profiles show that the inter-vehicle gaps are between 5–10 m for all except the last two vehicles. This implies that the tail vehicles are less fuel-efficient, which is acceptable since safety takes precedence over fuel efficiency. The tail vehicles in the platoon have longer gaps due to experiencing *poor* communication quality with the LV. Hence, the tail vehicles toggle between the CACC and CACC & GA states most of the time. On the contrary, vehicles V_1 and V_2 in the platoon (d_1 and d_2 in the distance profiles) maintain short inter-vehicle gaps during the entire time when *fair*, *poor* thresholds are higher. The reason is that the front vehicles have *fair* or *good* communication quality with the LV, which is required to use the PLATOON controller. V_3 and V_4 can also maintain comparatively shorter distances for higher thresholds for the same reason.

To better understand how the RTM governs switching between the states in Figure 2, let us take a closer look at the distance profiles of vehicle 6 (d_6) with thresholds (2, 3) and (4, 6) in Figure 7 as examples. Note that the inter-vehicle gap in the PLATOON state is 5 m and 15.89 m (at 100 kmh^{-1}) in the CACC state. In the PLATOON & GA and CACC & GA states, the gaps are increased by 25%; therefore, the inter-vehicle gap in the PLATOON & GA state is 6.25 m and 19.86 m (at 100 kmh^{-1}) in the CACC & GA state. With thresholds (2, 3), the gap d_6 quickly increases in the beginning and remains around 15 m during the simulation period of 70-80 s. The reason is that vehicle 6 uses the CACC state during 70-80 s due to experiencing *poor* communication quality with the LV. In the distance profiles representing thresholds (4, 6), vehicle 6 (d_6) experiences *poor* communication quality with the LV around 55 s into the simulation time, and d_6 starts increasing to adopt the gap of the CACC state (15.89 m). However, at around 65 s, vehicle 6 starts experiencing *fair* communication quality with the LV. The state machine in Figure 2 dictates that a vehicle

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958



(a) Speed profiles for various *fair*, *poor* thresholds.

(b) Corresponding inter-vehicle distance profiles.

Fig. 8: Speed (ms^{-1}) and inter-vehicle distance (m) profiles during cruising in a sparse data and road traffic scenario for various *fair*, *poor* thresholds using the GDU method; neighboring vehicles = 250, beacon frequency of neighboring vehicles = 25 Hz, vehicle density = 65 vehicles/km, oscillation frequency = 0.2 Hz, oscillation amplitude = 10 km^{-1}, ACC CTG = 1.2 s, CACC CTG = 0.5 s, PLATOON CDG = 5 m, speed = 100 kmh^{-1}.

should adopt the PLATOON & GA state when *c2l* is *fair* and *c2f* is *good*. To this end, vehicle 6 upgrades its performance by adopting the PLATOON & GA state, and d_6 becomes 6.25 m at around 73 s, continuing until 78 s. Then communication with the LV becomes *poor* again, and vehicle 6 increases its gap to degrade its performance in order to prioritize safety over fuel efficiency by adopting the CACC state. Notice that the tail vehicles in Figure 7 do not maintain a stable distance in one particular state due to frequent state switching caused by the time-varying communication quality. We also note that with a higher *poor* threshold, the platooning vehicles rarely transition to the ACC state as it requires a higher number of packet losses with respect to the vehicle in front. For this reason, we do not see the inter-vehicle gaps reaching up to 35.35 m, which is the gap considered for the ACC controller in the simulations of Figure 7. This indicates that the thresholds should possibly be selected or adjusted based on a vehicle's position within the platoon.

In summary, the more frequent state switching that can be observed with smaller *fair*, *poor* thresholds cause inter-vehicle distances to vary frequently, affecting communication quality. However, there are fewer state transitions for higher values of the *fair* and *poor* thresholds. This aids fuel efficiency, string stability, and the ability to track the LV by preventing too frequent changes in communication topology. Note that the observations above are made in the context of dense data and road traffic scenarios. A reader may wonder how the GDU method would perform in a sparse data and road traffic scenario in which the PLATOON controller by itself would show good performance, i.e., no switching. To this end, we simulate a sparse data traffic scenario by considering 250 neighboring vehicles instead of 400, a vehicle density of 65 vehicles/km instead of 95 vehicles/km, and the neighboring vehicles now have 25 Hz beacon frequency instead of 50 Hz. To put things into perspective, the neighboring vehicles generate 1,625 beacons $s^{-1}\ km^{-1}$ in the sparse scenario compared to 4,750 beacons $s^{-1}\ km^{-1}$ in the dense scenario. The simulation results with thresholds (1, 5), (2, 6), (3, 6), and (4, 6) using the GDU method under the considered sparse scenario are presented in Figure 8. The speed profiles in Figure 8a show that the vehicles exhibit excellent performance in terms of safety, string stability, and LV tracking. In this case also, the higher *fair*, *poor* thresholds are safe and efficient because there are fewer packet losses in a sparse data and road traffic scenario. The corresponding inter-vehicle distance profiles in Figure 8b show that the vehicles can maintain stable gaps around 5 meters with higher *fair*, *poor* thresholds in the sparse scenario as well, which enables high fuel efficiency. Therefore, it is safe to conclude that the GDU method performs as good as the PLATOON controller by itself in a sparse scenario. However, in a dense traffic scenario under transient communication outages in which the PLATOON, CACC, and ACC controllers by themselves either lack safety, fuel efficiency, string stability, or LV tracking, the GDU method provides a balanced trade-off by degrading the performance of only those vehicles which experience temporary communication outage.

In concluding remarks, Figure 7 shows that the GDU method tackles the safety concern of the PLATOON controller, i.e., inter-vehicle collisions, by temporarily degrading the fuel efficiency and string stability of the last two vehicles that cause collisions in Figure 6a. However, the front five vehicles exhibit highly string-stable behavior and maintain short inter-vehicle gaps. In addition, the GDU method inherits the LV tracking capability of the PLATOON controller, which the CACC and

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

ACC controllers lack (see Figure 5). Furthermore, compared to the CACC and ACC controllers in which all the vehicles are less fuel-efficient due to longer gaps, only the rear vehicles that experience more communication outages are less fuel-efficient with the GDU method. Therefore, the GDU method exhibits safe platoon cruising by facilitating a balanced trade-off with string stability, fuel efficiency, and LV tracking.

