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

Real-Time Distributed Automation Of Road Intersections

Fabio Molinari * **Alexander Katriniok** ** **Jörg Raisch** *

* *Control Systems Group - Technische Universitat Berlin, Germany, (e-mail: molinari, raisch@control.tu-berlin.de)*
** *Ford Research & Innovation Center, 52072 Aachen, Germany, (e-mail: de.alexander.katriniok@ieee.org)*

Abstract: The topic of this paper is the design of a fully distributed and real-time capable control scheme for the automation of road intersections. State of the art Vehicle-to-Vehicle (V2V) communication technology is adopted. Vehicles distributively negotiate crossing priorities by running a Consensus-Based Auction Algorithm (CBAA-M). Then, each agent solves a nonlinear Model Predictive Control (MPC) problem that computes the optimal trajectory avoiding collisions with higher priority vehicles and deciding the crossing order. The scheme is shown to be real-time capable and able to respond to sudden priority changes, e.g. if a vehicle gets an emergency call. Simulations reinforce theoretical results.

Keywords: Autonomous Vehicles; Distributed control and estimation; Multi-agent systems; Multi-vehicle systems; Coordination of multiple vehicle systems.

1. INTRODUCTION

Among recent topics in automotive research, traffic automation is surely in the spotlight, see Baskar et al. (2011). At the current state of the art, automated vehicles (AV) base their maneuvers solely on sensor measurements and on predictions of other road users' movements, see, e.g., Fajardo et al. (2011). In the close future, thanks to the development of faster data connections like the 5G technology (see Hastürkoğlu and Lindenmeier (2017)), Vehicle-To-Vehicle (V2V) communication will be available. Letting vehicles communicate their future trajectories, rather then relying on predictions, significantly reduces uncertainties. Many control strategies have been proposed, which make use of V2V and let self-driving vehicles cross intersections without collisions. Campos et al. (2014) propose a distributed Model Predictive Control (MPC) scheme that determines vehicles' crossing order by solving two convex quadratic programs. Crossing order is established by means of some heuristics in Campos et al. (2017), thus pledging lower complexity and scalability (at the price of suboptimality). However, Campos et al. (2017), as well as Hult et al. (2016), require a central decision making for the crossing order, thus resulting in a non-fully distributed approach. A fully-distributed control scheme, in which vehicles distributively negotiate their crossing order can be found in Molinari and Raisch (2018). This is possible thanks to a Consensus-Based Auction Algorithm (CBAA-M) that let vehicles distributively agree on the crossing order. However, in all these contributions, real-time implementability is not explicitly addressed or guaranteed. This topic is analyzed in Katriniok et al. (2019), where a real-time capable distributed nonlinear MPC is designed. Each vehicle avoids collisions with vehicles having higher priority, and the crossing order is decided by the distributed MPCs. However, priorities are fixed and assigned in a centralized fashion.

Fully distributed and *real-time capable* are two paramount properties for AV's control schemes. In fact, besides being more robust to failures, a fully distributed solution (rather than a centralized one) does not exhibit an increasing complexity the more complex the road network becomes. On the other hand, real-time capability is an indispensable feature for practical implementation.

The topic of this paper is the design of a fully distributed and real-time capable control scheme for the automation of road intersections. The underlying idea is to bring together the CBAA-M algorithm by Molinari and Raisch (2018) for negotiating priorities and the real-time capable distributed nonlinear MPC scheme by Katriniok et al. (2019). The novelties carried by this paper are manifold: (i) unlike Katriniok et al. (2019), priorities are time-varying and distributively negotiated; (ii) differently than Molinari et al. (2019), vehicle trajectories and kinematics are not simplified; (iii) a proof of convergence for CBAA-M for a network topology modeled by a strongly connected directed graph is given (Molinari et al. (2019) proves convergence for connected undirected topologies); (iv) unlike Molinari and Raisch (2018), *priority* and *order* are two separate concepts. In fact, higher priority vehicles are not forced to cross earlier. The nonlinear MPC determines the crossing order, while avoiding collisions with higher priority vehicles. This gives an additional degree of freedom; moreover, (v) a real-time capability analysis for the scheme, given state-of-the-art technology, is addressed; (vi) the impact of the continuous negotiation on the overall result is also studied.

The remainder of the paper is structured as follows: Section 2 describes the problem, the kinematic model for vehicles, and the hierarchical control structure. Section 3

⋆ Parts of this work were funded by the Deutsche Forschungsgemeinschaft (DFG) within their priority programme SPP 1914 Cyber-Physical Networking (CPN), Grant Number RA516/12-1.

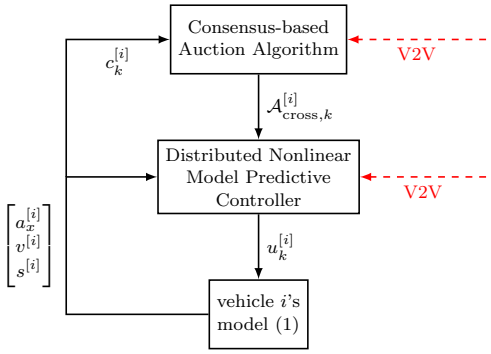

Fig. 1. Hierarchical control structure.

presents the CBAA-M and provides a proof of convergence. The nonlinear MPC is formalized in Section 4 and Section 5 contains simulation results. An analysis of real-time capability is given in Section 6. Final remarks and future work are the topic of Section 7.

2. PROBLEM DESCRIPTION

2.1 Intersection Coordination Problem

Our distributed control scheme relies on V2V communication. Each vehicle (or agent), after solving an optimal control problem for its own trajectory, transmits the result to other agents. We rely on the following assumptions, viable for in-vehicle implementation (Katriniok et al. (2019)):

Assumption 1. A1. Only single intersection scenarios are considered; A2. A single lane is available per direction; A3. The desired route of every agent together with its desired speed are determined by a high-level route planning algorithm; A4. Agents are equipped with V2V communication; A5. No communication failures or package dropouts occur; A6. The MPC solutions at time k are available to all agents at time $k+1$; A7. Vehicle states are measurable and not subject to uncertainty.

Joint satisfaction of collision avoidance (CA) constraints results in coupling between pairs of agents. This requires, traditionally, the solution of a dual optimization problem (which carries extra complexity). In Katriniok et al. (2019), to fully decouple the system, only one agent per each pair of conflicting agents imposes CA constraints. To this end, each agent is given a fixed priority. This way, each agent holds CA constraints only towards vehicles with higher priority. However, this scheme assigns priorities based on initial conditions and in a centralized fashion and does not account for the current traffic state. To tackle the online assignment of priorities, the hierarchical control scheme in Fig. 1 is designed. By interacting with other traffic participant, each agent runs *CBAA-M* thus obtaining a set of higher priority vehicles. The nonlinear model predictive controller holds CA constraints towards higher priority vehicles, thus deciding the crossing order, and therefore the input to each vehicle's kinematics.

