Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Case Study on Communication Based Cooperative Longitudinal Vehicle Guidance

Frank Schrödel * **Rick Voßwinkel** * **Robert Ritschel** *
Maximilian Gerwien * **Erik Gruschka** * **Norman Schwarz** *
Alexander Jungmann *

* *IAV GmbH, Chemnitz, Germany*
{*frank.schroedel, rick.vosswinkel, robert.ritschel, maximilian.gerwien, erik.gruschka, norman01.schwarz, alexander.jungmann*}*@iav.de*

Abstract: An approach of a proven concept for cooperative longitudinal vehicle guidance is presented. The paper briefly describes the communication pipeline as well as the incorporated communication technology. Moreover, the used longitudinal controller, which explicitly considers the communicated states of the vehicles ahead, is addressed. Here, we are expanding a classical Adaptive Cruise Controller (ACC) to realize the cooperative longitudinal guidance functionality simply and robustly. On the one hand, the current states of the vehicle in front are transmitted and on the other hand the desired ones. Results of a real-world test campaign of platooning vehicles illustrate the effectiveness of the proposed approach.

Keywords: Adaptive Cruise Control, Automated Guided Vehicles, Automobile Industry, Communication Applications, Communication Systems

1. INTRODUCTION

Connected automated vehicles have been drawing great attention from both academia and industry. Since cooperative longitudinal guidance is the basis of all platooning strategies. This offers a bunch of environmental, social and economic benefits. Such benefits are that fuel consumption is decreased and the improvement of the traffic flow. Another critical issue in modern traffic are accidents which are mainly arising from human errors. Cooperative guidance enables automatic breaking with very small reaction times coupled with permanent attention. Additionally, are connected vehicles more predictable for other road users which increases safety again.

Due to the great importance of that topic, numerous approaches for automated and connected driving had been proposed. An overview of different software architectures to realize automated vehicle guidance functionality can be found e.g in Ulbrich et al. (2015); Taş et al. (2016). Elementary parts of such a guidance system are the environment perception, decision making, and movement planning as well as the vehicle dynamic control. Information from sensors about the vehicle and its environment have to be processed first. Then, decision making and path planning functions can plan paths for different maneuvers. Finally, the actors perform based on lateral and longitudinal controllers the planned steering angle and longitudinal acceleration. In the present paper, the focus is on this acting layer, more precisely on the longitudinal control.

General longitudinal control approaches can be found e.g. in Rajamani (2012). Classical ACC controller architectures, including a fuzzy controller, are presented in Benalie et al. (2009); Ko and Lee (2007); Schrödel et al. (2019). More sophisticated approaches like model predictive control based ACC are presented in Stanger and del Re (2013); Corona and De Schutter (2008). A reinforcement learning-based ACC approach can be found in Desjardins and Chaib-draa (2011). Since the goal design of this study is a robust and easily tunable controller structure with low demands on computing power, such that advanced control strategies are omitted. Furthermore, first concepts of cooperative ACC functionality based on vehicle to vehicle communication are presented e.g. in Stanger and del Re (2013); Desjardins and Chaib-draa (2011); Esbensen et al. (2007); Lidstrom and Sjoberg (2012).

A study of synchronization and consensus effects is neglected in the current study. For such analyses please refer to the literature of Multi-Agent Systems (MAS). In general, the MAS information exchange aim is to reach a consensus, i.e. an agreement on certain values of interest that depend on the system states (Scardovi and Sepulchre, 2009). An analogous problem is the synchronization (see e.g. Li et al. (2010)), where the agents are demanded to converge to a common trajectory via feedback control (Trentelman et al., 2013). Ensuring synchronization for MAS is either investigated with a focus on designing proper controllers with a known communication structure (Olfati-Saber et al., 2007), or with a focus on finding proper communication structures using general properties of the communication network (Hermann et al., 2018). The classical and already established ACC control structure presented in Schrödel et al. (2019) were the starting point of the current approach. An easy integrable and upgradeable solution for already existing ACC systems is the focus of the paper. Therefore, we designed a quite easy to tune and modular function architecture, to realize good maintainability. Finally, the proposed approach is easy to tune and shows robust results during real-time test drives.

