Optimal Coordination of Automated Vehicles at Intersections: Theory and Experiments

Robert Hult, Mario Zanon, Sebastien Gros, and Paolo Falcone

***Abstract*— In this paper, we present a bilevel, model predictive controller for coordination of automated vehicles at intersections. The bilevel controller consists of a coordination level, where intersection occupancy timeslots are allocated, and a vehicle level, where the control commands for the vehicles are computed. We establish persistent feasibility and stability of the bilevel controller under some mild assumptions and derive conditions under which closed-loop collision avoidance can be ensured with bounded position uncertainty. We thereafter detail an implementation of the coordination controller on a three-vehicle test bed, where the intersection-level optimization problem is solved using a distributed Sequential Quadratic Programing method. We present and discuss results from an extensive experimental campaign, where the proposed controller was validated. The experimental results indicate the practical applicability of the bilevel controller and show that safety can be ensured for large positioning uncertainties.**

***Index Terms*— Distributed optimization, intersection coordination, model predictive control (MPC), networked mobile systems.**

I. Introduction

THE recent years have seen a rapid development in the field of automated driving, and such technologies are expected to penetrate the consumer market in the industrialized world in the upcoming decades. With the introduction of vehicle-to-vehicle (V2V) communication, *cooperative automated vehicles* (CAVs) offer new possibilities to increase both safety of passengers and efficiency of the traffic system. In particular, communication-based, cooperative strategies can augment the capabilities of autonomous vehicles and allow them to jointly resolve difficult and safety critical-traffic situations without relying on current traffic rules. Locations where roads cross or merge form a particularly problematic subset of the traffic system. It has, for instance, been reported that 21% of the traffic fatalities and 43% of the crashes occur in and around intersections in the European Union [1], and similar numbers have been found in USA. The risks associated with traffic intersections require a strict regulation of the involved vehicles, which is currently performed using traffic lights, signs, and right-of-way rules. In part due to the regulation, intersections tend to form bottlenecks in the traffic system and are commonly one of the causes of congested traffic. As a consequence, energy is wasted (e.g., through deceleration/acceleration and idling [2]), and improvements to traffic flow typically require an expansion of the infrastructure. The introduction of CAVs enables a potential remedy; rather than using the current regulations, the CAVs could be controlled using automated coordination algorithms, which could guarantee collision avoidance, increase energy efficiency, and optimize the traffic flow. The CAVs are then envisioned to be able to travel through intersections at high speeds in tightly packed, interlocking streams. This requires the coordination algorithms to act on the individual vehicles rather than controlling the traffic flows through, e.g., adaptive traffic signal timings.

The problem of coordinating CAVs at traffic intersections poses a number of challenges [3], [4]. For instance, a coordination algorithm must rely on potentially lossy wireless communication [5], utilize imperfect measurements (in particular for positioning), and handle various forms of perturbations at the vehicle level. The coordination algorithm must therefore be executed in closed loop, and the control of the vehicles continuously be adjusted to incorporate up-to-date information about the vehicle states and the surroundings.

In this paper, we utilize an optimal control (OC) formulation of the coordination problem at intersections, first presented in [6], and propose its application to closed-loop, receding horizon control. In particular, we propose a bilevel controller consisting of: 1) an intersection-level control loop that allocates and updates optimal and collision-free timeslots during which each vehicle is allowed to occupy the intersection and 2) lower level control loops that provide optimal actuation commands for all vehicles given the allocated timeslots. We provide a study of the nominal closed-loop system and present conditions enforcing persistent feasibility and stability. We also propose a modification of the controller, which allows it to retain persistent feasibility in the presence of bounded perturbations, and present conditions ensuring robust collision avoidance. Furthermore, we describe an implementation of the proposed controller on an experimental test bed consisting of three communicating, automated vehicles. The implementation

Manuscript received November 14, 2017; revised July 8, 2018; accepted September 4, 2018. Manuscript received in final form September 17, 2018. This work was supported in part by The Swedish Research Council under Grant 2012-4038 and in part by Vinnova through the Copplar Project under GRANT 2015-04849 and through the AstaZero Program under Grant 2015-03075. Recommended by Associate Editor Y. Wang. *(Corresponding author: Robert Hult.)*

R. Hult, S. Gros, and P. Falcone are with the Department of Electrical Engineering, Chalmers University of Technology, 412 96 Gothenburg, Sweden (e-mail: robert.hult@chalmers.se; grosse@chalmers.se; paolo.falcone@chalmers.se).

M. Zanon is with the IMT School for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: mario.zanon@imtlucca.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2018.2871397

uses the Sequential Quadratic Programing (SQP) algorithm proposed in [7] and [8], which is solved in a semidistributed fashion using V2V communication. We thereafter present results from an experimental campaign, where we demonstrate the proposed controller's behavior and discuss the consistency of its performance as well as its ability to reject perturbations.

A. Related Work

The last decade has seen a number of contributions to the study of coordination algorithms in the context of automated vehicles, in most cases focusing on intersections. In the early work of Dresner and Stone [9], a system was presented, where the oncoming vehicles request a reservation of timeslots, within which they are allowed to pass the intersection, to a centralized intersection manager (IM). The IM thereafter performs a forward simulation of the relevant vehicles and rejects the reservation only if a collision is predicted to occur. In this case, the vehicle slows down and sends new reservation requests. A different and provably safe timeslot-reservation-based IM was proposed in [10], where the timeslot reservations are improved incrementally as the system evolves with the purpose of minimizing travel time. In [11], another method is presented, where an IM allocates timeslots to the vehicles by minimizing the total timeslot overlap. In particular, the IM uses constant acceleration predictions to determine when the vehicles would occupy the intersection and attempts to find the accelerations that give the smallest timeslot overlap. In [12], an algorithm is proposed, where a central entity first decides the crossing order and timeslots of the vehicles using results from polling system theory and, thereafter, computes their respective control commands. Gregoire and Frazzoli [13] propose a similar scheme, where the central component schedules occupancy timeslots within which the vehicles thereafter are controlled to pass the intersection at maximum speed. It is shown that the efficiency of the coordination scheme with respect to travel time delay is inherited from the scheduling algorithm used. It is noted in [14] that the gains of coordination diminish as the vehicles come closer to the intersection, and a method that controls the vehicles long before the intersection is proposed. In particular, the authors of [14] suggest to first cluster the vehicles into platoons and, thereafter, coordinate the intersection crossing of these platoons rather than the individual vehicles. The intersection coordination problem is addressed from a different angle in [15], where a supervisory control structure is proposed, which ensure collision avoidance. The proposed controller is designed to override the inputs of the drivers (or driver agents) when those take the system to a state from which a collision is unavoidable.

Recently, several authors have proposed OC and model predictive control (MPC) as frameworks in which the coordination problem can be formulated. For instance, in [16], an OC-based algorithm is developed, where the vehicles decide sequentially how the intersection should be crossed. In this scheme, each vehicle is required to find the optimal plan that avoids collision with the vehicles that precedes it in the decision order. The algorithm is applied to receding horizon control in a distributed fashion in [17]. Another distributed MPC scheme was proposed in [18], where the vehicles utilize the previous predictions from other vehicles to enforce collision avoidance. More precisely, each vehicle is assigned a priority and solves an OC problem, where the previously predicted positions of higher priority vehicles are used as constraints to ensure safety. Similarly, Kim and Kumar [19] propose to find a crossing order through a rule-based priority assignment and only let a vehicle cross the intersection if its future path is not occupied by a vehicle with higher priority. The priority assignment is incorporated in an MPC scheme through manipulation of terminal state constraint, which is set to a point beyond the intersection if the vehicle is allowed to cross or to a point before the intersection if it must come to a stop. Qian *et al.* [20] propose an MPC scheme that assumes a given crossing order and include constraints that restrict the control inputs of each vehicle to lie in an order-preserving set of inputs. This is shown to guarantee nominal collision avoidance. An MPC scheme for intersection crossing is proposed in [21], in which risk minimization is chosen as the objective. This leads to solutions where the vehicles cross the intersection with large margins with respect to collisions when possible, and small margins only when necessary. A continuous-time optimal controller for intersection coordination based on indirect methods is proposed in [22], where the crossing order is assumed to be given by a first-in first-out heuristic. Other related works include [23], where the problem is posed using a spatial rather than temporal OC formulation and [24] that uses a robust MPC formulation of the problem in the context of collision avoidance between automated and nonautomated vehicles. For a more thorough review of the state of the art on coordination problems for automated vehicles, the reader is referred to the two excellent survey papers [25], [26] and the references therein.

B. Contributions

The control strategy proposed in this paper builds on the OC formulation of [6] and [7] with the following novel contributions: 1) the formulation of a bilevel MPC for coordination at intersections; 2) an analysis of the stability and persistent feasibility properties of the bilevel controller; 3) an extension of the controller that provides robustness against perturbations and facilitate a practical implementation; 4) a description of a real-time implementation of the bilevel controller that use distributed computation; and 5) an experimental validation of the algorithm and an analysis of experimental results.

In contrast to the existing coordination strategies, we highlight that our approach uses a general formulation of the problem, which does not require specific motion profiles and can make use of a large class of prediction models, constraints, and objective functions. In addition, we are not aware of any other MPC-based coordination strategy, where the conditions for stability and persistent feasibility are established. Although experimental results on intersection coordination and collision avoidance algorithms have been reported before (see [27], [28], or [29]), this is, to the best of our knowledge, the first time an MPC-based coordination scheme has been validated on real vehicles. We point out that extensive comparisons with other existing approaches are beyond the scope

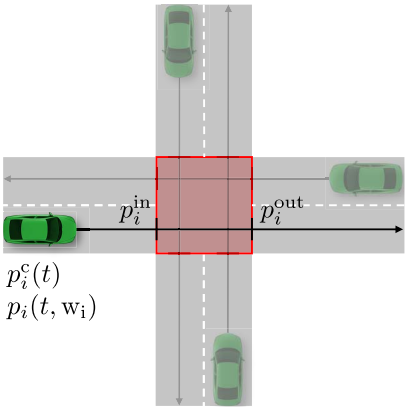


Fig. 1. Schematic of the scenarios considered in this paper. The arrows show the fixed paths of the vehicles, and the red square illustrates the zone inside the intersection, where collisions can occur.

of this paper and will be considered in the future research. Finally, while details on how the involved optimization problems are solved are indeed interesting and relevant, such details are largely left out of this paper, and we instead focus on the closed-loop control aspects. The interested reader is instead referred to [7] and [8].

C. Outline

The remainder of this paper is organized as follows. In Section II, we state the intersection coordination OC problem and introduce its application to receding horizon control. In Section III, we provide an analysis of the closed-loop system and present conditions under which the controller is both persistently feasible and stabilizes the system. In Section IV, we present a modification to the controller, which ensures persistent feasibility in the presence of perturbations, and present conditions under which robust collision avoidance can be ensured. In Section V, we describe the experimental setup used to validate the controller and present results from an experimental campaign. Finally, this paper is concluded in Section VI with a discussion of the contributions and final remarks.

II. Problem Formulation

In this paper, we are considering scenarios, such as that shown in Fig. 1, in which N_a CAVs need to cross an intersection. We assume that no noncooperative entities (vehicles, pedestrians, and so on) are present.

A. System Model

We assume that the vehicles move along predefined and fixed paths and model their motion along these paths as

$$\dot{x}_i^c(t) = f_i^c\left(x_i^c(t), u_i^c(t)\right) \tag{1}$$

where $x_i^c(t) \in \mathbb{R}^{n_i}$ denotes the state and $u_i^c(t)$ the control, and $f_i^c(\cdot)$ is continuously differentiable. Without loss of generality, we assume that the position $p_i^c(t)$ of the vehicle along its path is a state so that $x_i^c(t) = (p_i^c(t), z_i^c(t))$, where $z_i^c(t) \in \mathbb{R}^{n_i-1}$ collects all nonposition states. The states and controls are subject to constraints $h_i(x_i^c(t), u_i^c(t)) \leq 0$ capturing, e.g., actuator saturation and speed limits. We assume that (1) and $h_i(x_i^c(t), u_i^c(t)) \leq 0$ are such that $\dot{p}_i^c(t) > 0$ and $f_i^c(\cdot)$ is such that $p_i^c(t)$ does not depend on $u_i^c(t)$ directly.

