A Fault-Tolerant Controller Manager for Platooning Simulation

Shahriar Hasan[1], Muhammed Abdullah Al Ahad[1], Irfan Sljivo[1], Ali Balador[1,2],
Svetlana Girs[1], Elena Lisova[1]
[1]Mälardalen University, Sweden,
{firstname.lastname}@mdh.se, muhammed.abdullah.al.ahad@mdh.se,
[2]RISE SICS Västerås, Sweden

Abstract— **Recent development in wireless technology enabling communication between vehicles led to introduction of the concept of Cooperative Adaptive Cruise Control (CACC), which uses wireless vehicle-to-vehicle communication and aims at string stable behavior in a platoon of vehicles. Degradation cascades have been proposed as a way to maintain a certain level of the system functionality in presence of failures. Such degradation behaviour is usually controlled by a runtime/state manager that performs fault detection and transitions the system into states where it will remain acceptably safe. In this paper, we propose a dynamic controller manager that focuses on both safety and performance of the system. In particular, it monitors the channel quality within the platoon and reacts by degrading platoon performance in presence of communication failures, or upgrading the performance when the communication quality is high enough. The reaction can include, e.g., adjusting the inter-vehicle distance or switching to another suitable platoon controller to prevent collisions. We focus on the functional and operational safety and evaluate the performance of the dynamic controller manager under different scenarios and settings in simulation experiments to demonstrate that it can avoid rear-end collisions in a platoon, continue platooning operation even in dense traffic scenarios where the state-of-the-art controllers fail to do so.**

I. Introduction

A common phenomenon on congested roads is the formation of traffic shock waves, i.e., high-density waves traveling backwards with respect to the cruising direction. Traffic shock waves cause sudden emergency braking that can lead to chain collisions if the vehicles do not respect the required safety distance from preceding vehicles. It was shown before that usage of the concept of adaptive cruise control (ACC) has a positive impact on traffic safety and efficiency [1]. ACC relies on radar and/or Lidar sensor measurements in order to allow a vehicle to follow its predecessor by adapting the speed and inter-vehicle distance.

However, ACC does not sufficiently improve the string stability of vehicles, and requires maintenance of high inter-vehicle distances to achieve safety. A string of vehicles is called "string stable" when it can attenuate the propagation of any non-zero position, speed, and acceleration errors of an individual vehicle in the upstream direction [2]. Due to significant recent advances in computation, communication, and control technologies, connected and automated vehicles are becoming a reality. The limitations of sensor-based systems like ACC ameliorated the necessity of Cooperative Adaptive Cruise Control (CACC) which is a fusion of sensor measurements and wireless communications that has been proven to be able to close the inter-vehicle gap by sharing the vehicle parameters through inter-vehicle communication. The communicated information may include vehicle position, speed, acceleration, steering angle etc. With CACC and V2V communication it is possible to realize safer platooning, i.e., allow a group of vehicles to rally behind a lead vehicle with short inter-vehicle distances with the goal of minimizing fuel consumption, operating cost and enhancing road safety.

One of the main challenges in platooning application is to deal with the transient errors caused by the unreliable wireless communication [3]. In platooning, as the vehicles maintain short inter-vehicle distances, a temporary failure of V2V communication can cause a much more hazardous situation than sensor failures in an ACC system [4]. This is because the vehicles will collide long before the backup drivers can act as the typical reaction time of a driver is much higher than the required time headway in CACC systems [5], [6]. Degradation cascades have been previously proposed as a way to keep the platoon acceptably safe by switching to a lower mode in case of failures, e.g., switching from CACC to ACC when communication to the vehicle in front fails. The state machine capturing such mode switching can be specified in terms of assumption-guarantee safety contracts that realise the corresponding safety requirements [7]. Runtime Manager Concept (RMC) is proposed as a way to implement those contracts, and assure the related requirements [8]. According to this concept, the runtime manager monitors the parameters used within the contracts, evaluates the contract assumptions, and reacts accordingly with the behaviour specified in the contract guarantees.

To this end, we instantiate RMC as a *dynamic controller manager* that monitors the packet losses within the platoon, and looks for safety contract violation in which case, it adjusts the inter-vehicle distance, or switches to another suitable platoon controller to prevent collisions. The contributions of this work can be summarized as follows:

- We propose a dynamic controller manager as an instantiation of RMC that can monitor the communication status in a platoon during run-time, and switch between the controllers by both upgrading and degrading the performance to avoid rear-end collisions while the vehicles are still forming a platoon. The logic behind the controller switching or distance adjustment has also been presented in this paper.