VIII. Evaluation of Emergency Braking

In this section, the Enhanced Synchronized Braking (ESB) strategy is evaluated without RTM in terms of the fail-safe conditions (see Section V) and compared to the Synchronized Braking (SB) and Normal Braking (NB) strategies. To this end, the *braking scenario* is used in which the vehicles cruise using the PLATOON, CACC, or ACC controllers, and 70 s into the simulation, the vehicles transition into the NDFI, NDPI, or NDOI states in Figure 3 upon encountering an imaginary road hazard. Note that the evaluation is conducted under the assumption that an emergency may arise from any of the *States 2, 3*, or *4* in Figure 1. This implies that if a vehicle is at the NDFI state, it is possible to transition to the NDOI state while still maintaining the short gap from *State 2* of Figure 1. This section elaborates on *RQ3*.

A. Minimum Inter-Vehicle Distances at Full Stop

Table III presents the minimum inter-vehicle gaps at a complete standstill using the ESB, SB, and NB strategies in the ACC, CACC, and PLATOON states. Five simulation runs are shown for various CTGs and CDGs. We have chosen not to use the same CTGs for ACC and CACC states, respectively. This is because different controllers facilitate different time gaps, e.g., ACC is not suitable for having a 0.2 s time gap due to its reliance on onboard sensors. The negative values in Table III represent collision cases. The vehicle string or platoon exhibits poor performance in terms of collision avoidance while using the NB strategy; see Table III. On the contrary, the SB strategy alleviates the performance drastically compared to the NB strategy. However, there are still collision cases with the SB strategy for various CTGs and CDGs in some simulation runs. The reason is that if a vehicle does not receive a DENM within the τ_{wait} time with the SB strategy, then a full deceleration by the predecessor comes as a surprise, and the vehicle cannot avoid collisions unless the gap is adequate. The ESB strategy exhibits only two collision cases with 0.5 s CTG and 6 m CDG in the ACC and PLATOON states, respectively. This is due to an unusually long DENM delay experienced by the last vehicle, which is higher than the average waiting time. In general, there are fewer collision cases with the ESB strategy than with the SB strategy. The reason is that a vehicle can start soft-braking using the onboard sensors despite not receiving a DENM with the ESB strategy. The soft-braking gives a deceleration edge to a vehicle while using the ESB strategy compared to the SB strategy. However, if the DENM delay is too high and the gap is not sufficiently long, a collision occurs, which is why it is important to adjust the controller to the communication quality using the GDU method or something similar. Table III shows that the ESB strategy can avoid collisions in most situations, even with short CTGs and CDGs under dense data and road traffic scenarios.

B. Stopping Distance of the LV

Table IV presents the stopping distance of the LV (m) for various τ_{wait} times and soft-deceleration rates (ESB only) using the ESB and SB strategies. Note that the stopping distance of the LV with NB strategy is fixed and equal to 60.82 m at 100 kmh^{-1}. The ESB strategy demonstrates shorter stopping distances compared to the SB strategy. For instance, with the ESB strategy, the LV traverses 12.84 meters less than with the SB strategy when $\tau_{wait} = 1.12$ s (the waiting time at 0.5 s CTG). For a shorter waiting time, which is preferred in light data and road traffic scenarios, the difference in stopping distance with ESB and SB strategies is not significant, e.g., 1.05 m for $\tau_{wait} = 0.1$ s. Nevertheless, every meter counts in a safety-critical system. Moreover, recall that the ESB strategy shows better performance in terms of collision avoidance. On the other hand, the stopping distance of the LV is shorter with the NB strategy (60.82 m) compared to the ESB and SB strategies, as there is no waiting before emergency braking for synchronization purposes. However, platooning vehicles using normal braking requires decelerating slower to avoid collisions which ultimately increases the stopping distance of the LV significantly [19].

C. Total Time to Stop

Table V presents the average *total time to stop* the whole platoon t_{total} (s) for the same configurations as in Tables III and IV. The ESB strategy enables the platoon to transition into the fail-safe state faster than the SB and NB strategies due to soft-braking before full deceleration. In general, for all the braking strategies, the platoon requires a longer time to stop when in the ACC state compared to the CACC and PLATOON states. Moreover, it takes longer time to stop for longer time gaps and distance gaps as the tail vehicles experience more delays. Although the vehicles perform synchronized braking, if a vehicle does not receive a DENM within the τ_{wait} period, it starts braking later, and the total time to stop the whole platoon thereby increases. However, while braking from the PLATOON state, the total time to stop is significantly lower than the ACC and CACC states due to shorter CDGs that allow lower DENM delays.

IX. Evaluation of Autonomous Transitions between All States in the State Machine

So far, the GDU method and the selected braking strategies, e.g., NB, SB, and ESB, have been evaluated separately with or without the RTM and CEB modules. In this section, the GDU method and the selected braking strategies are evaluated together using the RTM and CEB modules of the extended Plexe simulator that we have developed. The aim is to evaluate the transition from cruising states to the fail-safe state through the emergency braking states.

We consider a scenario in which a platoon starts cruising using the PLATOON controller, and then the vehicles switch

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

TABLE III: *Minimum inter-vehicle gaps at full stop* (m) in an emergency braking scenario using ESB, SB, and NB strategies from the ACC, CACC, and PLATOON states. The results of various CTGs and CDGs are presented here; soft-deceleration rate = –2 ms^{-2}, full-deceleration rate = –8 ms^{-2}, speed = 100 kmh^{-1}.

Controllers		ACC CTG (s)				CACC CTG (s)				PLATOON CDG (m)			
	Gaps	0.4	0.5	0.6	0.8	0.2	0.3	0.4	0.5	3	4	5	6
ESB	run 0	12.17	11.29	19.69	24.13	4.01	6.86	12.17	2.08	0.3	3.92	0.92	4.02
	run 1	4.8	13.61	15.5	8.9	6.62	7.74	10.6	2.73	2.92	3.92	3.22	0.37
	run 2	10.23	-1	3.37	19.92	5.24	6.66	12.26	12.47	2.92	3.25	3.22	-1
	run 3	12.17	2.84	18.57	0.49	6.1	0.74	1.51	0.47	0.25	2.28	3.63	5.46
	run 4	12.17	1.17	1.74	21.07	3.02	8.06	10.46	14.11	1.89	3.92	4.92	2.71
SB	run 0	13.1	15.88	18.66	19.0	7.55	10.33	5.43	15.88	1.05	3.99	1.54	3.95
	run 1	1.6	15.88	3.1	24.21	7.55	10.33	13.11	10.69	1.73	3.99	-1	3.91
	run 2	-1	-1	-1	24.21	7.55	-1	-1	15.88	0.98	3.99	4.99	-1
	run 3	13.1	15.88	18.66	24.21	-1	2.03	13.11	-1	1.31	-1	4.99	5.99
	run 4	13.1	15.88	18.66	-1	7.55	10.33	13.11	15.88	-1	3.99	4.99	1.68
NB	run 0	-1	-1	-1	0.12	-1	-1	-1	0.37	0.3	-1	-1	-1
	run 1	-1	-1	-1	0.12	-1	-1	-1	-1	-1	-1	-1	-1
	run 2	-1	-1	-1	0.12	-1	-1	-1	-1	-1	-1	-1	2.12
	run 3	-1	-1	-1	0.12	-1	-1	-1	-1	-1	-1	-1	2.09
	run 4	-1	-1	-1	0.12	-1	-1	-1	-1	-1	-1	-1	-1

TABLE IV: The *stopping distance of the LV* (m) using the ESB and SB strategies for various waiting times and soft-deceleration rates; speed = 100 kmh^{-1}, full-deceleration rate = –8 ms^{-2}. Note that the stopping distance of the LV with NB strategy is 60.82 m at 100 kmh^{-1} speed as τ_{wait} = 0.