2.2 Kinematic Agent Model

We define the set of agents involved in the motion planning problem as $\mathcal{A} \triangleq \{1, \dots, N_A\}$ where N_A is a positive integer. Each agent $i \in \mathcal{A}$ is assumed to move along an *a priori* known path (Assumption A3) which is parameterized by its path coordinate $s^{[i]}$, see Fig. 2. For such kind of coordination problems, a simplified kinematic modeling approach is common in literature, see, e.g, Hult et al. (2016); Molinari and Raisch (2018). In this paper, however, the time evolution of the agent's velocity $v^{[i]}$ and path coordinate $s^{[i]}$ are described by a double integrator model whereas drivetrain dynamics is modeled by a first-order lag element and acceleration is denoted by $a_x^{[i]}$. This way, agent kinematics is modeled by the following linear time-invariant state space model

$$\frac{d}{dt}\begin{bmatrix} a_x^{[i]} \\ v^{[i]} \\ s^{[i]} \end{bmatrix} = \underbrace{\begin{bmatrix} -1/T_{a_x}^{[i]} & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}}_{A^{[i]}} \underbrace{\begin{bmatrix} a_x^{[i]} \\ v^{[i]} \\ s^{[i]} \end{bmatrix}}_{\mathbf{x}^{[i]}} + \underbrace{\begin{bmatrix} 1/T_{a_x}^{[i]} \\ 0 \\ 0 \end{bmatrix}}_{B^{[i]}} \underbrace{a_{x,\text{ref}}^{[i]}}_{u^{[i]}}, \quad (1)$$

where $T_{a_x}^{[i]}$ stands for the dynamic drivetrain time constant and $u^{[i]} = a_{x,\text{ref}}^{[i]}$ is the reference acceleration (sent to the actuator). System (1) is discretized using a zero-order hold, thus yielding

$$\mathbf{x}_{k+1}^{[i]} = A_d^{[i]}\mathbf{x}_k^{[i]} + B_d^{[i]}u_k^{[i]}, \quad (2)$$

where $A_d^{[i]} = e^{A^{[i]}T_s}$ and $B_d^{[i]} = \int_0^{T_s} e^{A^{[i]}\tau}B^{[i]}\mathrm{d}\tau$. While the kinematic agent model (2) describes Agent i's motion along its local path coordinate $s^{[i]}$, the respective global coordinates $(x_g^{[i]}(s^{[i]}), y_g^{[i]}(s^{[i]}))$ are used to formulate CA constraints. $\mathcal{F}_p^{[i]} : s^{[i]} \mapsto (x_g^{[i]}, y_g^{[i]})$ relates the local path coordinate $s^{[i]}$ to the corresponding global Cartesian coordinates. Likewise, the heading angle $\psi^{[i]}(s^{[i]})$ and the path curvature $\kappa^{[i]}(s^{[i]})$ are yielded by $\mathcal{F}_\psi^{[i]} : s^{[i]} \mapsto \psi^{[i]}$ and $\mathcal{F}_\kappa^{[i]} : s^{[i]} \mapsto \kappa^{[i]}$, respectively. A thorough description of such function can be found in (Katriniok et al., 2019, Sec. IIb).

2.3 Intersection Model

The intersection is divided in regions as in (Katriniok et al., 2019, Sec. IIc), see Fig. 2. In the *intersection control region* (ICR), that is, for $s_{\text{icr,in}}^{[i]} \leq s^{[i]} < s_{\text{icr,out}}^{[i]}$, the control scheme needs to avoid collisions with crossing agents and agents driving ahead in the same lane. When entering the *brake safe region* (BSR), defined by $s_{\text{bsr,in}}^{[i]} \leq s^{[i]} < s_{\text{bsr,out}}^{[i]}$, agents are still able to stop safely before the *critical region* (CR). Only in the CR, i.e., $s_{\text{cr,in}}^{[i]} \leq s^{[i]} < s_{\text{cr,out}}^{[i]}$, collisions with crossing agents may happen. When the CR has been passed, only rear-end collision avoidance needs to be enforced.

3. OBTAINING PRIORITIES

CBAA-M allows vehicles (agents) to negotiate priorities in a fully distributed fashion. At every sampling time k, agents in set $\{i \in \mathcal{A} \mid s_k^{[i]} \leq s_{\text{cr,out}}^{[i]}\}$ participate in a distributed auction and bid for having the highest possible priority. The biddable quantity is determined by agent's velocity and position. The underlying communication network topology at sampling time $k \in \mathbb{N}_0$ is modeled by the directed graph $(\mathcal{A}, \mathcal{C}(k))$, where $\mathcal{C}(k)$ is the set of arcs, i.e. $(i,j) \in \mathcal{C}(k)$ iff at sampling time k vehicle i transmits

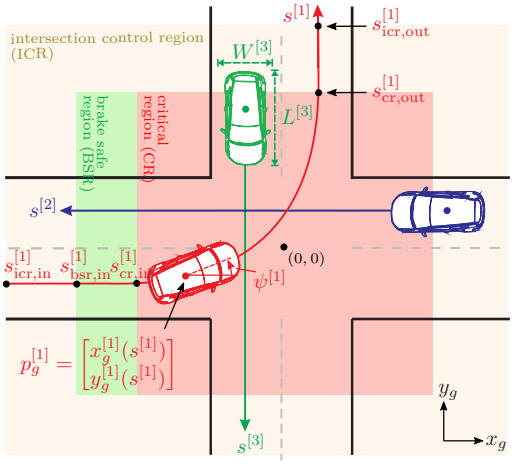

Fig. 2. Sketch of the intersection in global coordinates (x_g, y_g), with the origin at $(0,0)$. Intersection regions are illustrated for Agent 1, who is moving along its path coordinate $s^{[1]}$: inside ICR (beige), BSR (green), CR (red) and outside ICR (white).

information to vehicle j. The distributed technology allowing for a fast and reliable communication is discussed in Section 6. Note that employing a distributed algorithm allows vehicles to take all decisions, without the need of a centralized and expensive V2I (Vehicle to Infrastructure) technology. The result of this algorithm is that, at every $k \in \mathbb{N}_0$, each vehicle $i \in \mathcal{A}$ obtains a set of higher priority vehicles, i.e., $\mathcal{A}^{[i]}_{\text{cross},k} \subset \mathcal{A}$, towards which it will enforce CA constraints.

3.1 Bid computation

Reasonably, faster approaching vehicles (or vehicles closer to the BSR) should obtain higher priorities than vehicles driving more slowly (or being further away from the BSR). Moreover, a vehicle already inside the BSR must have higher priority than vehicles still outside. Accordingly, each vehicle $i \in \mathcal{A}$ at every sampling instant $k \in \mathbb{N}_0$ determines its own bid as follows:

$$c^{[i]}_k := \begin{cases} \alpha_1 v^{[i]}_k + \dfrac{\alpha_2}{(s^{[i]}_{bsr,in} - s^{[i]}_k)} & \text{if } s^{[i]}_{bsr,in} - s^{[i]}_k > \alpha_4 \\ \alpha_3 (s^{[i]}_k - s^{[i]}_{bsr,in}) + \alpha_5 & \text{else} \end{cases},$$

where $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5 \in \mathbb{R}_{>0}$ are design parameters.

Remark 1. $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5 \in \mathbb{R}_{>0}$ are chosen such that vehicles inside the BSR have always larger bids than vehicles outside, as in Fig. 3.