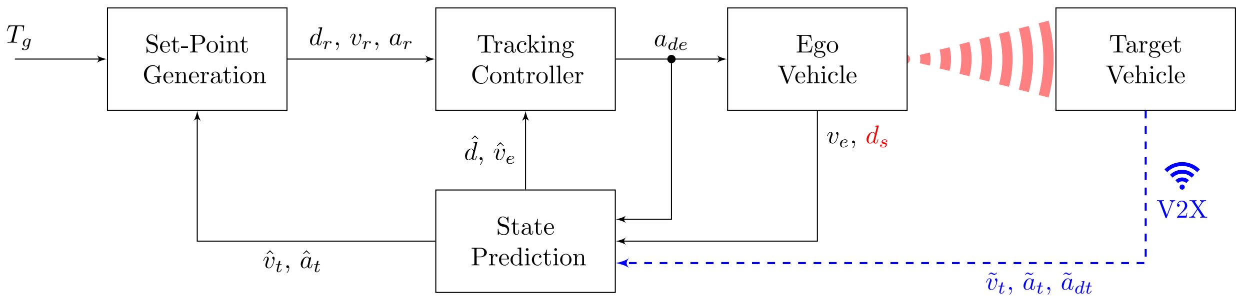

Fig. 1. Architecture for Longitudinal Control

For that purpose, the article is organized as follows. In the next section, the incorporated communication technique is presented. Based on this the longitudinal control approach is discussed in section 3. In section 4 the proposed concept is validated, by utilizing real-world measurement data from test drives, using automated prototypes of IAV GmbH.

2. V2X COMMUNICATION

Vehicle-to-Everything (V2X) communication is one major building block of our cooperative longitudinal vehicle guidance approach. Different direct communication technologies such as 802.11p and Cellular Vehicle-to-Everything (C-V2X) were subject to extensive discussions for the last years (cf. Festag (2014)). For the work at hand, we chose the standardized 802.11p approach, which is an adaption of 802.11a. It supports high vehicle speeds, works without any initialization procedures, and is based on broadcasting. Comparing the different communication technologies concerning communication latency in our real-world scenario is an important step in our future work, though. For more details regarding the overall communication infrastructure, please refer to Auerswald et al. (2019).

Applications implemented in today's available communication On-Board Units (OBUs) are mainly focusing on day 1 and day 1.5 use cases (cf. C-ITS Platform (2016)). Respective messages for implementing those applications are, e.g., Cooperative Awareness Message (CAM), Decentralized Environmental Notification Message (DENM), and Collective Perception Message (CPM). However, none of the existing message types currently fully supports our cooperative longitudinal vehicle guidance application. That is, none of the existing messages can transport the information required for our implementation. As a consequence, we decided to use 802.11p as the underlying communication technology but designed a proprietary message

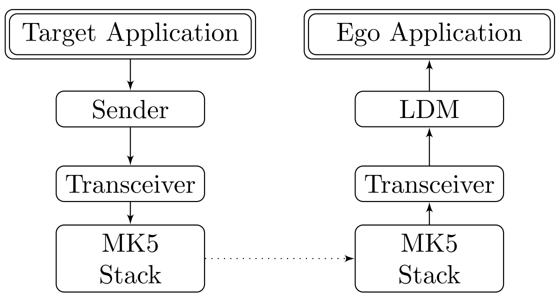

Fig. 2. V2X Communication Architecture

type. Thus, we can easily change the communicated information and identify the restrictions given by the currently standardized message types.

Fig. 2 briefly depicts the entire communication pipeline. The target application can be any automated driving function incorporating longitudinal control. Taken from the target vehicle interface, the required target vehicle data such as current or desired acceleration is sent via the transceiver to the actual communication hardware. For sending and receiving messages in our real-world tests, we use for each vehicle a Cohda wireless MK5 OBU combined with two Dedicated Short Range Communications (DSRC) antennas. After receiving the message on the physical layer and before processing the contained information in the ego application, the message is stored in our Local Dynamic Map (LDM) service. Roughly speaking, our LDM is responsible for maintaining and synchronizing all information received via communication.

