Any-Time Feasible Coordination for Multi-Vehicle Systems

Maximilian Kneissl, Adam Molin, Hasan Esen, and Sandra Hirche, *Fellow, IEEE*

***Abstract*—This article proposes a distributed control methodology for the coordination of multiple connected and automated vehicles (CAVs). Each vehicle computes its local trajectory based on a model predictive control (MPC) law and communicates the result with other relevant vehicles. In an iterative negotiation process, planned trajectories are optimized within a sampling time step. Inspired by the Jacobi over-relaxation (JOR) algorithm, we develop a distributed Jacobi over-relaxation algorithm (DJOR) for vehicle coordination. The modified algorithm exploits the structure of the distributed problem setting in which coupling occurs only in a bilateral way. Besides being able to guarantee any-time feasibility that implies collision-freeness, the algorithm scales well, unlike the standard JOR algorithm. The DJOR algorithm allows for significantly less conservatism in the choice of the update weightings. As a result, much faster convergence rates can be expected. Furthermore, the collision avoidance guarantee is extended for unforeseen scenarios such as emergency braking. Using an exact penalty function formulation ensures that the distributed optimization problem remains feasible even in previously unforeseen cases. Numerical simulations of an intersection crossing scenario illustrate the presented approach and show its benefits in comparison with standard traffic rules and a centralized computation.**

***Index Terms*—Distributed model predictive control, Multi-vehicle systems, Environmental uncertainty, Feasibility, Jacobi algorithm, Scalability.**

I. Introduction

THE rapid development of autonomous cars reveals a huge potential for increasing traffic safety, efficiency, and comfort. A significant contribution to these benefits arises from the ability of connected and automated vehicles (CAV) to share intended driving data with surrounding traffic participants. Substantial scenarios in order to coordination multiple CAVs are platooning [1], merging [2], intersection crossing [3], and navigating in parking environments [4]–[6]. The control community has worked intensively on solutions to these challenges. Often optimization-based methods (in a distributed fashion) are utilized to solve the coordination problem. Due to computational expensive solutions or a high amount of information exchange, most of the existing approaches lack in real-world applicability. To achieve this, the methods have to be safe and efficient in their intention, feasible to compute and communicate, as well as easy to implement.

M. Kneissl, A. Molin, and H. Esen are with the Corporate R&D department of DENSO Automotive Deutschland GmbH, Freisinger Str. 21-23, 85386 Eching, Germany {m.kneissl,a.molin,h.esen}@denso-auto.de

[2]S. Hirche and M. Kneissl are with the Institute for Information- oriented Control, Technische Universität München, Arcisstrae 21, 80290 München, Germany hirche@tum.de

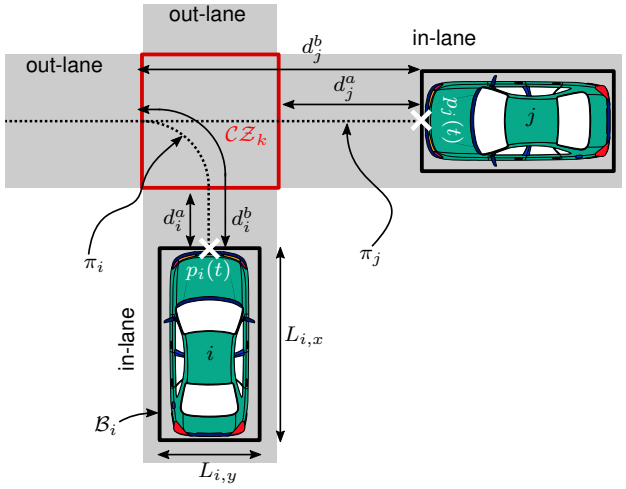

Fig. 1: Coordination Scenario.

Taking this into account, we propose a distributed trajectory negotiation method for vehicles with decoupled heterogenous dynamics and inter-vehicle coupling constraints. While following a pre-computed reference path, vehicles negotiate their intended trajectories with surrounding vehicles. Each vehicle applies online model predictive control (MPC) optimization problems for efficient driving decisions. These trajectory decisions are shared with neighbor vehicles and taken into account for consecutive decisions. A local update-step modification of the trajectories before sharing ensures system-wide feasible computations at any-time of the negotiation process. This implies a collision-free and communication-efficient solution with fast convergence. The local updates guarantee a scalabel and easy to implement methodology considering real-world applications. Proposed trajectories refer geometrically to common areas, named conflict zones. An example is the case of intersecting lanes, as illustrated in Fig. 1. The negotiation process is fully distributed between vehicles, while we assume a given decision on the crossing sequence at conflict zones.

A. Related Work

Significant contributions for multi-vehicle coordination have been proposed by solving the problem of automated intersection crossing. Initially, [7] suggested a protocol where vehicles approaching an intersection send a reservation request to an intersection manager. If access is granted, they cross according to the plan. If the previous reservation request was denied, they resubmit the request. Optimization-based control, in particular distributed MPC, is a relevant method to solve multi-vehicle

coordination problems, as it enables combined consideration of dynamical vehicle and safety constraints, as well as global coordination performance interests. Therefore, it has been widely applied in the literature, e.g., [8]–[10]. Coordination problems of several autonomous vehicles can be solved by restricting a local vehicle's position space with respect to an intersection area. The method proposed in [11] enables iteration-free and parallel computation for intersection crossing which is extended towards more general coordination scenarios in [12]. Alternatively, it has been proposed to coordinate vehicles referring to predicted crossing times of certain conflict areas. While authors of [13] compute analytical solutions to vehicles' exit times of intersection areas using Hamiltonian functions, authors of [14] compute occupancy time slots using constraint optimal control problems. Above literature and the problem setup of multi-vehicle scenarios suggest the applicability of distributed control strategies to these problems. A general overview of distributed optimization methods is presented in [15]. Guaranteeing feasibility of distributed optimization computations is in general not a trivial task. Feasibility in such setups can be ensured through sequential computations [16], [17], robust MPC formulations [18], or iterative algorithms [19], [20]. These approaches commonly come with the drawbacks of lacking in scalability, tending to conservative solutions, and resulting in communication extensive processes, respectively. The proposed methods in this article aim at overcoming these drawbacks to ensure applicability.

Unlike vehicle platooning or similar cooperative tasks with a fixed vehicle topology, coordination scenarios require a combinatoric vehicle sequence decision, i.e., in which order vehicles are sorted to merge or cross. Often, the sequence decision is assumed to be pre-defined, or simple heuristic rules are applied, such as first-come-first-served. In contrast, [21] discusses a modification of a consensus-based auction algorithm to suggest a crossing sequence for vehicles at intersections. [22] introduces heuristics in the framework of priority graphs, which encode vehicle sequences that are proven to be deadlock-free. Authors of [23] use mixed integer quadratic programs (MIQPs) to determine an approximate of the optimal crossing sequence. While pure huristic solutions can lead to significantly sub-optimal solutions, optimization-based methods quickly tend to be not implementable for large-scale systems. The approach in this article also depends on a pre-defined vehicle sequence decision. Therefore, we apply a scheduling problem (resource-constraint-project-scheduling) to compute this sequence, which has been introduced in [24] and suggests a trade-off between optimality and scalability of the solution.

A crucial property of multi-vehicle coordination is to guarantee safety even in the case of unforeseen scenarios. Therefore, it is common to plan with additional brake-safe distance to avoid collision with other traffic participants, e.g., proposed by [25] in a decentralized MPC setup, and through a safe velocity interval formulation without predictive methods in [26]. In [27] the authors propose a robust MPC formulation accounting for worst case actions of other vehicles in an intersection crossing scenario. Alternatively, [28], [29] design least restrictive supervisors using scheduling to intervene only if a driver would enter an unsafe set with his own action. A remaining challenge is how to handle the reaction to unforeseen events in distributed optimization for constraint-coupled problems. Even if a brake-safe distance is considered, a violation of a hard coupling-constraint due to an unforeseen event leads to infeasibility of the optimization problems.

B. Contributions

This article contributes by presenting a holistic multi-vehicle coordination approach with rear-end and side collision avoidance independent of a vehicle's prediction horizon length. The distributed iterative negotiation process based on Jacobi updates is any-time feasible, i.e., can be interrupted after each iteration with a guaranteed network-wide feasible solution. The distributed computational load and the possibility to stop iterating at any time are important properties for real-world implementations, which becomes clearly visable in experiments [30]. A Jacobi over-relaxation (JOR) decomposition for input-constraint distributed MPC problems is discussed in [31], which has been applied to general constraint optimization problems in [32] and to multi-vehicle coordination problems in [33]. We extend the JOR to a distributed Jacobi over-relaxation (DJOR) algorithm, which scales, different to the JOR, independently of the number of participating vehicles and iterates fully distributed, i.e., without the necessity of a central update step. Additionally, we extend previous work by ensuring collision avoidance for unforeseen events through prioritizing safety over the coordination process using exact penalty functions.

The remainder of this article is organized as follows. Section II introduces the coordination formulation and the open-loop optimal control problems. Thereafter, the iterative DJOR algorithm is introduced in Section III where also any-time feasibility and scalability properties are derived. Section IV extends the approach with the ability of handling uncertain environments, and Section V discusses numerical examples applying the proposed methodology.

Notation: Throughout this article $x(k|t)$ indicates a prediction of state x for time k computed at time t, and $x(:\,|t)$ refers to all elements of this vector. The set of integers $\mathbb{I}_{a:b}$ is defined by $\{a, a+1, ..., b\}$, and $\mathbb{I}_{a:b} = \varnothing$ if $a > b$. $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_\infty$ are the 1-, 2-, and infinity-norm, respectively. The weighted 2-norm is denoted by $\|x-\hat{x}\|_Q^2 = (x-\hat{x})^\mathsf{T} Q (x-\hat{x})$ with appropriate dimensions of vectors $x, \hat{x}$ and matrix Q. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$, and the i-th element of a vector v is indexed with $[v]_i$. $\{v_i\}_{i\in\mathcal{S}}$ is a set collecting all vectors v_i with indices i contained in the set $\mathcal{S}$.

II. Multi-vehicle Coordination Formulation

In this section, we introduce the multi-vehicle coordination model. First, we define conditions for inter-vehicle collision avoidance, and thereafter we state a central optimization problem including these conditions for the complete coordination scenario. Lastly, we decompose the centralized problem into local optimization problems which can be applied in each vehicle's control unit.

Throughout the article, we make the following assumptions. i) Local optimization problems are assumed to be pre-informed about the combinatoric vehicle sequence decisions, which result in a given communication structure of respective vehicles in the scenario. In the article, this is envisioned to be provided by a central infrastructure node as illustrated in Fig. 3 and discussed in [24]. Alternatively, distributed sequence decisions could replace the central infrastructure node.

ii) At the time a vehicle joins the distributed negotiation process, discussed in Section III, we assume that if the vehicle is traveling behind another vehicle it keeps a brake-safe distance to its predecessor. This is a natural assumption, as also human drivers would behave in the same way. The brake-safe distance will only be violated in unpredicted scenarios which for instance cause emergency braking. This is formalized in Assumption 1.

iii) During the negotiation process predicted vehicle trajectories will be referenced to geometrical zones on the road network (conflict zones $\mathcal{CZ}$). It is assumed that at the time a vehicle joins the distributed negotiation process it is able to propose a feasible trajectory that would lead to a full-stop before entering the respective zone. This requires to start the inter-vehicle communication with enough distance to such zones and is formalized in Assumption 2.