Assumption 2. For all distinct pairs of vehicles $i, j \in \mathcal{A}$, $\forall k \in \mathbb{N}_0$, $c^{[j]}_k \neq c^{[i]}_k$.

3.2 CBAA-M

CBAA-M is composed of two subsequent phases, a local auction phase in which vehicles place their bids, and a cooperative phase in which vehicles agree on the auction result. Let $\kappa \in \mathbb{N}$ denote the algorithm iteration. Each vehicle $i \in \mathcal{A}$ has two vectors, i.e., $\mathbf{v}^\kappa_i \in \{0 \dots N_A\}^{N_A}$ (containing the sorted list of agents) and $\mathbf{w}^\kappa_i \in \mathbb{R}^{N_A}_{>0}$ (containing the sorted list of bids), both initialized as null-vectors of dimension N_A. In order to avoid confusion of

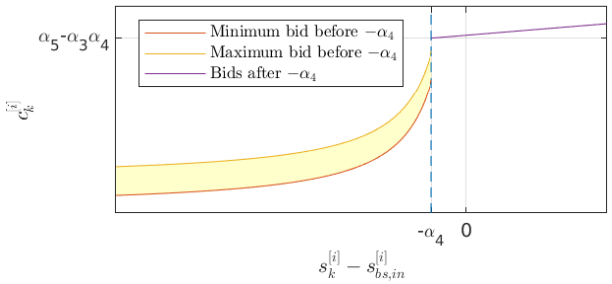

Fig. 3. Bid as function of vehicle's distance to the BSR. For $s^{[i]}_{bs,in} - s^{[i]}_k > \alpha_4$, the yellow area denotes all possible bids (since the bid is also a function of the speed). In this setting, $\alpha_1 = \frac{1}{10}, \alpha_2 = 5, \alpha_3 = \frac{1}{10}, \alpha_4 = 1, \alpha_5 = 7$.

time indices, for the analysis, we drop the time index k from $c^{[i]}_k$, thus focusing only on algorithm iteration index κ.

Phase 1. Local Auction: at every iteration κ, each $i \in \mathcal{A}$ places, if its index is not already stored in $\mathbf{v}^\kappa_i$, its own bid $c^{[i]}$ in the earliest possible position of vector $\mathbf{w}^\kappa_i$. In the same position, it stores its index in vector $\mathbf{v}^\kappa_i$.

Phase 2. Consensus over the lists: after the local auction, each agent has its own version of $\mathbf{v}^\kappa_i$ and $\mathbf{w}^\kappa_i$. The network needs to agree on them. To this end, each agent $i \in \mathcal{A}$ transmits its vectors to agents in set $\mathcal{N}^{\text{out}}_i \triangleq \{j \in \mathcal{A} \mid (i,j) \in \mathcal{C}(k)\}$ (namely, set of out-neighbors) and receives the vectors from agents in set $\mathcal{N}^{\text{in}}_i \triangleq \{j \in \mathcal{A} \mid (j,i) \in \mathcal{C}(k)\}$ (namely, set of in-neighbors). Then, via a max-consensus protocol, it selects the best bid for each row of $\mathbf{w}^\kappa_i$ and puts in the same position of $\mathbf{v}^\kappa_i$ the respective agent's index. After terminating Phase 2, $(\mathbf{w}^\kappa_i)_j$ is the j-th highest bid that Agent i is aware of at iteration κ, and $(\mathbf{v}^\kappa_i)_j$ is the index of the agent having placed that bid.

Algorithm 1 Phase 1. Local Auction: agents store locally their respective bid in the earliest possible position.

$\mathbf{v}^0_i = \mathbf{0}_{N_A}$, $\mathbf{w}^0_i = \mathbf{0}_{N_A}$
procedure BID($c^{[i]}$, $\mathbf{v}^{\kappa-1}_i$, $\mathbf{w}^{\kappa-1}_i$)
  $\mathbf{v}^\kappa_i \leftarrow \mathbf{v}^{\kappa-1}_i$
  $\mathbf{w}^\kappa_i \leftarrow \mathbf{w}^{\kappa-1}_i$
  loop:
  if $i \neq (\mathbf{v}^\kappa_i)_j, j = 1 \dots N_A$ **then**
    for $j = 1 \dots N_A$ **do**
      if $c^{[i]} > (\mathbf{w}^{\kappa-1}_i)_j$ **then**
        $(\mathbf{v}^\kappa_i)_j \leftarrow i$
        $(\mathbf{w}^\kappa_i)_j \leftarrow c^{[i]}$

Algorithm 2 Phase 2. Consensus: agents agree on the auction result.

SEND $(\mathbf{v}^\kappa_i, \mathbf{w}^\kappa_i)$ to $j \in \mathcal{N}^{\text{out}}_i$
RECEIVE $(\mathbf{v}^\kappa_h, \mathbf{w}^\kappa_h)$ from $h \in \mathcal{N}^{\text{in}}_i$
procedure UPDATE($\{\mathbf{v}^\kappa_h\}_{h \in \mathcal{N}^{\text{IN}}_i \cup \{i\}}$, $\{\mathbf{w}^\kappa_h\}_{h \in \mathcal{N}^{\text{IN}}_i \cup \{i\}}$)
  for $j = 1 \dots N_A$ **do**
    if $\max_h((\mathbf{w}^\kappa_h)_j) > 0$ **then**
      $(\mathbf{a}^\kappa_i)_j \leftarrow \arg\max_h((\mathbf{w}^\kappa_h)_j)$
      $(\mathbf{v}^\kappa_i)_j \leftarrow (\mathbf{v}^\kappa_{(\mathbf{a}^\kappa_i)_j})_j$
      $(\mathbf{w}^\kappa_i)_j \leftarrow \max_h((\mathbf{w}^\kappa_h)_j)$

3.3 Proof of convergence

The following proposition extends Molinari et al. (2019), as it shows convergence for strongly connected directed graphs.

Proposition 1. A multiagent system with a strongly connected network topology runs CBAA-M. Then, $\exists \bar{\kappa} \in \mathbb{N}$:

$$\forall i, j \in \mathcal{A},\ \forall \kappa > \bar{\kappa}, \quad \mathbf{v}_i^{\kappa} = \mathbf{v}_j^{\kappa} = \mathbf{v}^{\star} = \operatorname{argsort}(\mathbf{c}), \quad \mathbf{w}_i^{\kappa} = \mathbf{w}_j^{\kappa} = \mathbf{w}^{\star} = \operatorname{sort}(\mathbf{c}),$$

where $\mathbf{c}_i = c^{[i]}$ and $\bar{\kappa} \leq N_A \ell$, where $\ell \triangleq \max_{i,j}(p_{ij})$, and p_{ij} is the number of arcs in the shortest path [1] from j to i..

Proof. It is immediate to verify that, if $\exists \kappa_1$ such that, for one $i \in \mathcal{A}$, $\mathbf{v}_i^{\kappa_1} = \mathbf{v}^{\star}$ and $\mathbf{w}_i^{\kappa_1} = \mathbf{w}^{\star}$, then $\forall \kappa > \kappa_1$, $\mathbf{v}_i^{\kappa} = \mathbf{v}^{\star}$ and $\mathbf{w}_i^{\kappa} = \mathbf{w}^{\star}$ (see Molinari et al. (2019) for an extended discussion).

