2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

Distributed model predictive control for cooperative autonomous lane merging

Melanie Gallant[1⋆], Kevin Schmidt[1], Sven Reimann[1], Felix Berkel[1], Nina Majer[2], and Sören Hohmann[3]

***Abstract*— The use of communication technologies in advanced driver-assistance systems enables improving safety and control performance at the same time. Lane merging scenarios are typical traffic hubs which benefit from cooperative behavior but require guarantees on collision avoidance. To achieve these goals, we use a distributed model predictive control design with computationally efficient convex formulation of the optimal control problem. Safety guarantees are established through recursive feasibility using invariant terminal sets. Further, the framework of cooperative cost functions increases the global performance, such as the total time to merge, and maintains the formal guarantees. The performance of the proposed methods is illustrated by a numerical example, where the cooperative controller improves the overall cost by nearly 20%.**

I. Introduction

Advanced driver-assistance systems (ADAS) offer the opportunity to improve the traffic flow and reduce accidents [1]. Especially cooperative networked ADAS involving vehicle-to-everything (V2X) communication are a key technology towards autonomous driving [2]. Further, an increasing amount of computing capacities is enabled for real-time and safety-critical systems within the vision of reliable distributed systems [3]. This allows utilizing advanced algorithms such as distributed model predictive control (DMPC) to improve the system performance and safety at the same time. As an exemplary scenario, lane merging situations are considered throughout this article. These occur in various situations such as at highway on-ramps, at intersections or, if lanes are blocked due to road work. The mentioned traffic hubs can potentially benefit from connected and cooperative automated driving, as they represent bottlenecks and places of increased risk, where non-anticipatory and unsafe driving contributes to traffic congestion and accidents.

Due to the safety-critical nature of the control system, safety guarantees for the real system are essential and model predictive control (MPC) stands to reason. The control law is based on the solution of an optimal control problem which is solved at each time step in a receding horizon manner. This setting allows to address safety aspects such as collision avoidance by means of constraints in an explicit way.

[1]Robert Bosch GmbH, Corporate Sector Research and Advance Engineering, 71272 Renningen, Germany, Emails: {melanie.gallant, kevin.schmidt4, sven.reimann, felix.berkel}@de.bosch.com

[2]FZI Research Center for Information Technology, Haid-und-Neu-Straße 10-14, 76131 Karlsruhe, Germany, Email: majer@fzi.de

[3]Institute of Control Systems, Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany, Email: soeren.hohmann@kit.edu

Acknowledgements. This work was supported by the German Federal Ministry for Economic Affairs & Climate Action under grant no. 13IPC021.

In this article, we illustrate how the use of V2X communication makes it possible to apply the framework in a distributed fashion, which is beneficial due to enhanced fail-safe and scaling properties. This makes it possible to plan efficient anticipatory trajectories and to avoid the necessity of a central control unit. Consequently, no special infrastructural computing units are required.

State of the Art. Decentralized and distributed MPC has been successfully applied and investigated for lane merge scenarios, e.g., in [4], [5], [6], [7], [8], [9], [10], [11]. The approaches, however, do not ensure important theoretical properties such as recursive feasibility and constraint satisfaction and thus lack safety guarantees. In [12] recursive feasibility is ensured through the exchange of so-called contracts between the vehicles, i.e., sets which bound possible future trajectories. The approach requires exchanging multiple trajectories and solving the optimization problems several times which can be challenging in real-time applications. Another approach guaranteeing collision avoidance is presented in [13], where recursive feasibility is provided through the computation of forward reachable sets and an invariant terminal set. The local optimal control problems (OCPs) are posed as mixed-integer optimization problems, which are computationally expensive to solve.

Contributions. In this paper, a lane merge controller based on DMPC is proposed, combining both safety and performance. The OCPs in the distributed control algorithm are formulated as convex quadratically constrained quadratic programs, which are known to be solved efficiently. This allows the transfer to the application in a real-time context. Further, formal safety guarantees result from the proposed approach, which means that collisions are prevented at all times. For this purpose, a recursively feasible DMPC algorithm with suitable constraints and an invariant terminal set are designed. Finally, an extension to a cooperative DMPC is presented, where the global performance is improved regarding the total time until all vehicles have merged.

Structure of the Paper. This paper is structured as follows. In Section II, the lane merging scenario is introduced, the system model is shown in vehicle coordinates and in error coordinates as used in the context of terminal sets, and the objectives of this work are described. In Section III, the distributed MPC design is presented, starting at the sequential algorithm and the local OCP with the linear and terminal set constraints providing safety guarantees. Without violating results prior to this, an extension to include cooperative

behavior is then presented. Section IV evaluates the proposed approach using simulation of both the non-cooperative and the cooperative lane merge controller. Finally, conclusions and an outlook are given in Section V.

II. Problem Description and Objectives

A. Scenario Description

Figure 1 shows the lane merging scenario considered in this paper. The autonomous vehicles V_i with $i \in \{0, \dots, M-1\}$, where M is the number of vehicles, are distributed over the two lanes, the main lane L_0 and the merging lane L_1. Each vehicle is individually controlled by the distributed control algorithm designed in this work. It is assumed that vehicles on lane L_1 merge to lane L_0 at a fixed merging point s_m inside the control zone between the entry point s_in and the end point s_out, and no lane change maneuvers occur before this point. Moreover, the order in which the vehicles merge is fixed, e.g. by the first-in-first-out principle, and given before the first vehicle enters the control zone by a supervisory scheduling algorithm which is not considered in this paper. The vehicles V_i are numbered according to the merging order, V_0 being the first vehicle to pass s_m and V_{M-1} the last, as illustrated in Figure 1.

A vehicle can have up to four neighboring vehicles, consisting of the neighbors in the merging order and the neighbors in the same lane. The latter are the vehicles entering the control zone directly adjacent in the same lane, but not necessarily leaving the control zone directly adjacent. A vehicle has fewer than four neighbors if lane and merging order neighbors coincide, or if it is the first or last vehicle leaving the control zone. As the vehicles are numbered according to the merging order, in relation to any vehicle V_i, vehicle V_{i-1} is the front neighbor in the merging order and vehicle V_{i+1} is the rear neighbor in the merging order. The neighboring vehicles which enter the control zone in the same lane as V_i, which can be either L_0 or L_1, are the neighbors in the same lane. They are denoted as V_{i^-} for the front neighbor in the same lane and V_{i^+} for the rear neighbor in the same lane. If no vehicle merges between two neighbors in the same lane, these vehicles are also neighbors in the merging order. In addition to the global vehicle identifiers, Figure 1 shows an example of the notations used for neighboring vehicles in the same lane in gray from the point of view of V_2.

B. System Modeling

Vehicle Model. In this work, only longitudinal motion is considered. It is assumed there is an underlying low-level control loop regulating the acceleration and the lateral dynamics to match the path on the road and during the merging process. As frequently used in the literature, e.g. in [8], [11], [13], the vehicles V_i, $i \in \{0, \dots, M-1\}$ are modeled as point mass

$$\dot{s}_i(t) = v_i(t), \quad \dot{v}_i(t) = u_i(t), \tag{1}$$

where $s_i(t) \in \mathbb{R}$ is the longitudinal position, $v_i(t) \in \mathbb{R}$ the longitudinal velocity and $u_i(t) \in \mathbb{R}$ the desired longitudinal acceleration of V_i, at time $t \in \mathbb{R}_{\geq 0}$.

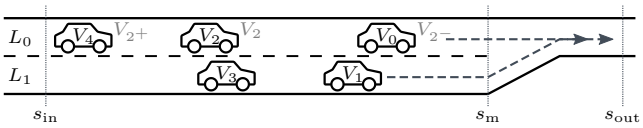


Fig. 1: Lane merging scenario with five vehicles $V_0, \dots, V_4$ and an example notation of neighboring vehicles in the same lane in relation to V_2 (gray).

