2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

A Feasible MPC-Based Negotiation Algorithm for Automated Intersection Crossing*

Maximilian Kneissl[1,2], Adam Molin[1], Hasan Esen[1], and Sandra Hirche[2]

Abstract— **We propose an intersection crossing algorithm for autonomous vehicles with vehicle to infrastructure (V2I) communication capability. All vehicles attempting to cross the intersection share their expected time of entering a critical zone based on decentralized model predictive control (MPC) results. These time suggestions are collected at a central intersection management (IM) unit, which is responsible for coordinating the vehicles. A time-based negotiation process between vehicles and IM is conducted to find a safe solution. An advantage of the approach is that model-based vehicle data is kept private, while the computational burden of the intersection coordination is distributed between the central IM and the vehicles. We prove the existence of a feasible solution and illustrate the introduced negotiation algorithm by simulation of an intersection crossing scenario with disturbances. The results show that vehicles remain in a safe distance without sharing private data.**

I. INTRODUCTION

A recent trend in the automotive industry is the development of autonomous vehicles. Commonly, these vehicles include the capability of communication with their environment to share information with other traffic participants and communicate with data clouds. Such applications are referred as vehicle to vehicle (V2V) and vehicle to infrastructure (V2I) technologies. A broad research effort has been spent on regulating traffic from a global perspective and it has been shown that those technologies have advantages compared to current traffic management systems [1]. Intersections play an important role and are often the bottleneck within urban traffic control. In general, vehicles approaching an intersection share information with either an intersection system (V2I) or with other vehicles (V2V). As a result, they receive or compute a feedback signal which describes the desired behavior and leads to a safe and smooth crossing. The use of traditional traffic lights and signs becomes unnecessary.

The goal of an intelligent guiding intersection is the resolution of conflicts among crossing vehicles, where conflicts are situations leading to an unsafe behavior of participants. Surveys on general motion coordination for distributed structures are provided in [2] and [3], while [4] treats the specific case of intersection management.

Significant work has been proposed to solve the scheduling problem for intersection crossing in order to define an appropriate crossing schedule for the vehicles [5]–[9].

However, above mentioned works make coarse assumptions on the local control of the vehicles what can result in the loss of feasibility of their solutions. A stronger focus on finding optimal or close-to-optimal solutions from a control perspective is proposed e.g. in [10]–[14]. These approaches assume either a given crossing order, or use a simplified scheduling method, such as first-come-first-serve.

Recent work also considers to find the system-wide optimum which contains the determination of the crossing order in relation with control optimization for the vehicles. Initially, [15] proposes a linear programming formulation incorporating a traffic model to solve the problem. Alternatively, mixed integer programs (MIP) are a natural way to combine the combinatorial scheduling nature with an optimal control problem of the vehicles [16], [17]. Last but not least, model predictive control (MPC) is commonly used to solve the intersection problem since it represents a powerful way to provide computationally tractable problems and result in close-to-optimal solutions [12], [14], [17]–[20].

In this paper, we present an algorithm for safe intersection crossing based on a decentralized V2I communication architecture. Thereby, we assume that local vehicle control units run model predictive controllers, connected to a centralized scheduling unit, referred to as intersection management (IM). The contribution of this work covers the consideration of privacy-related data exchange between the vehicles and the IM infrastructure, what is of interest in the field of vehicle coordination [21], [22]. Thereby, we use the term privacy to indicate that no vehicle information which enables to draw conclusions about local vehicle functions or behavior, such as model data or trajectory profiles, needs to be shared with third parties. By explicitly integrating a global timing coordination result as constraint in the local MPC formulation, the communication effort is kept low due to the little amount of information exchange. Additionally, the coupling between the coordination level and local control aims to find a close-to-optimal solution compared to a centralized approach. The global coordination is formulated as a convex quadratic optimization problem and thus scales well, while the local vehicle MPC problem can be formulated at arbitrary detail. The scheduling order is centrally defined based on predictions received from the local vehicle control units. Despite the low information exchange, we are able to

*This work has been conducted within the ENABLE-S3 project that has received funding from the ECSEL Joint Undertaking under grant agreement No 692455. This joint undertaking receives support from the European Union's HORIZON 2020 research and innovation programme and Austria, Denmark, Germany, Finland, Czech Republic, Italy, Spain, Portugal, Poland, Ireland, Belgium, France, Netherlands, United Kingdom, Slovakia, Norway

[1]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

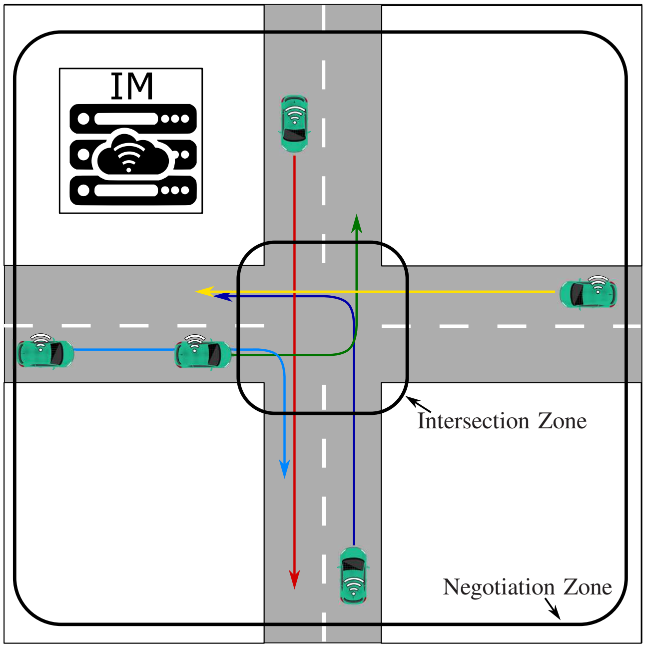

Fig. 1. Exemplary intersection scenario with an intersection management (IM) system and V2X capable vehicles.

provide a feasibility guarantee of the centralized scheduling proposition for the vehicle control problems.