Denote by $\iota_1 \in \mathcal{A}$ the first agent obtaining the solution, i.e. $\mathbf{v}_{\iota_1}^{\kappa_1} = \mathbf{v}^{\star}$ and $\mathbf{w}_{\iota_1}^{\kappa_1} = \mathbf{w}^{\star}$ for some $\kappa_1 \in \mathbb{N}_0$ (at the corresponding Phase 1). Agent ι_1 is clearly the first agent that stores, in the Phase 1 of iteration κ_1, the value $\min(\mathbf{c})$ in the last entry of vector $\mathbf{w}_{\iota_1}^{\kappa_1}$. Doing that, ι_1 stores also its own index in the last entry of $\mathbf{v}_{\iota_1}^{\kappa_1}$, thus proving $\iota_1 = \arg\min(\mathbf{c})$, namely ι_1 is the agent with the lowest bid. This is possible only if, at the end of Phase 2 of $\kappa - 1$, $\forall j = 1 \dots N_A - 1$, $(\mathbf{w}_{\iota_1}^{\kappa_1 - 1})_j = (\mathbf{w}^{\star})_j$ (respectively, $(\mathbf{v}_{\iota_1}^{\kappa_1 - 1})_j = (\mathbf{v}^{\star})_j$).

Let now $\iota_2 \in \mathcal{A}$ be the first agent such that

$$\forall j = 1 \dots N_A - 1, \qquad (\mathbf{w}_{\iota_2}^{\kappa_2})_j = (\mathbf{w}^{\star})_j$$

(respectively, $(\mathbf{v}_{\iota_2}^{\kappa_2})_j = (\mathbf{v}^{\star})_j$), for some $\kappa_2 \in \mathbb{N}_0$. As of above, it is also clear that $\iota_2 = \arg\min(\tilde{\mathbf{c}})$, where $\tilde{\mathbf{c}} := \{c \in \mathbf{c} \mid c > \min \mathbf{c}\}$, namely ι_2 is the agent with the second smallest bid.

By Nejad et al. (2009), since Phase 2 is based on max-consensus, propagating the first $(N_A - 1)$ entries of ι_2's vector starting at iteration κ_2 through the whole network (thus also to ι_1) requires at most $\ell \in \mathbb{N}$ iterations, where ℓ is the so-called *diameter of the graph*. This yields that $\kappa_1 \leq \kappa_2 + \ell$. Applying this recursively for every entry of the solution vectors, yields

$$\kappa_1 \leq \underbrace{\ell + \dots + \ell}_{N_A - 1},$$

equivalently, $\kappa_1 \leq (N_A - 1)\ell$. Propagating $\mathbf{v}^{\star}$ and $\mathbf{w}^{\star}$ to the network takes again at most ℓ iterations. This shows that $\bar{\kappa} \leq \kappa_1 + \ell \leq N_A \ell$, thus concluding the proof.

4. DISTRIBUTED MOTION PLANNING

As outlined in Section 2.1, all vehicles' objectives (tracking and comfort) and some of their constraints (actuator constraints) are independent of other agents. Conversely, CA constraints couple different agent's optimal control problems (OCPs). As in Katriniok et al. (2019), a primal decomposition technique is used to distribute the motion planning problem.

4.1 Separable Objectives and Constraints

Objectives. The local objectives of each agent, say $i \in \mathcal{A}$ are (i) to minimize the deviation of the agent's speed $v^{[i]}$ from the desired speed $v_{\text{ref}}^{[i]}$, and (ii) to ensure comfortable and efficient driving by minimizing the acceleration $u^{[i]} = a_{x,\text{ref}}^{[i]}$. The sum of these objectives along the prediction horizon (of length N) given time k is, $\forall j = 0, \dots, N-1$,

$$\ell_j^{[i]}(\mathbf{x}_{k+j|k}^{[i]}, u_{k+j|k}^{[i]}) \triangleq q^{[i]} \left(v_{k+j|k}^{[i]} - v_{\text{ref},k+j|k}^{[i]}\right)^2 + r^{[i]} \left(u_{k+j|k}^{[i]}\right)^2. \tag{3}$$

The terminal cost is

$$\ell_N^{[i]}(\mathbf{x}_{k+N|k}^{[i]}) \triangleq q_N^{[i]} \left(v_{k+N|k}^{[i]} - v_{\text{ref},k+N|k}^{[i]}\right)^2 \tag{4}$$

where $q^{[i]} > 0$, $q_N^{[i]} > 0$ and $r^{[i]} > 0$ are positive weights.

Constraints. To accommodate actuator limitations, the demanded longitudinal acceleration is bounded, i.e.,

$$u_{k+j|k}^{[i]} \in \mathcal{U}^{[i]} \triangleq \left\{ u \in \mathbb{R} \mid \underline{a}_x^{[i]} \leq u \leq \overline{a}_x^{[i]} \right\} \tag{5}$$

for $j = 0, \dots, N-1$ where $\overline{a}_x^{[i]}$ and $\underline{a}_x^{[i]}$ are upper and lower bounds. Vehicles should not drive backwards nor exceed the maximum speed, namely $\overline{v}^{[i]}$, thus

$$\mathbf{x}_{k+j|k}^{[i]} \in \mathcal{X}_{k+j|k}^{[i]} \triangleq \left\{ \mathbf{x} \in \mathbb{R}^3 \mid 0 \leq (\mathbf{x})_2 \leq \overline{v}^{[i]} \right\}, \tag{6}$$

for $j = 1, \dots, N$.

To guarantee comfort and vehicle stability while turning, the lateral acceleration $a_y^{[i]} \triangleq \kappa^{[i]}(s^{[i]}) v^{[i]^2}$ is constrained by

$$-\overline{a}_y^{[i]} \leq \kappa^{[i]}(s_{k+j|k}^{[i]}) \cdot (v_{k+j|k}^{[i]})^2 \leq \overline{a}_y^{[i]} \tag{7}$$

for $j = 1, \dots, N$ with an appropriate upper bound $\overline{a}_y^{[i]}$. Due to vehicle stability, the total acceleration should never exceed a reasonable maximum of $\overline{a}_{\text{tot}}^{[i]}$ to stay within the friction circle, see, e.g., Rajamani (2012), namely,

$$(a_{x,k+j|k}^{[i]})^2 + \left(\kappa^{[i]}(s_{k+j|k}^{[i]}) \cdot (v_{k+j|k}^{[i]})^2\right)^2 \leq (\overline{a}_{\text{tot}}^{[i]})^2 \tag{8}$$

for $j = 1, \dots, N$.

4.2 Coupling Constraints: Collision Avoidance

To decouple CA constraints, only one vehicle per pair of possibly colliding vehicles enforces the CA. To this end, we need to distinguish between collisions with crossing agents and rear-end collisions.

