2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

Nonlinear Model Predictive Control for Distributed Motion Planning in Road Intersections Using PANOC

Alexander Katriniok, Pantelis Sopasakis, Mathijs Schuurmans, Panagiotis Patrinos

Abstract— **The coordination of highly automated vehicles (or agents) in road intersections is an inherently nonconvex and challenging problem. In this paper, we propose a distributed motion planning scheme under reasonable vehicle-to-vehicle communication requirements. Each agent solves a nonlinear model predictive control problem in real time and transmits its planned trajectory to other agents, which may have conflicting objectives. The problem formulation is augmented with conditional constraints that enable the agents to decide whether to wait at a stopping line, if safe crossing is not possible. The involved nonconvex problems are solved very efficiently using the proximal averaged Newton method for optimal control (PANOC). We demonstrate the efficiency of the proposed approach in a realistic intersection crossing scenario.**

I. Introduction

Automated vehicles (AV) today either operate in a *reactive* way by solely basing their actions on sensor readings or exploit an uncertain estimate of future motion trajectories of other road users to be more *proactive*. With vehicle-to-vehicle (V2V) communication, more reliable information about future trajectories of the surrounding traffic can be adopted to operate the AV more efficiently [1]. Especially in intersections, throughput could significantly be improved when, instead of using traffic lights or signs, vehicles would negotiate intersection crossing using V2V communication.

In this paper, we address the control problem of coordinating vehicles (referred to as *agents*) in intersections by means of distributed nonlinear model predictive control (MPC), assuming that V2V communication is available for information exchange. To this end, optimization-based strategies such as MPC appear to be appropriate to deal with constrained motion planning problems incorporating anticipated trajectories of conflicting agents.

In recent years, MPC has been widely used for this purpose. The authors in [2] introduce a decentralized control scheme which relies on the solution of two convex quadratic programs (QP) to determine the order in which the agents cross the intersection. An MPC-based scheme is proposed in [3], which utilizes a central coordination unit to solve a high-level time slot allocation optimal control problem (OCP), while agent controls are determined as part of a nested low-level OCP. Essentially, every agent solves a QP and two linear programs (LP) and transmits the optimization results to the central coordinator which solves the high-level nonlinear program (NLP). This is extended in [4] with rear-end collision avoidance. A similar scheme is proposed in [5] where a centralized coordination unit is in charge of time slot allocation, while agents are controlled in a decentralized fashion and all involved OCPs are convex QPs. Moreover, [6] presents a decentralized consensus-based control strategy which determines the intersection crossing order as part of a high-level consensus algorithm and solves a convex optimization problem to determine vehicle controls on the lower level. Instead of using time slots, collision avoidance is ensured by bounding distances between agents from below. In [7], the authors have outlined a fully distributed MPC scheme where every agent solves a nonconvex quadratically constrained QP (QCQP). Instead of utilizing time slots, agents must keep a minimum distance to each other — defined with respect to their joint collision point.

A. Main Contribution and Outline

To orchestrate AVs in intersections, we rely on a distributed MPC scheme in which every agent solves its nonconvex optimal control problem simultaneously and broadcasts the optimized trajectories to the other agents via V2V communication. We believe that a distributed scheme is more flexible, resilient and scalable than a centralized one when it eventually comes to in-vehicle implementation.

In the literature, most control concepts rely on a formulation in which collision avoidance is tightly related to the intersection scenario, that is, through time slots or collision points [2], [3], [7]. For AV motion control, it is desirable to utilize a single controller for multiple scenarios. Therefore, we take a first step to generalize our problem formulation in [7] to be able to cover a wider range of use cases. Instead of using joint collision points along the agents' path coordinate, we propose to formulate collision avoidance in a Cartesian frame by mapping the agents' path coordinate to their respective global Cartesian coordinates. The authors in [6] are pursuing this direction using a linear mapping. By using B-spline functions, we allow for nonlinear mapping functions which are necessary to describe arbitrary realistic driving maneuvers. Eventually, we utilize the area overlap of the agents' bounding boxes (plus some safety margin) to define collision avoidance (CA) constraints. This formulation can easily be applied independent from intersections, e.g., for rear-end collision avoidance or lane change maneuvers.

A. Katriniok is with Ford Research & Innovation Center, 52072 Aachen, Germany, de.alexander.katriniok@ieee.org.

P. Sopasakis is with Queen's University Belfast, School of Electronics, Electrical Engineering and Computer Science, Centre For Intelligent Autonomous Manufacturing Systems (i-AMS), Belfast, Northern Ireland, UK, p.sopasakis@qub.ac.uk.

M. Schuurmans and P. Patrinos are with the Department of Electrical Engineering (ESAT-STADIUS), KU Leuven, 3001 Leuven, Belgium, mathijs.schuurmans@kuleuven.be, panos.patrinos@esat.kuleuven.be.

978-1-7281-1397-5/19/$31.00 ©2019 IEEE

Moreover, we introduce a methodology to embed *conditional constraints* into the motion planning problem, such as waiting at the stopping line if safe intersection crossing is impossible. Finally, we propose to exploit the proximal averaged Newton-type method for optimal control (PANOC) [8], [9] to solve the resulting nonconvex NLP in real-time.

The remainder of the paper is organized as follows: Section II introduces the vehicle kinematics model along with the underlying assumptions and the relationship between local and global coordinate frames. The distributed motion planning problem is then formulated in section III, while section IV outlines how an efficient numerical solution is obtained. Simulation results are finally discussed in section V.

B. Notation

Hereafter, $x_{k+j|k}$ will stand for the prediction of variable x at the future time step $k+j$ given information up to time k. For $x \in \mathbb{R}^n$ and $i \in \{1, \dots, n\}$, $[x]_i$ is the i-th entry of x. In addition, $\mathbb{N}^+$ is the set of positive integers, $[x]_+ \triangleq \max\{x, 0\}$ and $A^\top$ denotes the transpose of a matrix $A \in \mathbb{R}^{m \times n}$.

II. Modeling

We rely on the following fundamental assumptions:

Assumption 1. A1. Only single intersection scenarios are considered; A2. A single lane is available per direction; A3. The desired route of every agent is determined by a high-level route planning algorithm (Sec. II-B); 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.

Assumptions A1, A2, A5 and A7 are common in the literature and are used to reduce complexity [2], [3]. The use of a high-level planning algorithm which is postulated in A3 is quite common in AV architectures too [10]. Lastly, A4 is necessary for a distributed control scheme and A6 can be satisfied by choosing the MPC sampling time appropriately.