The discrete-time system equation

$$\boldsymbol{x}_i(k+1) = \boldsymbol{A}_i \boldsymbol{x}_i(k) + \boldsymbol{B}_i u_i(k), \tag{2}$$

with the state vector $\boldsymbol{x}_i(k) = \begin{pmatrix} s_i(k) & v_i(k) \end{pmatrix}^\top \in \mathbb{R}^2$ at time step $k = {}^t\!/_{t_\mathrm{s}} \in \mathbb{Z}_{\geq 0}$ with sample time $t_\mathrm{s} \in \mathbb{R}_{>0}$ is obtained using Euler's explicit method. The discrete-time local system and input matrices are

$$\boldsymbol{A}_i = \begin{pmatrix} 1 & t_\mathrm{s} \\ 0 & 1 \end{pmatrix}, \quad \boldsymbol{B}_i = \begin{pmatrix} 0 \\ t_\mathrm{s} \end{pmatrix}. \tag{3}$$

The global system of M vehicles with system equation

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) \tag{4}$$

consists of the stacked local state vectors $\boldsymbol{x}(k) = \mathrm{col}(\{\boldsymbol{x}_i(k)\}_{0 \leq i < M}) \in \mathbb{R}^{2M}$, the stacked local inputs $\boldsymbol{u}(k) = \mathrm{col}(\{u_i(k)\}_{0 \leq i < M}) \in \mathbb{R}^M$, the block diagonal system matrix $\boldsymbol{A} = \mathrm{diag}(\{\boldsymbol{A}_i\}_{0 \leq i < M}) \in \mathbb{R}^{2M \times 2M}$ and the block diagonal input matrix $\boldsymbol{B} = \mathrm{diag}(\{\boldsymbol{B}_i\}_{0 \leq i < M}) \in \mathbb{R}^{2M \times M}$.

Error Model. In the sequel of the paper it is advantageous in some cases to recast the system in error coordinates $\boldsymbol{z}_i \in \mathbb{R}^2, i \in \{0, \dots, M-1\}$, where the subsystems consist of the states

$$\boldsymbol{z}_0(k) = \begin{pmatrix} s_0(k) - s_\mathrm{r}(k) \\ v_0(k) - v_\mathrm{r} \end{pmatrix}, \tag{5a}$$

$$\boldsymbol{z}_i(k) = \begin{pmatrix} s_{i-1}(k) - s_i(k) - d_\mathrm{r} \\ v_i(k) - v_\mathrm{r} \end{pmatrix}, \ i = 1, \dots, M-1, \tag{5b}$$

with the reference position $s_\mathrm{r}(k) = k t_\mathrm{s} v_\mathrm{r} \in \mathbb{R}_{>0}$, the constant reference distance $d_\mathrm{r} \in \mathbb{R}_{>0}$ and the constant reference velocity $v_\mathrm{r} \in \mathbb{R}_{>0}$. The states of the error model correspond to the velocity error and the distance error with respect to the front neighboring vehicle in the merging order.

The subsystems in error coordinates are dynamically coupled. The system equation of a subsystem is

$$\boldsymbol{z}_0(k+1) = \boldsymbol{A}_0 \boldsymbol{z}_0(k) + \boldsymbol{B}_0 u_0(k), \tag{6a}$$

$$\boldsymbol{z}_i(k+1) = \tilde{\boldsymbol{A}}_{i,i-1} \boldsymbol{z}_{i-1}(k) + \tilde{\boldsymbol{A}}_{i,i} \boldsymbol{z}_i(k) + \boldsymbol{B}_i u_i(k), \quad i = 1, \dots, M-1, \tag{6b}$$

or in general terms $\boldsymbol{z}_i(k+1) = \tilde{\boldsymbol{A}}_{\mathcal{N}_i} \boldsymbol{z}_{\mathcal{N}_i}(k) + \boldsymbol{B}_i u_i(k)$, with the distributed system matrices $\tilde{\boldsymbol{A}}_{\mathcal{N}_0} = \boldsymbol{A}_0$ and $\tilde{\boldsymbol{A}}_{\mathcal{N}_i} = \begin{pmatrix} \tilde{\boldsymbol{A}}_{i,i-1} & \tilde{\boldsymbol{A}}_{i,i} \end{pmatrix}$ for $i = 1, \dots, M-1$. The state vectors $\boldsymbol{z}_{\mathcal{N}_0} = \boldsymbol{z}_0$ and $\boldsymbol{z}_{\mathcal{N}_i} = \begin{pmatrix} \boldsymbol{z}_{i-1}^\top & \boldsymbol{z}_i^\top \end{pmatrix}^\top \in \mathbb{R}^4$ for $i = 1, \dots, M-1$, are the stacked state vectors of error subsystems dynamically coupled to $\boldsymbol{z}_i$. The blocks of the distributed system matrices are defined as

$$\tilde{\boldsymbol{A}}_{i,i-1} = \begin{pmatrix} 0 & t_\mathrm{s} \\ 0 & 0 \end{pmatrix}, \quad \tilde{\boldsymbol{A}}_{i,i} = \begin{pmatrix} 1 & -t_\mathrm{s} \\ 0 & 1 \end{pmatrix}, \tag{7}$$

and combined into the global system matrix

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{A}_0 & & & \\ \tilde{\boldsymbol{A}}_{1,0} & \tilde{\boldsymbol{A}}_{1,1} & & \\ & \tilde{\boldsymbol{A}}_{2,1} & \tilde{\boldsymbol{A}}_{2,2} & \\ & & & \ddots \end{pmatrix}. \tag{8}$$

Using the stacked local state vector $\boldsymbol{z}(k) = \mathrm{col}(\{\boldsymbol{z}_i(k)\}_{0 \le i < M}) \in \mathbb{R}^{2M}$, the global system equation of M error subsystems is

$$\boldsymbol{z}(k+1) = \tilde{\boldsymbol{A}}\boldsymbol{z}(k) + \boldsymbol{B}\boldsymbol{u}(k). \tag{9}$$

Vehicle coordinates are obtained from error coordinates through the affine global transformation

$$\boldsymbol{z}(k) = \boldsymbol{T}\boldsymbol{x}(k) - \boldsymbol{z}_\mathrm{r}(k). \tag{10}$$

using a suitable transformation matrix $\boldsymbol{T} \in \mathbb{R}^{2M \times 2M}$ and the reference vector $\boldsymbol{z}_\mathrm{r}(k) = (s_r(k),\ v_r,\ d_r,\ \ldots,\ v_r)^\top$.

Constraints. The local accelerations $u_i(k)$ are constrained by the lower limit $u_\mathrm{min} \in \mathbb{R}_{<0}$ and the upper limit $u_\mathrm{max} \in \mathbb{R}_{>0}$, which is described by the local input constraint set

$$\mathcal{U}_i = \{u_i \in \mathbb{R} \mid u_\mathrm{min} \le u_i \le u_\mathrm{max}\}, \tag{11}$$

and the global input constraint set

$$\mathcal{U} = \{\boldsymbol{u} \in \mathbb{R}^M \mid u_i \in \mathcal{U}_i\ \forall i \in \{0, \ldots, M-1\}\}. \tag{12}$$

The local sets of state constraints consist of the constant limitation of the velocities by the limit values $v_\mathrm{min} \in \mathbb{R}_{\ge 0}$ and $v_\mathrm{max} \in \mathbb{R}_{>0}$ and of the restriction of a vehicle's position based on its neighbors' positions. To prevent collisions, a minimum distance $d_\mathrm{min} \in \mathbb{R}_{>0}$ is introduced, which must be maintained at all times between vehicles in the same lane, thus between V_i and V_{i^-} and between V_i and V_{i^+}. If the merging order neighbors V_{i-1} or V_{i+1} enter the control zone in another lane than V_i, the minimum distance is required only from the moment when the vehicle in front reaches s_m, with no distance constraint beforehand. In summary, the state constraint set is formulated as

$$\begin{aligned} \mathcal{X}_i = \{\boldsymbol{x}_i \in \mathbb{R}^2 \mid\ & v_\mathrm{min} \le v_i \le v_\mathrm{max}, \\ & s_{i^-} - s_i \ge d_\mathrm{min}, \quad s_i - s_{i^+} \ge d_\mathrm{min}, \\ & s_{i-1} - s_i \ge \begin{cases} d_\mathrm{min}, & \text{if } s_{i-1} \ge s_\mathrm{m} \\ -\infty, & \text{otherwise} \end{cases}, \\ & s_i - s_{i+1} \ge \begin{cases} d_\mathrm{min}, & \text{if } s_i \ge s_\mathrm{m} \\ -\infty, & \text{otherwise} \end{cases} \}, \end{aligned} \tag{13}$$

and the global set of state constraints as

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^{2M} \mid \boldsymbol{x}_i \in \mathcal{X}_i\ \forall i \in \{0, \ldots, M-1\}\}. \tag{14}$$