We describe the intersection as the interval $[p_i^{\text{in}}, p_i^{\text{out}}]$ on the path on each vehicle i and define the times t_i^{in}, t_i^{out} when the vehicle enters and exits the intersection through

$$p_i^c\left(t_i^{\text{in}}\right) = p_i^{\text{in}} \quad \text{and} \quad p_i^c\left(t_i^{\text{out}}\right) = p_i^{\text{out}}. \tag{2}$$

A sufficient condition for collision avoidance is that

$$t_i^{\text{out}} \leq t_j^{\text{in}} \quad \text{or} \; t_j^{\text{out}} \leq t_i^{\text{in}} \quad \forall i, j, \; i \neq j. \tag{3}$$

In our formulation, we will use the discretized dynamics

$$x_{i,k+1} = f_i(x_{i,k}, u_{i,k}, t_s) \tag{4}$$

where $x_{i,k} \in \mathbb{R}^{n_i}$ and $u_{i,k} \in \mathbb{R}^{m_i}$ are the state and control of vehicle i at time $t_k = kt_s$, where t_s is the sampling time. Here, $f_i(x, u, \tau)$ denotes the solution to (1) at time $t = \tau$, starting from x at $t = 0$ and using $u_i^c(t) = u$, $t \in [0, \tau]$. In the discrete-time setting, the evolution of the vehicle position is described by a sequence $(p_{i,0}, p_{i,1}, \ldots)$ rather than a continuous trajectory, thus requiring modification of the definition of $t_i^{\text{in}}, t_i^{\text{out}}$ in (2). To this end, we introduce the following continuous-time representation of the position in the discrete-time setting:

$$p_i(t, w_i) := [1, \mathbf{0}^{1\times n_i-1}] f_i(x_{i,k}, u_{i,k}, t - kt_s) \tag{5}$$

where $w_i := (x_{i,0}, u_{i,0}, x_{i,1}, u_{i,1}, \ldots)$, $k = \lfloor t/t_s \rfloor$, and $\lfloor \cdot \rfloor$ denotes rounding down to the closest integer. That is, the position at $t \in]kt_s, (k+1)t_s[$ is obtained through the integration of continuous-time dynamics from state $x_{i,k}$ using input $u_{i,k}$, where k is dependent on t. For a given w_i satisfying (4), $p_i(t, w_i)$ is continuous in t, and we define $t_i^{\text{in}}, t_i^{\text{out}}$ in the discrete-time case through $p_i(t_i^{\text{in}}, w_i) = p_i^{\text{in}}$ and $p_i(t_i^{\text{out}}, w_i) = p_i^{\text{out}}$. The discretization of the position and its properties is discussed at length in [7]. We clarify the above-mentioned definitions by Example 1.

Example 1 (Double-Integrator Dynamics): The vehicle is modeled as a point on the path coordinate, and its motion is described by $\ddot{p}_i^c(t) = u_i^c(t)$. The discrete-time representation is

$$x_{i,k+1} = \begin{bmatrix} 1 & t_s \\ 0 & 1 \end{bmatrix} x_{i,k} + \begin{bmatrix} \frac{1}{2}t_s^2 \\ t_s \end{bmatrix} u_{i,k} \tag{6}$$

where $x_{i,k} = [p_{i,k}, v_{i,k}]^\top$, $v_{i,k}$ being the velocity along the path. In this case, using $k = \lfloor t/t_s \rfloor$, (5) becomes

$$p_i(t, w_i) = p_{i,k} + (t - t_k)v_{i,k} + \frac{1}{2}(t - t_k)^2 u_{i,k}. \tag{7}$$

Remark 2: We remark that the shape of the predefined paths does not influence how the scenario is modeled. Straight paths are not a necessity, and scenarios where some vehicles travel on turning paths can be described similarly.

B. Open-Loop Optimal Coordination

In this section, we introduce the OC formulation of the intersection coordination problem, first presented in [6]. Due to, e.g., finite horizon effects, model mismatches and perturbations acting on the real system, it is convenient to differentiate between the open- and closed-loop trajectories. We therefore denote the open-loop prediction of the state at time $(k+n)t_s$ as $\bar{x}_{i,k+n}$, recalling that the actual state of the vehicle at time t_k is $x_{i,k}$. Moreover, we denote the predicted intersection occupancy timeslot as $T_i = (t_i^{\text{in}}, t_i^{\text{out}})$ and define the stacked timeslots as $\boldsymbol{T} := (T_1, \ldots, T_{N_a})$ as well as the stacked state of all vehicles, $X_k := (x_{1,k}, \ldots, x_{N_a,k})$. For a given time slot T_i, the predicted optimal state and control trajectories of vehicle i at time t_k are obtained as the minimizer of the following finite-time OC problem:

$$
\begin{aligned}
&V_i(x_{i,k}, T_i) \\
&\quad := \min_{w_i} \; V_i^f(\bar{x}_{k+N_i}) + \sum_{n=0}^{N_i-1} \ell_i(\bar{x}_{i,k+n}, \bar{u}_{i,k+n}) && \text{(8a)}\\
&\qquad \text{s.t. } \bar{x}_{i,k} = x_{i,k} && \text{(8b)}\\
&\qquad\quad \bar{x}_{i,k+n+1} = f_i(\bar{x}_{i,k+n}, \bar{u}_{i,k+n}), \quad n \in \mathbb{I}_{[0,N_i-1]} && \text{(8c)}\\
&\qquad\quad h_i(\bar{x}_{i,k+n}, \bar{u}_{i,k+n}) \le 0, \quad n \in \mathbb{I}_{[1,N_i]} && \text{(8d)}\\
&\qquad\quad p_i\big(t_i^{\text{in}}, w_i\big) - p_i^{\text{in}} \le 0 && \text{(8e)}\\
&\qquad\quad p_i^{\text{out}} - p_i\big(t_i^{\text{out}}, w_i\big) \le 0. && \text{(8f)}
\end{aligned}
$$

where $V_i^f(\cdot)$ is the terminal cost, $\ell_i(\cdot)$ is the stage cost, both continuously differentiable, $N_i \in \mathbb{N}$ is the prediction horizon, $w_i := (\bar{x}_{i,k}, \ldots, \bar{x}_{i,k+N_i}, \bar{u}_{i,k}, \ldots, \bar{u}_{i,k+N_i-1})$, and $\mathbb{I}_{[a,b]} = \{a, a+1, \ldots, b\}$ for integers a and b. Here, and in the remainder of this paper, we write $f_i(x,u,\tau) = f_i(x,u)$ when $\tau = t_s$.

We let $\mathcal{T}_i(x_{i,k})$ be the set of timeslots T_i for which the parametric optimization problem (8) is feasible for $x_{i,k}$ at time t_k and define the intersection crossing order as $S := (s_1, \ldots, s_{N_a})$. The crossing order S is a permutation of the set $\mathbb{I}_{[1,N_a]}$, such that vehicle s_i crosses the intersection before vehicle s_{i+1}. The optimal collision-free timeslots at t_k are obtained as the solution to

$$
\begin{aligned}
V(X_k) := \min_{\boldsymbol{T}, S} \; & \sum_{i=1}^{N_a} V_i(x_{i,k}, T_i) && \text{(9a)}\\
\text{s.t. } & T_i \in \mathcal{T}_i(x_{i,k}), \quad i \in \mathbb{I}_{[1,N_a]} && \text{(9b)}\\
& t_{s_i}^{\text{out}} \le t_{s_{i+1}}^{\text{in}} \quad i \in \mathbb{I}_{[1,N_a-1]} && \text{(9c)}\\
& S \in \text{perm}(\mathbb{I}_{[1,N_a]}) && \text{(9d)}
\end{aligned}
$$

where $\text{perm}(\mathbb{I}_{[1,N_a]})$ denotes the set of all permutations of the index set $\mathbb{I}_{[1,N_a]}$. The timeslot allocation problem (9) thus includes both finding the order S, in which the vehicles cross, and the continuous-time schedule $\boldsymbol{T}$, which minimizes the aggregated cost of all vehicles. We note that due to the constraint (9d), the problem (9) is combinatorial.

C. Receding Horizon Optimal Coordination

A closed-loop coordination is achieved by applying (8) and (9) in a receding horizon fashion. Consequently, at each time instant k, the optimization problems (8) and (9) are solved, and each vehicle applies the resulting minimizing control action $\bar{u}_{i,k}^*$. As illustrated in Fig. 2, we implement the closed-loop controller as a bilevel feedback structure, where the current state of a vehicle is used not only to compute that vehicle's own control action but, through the timeslots, also the feedback laws of all other vehicles. In particular, denoting the solution to (9) at X_k as $\boldsymbol{T}^*(X_k)$, the closed-loop dynamics for vehicle i is

$$
x_{i,k+1} = f_i\big(x_{i,k}, \bar{u}_{i,k}^*\big(x_{i,k}, T_i^*(X_k)\big)\big). \tag{10}
$$

In the following, we refer to the receding horizon application of (8) as the *vehicle-level* control loop, differentiating it from the *intersection-level* control loop, consisting of the calculation of $\boldsymbol{T}^*(X_k)$ through (9). Next, we study the properties of the nominal bilevel controller and establish persistent feasibility and stability for the closed-loop system (10).

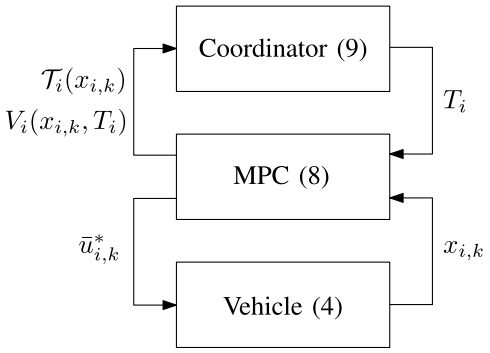


Fig. 2. Schematic of the bilevel control structure for one vehicle. The coordinator is in closed loop with all vehicles in the same way.

III. Closed-Loop Stability and Persistent Feasiblity

We first introduce the OC problem

$$
V_i^u(x_{i,k}) := \min_{w_i} \; \text{(8a)}, \quad \text{s.t. (8b)} - \text{(8d)} \tag{11}
$$

which is the vehicle-level optimization problem (8) without the position constraints (8e) and (8f). This corresponds to the optimal uncoordinated case, where the collisions are not explicitly avoided. Furthermore, we denote $\mathcal{N}_i := \{x_{i,k} \mid z_{i,k} = z_i^{\text{ref}}\}$, where z_i^{ref} is a reference, such that $\mathcal{N}_i \subset \mathcal{H}_i$ and $\mathcal{H}_i := \{x_{i,k} \mid \exists u_{i,k} : h_i(x_{i,k}, u_{i,k}) \le 0\}$. In addition, we define the distance from a point a to a set $\mathcal{A}$ as

$$
|a|_{\mathcal{A}} = \min_{b \in \mathcal{A}} \; ||a - b|| \tag{12}
$$

and say that a vehicle state is stabilized to $\mathcal{A}$ if $|x_{i,k}|_{\mathcal{A}} \to 0$ as $k \to \infty$ [30]. Finally, we make Assumptions 3 and 4.

Assumption 3 (Stability and Persistent Feasibility of Uncoordinated Vehicle-Level MPC): The vehicle-level MPC formulated based on (11) satisfies $\ell_i(x_{i,k}, u_{i,k}) \ge \alpha_{1,i}(|x_{i,k}|_{\mathcal{N}_i})$ for all feasible $x_{i,k}$ and $u_{i,k}$ and $V_i^f(x_{i,k}) \le \alpha_{2,i}(|x_{i,k}|_{\mathcal{N}_i})$ for $x_{i,k} \in \mathcal{H}_i$, where $\alpha_{1,i}$ and $\alpha_{2,i}$ are $\mathcal{K}_\infty$-functions. Moreover, there exists a controller $\kappa^f(x_{i,N})$, such that $V_i^f(f(x_{i,N}, \kappa^f(x_{i,N}))) - V_i^f(x_{i,N}) \le -\ell_i(x_{i,N}, \kappa^f(x_{i,N}))$ and $f_i(x_{i,k}, \kappa^f(x_{i,k})) \in \mathcal{H}_i$ for all $x_{i,N} \in \mathcal{H}_i$.

Assumption 4 (Regularity of Vehicle-Level Optimal Control Problems): For all $x_{i,k}$, such that Problem (11) is feasible, linear independence constraint qualifications and the second-order sufficient conditions hold at the solution.