A. Collision Avoidance Condition

Assume a set of vehicles $\mathcal{V} = \{1, 2, ..., N_v\}$ moving along pre-defined paths on a road network. The path of Vehicle $i \in \mathcal{V}$ is described by a set of N_{W_i} discrete waypoints $W_i = \{(p_x^1, p_y^1)^T, ..., (p_x^{N_{w_i}}, p_y^{N_{w_i}})^T\}$ with positions in the 2D $x-y-$plane. The reference point of the vehicle is assumed to be located at the center front of the vehicle. Let $\pi_i : I \rightarrow \mathbb{R}^2$, with interval $I = [1, N_{W_i}]$, describe a $\mathcal{C}^1$-curve which connects all waypoints W_i. Furthermore, assume that π_i can be perfectly tracked by Vehicle i.

Fig. 1 illustrates the coordination scenario for two vehicles, as introduced in the following. We represent a vehicle by a rectangular bounding-box $\mathcal{B}_i(p_i(t), L_{i,x}, L_{i,y})$ with parameters $p_i(t) \subset \pi_i$, which is the vehicle's time-dependent position on its path, as well as $L_{i,x}$ and $L_{i,y}$, which describe the length and width of the box (compare Fig. 1), respectively. By variable $t \in \mathbb{N}_0$ we denote the discrete time scale. Given the path and vehicle representation, we define Vehicle i's distance state $d_i^n(t)$ with the arc-length from a curve segment of π_i, such that

$$d_i^n(t) = \begin{cases} \int_m^n \parallel \pi_i^{'}(s) \parallel_2 \, ds, \text{ if } m \xrightarrow{\pi_i} n \\ -\int_m^n \parallel \pi_i^{'}(s) \parallel_2 \, ds, \text{ if } n \xrightarrow{\pi_i} m, \end{cases} \tag{1}$$

with $\pi_i(m) = p_i(t)$, $\pi_i(n) \in W_i$ a target reference waypoint on the curve, and $\pi_i^{'}(s)$ the derivative of π_i at s. Notation $m \xrightarrow{\pi_i} n$ means that $\pi_i(m)$ is located before $\pi_i(n)$ following the curve in driving direction and $n \xrightarrow{\pi_i} m$ accordingly. In other words, $d_i^n(t)$ describes the distance from the current vehicle position to a certain point n measured along the path. For example, in Fig. 1 $d_i^a(t)$, which is described formally below, is the distance from the vehicle to the beginning of the conflict zone $\mathcal{CZ}_k$.

Collisions between vehicles are avoided if

$$\mathcal{B}_i(t) \cap \mathcal{B}_j(t) = \varnothing, \ \forall t, \ i, j \in \mathcal{V}, \ i \neq j. \tag{2}$$

Areas where vehicle paths intersect, merge, and diverge are essential for coordinating vehicles on a road network. This is commonly the case at intersections, where condition (2) leads to areas in the 2D plane which can be accessed only by a single vehicle at a time. We refer to these zones as conflict zones $\mathcal{CZ}_i \subset \mathbb{R}^2$, for $i \in \mathbb{I}_{1:N_{cz}}$ with a total number of conflict zones N_{cz} in the coordination space. These conflict zones will serve as reference areas to guarantee a safe multi-vehicle coordination. Now, let Vehicle i crosses $\mathcal{CZ}_k$ before Vehicle j, then a collision avoidance condition with regard to the vehicles' distance states is

$$d_i^b(t) + L_{i,x} + d_{j,s} \leqslant d_j^a(t). \tag{3}$$

Here, $d_{j,s}$ is a scenario-dependent safety distance for Vehicle j defined in (7), $d_i^b(t)$ is the distance of Vehicle i to the exit of $\mathcal{CZ}_k$, defined according to (1), with b describing the first waypoint of W_i outside of $\mathcal{CZ}_k$ in driving direction, i.e.,

$$\begin{aligned} b = \min_{q \in \mathbb{I}_{1:N_{w_i}}} \quad & q \\ \text{s.t.} \quad & \hat{q} \leqslant q \leqslant N_{W_i} \\ & (p_x^q, p_y^q)^T \notin \mathcal{CZ}_k \\ & (p_x^{\hat{q}}, p_y^{\hat{q}})^T \in \mathcal{CZ}_k. \end{aligned} \tag{4}$$

Similarly, $d_j^a(t)$ is the distance of Vehicle j to the entrance of $\mathcal{CZ}_k$, i.e.,

$$\begin{aligned} a = \max_{q \in \mathbb{I}_{1:N_{w_i}}} \quad & q \\ \text{s.t.} \quad & 1 \leqslant q \leqslant \hat{q} \\ & (p_x^q, p_y^q)^T \notin \mathcal{CZ}_k \\ & (p_x^{\hat{q}}, p_y^{\hat{q}})^T \in \mathcal{CZ}_k. \end{aligned} \tag{5}$$

We assume that from both $d_i^b(t) + L_{i,x} \leqslant 0$ and $d_i^a(t) \geqslant 0$ follows $\mathcal{B}_i(t) \cap \mathcal{CZ}_k = \varnothing$. Thus, we neglect the possible skew of a vehicle in this notation. The simplification can be compensated by designing $L_{i,x}$ or the size of $\mathcal{CZ}_k$ considering appropriate spare dimensions. Let t_i^b be the time-instant just after Vehicle i leaves $\mathcal{CZ}_k$ (exit-time), such that

$$d_i^b(\tau_i^b) = -L_{i,x} \text{ and } t_i^b = \left\lceil \frac{\tau_i^b}{T_s} \right\rceil \tag{6}$$

holds, where $\tau_i^b \in \mathbb{R}_0^+$ is the exact continuous time solution and T_s the discrete sampling time.

Given the fact that vehicles move on pre-defined lanes on a road network, a time-dependent validity of constraints on the distance state can be specified dependent on different maneuvers. Therefore, we define the following four cases which distinguish the directions in which Vehicles i and j approach $\mathcal{CZ}_k$ (in-lane) and in which they leave it (out-lane) in the first row, the time instances for which the distance constraints need to be considered in the second row, and the definition of safety distance in (3) in the third row:

$$\textbf{c1:}\ \begin{cases}\text{from } \textbf{same} \text{ in-lane to } \textbf{same} \text{ out-lane}\\ \text{(3) holds } \forall t\\ d_{j,s} = d_{j,stop} - d_i^b(t) + d_i^a(t)\end{cases} \tag{7a}$$

$$\textbf{c2:}\ \begin{cases}\text{from } \textbf{same} \text{ in-lane to } \textbf{different} \text{ out-lane}\\ \text{(3) holds for } t < t_i^b\\ d_{j,s} = d_{j,stop} - d_i^b(t) + d_i^a(t)\end{cases} \tag{7b}$$

$$\textbf{c3:}\ \begin{cases}\text{from } \textbf{different} \text{ in-lane to } \textbf{same} \text{ out-lane}\\ \text{(3) holds for } t \geqslant t_i^b \ \vee\ d_j^a(t) > d_{j,s} \text{ holds for } t < t_i^b\\ d_{j,s} = d_{j,stop}\end{cases} \tag{7c}$$

$$\textbf{c4:}\ \begin{cases}\text{from } \textbf{different} \text{ in-lane to } \textbf{different} \text{ out-lane}\\ d_j^a(t) > d_{j,s} \text{ holds for } t < t_i^b\\ d_{j,s} = d_{j,stop},\end{cases} \tag{7d}$$

with $d_{j,stop}$ Vehicle j's stopping distance, discussed in Section IV. The second row in Fig. 2 illustrates exemplary geometric configuration for each of the four cases above.

B. Centralized Coordination Model

In this subsection, we state the coordination task as a centralized optimization problem considering longitudinal vehicle movements along pre-defined paths modeled by linear dynamics. For a single vehicle these time-discrete dynamics are

$$\underbrace{\begin{pmatrix} d_i \\ v_i \end{pmatrix}^+}_{x_i(t+1)} = \underbrace{\begin{pmatrix} 1 & -T_s \\ 0 & 1 \end{pmatrix}}_{A_i \in \mathbb{R}^{2\times 2}} \underbrace{\begin{pmatrix} d_i \\ v_i \end{pmatrix}}_{x_i(t)} + \underbrace{\begin{pmatrix} -T_s^2 \\ T_s \end{pmatrix}}_{B_i \in \mathbb{R}^{2\times 1}} \underbrace{a_i}_{u_i(t)}, \tag{8}$$

with distance state $d_i = d_i^{N_{W_i}}$ representing the distance to the end of Vehicle i's path, v_i its velocity, and a_i its acceleration input.

The centralized system is achieved by concatenating individual vehicle models, such that we achieve the system and input matrix

$$A = \begin{pmatrix} A_1 & & & \\ & A_2 & & \\ & & \ddots & \\ & & & A_{N_v} \end{pmatrix}, B = \begin{pmatrix} B_1 & & & \\ & B_2 & & \\ & & \ddots & \\ & & & B_{N_v} \end{pmatrix}, \tag{9}$$

and the state and input vectors

$$x(t) = \begin{pmatrix} x_1(t) \\ \vdots \\ x_{N_v}(t) \end{pmatrix} \text{ and } u(t) = \begin{pmatrix} u_1(t) \\ \vdots \\ u_{N_v}(t) \end{pmatrix}, \tag{10}$$

respectively. Now, we formulate the centralized coordination task as a finite horizon optimization problem

$$V^* = \min_{\bar{x},\bar{u}} \ \sum_{i=1}^{N_v} V_i \tag{11a}$$

s.t.