In addition, the constraint set $\mathcal{Z}$ is formulated in error coordinates, where all distances are required to comply with the minimum distance regardless of the vehicles' positions with respect to s_m. Fulfillment of $\mathcal{Z}$ guarantees compliance with the global state constraints $\mathcal{X}$. This set of constraints, formulated as

$$\begin{aligned} \mathcal{Z} = \{\boldsymbol{z} \in \mathbb{R}^{2M} \mid\ & v_\mathrm{min} \le v_i \le v_\mathrm{max}\ \forall i \in \{0, \ldots, M-1\}, \\ & s_{i-1} - s_i \ge d_\mathrm{min}\ \forall i \in \{1, \ldots, M-1\}\}, \end{aligned} \tag{15}$$

is crucial for the design of the terminal set applied to the last predicted values in the OCP formulated in Section III-B. The constant set of constraints $\mathcal{Z}$ does not unduly increase the conservatism of the controller, assuming a sufficiently large prediction horizon.

C. Objectives

The goal of this work is the design of a distributed controller guaranteeing collision prevention and compliance with the constraints at all times. The approach should be scalable and use only sparse communication as described in Section II without necessitating a central roadside control unit. On top of a distributed controller architecture, limited computational effort is a significant component of a scalable real-time control system.

The objectives of the control system are individual, as well as collective in nature. Regarding the former, minimization of acceleration efforts and deviations from a constant reference velocity and a safety distance to the vehicle in the front are desired. On the other hand, the negative impact on traffic of the lane merge bottleneck should be reduced by incorporating cooperation between vehicles for a decrease in the total time needed until each vehicle has merged and achieving a more even distribution of the merge's impact in terms of costs on all participating vehicles.

III. Distributed MPC design

A. Distributed Control Algorithm

A non-iterative sequential control algorithm is used, where the vehicles measure their states and apply their acceleration inputs simultaneously using a synchronized clock. To compute their optimal input signals, the vehicles solve their local OCPs sequentially according to the merging order. Algorithm 1 describes the lane merge control from the perspective of each vehicle and Figure 2 shows the necessary data from neighboring vehicles at time step k. In the following, the qualifier $(l|k)$ denotes the prediction of

Algorithm 1: Lane merge control for each $i \in \{0, \ldots, M-1\}$.

```
k = 0
while lane merge control is active do
    measure x_i(k)
    wait until OCPs of front neighbors are solved
    receive trajectories x*_{i-}(·|k) and x*_{i-1}(·|k)
    receive z*_{i-1}(k+N-1|k)
    update terminal set Z_{f,i}(k) according to (23)
    solve local OCP (16)
    send trajectory x*_i(·|k) to all neighbors
    send z*_i(k+N-1|k) to V_{i+1}
    send updated terminal set Z_{f,i}(k) to V_{i-1}
    wait until OCPs of rear neighbors are solved
    receive trajectories x*_{i+}(·|k), x*_{i+1}(·|k)
    receive updated terminal set Z_{f,i+1}(k)
    apply u*_i(k|k)
    k = k + 1
```

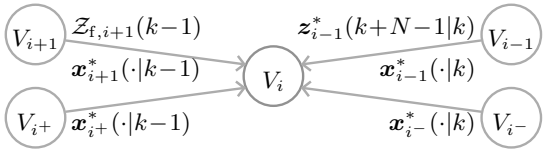


Fig. 2: Data received from neighboring vehicles used during the local control at time step k of vehicle V_i.

the respective quantity at future time step l, based on time step k. The optimal solution of an OCP is marked as $(\cdot)^\star$.

In the controller, a terminal set is used which is composed of local terminal sets that are updated at each time step according to Section III-D. Due to the sequential optimization, the front neighbors transmit their newly optimized trajectories $\boldsymbol{x}^\star_{i^-}(\cdot|k)$ and $\boldsymbol{x}^\star_{i-1}(\cdot|k)$ at time step k, while the rear neighbors' optimized trajectories $\boldsymbol{x}^\star_{i^+}(\cdot|k-1)$ and $\boldsymbol{x}^\star_{i+1}(\cdot|k-1)$ from the last optimization are used in the local OCP. Additionally, the local terminal set $\mathcal{Z}_{\mathrm{f},i+1}(k-1)$ from the rear error subsystem received at the previous time step is used, which is described in Section III-D. The update of the local terminal set depends on $\boldsymbol{z}^\star_{i-1}(k+N-1|k)$, which is transmitted from the front neighbor in the merging order.

B. Optimal Control Problem

The local OCP, solved on each vehicle's computing unit sequentially according to the merging order, is given by

$$\min_{\substack{u_i(\cdot|k),\boldsymbol{x}_i(\cdot|k),\\ \rho_{i,i^-}(\cdot|k),\\ \rho_{i,i-1}(\cdot|k)}} J_i(\boldsymbol{x}_i(k), u_i(k{:}k+N-1|k)) \tag{16a}$$

$$\text{s.t. } \boldsymbol{x}_i(k|k) = \boldsymbol{x}_i(k), \tag{16b}$$

$$\boldsymbol{z}_i(k+N|k) \in \mathcal{Z}_{\mathrm{f},i}(k), \tag{16c}$$

$$\boldsymbol{z}_i(k+N-1|k) \in \mathcal{Z}_{\mathrm{f},i}(k-1), \tag{16d}$$

$$\boldsymbol{z}_{i+1}(k+N-1|k) \in \mathcal{Z}_{\mathrm{f},i+1}(k-1), \tag{16e}$$

$$\text{for all } l = k,\dots,k+N-1:$$

$$\boldsymbol{x}_i(l+1|k) = \boldsymbol{A}_i\boldsymbol{x}_i(l|k) + \boldsymbol{B}_i u_i(l|k), \tag{16f}$$

$$u_i(l|k) \in \mathcal{U}_i, \tag{16g}$$

$$\boldsymbol{x}_i(l|k), \rho_{i,i^-}(l|k), \rho_{i,i-1}(l|k) \in \mathcal{X}'_i, \tag{16h}$$

$$\boldsymbol{z}_{i+1}(l|k) = \begin{pmatrix} s_i(l|k) - s^\star_{i+1}(l|k-1) - d_\mathrm{r} \\ v^\star_{i+1}(l|k-1) - v_\mathrm{r} \end{pmatrix}, \tag{16i}$$

$$\text{for all } l = k,\dots,k+N:$$

$$\boldsymbol{z}_i(l|k) = \begin{pmatrix} s^\star_{i-1}(l|k) - s_i(l|k) - d_\mathrm{r} \\ v_i(l|k) - v_\mathrm{r} \end{pmatrix}, \tag{16j}$$

where the prediction horizon is given by N. The predicted state sequence $\boldsymbol{x}_i(\cdot|k)$ is initiated by the measurement according to (16b) and follows the local system dynamics (16f). Constraints (16c), (16d) and (16e) with error subsystem variables specified in constraints (16j) and (16i) are the terminal constraints for recursive feasibility, as described in Section III-D. For the predicted values at time steps $l = k,\dots,k+N-1$, the constraint sets (16g) and (16h) are imposed, the former from Section II-B and the latter being described in Section III-C. The slack variables $\rho_{i,i^-} \in \mathbb{R}$ and $\rho_{i,i-1} \in \mathbb{R}$ are additional optimization variables mapping the lower deviation from the safety distance introduced in Section III-C.

The solution of the OCP is the optimal input trajectory $u^\star_i(k:k+N-1|k)$ producing the optimal nominal state trajectory $\boldsymbol{x}^\star_i(k:k+N|k)$ and $u^\star_i(k|k)$ is the acceleration applied to the vehicle after completion of the control iteration over all vehicles.