Strategy	soft-rate \ τ_{wait} (s)	0.1	0.25	0.3	0.433	0.5	0.6	0.8	1.0	1.12	1.5
ESB	–2 ms^{-2}	62.9	65.98	67	69.84	71.05	73.05	77.01	80.91	83.21	90.38
	–3 ms^{-2}	62.55	65.1	65.95	68.28	69.27	70.9	74.11	77.25	79.09	84.75
SB	None	63.6	67.77	69.16	73.04	74.71	77.49	83.05	88.61	91.93	102.49

TABLE V: *Total time to stop* (s) for different CTGs and CDGs using the ESB, SB, and NB strategies from the ACC, CACC, and PLATOON states (same configurations as in Table III).

Controllers	ACC CTG (s)				CACC CTG (s)				PLATOON CDG (m)			
Gaps	0.4	0.5	0.6	0.8	0.2	0.3	0.4	0.5	3	4	5	6
ESB	5.09	5.27	5.35	5.13	4.89	5.01	4.95	5.39	4.43	4.39	4.39	4.57
SB	5.41	5.33	5.53	5.47	5.23	5.29	5.27	5.31	4.51	4.57	4.61	4.55
NB	7.21	7.61	9.11	9.41	4.63	4.91	4.65	5.11	4.61	4.73	5.15	4.87

TABLE VI: Simulation of a normal braking scenario with RTM for various *fair*, *poor*-thresholds, and *braking scenario-1* and *braking scenario-2*.

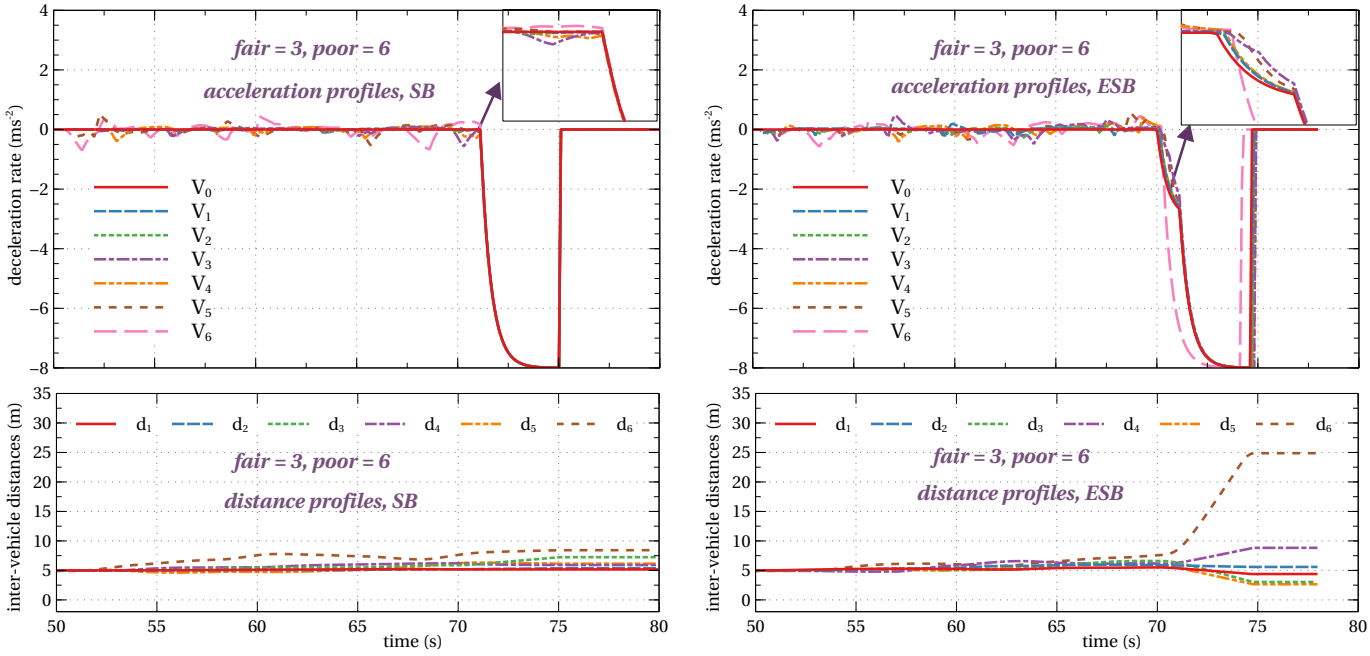
Scenarios	results \ *fair, poor*	1, 3	1, 4	1, 5	1, 6	2, 3	2, 4	2, 5	2, 6	3, 4	3, 5	3, 6	4, 5	4, 6
Braking scenario–1	collisions?	no	no	no	no	yes	yes	yes	yes	no	yes	yes	no	yes
	colliding vehicles	–	–	–	–	V_1, V_3, V_6	V_5	V_2, V_4	V_1	–	V_3, V_5	V_2	–	V_1, V_4
Braking scenario–2	collisions?	yes	yes	yes	yes	yes	yes	no	no	no	yes	no	no	no
	colliding vehicles	V_2	V_1, V_2	V_1	V_1	V_5, V_6	V_5, V_6	–	–	–	V_5	–	–	–

TABLE VII: *Braking scenario 3*: No. of collision cases out of 5 simulation runs using the RTM module with NB, SB, and ESB strategies for various *fair* and *poor*-thresholds. ACC CTG = 0.4 s, CACC CTG = 0.3 s, PLATOON CDG = 5 m, soft-deceleration rate = –3 ms^{-2}, full-deceleration rate = –8 ms^{-2}, speed = 100 kmh^{-1}.