A. Vehicle Kinematics

Let $\mathcal{A} \triangleq \{1, \dots, N_A\}$ be the set of agents where N_A is a positive integer. It is a common approach in the literature to use Assumption A3 to define the kinematics of Agent $i \in \mathcal{A}$ along paths which are parametrized by a scalar $s^{[i]}$ as shown in Fig. 1 [3], [6]. That said, the time evolution of velocity $v^{[i]}$ and path coordinate $s^{[i]}$ are described by a double integrator, while the drivetrain dynamics is modeled as a first-order system, yielding the following linear time-invariant state space representation

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

where $T_{a_x}^{[i]}$ denotes the dynamic drivetrain time constant and $u^{[i]} = a_{x,\text{ref}}^{[i]}$ is the requested acceleration (acceleration set-point). We may derive the exact discretization of (1) using zero-order hold, that is, $A_d^{[i]} = e^{A^{[i]}T_s}$ and $B_d^{[i]} = \int_0^{T_s} e^{A^{[i]}\tau} B^{[i]} \mathrm{d}\tau$ where T_s is the sampling time.

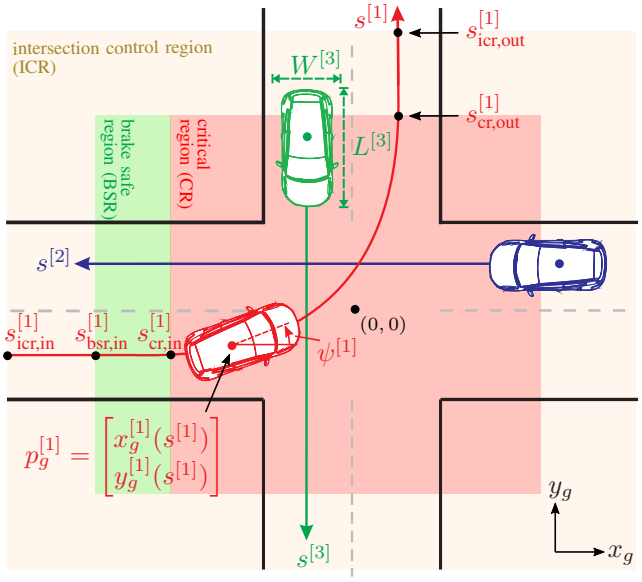


Fig. 1: Schematic of the intersection in the global frame, (x_g, y_g), with its origin $(0, 0)$ at the center of the intersection. For Agent 1, the respective intersection regions are illustrated: inside ICR (beige), BSR (green), CR (red) and outside ICR (white). The reference trajectory of Agent 1 is described by a B-spline.

B. Local and Global Coordinate Frames

Equation (1) describes the motion of every agent with respect to their local coordinate frame along the path coordinate $s^{[i]}$. In [7], collision avoidance is encoded in this local frame with respect to a joint collision point with another agent. This approach may not be suitable in multi-lane scenarios when a certain lateral distance to surrounding agents should be maintained. Then, a problem formulation in a global Cartesian frame appears to be more general and convenient. We, therefore, define a mapping function $\mathcal{F}_p^{[i]} : s^{[i]} \mapsto (x_g^{[i]}, y_g^{[i]})$, which maps the path coordinate $s^{[i]}$ of Agent i to its global Cartesian coordinates. Functions $\mathcal{F}_p^{[i]}$ can be chosen to be B-splines [11]

$$\begin{bmatrix} x_g^{[i]}(s^{[i]}) \\ y_g^{[i]}(s^{[i]}) \end{bmatrix} = \mathcal{F}_p^{[i]}(s^{[i]}) \triangleq \begin{bmatrix} \sum_{l=0}^{n_p} \alpha_{x,l}^{[i]} B_{x,l}^{[i],n}(s^{[i]}) \\ \sum_{l=0}^{n_p} \alpha_{y,l}^{[i]} B_{y,l}^{[i],n}(s^{[i]}) \end{bmatrix}, \quad (2)$$

where $B_{x,l}^{[i],n}$ and $B_{y,l}^{[i],n}$ are B-spline basis polynomials of degree n while $\alpha_{x,l}^{[i]}$ and $\alpha_{y,l}^{[i]}$ are spline coefficients. We assume that the initial condition $s^{[i]}(t_0) = 0$ holds. For natural driving maneuvers, we use a spline degree of $n = 3$. The number of spline coefficients, n_p, depends on the number of path points used for interpolating the agent's path [11]. According to Assumption A3, $\mathcal{F}_p^{[i]}$ is provided by a high-level route planning algorithm. Similarly, we assume that the heading angle $\psi^{[i]}(s^{[i]})$ and the path curvature $\kappa^{[i]}(s^{[i]})$ are provided as $\mathcal{F}_\psi^{[i]} : s^{[i]} \mapsto \psi^{[i]}$ and $\mathcal{F}_\kappa^{[i]} : s^{[i]} \mapsto \kappa^{[i]}$ respectively, either as separate spline curves or derived from the first and second derivative of $\mathcal{F}_p^{[i]}$ respectively.

C. Intersection Regions

For agent coordination, we divide the area around the intersection in regions as shown in Fig. 1. In the *intersection*

control region (ICR), that is, for $s^{[i]}_{\text{icr,in}} \leq s^{[i]} < s^{[i]}_{\text{icr,out}}$, the aim is to avoid collisions with crossing agents and to ensure rear-end collision avoidance at the same time. The ICR can further be subdivided into the *brake safe region* (BSR) and the *critical region* (CR). In the BSR, defined as $s^{[i]}_{\text{bs,in}} \leq s^{[i]} < s^{[i]}_{\text{cr,in}}$, it is always safe to stop before entering the CR. Only in the CR, that is, for $s^{[i]}_{\text{cr,in}} \leq s^{[i]} < s^{[i]}_{\text{cr,out}}$, collisions with crossing agents may happen and these need to be avoided by the control scheme. After leaving the CR and outside the ICR, only rear-end collisions must be prevented.

III. Distributed Motion Planning Problem

Following [7], the motion planning problem is separable with respect to the agents' individual objectives and constraints while only collision avoidance couples the agents among each other. This way, we separate the subproblems using a primal decomposition technique.