C. Linear State Constraints

The state constraints (13) form a non-convex constraint set with respect to the ego vehicle's position due to the switching distance constraints between neighbors in the merging order. Through the sequential update and the modifications in this section, they become linear in the optimization variables.

Additionally, the slack variables ρ_{i,i^-} and $\rho_{i,i-1}$ are added to the distance constraints to the front neighbors, to incentivize in (16) an increase of the distance up to the safety distance $d_\mathrm{min} + t_\mathrm{d} v_i(k)$ with the safety time gap $t_\mathrm{d} \in \mathbb{R}_{\geq 0}$. The distance d_min is maintained as the lower distance constraint where applicable, in order to operate the controller away from the constraints, increasing the degrees of freedom during the lane merge.

The resulting constraint set coupled to the optimized position trajectories of neighboring vehicles is

$$\mathcal{X}'_i = \{\boldsymbol{x}_i \in \mathbb{R}^{2M} \mid (18)\} \tag{17}$$

composed of the individual constraints

$$v_\mathrm{min} \leq v_i(l|k) \leq v_\mathrm{max}, \tag{18a}$$

$$s^\star_{i^-}(l|k) - s_i(l|k) \geq d_\mathrm{min} + t_\mathrm{d} v_i(l|k) - \rho_{i,i^-}(l|k), \tag{18b}$$

$$\rho_{i,i^-}(l|k) \leq t_\mathrm{d} v_i(l|k), \tag{18c}$$

$$s_i(l|k) - s^\star_{i^+}(l|k-1) \geq d_\mathrm{min}, \tag{18d}$$

$$s^\star_{i-1}(l|k) - s_i(l|k) \geq \begin{cases} d_\mathrm{min} + t_\mathrm{d} v_i(l|k) - \rho_{i,i-1}(l|k), & \text{if } s^\star_{i-1}(l|k) \geq s_\mathrm{m} \\ -\infty, & \text{otherwise,} \end{cases} \tag{18e}$$

$$\rho_{i,i-1}(l|k) \leq t_\mathrm{d} v_i(l|k), \tag{18f}$$

$$s_i(l|k) \leq \begin{cases} s_\mathrm{m} - d_\mathrm{min}, & \text{if } s^\star_{i-1}(l|k) \leq s_\mathrm{m} \\ \infty, & \text{otherwise,} \end{cases} \tag{18g}$$

$$s_i(l|k) - s^\star_{i+1}(l|k-1) \geq \begin{cases} d_\mathrm{min}, & \text{if } s^\star_{i+1}(l|k-1) > s_\mathrm{m} - d_\mathrm{min} \\ -\infty, & \text{otherwise.} \end{cases} \tag{18h}$$

D. Terminal Sets

In order to design a recursively feasible MPC algorithm, a structured positive invariant set and terminal feedback $\tilde{\boldsymbol{K}}_\mathrm{f} \in \mathbb{R}^{M\times 2M}$ are used, designed in a distributed manner according to [14]. The local terminal set is defined as

$$\mathcal{Z}_{\mathrm{f},i}(k) = \left\{\boldsymbol{z}_i \in \mathbb{R}^2 \mid \boldsymbol{z}_i^\top \boldsymbol{P}_i \boldsymbol{z}_i \leq \alpha_i(k)\right\}, \tag{19}$$

where $\boldsymbol{P}_i \in \mathbb{R}^{2\times 2}$ and the time-varying local set sizes $\alpha_i(k) \in \mathbb{R}_{\geq 0}$ with $\sum_{i=0}^{M-1} \alpha_i(k) \leq 1$ define the ellipsoidal set. The corresponding global set is $\mathcal{Z}_\mathrm{f} =$

$\left\{\boldsymbol{z} \in \mathbb{R}^{2M} \mid \boldsymbol{z}^\top \boldsymbol{P} \boldsymbol{z} \leq 1\right\}$ with block diagonal matrix $\boldsymbol{P} = \mathrm{diag}(\{\boldsymbol{P}_i\}_{0 \leq i < M}) \in \mathbb{R}^{2M \times 2M}$.

Local terminal controllers of the form $u_i = \tilde{\boldsymbol{K}}_{\mathrm{f},\mathcal{N}_i} \boldsymbol{z}_{\mathcal{N}_i}$ with $\tilde{\boldsymbol{K}}_{\mathrm{f},\mathcal{N}_0} = \tilde{\boldsymbol{K}}_{\mathrm{f},0,0}$ and $\tilde{\boldsymbol{K}}_{\mathrm{f},\mathcal{N}_i} = \begin{pmatrix} \tilde{\boldsymbol{K}}_{\mathrm{f},i,i-1} & \tilde{\boldsymbol{K}}_{\mathrm{f},i,i} \end{pmatrix}$ for $i \in \{1, \dots, M-1\}$ are considered, yielding the structured global terminal feedback matrix

$$\tilde{\boldsymbol{K}}_{\mathrm{f}} = \begin{pmatrix} \tilde{\boldsymbol{K}}_{\mathrm{f},0,0} & & & \\ \tilde{\boldsymbol{K}}_{\mathrm{f},1,0} & \tilde{\boldsymbol{K}}_{\mathrm{f},1,1} & & \\ & \tilde{\boldsymbol{K}}_{\mathrm{f},2,1} & \tilde{\boldsymbol{K}}_{\mathrm{f},2,2} & \\ & & & \ddots \end{pmatrix}. \tag{20}$$

The local terminal sets (19) and the feedback matrix (20) are to be designed such that the set $\mathcal{Z}_{\mathrm{f}}$ is positive invariant for the global closed-loop system

$$\boldsymbol{z}(k+1) = \tilde{\boldsymbol{A}}_{\mathrm{K_f}} \boldsymbol{z}(k) \tag{21}$$

with the closed-loop system matrix $\tilde{\boldsymbol{A}}_{\mathrm{K_f}} = \tilde{\boldsymbol{A}} + \boldsymbol{B}\tilde{\boldsymbol{K}}_{\mathrm{f}} \in \mathbb{R}^{2M \times 2M}$ according to the following definition.

Definition 3.1: A set $\mathcal{Z}_{\mathrm{f}}$ is positive invariant for system (21) with constraints $\boldsymbol{z} \in \mathcal{Z}$, $\tilde{\boldsymbol{K}}_{\mathrm{f}} \boldsymbol{z} \in \mathcal{U}$, if it holds that $\mathcal{Z}_{\mathrm{f}} \subseteq \mathcal{Z}$, $\tilde{\boldsymbol{K}}_{\mathrm{f}} \mathcal{Z}_{\mathrm{f}} \subseteq \mathcal{U}$ and $\boldsymbol{z}(k+1) = \tilde{\boldsymbol{A}}_{\mathrm{K_f}} \boldsymbol{z}(k) \in \mathcal{Z}_{\mathrm{f}}\ \forall \boldsymbol{z}(k) \in \mathcal{Z}_{\mathrm{f}}$.

Remark 3.1: From $\boldsymbol{z}(k) \in \mathcal{Z}$ follows for the transformed vehicle states $\boldsymbol{x}_i(k) \in \mathcal{X}_i'$, $\forall i \in \{0, \dots, M-1\}$, since $\mathcal{Z} \subseteq \mathcal{X}' = \{\boldsymbol{x} \in \mathbb{R}^{2M} \mid \boldsymbol{x}_i \in \mathcal{X}_i',\ \forall i \in \{0, \dots, M-1\}\}$.

The terminal sets are used under the following assumption.

Assumption 3.1: Given the local sets (19) and the structured global set $\mathcal{Z}_{\mathrm{f}}$, it holds that $\mathcal{Z}_{\mathrm{f},0}(k) \times \cdots \times \mathcal{Z}_{\mathrm{f},M-1}(k) \subseteq \mathcal{Z}_{\mathrm{f}},\ \forall k \geq 0$. Furthermore, there exists a suitable update function of the local terminal sets (19) for system (21), such that if $\boldsymbol{z}_i(k) \in \mathcal{Z}_{\mathrm{f},i}(k)\ \forall i \in \{0, \dots, M-1\}$, then $\boldsymbol{z}_i(k+1) \in \mathcal{Z}_{\mathrm{f},i}(k+1)\ \forall i \in \{0, \dots, M-1\}$. Using the sets of dynamic coupled subsystems $\mathcal{N}_0 = \{0\}$ and $\mathcal{N}_i = \{i-1, i\}$ and the corresponding update function, it holds that if $\boldsymbol{z}_j(k) \in \mathcal{Z}_{\mathrm{f},j}(k)\ \forall j \in \mathcal{N}_i$, then $\boldsymbol{z}_i(k+1) \in \mathcal{Z}_{\mathrm{f},i}(k+1)$.