The remainder of the paper is organized as follows. Section II introduces the vehicle control problems in Subsection II-A, the global IM optimization in Subsection II-B, and in Subsection II-C a simplified intersection representation. The following Section III describes the procedure of the proposed negotiation algorithm in Subsection III-B, after stating preliminary assumptions in Subsection III-A, and provides a feasibility guarantee of the solution in Subsection III-C. Next, Section IV illustrates the algorithm procedure numerically. Finally, Section V contains concluding remarks and an outlook for future work.

II. PROBLEM STATEMENT

Before formally setting up the problem, we provide an intuition on the solution approach. Figure 1 illustrates our system setup with an exemplary intersection crossing scenario. Each vehicle solves a local trajectory tracking control problem. At the point the vehicle approaches the intersection area, it shares its expected time of arrival at the intersection point with the IM, which receives data from all vehicles. If the IM detects a safety relevant conflict, it imposes updated reference times, specifying when the affected vehicles should arrive at the intersection. It is worth mentioning, that no model related information is exchanged but only time values. Thus, it is possible that the IM reference time is infeasible for vehicles. This is recognized and adjusted by the IM through a suggested feedback time from the vehicles regarding the reference. Conducting this negotiation iterations the deviation of the vehicle suggestions and the IM reference times converges towards zero assuming that negotiating vehicles are in a brake-safe distance to the intersection. While the vehicles are in the negotiation zone the procedure is repeated in an any-time fashion to cope for uncertainties in both the models and the environment.

Now, the components of the problem are formulated. First, we propose model-based in-vehicle (local) control subject to constraints connecting the local agents (vehicles) with the supervisory layer (IM), similar to [11], [12], [17]. Second, the optimization problem for the IM system is formulated. And finally, we explain the representation of the intersection itself.

A. Vehicles

We assume that each vehicle is equipped with a local MPC controller, which determines the vehicles' trajectories according to given paths.

Each vehicle $i \in \{1, ..., N\}$ uses a discrete-time private vehicle model for its local MPC problem. We describe these models by

$$x_i(k+1) = f_i\left(x_i(k), u_i(k)\right), \tag{1}$$

where $x_i(k) \in \mathbb{R}^{n_i}$ and $u_i(k) \in \mathbb{R}^{m_i}$ are the state and input vectors at discrete time instant k, respectively. An arbitrary state description can be assumed, however, it shall contain information on the vehicle's position and velocity.

The MPC problem is described by

$$\begin{aligned}
\underset{u_i}{\text{minimize}} \quad & \sum_{k=0}^{M-1} l_i\left(x_i(k), u_i(k)\right) + V_i(x_i(M)) && \text{(2a)}\\
\text{subject to} \quad & x_i(k+1) = f_i\left(x_i(k), u_i(k)\right), && \text{(2b)}\\
& x_i \in \mathbb{X}_i, \ \ u_i \in \mathbb{U}_i, && \text{(2c)}\\
& x_i(M) \in X_i \subset \mathbb{X}_i, && \text{(2d)}\\
& x(k_{ref,i}) = x_{ref,i}, && \text{(2e)}
\end{aligned}$$

with the prediction horizon M. Problem (2) optimizes the cost function $l_i\left(x_i(k), u_i(k)\right)$ and terminal cost $V_i(x_i(M))$ subject to the model dynamics (1), the sets $\mathbb{X}_i$ and $\mathbb{U}_i$ which constrain the states and input sequences x_i and u_i, respectively, as well as the terminal constraint (2d) with terminal set X_i. Furthermore, we introduce the time-based constraint (2e), where $k_{ref,i} \in \{1, ..., M\}$ is a reference time and $x_{ref,i} \in \mathbb{X}_i$ the corresponding reference state. These will be shared by the IM to coordinate the vehicles through the intersection area once the latter is reached by a vehicle's prediction horizon. Note that (2) without (2e) represents a standard MPC description which can be used for set point tracking control. By adding (2e) once the vehicle is in the vicinity of an intersection we make the problem suitable for the proposed intersection negotiation process.

According to [23], we know that there exists a minimum for the optimization (2a) - (2d) if

- M is finite,
- $f_i(\cdot)$, $l_i(\cdot)$, and $V_i(\cdot)$ are continuous,
- $\mathbb{U}_i$ is compact, and $\mathbb{X}_i$ as well as X_i are closed,
- the initial state $x_i(0) \in \mathbb{X}_i$.

Our feasibility analysis in Section III-C proofs the existence of a minimum when (2e) is added to the problem.