In our current setting, however, we completely neglect more intelligent data association and decision-making mechanisms. That is, we simply assume that the data broadcasted by the target vehicle is always relevant for the ego vehicle. Embedding our cooperative longitudinal vehicle guidance approach into a comprehensive automated driving system that deals with more complex traffic situations and not only a single (pre-defined) target vehicle will require more high-level logic in the future.

3. ACC APPROACH

A meshed control structure is utilized to realize the Cooperative Adaptive Cruise Control (CACC) functionality, see Fig. 1. Here, the CACC system can be decomposed into three main components:

- Tracking Controller: Realizes a good distance and velocity tracking behavior
- Set-Point Generation: Generation of reference values for the underlying tracking controller
- State Prediction: Prediction of the state values of the ego and the target vehicle

As shown in Fig. 1, input of the set-point generation is the desired time-gap T_g as well as the predicted velocity $\hat{v}_t$ and acceleration $\hat{a}_t$ of the target vehicle. Based on this values, the reference distance d_r, velocity v_r, and acceleration a_r is generated for the tracking controller. Moreover, the tracking controller receives the predicted distance $\hat{d}$ and the predicted velocity $\hat{v}_e$ of the ego vehicle from the state prediction. Note that the output of the

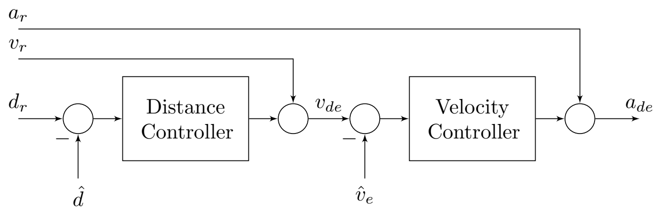

Fig. 3. Tracking Controller

state prediction varies depending on the approach used to implement the ACC functionality, see Section 3.4 for more details. The output value of the tracking controller is the desired acceleration of the ego vehicle a_{de}. This acceleration set-point is transmitted via vehicle CAN to the underlying acceleration controller, which computes the corespondent engine and brake torques. The resulting velocity of the ego vehicle is denoted with v_e.

For an intuitive presentation, Fig. 1 includes the target vehicle as well. The distance d_s to the target vehicle is measured with a radar sensor. In addition, the target vehicle states v_t, a_t and a_{dt} are transmitted to the ego vehicle via V2X communication, as described in Section 3.4. Note that this is the special aspect of the CACC approach.

3.1 Tracking Controller

We use a tracking controller based on the classical ACC architecture, presented e.g. in Rajamani (2012); Lidstrom and Sjoberg (2012); Schrödel et al. (2019). Thereby, the CACC is realized in a cascade structure, as presented in Fig. 3. This cascading allows the system to be divided into smaller subsystems. These subsystems are easier to handle, which increases control accuracy.

The outer loop controls the distance to the target vehicle by using a PD distance controller. It determines a delta velocity set-point, which is added to the reference velocity v_r to calculate the desired velocity v_{de} for the underlying velocity controller. The input signal of the distance controller is the difference between the reference distance d_r and the predicted distance $\hat{d}$. Further, an inner control loop, including a P controller, is determining the desired acceleration set-points of the ego vehicle a_{de}. Note that a_r and v_r are feed-forward signals.

To increase driving comfort and take the non-linear influence of vehicle speed on driving dynamic into account, both control loops were designed as gain scheduling controllers. Here, the controller gain is adjusted according to the current vehicle states, like the vehicle speed. Rear-end collisions in platoons are most often the result of slow and under-damped system reactions. Inevitably, it leads to larger distances between the platoon vehicles. To avoid large distances between the vehicles a proper controller tuning is necessary.

3.2 Set-Point Generation

The purpose of the set-point generation is to create set-point values for the underlying tracking controller, which are the reference distance to the vehicle in front d_r, the reference velocity v_r and the reference acceleration a_r.