A. Local Agent Objectives and Constraints

As local objectives, the motion planning regime should 1) maintain a desired speed (typically, close to the speed limit) and 2) enable efficient and comfortable driving. The former requirement translates into minimizing the deviation of the agent's speed, $v^{[i]}$, from a reference speed $v^{[i]}_{\text{ref}}$, while the latter is equivalent to reducing the demanded acceleration $u^{[i]} = a^{[i]}_{x,\text{ref}}$. Along a horizon of N steps, we can encode these requirements in the following stage cost at time $k+j$ for $j = 0, \dots, N-1$

$$\ell^{[i]}_j(x^{[i]}_{k+j|k}, u^{[i]}_{k+j|k}) \triangleq q^{[i]} \left(v^{[i]}_{k+j|k} - v^{[i]}_{\text{ref},k+j|k}\right)^2 + r^{[i]} \left(u^{[i]}_{k+j|k}\right)^2 \tag{3}$$

and the terminal cost

$$\ell^{[i]}_N(x^{[i]}_{k+N|k}) \triangleq q^{[i]}_N \left(v^{[i]}_{k+N|k} - v^{[i]}_{\text{ref},k+j|k}\right)^2 \tag{4}$$

where $q^{[i]} > 0$, $q^{[i]}_N > 0$ and $r^{[i]} > 0$ are positive weights. Besides local objectives, local agent constraints need to be accommodated as well. First, to account for actuator limitations, we constrain the demanded longitudinal acceleration by the input constraint

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

with some appropriate upper and lower bound $\overline{a}^{[i]}_x$ and $\underline{a}^{[i]}_x$, for $j = 0, \dots, N-1$.

Moreover, every agent should account for the speed limit on the current road section while driving backwards is prohibited. We encode this requirement as a state constraint of the form

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

with the upper bound $\overline{v}^{[i]}_{k+j|k}$, for $j = 1, \dots, N$.

For turning agents, the lateral acceleration $a^{[i]}_y$, being equal to the product of the curvature $\kappa^{[i]}(s^{[i]})$ with the squared velocity $v^{[i]}$, is bounded as follows

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

for $j = 1, \dots, N$, where $\overline{a}^{[i]}_y$ is an appropriately chosen upper bound. Additionally, it needs to be ensured that the total acceleration does not exceed the friction circle [12], that is,

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

for $j = 1, \dots, N$ and an appropriate upper bound $\overline{a}^{[i]}_{\text{tot}}$.

B. Collision Avoidance

1) Agent Conflict Sets: While the constraints in section III-A refer to the individual agent, CA constraints couple the agents among each other.

For crossing agents, we introduce time-invariant priorities on the agents that are determined once and held constant during the maneuver [7]. We define the bijective prioritization function $\gamma : \mathcal{A} \to \mathcal{A}$, which assigns a unique priority to every agent — a lower value corresponds to a higher priority. This way, we specify the set of agents $l \in \mathcal{A}$ which can collide with Agent i, but have a higher priority $\gamma(l) < \gamma(i)$, that is,

$$\mathcal{A}^{[i]}_{c,\gamma} \triangleq \left\{ l \in \mathcal{A} \mid \gamma(l) < \gamma(i) \right\}. \tag{9}$$

We refer to $\mathcal{A}^{[i]}_{c,\gamma}$ as the time-invariant *prioritized conflict set*. Moreover, we denote $\bar{\mathcal{A}}^{[i]}_{c,\gamma,k} \subseteq \mathcal{A}^{[i]}_{c,\gamma}$ as the set of higher priority agents that have not yet left the CR at time k.

In case of rear-end collision avoidance, we define $\mathcal{A}^{[i]}_{c,\text{ahead}}$ as the time-invariant set that contains the agents that are always in the same lane and ahead of Agent i. Furthermore, $\bar{\mathcal{A}}^{[i]}_{c,\text{ahead},k}$ refers to the time-varying set of agents $l \in \mathcal{A}$ at time k that have been crossing (i.e., $l \in \mathcal{A}^{[i]}_{c,\gamma}$ or $i \in \mathcal{A}^{[l]}_{c,\gamma}$) and are now in the same lane and ahead of Agent i.

To fully decouple the agents, we impose CA constraints only on one of two conflicting agents. Therefore, we consider the following cases at time k: **a)** If Agent i is **inside** the **ICR** and has **not yet passed the CR**, it imposes CA constraints on crossings agents and those that are driving ahead, i.e., on agents $l \in \mathcal{A}^{[i]}_{c,k} = \mathcal{A}^{[i]}_{c,\gamma} \cup \mathcal{A}^{[i]}_{c,\text{ahead}}$; **b)** If Agent i is **inside** the **ICR** and has **passed the CR**, it needs to avoid collisions with agents $l \in \mathcal{A}^{[i]}_{c,k} = \mathcal{A}^{[i]}_{c,\text{ahead}} \cup \bar{\mathcal{A}}^{[i]}_{c,\text{ahead},k} \cup \bar{\mathcal{A}}^{[i]}_{c,\gamma,k}$, that is, with those driving ahead and with higher priority agents that may be in the same lane and behind Agent i after leaving the CR; **c)** If Agent i is **outside** the **ICR**, we only impose rear-end CA constraints on agents $l \in \mathcal{A}^{[i]}_{c,k} = \mathcal{A}^{[i]}_{c,\text{ahead}} \cup \bar{\mathcal{A}}^{[i]}_{c,\text{ahead},k}$ that drive in the same lane and ahead of Agent i.

2) Safety Regions and CA Constraint Formulation: For each agent $i \in \mathcal{A}$, we define a safety region around the vehicle for every $l \in \mathcal{A}^{[i]}_{c,k}$, which must not be intersected by the bounding box of the conflicting Agent l. We denote by

$$n^{[l]}_\psi = R^{[i],\top}_\psi R^{[l]}_\psi \begin{bmatrix} 1 & 0 \end{bmatrix}^\top \tag{10}$$

the unit vector pointing in the direction of motion of Agent l in the Cartesian body frame of Agent i where $R^{[i]}_\psi$ and $R^{[l]}_\psi$ are the respective rotation matrices — see Fig. 2. We then define the safety region in the longitudinal direction in front of (xf) and behind (xr) Agent i as

$$d^{[i]}_{\text{safe,xf}} \triangleq \tilde{d}^{[i]}_{\text{safe,xf}} + v^{[i]} t^{[i]}_{\text{gap},x}, \tag{11a}$$

$$d^{[i]}_{\text{safe,xr}} \triangleq \tilde{d}^{[i]}_{\text{safe,xr}} + v^{[i]} t^{[i]}_{\text{gap},x} \cdot \max\left\{0, [1\ 0] n^{[l]}_\psi\right\}, \tag{11b}$$

where $\tilde{d}^{[i]}_{\text{safe,xf}}$ and $\tilde{d}^{[i]}_{\text{safe,xr}}$ are basic safety distances which are independent of the agents' motion. In addition, we want the agents to keep a velocity-dependent safety distance where $t_{\text{gap},x}$ is the respective time gap. The rear safety distance $d^{[i]}_{\text{safe,xr}}$ should only be increased if Agent l is driving in the same direction as Agent i, and not if it is crossing. That said, in (11b), $[1\ 0]n^{[l]}_{\psi}$ is the projection of Agent l's heading vector onto Agent i's body frame x-axis.