$$x(k+1|t) = Ax(k|t) + Bu(k|t) \qquad k \in \mathbb{I}_{t:t+M-1} \tag{11b}$$

$$x(t|t) = x(t) \tag{11c}$$

$$x(k|t) \in \mathbb{X} \qquad k \in \mathbb{I}_{t+1:t+M} \tag{11d}$$

$$u(k|t) \in \mathbb{U} \qquad k \in \mathbb{I}_{t:t+M-1} \tag{11e}$$

$$d_j^a(k|t) - d_{j,stop} \geqslant 0 \qquad \begin{cases}\textbf{c3}, \textbf{c4}\\ (i,j) \in \mathcal{T}_m\end{cases} \tag{11f}$$

$$d_i^b(k|t) + L_{i,x} + d_{j,s} - d_j^a(k|t) \leqslant 0 \qquad \begin{cases}\textbf{c1}, \textbf{c2}, \textbf{c3}\\ (i,j) \in \mathcal{T}_m\end{cases} \tag{11g}$$

$$m \in \mathbb{I}_{1:|\mathcal{T}|}, \tag{11h}$$

where the optimization variables are defined by

$$\bar{x} = (x(t+1|t)^\mathsf{T}, ..., x(t+M|t)^\mathsf{T}), \tag{12}$$

and

$$\bar{u} = (u(t|t), ..., u(t+M-1|t)), \tag{13}$$

for prediction horizon M. The objective function (11a) is the sum of the local vehicle objectives

$$V_i = \|x_i(M|t) - x_i^r\|_{P_i}^2 + \sum_{k=t}^{t+M-1} \Big(\|x_i(k|t) - x_i^r\|_{Q_i(k)}^2 + \|u_i(k|t)\|_{R_i(k)}^2 \Big), \tag{14}$$

with constant state references x_i^r and positive semi-definite weighting matrices P_i and $Q_i(k)$, as well as a positive definite input weighting matrix $R_i(k)$. Constraint (11b) is the central model containing all individual vehicle dynamics, (11c) is the initial state, $\mathbb{X}$ and $\mathbb{U}$ are polyhedral sets. Let $\mathcal{T}$ be a given tree in which a node is defined by a tuple (i,j) describing the order of two vehicles (Vehicle i before j) and each path from the root to a leaf node, $\mathcal{T}_m$, describes a feasible crossing sequence through a conflict zone, i.e.,

$$\mathcal{T}_m = \big((i,j),(j,k),(k,l),...\big),\ i,j,k,l \in \mathcal{V}. \tag{15}$$

Thus, optimizing over the integer variable m determines the best tree path with respect to (14). With a slight abuse of notation, $|\mathcal{T}|$ describes the number of paths (feasible crossing sequences) of $\mathcal{T}$ for a given scenario. Distance constraints (11f)–(11g) are valid for certain time intervals which are defined according to the Cases (7a) - (7d) and listed in Table I. Furthermore, Fig. 2 illustrates the feasible $d_i - d_j$ configuration space according to these distance constraints for a vehicle order tuple (i,j).

As (11a) is quadratic, (11b)–(11g) are linear, and (11h) are integers, (11) is a mixed integer quadratic program (MIQP). There are evident reasons why this problem is impractical to be solved online in real applications. First, the problem is computationally hard to solve, as scalability cannot be guaranteed through the central formulation, and the integer decision

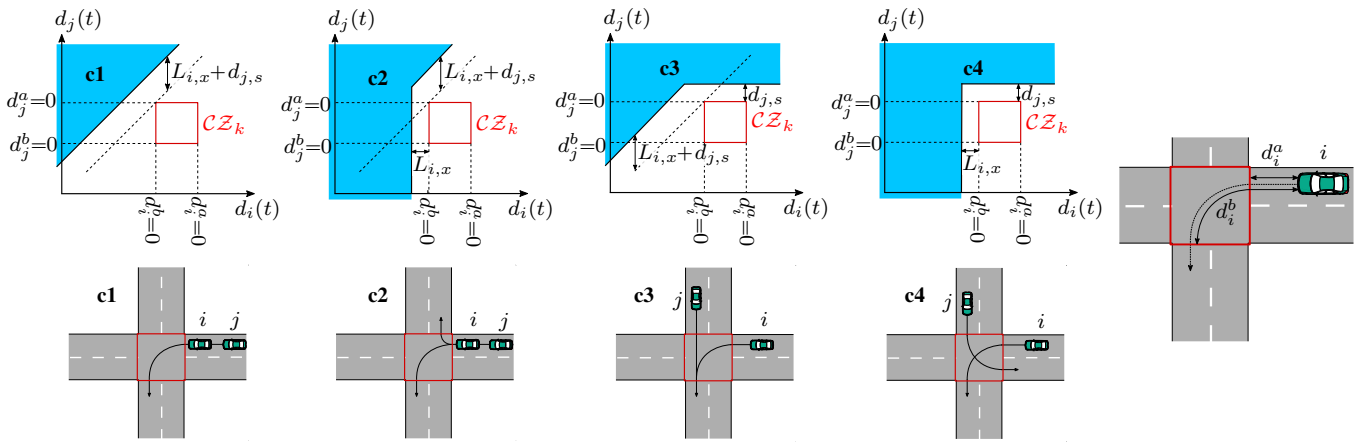

Fig. 2: Top row: feasible $d_i - d_j$ configuration space (blue shaded area) of coordination problem (11) for (i, j). Bottom row: exemplary vehicle constellation at an intersection. Right plot: geometric variables at an intersection.

TABLE I: Time intervals for distance constraints.

$k \in \ldots$	(11g),(18i)	(18j)	(11f)	(18h)
c1:	$\mathbb{I}_{t+1:t+M}$	$\mathbb{I}_{t+1:t+M}$	-	-
c2:	$\mathbb{I}_{t+1:t_i^b}$	$\mathbb{I}_{t+1:t_j^b}$	-	-
c3:	$\mathbb{I}_{t_i^b:t+M}$	$\mathbb{I}_{t_j^b:t+M}$	$\mathbb{I}_{t+1:t_i^b}$	$\mathbb{I}_{t+1:t_j^b}$
c4:	-	-	$\mathbb{I}_{t+1:t_i^b}$	$\mathbb{I}_{t+1:t_j^b}$

introduces a further computational burden. Additionally, the time intervals in Table I depend implicitly on the solution of (11) what requires the computation of a multi-level optimization. Solving such problems exactly is infeasible on real-time platforms [23]. Second, (11) requires central knowledge of all individual vehicle models, what is undesirable considering privacy of vehicle data requirements. Third, trajectories are safety-relevant decisions which are preferred to be computed on-board of a vehicle in order to guarantee a safe behavior even in the case of communication disruptions.

To overcome the drawbacks discussed above, we propose a decomposition of (11) into local optimization problems solved separately by each individual vehicle which is connected, and sharing resulting information with neighboring vehicles. The decomposition is designed to achieve computationally feasible sub-problems where vehicle models are kept private and trajectories can be verified locally to fulfill necessary safety criteria.

C. Distributed Coordination Model

The diagonal structure of (9) immediately suggests a decomposition of (11a) and (11b) into local sub-problems. However, constraints (11f)–(11g) contain inter-vehicle relations. To decompose these, we repeat respective coupling constraints in each coupled local sub-system and move the integer decision (11h) to the central infrastructure node. A scheduling-based method where m is computed in an approximate way given vehicles' trajectories is presented in [24]. In the following we will assume a given integer decision m, i.e., the order in which vehicles cross $\mathcal{CZ}_k$. The crossing sequence can be represented with a directed graph,

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \tag{16}$$

where the set of Vehicles $\mathcal{V}$ are the vertices and directed edges $(i, j) \in \mathcal{E}$, with $i, j \in \mathcal{V}$, mean that Vehicle i crosses $\mathcal{CZ}_k$ before Vehicle j. Given $\mathcal{G}$, we define the set of neighbors of Vehicle i,

$$\mathcal{N}_i = \{\mathcal{P}_i, \mathcal{S}_i\}, \tag{17}$$

where the set of predecessors $\mathcal{P}_i$ contains nodes connected via incoming edges to node $i \in \mathcal{V}$ and similarly successors $\mathcal{S}_i$ nodes from outgoing edges.

Now, (11) can be decomposed into local quadratic programming (QP) problems with coupling constraints between the problems. Thus, a local QP problem of Vehicle i has the form

$$V_i^* = \min_{\bar{x}_i, \bar{u}_i} \; V_i \tag{18a}$$

s.t.

$$x_i(k+1|t) = A_i x_i(k|t) + B_i u_i(k|t) \quad k \in \mathbb{I}_{t:t+M-1} \tag{18b}$$
$$x_i(t|t) = x_i(t) \tag{18c}$$
$$x_i(k|t) \in \mathbb{X}_i \qquad k \in \mathbb{I}_{t+1:t+M} \tag{18d}$$
$$u_i(k|t) \in \mathbb{U}_i \qquad k \in \mathbb{I}_{t:t+M-1} \tag{18e}$$
$$x_i(t+M|t) \in \mathbb{X}_T \tag{18f}$$
$$u_i(t+M-1|t) \in \mathbb{U}_T, \tag{18g}$$
$$d_i^a(k|t) - d_{i,stop} \geqslant 0 \qquad \textbf{c3}, \textbf{c4}, \; j \in \mathcal{P}_i \tag{18h}$$
$$d_i^b(k|t) + L_{i,x} + d_{j,s} - d_j^a(k|t) \leqslant 0 \quad \textbf{c1}, \textbf{c2}, \textbf{c3}, j \in \mathcal{S}_i \tag{18i}$$
$$d_j^b(k|t) + L_{j,x} + d_{i,s} - d_i^a(k|t) \leqslant 0 \quad \textbf{c1}, \textbf{c2}, \textbf{c3}, j \in \mathcal{P}_i, \tag{18j}$$

with local optimization variables

$$\bar{x}_i = (x_i(t+1|t)^\mathsf{T}, \ldots, x_i(t+M|t)^\mathsf{T}), \tag{19}$$
$$\bar{u}_i = (u_i(t|t), \ldots, u_i(t+M-1|t)), \tag{20}$$

and local polyhedral constraint sets (18d) and (18e). The definition of terminal constraints (18f) and (18g) will be discussed in Subsection III-B.

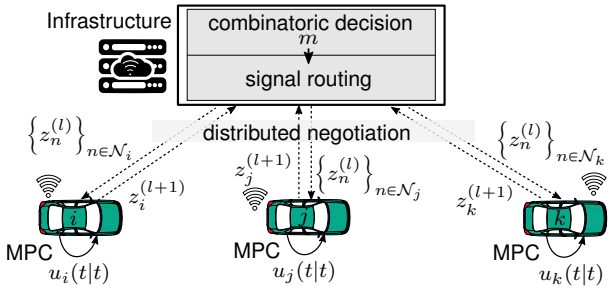

Fig. 3: Distributed Architecture.

To solve decomposition (18) efficiently, two remaining challenges need to be taken into account. The coupling constraints (18i)–(18j) are shared resources, i.e., Vehicle i optimizes the distance state with respect to information from Vehicle j and vice versa. Solving such problems in parallel requires awareness to guarantee feasibility of the distributed system. Additionally, the distance constraints are still implicit functions of the time vehicles exit $\mathcal{CZ}_k$. In the following section we present a solution to guarantee feasibility for the shared constraints while conducting parallel computations. Furthermore, we apply an approximation to remove the implicit time dependency in order to provide computationally viable problems.

III. Iterative Jacobi Negotiation

Every vehicle runs a local MPC unit where it solves its distributed finite horizon QP problem (18) in each iteration step. Fig. 3 illustrates the distributed computation architecture. The MPC law uses the optimized input $u_i(t|t)$ as control input and neglects the remaining optimization results for its local system, but shares the optimization result $(\bar{x}_i, \bar{u}_i)$ with neighboring vehicles. In the next time step, (18) is again solved with updated state measurements in a receding horizon fashion.