B. Intersection Management System

The IM computer has knowledge of the predicted arrival times at the intersection zone, $t_{sug,i}$, of all vehicles $i \in \{1, ..., N\}$. The crossing order is determined based on the sorted first time suggestions in the initialization phase of each negotiation round. It is worth to mention that the crossing order is determined in each control time step and therefore can change while vehicles are driving within the negotiation zone. The result is a order set O containing indexes $i \in \{1, ..., N\}$ inserted according the determined crossing order. Furthermore, its main responsibility is to ensure a specified safety time between crossing vehicles. We define the safety time between two consecutive vehicles i and $i+1$ as $t_{s,i,i+1}$.

Next, we propose a quadratic optimization problem for finding optimal intersection crossing times, by considering the safety times:

$$\underset{t_{ref}}{\text{minimize}} \quad (t_{ref} - t_{sug})^{\mathsf{T}} Q(\kappa) (t_{ref} - t_{sug}) + c^{\mathsf{T}} (t_{ref}) \tag{3a}$$

$$\begin{aligned} \text{subject to} \quad & t_{ref,1} + t_{s,1,2} \leq t_{ref,2} \qquad (3b) \\ & t_{ref,2} + t_{s,2,3} \leq t_{ref,3} \\ & \vdots \\ & t_{ref,N-1} + t_{s,N-1,N} \leq t_{ref,N} \end{aligned}$$

Thereby, $t_{ref} = (t_{ref,1}, ..., t_{ref,N})^{\mathsf{T}}$ is a vector in $\mathbb{R}_+^N$ which contains time reference values indicating when the vehicles are supposed to be at the intersection zone. On the contrary, $t_{sug} = (t_{sug,1}, ..., t_{sug,N})^{\mathsf{T}}$ is a vector in $\mathbb{R}_+^N$ with the stacked time predictions of the vehicles. These vectors, t_{ref} and t_{sug} are sorted according the index set O. The diagonal matrix $Q(\kappa) \in \mathbb{R}^{N \times N}$ weights the deviation of reference and suggested time values for each vehicle, thus $Q(\kappa)$ is of the form $diag\,(q_1(\kappa), ..., q_N(\kappa))$. Each $q_i(\kappa) \in \mathbb{R}$ is adjusted after a negotiation time step κ by

$$q_i(\kappa + 1) = q_i(\kappa) + \epsilon \mid t_{ref,i} - t_{sug,i} \mid, \tag{4}$$

where ϵ is an update constant. Similar, the constant vector $c = (c_1, ..., c_N)$ weights the time reference values. The constraints (3b) ensure the safety times between crossing vehicles. Note that also t_{sug} and t_{ref} depend on κ. For reasons of presentations, however, we skip this dependency and refer to it only when required for explanation. To conclude, the optimization (3) considers three intuitive tasks. The quadratic term orientates the reference result towards the vehicles' suggestions, while the linear term minimizes the reference time values itself to obtain a high vehicle throughput. In addition, the constraints ensure a safety distance between the vehicles.

C. Representation of Intersection

We simplify the intersection scenario shown in Figure 1 to an intersection problem of two vehicles. This is illustrated in Figure 2. Each vehicle follows a pre-defined path. The intersection point I represents the position where a vehicle enters the intersection zone, as illustrated in Figure 1.

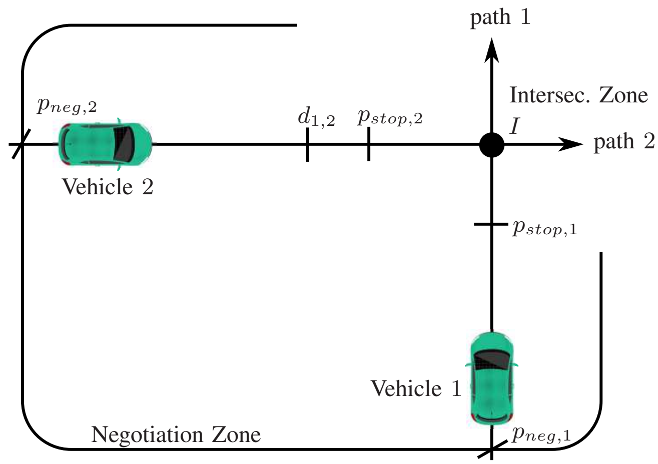

Fig. 2. Simplified illustration of the intersection scenario for the example $i \in 1, 2$. Presented are the paths of the vehicles, the intersection zone I reduced to a single point, the start of the negotiation zone $p_{neg,i}$, $d_{i,i+1}$, the minimum distance to I when the predecessor vehicle is at I, and the minimum stopping distance $p_{stop,i}$.

Furthermore, the negotiation zone introduced in the same figure is mapped on the vehicles' paths indicated by the points $p_{neg,i}$. The geometric configuration of the intersection and the chosen path of a vehicle as well as its dimensions and velocity influence the safety interval $t_{s,i,i+1}$ of two consecutive vehicles, introduced in Subsection II-B. In this paper we assume a constant approximation of this value. To ensure safe intersection crossing for general cases this shall be investigated in more detail in future work. The distance, $d_{i,i+1}$, describes the closest allowed position to I of vehicle $i+1$ at the time vehicle i approaches I if $t_{s,i,i+1}$ is fulfilled. The minimal stopping distance to I for vehicle i is illustrated with $p_{stop,i}$ and marks the position where a safe stop before I would still be possible according to the states and dynamics of the vehicle.