For each pair of **crossing agents**, say i and l, we leverage the output of CBAA-M. By virtue of Section 3, set $\mathcal{A}_{\text{cross},k}^{[i]}$ contains higher priority agents still inside of the CR. In contrast to Katriniok et al. (2019), $\mathcal{A}_{\text{cross},k}^{[i]}$ is now time-varying instead of being *a priori* fixed.

To avoid **rear-end collisions**, only the following agent imposes CA constraints towards the preceding agent. For each agent $i \in \mathcal{A}$, at each sampling time k, the set $\mathcal{A}_{\text{ahead}}^{[i]} \subset \mathcal{A}$ defines agents that are, currently, in the same lane and ahead of Agent i.

Agent i at time k imposes CA constraints on vehicles, depending on the particular scenario:

(1) Agent i inside of the ICR: CA constraints are imposed on agents

$$l \in \mathcal{A}_{c,k}^{[i]} \triangleq \mathcal{A}_{\text{cross},k}^{[i]} \cup \mathcal{A}_{\text{ahead}}^{[i]}. \tag{9}$$

[1] A path in $(\mathcal{A}, \mathcal{C}(k))$ is a sequence of nodes, such that each pair of adjacent nodes is connected by a directed arc.

(2) Agent i outside of the ICR: only rear-end CA constraints are imposed on agents

$$l \in \mathcal{A}_{c,k}^{[i]} \triangleq \mathcal{A}_{\text{ahead}}^{[i]} \tag{10}$$

All vehicles impose CA constraints towards current frontal vehicles. Additionally, vehicles inside of the ICR need to enforce CA constraints also towards higher priority vehicles.

Remark 2. It can happen that, at sampling time k_0, one agent, say i, crosses earlier than a higher priority agent, say l. This way, at time $k > k_0$, $(s_k^{[i]} - s_{bs,in}^{[i]}) > (s_k^{[l]} - s_{bs,in}^{[l]})$, which implies, by Section 3.1, that i ends up obtaining higher priority than l.

For every agent l in the conflict set $\mathcal{A}_{c,k}^{[i]}$ at time k, we examine the area overlap of Agent i's safety region and Agent l's bounding box, namely $A^{i,l}$, see Fig. 4. Agent i's safety region is composed of a fixed *basic* safety region and a *motion dependent* safety region which depends on the relative motion with respect to Agent l. Collision avoidance is ensured if the overlap is zero. To this end, we introduce the equality constraint

$$A_{k+j|k}^{i,l} = 0, \ \forall l \in \mathcal{A}_{c,k}^{[i]}. \tag{11}$$

for every time step $k+j$, $j = 1, \dots, N$ over the prediction horizon. We present a thorough analysis of this latter constraint in (Katriniok et al., 2019, Sec.IIIb).

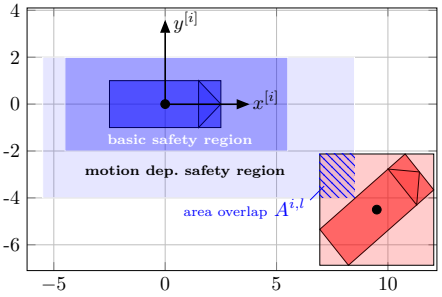

Fig. 4. Agent i's safety region along with Agent l's bounding box in Agent i's Cartesian body frame.

4.3 Minimum Spatial Preview

To ensure collision avoidance, the spatial preview of every agent i, that is the lookahead in meters along the path coordinate $s^{[i]}$, has to be long enough. Results contained in Katriniok et al. (2019) show that each agent, say i, has to leave the CR, at the latest, at the final time step $k+N$ of the prediction horizon, that is, $s_{k+N|k}^{[i]} \geq s_{\text{cr,out}}^{[i]}$. If this is not possible, then Agent i must stop before the stopping line, namely $s_{\text{stop}}^{[i]}$, before proceeding to the CR. This constraint can be expressed as

$$\left[-s_{k+N|k}^{[i]} + s_{\text{cr,out}}^{[i]}\right]_+ \cdot \left[s_{k+N|k}^{[i]} - s_{\text{stop}}^{[i]}\right]_+ = 0, \tag{12}$$

where $[x]_+ \triangleq \max\{0, x\}$.

4.4 Optimal Control Problem

By (Assumption A6), conflicting agents $l \in \mathcal{A}_{c,k}^{[i]}$ have transmitted at time $k-1$ their optimized position, velocity and heading trajectories, namely,

$$(x_{g,\cdot|k-1}^{[l],\star}, y_{g,\cdot|k-1}^{[l],\star}, \psi_{\cdot|k-1}^{[l],\star}, v_{\cdot|k-1}^{[l],\star}).$$

With this pieces of information at hand, every agent $i \in \mathcal{A}$ solves the following Optimal Control Problem (OCP) at time k

$$\underset{\{u_{k+j|k}\}_{j=0}^{N-1}}{\text{minimize}} \ \ell_N^{[i]}(\mathbf{x}_{k+N|k}^{[i]}) + \sum_{j=0}^{N-1} \ell_j^{[i]}(\mathbf{x}_{k+j|k}^{[i]}, u_{k+j|k}^{[i]}) \tag{13a}$$

$$\text{s.t.} \ \ \mathbf{x}_{k+j+1|k}^{[i]} = A_d^{[i]} \mathbf{x}_{k+j|k}^{[i]} + B_d^{[i]} u_{k+j|k}^{[i]} \tag{13b}$$

$$u_{k+j|k}^{[i]} \in \mathcal{U}^{[i]}, \qquad j = 0, \dots, N-1 \tag{13c}$$

$$\mathbf{x}_{k+j|k}^{[i]} \in \mathcal{X}_{k+j|k}^{[i]}, \qquad j = 1, \dots, N \tag{13d}$$

$$-\overline{a}_y^{[i]} \leq a_{y,k+j|k}^{[i]} \leq \overline{a}_y^{[i]}, \quad j = 1, \dots, N \tag{13e}$$

$$(a_{\text{tot},k+j|k}^{[i]})^2 \leq (\overline{a}_{\text{tot}}^{[i]})^2, \quad j = 1, \dots, N \tag{13f}$$

$$A_{k+j|k}^{i,l} = 0, \ \forall l \in \mathcal{A}_{c,k}^{[i]}, \quad j = 1, \dots, N \tag{13g}$$

$$\left[-s_{k+N|k}^{[i]} + s_{\text{cr,out}}^{[i]}\right]_+ \cdot \left[s_{k+N|k}^{[i]} - s_{\text{stop}}^{[i]}\right]_+ = 0, \tag{13h}$$

$$\mathbf{x}_{k|k}^{[i]} = \mathbf{x}_k^{[i]}, \tag{13i}$$

where $a_{\text{tot},k+j|k}^{[i]} = [(a_{x,k+j|k}^{[i]})^2 + (a_{y,k+j|k}^{[i]})^2]^{1/2}$ is the total acceleration in (8) and $a_{y,k+j|k}^{[i]} = \kappa^{[i]}(s_{k+j|k}^{[i]}) \cdot (v_{k+j|k}^{[i]})^2$ the lateral acceleration in (7). At every time instant k, Agent i solves the OCP (13), thus yielding the sequence of optimal control inputs $(u_{k|k}^{[i],\star}, \dots, u_{k+N-1|k}^{[i],\star})$, whose first element, $u_{k|k}^{[i],\star}$, is applied to the plant. After optimization, the resulting optimized trajectories $(x_{g,\cdot|k}^{[i]\star}, y_{g,\cdot|k}^{[i]\star}, \psi_{\cdot|k}^{[i]\star}, v_{\cdot|k}^{[i]\star})$ are transmitted to the other agents via V2V communication.