- We have performed simulation studies to demonstrate that the proposed dynamic controller manager can avoid rear-end collisions in a platoon, and continue platooning operation even in dense traffic scenarios while the state-of-the-art controllers fail to do so.

The rest of the paper is organized as follows: Section II outlines the background and the related works. Section III contains the system model of the proposed dynamic controller manager. The results of our simulation studies are presented in Section IV. Finally, Section V concludes the paper.

II. Background and Related Work

The design of efficient Fault Detection, Isolation, and Recovery (FDIR) system is indispensable to ensure the safety, and smooth functionality of ACC and CACC applications. The authors in [9] proposed a *Safety Checker* algorithm which is closely related to our work. This graceful degradation algorithm takes inaccuracies in acceleration, velocity and the distance to the front vehicle as input, and dynamically adjusts inter-vehicle distance in order to maintain a safe gap. However, only the results of sensor failure have been presented in this paper. A dynamic spacing policy upon V2V failure was proposed in [4] with the aim of maintaining operational and functional safety of a truck platoon. In [8], the authors applied safety analysis and risk assessment methods to determine what might go wrong in the communication between vehicles, and proposed a contract-based safety assurance method based on that. The dynamic controller manager proposed in this paper detects communication quality degradation, and thus takes proactive measures in order to maintain fail-operational state in the presence of a predicted communication failure. As a functional safety mechanism, a platooning vehicle switches between the controllers, and/or adjusts inter-vehicle gap.

In order to facilitate switching between the controllers, we take three popular controllers from the literature as reference. The ACC model is taken from Chapter 6 of [10] proposed by Rajamani. In this controller, the desired acceleration of a vehicle is determined by taking the time headway, relative speed, and distance error into consideration. However, the required distance to the preceding vehicle grows proportionally in this model with the speed to avoid collisions. In [11], the authors proposed a CACC model in which the kinematic parameters of the leader, and that of the preceding vehicle are communicated through V2V communication to the ego vehicle. The consideration of damping ratio and controller bandwidth along with the distance error and relative speed in its design, allows this controller to maintain an inter-vehicle distance as short as 5 meters in ideal scenarios. Ploeg *et. al* proposed a *one-vehicle look-ahead* CACC controller [12] in which a vehicle receives packets from its preceding vehicle only. The design of this controller emphasizes on maintaining string stability, and takes the communication delay in V2V communication into consideration. Test results show that a platoon of length six can exhibit string stable behavior for a time headway of 0.67 s in ideal scenarios.

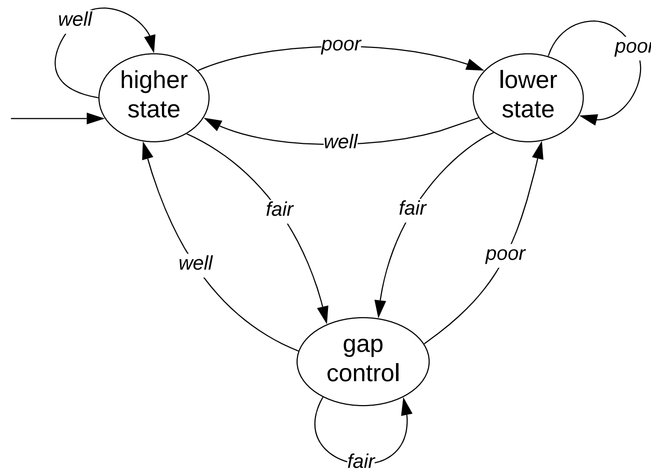


Fig. 1: Dynamic controller manager state machine.

The dynamic controller manager presented in this work manages the switching between the above mentioned controllers to prevent rear-end collisions in the platoon. From this point onwards, these controllers will be termed as ACC [10], CACC [11] and PLOEG [12]. These names have been kept consistent with the names used in the Plexe [13] simulator that was used to implement and evaluate the performance of the dynamic controller manager.