III. NEGOTIATION ALGORITHM

This section describes in Subsection III-B the procedure of the V2I-based negotiation algorithm for safe intersection crossing, after presenting some preliminary assumptions in Subsection III-A. A formal representation is provided in Algorithm 1, which is described in the following. Moreover, Figure 3 graphically illustrates the signal flow and optimization problems in both the IM system and the vehicles. Thereafter, in Subsection III-C, we investigate the algorithm convergence and provide a feasibility guarantee for the local MPC problems.

A. Preliminaries

We assume that vehicles follow their pre-defined paths and compute their local trajectories based on a MPC control law. If a vehicle enters the negotiation zone during $(k-1, k]$, it participates in the negotiation process starting at time k. Let us define the following state sets.

Algorithm 1 Intersection Crossing Negotiation

1: clock $\leftarrow k$
2: **procedure** INITIALIZATION
3: ***Vehicles:***
4: **for** $i \in \{1, ..., N\}$ **do** (in parallel)
5: compute MPC (2), determine $t_{sug,i}$
6: send $t_{sug,i} \rightarrow$ IM
7: ***IM:***
8: determine vehicle crossing order for current negotiation round

9: **procedure** NEGOTIATION
10: $\kappa = 1$ $\triangleright$ negotiation loop counter
11: $Q(\kappa) \leftarrow Q_{init}$
12: **repeat**
13: ***IM:***
14: compute t_{ref} considering t_s and t_{sug} (3)
15: broadcast $t_{ref,i} \rightarrow$ vehicles $i \in \{1, ..., N\}$

16: ***Vehicles:***
17: **for** $i \in \{1, ..., N\}$ **do** (in parallel)
18: compute MPC (2)
19: send $t_{sug,i} \rightarrow$ IM

20: ***IM:***
21: update cost $\forall i \in \{1, ..., N\}$:
22: $q_i(\kappa+1) \leftarrow q_i(\kappa) + \epsilon \mid t_{ref,i} - t_{sug,i} \mid$
23: **if** $\mid t_{ref,i} - t_{sug,i} \mid < \delta$ **then**
24: **return** success

25: **until** $\kappa = \kappa_{max}$

26: ***Vehicles:***
27: apply control
28: clock $\leftarrow k + 1$

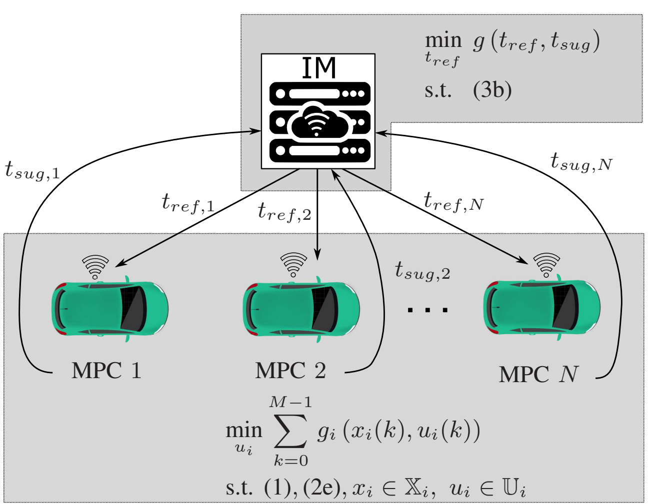

Fig. 3. Signal exchange and optimization problems during negotiation process.

Definition 1: [23] $X_{k \rightarrow f,i}$ is the set of states ($X_{k \rightarrow f,i} \subseteq \mathbb{X}_i$) which are steerable by a feasible $u_i \in \mathbb{U}_i$ to $X_{f,i}$ in k or less steps.

Additionally, we define a minimum-step set:

Definition 2: $X_{k_{min} \rightarrow f,i}$ is the set of states with a minimum possible k for which a feasible $u_i \in \mathbb{U}_i$ exists that steers the system to $X_{f,i}$ in exactly k_{min} steps and $X_{k_{min} \rightarrow f,i}$ is not empty.

Finally, a brake-safe set is defined.

Definition 3: $X_{stop,i} = \{x_i | v_i = 0, p_i < I\}$ is a terminal set wherein the velocity v_i and position p_i are elements of the state vector x_i, and I is the reference position of entering the intersection zone. Furthermore, $X_{inter,i} = \{x_i | p_i = I\}$.

For all vehicles $i \in \{1, ..., N\}$ in the negotiation process, the following assumption holds:

Assumption 1: If a vehicle participates in the intersection negotiation process, for its current state it holds $x_i \in X_{k \rightarrow stop,i}$ and $X_{k \rightarrow stop,i}$ is not empty.

Based on Assumption (1), we further assume:

Assumption 2: $X_{k \rightarrow stop,i} \subset X_{k_{min} \rightarrow inter,i}$, that means for all vehicles which have the ability to stop at $p_i < I$ there exists also a feasible control sequence u_i which steers the system to $p_i = I$ in k_{min} steps.