The above assumption holds true if the structured ellipsoidal set $\mathcal{Z}_{\mathrm{f}}$, the terminal feedback $\tilde{\boldsymbol{K}}_{\mathrm{f}}$ and the update matrices $\boldsymbol{\Gamma}_{\mathcal{N}_0} \in \mathbb{R}^{2\times 2}$ and $\boldsymbol{\Gamma}_{\mathcal{N}_i} \in \mathbb{R}^{4\times 4}$ for $i \in \{1, \dots, M-1\}$ of the terminal set update function

$$\alpha_i(k+1) = \alpha_i(k) + \boldsymbol{z}_{\mathcal{N}_i}^\top(k) \boldsymbol{\Gamma}_{\mathcal{N}_i} \boldsymbol{z}_{\mathcal{N}_i}(k) \tag{22}$$

are designed according to the approach from [14]. Additionally, the approach leads to a positive invariant set $\mathcal{Z}_{\mathrm{f}}$ according to Definition 3.1.

Application in Sequential Control Algorithm. The local terminal sets designed in [14] are applied to a distributed MPC problem with parallel optimization. In this work, the distributed OCPs are optimized sequentially, therefore the approach is modified here.

Terminal constraint (16c) guarantees feasibility at the next time step, while (16d) and (16e) ensure that subsequently optimized vehicles maintain feasibility of their propagated terminal values throughout the sequential control loop. The latter constraint requires knowledge of the local terminal set $\mathcal{Z}_{\mathrm{f},i+1}(k-1)$ of the rear neighbor in the merging order. The local terminal set $\mathcal{Z}_{\mathrm{f},i}(k)$ is updated before solving the OCP at time step k based on new information from vehicles in front consisting of $\boldsymbol{z}_{i-1}^\star(k+N-1|k)$ and $\boldsymbol{x}_{i-1}^\star(k+N-1|k)$, with the update function

$$\begin{aligned} \alpha_i(k) &= \alpha_i(k-1) \\ &+ \boldsymbol{z}_{\mathcal{N}_i}^{\star\top}(k+N-1|k-1^+) \boldsymbol{\Gamma}_{\mathcal{N}_i} \boldsymbol{z}_{\mathcal{N}_i}^\star(k+N-1|k-1^+), \end{aligned} \tag{23}$$

and initial value $\alpha_i(0) = \frac{1}{M}\ \forall i \in \{0, \dots, M-1\}$. The state vector used in the update function with $l = k+N-1$ is composed as $\boldsymbol{z}_{\mathcal{N}_0}^\star(l|k-1^+) = \boldsymbol{z}_0^\star(l|k-1)$ and

$$\boldsymbol{z}_{\mathcal{N}_i}^\star(l|k-1^+) = \begin{pmatrix} \boldsymbol{z}_{i-1}^\star(l|k) \\ \boldsymbol{z}_i^\star(l|k-1^+) \end{pmatrix}, \tag{24}$$

with

$$\boldsymbol{z}_i^\star(l|k-1^+) = \begin{pmatrix} s_{i-1}^\star(l|k) - s_i^\star(l|k-1) - d_{\mathrm{r}} \\ v_i^\star(l|k-1) - v_{\mathrm{r}} \end{pmatrix}, \tag{25}$$

for $i \in \{1, \dots, M-1\}$. The superscript $(\cdot)^+$ in the context of an optimized state vector of an error subsystem indicates that front vehicles V_j with $j \in \{0, \dots, i-1\}$ have already solved their OCPs and their newly optimized trajectories are considered. The ego vehicle and rear vehicles V_j with $j \in \{i, \dots, M-1\}$ have not yet solved their OCPs and optimized trajectories from the previous sampling time step are used.

The application of the terminal sets in the sequential distributed control algorithm is based on the following result, which can be inferred from the described approach of the update function. If $\boldsymbol{z}_0^\star(k+N-1|k-1^+) \in \mathcal{Z}_{\mathrm{f},0}(k-1)$ or

$$\boldsymbol{z}_{i-1}^\star(k+N-1|k) \in \mathcal{Z}_{\mathrm{f},i-1}(k-1), \tag{26a}$$

$$\boldsymbol{z}_i^\star(k+N-1|k-1^+) \in \mathcal{Z}_{\mathrm{f},i}(k-1), \tag{26b}$$

for $i \in \{1, \dots, M-1\}$, then it holds that

$$\boldsymbol{z}_i(k+N|k) = \tilde{\boldsymbol{A}}_{\mathrm{K_f},\mathcal{N}_i} \boldsymbol{z}_{\mathcal{N}_i}^\star(k+N-1|k-1^+) \in \mathcal{Z}_{\mathrm{f},i}(k), \tag{27}$$

using update function (23) and local closed-loop system matrix $\tilde{\boldsymbol{A}}_{\mathrm{K_f},\mathcal{N}_0} = \tilde{\boldsymbol{A}}_{\mathcal{N}_0} + \boldsymbol{B}_0 \tilde{\boldsymbol{K}}_{\mathrm{f},\mathcal{N}_0} \in \mathbb{R}^{2\times 2}$ and $\tilde{\boldsymbol{A}}_{\mathrm{K_f},\mathcal{N}_i} = \tilde{\boldsymbol{A}}_{\mathcal{N}_i} + \boldsymbol{B}_i \tilde{\boldsymbol{K}}_{\mathrm{f},\mathcal{N}_i} \in \mathbb{R}^{2\times 4}$ for $i \in \{1, \dots, M-1\}$.

E. Cost Function

The local trajectories are optimized with respect to the local actuation effort, the deviation from the reference velocity and the lower deviation from the safety distance to front vehicles. The components are each weighted with the factors p, q, $r \in \mathbb{R}_{\geq 0}$, yielding the cost function

$$\begin{aligned} J_i(\boldsymbol{x}_i(k), u_i(\cdot|k)) = \sum_{l=k}^{k+N-1} \big[& p \left(\rho_{i,i^-}(l|k)^2 + \rho_{i,i-1}(l|k)^2 \right) \\ & + q \left(v_i(l|k) - v_{\mathrm{r}} \right)^2 + r u_i(l|k)^2 \big]. \end{aligned} \tag{28}$$

F. Main Results

The main results of the proposed control approach are the computational efficiency due to the formulation of the OCP and the safety guarantees.

As outlined in Sections III-C, III-D and the following Section III-E, the local OCP is composed of a quadratic cost function (16a), ellipsoidal terminal constraints (16c), (16d) and (16e) and the linear input and state constraints (16g),

(16h) and thus forms a convex quadratically constrained quadratic program which can be solved efficiently, e.g., using an interior-point solver [15].

Furthermore, safety with respect to collision prevention is guaranteed at all times, starting from an initial feasible solution of OCP (16) for all vehicles, which exists under the following assumption.

Assumption 3.2: It is assumed that the prediction horizon N of the OCPs is sufficiently large for an initial feasible solution of (16) $u_i^\star(k{:}k{+}N{-}1|k)$, $\boldsymbol{x}_i^\star(k{:}k{+}N|k)$, $\rho_{i,i^-}^\star(k{:}k{+}N{-}1|k)$, $\rho_{i,i-1}^\star(k{:}k{+}N{-}1|k)$ to exist at time $k=0$ for all $i \in \{0,\dots,M{-}1\}$.

Remark 3.2: This assumption is necessary due to $\boldsymbol{z}(k{+}N|k) \in \mathcal{Z}_\mathrm{f} \subseteq \mathcal{Z}$, where all vehicles are required to keep the minimum distance d_min regardless of their positions with respect to the merging point s_m. The prediction horizon required to fulfill Assumption 3.2 is dependent on factors such as the number of vehicles M, the reference values d_r as well as v_r and should be increased if no feasible solution is found at time $k{=}0$, for any $i \in \{0,\dots,M{-}1\}$.