fair, poor	*1, 3*	*1, 4*	*1, 5*	*1, 6*	*2, 3*	*2, 4*	*2, 5*	*2, 6*	*3, 4*	*3, 5*	*3, 6*	*4, 5*	*4, 6*
ESB	0	0	0	0	0	0	0	1	0	0	0	0	0
SB	0	0	0	0	0	0	0	1	0	0	0	0	0
NB	3	3	2	2	4	4	3	3	4	4	4	4	4

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958



(a) RTM, SB

(b) RTM, ESB

Fig. 9: Acceleration (ms^{-2}) and distance (m) profiles using the RTM module with the SB and ESB strategies for the same configurations as in Table VII; the results of one simulation run (*fair* = 3, *poor* = 6) are presented.

controllers and/or adjust gaps according to the GDU method based on experienced communication quality and *fair*, *poor* - thresholds. Also in this case, the platoon performs emergency braking 70 s into the simulation time upon encountering an imaginary road hazard using either of the NB, SB, or ESB strategies (*braking scenario*). We simulate three different settings named *braking scenario–1*, *braking scenario–2*, and *braking scenario–3*. *Braking scenario–1* has PLATOON CDG = 5 m, CACC CTG = 0.5 s, and ACC CTG = 1.2 s, i.e., using the values as suggested in the literature. *Braking scenario–2* uses PLATOON CDG = 10 m, CACC CTG = 1.0 s, and ACC CTG = 1.2 s, i.e., less fuel-efficient inter-vehicle distances. Finally, *braking scenario–3* uses a more challenging configuration, e.g., PLATOON CDG = 5 m, CACC CTG = 0.3 s, and ACC CTG = 0.4 s. The full-deceleration rate is –8 ms^{-2}, the soft-deceleration rate is –3 ms^{-2} (for ESB only), and the speed is 100 kmh^{-1}.

Table VI summarizes the results of *braking scenario–1* and *braking scenario–2* for 13 combinations of *fair* and *poor* thresholds. It shows which combinations exhibit collisions together with the colliding vehicles. If we carefully look for a pattern in this table, it is evident that there are no collisions when the threshold *fair* = 1 is used in *braking scenario-1*, and as the *fair* threshold increases, more collision cases can be noticed. In contrast, when the *fair* threshold is small in *braking scenario-2*, the platooning vehicles undergo collisions. These results suggest that when an initial inter-vehicle distance as short as 5 m is used, the RTM must react to packet losses fast by increasing the distance to the front vehicle. To this end, a small value for *fair* threshold should be chosen, e.g., (1, 3), (1, 4), (1, 5), (1, 6) when the inter-vehicle distances are small. However, when the initial CDG is larger, e.g., 10 m, increasing the inter-vehicle distance too early further increases packet losses due to path loss and fading effects and eventually causes the adoption of the ACC controller. As the ACC controller does not perform well with a normal braking strategy unless the gap is sufficiently high, collisions may occur. There are two important observations that these results suggest. First, although *braking scenario–1* avoids collisions when using a *fair* threshold of 1, such a threshold is not suitable for string stability and fuel efficiency for the considered scenario, as discussed in Section VII-D. Second, normal braking exhibits poor performance in terms of collision avoidance despite using the GDU method. Moreover, the simulation results corresponding to *braking scenario–2* suggest that using longer inter-vehicle gaps does not necessarily ensure collision avoidance with normal braking when the platoon requires to decelerate stronger in a dense data and road traffic scenario. Note that the collision cases presented in Table VI happen during emergency braking, not while the platoon is cruising, i.e., the GDU method is still robust at avoiding collisions during cruising.

As *braking scenario–3* is more challenging compared to the other two in terms of inter-vehicle gaps, we use this scenario to put the proposed GDU method and ESB strategy to the test and also compare them with the other braking strategies, i.e., NB and SB. Table VII shows the number of collisions out of five simulation runs for 13 different combinations of *fair* and *poor* thresholds (total 65 simulation runs for each braking strategy). We can see that both the SB and ESB strategies can avoid collisions for 64 out of 65 simulation runs. However, with (*fair*, *poor*)-thresholds (2, 6), there is one collision case caused by the last vehicle in the platoon. This is due to a high DENM delay (1.6 s and 2.8 s in the ESB and SB strategies, respectively) and a high *poor* threshold that generates less state switching and results in inadequate inter-vehicle distances considering collision avoidance. Moreover, notice that the

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

TABLE VIII: Strong contracts representing the overall safety goals in Figures 1 and 2.

A_1	Maximum deceleration $-\ddot{x}_{i\_max} = -8\ ms^{-2}$ AND minimum deceleration $-\ddot{x}_{i\_min} = -3\ ms^{-2}$;
G_1	The deceleration rate is within [–3 ms^{-2}, –8 ms^{-2}];
A_2	–
G_2	(PLATOON, CACC, or ACC controller is active) $\longrightarrow$ The inter-vehicle gap $d_i > 0$ m both while the platoon is cruising AND at a complete standstill AND $d_L < d_{hazard}$ at full stop;

inter-vehicle distances used in *braking scenario–3* are shorter than what is recommended in the literature. These simulation results suggest that the selection of the *good*, *fair*, and *poor* thresholds should be adjusted to the deceleration capacity of the vehicles and the selected inter-vehicle distances in the different states, in order to always prioritize safety over fuel efficiency. The normal braking strategy exhibits many collision cases with *braking scenario–3* as seen in Table VII. However, these collision cases are not during platoon cruising but instead during emergency braking.

To better highlight the strategy of the ESB protocol, Figure 9 presents the acceleration and distance profiles with the SB and ESB strategies for one representative simulation run (*fair* = 3, *poor* = 6) as an example. The acceleration profiles for the ESB strategy show that vehicle 6 brakes at a full-deceleration rate as soon as it receives a DENM without waiting until the τ_{wait} period as per the ESB algorithm. The other vehicles perform soft-deceleration until the τ_{wait} time is reached and then brake at the full-deceleration rate simultaneously. As vehicle 6 brakes long before the other vehicles, it leaves a very high inter-vehicle gap at a complete standstill; see the ESB distance profiles in Figure 9b. Due to higher *fair*, *poor* thresholds, we do not see the vehicles adjusting the inter-vehicle gaps too frequently. In the SB case, all the vehicles receive the DENMs within τ_{wait} and brake at the full-deceleration rate simultaneously.

Please notice the frequent acceleration change between 50-70 s. This is because the RTM instructs the vehicles to accelerate or decelerate to attain the state and controller-specific gaps based on the experienced communication quality levels.