B. Procedure Description

The negotiation rounds are conducted within the inter-sampling period while vehicles are driving. Thus, an achieved negotiation result during $(k-1, k]$ affects the control input $u_i(k)$, where the clock k is assumed to be synchronized system-wide.

At the beginning of each control time step k for $k \in \mathbb{N}_0$, every vehicle conducts a MPC computation according to (2) and shares the resulting predicted time, $t_{sug,i}$, at I (compare Figure 2) with the IM system (lines 2 - 8 of Algorithm 1). Based on the collected time suggestions of all participating vehicles, the IM computer determines the crossing order of the vehicles through the intersection zone. If a vehicle participates the first time, since it just entered the negotiation zone, $t_{sug,i}$ is computed based on (2) without considering (2e). Next, the cost matrix is initialized with Q_{init}, then the negotiation process starts by computing the reference times t_{ref} according to problem (3), where the suggestions of the vehicles t_{sug} and the safety time t_s are considered. The resulting $t_{ref,i}$s are broadcast to the respective vehicles (lines 13 - 15). Consequently, the vehicles try to realize the global time suggestions in their local problem by applying the hard constraint (2e) within (2). By using methods such as exact penalty functions the possibility of an infeasible constraint (2e) can be considered [24]. This leads to a new time suggestion $t_{sug,i}$ from the vehicles, where in the feasible case it equals the reference $t_{ref,i}$ and else the resulting $t_{sug,i}$ presents an attempt to achieve $t_{ref,i}$ as close as possible (lines 16 - 19). After receiving the new vehicle suggestions the IM updates the cost matrix according to line 22 of Algorithm 1. If $t_{sug,i}$ and $t_{ref,i}$ differ, the cost $q_i(\kappa+1)$ increases (lines 20 - 22). We call the negotiation terminated when the deviation of $\mid t_{ref,i} - t_{sug,i} \mid$ is bounded by δ

(lines 23 - 24). Otherwise, it stops after a maximum number of iterations κ_{max} (line 25). Finally, the receding horizon-fashioned control can be applied for time step k and time proceeds to $k+1$ (lines 26 - 28). We can summarize the order of time values during the negotiation procedure as *Initialization* $\rightarrow ... \rightarrow t_{ref}(\kappa) \rightarrow t_{sug}(\kappa) \rightarrow Q(\kappa+1) \rightarrow t_{ref}(\kappa+1) \rightarrow t_{sug}(\kappa+1) \rightarrow ...$.

Remark 1: In order to cope with the conversion between the discrete domain in problem (2) to continuous variables in (3) we use discrete values in the continuous problems, i.e. $t_{sug} = k_{sug}$. Contrary, we apply the rounded values (*floor* function) of t_{ref} as k_{ref} in the constraint (2e) and in the cost update (4).

C. Termination and Feasibility

Before showing that the algorithm terminates, we make a statement for the time suggestions from the local control units.

Proposition 1: For the set of feasible times $T_{f,i} = \{t|p_i = I\}$ it holds $T_{f,i} \in [t_{min,i}; \infty)$.

Proof: From Assumptions 1 and 2 we know that there exists a $t_{min,i} = k_{min,i}$. Furthermore, as Assumption 1 holds, the set of states $X_{stop,i} = \{x_i | v_i = 0, p_i < I\}$ is a feasible set which leads to $\{t|p = I\} \rightarrow \infty$. ■

Theorem 1: Given Assumptions 1 and 2, the intersection negotiation algorithm terminates and its solution is feasible for the local optimization problems (2), such that $t_{ref,i} \in T_{f,i}$ for all $i \in \{1, ..., N\}$.

Proof: We show the asymptotic convergence of Algorithm 1, where for brevity of notation δ (line 23) is neglected. Thus, $t_{ref,i} = t_{sug,i}, i \in \{1, ..., N\}$ is used to proof Theorem 1 and ensure feasibility of problem (2).

From constraint (2e) we know $t_{sug,i}(\kappa) = t_{ref,i}(\kappa)$ if $t_{ref,i}(\kappa) \in T_{f,i}$, since we assume that the stated conditions in Subsection II-A are met. Furthermore, since (4) is a non-decreasing function $q_i(\kappa)$ increases with growing κ if $t_{ref,i} \neq t_{sug,i}$. Knowing that $c^\intercal$ in (3a) is constant, we assume that $q_i \gg c_i$, for $i \in \{1, ..., N\}$, holds and thus the cost function of (3) reduces to $(t_{ref} - t_{sug})^\intercal Q(\kappa)(t_{ref} - t_{sug}) + c^\intercal (t_{ref}) \approx (t_{ref} - t_{sug})^\intercal Q(\kappa)(t_{ref} - t_{sug})$.

Now, consider the case $t_{ref,i}(\kappa) \geq t_{sug,i}(\kappa - 1)$. We have $t_{min,i} \leq t_{sug,i}(\kappa - 1) \leq t_{ref,i}(\kappa)$ and consequently $t_{ref,i}(\kappa) \in T_{f,i}$. This results in $t_{sug,i}(\kappa) = t_{ref,i}(\kappa)$ and the cost value, $q_i(\kappa+1) = q_i(\kappa)$, remains constant.