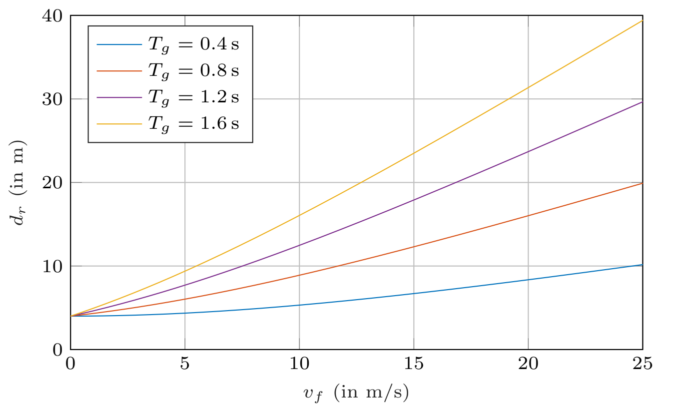

Fig. 4. Reference distance d_r as a function of the filtered velocity v_f for different time-gaps T_g

Input data for this calculations are predicted vehicle states and the desired time-gap T_g, see Fig. 1.

To describe the ACC reference distance d_r we use an adapted generalized logistic function (Richards, 1959)

$$d_r(x) = k_0 + \frac{T_g v_f - k_0}{1 + e^{-k_1(v_f - k_2)}} \tag{1a}$$

with

$$k_0 = d_0\,(1 + e^{-k_1 k_2}) \tag{1b}$$

$$k_1 = \frac{2\pi}{v_{ub} - v_{lb}} \tag{1c}$$

$$k_2 = v_{ub} - \frac{\pi}{k_1} = \frac{v_{ub} + v_{lb}}{2}, \tag{1d}$$

where $x = [T_g, v_f]^\top$ is the input vector, which consists of time-gap T_g and the filtered velocity v_f. The parameters of this function are the minimum distance $d_o > 0$ as well as the upper velocity bound v_{ub} and the lower velocity bound $v_{lb} < v_{ub}$. Note that $d_r(x)$ equals d_0 for $v_f = 0$ and approaches $T_g v_f$ for increasing v_f. There is a smooth transition between these both extremes, whose beginning and end is defined by v_{ub} and v_{lb}. The aim of the Definition (1a) is that $d_r(x)$ depends only on T_g and v_f for higher velocities but has a minimum d_0 for $v_f = 0$.

The parametrization of the Function (1a) used for our experiments is visualized in Fig. 4. Here, the influence of the filtered vehicle velocity v_f and the desired time-gap T_g on the reference distance d_r can be seen. As expected, the reference distance increases by increasing vehicle velocity and increasing time-gap. Moreover, it can be seen that the reference distance converges to the minimum distance $d_o = 4\,\mathrm{m}$ if $v_f \to 0$.

Please note, a second order dynamic

$$G_f(s) = \frac{v_f}{\hat{v}_t} = \frac{1}{T_f^2 s + 2T_f\zeta_f s + 1}, \tag{2}$$

with the time constant T_f and the damping ratio ζ_f is utilized to create the filtered velocity signal v_f based on the predicted vehicle velocity $\hat{v}_t$ of the target. This is necessary to reduce the effect of the sensor noise. Additionally, it is used to smooth the generated reference distance signal $d_r(t)$ in order to adjust the tracking behavior.

For the calculation of the relative velocity we use

$$\dot{d}_r = J(x)\dot{x} \qquad \text{with} \qquad J(x) = \frac{\partial d_r(x)}{\partial x}. \tag{3}$$

Moreover, for the relative acceleration follows

$$\ddot{d}_r = J(x)\ddot{x} + \dot{J}(x,\dot{x})\dot{x} \quad \text{with} \quad \dot{J}(x,\dot{x}) = \frac{\partial J(x)}{\partial x}\dot{x}. \tag{4}$$

Finally, the set-points for the underlying tracking controller result in the subsequent relations for the reference velocity

$$v_r = \hat{v}_t - \dot{d}_r \tag{5}$$

and the reference acceleration

$$a_r = \hat{a}_t - \ddot{d}_r, \tag{6}$$

where $\hat{v}_t$ and $\hat{a}_t$ represent the predicted values of the target vehicle. The usage of these reference values simplifies the task of distance and speed control dramatically. They are used as feed-forward signals. Consequently, the tracking controller is only needed to realize a proper disturbance reaction behavior.