In the lateral direction, i.e., left (yl) and right (yr) from Agent i, we consider the basic safety distances $\tilde{d}^{[i]}_{\text{safe,yl}}$ and $\tilde{d}^{[i]}_{\text{safe,yr}}$ which are independent of the agents' motion. If Agent l is moving from the side towards Agent i, the safety distance increases depending on the speed and heading of Agent l, that is,

$$d^{[i]}_{\text{safe,yl/yr}} \triangleq \tilde{d}^{[i]}_{\text{safe,yl/yr}} + v^{[l]} t^{[i]}_{\text{gap},y} \cdot \max\{0, \mp[0\ 1]n^{[l]}_{\psi}\}, \quad (12)$$

where $t_{\text{gap},y}$ is an appropriate time gap, and $[0\ 1]n^{[l]}_{\psi}$ is the projection of Agent l's heading vector onto Agent i's body frame y-axis. Applying this formulation, the safety distance to Agent l increases, even as it moves away from Agent i. In this case, though, the problem solution is not affected.

We now define the lower-left and upper-right corner points, $\underline{p}^{[i]}$ and $\overline{p}^{[i]}$ of Agent i's safety region (see Fig. 2) in its Cartesian body frame as

$$\underline{p}^{[i]} \triangleq \begin{bmatrix} -\frac{L^{[i]}}{2} - d^{[i]}_{\text{safe,xr}} \\ -\frac{W^{[i]}}{2} - d^{[i]}_{\text{safe,yr}} \end{bmatrix}, \ \overline{p}^{[i]} \triangleq \begin{bmatrix} \frac{L^{[i]}}{2} + d^{[i]}_{\text{safe,xf}} \\ \frac{W^{[i]}}{2} + d^{[i]}_{\text{safe,yl}} \end{bmatrix}, \quad (13)$$

where $L^{[i]}$ and $W^{[i]}$ are the length and width of Agent i, respectively. To determine collisions between agents, we examine the area overlap of Agent i's safety region and Agent l's bounding box. If there is no overlap, both agents are safe. If the edges of both rectangles are not perpendicular to each other, a closed-form expression is hard to determine. For that reason, we resort to over-approximating Agent l with a bounding box whenever the rectangle edges are not perpendicular to each other as shown in Fig. 2. We denote the lower right and upper left corner point of the over-approximated rectangle in Agent i's reference frame as $\underline{p}^{[l]}$ and $\overline{p}^{[l]}$. Then, the length $L^{i,l}$ and width $W^{i,l}$ of the overlapping area can easily be obtained as follows

$$L^{i,l} \triangleq \min\{[\overline{p}^{[i]}]_1, [\overline{p}^{[l]}]_1\} - \max\{[\underline{p}^{[i]}]_1, [\underline{p}^{[l]}]_1\}, \quad (14a)$$

$$W^{i,l} \triangleq \min\{[\overline{p}^{[i]}]_2, [\overline{p}^{[l]}]_2\} - \max\{[\underline{p}^{[i]}]_2, [\underline{p}^{[l]}]_2\}. \quad (14b)$$

If there is no overlap, $L^{i,l}$ and/or $W^{i,l}$ are less than zero. Therefore, the overlap of Agent i's safety region and Agent l's bounding box is

$$A^{i,l} \triangleq \max\{0, L^{i,l}\} \cdot \max\{0, W^{i,l}\}. \quad (15)$$

Finally, the following equality constraint needs to be satisfied to guarantee collision avoidance for every agent in the conflict set $\mathcal{A}^{[i]}_{c,k}$ (see section III-B) and for every time step $k+j$, $j = 1, \ldots, N$ over the prediction horizon:

$$A^{i,l}_{k+j|k} = 0, \ \forall l \in \mathcal{A}^{[i]}_{c,k}. \quad (16)$$

Remark 1. Over-approximating Agent l's bounding box does not entail any undesired conservatism. A tighter approximation, though, might be preferable for future use cases such as multi-lane scenarios.

C. Minimum Spatial Preview

To guarantee collision avoidance in the distributed setting, the spatial preview of every agent i has to be of sufficient length while crossing the intersection, see [7]. Particularly, if Agent i has approached its BSR, i.e., $s^{[i]}_{\text{bsr,in}} \leq s^{[i]}_{k} < s^{[i]}_{\text{cr,in}}$, it is safe to stop before entering the CR where collision can happen. If there are still crossing agents with higher priority that have not yet left the CR, that is, if $\bar{\mathcal{A}}^{[i]}_{c,\gamma,k} \neq \emptyset$, the preview of Agent i has at least to cover the agent's CR to avoid unforeseen conflicts with other agents. This requirement is equivalent to the constraint that Agent i leaves the CR at the latest at the final time step $k+N$ of the prediction horizon, i.e., $s^{[i]}_{k+N|k} \geq s^{[i]}_{\text{cr,out}}$. If this constraint cannot be satisfied, Agent i should stop before entering the CR. In essence, we need to encode the *conditional constraint*

$$\textbf{IF NOT } s^{[i]}_{k+N|k} \geq s^{[i]}_{\text{cr,out}} \textbf{ THEN } s^{[i]}_{k+N|k} \leq s^{[i]}_{\text{stop}}, \quad (17)$$

where $s^{[i]}_{\text{stop}} < s^{[i]}_{\text{cr,in}}$ refers to the stop line in the BSR. If all agents with higher priority have already left the critical region, we set $s^{[i]}_{\text{cr,out}}$ to zero, thus allowing Agent i to finally pass the critical region, referred to as liveness of the control scheme [13]. To encode conditional constraints of the form

$$\textbf{IF NOT } g(x) \geq 0 \textbf{ THEN } h(x) \leq 0, \quad (18)$$

with the constraint functions $g : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}$, we can write (18) equivalently as $\neg(g(x) \geq 0) \Rightarrow (h(x) \leq 0)$, or, what is the same

$$(-g(x) \leq 0) \vee (h(x) \leq 0) \quad (19)$$

where $\neg$ denotes the logical NOT operator. By using the plus operator $[\cdot]_+$, condition (19) can be cast as an equality constraint of the form

$$[-g(x)]_+ \cdot [h(x)]_+ = 0. \quad (20)$$

By virtue of (20), condition (17) can then be rewritten as

$$\left[-s^{[i]}_{k+N|k} + s^{[i]}_{\text{cr,out}}\right]_+ \cdot \left[s^{[i]}_{k+N|k} - s^{[i]}_{\text{stop}}\right]_+ = 0. \quad (21)$$