In the following, a theorem on the recursive feasibility of the developed control algorithm is stated. The corresponding proof can be found in Appendix A.

Theorem 3.1 (Recursive Feasibility): Consider system (4) with the control law defined in Algorithm 1. Let Assumption 3.1 and 3.2 hold, then there exist feasible sequences $\hat{u}_i(k{:}k{+}N{-}1|k)$, $\hat{\boldsymbol{x}}_i(k{:}k{+}N|k)$, $\hat{\rho}_{i,i^-}(k{:}k{+}N{-}1|k)$, $\hat{\rho}_{i,i-1}(k{:}k{+}N{-}1|k)$ for all $k > 0$. Moreover, the input and state comply with the constraints (16g) and (16h).

G. Cooperative Lane Merging

Cooperative behavior is used to achieve a more balanced and globally improved control. Due to cooperative cost functions, vehicles accept an increase in their individual costs if this results in an overall reduction when taking into account the rear neighboring vehicles' costs. This is implemented with additional slack variables $\rho_{i^+,i} \in \mathbb{R}$ and $\rho_{i+1,i} \in \mathbb{R}$ mapping the lower deviation from the rear neighbor's safety distances to the currently optimizing vehicle, which enables the retention of the sequential control loop. For this purpose, OCP constraint (18d) from $\mathcal{X}_i'$ is extended to

$$\begin{aligned} &s_i(l|k) - s_{i^+}^\star(l|k{-}1) \geq \\ &\quad d_\mathrm{min} + t_\mathrm{d} v_{i^+}^\star(l|k{-}1) - \rho_{i^+,i}(l|k), \end{aligned} \tag{29a}$$

$$\rho_{i^+,i}(l|k) \leq t_\mathrm{d} v_{i^+}^\star(l|k{-}1). \tag{29b}$$

Similarly, the constraint (18h) is extended to

$$\begin{aligned} &s_i(l|k) - s_{i+1}^\star(l|k{-}1) \geq \\ &\begin{cases} d_\mathrm{min} + t_\mathrm{d} v_{i+1}^\star(l|k{-}1) & \text{if } s_{i+1}^\star(l|k{-}1) > \\ \qquad - \rho_{i+1,i}(l|k), & \qquad s_\mathrm{m} - d_\mathrm{min} \\ -\infty, & \text{otherwise,} \end{cases} \end{aligned} \tag{30a}$$

$$\rho_{i+1,i}(l|k) \leq t_\mathrm{d} v_{i+1}^\star(l|k{-}1). \tag{30b}$$

The cooperative cost function is a weighted sum of the individual costs (28) and the variable parts of the rear neighbors' costs, defined as

$$\begin{aligned} J_{\mathcal{N}_i}(\boldsymbol{x}_i(k), u_i(\cdot|k)) &= \omega_\mathrm{i} J_i(\boldsymbol{x}_i(k), u_i(\cdot|k)) \\ &+ \sum_{l=k}^{k+N-1} \left(\omega_\mathrm{n} \rho_{i^+,i}(l|k)^2 + \omega_\mathrm{o} \rho_{i+1,i}(l|k)^2\right). \end{aligned} \tag{31}$$

The weights ω_i, ω_n, $\omega_\mathrm{o} \in \mathbb{R}_{\geq 0}$ adjust the cooperative behavior, where $\omega_\mathrm{n} = \omega_\mathrm{o} = 0$ reduces the cooperative controller to the individual one. As the minimum required distances remain the same, the local OCPs are only changed with respect to the costs and the main results presented in Section III-F hold.

IV. Numerical Example

A. Setup

The proposed DMPC algorithm for autonomous lane merging is simulated in MATLAB. The ellipsoidal terminal sets $\mathcal{Z}_{\mathrm{f},i}$ computed at the beginning of the simulation and the distributed local OCPs are solved using the toolbox YALMIP [16] with the solver MOSEK [17].

The constant parameter values used to define the scenario, the vehicles and the MPC framework are shown in Table I. For the given simulation parameters, the computation time of OCP (16) for a vehicle at each time step averages $36\,\mathrm{ms}$ (Recorded on a device with Intel Core i7-1270P, $32\,\mathrm{GB}$ RAM).

The initial longitudinal positions are random parameters, where the initial distances between neighboring vehicles in the same lane are distributed uniformly between $30\,\mathrm{m}$ and $60\,\mathrm{m}$, thus on average slightly below the reference distance. In order to ensure sufficient distance after merging the two lanes, significant control action is required.

B. Results and Discussion

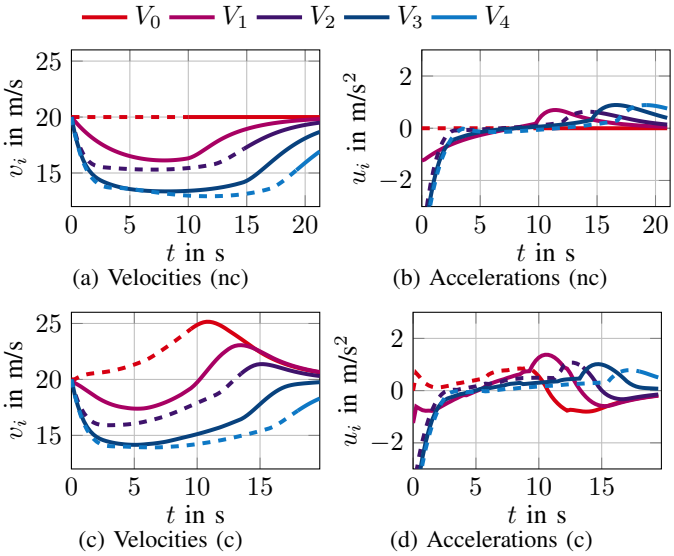
Simulation is used to verify the safety guarantees of the control algorithm and to demonstrate the improvement with cooperative cost functions in comparison to the same controller without cooperative behavior. The former is shown by the fact that in the course of the simulations no infeasibilities occurred during the control.

The line plots in Figure 3 show the state and input trajectories of each vehicle in the respective vehicle color, where the line is drawn dashed for vehicles in the merging lane L_1 until merging to the main lane L_0 and vehicles in L_0 are drawn as solid lines.

Cooperative Control. For the simulation of cooperative control as described in Section III-G, the cost function weights

TABLE I: Parameter values.

name	value	name	value	name	value
s_m	$200\,\mathrm{m}$	u_min	$-10\,\mathrm{m/s^2}$	M	5
s_out	$230\,\mathrm{m}$	u_max	$10\,\mathrm{m/s^2}$	N	60
v_min	$0\,\mathrm{m/s}$	d_min	$10\,\mathrm{m}$	p	$7 \cdot 10^{-4}$
v_max	$35\,\mathrm{m/s}$	d_r	$50\,\mathrm{m}$	q	$8.2 \cdot 10^{-4}$
v_r	$20\,\mathrm{m/s}$	t_d	$2\,\mathrm{s}$	r	10^{-2}
$v_i(0)$	$20\,\mathrm{m/s}$	t_s	$0.25\,\mathrm{s}$		



(a) Velocities (nc) (b) Accelerations (nc)
(c) Velocities (c) (d) Accelerations (c)

Fig. 3: Comparison of velocity/acceleration trajectories with cooperative (c) vs. non-cooperative (nc) control approach.

$\omega_\mathrm{i} = 1$, $\omega_\mathrm{n} = p$ and $\omega_\mathrm{o} = 5p$ are defined. A simulation with cooperative cost functions and a simulation with non-cooperative cost functions ($\omega_\mathrm{n} = \omega_\mathrm{o} = 0$) and besides that identical parameterizations are compared in Figure 3 to show the effect of cooperative behavior. In the top row, the first vehicle's constant velocity $v_0(t) = v_\mathrm{r}$ necessitating a significant reduction in velocity of the following vehicles is illustrated. The cooperative cost functions used in the simulation shown in the bottom row lead to an increase in velocity above the reference velocity of the front vehicles, thus leading to a more moderate decrease in velocity of the following vehicles. In order to revert to the reference velocity a deceleration is required for vehicle V_0, which is shown in the bottom right plot.