3.3 State Prediction

The state prediction function aims to forecast the future state values of the ego vehicle and the target vehicle used by the tracking controller and the set-point generation functionality. The objective is to compensate sensor and communication time as well as vehicle actuator delays. For this purpose, the actuator dynamic are modeled as time delay systems

$$a_e(t) = a_{de}(t - T_e), \qquad a_t(t) = a_{dt}(t - T_t), \tag{7}$$

where T_e and T_t represent the input delays of the ego vehicle and the target vehicle, respectively. The desired prediction time is denoted as T_p. We use the following three assumptions for the state prediction:

- The acceleration of the target vehicle will change only at low frequencies so that the modeling of the vehicle dynamic as time delay systems (7) is sufficient.
- All system delays (due to V2X-communication, computational times, etc.) are static.
- For the desired prediction time T_p, $T_p \leq T_e$ applies.

To predict the ego velocity we use

$$\hat{v}_e(t + T_p) = v_e(t) + \int_{t-T_p}^{t} \hat{a}_e(\tau + T_p)\,\mathrm{d}\tau, \tag{8}$$

where

$$\hat{a}_e(t + T_p) = a_{de}(t + T_p - T_e) \tag{9}$$

is the predicted acceleration of the ego vehicle. Furthermore, the predicted target velocity results in

$$\hat{v}_t(t + T_p) = \tilde{v}_t(t) + \int_{t-(T_p+T_c)}^{t} \hat{a}_t(\tau + T_p)\,\mathrm{d}\tau, \tag{10}$$

where

$$\hat{a}_t(\tau + T_p) = \begin{cases} \tilde{a}_{dt}(\tau + T_p - \Delta T) & \tau \leq t - (T_p - \Delta T) \\ \tilde{a}_{dt}(t) & \text{otherwise} \end{cases} \tag{11}$$

is the predicted acceleration and $\Delta T = T_t - T_c$. Here, $\tilde{v}_t(t) = v_t(t - T_c)$ and $\tilde{a}_{dt}(t) = a_{dt}(t - T_c)$ are the target velocity and the target acceleration, which are received via V2X. The time delay of the V2X communication is denoted with T_c. Note that the integration in (10) starts at $t - (T_p + T_c)$ to consider the communication delay in the prediction. Please also note that in (11) we assume that the acceleration $\tilde{a}_{dt}(t)$ will retain the last known value for future times.

Accordingly, the predicted relative velocity is calculated using the relation

$$\Delta\hat{v}(t + T_p) = \hat{v}_t(t + T_p) - \hat{v}_e(t + T_p). \tag{12}$$

Based on this, the distance to the target vehicle can be predicted to

$$\hat{d}(t + T_p) = d_s(t) + \int_{t-(T_p+T_s)}^{t} \Delta\hat{v}(\tau + T_p)\,\mathrm{d}\tau, \tag{13}$$

where $d_s = d(t - T_s)$ is the sensed distance, which is measured by a sensor that has the delay T_s.

3.4 ACC Expansion Stages

In the following, three different approaches for the realization of the ACC functionality are considered. These approaches are referred to as *Basic ACC*, *CACC* and *CACC+* below.

With the *Basic ACC* approach, only a standard radar sensor is used to obtain information regarding the current states of the target vehicle, such as distance and speed. This solution is already commercially available on the market. The disadvantage of such an approach is that the information regarding the current states of the target vehicle has some delays (computational and sensor delay). For the set-point generation, see Section 3.2, we use

$$\hat{d}(t) = d_s(t) = d(t - T_s), \tag{14a}$$

$$\hat{v}_t(t) = v_e(t - T_s) + \dot{d}_s(t), \tag{14b}$$

$$\hat{a}_t(t) = 0, \tag{14c}$$

$$\hat{v}_e(t) = v_e(t). \tag{14d}$$

Note that there is no prediction of future vehicle states. Only the velocity of the ego vehicle, as well as the measured distance d_s and its derivative $\dot{d}_s$, are used.