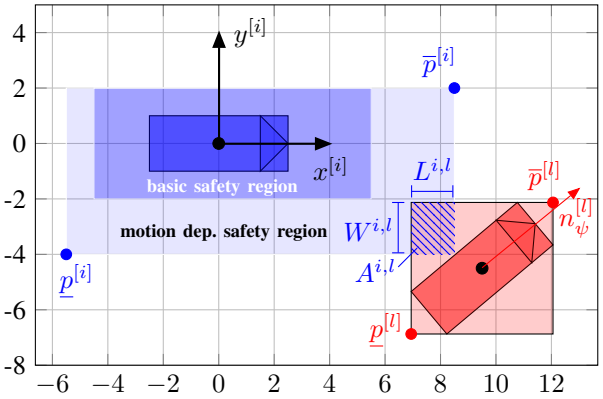


Fig. 2: Agent i's safety region (basic and motion dependent) along with Agent l's bounding box in Agent i's Cartesian body frame. $A^{i,l}$ represents their overlap.

D. Optimal Control Problem

Based on the optimized position, velocity and heading trajectories $(x^{[l]}_{g,\cdot|k-1}, y^{[l]}_{g,\cdot|k-1}, \psi^{[l]}_{\cdot|k-1}, v^{[l]}_{\cdot|k-1})$ of conflicting agents $l \in \mathcal{A}^{[i]}_{c,k}$ that have been transmitted at time $k-1$ (Assumption A6), every agent $i \in \mathcal{A}$ solves the following OCP at time k

$$\underset{\{u_{k+j|k}\}_{j=0}^{N-1}}{\text{minimize}} \ \ell_N^{[i]}(x^{[i]}_{k+N|k}) + \sum_{j=0}^{N-1} \ell_j^{[i]}(x^{[i]}_{k+j|k}, u^{[i]}_{k+j|k}) \quad (22a)$$

$$\text{s.t.} \ \ x^{[i]}_{k+j+1|k} = A_d^{[i]} x^{[i]}_{k+j|k} + B_d^{[i]} u^{[i]}_{k+j|k} \quad (22b)$$

$$u^{[i]}_{k+j|k} \in \mathcal{U}^{[i]}, \qquad j = 0, \dots, N-1 \quad (22c)$$

$$x^{[i]}_{k+j|k} \in \mathcal{X}^{[i]}_{k+j|k}, \qquad j = 1, \dots, N \quad (22d)$$

$$-\overline{a}_y^{[i]} \le a^{[i]}_{y,k+j|k} \le \overline{a}_y^{[i]}, \quad j = 1, \dots, N \quad (22e)$$

$$(a^{[i]}_{\text{tot},k+j|k})^2 \le (\overline{a}^{[i]}_{\text{tot}})^2, \qquad j = 1, \dots, N \quad (22f)$$

$$A^{i,l}_{k+j|k} = 0, \ \forall l \in \mathcal{A}^{[i]}_{c,k}, \quad j = 1, \dots, N \quad (22g)$$

$$\left[-s^{[i]}_{k+N|k} + s^{[i]}_{\text{cr,out}}\right]_+ \cdot \left[s^{[i]}_{k+N|k} - s^{[i]}_{\text{stop}}\right]_+ = 0, \quad (22h)$$

$$x^{[i]}_{k|k} = x^{[i]}_k, \quad (22i)$$

where $a^{[i]}_{\text{tot},k+j|k} = [(a^{[i]}_{x,k+j|k})^2 + (a^{[i]}_{y,k+j|k})^2]^{1/2}$ refers to the total acceleration in (8) and $a^{[i]}_{y,k+j|k} = \kappa^{[i]}(s^{[i]}_{k+j|k}) \cdot (v^{[i]}_{k+j|k})^2$ to the lateral acceleration in (7). At every time instant k, Agent i solves the above optimization problem and obtains an optimal sequence $(u^{[i],\star}_{k|k}, \dots, u^{[i],\star}_{k+N-1|k})$, the first element of which, $u^{[i],\star}_{k|k}$, is applied to the plant. Moreover, the agents' optimized trajectories $(x^{[i]\star}_{g,\cdot|k}, y^{[i]\star}_{g,\cdot|k}, \psi^{[i]\star}_{\cdot|k}, v^{[i]\star}_{\cdot|k})$ are transmitted to the other agents via V2V communication.

Due to (22f), (22g) and (22h), Problem (22) is nonconvex. It can, however, be reformulated in a form so that it can be solved efficiently in real time using PANOC.

IV. Fast Numerical NMPC Solution

The proximal averaged Newton-type method for optimal control (PANOC) has been proposed by the authors in [9] (see also [8], [14] for applications of PANOC to obstacle avoidance MPC). PANOC is a first-order method that combines projected gradient iterations with quasi-Newtonian directions for fast convergence. The key to guaranteeing global convergence of PANOC is the forward-backward envelope (FBE) introduced in [15] for convex problems and further extended to nonconvex problems in [16], [17].

To solve problem (22) for Agent i with PANOC, it is reformulated and cast in the form

$$\underset{u^{[i]}_{\cdot|k} \in U^{[i]}_k}{\text{minimize}} \ \phi_k^{[i]}(u^{[i]}_{\cdot|k}; z^{[i]}_k), \quad (23)$$

where $u^{[i]}_{\cdot|k} = [u^{[i]}_{k|k}, \dots, u^{[i]}_{k+N-1|k}]^\top$ is the vector of predicted control actions of Agent i, and $z^{[i]}_k = [x^{[i],\top}_k, (x^{[l],\top}_{g,\cdot|k-1}, y^{[l],\top}_{g,\cdot|k-1}, \psi^{[l],\top}_{\cdot|k-1}, v^{[l],\top}_{\cdot|k-1})_{l \in \mathcal{A}^{[i]}_{c,k}}]^\top$ is a parameter vector which provides to Agent i all necessary measured information. PANOC requires that $\phi_k^{[i]}$ are continuously differentiable functions in $u^{[i]}_{\cdot|k}$ with Lipschitz-continuous gradient and that sets $U_k^{[i]}$ are closed and such that we can easily compute projections thereon.