III. Dynamic Controller Manager

A. System Model

The proposed *dynamic controller manager* handles switching between controllers at run-time based on the packet losses induced by channel conditions, and interference from non-platooning vehicles. Consider a scenario in which a platoon is cruising at a constant speed of 100 kmh^{-1} with an inter-vehicle distance of 5 metres in accordance with the CACC control law. A few consecutive packet losses can cause rear-end collisions in the platoon [8]. The controller manager dynamically monitors the communication link between a particular vehicle with its predecessor, and the leader, logs the packet losses, and looks for safety contract violation. In case of such a violation, the controller manager takes proactive measures by either increasing the inter-vehicle distance, or switching to a lower state, e.g., ACC. The *connection to the front (c2f)* vehicle and the *connection to the leader (c2l)* are categorized into three categories, namely *well*, *fair* and *poor*. If a platooning vehicle does not experience more than a predefined number of packet losses with respect to its predecessor and the leader, then the connection status is set to *well*. The connection is considered *fair* or *poor* in case of higher number of packet losses. The boundary between *fair* and *poor* depends on a scenario under consideration and should be carefully chosen based on the simulation results. We have presented the results for different combinations of *fair* and *poor* values in Section IV. In order to explain the idea of dynamic controller manager, we simplify the conditions for controller switching. In practice, this is more complex and will be explained later in this section.

The finite state machine of dynamic controller manager as shown in Fig. 1, can be represented by 5-tuple $(Q, \sum, \delta, q_0, F)$, where Q is the finite set of states, $\sum$ is the set of input alphabets, δ is the transition function, q_0 is the initial state, and F is the set of final states such that $F \subseteq Q$. The set Q consists of *higher state*, *lower state*, and *gap control*. CACC and PLOEG controllers are considered to be the higher states because they can maintain a comparatively short inter-vehicle distance and string stability. ACC controller, on the other hand, is considered to be the lower state because of its high time headway requirement. Gap control signifies inter-vehicle distance adjustment without switching to the lower or higher state. In fact, we only switch to the lower state if the connection to the front vehicle is *poor*. If the connection quality is *fair* then the inter-vehicle distance is increased to avoid any potential accident. The controller manager keeps monitoring the *c2f* and *c2l*, and switches back to the higher state, or minimizes the gap if the connection becomes *well* with time. In the proposed dynamic controller manager, platooning always starts with CACC controller, a higher state, and can then switch to any of the lower or gap control states depending on the connection.

The transition state tables for controller switching in dynamic controller manager are depicted in Table I. The rules of transition were determined based on the control law of ACC, PLOEG and CACC. Please recall, in ACC there is no V2V communication and the distance to the preceding vehicle is maintained based on radar measurements. In case of PLOEG controller, only the preceding vehicle communicates its kinematic parameters through IEEE 802.11p Cooperative Awareness Messages (CAMs). A vehicle using CACC controller requires connection to both the front and the leading vehicle. So, when a platooning vehicle is at a higher state and its connection to the front vehicle is *poor*, it switches to the ACC controller in order to avoid collision with its predecessor as shown in Table I (a). However, if the connection to the leader is *poor*, a vehicle switches from CACC to PLOEG as the PLOEG controller does not require connection to the lead vehicle. While using PLOEG, no matter what the connection quality with respect to the leader is, a vehicle increases the gap with its predecessor if the connection quality to the front vehicle deteriorates to *fair*, Table I (b). The conditions for switching to a higher state are presented in Table I (c). If the connection to both front and the leader becomes *well* with time, then the vehicle switches to CACC controller. For *poor* connection to the leader and *well* connection to the front vehicle, it switches to PLOEG controller. As the CACC controller uses a short inter-vehicle distance, due to *fair* connection with the vehicle in the front, the controller manager switches to PLOEG controller and at the same time increases the gap to the front vehicle, Table I (d). The PLOEG controller is considered a lower state in this case, as the required time headway for this controller is much higher than that of CACC controller. Similarly, for a connection quality of *fair* with the predecessor, the vehicle switches to PLOEG but increases the gap, while due to *well* connection quality to the predecessor and *fair* connection to the leader, the state is changed to CACC in combination with increased gap to the front vehicle. In both cases, the gap is increased by a user defined factor to minimize too frequent state switching. The necessary conditions for switching to a higher state with gap control are depicted in Table I (e).

IV. Simulations and Performance Evaluation

We have carried out simulation studies to analyze the performance of the dynamic controller manager. To this end, we use the Plexe [13] simulator that is built as an extension of VANET simulation framework Veins [14] for realistic simulation of platooning applications. Veins is an OMNeT++ [15] based simulator that provides a complete IEEE 802.11p protocol stack along with different physical layer channel models, and realistic mobility for its nodes with the aid of the road traffic simulator SUMO [16]. In Veins, a SUMO vehicle is replicated by an OMNeT++ node, and they communicate through TraCI interface, a TCP-based client/server model. The communication protocols and application logics necessary for platooning application are implemented in the OMNeT++ part of Plexe. On the other hand, different car-following models proposed in the state-of-the-art are implemented in the SUMO part. The dynamic controller manager has been implemented as a separate module in Veins. It is designed to perform three major tasks: logging, monitoring, and taking actions based on monitored state of *c2f* and *c2l* of the controlled vehicle. The *onPlatoonBeacon* method implemented in the dynamic controller manager acts as a bridge between Plexe and dynamic controller manager, and is invoked every time a platooning vehicle receives a beacon.