Neither Assumption 3 nor 4 is restrictive. Assumption 3 states that the receding horizon application of (11) is persistently feasible and implies that $V_i^u(\cdot)$ is a Lyapunov function for the closed-loop system so that the vehicle state is stabilized to the set $\mathcal{N}_i$. That is, if no other cars are present and no coordination is required, the vehicle will track the reference z_i^{ref}. Assumption 4 is standard in the MPC context and means that MPC problem (11) is well posed.

A. Persistent Feasibility

While a receding horizon controller based on (11) is persistently feasible by Assumption 3, this is not necessarily the case for the receding-horizon application of (8). This is due to the presence of the position constraints (8e) and (8f). We note, in particular, that due to (5), (8e) and (8f) are formulated using the state and control at time index $k_i^{\text{in}} = \lfloor t_i^{\text{in}}/t_s \rfloor$ and $k_i^{\text{out}} = \lfloor t_i^{\text{out}}/t_s \rfloor$, respectively. That is, at time instant k, (8e) and (8f) constrain the predicted state and control at $n_i^{\text{in}} = k_i^{\text{in}} - k$ and $n_i^{\text{out}} = k_i^{\text{out}} - k$ time steps in the future. Consequently, as the system evolves and k increases, n_i^{in} and n_i^{out} decrease if t_i^{in} and t_i^{out} are constant, and the constraints (8e) and (8f) are "shifted" toward the beginning of the prediction horizon. These issues are addressed in Proposition 5, where its proof is given in Appendix A.

Proposition 5 (Nominal Persistent Feasibility): Suppose that Assumption 3 holds that the vehicle states X_k at time t_k are such that a feasible solution to (9) exists and that the bilevel controller (10) is applied to the vehicles. Then, the optimization problems (8) and (9) are feasible at time t_{k+n}, $n > 0$.

B. Nominal Stability

By Assumption 3, an MPC based on (11) stabilizes the vehicle state to the set $\mathcal{N}_i$ $\forall i$. However, the assumption does not directly imply stability in the coordinated case, where the MPC is based on (8), due to the introduction of the constraints (8e) and (8f). Consider, for instance, the case where the closed-loop system is initialized so that $x_{i,k} \in \mathcal{N}_i$, and collisions occur if the vehicles are not coordinated. By Assumption 3, the uncoordinated closed-loop system will remain in $\mathcal{N}_i$ by use of an MPC based on (11). However, for a collision-free timeslot schedule $\boldsymbol{T}$, $x_{i,k}$ can be forced to deviate from $\mathcal{N}_i$ to satisfy (8e) and (8f) when an MPC based on (8) is used. Consequently, for collision-free stabilization to $\mathcal{N}_i$, both $V_i^u(x_{i,k})$ and $V_i(x_{i,k}, T_i)$ can increase, and therefore, they are not Lyapunov functions for the coordinated closed-loop system. For this reason, the notion of stability is modified, and we consider stabilization to the sets $\mathcal{Q}_i = \mathcal{N}_i \cap \mathcal{P}_i$, where $\mathcal{P}_i = \{x_{i,k} \mid p_{i,k} \geq p_i^{\text{out}}\}$. We formalize the stability result in Theorem 6, where its proof is given in Appendix B.

Theorem 6 (Stability of the Bilevel Controller): Suppose that the timeslots are updated through the solution of (9) and Assumptions 3 and 4 hold. Then, the bilevel controller stabilizes X_k to $\mathcal{Q} = \prod_{i=1}^{N_a} \mathcal{Q}_i$, where $\prod$ denotes the Cartesian product. Moreover, the bilevel controller is stabilizing if:1) only $\boldsymbol{T}$ is optimized and S is fixed; 2) $\boldsymbol{T}$ and/or S optimized at a lower rate than the vehicle-level problems; or 3) both $\boldsymbol{T}$ and S are fixed.

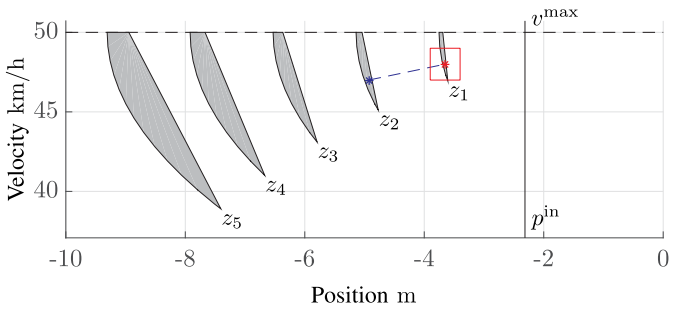


Fig. 3. Illustration of the feasible sets of (8) for a given T_i using the double-integrator system of Example 1 and $t_s = 0.1$. The set z_j is the set of states $x_{i,k}$ for which (8) has a solution when $t_i^{\text{in}} - t_k = jt_s$. The red star is a one-step prediction from the blue star and the red box the bounds of a modest uniform uncertainty (± 1 km/h and ± 0.25 m), stemming from, e.g., model mismatch, measurement errors, or external perturbations. This figure illustrates that since z_j is shrinking as the vehicle approach the intersection, infeasibility of (8) is more likely.

IV. Robustness With Respect to Perturbations

The vehicle-level MPC Problem (8) differs from standard MPC problem formulations in that the position constraints (8e) and (8f) force the system to be at a prescribed position at a given time. In the presence of perturbations, e.g., process noise, or measurement errors, satisfaction of such constraints is difficult in practice, and the persistent feasibility guarantees of the nominal case no longer hold. Since violations of the position constraints (8e) and (8f) by the vehicles impliy a risk of collisions inside the intersection, the issue must be resolved for the bilevel controller to be useful in practice. In this section, we present a relaxation of the bilevel control formulation, which ensures persistent feasibility of the optimization problems (8) and (9). Furthermore, we also state the conditions under which the closed-loop system is collision-free.

A. Ensuring Optimization Problem Feasibility

As described in Section III-A, the position constraints (8e) and (8f) move closer to the first prediction stage as the vehicle approachs the intersection and t_k approaches t_i^{in} (and t_i^{out}). As a result, the number of stages at which the controller has authority to affect the satisfaction of the position constraints (8e) and (8f) decreases. As illustrated in Fig. 3, this causes the set of states for which the vehicle-level MPC problem (8) has a solution for a given T_i to shrink as the closed-loop system evolves. Consequently, with measurement uncertainty, external perturbations, and model-plant mismatches, it is likely that (8) becomes infeasible for some $x_{i,k}$. In particular, the risk increases when the vehicle is close to the intersection. To ensure feasibility of the optimization problems (8) and (9) along the closed-loop system trajectories, we propose the following relaxation of the constraints of the

vehicle MPC problem (8), where $\sigma_i = (\sigma_i^{\text{in}}, \sigma_i^{\text{out}})$ are slack variables and $\rho_i(\cdot)$ is an exact penalty function:

$$
\begin{aligned}
V_i^r(x_{i,k}, T_i) := \min_{w_i, \sigma_i} \ & \rho_i(\sigma_i) + (8a) && \text{(13a)}\\
\text{s.t. } & (8b), (8c), (8d) && \text{(13b)}\\
& p_i\big(t_i^{\text{in}}, w_i\big) - p_i^{\text{in}} \le \sigma_i^{\text{in}} && \text{(13c)}\\
& p_i^{\text{out}} - p_i\big(t_i^{\text{out}}, w_i\big) \le \sigma_i^{\text{out}} && \text{(13d)}\\
& \sigma_i^{\text{out}} \ge 0, \quad \sigma_i^{\text{in}} \ge 0. && \text{(13e)}
\end{aligned}
$$

The solution to (13) has the following properties.

Proposition 7 (Characteristics of Solutions to the Relaxed Vehicle Problem): Provided that $\nabla\rho_i(0)$ is large enough, the relaxed problem has the following properties.

1) Whenever there exists a feasible solution for the original problem (8), the relaxed problem (13) yields the same primal solution.
2) Otherwise, the relaxed problem (13) yields a solution that minimizes $||\sigma_i||_\infty$.

Proof: Claim 1) is well-known, and a proof can be found in, e.g., [31, Th. 14.3.1]. However, we could not find a proof for 2) in the literature and, therefore, provide one in Appendix C, where we formalize the result as Theorem 13. □

We note that with the relaxation of the constraints, the set of $x_{i,k}$ for which (13) is feasible is the same as for the uncoordinated problem (11), i.e., the timeslot T_i does not affect feasibility. As a consequence, the set of feasible times $\mathcal{T}_i(x_{i,k})$ of the modified problem becomes $\mathbb{R}^2$, and the intersection-level problem (9) can be written as

$$
\min_{\boldsymbol{T}, S} \ \sum_{i=1}^{N_a} V_i^r(x_{i,k}, T_i) \quad \text{s.t. } (9c), (9d). \tag{14}
$$

Consequently, the use of (13) instead of (8) relaxes (9), and the following result holds for the characteristics of its solutions.

Proposition 8 (Characteristics of Solutions to the Relaxed Timeslot Allocation Problem): If for all i, $\rho_i(\cdot)$ is chosen so that $\nabla\rho_i(0)$ is large enough, the solution to the relaxed timeslot allocation problem (14) coincides with the solution to (9), whenever a solution to (9) exists. When a solution to (9) does not exist, the solution $(S, \boldsymbol{T})$ to (14) is such that the largest constraint violation in (13) among all vehicles is minimized.

Proof: The intersection-level problem (9) can equivalently be stated as

$$
\begin{aligned}
\min_{\boldsymbol{T}, S, \sigma, W} \ & \sum_{i=1}^{N_a} \rho_i(\sigma_i) + \sum_{i=1}^{N_a} (8a) && \text{(15a)}\\
\text{s.t. } & (13b) - (13e), \quad i \in \mathbb{I}_{[1,N_a]} && \text{(15b)}\\
& t_{s_i}^{\text{out}} \le t_{s_{i+1}}^{\text{in}}, \quad i \in \mathbb{I}_{[1,N_a-1]} && \text{(15c)}\\
& S \in \text{perm}(\mathbb{I}_{[1,N_a]}) && \text{(15d)}
\end{aligned}
$$

where $\sigma := (\sigma_1, \ldots, \sigma_{N_a})$ and $W := (w_1, \ldots, w_{N_a})$. The result follow from Proposition 7. □

This means that if a previously feasible timeslot schedule $\boldsymbol{T}$ has become infeasible for one or more vehicles, the timeslot allocation controller (14) utilizes the authority it has over *all* vehicles to, if necessary, find the $(\boldsymbol{T}, S)$ that results in the smallest constraint violation $\|\sigma\|_\infty$.

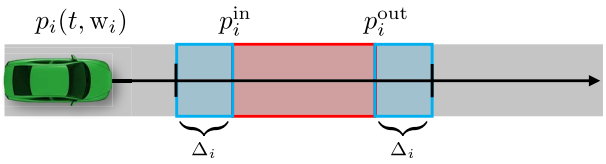


Fig. 4. Illustration of the enlargement of the intersection resulting in a tightening of constraints (13c) and (13d)

B. Ensuring Closed-Loop Collision Avoidance

Although the relaxation of (8e) and (8f) ensures persistent feasibility of (13), the closed-loop system is not guaranteed to satisfy the collision avoidance constraints. However, from Proposition 7, we know that the modified bilevel controller minimizes the violations of the position constraints (8e) and (8f). If an upper bound can be found to the largest violation of (8e) and (8f) by the closed-loop system, (13) can be modified to ensure the collision avoidance of the closed-loop system. In particular, consider the following modification of (13):

$$
\begin{aligned}
& V_i^t(x_{i,k}, T_i, \Delta_i)\\
& := \min_{w_i} \ (8a) && \text{(16a)}\\
& \quad \text{s.t. } (8b), (8c), (8d)(13e) && \text{(16b)}\\
& \qquad p_i\big(t_i^{\text{in}}, w_i\big) - p_i^{\text{in}} + \Delta_i \le \sigma_i^{\text{in}} && \text{(16c)}\\
& \qquad p_i^{\text{out}} - \Delta_i - p_i\big(t_i^{\text{out}}, w_i\big) \le \sigma_i^{\text{out}} && \text{(16d)}
\end{aligned}
$$

where $\Delta_i \ge 0$ is a tightening of the constraints, corresponding to an enlargement of the intersection, as illustrated in Fig. 4. We then have the following.