In the following, we present the detailed distributed MPC computation in form of a DJOR algorithm derived from [34, Ch. 2.4]. To simplify notation, we state the equivalent problem of (18) as

$$\begin{aligned} z_i^* = \underset{z_i}{\text{argmin}} \quad & V_i(z_i) && \text{(21a)}\\ \text{s.t. } & \mathcal{A}_i z_i - b_i \leqslant 0 && \text{(21b)}\\ & \mathcal{A}_{ij}^d z_i + \mathcal{C}_{ij}^d z_j - b_{ij}^d \leqslant 0, \quad j \in \mathcal{N}_i, && \text{(21c)} \end{aligned}$$

with $z_i = (\bar{x}_i, \bar{u}_i)$, constraints (18b)–(18h) collected in (21b), and coupling constraints (18i)–(18j) in (21c). Observe that coupling between system variables occurs only in coupling constraints and only in bilateral form, i.e., a constraint between system variable of the Vehicle i and a neighboring Vehicle j. Similarly, we achieve

$$\begin{aligned} (z^*, m^*) = \underset{z,m}{\text{argmin}} \quad & \sum_{i=1}^{N_v} V_i(z_i) && \text{(22a)}\\ \text{s.t. } & \mathcal{A} z - b \leqslant 0 && \text{(22b)}\\ & \mathcal{A}_m^d z - b_m^d \leqslant 0 && \text{(22c)}\\ & m \in \mathbb{I}_{1:|\mathcal{T}|}, && \text{(22d)} \end{aligned}$$

for (11) with $z = (\bar{x}, \bar{u})$, (22b) representing (11b)–(11e), and (22c) containing (11f)–(11g).

A. Distributed Over-Relaxation Algorithm

The DJOR algorithm iteratively negotiates between Vehicles $i \in \mathcal{V}$ towards a solution, while an interruption after each iteration results in a feasible solution. Therefore, let $z_i^{(l)}$ be the solution after the l-th inter-sampling iteration between t and $t+1$, $\hat{z}_i$ a feasible initial candidate at the beginning of iterations, and z_i^* the result of a local optimization problem (18). Algorithm 1 summarizes the DJOR procedure, where the

Algorithm 1 Distributed Jacobi Over-Relaxation

clock $\leftarrow t$
Initialization:
$l \leftarrow 0$
$\forall i \in \mathcal{V}$: receive $z_j^{(l)} = \hat{z}_j,\ j \in \mathcal{N}_i$
repeat
  Computation:
  $\forall i \in \mathcal{V}$ in parallel: compute $z_i^*\left(\left\{z_j^{(l)}\right\}_{j\in\mathcal{N}_i}\right)$
  determine next iterate with $\omega_i \geqslant 0,\ \omega_i + \omega_j = 1$:

$$z_i^{(l+1)} = \omega_i z_i^*\left(\left\{z_j^{(l)}\right\}_{j\in\mathcal{N}_i}\right) + (1-\omega_i) z_i^{(l)} \qquad (23)$$

  Synchronization:
  share $z_i^{(l+1)}$ with Vehicles $j \in \mathcal{N}_i$
  $l \leftarrow l+1$
until $l > l_{max} \vee V_i\left(z_i^{(l-1)}\right) - V_i\left(z_i^{(l)}\right) < \gamma\ \forall i \in \mathcal{V}$
apply $u_i(t|t),\ \forall i \in \mathcal{V}$ to local vehicle systems
clock $\leftarrow t+1$

update variable ω_i defines the degree of over-relaxation and γ a termination condition. Solving the optimization problems and the iterative updates can be conducted fully distributed with trajectory exchange between neighboring vehicles after each iteration. To illustrate this, compare (23) with

$$z^{(l+1)}\left(\omega_i, z_i^*, z^{(l)}\right),\ i \in \mathcal{V},\ \omega_i \geqslant 0,\ \sum_{i=1}^{N_v} \omega_i = 1, \qquad (24)$$

which states the structure of the standard JOR update step. It requires central knowledge of z and depends on the total number of vehicles N_v.

In the next subsection we present the construction of initial candidates $\hat{z}_j$, which is an essential point to guarantee feasibility. Thereafter, we discuss the properties Feasibility, Scalability, and Convergence of Algorithm 1.

B. Feasible Initial Guess

At the beginning of a time step each vehicle suggests a feasible initial candidate $\hat{z}_i$ for the inter-sampling iterations, which will be shared with its neighboring Vehicles $j \in \mathcal{N}_i$. In general it is difficult to determine candidates which preserve feasibility among the complete distributed system because in our setup local vehicles do not have information about their neighbors' dynamical capabilities. To resolve this, authors of [31], [32] propose using the system's steady-state as terminal constraint since they handle stabilizing problems such as spring-damper systems. In order to be applicable in a multi-vehicle scenario this idea requires adjustments. Therefore, we use a stand-still condition as terminal constraints (18f) and (18g), such that

$$\mathbb{X}_T = \{x_i | v_i = 0\} \subset \mathbb{X}_i \text{ and } \mathbb{U}_T = 0 \subset \mathbb{U}_i. \tag{25}$$

Now, we suggest the following initial candidate for the upcoming time step $t+1$

$$\hat{z}_i(:|t+1) = \left(z_i^{(l)}(t+1:t+M|t), (x_i'^{\mathsf{T}}, u_i') \right), \tag{26}$$

with the last negotiation result $z_i^{(l)}$ from time step t and the stand-still extension $(x_i'^{\mathsf{T}}, u_i') = (d_i(t+M|t), 0, 0)$.

However, it is not the actual target to bring all vehicles to stand-still. To this end, the terminal condition will be solely used to guarantee system-wide feasibility, while stand-still is not desired to be applied during a nominal coordination scenario. In the MPC framework only the first element of the computed trajectory is applied and all remaining elements are discarded. Thus, we want to ensure the computation of a nominal driving behavior in the beginning of the trajectory, while the stand-still (braking) action is moved to the back of the trajectory as far as possible. To do this, the time-dependent weights

$$Q_i(k)\,[R_i(k)] = \begin{cases} Q_i^n\,[R_i^n] & \text{for } 1 \leqslant k < k_{brake} \\ 0 & \text{for } k_{brake} \leqslant k \leqslant M, \end{cases} \tag{27}$$

are applied in (14), with k_{brake} at the latest possible time-instant for which a vehicle can reach $\mathbb{X}_T$ and $\mathbb{U}_T$ for a desired trajectory $\tilde{z}_i$, which is the solution of (18) without (18f) and (18g). In (27), Q_i^n and R_i^n are constant weighting matrices for the nominal behavior. A set-projection algorithm can be used to calculate k_{brake} and the interested reader is referred to [33] for a detailed presentation of this method.

C. Algorithmic Properties

First, we provide algorithmic properties with respect to Case **c1** where vehicles arrive at a conflict zone $\mathcal{CZ}_k$ from the same in-lane and also leave it on the same out-lane. We will, thereafter, generalize the results to Cases **c2**,**c3**, and **c4** in Subsection III-D.

1) Feasibility: This paragraph shows that each iteration in Algorithm 1 results in a recursive feasible solution for local vehicle problems (18).

Lemma 1. *Given globally feasible initial solutions $z_i^{(0)}$, with $i \in \mathcal{V}$, the iterations* (23) *are feasible for any Vehicle i and respective neighbor Vehicles $j \in \mathcal{N}_i$.*

Proof. Based on the assumption of feasible initial solutions $z_i^{(0)}$, with $i \in \mathcal{V}$, suppose all subsystems conducted a local optimization and compute the combination in (23). Select any subsystem i and any coupled constraint in (21c) for $j \in \mathcal{N}_i$, then it holds that

$$\begin{aligned} \begin{pmatrix} z_i^{(1)} \\ z_j^{(1)} \end{pmatrix} &= \begin{pmatrix} \omega_i z_i^* + (1-\omega_i) z_i^{(0)} \\ \omega_j z_j^* + (1-\omega_j) z_j^{(0)} \end{pmatrix} \\ &= \begin{pmatrix} \omega_i z_i^* + \omega_j z_i^{(0)} \\ \omega_j z_j^* + \omega_i z_j^{(0)} \end{pmatrix} \\ &= \omega_i \begin{pmatrix} z_i^* \\ z_j^{(0)} \end{pmatrix} + \omega_j \begin{pmatrix} z_i^{(0)} \\ z_j^* \end{pmatrix}. \end{aligned} \tag{28}$$

Both solution vectors in the last line are feasible with respect to the coupled constraint. This is because in the first vector z_i^* is feasible as (21) is a QP problem and thus there exists a unique solution. $z_j^{(0)}$ are feasible for the problem (by assumption) and thus $z_i^*\left(z_j^{(0)}\right)$ is a feasible (and optimal) solution. A similar argumentation holds for the second vector. Due to linearity of the constraints with respect to z_j and z_i, the convex combination with $\omega_i + \omega_j = 1$ is also a feasible solution for (21c). The feasibility guarantee for $l > 1$ follows by induction. This completes the proof. □

Remark 1. *For readability, we simplify the notation of $z_i^*\left(z_j^{(0)}\right)$ by omitting its neighbor dependency in* (28) *and similar for z_j^*.*

Algorithm 1 iterates in between two sampling time steps. At the beginning of each iteration it requires initial candidates which are feasible solutions for the local optimization problems (21). Therefore, the following lemma proves feasibility for a time step transition $t+1$ by using trajectory candidates (26).

Lemma 2. *The trajectory candidate $\hat{z}_i(:|t+1)$ from* (26) *is feasible for* (21) *at iteration $l = 0$ and time step $t+1$.*

Proof. The last iteration of time step t, $z_i^{(l)}(k|t)$ with $k \in \mathbb{I}_{t:t+M}$ was feasible according to Lemma 1. Thus, $z_i^{(l)}(k|t)$ with $k \in \mathbb{I}_{t+1:t+M}$ will be feasible at time step $t+1$ as model uncertainty is neglected. It holds that $z_i(t+M|t) \in \mathbb{X}_T \times \mathbb{U}_T$. Furthermore, the final element of $\hat{z}_i(:|t+1)$, $(x_i'^{\mathsf{T}}, u_i') \in \mathbb{X}_T \times \mathbb{U}_T$, extends the stand-still condition. This concludes that (26) is feasible for (21). □

For Case **c1** we make the following assumption on the inter-vehicle distance:

Assumption 1. *If Vehicle j is driving on the same lane behind Vehicle i, they are traveling with an inter-vehicle distance $d_{inter}(k|t) > d_{j,stop},\ k \in \mathbb{I}_{t:t+M}$.*

Definition 1. *An MPC problem computed with* (18) *is recursive feasible if*

$$\begin{aligned} &x_i(k|t) \in \mathbb{X}_i,\ k \in \mathbb{I}_{t:t+M} \wedge x_i(t+M|t) \in \mathbb{X}_T \wedge \\ &u_i(k|t) \in \mathbb{U}_i,\ k \in \mathbb{I}_{t:t+M-1} \\ &\Rightarrow \exists u_i(k|t+1) \in \mathbb{U}_i,\ k \in \mathbb{I}_{t+1:t+M} \text{ such that} \\ &x_i(k|t+1) \in \mathbb{X}_i,\ k \in \mathbb{I}_{t+1:t+M+1} \wedge x_i(t+1+M|t+1) \in \mathbb{X}_T. \end{aligned}$$

Theorem 1. *Let Case **c1** and Assumption 1 hold. Given Lemma 1 and Lemma 2, each iteration of Algorithm 1 leads to a recursive feasible solution for all Vehicles i in $\mathcal{G}$ with local problems* (21)*.*

Proof. Replacing $d_{j,s}$ in (18i) with the definition from (7a) gives

$$d_j^a(t) - (d_i^a(t) + L_{i,x}) \geqslant d_{j,stop}. \tag{29}$$

Given Assumption 1 and following a similar argumentation for (18j) ensures the existence of feasible solutions $z_i^{(0)}$ and $z_j^{(0)}$ for $i, j \in \mathcal{V}$ of problem (21). Following inter-sampling iterations remain feasible for local problems i according to Lemma 1. Moreover, following the reasoning in Lemma 2, feasible trajectories exist for time step transitions $t \to t+1$. Consequently, Definition 1 is fulfilled for all Vehicles $i \in \mathcal{V}$ with problems (21) interacting according to Algorithm 1. □

2) Scalability: The standard JOR algorithm requires a centralized update step (24). Moreover, through the condition $\sum_{i=1}^{N_v} \omega_i = 1$ the speed of convergence depends on the number of vehicles in the network $\mathcal{G}$.