Due to (13f), (13g) and (13h), Problem (13) is nonconvex. By replacing equality and inequality constraints with penalty functions, see Nocedal and Wright (2006), it has been shown in (Katriniok et al., 2019, Sec. IV) that (13) can be recast as a box constrained nonconvex problem of the form

$$\underset{u_{\cdot|k}^{[i]} \in U_k^{[i]}}{\text{minimize}} \ \phi^{[i]}(u_{\cdot|k}^{[i]}; z_k^{[i]}). \tag{14}$$

where $\phi^{[i]}$ is a continuous differentiable function. We apply the proximal averaged Newton method for optimal control (PANOC) (Sathya et al. (2018); Stella et al. (2017)) to compute a stationary point, that is, a local solution of (14). To enforce constraint satisfaction, the quadratic penalty method, see (Nocedal and Wright, 2006, Chap. 17), is applied. In (14), $u_{\cdot|k}^{[i]} = [u_{k|k}^{[i]}, \dots, u_{k+N-1|k}^{[i]}]^\top$ is the vector of predicted control actions of Agent i, and

$$z_k^{[i]} = [x_k^{[i],\top}, (x_{g,\cdot|k-1}^{[l],\top}, y_{g,\cdot|k-1}^{[l],\top}, \psi_{\cdot|k-1}^{[l],\top}, v_{\cdot|k-1}^{[l],\top})_{l \in \mathcal{A}_{c,k}^{[i]}}]^\top$$

is a parameter vector which provides to Agent i all necessary measured information. A detailed analysis of the method is contained in (Katriniok et al., 2019, Sec.IV).

5. SIMULATION RESULTS

5.1 Simulation Setup

A realistic intersection scenario with four agents as shown in Fig. 5a is considered. Agent 1 (red) crosses the intersection straight from North to South, Agent 2 (blue)

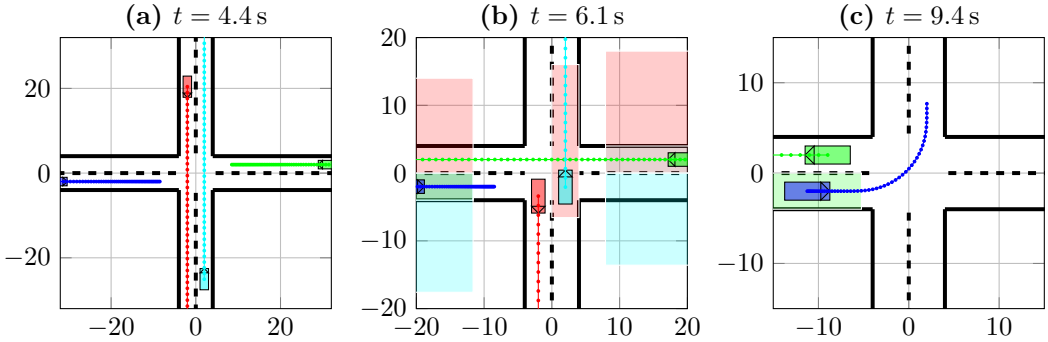

Fig. 5. Snapshots Use Case 1: (Left) Agent 1 (red) and Agent 4 (cyan) cross first; (Middle) Agent 3 (green) proceeds after Agent 4 (cyan); (Right) Agent 2 (blue) turns left after Agent 3 (green) has left the CR. The middle and right figures show the safety region of each agent i in the color of the conflicting Agent l.

approaches the intersection from the West and turns left while Agent 3 (green) and Agent 4 (cyan) crosses the intersection straight from East to West and South to North, respectively. Each agent has the same dimensions of $L^{[i]} = 5\,\mathrm{m}$ and $W^{[i]} = 2\,\mathrm{m}$ and the same drivetrain time constant of $T_{a_x}^{[i]} = 0.3\,\mathrm{s}$. The initial positions in the global frame are: $(-2, 82)$ for Agent 1, $(-84, -2)$ for Agent 2, $(81, 2)$ for Agent 3 and $(2, -84)$ for Agent 4. Moreover, all agents have the same initial and reference velocity of $14\,\mathrm{m/s}$ while the maximum speed is $15\,\mathrm{m/s}$. Sampling time is $T_s = 0.1\,\mathrm{s}$. MPC's prediction horizon consists of $N = 50$ steps. MPC's weights are chosen equally for every agent, that is, $q^{[i]} = q_N^{[i]} = 1$ and $r^{[i]} = 20$. Agents' safety region is parameterized as in Katriniok et al. (2019). Finally, we constrain the demanded longitudinal acceleration between -7 and $4\,\mathrm{m/s^2}$ while the absolute lateral acceleration has to be less or equal to $3.5\,\mathrm{m/s^2}$ and the total acceleration is bounded from above by $7\,\mathrm{m/s^2}$. All simulations are run on Intel i7 machine at 2.9 GHz with Matlab R2018b while the nonlinear MPC controllers run in `C89` using the open source code generation tool `nmpc-codegen`, available at `github.com/kul-forbes/nmpc-codegen`.

To evaluate the interplay of the auction based algorithm and the distributed MPC control scheme, we investigate two use cases for the scenario outlined above:

(1) Regular priority negotiation: the agents are negotiating priorities as in Section 3.
(2) Emergency vehicle: Agent 2 is an emergency vehicle, requesting the highest priority at $t = 0.5\,\mathrm{s}$.

5.2 Use Case 1: Regular Priority Negotiation

Fig. 7 illustrates in each row i the optimized motion trajectories of Agent i; Fig. 5 shows three snapshots of the maneuver in the global coordinate frame along with agents' safety regions. In addition, Fig. 6 highlights agents' negotiated priorities until 8 s (when, after agents enter their respective BSR, priorities turn out to be constant). Initially, Agent 1 (red) exhibits the highest priority, followed by Agent 3 (green), Agent 4 (cyan) and Agent 4 (blue). By getting closer to the intersection, at $t = 2\,\mathrm{s}$, Agent 1 (red) and Agent 4 (cyan), driving in the North/South direction, obtain the highest and second highest priority, respectively; Agent 2 (blue) and Agent 3 (green) are assigned the next higher priorities. At $t = 3.8\,\mathrm{s}$, Agent 3 (green) gets a higher priority than Agent 2 (blue), mainly because of Agent 2 (blue)'s deceleration before turning left. After $t = 4.1\,\mathrm{s}$, all agent priorities turn out to be fixed, since, inside of the CR, they are proportional only to the distance. Priorities also reflect in the average speeds. In fact, Agent 2 (blue) has both the lowest priority and the lowest average speed (same for other vehicles).
The optimized motion trajectories in Fig. 7 prove that the time-varying (until $t = 4\,\mathrm{s}$) negotiated priorities do not cause discontinuities in acceleration or velocity. By comparing Fig. 5 and Fig. 6, one can observe that the crossing order is determined by the negotiated priority. Initially, the acceleration of Agent 4 (cyan) is slightly negative; this is the case until it is assigned the 2nd highest priority after $t = 1\,\mathrm{s}$. Agent 2 (blue) yields the way to Agent 3 (green), see Fig. 5c, thus decelerating until before $t = 10\,\mathrm{s}$. On its way through the intersection, Agent 2 (blue) satisfies its lateral acceleration constraint (depicted in the third column). Concerning CA, the first column provides evidence that agent trajectories are safe as the distance between the agent's safety region and the other agent's bounding box is always greater or equal to zero. Moreover, velocities and accelerations are smooth and inside their designated bounds.