The bar chart in Figure 4 shows the averaged values of the individual vehicle costs over 10 simulations, where the actual trajectories of the vehicles over the entire simulation period are inserted into (28). The impact of the lane merge lies disproportionately on the rearmost vehicles in the non-cooperative case. The application of cooperative cost functions leads to a more even distribution of the individual costs among the vehicles, with a reduction of the cumulated individual costs by 19.9 %. Additionally, the average length of time from the first vehicle entering the control zone until the rearmost vehicle passes s_out is reduced by 6.1 %, from 21.3 s to 20.0 s.

V. Summary and Outlook

Lane merging situations are ubiquitous in urban traffic and contribute to congestion and potential accidents. In this article, a distributed model predictive control framework is proposed to induce an anticipatory and cooperative behavior. Formal safety guarantees, such as collision avoidance, are achieved by means of recursive feasibility. In our approach, the communication between the involved vehicles is limited to the direct neighbors, which implies scalability even for large convoys. In a numerical example, we illustrate how

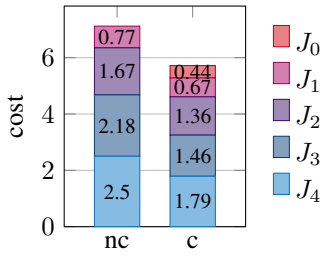


Fig. 4: Comparison of individual vehicle costs with cooperative (c) vs. non-cooperative (nc) control approach.

the overall cost can be reduced by nearly 20 % using cooperative cost functions. Future research directions are distributed stochastic or robust model predictive control in order to extend the control design to disturbances and model uncertainties, see, e.g., [18].

Appendix

A. Proof of Theorem 3.1

In the following, the proof of Theorem 3.1 is shown. *Proof:* Given a feasible optimal control sequence $u_i^\star(k:k+N-1|k)$ and feasible state and slack trajectories $\boldsymbol{x}_i^\star(k:k+N|k)$, $\rho_{i,i^-}^\star(k:k+N-1|k)$, $\rho_{i,i-1}^\star(k:k+N-1|k)$ at time step k for all $i \in \{0,\dots,M-1\}$ for OCP (16), a candidate solution can be recursively constructed that is a feasible solution for OCP (16) at time step $k+1$ for all $i \in \{0,\dots,M-1\}$.

The candidate input sequence for the optimization at time step $k+1$ is defined as

$$\hat{u}_i(l|k+1) = \begin{cases} u_i^\star(l|k), & k < l < k+N, \\ \boldsymbol{K}_{\mathrm{f},\mathcal{N}_i} \hat{\boldsymbol{z}}_{\mathcal{N}_i}(l|k+1), & l = k+N. \end{cases} \tag{32}$$

The appended candidate value for $l = k+N$ is generated using the terminal feedback. The candidate state sequence in vehicle coordinates is

$$\hat{\boldsymbol{x}}_i(l|k+1) = \boldsymbol{x}_i^\star(l|k), \quad k < l \leq k+N. \tag{33}$$

The slack variables' candidate solutions are

$$\hat{\rho}_{i,i^-}(l|k+1) = \hat{\rho}_{i,i-1}(l|k+1) = t_\mathrm{d} v_i^\star(l|k) \tag{34}$$

for $k < l \leq k+N$. In error coordinates, a subsystem's state vector comprises two vehicles' states. The candidate solution of subsystem i is defined as

$$\hat{\boldsymbol{z}}_i(l|k+1) = \boldsymbol{z}_i^\star(l|k^+), \quad k < l \leq k+N, \tag{35}$$

where the optimized trajectory from vehicle V_{i-1} at time $k+1$ and optimal data from vehicle V_i at time k are used, see (25). The candidate solution of the rear error subsystem used in constraint (16e) does not include newly optimized values with

$$\hat{\boldsymbol{z}}_{i+1}(l|k+1) = \boldsymbol{z}_{i+1}^\star(l|k), \quad k < l \leq k+N. \tag{36}$$

The new appended terminal value of the state candidate solution is computed in error coordinates as

$$\hat{\boldsymbol{z}}_i(k+N+1|k+1) = \tilde{\boldsymbol{A}}_{\mathrm{K_f},\mathcal{N}_i} \hat{\boldsymbol{z}}_{\mathcal{N}_i}(k+N|k+1) \tag{37}$$

$$= \tilde{\boldsymbol{A}}_{\mathrm{K_f},\mathcal{N}_i} \boldsymbol{z}_{\mathcal{N}_i}^\star(k+N|k^+). \tag{38}$$

The candidate solutions are verified regarding constraints (16g), (16h), (16c), (16d) and (16e).

First, constraint (16g) is examined for all $k < l \leq k+N$. The constraint holds for $k < l < k+N$ with $\hat{u}_i(l|k+1) = u_i^\star(l|k) \in \mathcal{U}_i$. If constraint (16d) is fulfilled for V_i and V_{i-1}, then it holds that $\hat{\boldsymbol{z}}_i(k+N|k+1) \in \mathcal{Z}_{\mathrm{f},i}(k)$ and $\boldsymbol{z}_{i-1}^\star(k+N|k+1) \in \mathcal{Z}_{\mathrm{f},i-1}(k)$. Then $\hat{u}_i(k+N|k+1) = \tilde{\boldsymbol{K}}_{\mathrm{f},\mathcal{N}_i}\hat{\boldsymbol{z}}_{\mathcal{N}_i}(k+N|k+1) \in \mathcal{U}_i$ holds, due to (25) and constraint satisfaction in the invariant set according to Definition 3.1.

Constraint set (16h) comprises individual constraints (18a) to (18h). (18a) is fulfilled with $\hat{v}_i(l|k+1) = v_i^\star(l|k),\ k < l \leq k+N$, since $v_i^\star(l|k) \in \mathcal{X}_i'$ for $k < l \leq k+N-1$ and $v_i^\star(k+N|k) \in \mathcal{Z}_{\mathrm{f},i}(k)$, where the constraint on velocity holds according to Definition 3.1 and constraint set (15).

Switching constraint (18e) states $s_{i-1}^\star(l|k+1) - \hat{s}_i(l|k+1) \geq d_{\min} + t_{\mathrm{d}}\hat{v}_i(l|k+1) - \hat{\rho}_{i,i-1}(l|k+1)$, if $s_{i-1}^\star(l|k+1) \geq s_{\mathrm{m}}$. The other case does not restrict the position. The candidate solution is inserted to yield $s_{i-1}^\star(l|k+1) - s_i^\star(l|k) \geq d_{\min} + t_{\mathrm{d}}v_i^\star(l|k) - t_{\mathrm{d}}v_i^\star(l|k) = d_{\min}$. From feasibility of the OCP of vehicle V_{i-1} at time step $k+1$, it follows that $s_{i-1}^\star(l|k+1) - s_i^\star(l|k) \geq d_{\min}$, if $s_i^\star(l|k) > s_{\mathrm{m}} - d_{\min}$ as it is required in (18h). A violation of (18e) would mean $s_{i-1}^\star(l|k+1) - s_i^\star(l|k) < d_{\min}$ and $s_{i-1}^\star(l|k+1) \geq s_{\mathrm{m}}$ are fulfilled at the same time. Rearranged for $s_i^\star$, this yields $s_i^\star(l|k) > s_{i-1}^\star(l|k+1) - d_{\min} > s_{\mathrm{m}} - d_{\min}$. This activates constraint (18h) of vehicle V_{i-1}, requiring $s_{i-1}^\star(l|k+1) - s_i^\star(l|k) \geq d_{\min}$ and thus preventing a constraint violation.

Constraint (18f) is fulfilled with $\hat{\rho}_{i,i-1}(l|k+1) = t_{\mathrm{d}}v_i^\star(l|k) \leq t_{\mathrm{d}}\hat{v}_i(l|k+1) = t_{\mathrm{d}}v_i^\star(l|k),\ k < l \leq k+N$.