A. Simulation Settings

A platoon of length eight has been simulated in presence of 50 non-platooning vehicles. The non-platooning vehicles generate interference and increase contention to create a dense traffic scenario. Most of the communication and controller parameters have been kept the same as the default Plexe simulator, as listed in Table II, and these parameters comply with the IEEE 802.11p standard. The path loss model used here is Free Space Path Loss (FSPL) model, according to which the received signal strength is inversely proportional to the square of the distance. This means that the tail vehicles, which require packets from the leader to operate in a platoon using the CACC controller, are likely to experience more packet losses if the platoon is long. We have simulated the sinusoidal scenario of Plexe in which the leader accelerates and decelerates in a sinusoidal fashion with an oscillation frequency of 0.2 Hz. The configuration parameters for dynamic controller manager are also listed in Table II. The packet loss is monitored every 100 ms. In the gap control state, the inter-vehicle distance is increased by the factor of 0.25 for both CACC and PLOEG controller in case of *fair* connection quality. *fair* and *poor* connections are dependent on the number of consecutive packet losses, and these are defined by $nPacketLossFair$ and $nPacketLossPoor$ parameters respectively in our implementation. We have simulated different

TABLE I: Transition state tables for dynamic controller switching in run-time.

(a) Switching to a lower state.

Initial state	c2f	c2l	Final state
PLOEG	*poor*	*well*	ACC
PLOEG	*poor*	*fair*	ACC
PLOEG	*poor*	*poor*	ACC
CACC	*well*	*poor*	PLOEG
CACC	*poor*	*well*	ACC
CACC	*poor*	*fair*	ACC
CACC	*poor*	*poor*	ACC

(b) Gap control.

Initial state	c2f	c2l	dist2Pred
PLOEG	*well*	*poor*	*Default*
PLOEG	*fair*	*well*	*Increase*
PLOEG	*fair*	*fair*	*Increase*
PLOEG	*fair*	*poor*	*Increase*
CACC	*well*	*well*	*Default*
CACC	*well*	*fair*	*Increase*

(c) Switching to a higher state.

Initial state	c2f	c2l	Final state
ACC	*well*	*poor*	PLOEG
ACC	*well*	*well*	CACC
PLOEG	*well*	*well*	CACC
CACC	*well*	*well*	CACC

(d) Switching to a lower state and gap control.

Initial state	c2f	c2l	Final state	dist2Pred
CACC	*fair*	*well*	PLOEG	*Increase*
CACC	*fair*	*fair*	PLOEG	*Increase*
CACC	*fair*	*poor*	PLOEG	*Increase*

(e) Switching to a higher state and gap control.

Initial state	c2f	c2l	Final state	dist2Pred
ACC	*fair*	*well*	PLOEG	*Increase*
ACC	*fair*	*fair*	PLOEG	*Increase*
ACC	*fair*	*poor*	PLOEG	*Increase*
ACC	*well*	*fair*	CACC	*Increase*
PLOEG	*well*	*fair*	CACC	*Increase*

TABLE II: Simulation and analysis configuration parameters.

Category	Parameters	Value	Parameters	Value
Communication	Tx power	100 mW	Sensitivity	- 94 dBm
	Thermal noise	- 102 dBm	Path loss model	Free space (α = 2.0)
	Frequency	5.89 GHz	Bit rate	6 Mbps
	Packet size	200 B	Bit rate (non-platooning vehicle)	3 Mbps
Mobility	Leader speed	100 kmh^{-1} (27.778 ms^{-1})	No. of non-platooning vehicles	50
	Platoon size	8	Leader oscillation freq.	0.2 Hz
Controller	Controller name	CACC, PLOEG, ACC	Headway distance	3 ~ 20 m
Dynamic controller manager	Monitor interval	0.1 s	Expected beacon interval	0.105 s
	CACC spacing factor	0.25	PLOEG time headway factor	0.25

combinations of $nPacketLossFair$ and $nPacketLossPoor$ values. However, in this paper, we only demonstrate the results for $nPacketLossFair = 2, nPacketLossPoor > 2$ and $nPacketLossFair = 3, nPacketLossPoor > 3$, i.e., more than 2 and 3 packet losses are considered as *poor* connection in the first and second combination respectively.