To that end, we first eliminate the state sequence using (22b), that is, we substitute

$$x^{[i]}_{k+j|k}(u^{[i]}_{\cdot|k}) = (A_d^{[i]})^j x^{[i]}_{k|k} + \sum_{\iota=0}^{j-1} (A_d^{[i]})^{j-1-\iota} B_d^{[i]} u^{[i]}_{k+\iota|k}.$$

For the input constraints (22c) we define the set $U_k^{[i]} \triangleq \{u^{[i]}_{\cdot|k} \mid u^{[i]}_{k+j|k} \in \mathcal{U}^{[i]}, j = 0, \dots, N-1\}$, which is a rectangle, because of the definition of $\mathcal{U}^{[i]}$ in (5). The remaining equality and inequality constraints, namely (22d) to (22h), are modeled as *soft constraints* and the quadratic penalty method [18, Chap. 17] is used to ensure their satisfaction.

In particular, equality constraints (22g) and (22h) can be concisely written as $h_s(u^{[i]}_{\cdot|k}) = 0$ for $s = 1, \dots, n_{\text{eq}}$. We introduce the associated penalty functions

$$\psi_{\text{eq},s}(u^{[i]}_{\cdot|k}, \beta_{\text{eq},s}) \triangleq \beta_{\text{eq},s} \cdot h_s(u^{[i]}_{\cdot|k})^2$$

with positive weights $\beta_{\text{eq},s} > 0$ for $s = 1, \dots, n_{\text{eq}}$. Inequality constraints can be concisely stated as $g_s(u^{[i]}_{\cdot|k}) \le 0$, for $s = 1, \dots, n_{\text{ineq}}$. For each such constraint, we define the penalty function

$$\psi_{\text{ineq},s}(u^{[i]}_{\cdot|k}, \beta_{\text{ineq},s}) \triangleq \beta_{\text{ineq},s} \cdot [g_s(u^{[i]}_{\cdot|k})]_+^2,$$

with positive weights $\beta_{\text{ineq},s} > 0$. This way, we define the modified cost

$$\begin{aligned} &\phi_k^{[i]}(u^{[i]}_{\cdot|k}; z^{[i]}_k, (\beta_{\text{eq},s})_s, (\beta_{\text{ineq},s})_s) \\ &\quad \triangleq \ell_N^{[i]}(x^{[i]}_{k+N|k}(u^{[i]}_{\cdot|k})) + \sum_{j=0}^{N-1} \ell_j^{[i]}(x^{[i]}_{k+j|k}(u^{[i]}_{\cdot|k}), u^{[i]}_{k+j|k}) \\ &+ \sum_{s=1}^{n_{\text{eq}}} \psi_{\text{eq},s}(u^{[i]}_{\cdot|k}, \beta_{\text{eq},s}) + \sum_{s=1}^{n_{\text{ineq}}} \psi_{\text{ineq},s}(u^{[i]}_{\cdot|k}, \beta_{\text{ineq},s}). \end{aligned} \quad (24)$$

Now, the problem is in the form (23) and can be solved using PANOC. Note that the constraints are satisfied if and only if $\psi_{\text{eq},s}(u^{[i]}_{\cdot|k}, \beta_{\text{eq},s}) = 0$ and $\psi_{\text{ineq},s}(u^{[i]}_{\cdot|k}, \beta_{\text{ineq},s}) = 0$ for some weight parameters and for all s. In order for the constraints to be satisfied, we select a tolerance $\epsilon_s > 0$ for the constraints and we require that $\psi_{\text{eq},s}/\beta_{\text{eq},s} < \epsilon_s$ for all s (similarly, $\psi_{\text{ineq},s}/\beta_{\text{ineq},s} < \epsilon_s$ for the inequality constraints). If this is not satisfied for some s, we multiply the corresponding weight by 10 and solve the problem, warm-starting with the previous approximate solution.

V. Simulation Results

A. Simulation Setup

To demonstrate the efficiency of our approach, we present a realistic intersection crossing scenario which involves four agents as illustrated in Fig. 4. According to section III-B, crossing priorities are time-invariant with $\gamma(1) = 3$, $\gamma(2) = 1$, $\gamma(3) = 4$ and $\gamma(4) = 2$. In the considered scenario, Agent 1 (red) crosses the intersection straight from North to South, Agent 2 (blue) approaches the intersection from the West and turns left while Agent 3 (green) and Agent