Note that, in general, the neighbor sets in (17) contain several vehicles, i.e., $|\mathcal{P}_i| \geqslant 1$ and $|\mathcal{S}_i| \geqslant 1$. This results in a negotiation between several interdependent vehicles according to (18h) – (18j). However, applying (18) with Algorithm 1 enables a scalable and fully distributed computation, as guaranteed by the following theorem.

Theorem 2. *$\omega_i = 0.5,\ \forall i \in \mathcal{V}$ is a valid choice for the step size in* (23) *independent of $\mathcal{G}$.*

Proof. By construction of the coordination problem, it holds for the central inter-vehicle distance constraints $\mathcal{A}_m^d = (\alpha_{ij}) \in \{-1, 0, 1\}^{p \times q}$ and thus

$$\| \mathcal{A}_m^d \|_\infty = \max_{i \in \mathbb{I}_{1:p}} \sum_{j \in \mathbb{I}_{1:q}} | \alpha_{ij} | = 2.$$

This means that a distance constraint at an optimization stage is shared at most by two vehicles. Consequently, each of such constraints, i.e., each row in (22c), is shared between a Vehicle $i \in \mathcal{V}$ and at most one other Vehicle $j \in \mathcal{N}_i$ in the distributed setup. For such a setup, we know from Lemma 1 that the DJOR update (23) between Vehicles i and j is feasible. For any neighbor permutation $i, j \in \mathcal{V}$ and $i \neq j$ (where indices may mutually vary), $\omega_i = \omega_j = 0.5$ thus fulfills the convexity condition $\omega_i + \omega_j = 1$. □

Remark 2. *Knowing that Vehicle i is predecessor of Vehicle j and at the same time Vehicle j is successor of Vehicle i for all pairs $(i, j) \in \mathcal{E}$ and all prediction time steps k, enables the relaxing of condition $\omega_i = \omega_j = 0.5$ to the more general condition $\omega_i + \omega_j = 1,\ \omega_i > 0, \omega_j > 0$.*

3) Convergence: Convergence for the overall system is guaranteed by the fact that local cost functions are decoupled as well as the following proposition.

Proposition 1. *The DJOR iterations* (23) *applied to the distributed systems* (21) *converge for $l \to \infty$.*

Proof. First, we show monotonicity between two iteration steps of individual decoupled vehicle costs:

$$\begin{aligned} V_i\left(z_i^{(l+1)}\right) &= V_i\left(\omega_i z_i^* + (1-\omega_i) z_i^{(l)}\right) \\ &\leqslant \omega_i V_i\left(z_i^*\right) + (1-\omega_i) V_i\left(z_i^{(l)}\right) \\ &\leqslant \omega_i V_i\left(z_i^{(l)}\right) + (1-\omega_i) V_i\left(z_i^{(l)}\right) \\ &= V_i\left(z_i^{(l)}\right). \end{aligned}$$

This holds for all Vehicles $i \in \mathcal{V}$. The first line applies (23), the second line follows from convexitiy of the cost functions, and the third line follows from optimality of local problems (21). As all individual quadratic cost functions V_i are bounded below, V is also bounded, and thus convergence can be guaranteed as $l \to \infty$. □

D. Fulfillment of Coordination Conditions

This subsection discusses the extension of the feasibility results of Algorithm 1 to the coordination Cases **c2**, **c3**, **c4**. In the first step, we present an approximate solution to compute the crossing time steps.

1) Crossing time approximation: To avoid the computation of a multi-level optimization problem, we utilize the receding horizon nature of MPC. This enables an approximation of the exit time defined in (6) by referring to the trajectory computed at the previous time step $t-1$ such that

$$\tilde{t}_i^b := \begin{cases} \infty, & \text{if } d_i^b(t+M|t-1) + L_{i,x} > 0 \\ \underset{d_i^b(k|t-1) \leqslant -L_{i,x}}{\operatorname{argmin}} \ d_i^b(k|t-1), & \text{else,} \end{cases} \tag{30}$$

and we substitute $t_i^b = \tilde{t}_i^b$ with its approximation.

2) Case distinction: For Case **c2**, where Vehicles i and j approach from the same lane (Vehicle i driving in front of Vehicle j) but leave $\mathcal{CZ}_k$ on different lanes, Theorem 1 is still valid, as the initial conditions do not change. The difference is solely that constraints (18i) and (18j) are not formulated for the complete prediction horizon, but only from time step $t+1$ until time step t_i^b.

In Case **c3** vehicles approach from different lanes and merge onto the same lane in $\mathcal{CZ}_k$. Thus, coupling constraints are not required until the time of merging, i.e., only from t_i^b until $t+M$ (assuming Vehicle i crosses before Vehicle j). Constraint (18h) ensures consistency of the vehicle order by guaranteeing that a vehicle remains in front of the conflict zone $\mathcal{CZ}_k$ until its predecessor has predicted to have crossed it, in what we assume to be a feasible solution:

Assumption 2. *$d_j^b(t_0 + M|t_0) \geqslant d_{j,stop}$ is feasible for Vehicle j, with t_0 being the time of starting the negotiation according to Algorithm 1 with its neighbor vehicles.*

This enables the feasibility guarantee for Case **c3**.

Theorem 3. *Let Case **c3** and Assumption 2 hold. A partial coupling along the horizon, $\mathbb{I}_{t_i^b:t+M}$, of constraints* (18i)–(18j) *results in feasible solutions for Vehicles i and j.*

Proof. Let Vehicle i cross $\mathcal{CZ}_k$ before Vehicle j. For the case $t_i^b = \infty$, i.e., the horizon of Vehicle i does not yet predict to cross the conflict zone $\mathcal{CZ}_k$, the set $\mathbb{I}_{t_i^b:t+M} = \varnothing$ is empty and problems (18) of Vehicles i and j are not coupled through (18i) and (18j). Without coupling, it follows feasibility for the local problem (18a)–(18g) of the first Vehicle i. Furthermore, given Assumption 2 we can also guarantee feasibility for Vehicle j with problem (18a)–(18h).

If the prediction of Vehicle i crosses $\mathcal{CZ}_k$ at time step $t-1$, i.e., $t_i^b \leqslant t+M$ according to (30), it follows with Lemma 2 that the same solution will be a feasible candidate at time step t. Before i and j become coupled, (18h) was feasible for Vehicle j.

Now, for Vehicle i's problem we know from (18i):

$$\underbrace{d_i^b(t_i^b) + L_{i,x}}_{\leqslant 0} \leqslant \underbrace{d_j^a(t_i^b) - d_{j,s}}_{\geqslant 0}, \tag{31}$$

where the left side of the inequality follows from the condition that vehicles become coupled and the right side from Vehicle j's constraint (18h). The same argumentation holds from Vehicle j's perspective referring to constraint (18j). The feasibility of activated coupling constraints for prediction steps $\mathbb{I}_{t_i^b:t+M}$ between vehicles i and j follows from the reasoning in Theorem 1.

The discussion for the scenario where Vehicle j crosses $\mathcal{CZ}_k$ before Vehicle i, follows analogously. □

Feasibility for Case **c4** follows directly from Assumption 2.

IV. Environmental Uncertainty Handling

In Section III we proposed a distributed negotiation algorithm for multi-vehicle coordination which guarantees a feasible, and thus safe solution after each iteration step. The algorithmic guarantees are provided in absence of model and environmental uncertainties. Long prediction horizons are desirable for increasing the performance of the coordination procedure as, e.g., the approximation (30) will become more accurate with longer horizons. Yet, predicting the environment of an autonomous vehicle with the required confidence will only be possible for shorter horizons. Reacting to uncertain events in the environment, such as a pedestrian appearing in a vehicle's sensor view, might require deviating from the plan with respect to the long horizon which has been agreed on with other vehicles in the network. This can cause infeasiblity in the network due to a possible violation of the coupling constraints (18i)–(18j). However, it does not mean that such a violation of the long horizon plan leads to an unsafe behavior, as vehicles travel with a safety distance $d_{i,s}$ between each other and they can still react locally to unforeseen events.

We propose to induce this local reaction behavior by relaxing the coupling constraints in the form of exact penalty functions. If possible, vehicles will fulfill the formulated inter-vehicle coupling constraints and conduct the long horizon plan negotiated with its neighboring vehicles. Whenever unforeseen events occur and no feasible agreement with the neighboring vehicles can be found, vehicles can violate the coupling constraints to conduct, for example, an emergency braking maneuver. Thus, a prioritization of safety over coordination is achieved. After such a local reaction the vehicles automatically recover from the coupling constraint violation and return to a network-wide feasible solution.

In the following, we introduce the concept of exact penalty functions and thereafter describe how they are integrated into the DJOR algorithm.

A. Exact Penalty Functions

First, we recast the local QP problem (21) in an optimization problem with soft constrained inter-vehicle distances:

$$\overline{z}_i^* = \underset{z_i}{\operatorname{argmin}} \;\; V_i(z_i) + \delta_i \Big\| \sum_{j \in \mathcal{N}_i} \left(\mathcal{A}_{ij}^d z_i + \mathcal{C}_{ij}^d z_j - b_{ij}^d \right)^+ \Big\|_1 \tag{32a}$$

$$\text{s.t.} \qquad \mathcal{A}_i z_i - b_i \leqslant 0, \tag{32b}$$

with penalty weight $\delta_i \in \mathbb{R}$ and

$$\left[\left(\mathcal{A}_{ij}^d z_i + \mathcal{C}_{ij}^d z_j - b_{ij}^d \right)^+ \right]_n := \max\left(\left[\mathcal{A}_{ij}^d z_i + \mathcal{C}_{ij}^d z_j - b_{ij}^d \right]_n, 0 \right), \tag{33}$$

where index n indicates the n-th element of vector $\mathcal{A}_{ij}^d z_i + \mathcal{C}_{ij}^d z_j - b_{ij}^d$. Let λ_i^* be the Lagrangian vector corresponding to the feasible and optimal solution z_i^* of (21).