X. Assume/Guarantee Contracts

In Section IV-C, we discussed safety contracts that capture the operation modes of the system components in a degradation cascade. In this part, we derive a set of safety contracts based on the GDU method presented in Figure 2, the safety requirements, and the simulation results obtained above. Initially, the safety contracts suggested in [20], which are proposed based on domain knowledge, were taken as benchmarks. These were then refined and fine-tuned based on rigorous simulation studies of the controllers, as well as communications and vehicle kinematic parameters. First, a set of strong contracts are defined that represent the overall safety goal, Table VIII. The strong contracts $C^{strong} =< A, G >$ signify that the assumptions A_i shall always be met AND the guarantees G_i shall always hold. On the other hand, the weak contracts $C_i^{weak} =< B, H >$ imply that the guarantees H_i only require holding when the assumptions B_i are fulfilled, and the weak assumptions are not always required to hold [65]. For brevity, only the weak contracts related to the degradation cascade are presented in Table IX.

TABLE IX: A set of weak contracts in the PLATOON, CACC, and ACC modes.

B_{P1}	PLATOON controller active AND *c2l* and *c2f* are *good*;
H_{P1}	$gap_{des} \geq 5$ m AND the speed is 100 kmh^{-1};
B_{P2}	PLATOON controller active AND *c2f* is *good* AND *c2l* is *fair*;
H_{P2}	Retain PLATOON controller AND issue the increase of CDG within 10 ms;
B_{P3}	PLATOON controller active AND *c2f* is *good* AND *c2l* is *poor*;
H_{P3}	Issue the transition to the CACC controller within 10 ms;
B_{P4}	PLATOON controller active AND *c2l* and *c2f* are *good* for all the vehicles AND $gap_{des} = 5$ m;
H_{P4}	the platoon exhibits high level of fuel efficiency and string stability;
B_{CACC1}	CACC controller active AND *c2f* is *good* AND *c2l* is *poor*;
H_{CACC1}	time gap $T \geq 0.3$ s AND the speed is 100 kmh^{-1};
B_{CACC2}	CACC controller active AND *c2f* is *fair*;
H_{CACC2}	Retain CACC controller AND issue the increase of CTG within 10 ms;
B_{CACC3}	CACC controller active AND *c2f* is *poor*;
H_{CACC3}	Issue the transition to the ACC controller within 10 ms;
B_{CACC4}	CACC controller active AND *c2f* is *good* and *c2l* is *poor* for all the vehicles AND time gap $T = 0.5$ s;
H_{CACC4}	the vehicle string exhibits string stability;
B_{ACC1}	ACC controller active AND no onboard sensor failure;
H_{ACC1}	time gap $T \geq 0.4$ s AND the speed is 100 kmh^{-1};

The weak assumptions present input conditions in the PLATOON, CACC, and ACC modes, and the guarantees address the system component behaviors, which also represent the safety requirements. Based on the communication quality with the vehicle in front or the LV, a vehicle can degrade its performance by either increasing the gap to the vehicle in front and/or switching to a more suitable controller. The act of increasing the gap first as a response to *fair* communication quality with the lead vehicle is regarded as *graceful degradation*. Note that we suggest some numbers, such as 5 m CDG for the PLATOON controller, 0.3 s CTG for the CACC controller, and 0.4 s CTG for the ACC controller, etc., while defining the guarantees in Tables VIII and IX. These are not randomly chosen but instead obtained from extensive simulation studies (see the simulation results in Section IX). However, these assume/guarantee pairs do not necessarily mean that such constant distance gaps or time gaps cannot be used unless the specified assumptions are fulfilled [65]. These contracts rather represent the fact that the component behaviors (Guarantees) are known, given that the assumptions are satisfied.

XI. Discussions

In this section, we provide an analysis of the control algorithms and our proposed approaches, e.g., the GDU method

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

and the ESB strategy, based on the simulation parameters and the obtained simulation results. The analysis is made in terms of platoon safety, fuel efficiency, string stability, and the ability to track the LV. To evaluate the robustness of our proposed approaches, we selected two challenging simulation scenarios, one for cruising and one for braking, both of which use high speed (100 kmh^{-1}) and shorter inter-vehicle gaps than what is recommended in the literature. Moreover, the simulations are performed under a dense data and road traffic scenario with 400 additional non-platooning vehicles, which contribute to high communication delays.

When the state-of-the-art controllers, e.g., PLATOON [25], CACC [42], or ACC [39], are used independently, the platooning vehicles either lack safety, string stability, fuel efficiency, or LV tracking ability. More specifically, the tail vehicles in the platoon using the PLATOON controller undergo collisions due to transient communication outages with the LV and the short inter-vehicle gaps (5 m). However, the PLATOON controller enables high fuel efficiency, string stability, and LV tracking. Note that the PLATOON controller does not necessarily exhibit safe behavior if longer gaps are used, e.g., 10 m, because the rear vehicles experience even more communication outages as they are further away from the LV. In addition, string stability and fuel efficiency become worse with longer gaps when using the PLATOON controller. On the other hand, the CACC controller exhibits moderate string stability but high tracking error with the LV and the vehicles are less fuel-efficient due to the requirement of using longer gaps (15.89 m). The ACC controller is even less fuel-efficient and exhibits tracking error when maintaining 35.35 m gaps at 100 kmh^{-1}. The proposed GDU method ameliorates the overall performance of the platoon by degrading the fuel efficiency and string stability of only a subset of the vehicles, the rear vehicles in the platoon, when these experience communication outages and would cause collisions in case the PLATOON controller was used. Using the GDU method, these rear vehicles either increase the gap to the vehicle in front or adopts the CACC controller to provide a sufficient level of safety. However, the front vehicles inherit the fuel efficiency, string stability, and LV tracking ability of the PLATOON controller while maintaining safety. Moreover, the performance degradation of the rear vehicles is temporary, just like the temporary communication outage; the GDU method adopts the PLATOON controller again or reduces the gap when the communication quality improves. Our simulation results show that when higher values for the *fair*, *poor* thresholds are used in the GDU method, the vehicles demonstrate better string stability, fuel efficiency, and LV tracking ability. On the other hand, lower *fair*, *poor* thresholds cause too frequent state switching, which aids safety but worsens fuel efficiency and string stability. However, the state machine and the autonomous transitions using the GDU method can avoid collisions *during cruising* for all simulations conducted in this paper for all choices of the *fair*, *poor* thresholds. Therefore, the GDU method is very robust in maintaining platoon safety, which is the primary concern. Moreover, the GDU method uses the best of the PLATOON, CACC, and ACC controllers to provide a balanced trade-off between safety, fuel efficiency, string stability, and LV tracking.