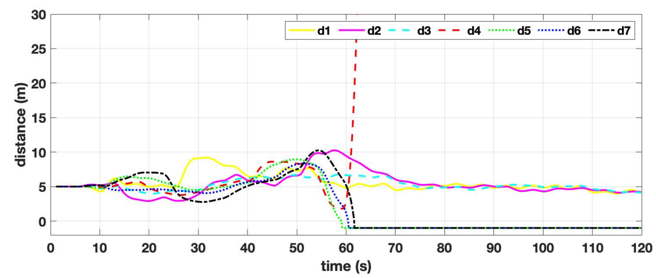
B. Results And Analysis

In this section, we analyze the inter-vehicle distance profiles of platooning vehicles to demonstrate the benefits of the dynamic controller manager over a single controller like CACC or PLOEG without additional safety measures. The inter-vehicle distance between two platooning vehicles is calculated as:

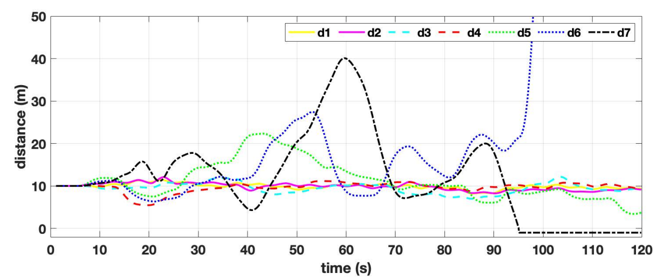
$$d_i(t) = x_{i-1}(t) - x_i(t) - l_{i-1}, \quad (1)$$

where, x_i and x_{i-1} are the positions of i^{th} vehicle and its predecessor $i-1$ respectively, and l_{i-1} is the length of the vehicle in the front. In case of a collision with the vehicle in front, $d_i(t)$ is considered to be equal to zero.

We have simulated the CACC controller [11] implemented in Plexe with the simulation settings listed in Table II for 120 seconds with initial inter-vehicle distances of 5 and 10 metres. The resulting distance profiles are depicted in Fig. 2, where, d_1 is the inter-vehicle distance between the lead vehicle, V_0, and the second vehicle, V_1. For CACC spacing of 5 meters, at the 58^{th} second of simulation time, vehicle 5 hits the vehicle 4 from behind and brings it into a complete standstill, Fig. 2a. As a result, vehicles 6 and 7 also undergo rear-end collisions. Vehicles 0-3, on the other hand, keep cruising, and



(a) Initial inter-vehicle distance = 5 m

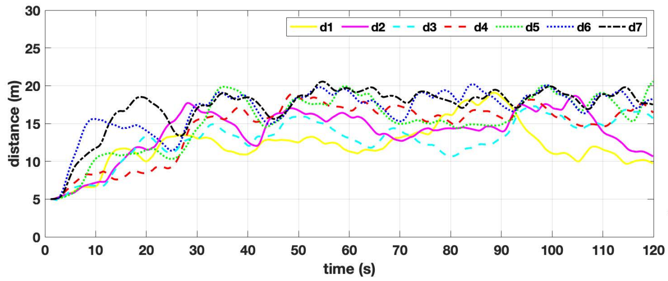


(b) Initial inter-vehicle distance = 10 m

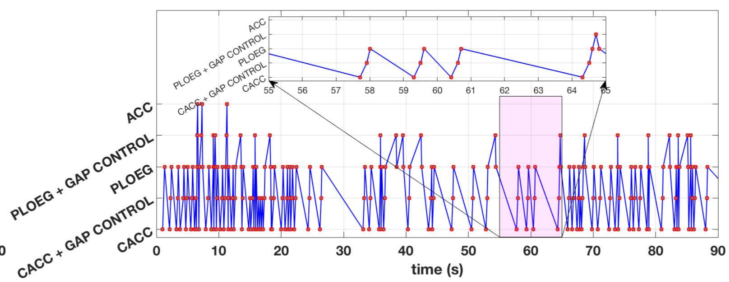
Fig. 2: Distance profiles, $d_i(t)$ of vehicles in CACC platoon mode with no dynamic controller manager in play.

this increases the distance between vehicles 3 and 4 sharply (the red curve, d4). As the non-colliding vehicles (0-3) move away from the rest, this platoon of three vehicles becomes more string stable and can maintain the default 5 meters gap. This is because all the vehicles are close to the leader and receive the beacons successfully. Similar behavior can be observed for initial inter-vehicle distance of 10 meters for CACC controller as shown in Fig. 2b. As the vehicles 6 and 7 are far away from the leader, they experience more packet losses due to path loss effect. At around 95^{th} second of simulation time, vehicle 7 hits the vehicle 6 from behind. The rest of the platooning vehicles keep moving and maintain the default 10 meters gap, exhibit string stable behavior. The dynamic controller manager proposed in this paper detects the packet losses that cause these collisions and tries to apply the most suitable cruising method in order to avoid them.