In the *CACC* approach, information regarding the current velocity v_t and the current acceleration a_t of the target vehicle are transmitted by using V2X communication, see Fig. 1. This information is available in addition to the measured distance, enabling prediction of future vehicle states and compensation of time delays. For the set-point generation we assume that the acceleration of the target vehicle will be constant for the prediction horizon, such that

$$\hat{a}_t(t + T_p) = \tilde{a}_t(t) \tag{15}$$

applies, where $\tilde{a}_t(t) = a_t(t - T_c)$ is the received acceleration of the target vehicle. For the the predicted target velocity follows

$$\hat{v}_t(t + T_p) = \tilde{v}_t(t) + \tilde{a}_t(t)(T_p + T_c). \tag{16}$$

The other states are predicted with (8) and (13) according to Section 3.3. For the desired prediction we use $T_p = T_e$ in order to compensate the actuator delay T_e and thus minimize the tracking error $e_d = d_r - d$. Note that we

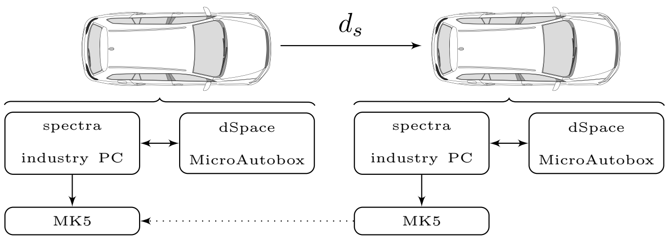

Fig. 5. Test setup and additional hardware components, with ego on the left and target vehicle on the right

assume that the acceleration of the target vehicle will be constant for the prediction horizon.

The *CACC+* approach is an extension of the *CACC* approach in which the desired acceleration a_{dt} is transmitted via V2X instead of the current acceleration a_t. This allows a further improvement of the prediction accuracy compared to the *CACC* approach. Now we predict the target velocity and the target acceleration with (10) and (11).

We are focusing on all three realization approaches since it will take several times until all vehicles will have V2X communication units and CACC functionality. In the meantime, we will have a mixed traffic situation where different CACC techniques will be used.

4. CASE STUDY

The proposed CACC approach was verified using two test vehicles of IAV on a test track. Both test vehicles used in the current study are VW Passat Variant equipped with additional computational hardware (dSpace MicroAutoBox and spectra industry PC) in the trunk. These components receive data directly from the bus systems of the vehicle and additional sensors. Fig. 5 shows the tested hardware set up respectively.

In our study, the first vehicle (target vehicle) was following the predefined velocity profile v_t, presented in Fig. 6 in green color. The second vehicle (ego vehicle) has utilized different time-gaps and was following the target vehicle. The effect of different time-gaps on the reference velocity can be studied in Fig. 6. Here, it can be seen that over-/under-shots occur in the reference velocity of large time-gaps if rapid velocity changes of the target vehicle are

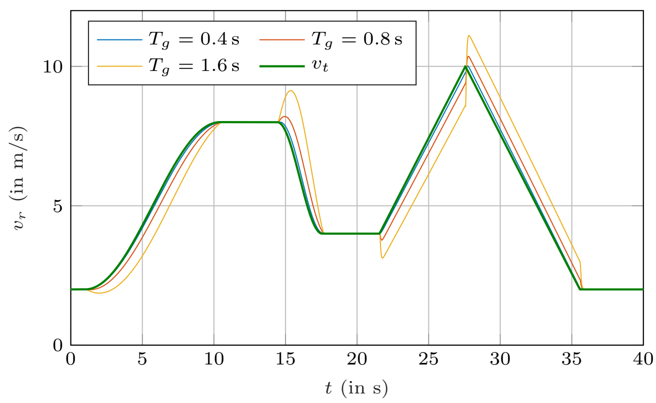

Fig. 6. Reference velocity v_r for different time-gaps

realized. On the one hand side, these effects results, since the tracking controller tries to eliminate the tracking error first before the vehicle reacts to the new reference velocity input. On the other hand, this over-/under-shot occur due to the fact, that a rapid velocity change results also in an un-intuitive time-gap change. For example at the second 15 in Fig. 6, the velocity of the target vehicle decreases. Consequently, the desired distance defined by the time-gap decreases. To track the decreased distance accurately, the ego vehicle increases his velocity (even though the target vehicle is slowing down). The corresponding reference distances are visualized in Fig. 7. Note that v_r and d_r in Fig. 6 and Fig. 7 are determined based on the simulated target velocity v_t to neglect measurement noise for the previous discussion.

In order to investigate the vehicle's following behavior, we applied different kinds of the proposed CACC functionality to the ego vehicle, as discussed in Section 3.4. The effect of the utilized CACC approaches on the closed-loop system behavior is illustrated in Fig. 8 and Fig. 9.