The ESB strategy shows good performance in attaining the fail-safe state both in terms of avoiding collisions and stopping the platoon fast. In 64 out of 65 simulation runs, the platoon avoids collisions during emergency braking when using GDU and ESB together. The collision experienced in one simulation run occurs due to a high communication delay coupled with a high selected value of the *poor* threshold (six consecutive packet losses). There are 44 collision cases out of 65 simulation runs with the normal braking strategy despite using the GDU method. Note that these collisions happen during emergency braking, not while the platoon is cruising. Therefore, the normal braking strategy is unsuitable for emergency braking in a challenging scenario, whereas the SB and ESB strategies are efficient at collision avoidance. In addition to collision avoidance, the proposed ESB strategy exhibits 12.84 meters shorter stopping distance of the LV than its predecessor SB strategy.

Finally, the safety contracts derived from the simulation results concerning the GDU method suggest some quantitative performance targets on the inter-vehicle distances while cruising with the PLATOON, CACC, and ACC controllers. The vehicles can maintain gaps, e.g., 5, 10.33, and 13.11 meters with PLATOON, CACC, and ACC controllers, respectively, while cruising at a speed of 100 kmh^{-1}, given the switching conditions between the states in GDU method are known.

XII. Conclusions and Future Work

In this paper, we propose a strategy for classifying the transient communication outages in vehicle platooning into states in a state machine that captures different platooning modes and performance levels as a function of the communication quality levels. In order to keep a platoon fail-operational, a Graceful Degradation and Upgradation (GDU) method has also been proposed that regulates the transitions between different cruising states during transient connectivity problems. Instead of the traditional way of classifying wireless connectivity as successful or failed, the GDU method considers *good*, *fair*, and *poor* communication qualities with the LV and the vehicle in front to facilitate the transitions between the states in the state machine. We have performed a detailed analysis of how the *fair*, *poor* communication thresholds can be selected and how they can be used to keep a platoon fail-operational in terms of safety while facilitating a sufficient level of fuel efficiency, string stability, and LV tracking. Moreover, an emergency braking strategy named Enhanced Synchronized Braking (ESB) is proposed and evaluated, aiming to facilitate the transition of the platooning vehicles from any of the cruising states to a fail-safe state even during challenging communication scenarios. Last but not least, we derive a set of safety contracts that capture the operation modes of the GDU method.

The rigorous simulation studies we have conducted demonstrate that the GDU method can keep a platoon fail-operational in the presence of transient connectivity problems and that the ESB strategy can avoid collisions and reduce the stopping distance of the platoon also under dense data and road traffic

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

scenarios. The best performance in terms of fuel efficiency, string stability, and safety is achieved when the ESB strategy and the GDU method are combined with insightfully selected values of the communication thresholds, which are adapted to the CAM rate and the deceleration capabilities of the vehicles. Hence, the suggested state machine can enable automated platooning while ensuring fault tolerance during transient connectivity problems.

In this paper, we analyze the effects of *fair*, *poor* thresholds on safety, string stability, fuel efficiency, and LV tracking ability and provide some guidelines on the choice of the thresholds in both dense and sparse traffic scenarios. Exactly how these thresholds should be adjusted for different inter-vehicle gaps with different controllers is left for future investigation. In particular, since the communication outage can be caused artificially in situations when DCC algorithms, which cause a reduction in the CAM rate, are mandatory. Furthermore, it would be worth investigating the performance of the GDU method by assigning different *fair*, *poor* thresholds to different vehicles based on needs and distances to the vehicle in front and the LV. Moreover, how the *fair*, *poor* thresholds can be adapted to the changes in the vehicles' mobility parameters, e.g., acceleration change, is worth investigating. In addition, theoretical studies on how the platoon members functioning in a distributed manner with different types of CACC controllers and different gap policies should pursue a global stabilizing condition require research attention from a control theory perspective. Furthermore, it could be beneficial to relay the packets from the LV when the platoon is long. Moreover, an in-depth comparative analysis of the state-of-the-art emergency braking strategies in terms of fail-safe conditions is required.

REFERENCES

[1] 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, 2015.

[2] J. Axelsson, "Safety in vehicle platooning: A systematic literature review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1033–1045, 2016.

[3] J. C. Knight, "Safety critical systems: challenges and directions," in *Proc. of the 24th International Conference on Software Engineering. ICSE 2002*, pp. 547–550, 2002.

[4] T. Stolte, S. Ackermann, R. Graubohm, I. Jatzkowski, B. Klamann, H. Winner, and M. Maurer, "A taxonomy to unify fault tolerance regimes for automotive systems: Defining fail-operational, fail-degraded, and fail-safe," *IEEE Transactions on Intelligent Vehicles*, pp. 1–1, 2021.

[5] S. Hasan, M. A. Al Ahad, I. Sljivo, A. Balador, S. Girs, and E. Lisova, "A fault-tolerant controller manager for platooning simulation," in *Proc. 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*, pp. 1–6, Graz, Austria, 2019.

[6] S. Girs, I. Sljivo, and O. Jaradat, "Contract-based assurance for wireless cooperative functions of vehicular systems," in *Proc. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*, pp. 8391–8396, 2017.

[7] R. Wanhill, L. Molent, and S. Barter, "Milestone case histories in aircraft structural integrity," in *Reference Module in Materials Science and Materials Engineering*, Elsevier, 2016.

[8] C. Nowakowski, S. E. Shladover, X.-Y. Lu, D. Thompson, and A. Kailas, "Cooperative Adaptive Cruise Control (CACC) for Truck Platooning: Operational Concept Alternatives," *PATH research report*, 2015.

[9] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cooperative adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 488–497, 2015.

[10] D. K. Murthy and A. Masrur, "Exploiting space buffers for emergency braking in highly efficient platoons," in *Proc. 2017 IEEE 23rd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 1–10, Hsinchu, Taiwan, 2017.

[11] C. Bergenhem, K. Meinke, and F. Ström, "Quantitative safety analysis of a coordinated emergency brake protocol for vehicle platoons," in *Proc. International Symposium on Leveraging Applications of Formal Methods*, pp. 386–404, Springer, 2018.

[12] W. Yu, X. Hua, and W. Wang, "Investigating the longitudinal impact of cooperative adaptive cruise control vehicle degradation under communication interruption," *IEEE Intelligent Transportation Systems Magazine*, pp. 2–20, 2021.

[13] F. Miekautsch, F. Seeland, J. Horn, and A. Fay, "Situation-aware switching of the communication topology in heterogeneous platooning applications," *IFAC-PapersOnLine*, vol. 54, no. 2, pp. 306–313, 2021. 16th IFAC Symposium on Control in Transportation Systems CTS 2021.

[14] R. Zheng, K. Nakano, S. Yamabe, M. Aki, H. Nakamura, and Y. Suda, "Study on emergency-avoidance braking for the automatic platooning of trucks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 4, pp. 1748–1757, 2014.