Next, we investigate all $t_{ref,i}(\kappa) < t_{sug,i}(\kappa - 1)$. For the consecutive time suggestion it holds $t_{sug,i}(\kappa) \leq t_{sug,i}(\kappa - 1)$ and $t_{ref,i}(\kappa) \leq t_{sug,i}(\kappa)$, with equality if $t_{ref,i}(\kappa) \in T_{f,i}$ what results again in an constant cost update. However, if strict inequality holds $(t_{ref,i}(\kappa) < t_{sug,i}(\kappa))$ the cost increases, $q_i(\kappa+1) > q_i(\kappa)$. We summarize that at negotiation step κ the time suggestion, $t_{sug,i}(\kappa)$ has either converged to $t_{ref,i}(\kappa)$, or it holds $t_{sug,i}(\kappa) > t_{ref,i}(\kappa)$ with $q_i(\kappa+1) > q_i(\kappa)$. Because of this cost increase of all $t_{ref,i}(\kappa+1) \notin T_{f,i}$ and constant costs for all $t_{ref,i}(\kappa+1) \in T_{f,i}$, we can draw the conclusion $t_{ref,i}(\kappa+1) > t_{ref,i}(\kappa)$ if $t_{ref,i}(\kappa) \notin T_{f,i}$. Note, that the increase of the reference-time $t_{ref,i}(\kappa+1) > t_{ref,i}(\kappa)$ does not affect the convergence of any consecutive vehicles $j > i$. If due to an active constraint (3b) such a increase causes $t_{ref,j}(\kappa+1) > t_{ref,j}(\kappa)$ and $t_{ref,j}(\kappa) \in T_{f,j}$, then $t_{ref,j}(\kappa+1)$ remains in $T_{f,j}$. For all $t_{ref,j}(\kappa) \notin T_{f,j}$ the previous reasoning for i holds. Therefore, $| t_{ref,i}(\kappa+1) - t_{sug,i}(\kappa+1) | = 0$ if $t_{ref,i}(\kappa+1) \in T_{f,i}$. Else for $t_{ref,i}(\kappa+1) \notin T_{f,i}$ we find asymptotic convergence $| t_{ref,i}(\kappa+1) - t_{sug,i}(\kappa+1) | < | t_{ref,i}(\kappa) - t_{sug,i}(\kappa) |$ which will result in $t_{ref,i}(\kappa+1) \in T_{f,i}$.

■

IV. NUMERICAL RESULTS

For the simulation setup, we assume perfect path tracking capability of the vehicles, what enables us to model the vehicles driving on a straight line. We further model the car fleet, consisting of homogeneous vehicles, with the discrete and linear time invariant point mass model

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k), \tag{5}$$

where the system and input matrices are defined as

$$A_i = \begin{pmatrix} 1 & T_s \\ 0 & 1 \end{pmatrix} \text{ and } B_i = \begin{pmatrix} T_s^2/2 \\ T_s \end{pmatrix}, \tag{6}$$

respectively. Therein, $T_s = 0.1s$ is the control sampling time and $x_i = (p_i, v_i)^\intercal \in \mathbb{R}^2$ contains the position state p_i and velocity state v_i of vehicle i. The input $u_i \in \mathbb{R}$ in (5) is the vehicle acceleration. We specify the MPC problem (2) as

$$\begin{aligned} \underset{u_i}{\text{minimize}} \quad & \sum_{k=0}^{M-1} (v_i(k) - v_{ref,i}(k))^2 && \text{(7a)} \\ \text{subject to} \quad & x_i(k+1) = A_i x_i(k) + B_i u_i(k), && \text{(7b)} \\ & x_i \in \mathbb{X}_i, \ u_i \in \mathbb{U}_i && \text{(7c)} \\ & p_i(k_{ref,i}) = p_{ref,i}, && \text{(7d)} \end{aligned}$$

where $\mathbb{X}_i$ is the set $\{(p_i, v_i) \in \mathbb{R}^2 \mid p_i \in path_i, v_i \in [0, 15\ m/s]\}$ and $\mathbb{U}_i = \{u_i \in \mathbb{R} \mid u_i \in [-4\ m/s^2, 4\ m/s^2]\}$. We use a horizon of $M = 100$ and $v_{ref,i}(k) = 8.3 m/s$ is used to integrate an environmental velocity reference. The conditions for the existence of a minimum in Subsection II-A are met and we have a solution to the optimization problem. Note that we simplified the MPC formulation by neglecting the terminal constraint and cost. Furthermore, in (7d), we reduce the time constraint (2e) by only considering the position component with a reference position $p_{ref,i}$ at a reference time $k_{ref,i}$. Next, we define the vector $t_s = (0.5\ s, ..., 0.5\ s)^\intercal \in \mathbb{R}^{N-1}$ with constant safety time between vehicles. During simulation the weights c of the reference times in (3) are kept constant at value one, while $Q(\kappa)$ is dynamically updated according to Algorithm 1.