5.3 Use Case 2: Emergency Vehicle

Initial conditions and scenario of the Use Case 2 are the same as Use Case 1, apart from Agent 2 (blue) — assumed to be an ambulance — receiving an emergency call at $t = 0.5\,\mathrm{s}$. The topic of this section is to investigate how a sudden change in priority affects the entire traffic. Because

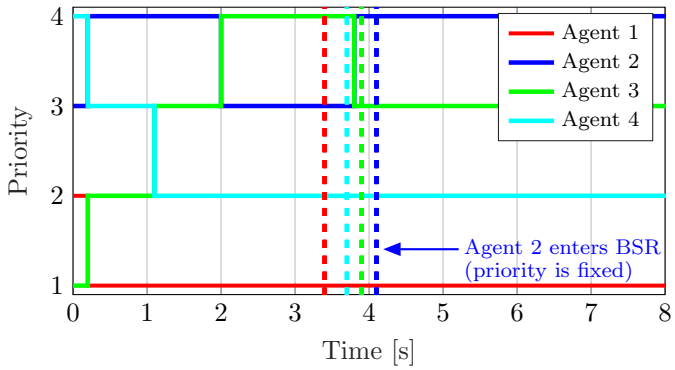

Fig. 6. Use Case 1: Agent priorities are negotiated until the agents enter their BSR (indicated by dashed vertical line), then they remain constant.

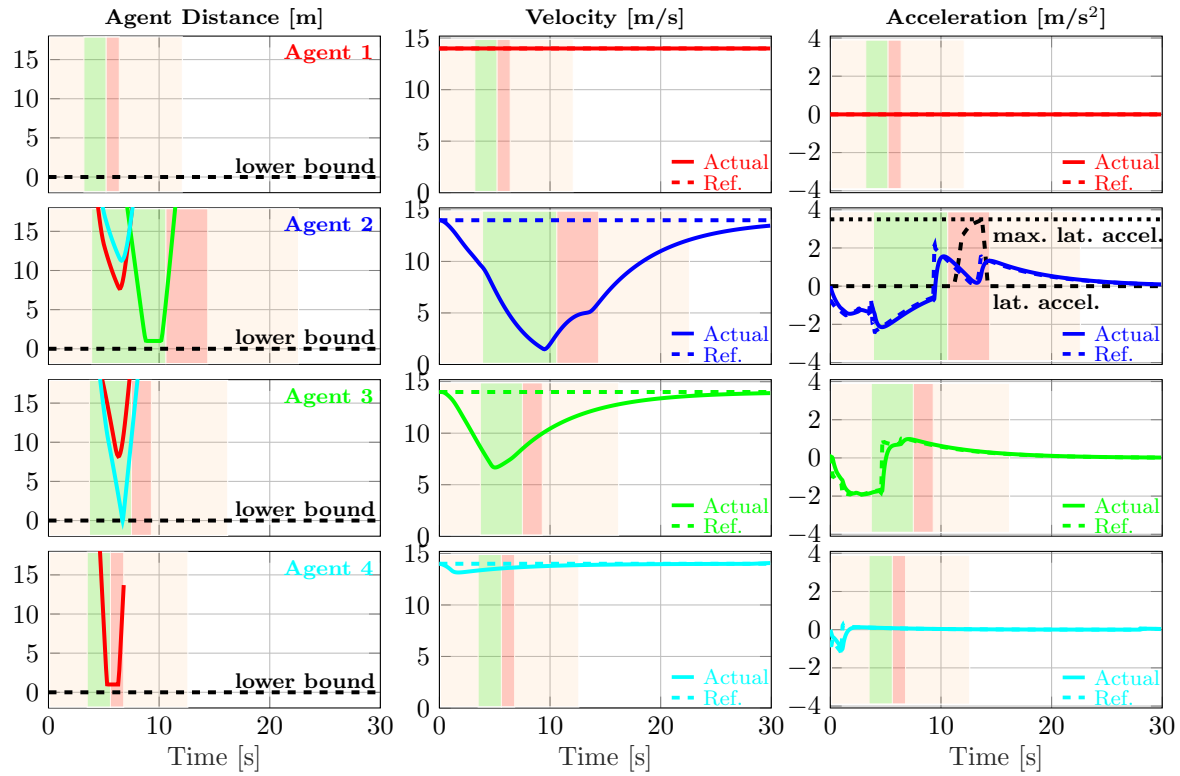

Fig. 7. Use Case 1: From left to right in row i: (1) Distances between safety region and bounding box of agents i and l, (2) velocity and (3) acceleration of Agent i. The intersection regions are indicated by colored patches: inside ICR (beige), BSR (green), CR (red) and outside ICR (white).

of the limited space, relevant trajectories and priorities are condensed in Fig. 8. Until $t = 0.5\,\mathrm{s}$, Agent 2 (blue) retains the 3rd negotiated priority. At $t = 0.5\,\mathrm{s}$, Agent 2 (blue) receives an emergency call, gets an arbitrarily high bid and negotiates the 1st priority. Fig. 8 confirms that acceleration and velocity do not show any discontinuities as priorities are changed suddenly. Agent 2 (blue) crosses the intersection without taking care of any agent. However, in contrast to Use Case 1, priorities do not determine the crossing order. In fact, Agent 3 (green) crosses first (the vertical lines in the priority plot indicate when the agents enter their BSR) without the need to decelerate. Clearly, by (9), Agent 3 (green) needs to enforce CA constraints inside of the intersection towards Agent 2 (blue). This is an advantage (and an additional *degree of freedom*) of our approach, with regards to, e.g., Molinari et al. (2019). From the above discussion, it is evident that requirements in terms of comfort, performance, and safety are satisfied.

6. REAL-TIME CAPABILITY

Real-time capability depends on two distinct components: the convergence of CBAA-M to a solution and the solving time of the nonlinear MPC.

Concerning CBAA-M, Proposition 1 states that, at each sampling time, the network gets to the agreement in at most $N_A \ell$ iterations. We deal with state-of-the-art communication technology, i.e., 5G network used for automated driving. The technical specification 3GPP.org (2019) [2] contains the newest global specifications for automated driving. Accordingly, for the scenario *Emergency trajectory alignment between UEs supporting V2X application*, the following performance aspects are defined:

- *reliability of communication links* is 99.999%;
- *max end-to-end latency* is 3 ms.