[15] N. An, J. Mittag, and H. Hartenstein, "Designing fail-safe and traffic efficient 802.11p-based rear-end collision avoidance," in *Proc. IEEE Vehicular Networking Conference*, Paderborn, Germany, Dec 2014, pp. 9-16.

[16] M. Segata, B. Bloessl, S. Joerer, C. Sommer, M. Gerla, R. Lo Cigno, and F. Dressler, "Toward communication strategies for platooning: Simulative and experimental evaluation," *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 5411–5423, Dec 2015.

[17] V. R. S. Yellapantula, K. B. Devika, and S. C. Subramanian, "Communication latency and speed-dependent minimum time headway for connected heavy road vehicle collision avoidance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4739–4748, 2020.

[18] D. K. Murthy and A. Masrur, "A cyber-physical approach for emergency braking in close-distance driving arrangements," *ACM Trans. Cyber-Phys. Syst.*, Mar 2022.

[19] S. Hasan, A. Balador, S. Girs, and E. Uhlemann, "Towards emergency braking as a fail-safe state in platooning: A simulative approach," in *Proc. 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, pp. 1–5, Honolulu, HI, USA, 2019.

[20] I. Sljivo, B. Gallina, and B. Kaiser, "Assuring degradation cascades of car platoons via contracts," in *Proc. 6th International Workshop on Next Generation of System Assurance Approaches for Safety-Critical Systems*, Magdeburg, Germany, Sept. 2017, pp 317–329.

[21] Y. Li, C. Tang, K. Li, S. Peeta, X. He, and Y. Wang, "Nonlinear finite-time consensus-based connected vehicle platoon control under fixed and switching communication topologies," *Transportation Research Part C: Emerging Technologies*, vol. 93, pp. 525–543, 2018.

[22] K. Li, Y. Bian, S. E. Li, B. Xu, and J. Wang, "Distributed model predictive control of multi-vehicle systems with switching communication topologies," *Transportation Research Part C: Emerging Technologies*, vol. 118, p. 102717, 2020.

[23] F. Gao, F.-x. Lin, and B. Liu, "Distributed H∞ control of platoon interacted by switching and undirected topology," *International Journal of Automotive Technology*, vol. 21, no. 1, pp. 259–268, 2020.

[24] S. E. Shladover, C. Nowakowski, X.-Y. Lu, and R. Ferlis, "Cooperative adaptive cruise control: Definitions and operating concepts," *Transportation Research Record*, vol. 2489, no. 1, pp. 145–152, 2015.

[25] R. Rajamani, *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[26] M. Segata, *Safe and Efficient Communication Protocols for Platooning Control*. PhD Thesis, University of Innsbruck, Feb 2016.

[27] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic set of applications; part 2: Specification of cooperative awareness basic service," ETSI EN 302 637-2 V1.3.1, Nov 2014.

[28] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; part 3: Specification of decentralized environmental notification basic service," ETSI EN 302 637-3 V1.2.1, Nov 2014.

[29] E. Uhlemann, "Communication requirements of emerging cooperative driving systems," in *Proc. 2011 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 281–282, IEEE, 2011.

[30] E. van Nunen, J. Ploeg, A. M. Medina, and H. Nijmeijer, "Fault tolerancy in cooperative adaptive cruise control," in *Proc. 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pp. 1184–1189, IEEE, 2013.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

[31] B. Kaiser, B. Monajemi, D. Kusche, and H. Schulte, "Systematic design and validation of degradation cascades for safety-relevant systems," in *Proc. The 2nd International Conference on Engineering Sciences and Technologies*, pp. 527–527, 06 2017.

[32] E. van Nunen, D. Tzempetzis, G. Koudijs, H. Nijmeijer, and M. van den Brand, "Towards a safety mechanism for platooning," in *Proc. 2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 502–507, IEEE, 2016.

[33] M. Segata, R. Lo Cigno, T. Hardes, J. Heinovski, M. Schettler, B. Bloessl, C. Sommer, and F. Dressler, "Multi-technology cooperative driving: An analysis based on plexe," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2022.

[34] K. Bilstrup, E. Uhlemann, E. Ström, and U. Bilstrup, "On the ability of the 802.11 p mac method and stdma to support real-time vehicle-to-vehicle communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2009, no. 1, p. 902414, 2009.

[35] W. Yu, X. Hua, and W. Wang, "Investigating the longitudinal impact of cooperative adaptive cruise control vehicle degradation under communication interruption," *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 4, pp. 183–201, 2022.

[36] H. Chehardoli and M. R. Homaeinezhad, "Stable control of a heterogeneous platoon of vehicles with switched interaction topology, time-varying communication delay and lag of actuator," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 231, no. 22, pp. 4197–4208, 2017.

[37] T. Alkim, H. Schuurman, and C. Tampere, "Effects of external cruise control and co-operative following on highways: an analysis with the mixic traffic simulation model," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No.00TH8511)*, pp. 474–479, 2000.

[38] S. Hasan, "Fail-Operational and Fail-Safe Vehicle Platooning in the Presence of Transient Communication Errors," tech. rep., Mälardalen University, Västerås, Sweden, March 2022.

[39] P. Ioannou and C. Chien, "Autonomous intelligent cruise control," *IEEE Transactions on Vehicular Technology*, vol. 42, no. 4, pp. 657–672, 1993.

[40] S. E. Shladover, C. Nowakowski, and X.-Y. Lu, "Using cooperative adaptive cruise control (cacc) to form high-performance vehicle streams. definitions, literature review and operational concept alternatives," in *UC Berkeley: California Partners for Advanced Transportation Technology*, 2014.

[41] S. Santini, A. Salvi, A. S. Valente, A. Pescapé, M. Segata, and R. L. Cigno, "A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 3, pp. 1985–1999, 2016.

[42] J. Ploeg, B. T. M. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Proc. 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Washington, DC, USA, October 5-7, 2011, pp 260–265.

[43] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, 2014.

[44] A. Ali, G. Garcia, and P. Martinet, "The flatbed platoon towing model for safe and dense platooning on highways," *IEEE Intelligent Transportation Systems Magazine*, vol. 7, no. 1, pp. 58–68, 2015.

[45] G. Lee, S. Kim, Y. Yim, J. Jung, S. Oh, and B. Kim, "Longitudinal and lateral control system development for a platoon of vehicles," in *Proceedings 199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, pp. 605–610, 1999.

[46] Xiangheng Liu, A. Goldsmith, S. S. Mahal, and J. K. Hedrick, "Effects of communication delay on string stability in vehicle platoons," in *Proc. ITSC 2001. 2001 IEEE Intelligent Transportation Systems (Cat. No.01TH8585)*, pp. 625–630, 2001.