In Fig. 8, the distance tracking error e_d is shown for time-gaps $T_g = 0.8\,\mathrm{s}$. As expected, we had the smallest distance tracking error if we are using the transmitted desired vehicle movement states of the target vehicle (*CACC+*). The distance error increases in the case if only the actual vehicle states are transmitted from the target vehicle (*CACC*). This effect occurs due to prediction mismatches. Although the differences are generally small the resulting absolute values of the distance tracking error can double,

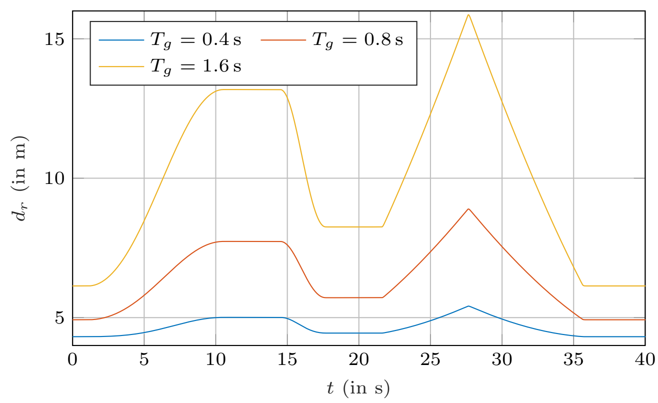

Fig. 7. Reference distance d_r for different time-gaps

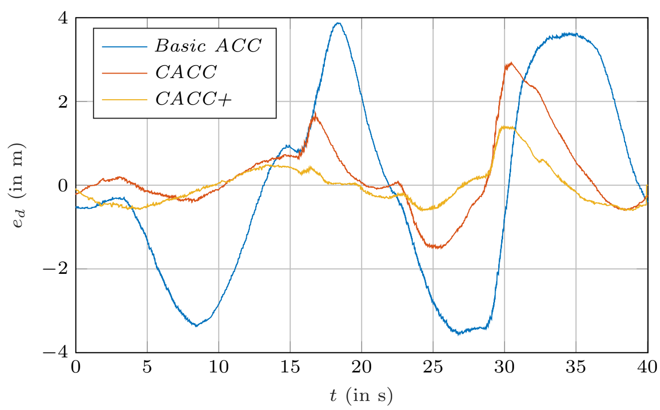

Fig. 8. Distance tracking error for $T_g = 0.8\,\mathrm{s}$

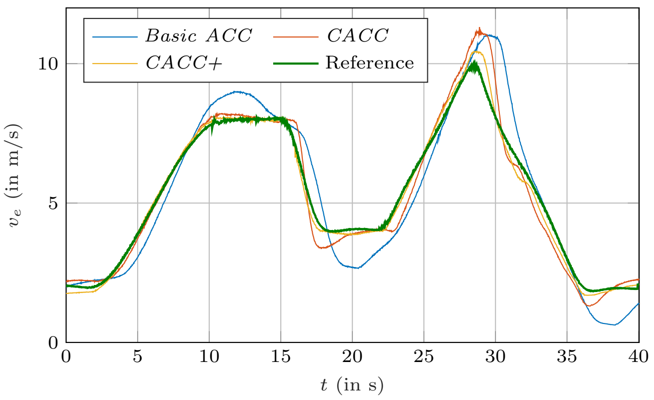

Fig. 9. Velocity tracking for $T_g = 0.8\,\mathrm{s}$

e. g. at 30 s. The biggest distance tracking error occurs if only the radar sensor and no V2X communication is used (*Basic ACC*). Overall, the distance tracking error using the radar is less than 4 m and can be reduced to a value smaller than 0.5 m with V2X communication.

The same effect can be seen in Fig. 9. Here, the reference velocity (green) is visualized, which is calculated based on the measured velocity of the target vehicle. Additionally, the ego velocity v_e for the different approaches is plotted. Similarly to the previous discussion, it is evident that the smallest velocity tracking error occurs in the case when the vehicle intention is transmitted (*CACC+*).