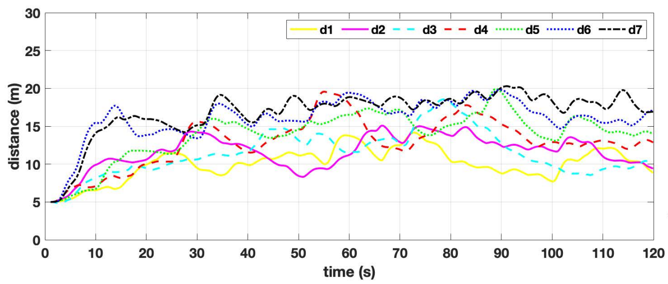
The same scenario as in Fig. 2a has been simulated with



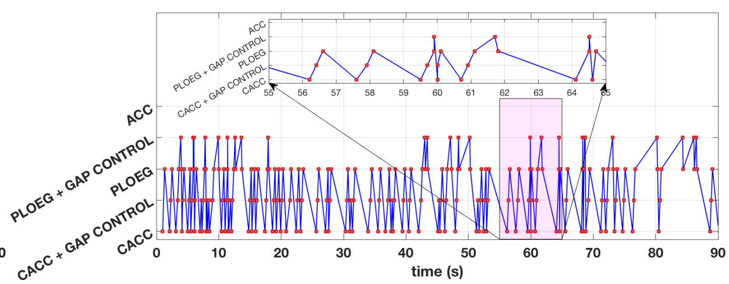
(a) $nPacketLossFair = 2, nPacketLossPoor > 2$.



(c) $nPacketLossFair = 2, nPacketLossPoor > 2$.

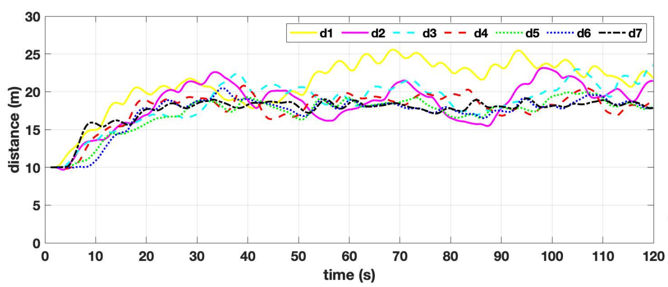


(b) $nPacketLossFair = 3, nPacketLossPoor > 3$.

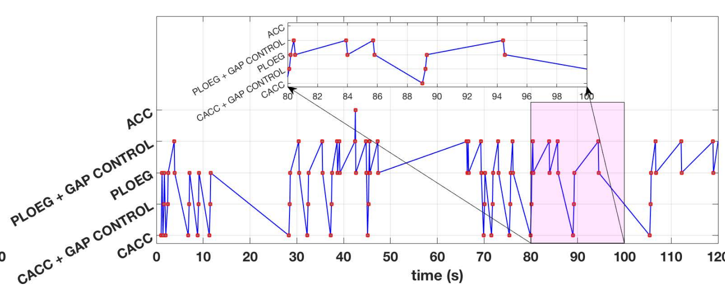


(d) $nPacketLossFair = 3, nPacketLossPoor > 3$.

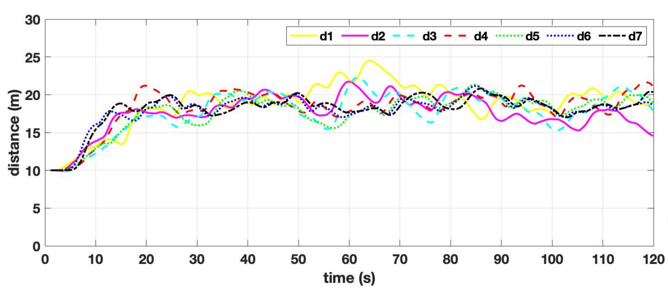
Fig. 3: Left column represents the inter-vehicle distance profile of the platoon, and the right column represents the state of dynamic controller manager for Vehicle 5; ACC spacing = 27.78m, PLOEG spacing = 16.67m, CACC spacing = 5m.