[47] P. Fernandes and U. Nunes, "Platooning with IVC-enabled autonomous vehicles: Strategies to mitigate communication delays, improve safety and traffic flow," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 91–106, 2012.

[48] 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, 2000.

[49] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *Proc. 2014 IEEE Vehicular Networking Conference (VNC)*, Paderborn, Germany, December 2014, pp. 53–60.

[50] P. Pop, D. Scholle, H. Hansson, G. Widforss, and M. Rosqvist, "The safecop ecsel project: Safe cooperating cyber-physical systems using wireless communication," in *Proc. 2016 Euromicro Conference on Digital System Design (DSD)*, pp. 532–538, 2016.

[51] A. Benveniste, B. Caillaud, D. Nickovic, R. Passerone, J.-B. Raclet, P. Reinkemeier, A. Sangiovanni-Vincentelli, W. Damm, T. Henzinger, and K. G. Larsen, "Contracts for System Design," *Research Report RR-8147, INRIA*, Nov. 2012.

[52] S. Magdici and M. Althoff, "Adaptive cruise control with safety guarantees for autonomous vehicles," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5774–5781, 2017. 20th IFAC World Congress.

[53] J. Ligthart, E. Semsar-Kazerooni, J. Ploeg, M. Alirezaei, and H. Nijmeijer, "Controller design for cooperative driving with guaranteed safe behavior," in *Proc. 2018 IEEE Conference on Control Technology and Applications (CCTA)*, pp. 1460–1465, 2018.

[54] R. Zheng, K. Nakano, S. Yamabe, and Y. Suda, "Safety evaluation of system failures in formation and separation processes of automatic platooning of trucks," in *Proc. 20th ITS World Congress, ITS Japan*, 2013.

[55] D. K. Murthy and A. Masrur, "Braking in close following platoons: The law of the weakest," in *Proc. 2016 Euromicro Conference on Digital System Design (DSD)*, pp. 613–620, 2016.

[56] J. Thunberg, N. Lyamin, K. Sjöberg, and A. Vinel, "Vehicle-to-Vehicle Communications for Platooning: Safety Analysis," *IEEE Networking Letters*, vol. 1, no. 4, pp. 168–172, 2019.

[57] G. Sidorenko, J. Thunberg, K. Sjöberg, and A. Vinel, "Vehicle-to-Vehicle Communication for Safe and Fuel-Efficient Platooning," in *Proc. 2020 IEEE Intelligent Vehicles Symposium (IV)*, pp. 795–802, 2020.

[58] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved IVC analysis," *IEEE Transaction Mobile Computing*, vol. 10, pp. 3–15, Jan 2011.

[59] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo–simulation of urban mobility: an overview," in *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*, 2011.

[60] A. Wegener, M. Piórkowski, M. Raya, H. Hellbrück, S. Fischer, and J.-P. Hubaux, "TraCI: an interface for coupling road traffic and network simulators," in *Proc. the 11th communications and networking simulation symposium*, pp. 155–163, 2008.

[61] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, and P. Mudalige, "Mobile vehicle-to-vehicle narrow-band channel measurement and characterization of the 5.9 ghz dedicated short range communication (dsrc) frequency band," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1501–1516, 2007.

[62] ETSI, "Intelligent Transport Systems (ITS); ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band," ETSI EN 302 663 V1.3.1, Oct 2019.

[63] L. Zhang, F. Chen, X. Ma, and X. Pan, "Fuel economy in truck platooning: a literature overview and directions for future research," *Journal of Advanced Transportation*, vol. 2020, 2020.

[64] G. Bansal, J. B. Kenney, and C. E. Rohrs, "Limeric: A linear adaptive message rate algorithm for dsrc congestion control," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 9, pp. 4182–4197, 2013.

[65] I. Sljivo, B. Gallina, J. Carlson, and H. Hansson, "Strong and weak contract formalism for third-party component reuse," in *Proc. 2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*, pp. 359–364, 2013.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

This article has been accepted for publication in IEEE Open Journal of Intelligent Transportation Systems. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/OJITS.2023.3237958

Shahriar Hasan (M'19) received the BSc degree in Computer Science and Engineering from Islamic University of Technology, Dhaka, Bangladesh, in 2013 and the joint Master's degree in Internet Technologies and Architecture from Sorbonne Université, Paris, France and University of Trento, Italy, in 2017. He is currently pursuing the PhD degree in Computer Science and Engineering at Mälardalen University, Västerås, Sweden.

From 2013 to 2015, he was a Software Engineer with the Samsung R&D Institute Bangladesh Ltd., Dhaka, Bangladesh. His research interest includes connected vehicles, wireless vehicular communications, fault tolerance methods in safety-critical systems, software-defined networking, network function virtualization, 5G radio access networks, and network simulation and modelling.

Svetlana Girs (M'12) is a senior lecturer at the School of Innovation, Design and Engineering at Mälardalen University, Västerås, Sweden, where she is currently leading the Networked and Embedded Systems Division. Svetlana holds B.Sc. and M.Sc. degrees in Telecommunications from Saint-Petersburg State Polytechnic University, Russia from 2009 and 2011 respectively. In February 2016 Svetlana received her Ph.D. in Computer Science and Engineering from Mälardalen University. Svetlana has been a visiting researcher at University of Canterbury, Christchurch, New Zealand for three months in 2014. Svetlana's research interests include real-time communication, cooperative relay networks and reliable wireless communication for industrial systems, vehicular communication. Svetlana is also a Co-chair of the Subcommittee on Industrial Communication Systems within the IEEE IES Technical Committee on Factory Automation.

Elisabeth Uhlemann received her PhD degree in Communications Theory from Chalmers University of Technology, Gothenburg, Sweden, in 2004. She has held visiting positions at the University of South Australia in 2005, the Technical University of Berlin in 2007 and the University of Canterbury, New Zealand in 2011. She has also worked as a consultant at Volvo Technology during 2005-2009 dealing with connected vehicles, as a consultant at Ikanos Communications, CA, USA, in 2005 with VDSL protocols and at Free2move, Sweden, during 2009-2010 with wireless audio. Currently, she is a full professor in Data Communications at Malardalen University, Sweden. She has been involved in several European projects studying communication requirements for traffic-safety applications in vehicular networks. She also contributed to the European ITS communications architecture and has served as a technical expert in standardization within ETSI TC ITS. She currently serves as senior editor for IEEE VT Magazine on Connected and Automated Vehicles and has served as a member of different PhD examination committees over 20 times in five different countries. She is a senior member of the IEEE and has served as vice chair of the Swedish VT/COM/IT chapter for several years.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/