Proposition 9: Suppose that $\boldsymbol{T}$ satisfies (9c) for some S and that $\forall i$, at t_0, (16) is feasible for $(x_{i,k}, T_i, \Delta_i)$. Then, the closed-loop application of (16) ensures that no collisions occur if Δ_i is chosen so that $\Delta_i \ge \sigma_i^{\text{in}}$ and $\Delta_i \ge \sigma_i^{\text{out}}$ $\forall k$ and $\forall i$.

Proof: Denoting the position of the closed-loop system at time t_i^{in} as $\bar{p}_i(t_i^{\text{in}})$, we have $\bar{p}_i(t_i^{\text{in}}) \le p_i^{\text{in}} - \Delta_i + \sigma_i^{\text{in}}$, which due to $\Delta_i \ge \sigma_i^{\text{in}}$ gives that $\bar{p}_i(t_i^{\text{in}}) \le p_i^{\text{in}}$. Similarly, due to $\Delta_i \ge \sigma_i^{\text{out}}$, we have that $\bar{p}_i(t_i^{\text{out}}) \ge p_i^{\text{out}}$. Consequently, $\bar{p}_i(t) \in [p_i^{\text{in}}, p_i^{\text{out}}] \Rightarrow t \in [t_i^{\text{in}}, t_i^{\text{out}}]$ and $t \notin [t_i^{\text{in}}, t_i^{\text{out}}] \Rightarrow \bar{p}_i(t) \notin [p_i^{\text{in}}, p_i^{\text{out}}]$. Since $[t_i^{\text{in}}, t_i^{\text{out}}] \cap [t_j^{\text{in}}, t_j^{\text{out}}] = \emptyset, j \ne i$ due to (9c), $\bar{p}_i(t) \in [p_i^{\text{in}}, p_i^{\text{out}}] \Rightarrow \bar{p}_j(t) \notin [p_j^{\text{in}}, p_j^{\text{out}}]$, $i \ne j$ ensuring that no collisions can occur for the closed-loop system. □

Remark 10: While easy to include in the control formulation, finding a tight upper bound Δ_i can be challenging, as it includes characterizing the closed-loop systems response, in particular the satisfaction of constraints (13c) and (13d) to model mismatch, measurement errors, and external perturbations.

V. Experimental Validation

In this section, we present experimental results from a three-way intersection scenario, obtained from an implementation of the bilevel controller on a test setup consisting of three automated vehicles equipped with GPS positioning and

V2V communication systems. For ease of implementation, the intersection-level problem (9) was solved for the fixed order S during the experiments. However, we illustrate, in simulation, the behavior of the closed-loop system when the intersection-level control loop includes computation of S.

A. Experimental Setup

The experimental validation of the bilevel, closed-loop controller was performed at the Asta Zero test track outside Gothenburg, Sweden, on a heterogeneous test setup consisting of three different Volvo vehicles. In all experiment runs, the vehicles were initialized in configurations that would lead to collisions if no action was taken. From these initial configurations, the bilevel controller was used to control the vehicles through the intersection. The experimental validation was carried out in two different modes: first, in a parallel configuration on a highway stretch, where the intersection was represented by a mutually exclusive segment of the road, shown in Fig. 5(a), and second, in an actual intersection, shown in Fig. 5(b). In particular, the parallel configuration was used to safely carry out the validation at higher speeds. Due to collision risks, the controller was only validated at speeds up to 50 km/h and with added safety margins in the crossing configuration. We emphasize that there were no differences in the controller between the two configurations.

1) Test Platform: The test setup consisted of a central computer, acting as a coordinator, and three Volvo Cars: one Volvo S60 D5 turbo-diesel, one Volvo S60 T6 petrol-turbo, and one Volvo XC90 T6 petrol-turbo. Each vehicle were commanded through an interface to a low-level controller, which tracked commanded acceleration inputs. The vehicles were equipped with real-time kinematic GPS receivers, wheel encoders and inertial measurement units, and V2V communication devices from RENDITS [33]. Furthermore, to improve positioning accuracy, the measurements were fused using extended Kalman filters (EKFs), as described in [29]. The 1-D position $p_{i,k}$ was constructed by projecting the current estimate of the vehicle's global position onto a reference path along the road. Finally, the clocks in the three vehicles were synchronized through the GPS receivers.

2) Prediction Model and Objective Function: For simplicity, we used the double-integrator dynamics introduced in Example 1 as the prediction model in (8). The choice of such a simple model removed the need to identify any parameters of the real systems and allowed the same controller to be used on all cars. Furthermore, the path constraints (8d) were $0 \leq v_{i,k}$, $u_{i,k} \geq u_i^{\min} = -3$ m/s^2, and $u_{i,k} \leq u_i^{\max} = 1.6$ m/s^2 $\forall i, k$, where the input bounds reflect bounds present in the vehicles' actuation interfaces. Finally, the objective was chosen as

$$J_i(w_i) = \left(v_{i,N} - v_i^{\text{ref}}\right)^2 Q_i^f + \sum_{i=0}^{N_i-1} \left(v_{i,k} - v_i^{\text{ref}}\right)^2 Q_i + u_{i,k}^2 R_i \tag{17}$$

where R_i, Q_i, and Q_i^f are positive constants and v_i^{ref} is a reference speed. To ensure that the vehicle-level controller remained feasible throughout the experiments, the relaxed formulation (13) was used to compute the control commands $u_{i,k}$. The exact penalty function

$$\rho_i(\sigma_i) = \frac{1}{2}\sigma_i^\top \sigma_i \phi_i^q + \phi_i \mathbf{1}^\top \sigma_i \tag{18}$$

was used, where ϕ_i and ϕ_i^q are positive weights. With this prediction model, objective, and penalty function, (13) and (8) are the convex quadratic programs (QPs).

3) Algorithmic Implementation: The solution to (13) and (8) was obtained using the dedicated MPC-QP-solver HPMPC [34]. To solve the fixed-order timeslot allocation problem (9), we employed the semidistributed SQP method developed in [7] and [8], which we briefly describe here. As noted in [7], the constraint (9b) can be written as $g_i(x_{i,k}, T_i) \leq 0$, $\forall i$ and evaluated by solving four linear programs. With $\mathcal{L}_i(x_{i,k}, T_i, \lambda_i) = V_i(x_{i,k}, T_i) + \lambda_i^\top g_i(x_{i,k}, T_i)$, where λ_i is the Lagrange multiplier corresponding to (9b) and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{N_a})$, the Lagrange function of (9) is

$$\mathcal{L}(\boldsymbol{T}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i=1}^{N_a} \mathcal{L}_i(x_{i,k}, T_i, \lambda_i) + \sum_{i=1}^{N_a-1} \mu_i \left(t_{s_i}^{\text{out}} - t_{s_{i+1}}^{\text{in}}\right) \tag{19}$$

where μ_i is the Lagrange multiplier corresponding to the precedence constraint (9c) and $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_{N_a-1})$. To solve the intersection-level problem, (9), the iteration

$$\boldsymbol{T}^+ \leftarrow \boldsymbol{T} + \alpha \Delta \boldsymbol{T} \tag{20a}$$

$$\boldsymbol{\lambda}^+ \leftarrow \boldsymbol{\lambda}(1-\alpha) + \alpha \boldsymbol{\lambda}_{QP} \tag{20b}$$

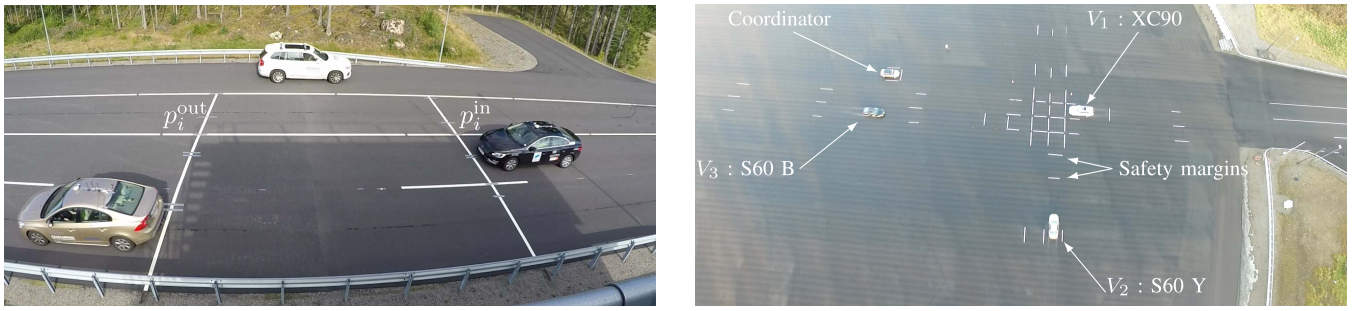
is performed until an exit criterion is met. Here, $\Delta \boldsymbol{T} = (\Delta T_1, \ldots, \Delta T_{N_a})$ and $\boldsymbol{\lambda}_{QP} = (\lambda_{i,QP}, \ldots, \lambda_{N_a,QP})$ are the primal-dual solution to the QP

$$\min_{\Delta \boldsymbol{T}} \sum_{i=1}^{N_a} \frac{1}{2} \Delta T_i^\top H_i \Delta T_i^\top + \nabla_{T_i} V_i^r(x_{i,k}, T_i)^\top \Delta T_i \tag{21a}$$

$$\text{s.t. } g_i(x_{i,k}, T_i) + \nabla_{T_i} g_i(x_{i,k}, T_i)^\top \Delta T_i \leq 0, \quad i \in \mathbb{I}_{[1,N_a]} \tag{21b}$$

$$t_{s_i}^{\text{out}} \leq t_{s_{i+1}}^{\text{in}}, \quad i \in \mathbb{I}_{[1,N_a-1]} \tag{21c}$$

where $\lambda_{i,QP}$ is the Lagrange multipliers of the constraint (21b). Furthermore, the step size α is in each iteration chosen through the line search procedure described in [8], and H_i is the possibly regularized Lagrange function Hessian $\nabla_{T_i}^2 \mathcal{L}_i(x_{i,k}, T_i, \lambda_i)$. The SQP procedure is summarized in Algorithm 1. Note that while a central coordinator is used, most computations, consisting of the evaluation of $\nabla_{T_i} V_i^r(x_{i,k}, T_i)$, $g_i(x_{i,k}, T_i)$, $\nabla_{T_i} g_i(x_{i,k}, T_i)$, and H_i, are performed in parallel on-board the vehicles, and the result is sent to the coordinator using the V2V communication. In particular, the communication between the coordinator and the vehicles is performed at lines 3, 5, and 7 in Algorithm 1. The only computations performed by the coordinator are the solution of the small SQP subproblems (21). We remark that if the order S is not fixed, the coordinator is required to solve (9) for both S and $\boldsymbol{T}$. In this case, the problem is combinatorial, and while distribution is still possible, the problem's exponential complexity is not affected by this fact. An approximate method that finds an order using the information of the first SQP iterate is presented in [35].



(a) (b)

Fig. 5. Photographs of the two different configurations used in the experimental validation. (a) Photograph of the parallel configuration used in the experimental validation. The white lines mark the beginning and end of the intersection, p_i^{in} and p_i^{out}, and collisions are thereby avoided when only one vehicle is between the two white lines at a given time. (b) Aerial photograph of the crossing configuration used in the experimental validation. The white lines before the square representing the intersection illustrate the different safety margins employed. Video material from the experiments can be found in [32].

Algorithm 1 Schematic description of the Distributed SQP. The arguments of the functions are removed for brevity.

1: Coordinator initializes the problem.
2: **while** exit conditions not fulfilled **do**
3: Coordinator broadcasts $\boldsymbol{T}$.
4: Each vehicle solves (8)
5: Each vehicle returns $\nabla V_i, \nabla^2 V_i, g_i, \nabla g_i, \nabla^2 g_i$.
6: Coordinator solves the SQP sub-problem (21).
7: Coordinator and vehicles compute α.
8: Coordinator takes step (20).

4) Parameters: The horizon and sampling time were set to $N_i = 200, t_s = 0.1$ s. Moreover, the size of the intersection was varied between and within the two different configurations: in the parallel configuration it was set to 10.7 m, and in the crossing configuration 17 and 12 m, corresponding to two different safety margins [see Fig. 5(b)].[1] R_i, Q_i, Q_i^f, and v_i^{ref} were varied between different experiment runs.