Theorem 4. *If $\delta_i > \|\lambda_i^*\|_\infty$, then the minimizers z_i^* and $\overline{z}_i^*$ are identical.*

Proof. See [35, Theorem 14.3.1] □

How to compute apropriate values of δ_i in an MPC setting is discussed in [36], [37].

The non-smoothness of the 1-norm in Problem (32) ensures its exactness with respect to the original Problem (21), yet a non-smooth optimization problem cannot be computed using standard algorithms. Therefore, an equivalent problem is formulated using the slack variables vector ϵ_i to become a QP problem again which can be efficiently solved [38], [39]:

$$\overline{z}_i^* = \underset{z_i, \epsilon_i}{\operatorname{argmin}} \;\; V_i(z_i) + \delta_i \|\epsilon_i\|_1 \tag{34a}$$

$$\text{s.t.} \qquad \mathcal{A}_i z_i - b_i \leqslant 0 \tag{34b}$$

$$\mathcal{A}_{ij}^d z_i + \mathcal{C}_{ij}^d z_j - b_{ij}^d \leqslant [\epsilon_i]_l, \;\; j \in \mathcal{N}_i, \;\; l \in \mathbb{I}_{1:|\mathcal{N}_i|} \tag{34c}$$

$$-\epsilon_i \leqslant 0. \tag{34d}$$

B. Integration in Jacobi Negotiation

If a solution to (21) exists, then this solution will be (exactly) found by (34). If no solution exists, (34) will compute the closest possible solution, i.e. the solution with the least possible constraint violation. This means it will attempt to keep the coupling distances to its neighbor vehicles, represented by (34c), whenever possible and violates it only if necessary due to, e.g., environmental uncertainties.

Given the responsibility of a vehicle to avoid a collision with its preceding vehicle and the fact that the predecessor's dynamics are not known locally by the following vehicle, a collision can be avoided if the inter-vehicle distance is larger than the required stopping distance. In an offline manner, we can compute the stopping distance of Vehicle j (which is

assumed to be scheduled behind Vehicle i), such that it holds

$$d_{j,stop} = \min_{z_j, t_{stop}} \quad d_i(t_{stop}) \tag{35a}$$
$$\text{s.t. (18b), (18d), (18e)} \tag{35b}$$
$$t_{stop} \in \mathbb{I}_{0:M} \tag{35c}$$
$$v_j(\bar{t}) = 0 \qquad \bar{t} \in \mathbb{I}_{t_{stop}:M} \tag{35d}$$
$$v_j(0) = v_{max} \tag{35e}$$
$$d_j(0) = 0, \tag{35f}$$

where it is assumed that the horizon is large enough, such that the optimization problem has a solution, v_{max} is the maximum allowed velocity in the scenario, and z_j is determined by choosing $u_j(t) = \min_{u \in \mathbb{U}_j} u$ until the vehicle stops.

Remark 3. *Computing the stopping distance offline with* (35) *leads to reasonable values for low-speed maneuvers (e.g., up to* $30km/h$*). To reduce the conservatism for higher speed driving* (35) *can be computed online by replacing* (35e) *with the current velocity* $v_j(t)$ *and* z_j *with the vehicle's current trajectory. Additionally, the Responsibility-Sensitive Safety (RSS) framework suggests rules for inter-vehicle distances [40], with reduced conservatism as also assumptions on the braking dynamics of the predecessor vehicle are considered.*

A potential constraint violation needs to be considered in the DJOR negotiation to guarantee collision avoidance between vehicles. A slack variable unequal to zero, $\epsilon_i > 0$, after the local optimization (line 7 in Algorithm 1) indicates that the coupling constraints need to be violated. If this is the case, the negotiation step (23) has to be modified by choosing $\omega_i = 1$ to ensure the validity of the stopping distance $d_{i,s}$.

After an unexpected event, (34) will automatically recover by computing a solution which fulfills the coupling constraints again as soon as possible. Once $\epsilon_i = 0$ holds, the recovery is completed and the negotiation can be switched back to $\omega_i = 0.5$.

Remark 4. *Constraint softening, such as exact penalty functions, is commonly applied to all constraints and used to avoid infeasibilities of the optimization problem triggered thorough model uncertainties or sensor noise. In this article, however, we soften solely the coupling constraints to emphasize the methodology of avoiding coordination infeasibilities.*

Algorithm 2 summarizes the essential steps of the proposed multi-vehicle coordination procedure.

Algorithm 2 Overall Coordination Procedure

1: Given $d_{i,stop}$ from (35), $\hat{z}_i(:|t+1)$ form (26) $\forall i \in \mathcal{V}$
2: ***Central Infrastructure:***
3: Determine crossing sequence (16) and $\tilde{t}_i^b$ with (30)
4: Share $\mathcal{N}_i$ and respective Cases (7) with vehicles $i \in \mathcal{V}$
5: ***Local Vehicles:***
6: Conduct Steps 2 – 13 of Algorithm 1
7: **if** $\epsilon_i = 0$ (at Step 8) **then**
8: $\omega_i = 0.5$
9: **else**
10: $\omega_i = 1$

V. Numerical Examples

Numerical simulations were conducted to illustrate the performance and functionality of the methodology introduced in this article. The simulation PC contains an Intel Core i5 double core processor with $2.5GHz$ and $8GB$ RAM memory. Simulations were conducted with MATLAB and its quadprog solver to compute the QP problems.

Table II summarizes the applied simulation parameter for each of the following subsections, where $x_{i,v}^r$ is the state reference value referring to the velocity state, and similar for the constraint set $\mathbb{X}_{i,v}$. The column labeled with **ID** indicates the vehicle IDs. In all setups the sampling time was $T_s = 0.1s$.

A. Prediction Horizon Evaluation

In the first step, we evaluate the influence of the prediction horizon length M on the coordination procedure. Therefore, two vehicles are simulated to cross a common $\mathcal{CZ}_k$ in a scenario setup as shown in Fig. 1 with local MPC laws (18) and the negotiation according to Algorithm 1. Fig. 4 illustrates the resulting vehicle distance-state in the $d_1 - d_2$ configuration space for Cases **c3** and **c4** with horizon lengths $M = 30, 50, 100$ and Vehicle 1 crossing $\mathcal{CZ}_k$ before Vehicle 2. For each horizon length the distributed optimization results in a feasible and safe coordination. Yet, increasing M leads to a smoother coordination result, since the exit time t_i^b in (30) can be evaluated earlier than that with shorter horizons. This becomes visible through the dots in Fig. 4 which show the terminal states for respective planning steps.

B. Coordination Performance in an Intersection Crossing Scenario

The considered intersection scenario is illustrated in Fig. 5. We assume a set of autonomous vehicles crossing the intersection along pre-defined paths, where all possible paths are drawn in the figure. Vehicles communicate with an infrastructure unit which computes the crossing sequence for the intersection area once vehicles enter a certain scheduling zone (compare Fig. 3). The intersection area is divided into five conflict zones. To take this into consideration, we extend the introduced coordination concept for a single conflict zone by repeating constraints (18h)–(18j) for each conflict zone $\mathcal{CZ}_k,\ k \in \mathbb{I}_{1:5}$ in the intersection area which a vehicle passes.

To evaluate the performance, we simulate 200 randomly generated scenarios. In each scenario 6 vehicles are placed on incoming lanes E and W towards the intersection (Vehicles $1, 2, 3$ on lane E, Vehicles $4, 5, 6$ on lane W) with random initial distances, between $15m$ and $65m$, to the intersection zones and a minimum initial inter-vehicle distance of $5m$, as well as stand-still initial conditions. Furthermore, the maneuvers, i.e., *right turn, straight, left turn*, are determined randomly for each vehicle and scenario. For each of the 200 test cases five different control methods are simulated: (i) overpass, where no interaction between vehicles is required; (ii) centralized: the result (11) with a given decision m; (iii)

TABLE II: Simulation Parameters

Parameter	ID	M	$x^r_{i,v}$	Q^n_i	R^n_i	$\mathbb{U}_i$	$\mathbb{X}_{i,v}$	$\mathbb{U}^T_i$	$\mathbb{X}^T_{i,v}$	$d_{i,s}$	ω_i	l_{max}	δ_i
Subsection V-A													
Value	1	-	$7m/s$	diag(0, 5)	1	$[-7m/s^2, 4m/s^2]$	$[0m/s, 9m/s]$	(25)	(25)	$2m$	0.5	4	
	2	-	$8.5m/s$	”	”	”	”	”	”	”	”	”	
Subsection V-B													
	1, 4	50	$5m/s$	diag(0, 5)	12	$[-7m/s^2, 4m/s^2]$	$[0m/s, 9m/s]$	(25)	(25)	$2m$	0.5	-	
Value	2, 5	”	$6m/s$	”	”	”	”	”	”	”	”	-	
	3, 6	”	$7m/s$	”	”	”	”	”	”	”	”	-	
Subsection V-C													
	1	50	$7m/s$	diag(0, 5)	10	$[-7m/s^2, 4m/s^2]$	$[0m/s, 10m/s]$	(25)	(25)	$2m$	0.5	4	$4e3$
Value	2	”	$8m/s$	diag(0, 10)	”	”	”	”	”	”	”	”	”
	3	”	$9m/s$	diag(0, 50)	”	$[-5m/s^2, 4m/s^2]$	”	”	”	”	”	”	”

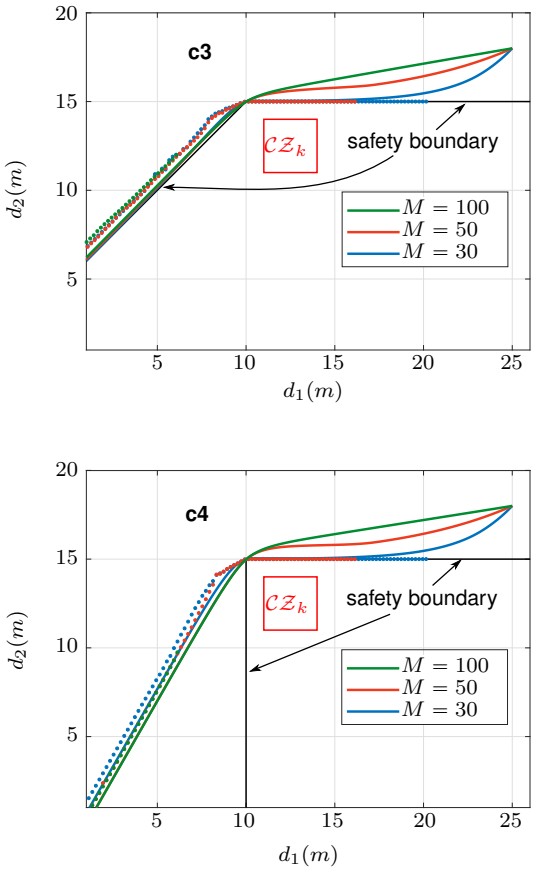

Fig. 4: Trajectories for varying horizon lengths M in the $d_1 - d_2$ configuration space for vehicles 1 and 2 crossing $\mathcal{CZ}_k$ in Cases **c3** (top) and **c4** (bottom), with safety boundaries according to Fig. 2. Vehicle 1 crosses before vehicle 2. Solid lines are the actual distance states, and dots show the terminal states for each planning step.