Constraint (18g) with the candidate solution inserted states $\hat{s}_i(l|k+1) = s_i^\star(l|k) \leq s_{\mathrm{m}} - d_{\min}$, if $s_{i-1}^\star(l|k) \leq s_{\mathrm{m}}$. As above, constraint (18h) in the OCP of vehicle V_{i-1} is formulated as $s_{i-1}^\star(l|k+1) - s_i^\star(l|k) \geq d_{\min}$, if $s_i^\star(l|k) > s_{\mathrm{m}} - d_{\min}$. Therefore, if $s_i^\star(l|k) > s_{\mathrm{m}} - d_{\min}$, then rearranged constraint (18h) of V_{i-1} yields $s_{i-1}^\star(l|k+1) \geq s_i^\star(l|k) + d_{\min} > s_{\mathrm{m}} - d_{\min} + d_{\min} = s_{\mathrm{m}}$. Thus, $s_i^\star(l|k) > s_{\mathrm{m}} - d_{\min}$ implies $s_{i-1}^\star(l|k+1) > s_{\mathrm{m}}$ and (18g) is satisfied for $k < l \leq k+N$.

Constraint (18h) requires $\hat{s}_i(l|k+1) - s_{i+1}^\star(l|k) = s_i^\star(l|k) - s_{i+1}^\star(l|k) \geq d_{\min}$, if $s_{i+1}^\star(l|k) > s_{\mathrm{m}} - d_{\min}$. For verification, constraint (18e) in the OCP of vehicle V_{i+1} at time step k, using constraint (18f) to reduce the slack variables, is formulated as $s_i^\star(l|k) - s_{i+1}^\star(l|k) \geq d_{\min} + t_{\mathrm{d}}v_{i+1}^\star(l|k) - \rho_{i+1,i}^\star(l|k) \geq d_{\min}$, if $s_i^\star(l|k) \geq s_{\mathrm{m}}$. Additionally, constraint (18g) of vehicle V_{i+1} is formulated as $s_{i+1}^\star(l|k) \leq s_{\mathrm{m}} - d_{\min}$, if $s_i^\star(l|k) \leq s_{\mathrm{m}}$. Therefore, $s_{i+1}^\star(l|k) > s_{\mathrm{m}} - d_{\min}$ is only possible if $s_i^\star(l|k) > s_{\mathrm{m}}$, which leads to constraint $s_i^\star(l|k) - s_{i+1}^\star(l|k) \geq d_{\min}$ of vehicle V_{i+1}, which verifies constraint (18h) of vehicle V_i for $k < l \leq k+N-1$. The constraint is fulfilled at $l = k+N$, due to $\boldsymbol{z}_j^\star(k+N|k) \in \mathcal{Z}_{\mathrm{f},j}(k)\ \forall j \in \{0, \ldots, M-1\}$, where the minimum distance constraint holds between all vehicles. The compliance of constraints (18b), (18c) and (18d) can be proven similarly.

Constraint (16c) requires $\hat{\boldsymbol{z}}_i(k+N+1|k+1) \in \mathcal{Z}_{\mathrm{f},i}(k+1)$. Using definition (37) and (27) in combination with (26), this holds true if $\boldsymbol{z}_{i-1}^\star(k+N|k+1) \in \mathcal{Z}_{f,i-1}(k)$ and $\boldsymbol{z}_i^\star(k+N|k^+) \in \mathcal{Z}_{\mathrm{f},i}(k)$ are fulfilled. From feasibility of the OCP of vehicle V_{i-1} at time $k+1$ follows $\boldsymbol{z}_{i-1}^\star(k+N|k+1) \in \mathcal{Z}_{f,i-1}(k)$ from constraint (16d). The second condition is fulfilled through constraint (16e) in the OCP of vehicle V_{i-1}, which states $\boldsymbol{z}_i(k+N|k+1) \in \mathcal{Z}_{\mathrm{f},i}(k)$. After optimization of V_{i-1}, this yields $\boldsymbol{z}_i^\star(k+N|k^+) \in \mathcal{Z}_{\mathrm{f},i}(k)$, see (25).

Constraint (16d) requires $\hat{\boldsymbol{z}}_i(k+N|k+1) = \boldsymbol{z}_i^\star(k+N|k^+) \in \mathcal{Z}_{\mathrm{f},i}(k)$. This follows from feasibility of (16e) in the OCP of vehicle V_{i-1} for $k+1$ as shown in the paragraph above.

Constraint (16e) requires $\hat{\boldsymbol{z}}_{i+1}(k+N|k+1) = \boldsymbol{z}_{i+1}^\star(k+N|k) \in \mathcal{Z}_{\mathrm{f},i+1}(k)$. This holds due to feasibility of OCP of vehicle V_{i+1} at time step k w.r.t. constraint (16c) which yields $\boldsymbol{z}_{i+1}^\star(k+N|k) \in \mathcal{Z}_{\mathrm{f},i+1}(k)$. This completes the proof. ■

REFERENCES

[1] B. Van Arem, C. Van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on intelligent transportation systems*, 2006.

[2] I. Llatser, T. Michalke, M. Dolgov, F. Wildschütte, and H. Fuchs, "Cooperative automated driving use cases for 5G V2X communication," in *Proc. IEEE 2nd 5G World Forum*, 2019.

[3] P. Mundhenk, A. Hamann, A. Heyl, and D. Ziegenbein, "Reliable distributed systems," in *Proc. Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2022.

[4] V. Bhattacharyya and A. Vahidi, "Automated Vehicle Highway Merging: Motion Planning via Adaptive Interactive Mixed-Integer MPC," in *Proc. American Control Conference (ACC)*, 2023.

[5] W. Cao, M. Mukai, and T. Kawabe, "Merging trajectory generation method using real-time optimization with enhanced robustness against sensor noise," *Artificial Life and Robotics*, 2019.

[6] W. Cao, M. Muka, T. Kawabe, H. Nishira, and N. Fujiki, "Merging trajectory generation for vehicle on a motor way using receding horizon control framework — consideration of its applications," in *Proc. IEEE Conference on Control Applications*, 2014.

[7] J. Lubars, H. Gupta, S. Chinchali, L. Li, A. Raja, R. Srikant, and X. Wu, "Combining Reinforcement Learning with Model Predictive Control for On-Ramp Merging," in *Proc. IEEE International Intelligent Transportation Systems Conference*, 2021.

[8] M. Mukai, H. Natori, and M. Fujita, "Model predictive control with a mixed integer programming for merging path generation on motor way," in *Proc. IEEE Conference on Control Technology & Applications*, 2017.

[9] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transportation Research C: Emerging Technologies*, 2016.

[10] S. H. Wang, M. Zhao, D. H. Sun, and X. Liu, "A Merging Strategy Based on Optimal Control of Main-lane Downstream and On-ramp Vehicles," *KSCE Journal of Civil Engineering*, 2022.

[11] Q. Zhang, R. Langari, H. Tseng, D. Filev, S. Szwabowski, and S. Coskun, "A Game Theoretic Model Predictive Controller With Aggressiveness Estimation for Mandatory Lane Change," *IEEE Transactions on Intelligent Vehicles*, 2020.

[12] S. Blasi, M. Kögel, and R. Findeisen, "Distributed Model Predictive Control Using Cooperative Contract Options," in *Proc. IFAC Conference on Nonlinear Model Predictive Control*, 2018.

[13] X. Chen and J. Mårtensson, "Heterogeneous Traffic Intersection Coordination Based on Distributed Model Predictive Control with Invariant Safety Guarantee," in *Proc. IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2022.

[14] C. Conte, N. Voellmy, M. Zeilinger, M. Morari, and C. Jones, "Distributed synthesis and control of constrained linear systems," in *Proc. American Control Conference (ACC)*, 2012.

[15] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[16] J. Löfberg, "YALMIP: A toolbox for modeling and optimization in Matlab," in *Proc. IEEE Conference on Robotics and Automation*, 2004.

[17] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 10.1.*, 2024. [Online]. Available: http://docs.mosek.com/latest/toolbox/index.html

[18] F. Berkel and S. Liu, "An event-triggered cooperation approach for robust distributed model predictive control," *IFAC Journal of Systems and Control*, vol. 6, pp. 16–24, 2018.