Due to the latencies introduced by the communication system and limitations of the hardware platforms used, the intersection-level control loop was closed at a lower rate than the vehicle-level control loop (8), and (9) was solved every $t_s^c = 3$ s. In particular, in most cases, the time required by Algorithm 1 ranged between 0.5 and 1.5 s. To compensate for this large delay, the timeslots, which were to be applied at time kt_s^c, were obtained by solving the intersection-level problem (9) between $(k-1)t_s^c$ and kt_s^c using a prediction of the system state X at kt_s^c. Finally, the intersection-level controller was applied until one vehicle reached a distance of 50 m before the intersection, after which the timeslot schedule was frozen.

B. Results

We index the vehicles with 1 for the XC90 and 2 and 3 for the two S60's, respectively, and consider scenarios, where the crossing order is $S = (1, 2, 3)$.

[1]The GPS antennas were mounted on the center of each vehicle, and the beginning and end of the intersection used in (8e) and (8f) were therefore set to $p_i^{\text{in}} - L_i/2$ and $p_i^{\text{out}} + L_i/2$, respectively, where L_i is the length of the vehicle. Satisfaction of (8e) consequently meant that the front of the vehicle was before p_i^{in} at t_i^{in}, and satisfaction of (8f) meant that the rear of the vehicle had passed p_i^{out} at t_i^{out}. This can be observed in Fig. 5(a).

1) Example Scenario: In one scenario, all vehicles were controlled to be at 200 m from the intersection at the same time while moving at 50 km/h, after which the bilevel controller was applied. The objective function weights were set to $Q_1 = 100$, $R_1 = 10$, and $Q_2 = Q_3 = 10$, $R_2 = R_3 = 1$, and the reference velocity to $v_i^{\text{ref}} = 50$ km/h, $i = 1, 2, 3$. The scenario was thereby one where all vehicles started at the reference velocity, equidistant from the intersection, so that all vehicles would enter the intersection at the same time and cause a three-way collision in the uncontrolled case.

Due to the weights, vehicles 2 and 3 were expected to use more control effort and deviate more from the reference than vehicle 1 in order to avoid collisions. As seen in Fig. 6(b) and (c), this was indeed the case: vehicle 1 increased its velocity slightly from the reference, whereas vehicles 2 and 3 made larger deviations and required larger accelerations The position trajectories in Fig. 6(a) illustrates that the bilevel level controller thereby satisfied the precedence constraints (9c). As emphasized by the cutout of Fig. 6(a), the timeslots of the closed-loop trajectories were packed tightly together but not overlapping. Since not controlling the vehicles would have resulted in a complete overlap, the tight packing of the timeslots additionally illustrates that with the objective function (17), the vehicles are not actuated more than necessary to avoid collisions.

Due to sensor noise, model-plant mismatch, and external disturbances, the one-step predictions and measured actual evolution of the vehicles differed. In addition to being continuously counteracted in the lower control loop, this discrepancy prompted the intersection-level control loop to perform adjustments to the timeslot schedule $\boldsymbol{T}$ as the vehicles approached the intersection. The difference between the optimal $\boldsymbol{T}$ at times kt_s^c and $(k+1)t_s^c$ was, in general, small and decreased as the vehicles got closer to the intersection. An illustration of the change in the optimal timeslot schedule $\boldsymbol{T}$ is provided in Fig. 6(d), clearly demonstrating this fact.

2) Perturbed Example Scenario: To highlight the ability of the controller to counteract perturbations, the same scenario was executed with a large input disturbance introduced to vehicle 1. After the bilevel controller had been initialized, the driver of the first vehicle pressed the brake pedal and thereby suspended the MPCs authority to control that vehicle

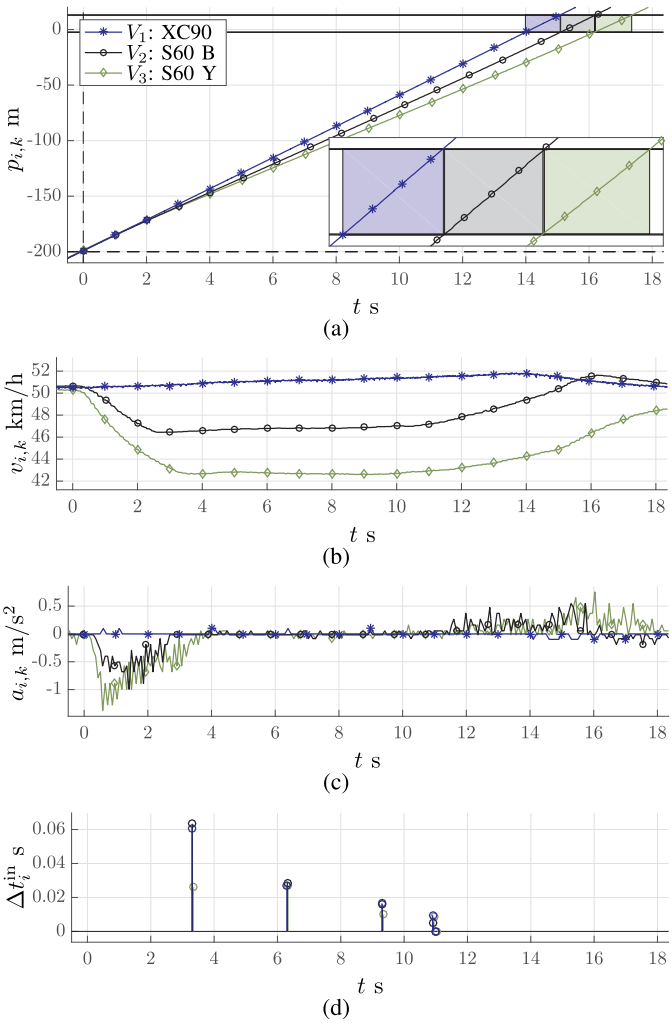


Fig. 6. Example from nonperturbed scenario. (a) Distance to intersection; the solid horizontal lines indicates the beginning and end of the intersection, and the dashed lines the scenario starting time and position. The colored blocks illustrate the timeslot used by each vehicle. (b) Velocity. (c) Acceleration. (d) Timeslot changes; difference in the optimal t_i^{in} between two executions of the intersection level control loop, as sent out by the coordinator.

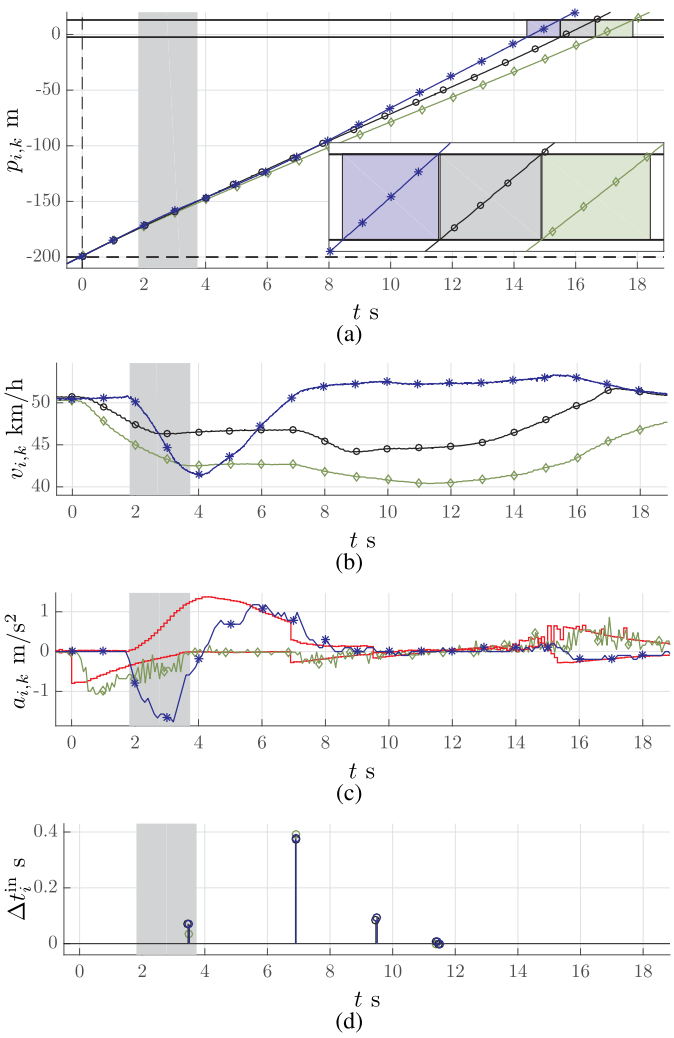


Fig. 7. Example from a perturbed scenario. (a) Distance to intersection. (b) Velocity. (c) Acceleration. (d) Timeslot changes. The gray bar illustrates the time during which the MPC controller lost authority of vehicle 1 because the driver overrode the system by pressing the brake pedal. The difference between the actual acceleration $a_{i,k}$ and the MPC command $u_{i,k}$ for vehicle 1 and 3 is illustrated in (c).

between $t = 1.8$ s and $t = 3.8$ s. The closed-loop system response is shown in Fig. 7. Fig. 7(d) illustrates the reaction of the intersection-level control loop to the disturbance and shows larger adjustments to the timeslots of all vehicles compared to the unperturbed case [see Fig. 6(d)]. Recall that since the time required to solve the intersection-level problem (9) is nonnegligible, the timeslot applied at kt_s^c is computed using information known at $(k-1)t_s^c$ (see Section V-A for details). Although the disturbance starts at $t = 1.8$ s, the timeslot adjustments are therefore not applied until $t = 6$ s

The difference between the actual acceleration resulting from the driver intervention and the acceleration commanded by the MPC is illustrated in Fig. 7(c). Note that the vehicle-level control loop at first attempts to counteract the perturbation with large acceleration commands to satisfy the timeslots commanded at $t = 0$ and $t = 3$. When the recomputed timeslot is applied shortly after $t = 6$, the control effort commanded by the vehicle-level controller loop of vehicle 1 is reduced noticeably, while it is increased for vehicle 3. This illustrates the ability of the intersection-level control loop to distribute the effort needed to counteract the perturbation between the vehicles according to the objective (17).

To further demonstrate the capabilities of the bilevel controller, results from a simulation of the perturbed scenario, where the intersection-level control loop includes recomputing the order s, are given in Fig. 8. In particular, the vehicles are initialized as in the experimental scenario, and the recorded perturbation from the experiment is applied to the simulated vehicle 1. Here, the vehicles were simulated using a double integrator with first-order actuator dynamics, i.e., $\dot{p}_i(t) = v_i(t)$, $\dot{v}_i(t) = a_i(t)$, and $\dot{a}_i(t) = (1/\tau)(u_i(t) - a_i(t))$, where $\tau = 0.5$ s for the two S60s and $\tau = 1$ s for the larger XC90, and the combinatorial problem of finding S was solved by enumeration. The prediction model used and the remaining parameters of the simulated vehicles (e.g., input bounds and objective weights) were set according to Section V-A2. In this specific scenario, the perturbation caused the crossing order to change to $S = (2, 1, 3)$ at $t = 3$, resulting in a change in $\boldsymbol{T}$