Jacobi $4-$iter, applying problems (18) with Algorithm 1 and $l_{max} = 4$; (iv) Jacobi $1-$iter, the same as (iii) with $l_{max} = 1$; and (v) traffic rules, where vehicles have to yield the right-of-way which in these scenarios is the case for left turns (if two vehicles want to turn left at the same time lane E is prioritized). The crossing sequence in (ii), (iii), and (iv) is computed according the scheduling rule introduced in [24].

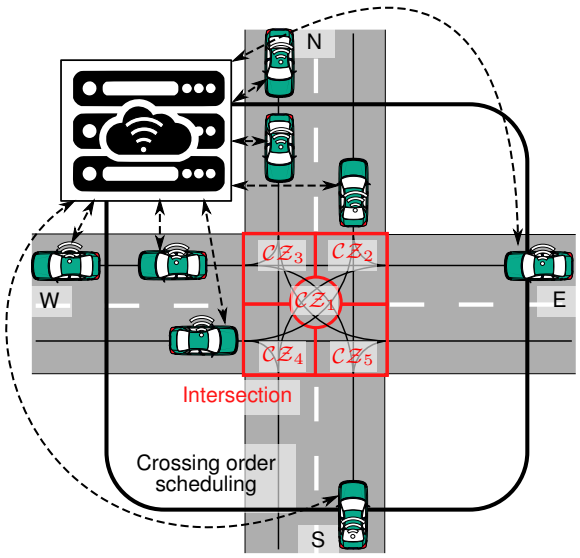

Fig. 5: Intersection Scenario.

Fig. 6 and Fig. 7 show a single example test case by comparing control method (iii) and (v). Thereby, Fig. 6 illustrates the distance trajectories, while Fig. 7 plots the intersection scenarios for both approaches with the vehicles' predictions at time $t = 7.0s$ and $t = 9.4s$, respectively. The example shows that the inter-vehicle coordination using the Jacobi approach enables crossing the intersection more efficient (faster), compared to the rule-based approach.

The overall evaluation of the 200 test cases is summarized in Fig. 8. The left plot shows the average percentage increase of the control methods (ii)–(v) (bars from left to right) compared with the overpass method (i), which is the lower bound for the performance measures. As performance measures the average crossing time, i.e., the time it takes until all vehicles have crossed the intersection area, and the average cumulative acceleration effort, i.e the sum of all absolute acceleration values from all vehicles, are evaluated. The right plot illustrates the average computation time of the centralized computation (ii) and methods (iii) and (iv), where each sub-stack (in 4-iter and 1-iter) corresponds to the mean computation time of a single vehicle.

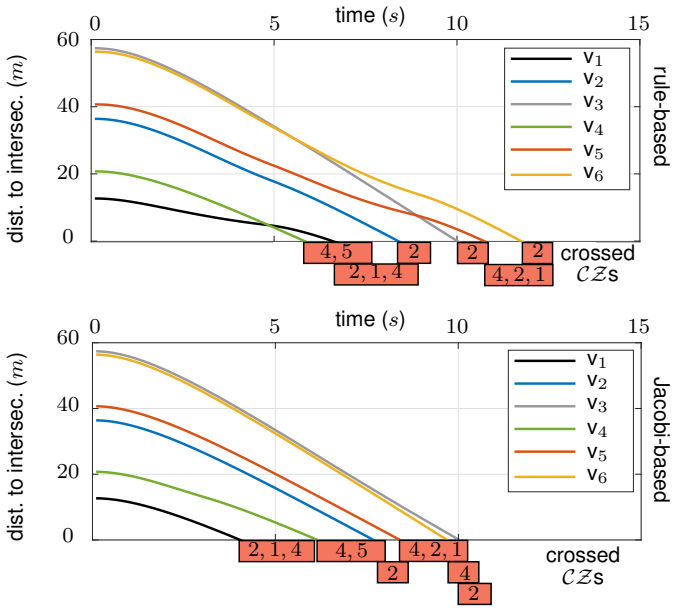

Fig. 6: Distance trajectories for random test case with vehicles $1-6$ (v_1-v_6). Top: traffic rule simulation. Bottom: Jacobi algorithm with $l_{max} = 4$ inter-sampling iterations.

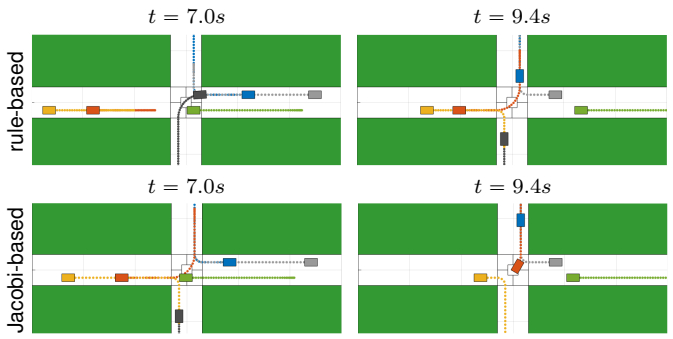

Fig. 7: Scenario visualization of test case plotted in Fig. 6 for time steps $t = 7.0s$ and $t = 9.4s$. Dots show predictions of vehicles at the respective time step. Top: traffic rule simulation. Bottom: Jacobi algorithm with $l_{max} = 4$ inter-sampling iterations.

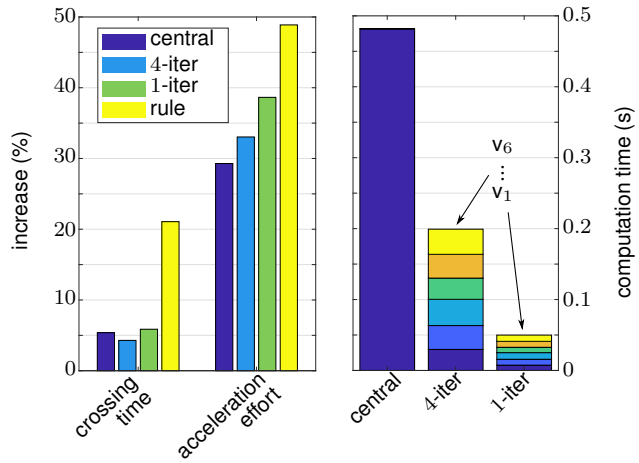

Fig. 8: Evaluation intersection performance simulation.

The crossing time evaluation shows similar results for methods (ii)–(iv). The crossing time itself is not the foremost optimzation objective, which becomes visible as the average central methods (ii) finds longer results than (iii), the 4-iter method. However, methods (ii)–(iv) result in significantly shorter crossing times than the traffic rule method (v). The acceleration effort is the lowest for the central computation result (ii) which is about 20% less than in the traffic rule simulation (v). Also, a performance improvement effect of increased inter-sampling iterations of the Jacobi approach becomes visible by comparing method (iii) and (iv). Investigating the computation time reveals a significant benefit from using distributed optimization. While a centralized computation consumes on average $480ms$, 4 inter-sampling iterations require between $30ms$ and $40ms$, and a single iteration between $8ms$ and $9ms$ for each vehicle. Remember that the solution for each vehicle in methods (iii) and (iv) can be computed in parallel. The low computation times of the distributed implementations leave significant room for the inter-vehicle communication within a sampling time interval of $T_s = 100ms$.

C. Uncertainty Simulation

This subsection illustrates simulation results with exact penalty functions on the inter-vehicle coupling constraint. We model three vehicles moving in a platoon with the following order: Vehicle 1 $\rightarrow$ Vehicle 2 $\rightarrow$ Vehicle 3. At time $t = 5s$ Vehicle 2 conducts an emergency braking maneuver with $a_2 = a_{2,min} = -7m/s^2$, which is shown in the top middle plot of Fig. 9. Before this maneuver, Vehicles 2 and 3 drive with minimum inter-vehicle distance $d_{3,s} = 2m$ (lower plot of Fig. 9). Lower braking capability of vehicle 3, i.e., $a_{3,min} = -5m/s^2$, would lead to infeasible solutions of the distributed optimization without softened constraints due to an inter-vehicle constraint violation. Fig. 10 shows how the respective slack variable ϵ_3 becomes active during the braking phase and the automatic recovery after $t = 7.9s$. Once the distributed system has recovered into a feasible area, the negotiation continues as it would in the nominal case (with hard coupling constraints) due to the exactness of the penalty function.

VI. Conclusion

In this article, we have introduced an iterative coordination methodology for connected and automated vehicles using distributed model predictive control computations with distributed Jacobi over-relaxation (DJOR) updates, which ensure feasibility after each inter-sampling iteration step. We have shown that through the specific problem setup the algorithm scales well independently of the interaction topology. The coordination is conducted with respect to conflict zones. The inter-vehicle coupling is softened using exact penalty functions to prioritize safety in unforeseen scenarios over the coordination performance, while feasibility is preserved at all times. The methodology is illustrated and evaluated by numerical simulations including an intersection crossing scenario.

Guaranteed properties, including feasibility and scalability, together with the distributed computation qualify the presented

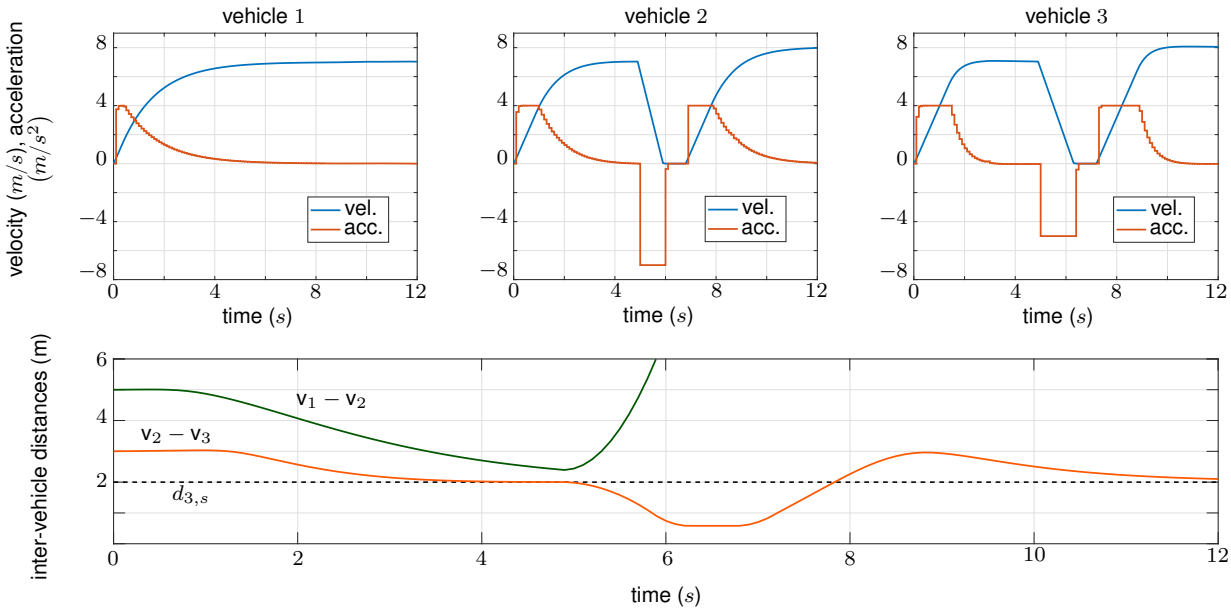

Fig. 9: Emergency braking maneuver in distributed vehicle setup. Top row: vehicles' acceleration and velocity profiles. Bottom: inter-vehicle distances with minimum distance $d_{3,s}$ between Vehicles 2 and 3.