Figure 4 illustrates the negotiation process introduced in Algorithm 1 for the worst case scenario where all vehicles request to enter the intersection zone at the same time. In this case 4 vehicles are in the negotiation zone and suggest an arrival time of $t_{sug} = 1.1s$, which is the initial negotiation iteration 0 (lines 2 - 8 of Algorithm 1). Subsequently, the

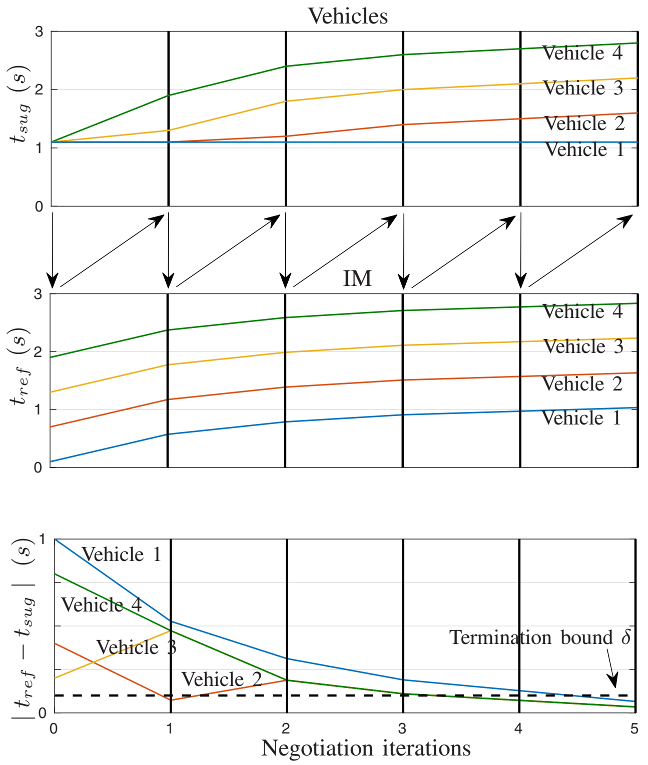

Fig. 4. Negotiation process for the time of entering the intersection zone. Top plot: First Vehicle suggestions (t_{sug}) at iteration step 0 and adjustments during negotiation (iterations $1-5$) for 4 vehicles. Middle plot: Reference time resulting form IM optimization problem. Bottom plot: Deviation between reference and suggestions and a bound representing the termination condition.

IM splits these suggestions according to the safety distance t_s. During negotiation, the increasing costs $q_1(\kappa)$ orients $t_{ref,1}$ towards $t_{sug,1}$ (line 22 in Algorithm 1). Once all deviation values are below a certain bound (convergence bound in bottom plot of Figure 4), the algorithm is converged and the latest suggested time values can be applied to the local control of the vehicles. After explaining the negotiation process, we show the simulation of a complete intersection crossing scenario with any-time replanning in the disturbed case. Therefore, we model vehicles approaching to an intersection from tree different lanes *lane* 1, *lane* 2, and *lane* 3. Vehicles are added randomly to the entrance of the intersection negotiation zone at position $-15m$. After the vehicle negotiation converged, as described above, the respective control input is applied to the systems. A supervisory function implemented in the IM computer checks the divergence of t_{sug} and t_{ref} during every control time step. If necessary, a new negotiation round is started. We model this necessity by applying a random disturbance (marked with $\lightning$ in Figure 5) in the speed state v_i of vehicles v_i. For example, such a disturbance appears for the second vehicle v_2 at time $t = 1.2s$. This leads to a new negotiation process and consequently to a different crossing order of the vehicles as originally planned.

TABLE I

TIME SUGGESTIONS FROM VEHICLES BEFORE AND AFTER APPEARANCE OF DISTURBANCE

vehicle	disturbance at $t = 1.2s$ t_{sug} before	t_{sug} after
v_1	$2.1s$	$2.1s$
v_2	$2.6s$	$3.7s$
v_3	$3.2s$	$2.6s$
v_4	$3.8s$	$3.1s$
v_5	$4.3s$	$4.3s$

Table I lists the suggested times $t_{sug,i}$ for vehicles v_1 - v_5 after the converged negotiation rounds. The middle column shows the suggested arrival times at the entrance of the intersection zone before the disturbance emerges, while the right column illustrates the changed order as vehicle v_2 was disturbed.

At the intersection zone (position 0 in Figure 5) all vehicles have the specified safety distance $t_s = 0.5s$. The reference velocity for crossing the intersection is $v_{ref} = 8.3m/s$, marked by the dashed line in Figure 5. If necessary, the vehicles differ from v_{ref} to meet the imposed time constraints from the IM. This results in the speed profiles illustrated in the lower plot of Figure 5.

V. CONCLUSIONS

We introduced a negotiation approach for safe intersection crossing of automated vehicles. Local model-based control predictions from vehicles are shared with a central IM computer, which detects safety critical scenarios. Furthermore, it advises the vehicles to adjust their control signals to achieve an appropriate arrival time in case of unsafe scenarios. If the reference time constraint from the IM is not feasible for the vehicles, they suggest a new value considering the reference as a soft constraint in their local MPC problem. Therefore, the resulting negotiation procedure uses no private crucial vehicle data to solve the decentralized optimization problem.

Subject to future work will be the consideration of the influence of coupled vehicle dynamics due to e.g. platooning scenarios. Furthermore, methods for defining scenario dependent safety times will be investigated.

REFERENCES

[1] M. Li, K. Boriboonsomsin, G. Wu, W.-B. Zhang, and M. Barth, "Traffic energy and emission reductions at signalized intersections: a study of the benefits of advanced driver information," *International Journal of Intelligent Transportation Systems Research*, vol. 7, no. 1, pp. 49–58, 2009.