5. CONCLUSION AND OUTLOOK

In this paper, a proven approach for cooperative longitudinal vehicle guidance has been presented. Thereby, a classical ACC has been extended by a communication interface, to access the desired acceleration of the vehicle in front. In parallel with the detailed discussion of the utilized control structure, the communication method is addressed.

The practical example of platooning vehicles illustrated the performance of the approach. Further work will focus on a detailed, practical evaluation in public traffic of the presented method and the impact of platoons with a higher amount of vehicles.

REFERENCES

Auerswald, R., Busse, R., Dod, M., Fritzsche, R., Jungmann, A., Klöppel-Gersdorf, M., Krems, J.F., Lorenz, S., Schmalfuß, F., Springer, S., and Strobl, S. (2019). Cooperative driving in mixed traffic with heterogeneous communications and cloud infrastructure. In *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS)*, 95–105.

Benalie, N., Pananurak, W., Thanok, S., and Parnichkun, M. (2009). Improvement of adaptive cruise control system based on speed characteristics and time headway. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2403–2408.

C-ITS Platform (2016). Final report. Technical report, C-ITS Platform. Accessed: 2019-11-11.

Corona, D. and De Schutter, B. (2008). Adaptive Cruise Control for a SMART Car: A Comparison Benchmark for MPC-PWA Control Methods. *IEEE Transactions on Control Systems Technology*, 16(2), 365–372.

Desjardins, C. and Chaib-draa, B. (2011). Cooperative adaptive cruise control: A Reinforcement Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, 12(4), 1248–1260.

Esbensen, T., Jensen, B., Niss, M., and Sloth, C. (2007). Comparison of Longitudinal Control of Vehicles with and without Inter-vehicle Communication. Technical Report December, Section of Automation & Control, Department of Electronic Systems, Aalborg University, Aalborg.

Festag, A. (2014). Cooperative intelligent transport systems standards in europe. *IEEE Communications Magazine*, 52(12), 166–172.

Hermann, J., Bernhard, S., Konigorski, U., and Adamy, J. (2018). Designing communication topologies for optimal synchronization trajectories of homogeneous linear multi-agent systems. *European Control Conference*, 95(1), 1454–1461.

Ko, S. and Lee, J. (2007). Fuzzy logic based adaptive cruise control with guaranteed string stability. In *International Conference on Control, Automation and Systems*, 15–20.

Li, Z., Duan, Z., Chen, G., and Huang, L. (2010). Consensus of multiagent systems and synchronization of complex networks: A unified viewpoint. *IEEE Transaction on Circuits and Systems*, 57(1), 213–224.

Lidstrom, K. and Sjoberg, K. (2012). A modular cacc system integration and design. *IEEE Transactions on Intelligent Transport Systems*, 13(3), 1050–1061.

Olfati-Saber, R., Fax, J.A., and Murray, R.M. (2007). Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1), 215–233.

Rajamani, R. (2012). Vehicle dynamics and control. *Springer US*.

Richards, F.J. (1959). A Flexible Growth Function for Empirical Use. *Journal of Experimental Botany*, 10(2), 290–301.

Scardovi, L. and Sepulchre, R. (2009). Synchronization in networks of identical linear systems. *Automatica*, 45, 2557–2562.

Schrödel, F., Herrmann, P., and Schwarz, N. (2019). An improved multi-object adaptive cruise control approach. *10th IFAC Symposium on Intelligent Autonomous Vehicles.*

Stanger, T. and del Re, L. (2013). A model predictive Cooperative Adaptive Cruise Control approach. In *American Control Conference*, 1374 – 1379. Washington, DC, USA.

Taş, Ö.Ş., Kuhnt, F., Zöllner, J.M., and Stiller, C. (2016). Functional system architectures towards fully automated driving. In *2016 IEEE Intelligent Vehicles Symposium*, 304–309.

Trentelman, H.L., Takaba, K., and Monshizadeh, N. (2013). Robust synchronization of uncertain linear multi-agent systems. *IEEE Transaction on Automatic Control*, 58(6), 1511–1523.

Ulbrich, S., Menzel, T., Reschka, A., Schuldt, F., and Maurer, M. (2015). Defining and substantiating the terms scene, situation, and scenario for automated driving. In *IEEE 18th International Conference on Intelligent Transportation Systems*, 982–988.