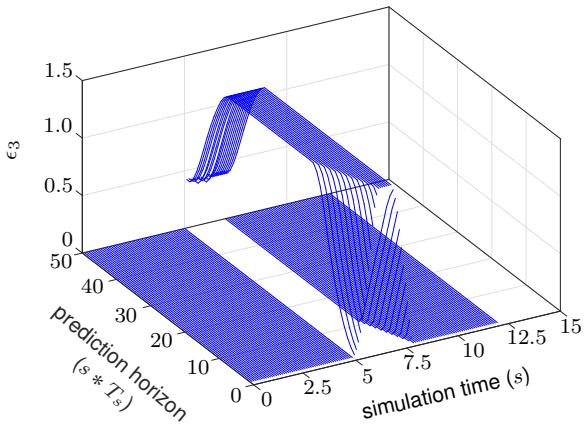

Fig. 10: Slack variable ϵ_3 of constraint (18j) during emergency braking maneuver.

methods for real-world automated vehicle coordination scenarios. A proof-of-concept of such scenarios is the subject of future work. Moreover, the proposed distributed problem setting with its special structure is expected to match for many other multi-agent coordination tasks. It would be of interest to apply the DJOR algorithm and exploit its benefits in other domains beyond multi-vehicle tasks.

REFERENCES

[1] P. Kavathekar and Y. Chen, "Vehicle platooning: A brief survey and categorization," in *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. 829–845, American Society of Mechanical Engineers Digital Collection, 2011.

[2] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2016.

[3] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2015.

[4] X. Shen, X. Zhang, and F. Borrelli, "Autonomous parking of vehicle fleet in tight environments," in *2020 American Control Conference (ACC)*, pp. 3035–3040, IEEE, 2020.

[5] Y. Li, K. H. Johansson, and J. Mårtensson, "A hierarchical control system for smart parking lots with automated vehicles: Improve efficiency by leveraging prediction of human drivers," in *2019 18th European Control Conference (ECC)*, pp. 2675–2681, IEEE, 2019.

[6] M. Kneissl, A. K. Madhusudhanan, A. Molin, H. Esen, and S. Hirche, "A multi-vehicle control framework with application to automated valet parking," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[7] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[8] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2016.

[9] K.-D. Kim and P. R. Kumar, "An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, 2014.

[10] W. B. Dunbar and R. M. Murray, "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.

[11] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940–5946, 2017.

[12] A. Katriniok, P. Sopasakis, M. Schuurmans, and P. Patrinos, "Nonlinear model predictive control for distributed motion planning in road intersections using panoc," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 5272–5278, IEEE, 2019.

[13] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras, "Optimal control and coordination of connected and automated vehicles at urban traffic intersections," in *2016 American Control Conference (ACC)*, pp. 6227–6232, IEEE, 2016.

[14] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Primal decomposition of the optimal coordination of vehicles at traffic intersections," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 2567–2573, IEEE, 2016.

[15] G. Notarstefano, I. Notarnicola, and A. Camisa, "Distributed optimization for smart cyber-physical networks," *arXiv preprint arXiv:1906.10760*, 2019.

[16] G. R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative receding horizon conflict resolution at traffic intersections," in *53rd IEEE Conference on Decision and Control*, pp. 2932–2937, IEEE, 2014.

[17] A. Richards and J. How, "A decentralized algorithm for robust constrained model predictive control," in *Proceedings of the 2004 American control conference*, vol. 5, pp. 4261–4266, IEEE, 2004.

[18] M. Farina and R. Scattolini, "Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems," *Automatica*, vol. 48, no. 6, pp. 1088–1096, 2012.

[19] M. Zanon, S. Gros, H. Wymeersch, and P. Falcone, "An asynchronous algorithm for optimal vehicle coordination at traffic intersections," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12008–12014, 2017.

[20] M. Kneissl, A. Molin, H. Esen, and S. Hirche, "A feasible mpc-based negotiation algorithm for automated intersection crossing," in *2018 European Control Conference (ECC)*, pp. 1282–1288, IEEE, 2018.

[21] F. Molinari and J. Raisch, "Automation of road intersections using consensus-based auction algorithms," in *2018 Annual American Control Conference (ACC)*, pp. 5994–6001, IEEE, 2018.

[22] J. Gregoire, S. Bonnabel, and A. De La Fortelle, "Priority-based coordination of robots," 2014.

[23] R. Hult, M. Zanon, S. Gras, and P. Falcone, "An miqp-based heuristic for optimal coordination of vehicles at intersections," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 2783–2790, IEEE, 2018.

[24] M. Kneissl, A. Molin, H. Esen, and S. Hirche, "Combined scheduling and control design for the coordination of automated vehicles at intersections," *IFAC-PapersOnLine*, vol. 51, no. 1, 2020.

[25] X. Qian, J. Gregoire, A. De La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *2015 European Control Conference (ECC)*, pp. 3452–3458, IEEE, 2015.

[26] F. Belkhouche, "Collaboration and optimal conflict resolution at an unsignalized intersection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2301–2312, 2018.

[27] G. Schildbach, M. Soppert, and F. Borrelli, "A collision avoidance system at intersections using robust model predictive control," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 233–238, IEEE, 2016.

[28] A. Colombo and D. Del Vecchio, "Least restrictive supervisors for intersection collision avoidance: A scheduling approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 6, pp. 1515–1527, 2014.

[29] H. Ahn and D. Del Vecchio, "Semi-autonomous intersection collision avoidance through job-shop scheduling," in *Proceedings of the 19th International Conference on Hybrid Systems: Computation and Control*, pp. 185–194, 2016.

[30] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2510–2525, 2018.

[31] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia, "Cooperative distributed model predictive control," *Systems & Control Letters*, vol. 59, no. 8, pp. 460–469, 2010.

[32] M. D. Doan, M. Diehl, T. Keviczky, and B. De Schutter, "A jacobi decomposition algorithm for distributed convex optimization in distributed model predictive control," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4905–4911, 2017.

[33] M. Kneissl, A. Molin, H. Esen, and S. Hirche, "A one-step feasible negotiation algorithm for distributed trajectory generation of autonomous vehicles," in *Proceedings of the Conference on Decision and Control (CDC)*, 2019.

[34] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*, vol. 23. Prentice hall Englewood Cliffs, NJ, 1989.

[35] R. Fletcher, *Practical methods of optimization*. John Wiley & Sons, 1987.

[36] E. C. Kerrigan and J. M. Maciejowski, "Soft constraints and exact penalty functions in model predictive control," in *UKACC Int. Conf. (Control 2000)*, (Cambridge), 2000.

[37] M. Hovd, "Multi-level programming for designing penalty functions for mpc controllers," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 6098–6103, 2011.

[38] N. M. de Oliveira and L. T. Biegler, "Constraint handing and stability properties of model-predictive control," *AIChE journal*, vol. 40, no. 7, pp. 1138–1155, 1994.

[39] P. O. Scokaert and J. B. Rawlings, "Feasibility issues in linear model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1649–1659, 1999.

[40] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "On a formal model of safe and scalable self-driving cars," *arXiv preprint arXiv:1708.06374*, 2017.

Maximilian Kneissl is currently working as a Research Engineer at DENSO AUTOMOTIVE Deutschland GmbH in Eching, Germany. He received his B.Sc., M.Sc., and Ph.D. degree in Electrical and Computer Engineering from the Technical University of Munich, Germany, in 2013, 2016, and 2021, respectively. The Ph.D. thesis was conducted at the chair of Information-Oriented Control, Department of Electrical and Computer Engineering, Technical University of Munich, Germany, in collaboration with DENSO AUTOMOTIVE Deutschland GmbH. His research interests include distributed control and planning for cooperative vehicles in the field of autonomous driving, model-based system engineering, and simulation methods for distributed control systems.

Adam Molin is currently a Technical Manager at DENSO AUTOMOTIVE Deutschland GmbH in Eching, Germany. Prior to that, he was a postdoctoral researcher at the Department of Automatic Control, Royal Institute of Technology (KTH), Stockholm, Sweden, from 2014 to 2016. He received his Diplom degree in electrical engineering in 2007 and his Doctor of Engineering degree in 2014, both from the Department of Electrical Engineering and Information Technology, Technical University of Munich (TUM), Germany. His PhD thesis was awarded with the Kurt-Fischer-Prize by the Department of Electrical Engineering and Information Technology, TUM, in 2014. His main research interests include the development of testing and design methods for networked control and cyberphysical systems with applications for automotive systems.

Hasan Esen is a Technical Manager in the Corporate R&D Department at DENSO AUTOMOTIVE Deutschland GmbH. He is responsible for the company's advanced control and system engineering R&D activities in Europe. He received his PhD in Control Engineering from the Technical University of Munich, Germany, MSc in Mechatronics from Technical University of Hamburg-Harburg, Germany, and a BSc in Mechanical Engineering from Technical Universtiy of Istanbul, Turkey.

Sandra Hirche (M03-SM11-F20) received the Diplom-Ingenieur degree in aeronautical engineering from Technical University Berlin, Germany, in 2002 and the Doktor-Ingenieur degree in electrical engineering from Technical University Munich, Germany, in 2005. From 2005 to 2007 she was awarded a Postdoc scholarship from the Japanese Society for the Promotion of Science at the Fujita Laboratory, Tokyo Institute of Technology, Tokyo, Japan. From 2008 to 2012 she has been an associate professor at Technical University Munich. She has been a TUM Liesel Beckmann Distinguished Professor since 2013 and heads the Chair of Information-Oriented Control in the Department of Electrical and Computer Engineering at Technical University Munich. Her main research interests include cooperative, distributed and networked control with applications in human-robot interaction, multi-robot systems, and general robotics. She has published more than 150 papers in international journals, books and refereed conferences. Dr. Hirche has served on the Editorial Boards of the IEEE Transactions on Control of Network Systems, IEEE Transactions on Control Systems Technology, and the IEEE Transactions on Haptics.