Because of the high link reliability, it is straightforward to assume the underlying topology to be *fully-connected*, thus $\ell = 1$. Agreement is achieved in N_A iterations, thus

$$T_{\mathrm{CBAAM}} \leq N_A \cdot 3 \text{ ms}.$$

For the case at hand, we assume indeed $T_{\mathrm{CBAAM}} = 12$ ms. Concerning the solving time of the nonlinear MPC, we refer to Fig. 9, which depicts agents' computation times for *Use Case 1*, simulated on a machine with the specifications in Section 5.1. T_{MPC} is at most 72 ms. Note that a dedicated setup would lead to an even faster convergence. However, also in a non-dedicated simulation environment, the overall execution time for the hierarchical controller is

$$T_{\mathrm{CBAAM}} + T_{\mathrm{MPC}} \leq 12 \text{ ms} + 72 \text{ ms} \leq 84 \text{ ms},$$

which is strictly less than the sampling time of 100 ms.

7. CONCLUSION

This paper has proposed a distributed and real-time capable control scheme for automating a road intersection. Vehicles participate in a distributed auction for determining priorities. Afterwards, a distributed nonlinear MPC enforces collision avoidance towards higher priority vehicles and determines the crossing order. Simulation results support the initial claims. State-of-the-art communication technology allows for real-time implementability.

[2] 3GPP is a standards organization which develops protocols for mobile telephony.

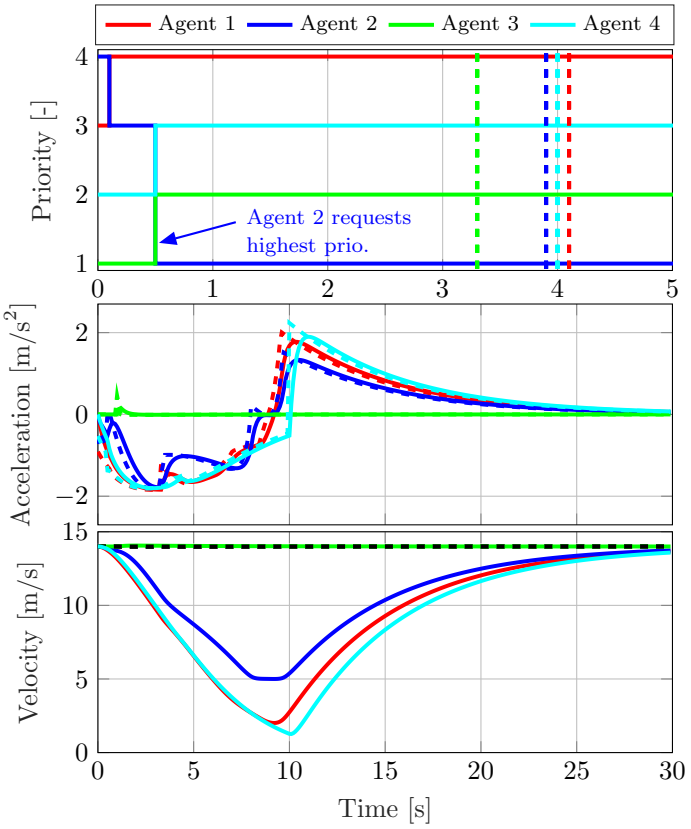

Fig. 8. Use Case 2: Emergency vehicle switches from priority 3 to highest priority at $t = 0.5\,\mathrm{s}$.

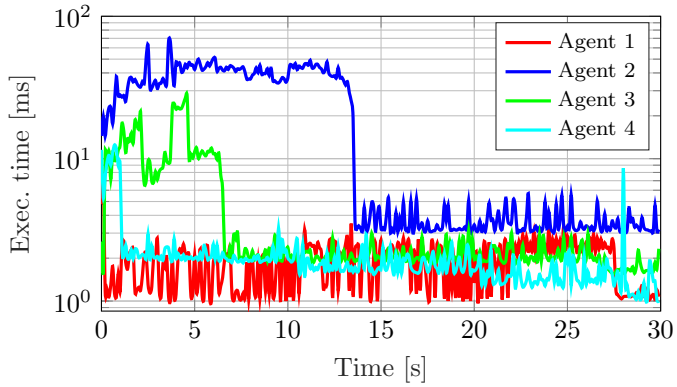

Fig. 9. Use Case 1: Agent computation times.

REFERENCES

3GPP.org (2019). Enhancement of 3GPP support for V2X scenarios. 3GPP TS 22.186, V16.2.0. URL `https://portal.3gpp.org /desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3180`.

Baskar, L.D., De Schutter, B., Hellendoorn, J., and Papp, Z. (2011). Traffic control and intelligent vehicle highway systems: a survey. *IET Intelligent Transport Systems*.

Campos, G.R., Falcone, P., Wymeersch, H., Hult, R., and Sjöberg, J. (2014). Cooperative receding horizon conflict resolution at traffic intersections. In *53rd IEEE Conference on Decision and Control*, 2932–2937.

Campos, G.R., Falcone, P., Hult, R., Wymeersch, H., and Sjöberg, J. (2017). Traffic coordination at road intersections: Autonomous decision-making algorithms using model-based heuristics. *IEEE Intelligent Transportation Systems Magazine*, 9(1), 8–21.

Fajardo, D., Au, T.C., Waller, S.T., Stone, P., and Yang, D. (2011). Automated intersection control: Performance of future innovation versus current traffic signal control. *Transportation Research Record*, 2259(1), 223–232.

Hastürkoğlu, S. and Lindenmeier, S. (2017). A wideband automotive antenna for actual and future mobile communication 5g/lte/wlan with low profile. In *2017 11th European Conference on Antennas and Propagation (EUCAP)*, 602–605. IEEE.

Hult, R., Zanon, M., Gros, S., and Falcone, P. (2016). Primal decomposition of the optimal coordination of vehicles at traffic intersections. In *Conference on Decision and Control*, 2567–2573.

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

Katriniok, A., Sopasakis, P., Schuurmans, M., and Patrinos, P. (2019). Nonlinear Model Predictive Control for Distributed Motion Planning in Road Intersections Using PANOC. In *IEEE Conference on Decision and Control*.

Molinari, F., Dethof, A.M., and Raisch, J. (2019). Traffic automation in urban road networks using consensus-based auction algorithms for road intersections. In *2019 18th European Control Conference (ECC)*, 3008–3015. IEEE.

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

Nejad, B.M., Attia, S.A., and Raisch, J. (2009). Max-consensus in a max-plus algebraic setting: The case of fixed communication topologies. In *2009 XXII International Symposium on Information, Communication and Automation Technologies*, 1–7. IEEE.

Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer, second edition.

Rajamani, R. (2012). *Vehicle Dynamics and Control*, volume 2. Springer.

Sathya, A., Sopasakis, P., Parys, R.V., Themelis, A., Pipeleers, G., and Patrinos, P. (2018). Embedded nonlinear model predictive control for obstacle avoidance using PANOC. In *ECC*, 1523–1528.

Stella, L., Themelis, A., Sopasakis, P., and Patrinos, P. (2017). A simple and efficient algorithm for nonlinear model predictive control. In *Conference on Decision and Control*, 1939–1944.