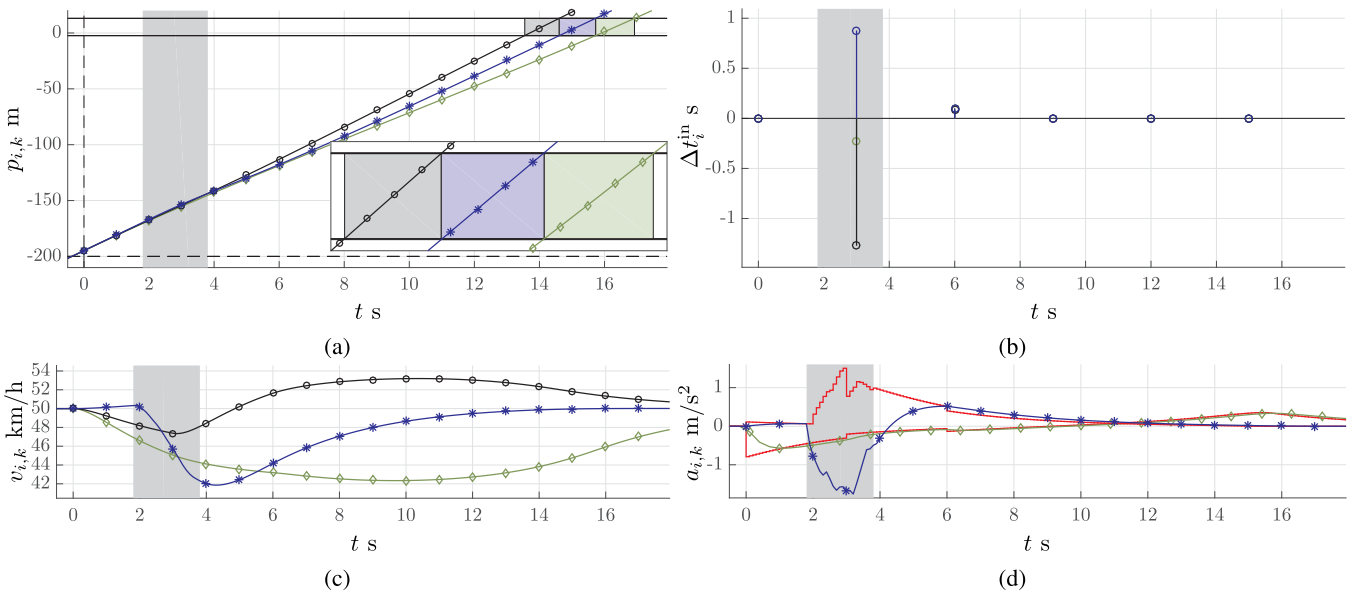


Fig. 8. Example from perturbed simulated scenario including recalculation of the order S. The disturbance from Fig. 7 is applied to the simulated vehicle 1. (a) Distance to intersection. (b) Timeslot changes. (c) Velocity. (d) Acceleration.

that is significantly larger than in the experiment. The effect of the large delays present during the experiment can be seen by comparing Figs. 7(d) and 8(b); in the simulated scenario, the intersection-level control loop can react instantaneously to the disturbance and start to counter it already at $t = 3$.

3) Controller Consistency and Constraint Violations: As illustrated in Fig. 7(a), the introduction of the large perturbation did not significantly degrade the ability of the closed-loop system to satisfy the precedence constraints (9c). During the experimental campaign, more than 75 runs were performed, which involved speeds ranging from 30 to 90 km/h, more than 20 different sets of parameters (objective function weights, starting configurations, and so on), and, in several cases, larger perturbations. With a few exceptions, the bilevel controller consistently managed to command the vehicles so that little or no overlap of the timeslots occurred. In 56.7%[2] of the evaluations, the position constraints (8e) and (8f) were satisfied by the closed-loop system trajectories, whereas in the remaining 40.2%, the constraint violation remained below 0.7 m. In 3.1% of the cases, the constraint violations were larger than 1 m as the result of infeasible initial conditions, too large perturbations close to the intersection or experimental parameter tuning. While these runs were considered failures, the results were expected. The distribution of the constraint violations are shown in Fig. 9, and we note especially that for 90% of the cases, where the constraints were not satisfied, the violations remained below 0.4 m. We want to highlight that these violations consist of the front of the entering vehicle protruding a few decimeters into the intersection while a few decimeters of the rear of the leaving vehicle is still inside.

[2]The data were collected during 75 experimental runs, each with three involved vehicles and two position constraint evaluations per vehicle. Of the 450 total evaluations, 255 satisfied the constraints, 181 had a violations smaller than 1 m, and 14 had a violation larger than 1 m.

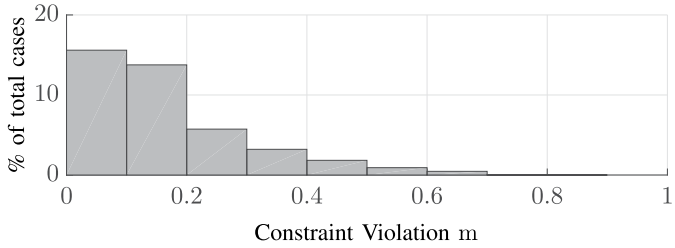


Fig. 9. Constraint violation statistics from 75 experiment runs, corresponding to 450 evaluations of the position constraints (8e) and (8f) by the closed-loop system. In 255 out of 450 evaluations, the constraints were satisfied by the closed-loop system. A histogram of the constraint violations in the remaining cases is given here. The plot only shows the data from successful runs, where the violation was smaller than 1 m.

In Fig. 10, we illustrate the consistency of the controller by reporting the position trajectories of the closed-loop system in 34 of the successful experimental runs performed in the parallel configuration. Fig. 10 shows an overlay of the system trajectories in the configuration spaces of the vehicle pairs $(1, 2)$ and $(3, 2)$ for each experiment so that the x-axis is the position of vehicle 1 and 3 and the y-axis is the the position of vehicle 2. The intersection is represented as the rectangle $\mathcal{R} = ([p_1^{\text{in}}, p_1^{\text{out}}] \times [p_2^{\text{in}}, p_2^{\text{out}}]) \cup ([p_3^{\text{in}}, p_3^{\text{out}}] \times [p_2^{\text{in}}, p_2^{\text{out}}])$, and consequently, if a trajectory is inside $\mathcal{R}$, more than one vehicle is inside the intersection at the same time. The small constraint violations reported in Fig. 9 are noticeable in Fig. 10 as the overlaps of the trajectories on the corners of $\mathcal{R}$.

4) Robustness Aspects: The fact that the constraint violation statistics shown in Fig. 9 are collected over a range of parameter settings, including speeds and intersection sizes, is an indication that the accuracy of the constraint enforcement is a property of the closed-loop system. Such data could therefore be used to determine an appropriate size of the constraint tightening parameter Δ_i, as discussed in Section IV-B.

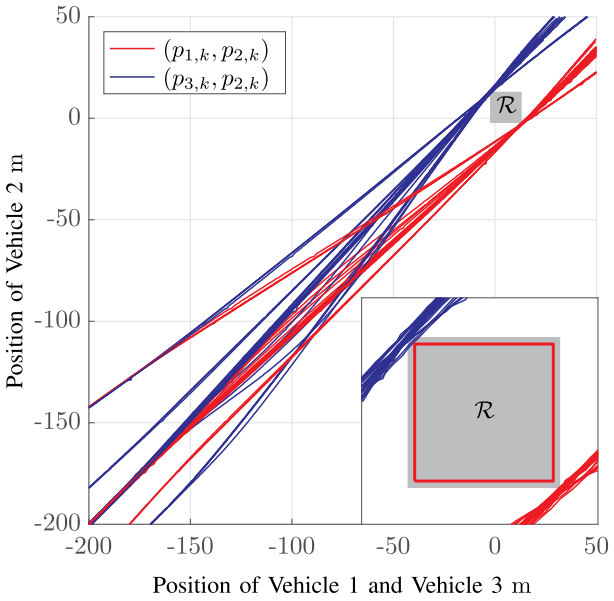


Fig. 10. Position trajectories from 34 experimental runs with three vehicles. The trajectories of the system in the position space of vehicles 1 and 2 are shown in red lines, whereas the corresponding trajectories for vehicles 2 and 3 are shown in blue lines. The gray rectangle $\mathcal{R} = ([p_1^{\text{in}}, p_1^{\text{out}}] \times [p_2^{\text{in}}, p_2^{\text{out}}]) \cup ([p_3^{\text{in}}, p_3^{\text{out}}] \times [p_2^{\text{in}}, p_2^{\text{out}}])$ corresponds to the intersection, and consequently, when a trajectory is inside it, at least one involved vehicle violates one of the position constraints (8e) and (8f). The red square illustrates an intersection, where the width is 1.4 m less than $\mathcal{R}$ in all dimensions.

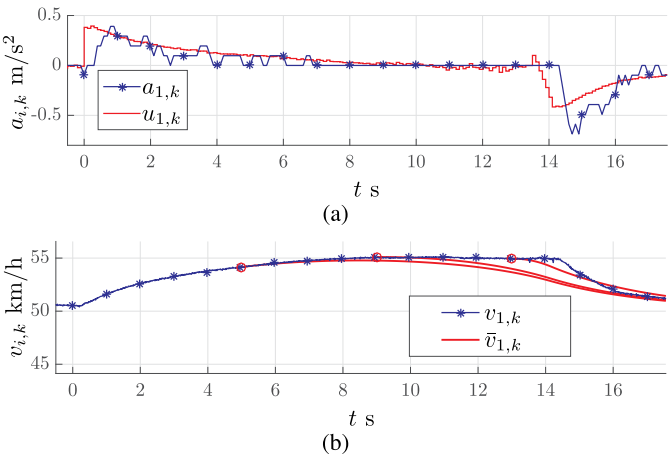


Fig. 11. Illustration of the model-plant mismatch. (a) Demanded, $u_{i,k}$, and actual acceleration, $a_{i,k}$, where the difference between the two highlights the nonmodeled dynamics. (b) Corresponding velocity profile, where the prediction of $\bar{v}_{i,k}$ at $t = 5, 9, 13$ is displayed.

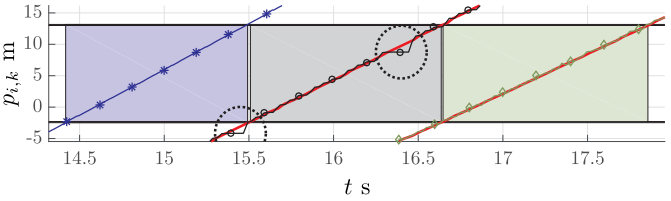


Fig. 12. Position trajectories of one experimental run illustrating the positioning issues in vehicles 2 (black line) and 3 (green line). Vehicle 2 suffered recurring large position errors, which are encircled here. For reference, more accurate position trajectories are given as red lines, which have been reconstructed off-line based on raw measurements acquired during the experimental run.

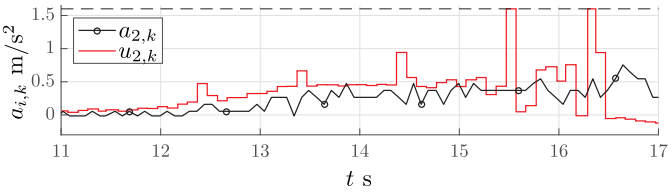


Fig. 13. Desired and actual acceleration of vehicle 2 corresponding to the position trajectories in Fig. 12. The effects of the faulty position estimates are visible as spikes in the demanded acceleration between $t = 12$ s and $t = 17$ s.

In particular, the red square in the cutout of Fig. 10 illustrates the results that would have been obtained if the intersection was 1.4 m shorter on all roads, while the vehicles were controlled using (16) with $\Delta = 0.7$ m. As can be seen in Fig. 10, the smaller intersection would, in this case, have been completely mutually exclusive for all experimental runs, and the closed-loop system would have satisfied all position constraints (8e) and (8f).

5) Causes of Constraint Violation: Several factors most likely contributed to the small constraint violations observed during the experimental campaign. For instance, the simplistic prediction model used failed to capture important aspects of the vehicles' behavior, causing the predicted and actual state evolution to differ. In particular, the actual actuator dynamics contained several unknown nonlinearities due to, e.g., turbo-lag, gear shifts, and the vehicle interface logics, and exhibited different dynamical behavior in acceleration and braking. This is clearly seen in Fig. 11(a), which displays the commanded, $u_{i,k}$, and actual, $a_{i,k}$, acceleration for a successful run. Note, in particular, the delayed first-order behavior between $t = 0$ s and $t = 1$ s, the dead-zone between $t = 7$ s and $t = 13.5$ s, and the delayed but significantly faster dynamics at $t = 14$ s resulting from the application of the friction brakes. The corresponding velocity profile is given in Fig. 11(b), where an open-loop prediction is shown for comparison. The effects of the nonmodeled dynamics are clearly visible in the difference between the open- and closed-loop trajectories. Moreover, the upper bound on admissible actuation commands, $u_1^{\max}$, was erroneously identified for vehicle 1 and the vehicle acceleration saturated at 1 m/s^2 rather than 1.6 m/s^2, which was used in (8). Due to these model-plant mismatches, the nominal prediction could satisfy the position constraints (8e) and (8f), while the actual system would cause constraint violations.