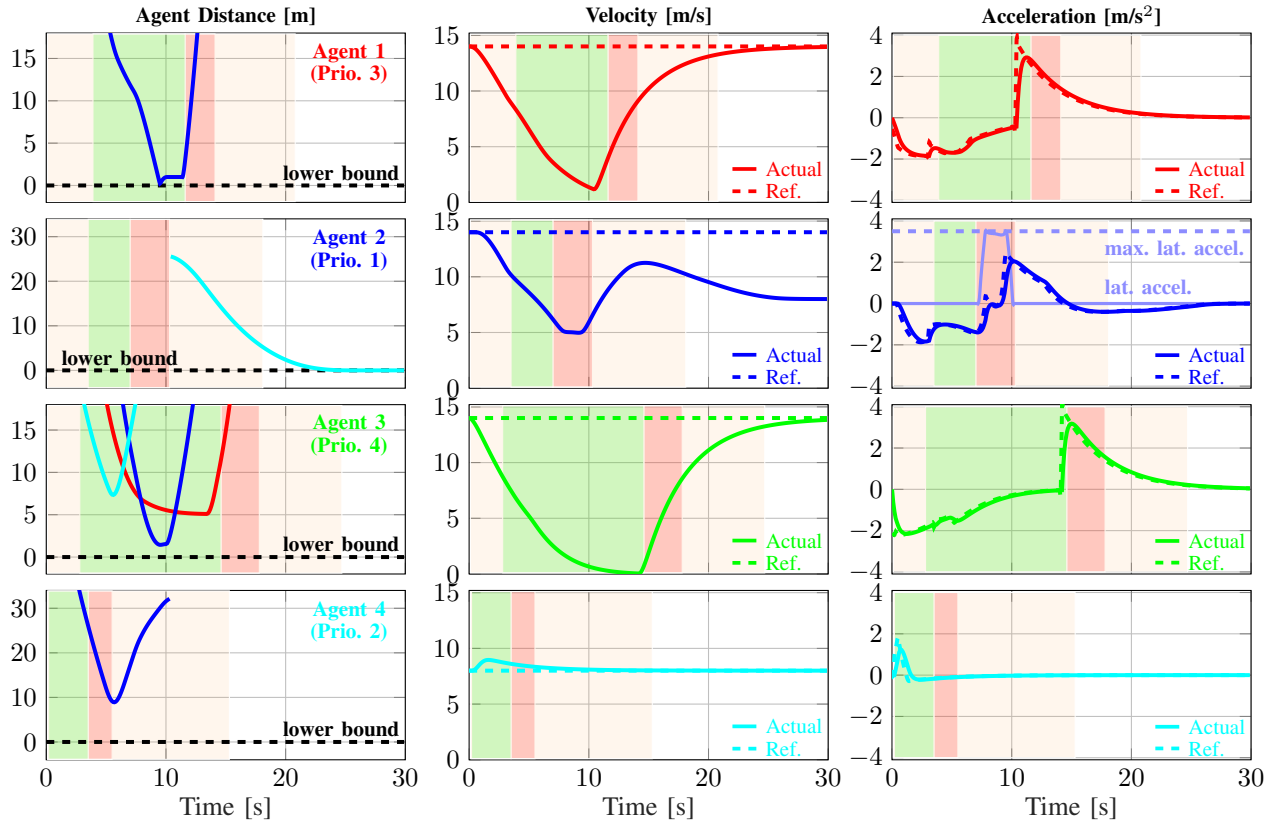


Fig. 3: From left to right in row i: (1) Distances between the safety region and bounding box of agents i and l, (2) velocity and (3) acceleration of Agent i. Agent 2 is turning left, therefore, we additionally depict its lateral acceleration. Color patches refer to intersection regions: inside ICR (beige), BSR (green), CR (red) and outside ICR (white).

4 (cyan) cross the intersection straight from East to West and South to North, respectively — as shown in Fig. 4a.

All agents have dimensions $L^{[i]} = 5\,\mathrm{m}$ and $W^{[i]} = 2\,\mathrm{m}$ and the same dynamic time constant of $T_{a_x}^{[i]} = 0.3\,\mathrm{s}$. The initial and reference velocity is $14\,\mathrm{m/s}$ for agents 1 to 3 and $8\,\mathrm{m/s}$ for Agent 4. The initial positions in the global frame are: $(-2, 82)$ for Agent 1, $(-82, -2)$ for Agent 2, $(69, 2)$ for Agent 3 and $(2, -39)$ for Agent 4. The agents' MPC controllers have the same parameterization: a sampling time of $T_s = 0.1\,\mathrm{s}$, a horizon length of $N = 50$ and the weights $q^{[i]} = q_N^{[i]} = 1$ and $r^{[i]} = 10$. The sampling time is chosen in accordance to the commonly applied V2V communication frequency of $10\,\mathrm{Hz}$. We parameterize the safety region by $\tilde{d}^{[i]}_{\mathrm{safe,xf/xr}} = 2\,\mathrm{m}$, $\tilde{d}^{[i]}_{\mathrm{safe,yl/yr}} = 1\,\mathrm{m}$ and $t^{[i]}_{\mathrm{gap},x/y} = 1\,\mathrm{s}$. Moreover, the longitudinal acceleration is constrained between -7 and $4\,\mathrm{m/s^2}$, the absolute lateral acceleration should not exceed $3.5\,\mathrm{m/s^2}$ and the total acceleration is bounded from above by $7\,\mathrm{m/s^2}$. The maximum velocity is set to $15\,\mathrm{m/s}$. For equality and inequality constraints, a tolerance of $\epsilon_s = 10^{-4}$ has been selected. Simulations are run on an Intel i7 machine at 2.9 GHz with Matlab R2018b, while the controllers run in `C89` using the open source code generation tool `nmpc-codegen`, available at `github.com/kul-forbes/nmpc-codegen`.

B. Discussion of Results

For the given scenario in row i, Fig. 3 illustrates the resulting trajectories and Fig. 4 shows three snapshots of the scenario along with the agents' optimized trajectories and their safety regions. As we may observe in Fig. 3 and Fig. 4, Agent 4 is able to cross the intersection before Agent 2 by increasing its velocity temporarily from $8\,\mathrm{m/s}$ to $9\,\mathrm{m/s}$. Indeed, the lower priority of Agent 4 allows the agent to cross the intersection before Agent 2. After Agent 2 has left the CR, the CA constraint on Agent 2 is dropped by Agent 4 and the distance in the first column is no longer determined.

While Agent 2 crosses the intersection, it has to decelerate in order to satisfy the constraint on its lateral acceleration, which is depicted in the third column (light blue solid) along with its upper bound (light blue dashed). When passing the CR, no CA constraints are imposed on other agents as Agent 2 has the highest priority and $\mathcal{A}^{[i]}_{\mathrm{c,ahead}} = \emptyset$. However, after leaving the CR, Agent 2 needs to avoid rear-end collisions with Agent 4 who is now driving in the same lane. It can be seen that Agent 2 is closing up to Agent 4, however, eventually, Agent 2 is driving at the same speed as Agent 4 to avoid a rear-end collision as shown in Fig. 4c.

Agent 1, who has lower priority than Agent 2, must wait for Agent 2 to leave the intersection. Although Agent 1 needs to decelerate to $1.2\,\mathrm{m/s}$, it stays as close as possible to Agent 2 to maximize its own speed as shown in Fig. 4b.

Lastly, Agent 3 is able to cross. With the lowest priority, it has to impose CA constraints on all other agents. Note that Agent 3 has to wait at the stopping line as it does not satisfy the minimum spatial preview constraint (17) — see Fig. 4b. After Agent 1 has cleared the intersection, Agent 3 is finally able to proceed. In essence, the distributed intersection coordination scheme satisfies all our requirements by being