[2] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, 2016.

[3] S. Martinez, J. Cortes, and F. Bullo, "Motion coordination with distributed information," *IEEE Control Systems*, vol. 27, no. 4, pp. 75–88, 2007.

[4] L. Chen and C. Englund, "Cooperative intersection management: a survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 570–586, 2016.

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

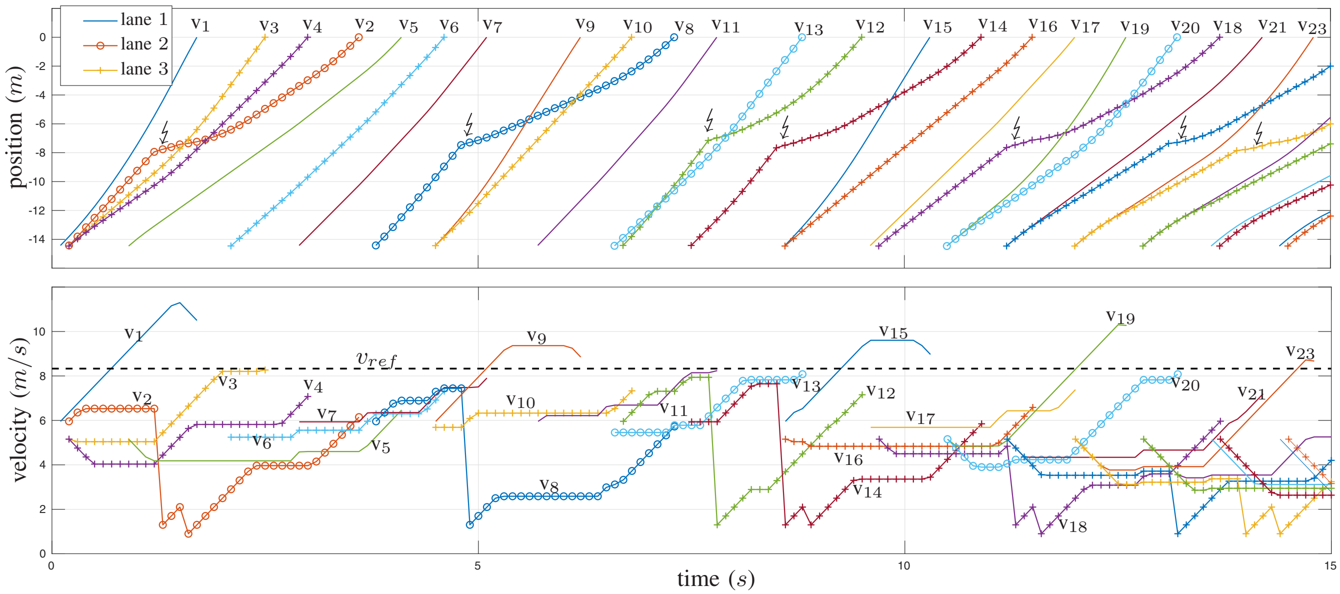

Fig. 5. Simulation of a three lane intersection scenario with modeled disturbances (↯). Solid, circled, and crossed lines represent the states of vehicles v_i on lane 1, lane 2, and lane 3, respectively.

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

[7] K. Yang, S. I. Guler, and M. Menendez, "Isolated intersection control for various levels of vehicle technology: Conventional, connected, and automated vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 109–129, 2016.

[8] E. R. Müller, R. C. Carlson, and W. K. Junior, "Intersection control for automated vehicles with milp," *IFAC-PapersOnLine*, vol. 49, no. 3, pp. 37–42, 2016.

[9] A. A. Hassan and H. A. Rakha, "A fully-distributed heuristic algorithm for control of autonomous vehicle movements at isolated intersections," *International Journal of Transportation Science and Technology*, vol. 3, no. 4, pp. 297–309, 2014.

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

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

[12] M. Zanon, S. Gros, P. Falcone, and H. Wymeersch, "An asynchronous algorithm for optimal vehicle coordination at traffic intersections," in *20th IFAC World Congress*, 2017.

[13] A. I. M. Medina, N. Van De Wouw, and H. Nijmeijer, "Automation of a t-intersection using virtual platoons of cooperative autonomous vehicles," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*, pp. 1696–1701, IEEE, 2015.

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

[15] F. Zhu and S. V. Ukkusuri, "A linear programming formulation for autonomous intersection control within a dynamic traffic assignment and connected vehicle environment," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 363–378, 2015.

[16] F. Tedesco, D. M. Raimondo, A. Casavola, and J. Lygeros, "Distributed collision avoidance for interacting vehicles: a command governor approach," *IFAC Proceedings Volumes*, vol. 43, no. 19, pp. 293–298, 2010.

[17] R. Hult, G. R. Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *American Control Conference (ACC), 2015*, pp. 763–768, IEEE, 2015.

[18] G. R. De Campos, P. Falcone, and J. Sjoberg, "Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing," *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*, pp. 1456–1461, 2013.

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

[20] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach,"

[21] J.-P. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security & Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[22] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pp. 1–8, IEEE, 2014.

[23] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, 2000.

[24] J. Nocedal and S. J. Wright, "Numerical optimization 2nd," 2006.