Another explanation to the occurrence of constraint violations lies in the accuracy of the positioning of vehicles 2 and 3, which was poor at times. This is illustrated in Fig. 12, where the position estimate used by the controller is overlayed on an off-line, reconstruction of the position with higher accuracy. The better positioning of vehicle 1 is noticeable in the constraint violation statistics; in 90% of the evaluations of the positioning constraints (8e) and (8f) for vehicle 1, the violations were smaller than 0.18 m, compared to 0.38 and 0.41 m for vehicles 2 and 3. The issues were most pronounced in vehicle 2, where measurement errors on the order of meters

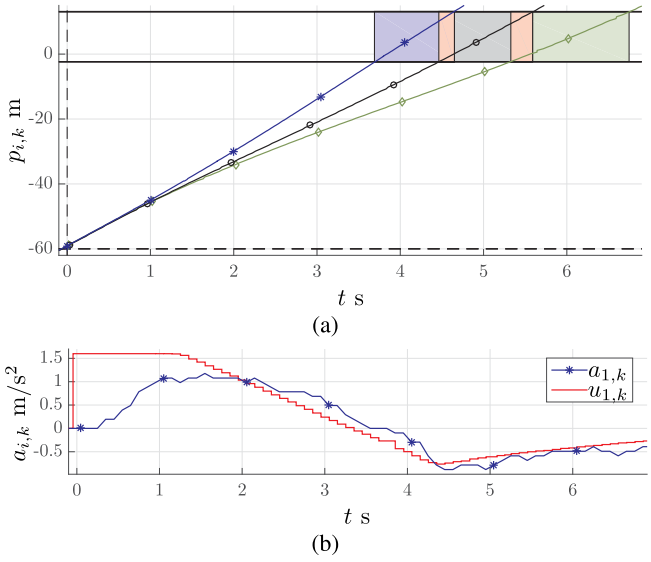


Fig. 14. Illustration of controller behavior for vehicle 1 with a penalty function, where $\nabla\rho_i(\sigma i)$ is too small. (a) Distance to intersection; The light-red colored blocks illustrate the times during which more than one vehicle is inside the intersection (i.e., where the collision avoidance constraints are violated). (b) Acceleration.

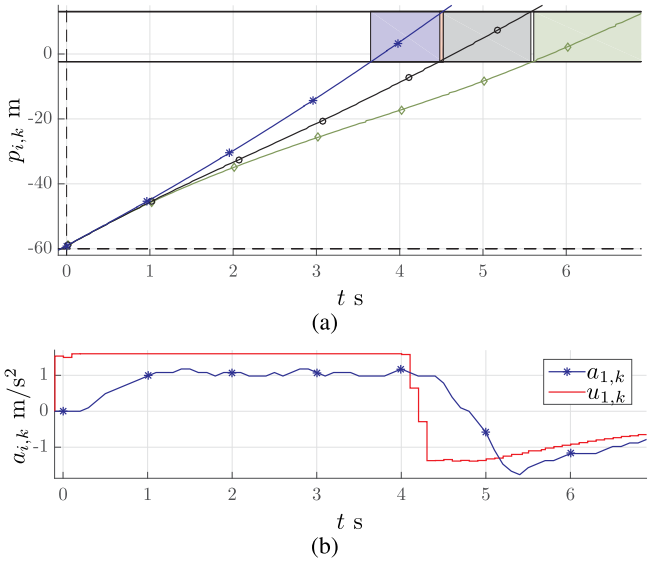


Fig. 15. Illustration of controller behavior for vehicle 1 with a penalty function, where $\nabla\rho_i(\sigma_i)$ is large enough. (a) Distance to intersection. (b) Acceleration. Note the discrepancy between the actual and modeled input saturation level $u_1^{\max}$ in (b), as discussed in Section V-B3.

occurred with a frequency of 1 Hz, which is visible around times $t = 15.5$ s and $t = 16.5$ s in Fig. 12. The error occurred due to faulty parsing of the data supplied by the GPS receiver, which caused the same position measurements to be used more than once in the update of the EKF. Due to this, the vehicle temporarily appeared stationary.

6) Constraint Violation Minimization: Due to the large positioning errors in vehicle 2, the control commands were occasionally computed based on positions that were further away from the intersection than the actual system state. To ensure satisfaction of the position constraints (8e) and (8f), the controller attempted to compensate the faulty position by commanding higher acceleration from the vehicle. The behavior can be seen in Fig. 13, where spikes in the demanded acceleration are present with frequency of 1 Hz, corresponding to the occurrence of the large positioning errors. Moreover, the amplitude of the spikes increases as the distance to the intersection decreases since the control required to counteract the perturbations has to performed over fewer and fewer control stages. This exemplifies the reduction of the control authority discussed in Section IV.

In particular, at $t = 15.47$ s, a control command is computed based on the position $p_{2,k} = -4.168$ m and velocity $v_{2,k} = 48.79$ km/h, while the vehicle is required to have passed $p_2^{\text{in}} = -2.314$ m at $t_2^{\text{in}} = 15.484$ s. The acceleration required to satisfy the position constraints (8e) far exceeds the upper bound on the input, and the MPC problem (8) is therefore infeasible. The situation thus constitutes an example of the infeasibility issues discussed in Section IV-A and illustrated in Fig. 3. However, since the relaxed formulation of the vehicle-level MPC (13) was used, the infeasibility issue is avoided. Instead, Fig. 13 demonstrates the constraint-violation minimization mode of the controller, which was discussed in Section IV-A; that is, while no input exists such that the position constraints (8e) and (8f) are satisfied, the controller saturates the input in order to make the constraint violation as small as possible. As discussed in Section IV-A, the controller minimizes the constraint violation, provided that $\nabla\rho_i(\sigma_i) = \phi_i\mathbf{1} + \phi_i^q\sigma_i$ [see (18)] is chosen large enough. To illustrate the effect on the controller performance, we provide results from a run, where ϕ_i was set too small ($\phi_i = 100$), and from a run, where ϕ_i was adequately chosen ($\phi_i = 1000$) in Figs. 14 and 15, respectively. In both cases, the bilevel controller is applied from the same starting configuration, which is such that significant control effort is needed to ensure collision avoidance. As illustrated in Fig. 14(b), with too small ϕ_i, the controller does not utilize the full control authority to avoid collisions. With ϕ_i chosen large enough, the input is instead saturated, as shown in Fig. 15(b). As a result, the closed-loop controller with small ϕ_i causes significant violations of the collision avoidance constraints, as illustrated in Fig. 14(a), while an adequately chosen ϕ_i results in only minor constraint violations, as shown in Fig. 15(a).

VI. Conclusion and Discussion

In this paper, we first proposed, analyzed, and established the properties of a bilevel, closed-loop controller for coordination of automated vehicles at intersections. In particular, we proved that the bilevel controller is nominally stable and persistently feasible under mild assumptions and showed that the controller can be modified to ensure robust persistent feasibility and collision avoidance. We thereafter demonstrated the applicability of the controller through experimental validation. In particular, we showed that the controller performance was consistent, it managed to counteract both large and small perturbations, and the violations of the collision avoidance constraints were small. We also discussed the causes of the constraint violations and illustrated how collision avoidance could be guaranteed by use of constraint tightening.

A. Comments on the Experimental Results

The precision observed in the enforcement of the position constraints (8e) and (8f) during the experimental campaign was surprising, considering the simplicity of the prediction model and the measurement noise. Most likely, inclusion of actuator dynamics in the prediction model would have yielded even higher precision, as would more accurate positioning. While access to accurate positioning systems is a strong assumption outside the experimental domain, we emphasize that the controller presented in this paper could handle large uncertainties by appropriately choosing the constraint tightening parameter Δ_i. Most likely, a practical implementation of the controller would even utilize margins well beyond the constraint satisfaction accuracy, if nothing else than for the confidence of the passengers. Furthermore, we want to emphasize that although the intersection-level control loop exhibited large delays in its reaction to perturbations, this was largely a consequence of implementation specifics that could be improved. For instance, while Algorithm 1 usually required 1–1.5 s to converge, well below 0.1 s was spent on actual computations. The remaining time was largely spent in various waiting or idling states and, to a small extent, on communication. As a final comment on the experimental results, we note that hardware availability limited the number of vehicles to 3. The interested reader can find simulations, including more cars in [7], [8], or [35].

B. Outlook and Future Research

As opposed to many other intersection coordination schemes, the controller presented in this paper does not depend on a specific dynamic equation [see (8c)], specific constraints [see (8d)], nor a specific objective function [see (8a)]. The coordination scheme therefore allows a designer to directly incorporate complex models of vehicle dynamics and select an objective to *a priori* specify the desired vehicle behavior. Due to this flexibility, the controller presented in this paper could be used as a component in a future intelligent transportation system, where a higher level traffic flow controller adjusts the objective of the coordination online, based on the current conditions. This could, for instance, be a prioritization of intersection throughput during rush hour traffic and an emphasis on energy efficiency in low-intensity situations. However, the stability properties discussed in this paper are relevant for batch-type problems, where no new vehicles arrive at the intersection as the system evolves. With continuously oncoming vehicles, additional notions of stability and feasibility become important. For instance, it is well known in the field of traffic flow control that current intersections have limits on the rate of cars that can be served without the formation of unstable queues. Such limits will exist also when the intersections are controlled by the approach proposed in this paper, and a future direction of research is to study the behavior of the proposed controller on a larger scale with continuously oncoming cars. In particular, we plan to conduct a comparative study between the proposed controller and both current regulatory mechanisms (i.e., traffic lights) and other coordination schemes in this setting. While the proposed controller is optimal in the performance metric (8a) by construction, a quantification of the performance advantage is necessary. Moreover, while the coordination controller presented in this paper does not rely on a specific objective function, Assumption 3 imposes some restrictions on its characteristics. It is our intention to incorporate more general, economic objectives to the coordination controller in the future work, and we have presented an initial study in [36]. Due to the safety-critical reliance on V2V communication, a through study of the impact on the controller by lossy or intermittent communication is also necessary. Communication deficiencies will, in practice, limit the sampling frequency of the intersection-level control loop and, in case of distributed solution of (9), also how the problem is solved. Therefore, an adaptation of the algorithm that explicitly accounts for limited-capacity communication channels is highly desirable.

Finally, while we did not discuss algorithms for the solution of (9) in depth in this paper, we are currently developing a distributed primal-dual interior point method tailored for the problem with a fixed order S. It is our intention to use this as the basis of a distributed mixed-integer nonlinear program (NLP) solver to address (9), including finding the order S.

APPENDIX

A. Proof of Proposition 5

A timeslot schedule $\boldsymbol{T}_0 = (T_{0,1}, \ldots, T_{0,N_a})$, which is feasible in (9) for the state X_k, exists by assumption. We label Problem (8) at t_k with parameters $(x_{i,k}, T_{0,i})$ as $\mathbb{P}_{i,k}$ and similarly label Problem (8) at t_{k+1} with parameters $(x_{i,k+1}, T_{0,i})$ as $\mathbb{P}_{i,k+1}$. By assumption, $\mathbb{P}_{i,k}$ is feasible, and we denote its solution $w^*_{i,k} = (x_{i,k}, \bar{u}^*_{i,k}, \bar{x}^*_{i,k+1}, \ldots, \bar{x}^*_{i,k+N_i})$. Since we are considering the nominal case, we have that $x_{i,k+1} = \bar{x}^*_{i,k+1}$ and need to establish that $\mathbb{P}_{i,k+1}$ is feasible. Assumption 3 gives that $w_{i,k+1} = (x_{i,k+1}, \bar{u}^*_{i,k+1}, \ldots, \bar{x}^*_{i,k+N}, \kappa^f(\bar{x}_{i,k+N}), f(\bar{x}_{i,k+N}, \kappa^f_i(\bar{x}_{i,k+N})))$ satisfies (8b)–(8d) in $\mathbb{P}_{k+1}$. Furthermore, in $\mathbb{P}_{i,k}$, (8e) constrains the predicted state and controls n_i^{in} time steps into the future. In $\mathbb{P}_{i,k+1}$, on the other hand, (8e) constrains the predicted state and controls $n_i^{\text{in}} - 1$ time steps into the future. However, since the $n_i^{\text{in}} - 1$:th entry of $w_{i,k+1}$ is identical to the n_i^{in}:th entry of $w^*_{i,k}$, $w_{i,k+1}$ satisfies (8e) in $\mathbb{P}_{i,k+1}$. The same applies for (8f), and consequently, $w_{i,k+1}$ is feasible in $\mathbb{P}_{i,k+1}$. Since $\mathbb{P}_{k+1}$ is feasible $\forall i$, (9) has at least the solution $\boldsymbol{T}_0$ at t_{k+1}. We note that by construction, a feasible solution exists to (8) with $(x_{i,k+1}, T^*_i(X_{k+1}))$, recalling that $\boldsymbol{T}^*(X_{k+1})$ is the optimal solution to (9) for X_{k+1}.