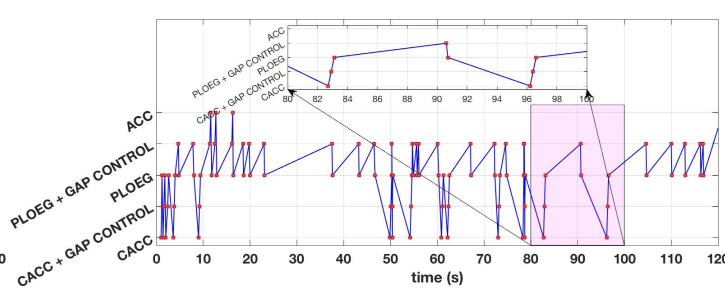
(a) $nPacketLossFair = 2, nPacketLossPoor > 2$.



(c) $nPacketLossFair = 2, nPacketLossPoor > 2$.



(b) $nPacketLossFair = 3, nPacketLossPoor > 3$.



(d) $nPacketLossFair = 3, nPacketLossPoor > 3$.

Fig. 4: Left column represents the inter-vehicle distance profile of the platoon, and the right column represents the state of dynamic controller manager for Vehicle 7; ACC spacing = 27.78m, PLOEG spacing = 16.67m, CACC spacing = 10m.

dynamic controller manager which always starts with the CACC controller with initial inter-vehicle spacing of 5 meters. The distance profiles, $d_i(t)$ of the platooning vehicles with $nPacketLossFair$ values of 2 and 3 are depicted in Figures 3a and 3b. It is clear from the figures that the dynamic controller manager avoids the collision at the 58^{th} second that occurred in Fig. 2a. In order to explain the proactive measures taken by the controller manager to avoid the collision that is caused by vehicle 5, we present the state profiles of the vehicle 5 in Figures 3c and 3d, and zoom in the window between 55-65 s. In Fig. 3c, just before the 58^{th} second, the vehicle is using CACC controller, and due to 2 packet losses it increases the gap to the front vehicle. In the meantime, dynamic controller manager notices 3 packet losses, which is considered as *poor* connection, and switches to PLOEG controller at the 58^{th} second. In Fig. 3d, as the $nPacketLossFair$ threshold is set to 3, due to 3 packet losses the vehicle just increases the gap but still can avoid collision. As the vehicle is using

PLOEG controller now, and the connection to the lead vehicle gets better, it switches back to CACC controller again. This way, vehicle 5 hovers between PLOEG and CACC controllers to both minimize the gap and avoid collision. The dynamic controller manager results corresponding to Fig. 2b with initial CACC spacing of 10 meters are depicted in Fig. 4. It is apparent from the figure that the vehicle 7, which caused collision at the 95^{th} second in Fig. 2b, avoids collision in this case. During that time, the vehicle switches to PLOEG controller due to *poor* connection to the leader, as shown in Fig. 4c. For $nPacketLossFair$ threshold 3, vehicle 7 switches to CACC controller due to *well* connection to both the front and the lead vehicles. Although the sequence of state changing is different for different $nPacketLossFair$ values, in both cases vehicle 7 can avoid the collision by switching to PLOEG controller in time.

Fig. 4 uses a CACC spacing of 10 meters whereas, it is 5 meters in case of Fig. 3. A careful look at these figures shows that the scenarios represented by Fig. 4 has fewer state switching than the scenarios in Fig. 3. This is because of the initial CACC spacing. Whenever vehicle 7 in Fig. 4c or 4d switches to CACC controller, the connection to the leader becomes bad due to path loss effect, and it switches to PLOEG controller in the immediate next monitor interval. This is why most of the time, the vehicles in Fig. 4 use PLOEG controller as it does not require connection to the leader, and exhibits a consistent inter-vehicle spacing in comparison to the vehicles in Fig. 3. Due to shorter CACC spacing (5 meters), the vehicles in the front of the platoon in Fig. 3 are closer to the leader while using CACC controller. If the connection to the leader becomes bad, they switch to PLOEG or CACC + Gap control state. Thus, these vehicles alternate between CACC and PLOEG controller quite frequently and maintain a shorter inter-vehicle distance on an average throughout the simulation time. This is good from a fuel efficiency point of view. However, from a safety and string stability point of view, less state switching and thus consistent inter-vehicle distance like in Fig. 4 is better. Moreover, Figures 3c, 3d, 4c, 4d exhibit an aggressive increase in inter-vehicle distance from the very beginning of the simulation. This is because the non-platooning vehicles generate interference throughout the simulation time causing packet losses. The dynamic controller manager successfully detects these and takes necessary safety measures the whole time.