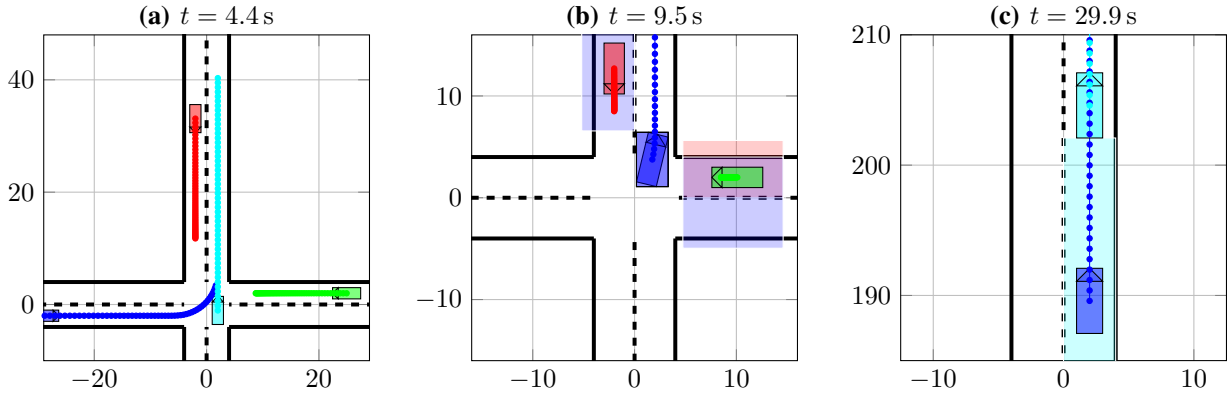


Fig. 4: Snapshots: (Left) Agent 4 (cyan) crosses first; Note that agents 2 (blue) and 3 (green) decide to wait; (Middle) Agent 1 (red) and Agent 3 (green) yield to Agent 2 (blue); (Right) Agent 2 (blue) avoids rear-end collisions with Agent 4 (cyan). The middle and right figures show the safety region of each agent i in the color of the conflicting Agent l.

capable of avoiding collisions in a complex realistic driving scenario and by ensuring smooth velocity and acceleration trajectories. In its current formulation, the control scheme avoids collisions with crossing agents in intersections and acts as a conventional adaptive cruise control system (ACC) outside intersection areas. Thus, we have proposed a generalized formulation of the collision avoidance problem which will serve as a basis for future extensions.

To evaluate the associated computational complexity, in Fig. 5 we show the computation times for every agent. Evidently, the entire scheme is real time capable having a maximum computation time of $44.1\,\mathrm{ms}$, while the sampling time is $100\,\mathrm{ms}$. An increased computational demand can be recognized when constraints are active for almost the entire horizon. This can be observed for Agent 2 during rear-end collision avoidance ($t \geq 20\,\mathrm{s}$) and for Agent 3 before entering the intersection ($5\,\mathrm{s} \leq t \leq 14\,\mathrm{s}$).

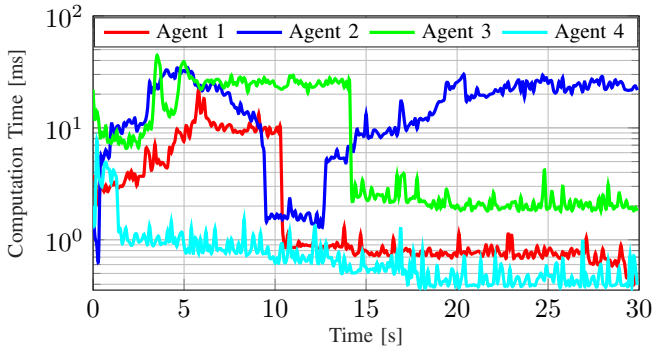


Fig. 5: Computation times for every agent for the entire simulation run.

VI. Conclusion and Future Work

For the coordination of agents, we have proposed a distributed MPC scheme which relies on V2V communication. Every agent determines its local control actions by efficiently solving a nonconvex nonlinear MPC problem using PANOC. To obtain a more general problem formulation, CA constraints are stated in a global Cartesian frame using B-spline functions. The OCP is augmented with conditional constraints which allows us to encode rules. Simulation results demonstrate the efficiency in terms of control performance and computational complexity.

In the future, we envision to further extend the control scheme to multi-lane/multi-intersection scenarios and to investigate how to deal with dynamic priority negotiation. Finally, we intend to conduct experimental tests.

References

[1] L. Hobert, A. Festag, I. Llatser, L. Altomare, F. Visintainer, and A. Kovacs, "Enh. of V2X comm. in support of cooperative autonomous driving," *IEEE Comm. Magazine*, vol. 53, no. 12, pp. 64–70, 2015.

[2] G. R. Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative Receding Horizon Conflict Resolution at Traffic Intersections," in *IEEE Conf. on Decision and Control*, 2014, pp. 2932–2937.

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

[4] J. Shi, Y. Zheng, Y. Jiang, M. Zanon, R. Hult, and B. Houska, "Distributed control algorithm for vehicle coordination at traffic intersections," in *European Control Conference*, 2018, pp. 1166–1171.

[5] M. Kneissl, A. Molin, H. Esen, and S. Hirche, "A Feasible MPC-Based Negotiation Algorithm for Automated Intersection Crossing," in *European Control Conference*, June 2018, pp. 1282–1288.

[6] F. Molinari and J. Raisch, "Automation Of Road Intersections Using Consensus-based Auction Algorithms," in *American Control Conference*, 2018, pp. 5994–6001.

[7] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed Model Predictive Control for Intersection Automation Using a Parallelized Optimization Approach," in *IFAC World Congress*, vol. 50, no. 1, 2017, pp. 5940 – 5946.

[8] A. Sathya, P. Sopasakis, R. V. Parys, A. Themelis, G. Pipeleers, and P. Patrinos, "Embedded nonlinear model predictive control for obstacle avoidance using PANOC," in *ECC*, 2018, pp. 1523–1528.

[9] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, "A simple and efficient algorithm for nonlinear model predictive control," in *Conference on Decision and Control*, 2017, pp. 1939–1944.

[10] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 613–626, 2018.

[11] C. de Boor, *A Practical Guide to Splines*. Springer, 1978, vol. 1.

[12] R. Rajamani, *Vehicle Dynamics and Control*. Springer, 2012, vol. 2.

[13] K. D. Kim and P. R. Kumar, "An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Trans. on Autom. Control*, vol. 59, pp. 3341–3356, 2014.

[14] B. Hermans, P. Patrinos, and G. Pipeleers, "A penalty method based approach for autonomous navigation using nonlinear model predictive control," in *IFAC World Congress*, no. 20, 2018, pp. 234–240.

[15] P. Patrinos and A. Bemporad, "Proximal Newton methods for convex composite optimization," in *IEEE 52nd Annual Conference on Decision and Control*, 2013, pp. 2358–2363.

[16] L. Stella, A. Themelis, and P. Patrinos, "Forward-backward quasi-Newton methods for nonsmooth optimization problems," *Computational Optimization & Applications*, vol. 67, no. 3, pp. 443–487, 2017.

[17] A. Themelis, L. Stella, and P. Patrinos, "Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms," *SIAM Journal on Optimization*, vol. 28, no. 3, pp. 2274–2303, 2018.

[18] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. Springer, 2006.