V. Conclusions

In this paper, we propose a dynamic controller manager as an instantiation of Runtime Manager concept and simulate its performance by means of the Plexe simulator. The concept was also extended by considering both upgradation and degradation of platoon performance. Simulation results show that there are collisions in the platoon while the vehicles are moving with inter-vehicle distances of 5 and 10 meters due to packet losses in dense traffic scenarios. However, for the same simulation settings, the platooning vehicles can avoid the collisions by degrading platoon performance transiently using the dynamic controller manager. Moreover, it has been observed that better string stability can be achieved by increasing the packet loss threshold for controller switching. This also depends on the inter-vehicle gap and the communication strategy of the used controllers. Regarding future study, we will investigate the effectiveness of the dynamic controller manager in emergency braking scenarios, and take more realistic channel models into consideration.

Acknowledgment

The work in this paper was funded from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 764951, KKS via the ELECTRA project, SSF via project Future Factories in the Cloud (FiC), and the SafeCOP project which is funded from the ECSEL Joint Undertaking under grant agreement n^0 692529 and from National funding.

References

[1] I. R. Wilmink, G. A. Klunder, and B. van Arem, "Traffic flow effects of integrated full-range speed assistance (irsa)," in *Proc. IEEE IV*, Istanbul, Turkey, June 2007, pp. 1204–1210.

[2] A. Bose and P. Ioannou, "Analysis of traffic flow with mixed manual and intelligent cruise control (icc) vehicles: Theory and experiments. california path report," tech. rep., UCB-ITS-PRR-2001-13, 2001.

[3] N. An, J. Mittag, and H. Hartenstein, "Designing fail-safe and traffic efficient 802.11 p-based rear-end collision avoidance," in *Proc. IEEE VNC*, Paderborn, Germany, December 2014, pp. 9–16.

[4] E. Nunen, R. Koch, L. Elshof, and B. Krosse, "Sensor safety for the european truck platooning challenge," in *Proc. ITS World Congress*, Melbourne, Australia, Oct 2016.

[5] P. Lemmen, H. Fagerlind, T. Unselt, C. Rodarius, E. Infantes, and C. van der Zweep, "Assessment of integrated vehicle safety systems for improved vehicle safety," *Procedia-Social and Behavioral Sciences*, vol. 48, pp. 1632–1641, 2012.

[6] X. Ma and I. Andréasson, "Estimation of driver reaction time from car-following data: Application in evaluation of general motor–type model," *Transportation research record*, vol. 1965, no. 1, pp. 130–141, 2006.

[7] I. Sljivo, B. Gallina, and B. Kaiser, "Assuring degradation cascades of car platoons via contracts," in *proc. SASSUR*, Magdeburg, Germany, Sept 2017, pp. 317–329.

[8] S. Girs, I. Sljivo, and O. Jaradat, "Contract-based assurance for wireless cooperative functions of vehicular systems," in *Proc. IECON*, Beijing, China, Oct 2017, pp. 8391–8396.

[9] E. van Nunen, J. Ploeg, A. M. Medina, and H. Nijmeijer, "Fault tolerancy in cooperative adaptive cruise control," in *Proc. IEEE ITSC*, The Hague, Netherlands, Oct 2013, pp. 1184–1189.

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

[11] 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 Trans Control Sys. Tech.*, vol. 8, no. 4, pp. 695–708, 2000.

[12] J. Ploeg, B. T. M. Scheepers, E. Nunen, N. Van De Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," *in Proc. IEEE ITSC*, Washington, DC, USA, Oct 2011, pp. 260–265.

[13] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. Lo Cigno, "PLEXE: A Platooning Extension for Veins," in *Proc. IEEE VNC*, Paderborn, Germany, Dec 2014, pp. 53-60.

[14] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Trans Mob. Comp.*, vol. 10, no. 1, pp. 3–15, 2011.

[15] A. Varga, "The omnet++ discrete event simulation system," *Proc. ESM'2001*, vol. 9, 01 2001.

[16] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "Sumo (simulation of urban mobility)-an open-source traffic simulation," in *Proc. MESM20002*, pp. 183–187, 2002.