In summary, if (9) has a solution for X_k at t_k, (8) is feasible $\forall i$ at t_k. Due to this, (9) also has a solution at X_{k+1}, whereby (8) are feasible $\forall i$ at t_{k+1}. The result for $k+n, n > 1$ and persistent feasibility follows.

B. Proof of Theorem 6

To prove the nominal stability of the bilevel controller, we first establish the stability of the receding horizon controller for a single vehicle and a fixed timeslot T_i.

Proposition 11 (Stability of One Vehicle Under a Fixed Timeslot): Suppose that Assumptions 3 and 4 hold and that

a feasible solution to (8) exists for $(x_{i,k}, T_i)$. Then, the MPC based on (8) stabilizes the vehicle state to $\mathcal{Q}_i = \mathcal{N}_i \cap \mathcal{P}_i$.

Proof: Consider the candidate Lyapunov function

$$\bar{V}_{i,k}(x_{i,k}, T_i) = V_i(x_{i,k}, T_i) + \alpha_{1,i}\big(\max\big(0, p_i^{\text{out}} - p_{i,k}\big)\big). \quad (22)$$

Since the feasibility set of (8) is a subset of the feasibility set of (11), we obtain

$$V_i(x_{i,k}, T_i) \geq V_i^u(x_{i,k}) \geq \alpha_{1,i}(|x_{i,k}|_{\mathcal{N}_i}) \quad (23)$$

where the last inequality is due to Assumption 3, which implies that $V_i^u(\cdot)$ is a Lyapunov function for the closed-loop system [30]. Then, there exists a $\mathcal{K}_\infty$-function $\bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) \leq \bar{V}_i(x_{i,k}, T_i)$

$$\bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) := \alpha_{1,i}\left(\frac{1}{2}|x_{i,k}|_{\mathcal{Q}_i}\right) \quad (24a)$$

$$\leq \alpha_{1,i}\left(\frac{1}{2}\big(|x_{i,k}|_{\mathcal{N}_i} + \max\big(0, p_i^{\text{out}} - p_{i,k}\big)\big)\right) \quad (24b)$$

$$\leq \alpha_{1,i}\big(|x_{i,k}|_{\mathcal{N}_i}\big) + \alpha_{1,i}\big(\max\big(0, p_i^{\text{out}} - p_{i,k}\big)\big) \quad (24c)$$

$$\leq \bar{V}_i(x_{i,k}, T_i) \quad (24d)$$

where (24c) follows from the property $\alpha(a+b) \leq \alpha(2a) + \alpha(2b)$ of $\mathcal{K}_\infty$-functions, and (24b) is due to $|x|_{\mathcal{N}_i} + \max(0, p_i^{\text{out}} - p_{i,k}) = |x|_{\mathcal{N}_i} + |x|_{\mathcal{P}_i} \geq |x|_{\mathcal{Q}_i}$.[3] We now prove the existence of a $\mathcal{K}_\infty$-function $\bar{\alpha}_{2,i}(|x_{i,k}|_{\mathcal{Q}_i}) \geq \bar{V}_i(x_{i,k}, T_i)$ by showing that $\bar{V}_i(x_{i,k}, T_i) = 0$ for $x_{i,k} \in \mathcal{Q}_i$ and that $\bar{V}_i(x_{i,k}, T_i)$ is continuous on $\partial\mathcal{Q}_i$. First, we note that $V_i^u(x_{i,k}) = V_i(x_{i,k}, T_i)$ for $x_{i,k} \in \mathcal{P}_i$. Due to Assumption 3, $V_i^u(x_{i,k}) = 0, \forall x_{i,k} \in \mathcal{N}_i$, which gives that $\bar{V}(x_{i,k}, T_i) = V_i(x_{i,k}, T_i) = 0, \forall x_{i,k} \in \mathcal{Q}_i$. Since $\alpha_{1,i}(\cdot)$ and $\max(\cdot,\cdot)$ are continuous, continuity of $\bar{V}(x_{i,k}, T_i)$ is inherited from $V(x_{i,k}, T_i)$. There are two cases of interest when $x_{i,k} \in \partial\mathcal{Q}_i$: 1) when $x_{i,k}$ is in the interior of $\mathcal{P}_i$ and 2) when $x_{i,k} \in \partial\mathcal{P}_i$. For case 1, we note that $\bar{V}_i(x_{i,k}, T_i) = V_i(x_{i,k}, T_i) = V_i^u(x_{i,k})$, and that continuity follows from Assumption 3. For case 2, (8e) is removed, and (8f) holds with equality at the initial state, i.e., $p_{i,k} = \bar{p}_{i,k} = p_i^{\text{out}}$. In directions of increasing $p_{i,k}$, i.e., to the interior of $\mathcal{P}_i$, t_i^{out} and (8f) are removed from the problem. Continuity of $V_i(x_{i,k}, T_i)$ in this direction follows from $V_i(x_{i,k}, T_i) = V_i(x_{i,k})$ on $\partial\mathcal{P}_i$ and continuity of $V_i(x_{i,k})$ is due to Assumption 3. We now turn to directions, where $p_{i,k} < p_i^{\text{out}}$ and t_i^{out} are such that (8) is feasible. Standard results from parametric programing establish that the optimal value function of a parametric NLP is continuous in the problem parameters if the Mangasarian–Fromovitz Constraint Qualification (MFCQ) holds [37, Th. 4.2]. Unfortunately, MFCQ fails at $\partial\mathcal{P}_i$, i.e., continuity does not directly follow from the formulation (8). However, MFCQ does hold if the position component of the initial condition constraint (8b) is moved to the objective using an exact penalty function, and its optimal value function is continuous in $(p_{i,k}, t_i^{\text{out}})$. By Theorem 12, this relaxed formulation is equivalent to (8), and therefore, the continuity of $V_i(x_{i,k}, T_i)$ follows. Finally, we prove the decrease of $\bar{V}_i(x_{i,k}, T_i)$ along the closed-loop system trajectories, which follows from the standard arguments in MPC, found in, e.g., [30]. We denote the solution to (8) at t_k as $w_i^* := (x_{i,0}^*, \ldots, x_{i,N_i}^*, u_{i,0}^*, \ldots, u_{i,N_i-1}^*)$. Since $x_{i,k+1} = x_{i,1}^*$ and Proposition 5 states that closed-loop system is persistently feasible, we have that

$$\begin{aligned} V_i(x_{i,k+1}, T_i) \leq\ & V_i(x_{i,k}, T_i) - \ell_i\big(x_{i,k}, u_{i,0}^*\big) \\ & - V_i^f\big(x_{i,N_i}^*\big) + \ell_i\big(x_{i,N}^*, \kappa_i^f\big(x_{i,N_i}^*\big)\big) \\ & + V_i^f\big(f_i\big(x_{i,N_i}^*, \kappa_i^f\big(x_{i,N_i}^*\big)\big)\big). \end{aligned} \quad (25)$$

By Assumption 3, this gives

$$V_i(x_{i,k+1}, T_i) - V_i(x_{i,k}, T_i) \leq \ell_i\big(x_{i,k}, u_{i,0}^*\big). \quad (26)$$

By construction, $p_{i,k}$ is nondecreasing in k, and therefore, $\alpha_{1,i}(\max(0, p_i^{\text{out}} - p_{i,k}))$ is nonincreasing along the closed-loop system's trajectories. Consequently, $\bar{V}_i(x_{i,k}, T_i)$ is a Lyapunov function for the closed-loop system, and $|x_{i,k}|_{\mathcal{Q}_i} \to 0$ as $k \to \infty$. □

We now provide the proof for Theorem 6.

Proof of Theorem 6: Consider the Lyapunov function candidate

$$\bar{V}(X_k) = \sum_{i=1}^{N} \bar{V}_i(x_{i,k}, T_i). \quad (27)$$

First, by Proposition 11, the terms in (27) are such that $\bar{V}(X_k) = 0, \forall X_k \in \mathcal{Q}$ and such that $\bar{V}(X_k)$ is continuous on $\partial\mathcal{Q}$. Consequently, $\exists$ $\mathcal{K}_\infty$-function $\tilde{\alpha}_{2,i}(|X_k|_{\mathcal{Q}}) \geq \bar{V}(X_k)$. Second, $\exists\mathcal{K}_\infty$-function $\tilde{\alpha}$ for all feasible $\boldsymbol{T}$ such that

$$\sum_{i=1}^{N_a} \bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) \geq \tilde{\alpha}\left(\sum_{i=1}^{N_a} |x_{i,k}|_{\mathcal{Q}_i}\right) \geq \tilde{\alpha}(|X_k|_{\mathcal{Q}}) \quad (28)$$

and since

$$\bar{V}(X_k) = \sum_{i=1}^{N} \bar{V}_i(x_{i,k}, T_i) \geq \sum_{i=1}^{N_a} \bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) \quad (29)$$

we have that $\bar{V}(X_k) \geq \tilde{\alpha}(|X_k|_{\mathcal{Q}})$. Finally, we note that for a fixed timeslot schedule $\boldsymbol{T}$, Proposition 11 ensures the decrease of $\bar{V}(X_k)$ since $\forall i$, $\bar{V}_i(x_{i,k}, T_i)$ decreases. Denoting the solution to (9) at X_k as $T_{i,k}^*, i \in \mathbb{I}_{[1,N_a]}$, we have by optimality that

$$\sum_{i=1}^{N_a} V_i\big(x_{i,k+1}, T_{i,k+1}^*\big) \leq \sum_{i=1}^{N_a} V_i(x_{i,k+1}, T_i) < \sum_{i=1}^{N_a} V_i(x_{i,k}, T_i). \quad (30)$$

Since this holds for all feasible T_i, then $\sum_{i=1}^{N_a} V_i(x_{i,k+1}, T_{i,k+1}^*) < \sum_{i=1}^{N_a} V_i(x_{i,k}, T_{i,k}^*)$. Finally, since $\forall i$, $\alpha_{1,i}(\max(0, p_i^{\text{out}} - p_{i,k}))$ is nonincreasing, $\bar{V}(X_k)$ must be decreasing also when the intersection-level loop is closed, and $\boldsymbol{T}$ is recomputed. Consequently, $\bar{V}$ is a Lyapunov function for the bilevel closed-loop system, and $|X_k|_{\mathcal{Q}} \to 0$ as $k \to \infty$. Besides the case where $\boldsymbol{T}$ and S are optimized, the decrease condition (30) holds also when $\boldsymbol{T}$ is optimized and S is fixed when S is optimized and/or $\boldsymbol{T}$ is updated more rarely than t_s, as well as when both $\boldsymbol{T}$ and S are fixed. Consequently, the bilevel controller is stabilizing under these conditions as well. □

[3] Note that a triangle inequality holds for $|a|_{\mathcal{A}}$; if $\mathcal{A} = \mathcal{B} \times \mathcal{C}$, where $\times$ is the Cartesian product, we have that $|a|_{\mathcal{B}} + |a|_{\mathcal{C}} \geq |a|_{\mathcal{A}}$.

Note that if S is updated using an external heuristic, the bilevel controller is stabilizing if for S, $\boldsymbol{T}$ can be found, which ensures (30). This is particularly important as the solution of the combinatorial problem (9) might be prohibitively hard and reliance on heuristics for closed-loop control might be necessary.

C. Proof of Proposition 7

Consider the NLP

$$\min_{w\in\mathcal{W}} \quad f(w) \tag{31a}$$
$$\text{s.t. } h(w) \geq 0 \tag{31b}$$

and the relaxed NLP

$$\min_{w\in\mathcal{W},\sigma} \quad f(w) + p(\sigma) \tag{32a}$$
$$\text{s.t. } h(w) + \sigma \geq 0 \tag{32b}$$
$$\sigma \geq 0 \tag{32c}$$

where $\mathcal{W} = \{w \mid g(w) \geq 0,\ g_e(w) = 0\}$. We denote $\lambda \geq 0$ as the Lagrange multiplier associated with constraint (31b) and let $\lambda_{\max}$ be the largest value it can attain. In the following, we are interested in analyzing the properties of NLP (32) in the two cases, in which NLP (31) is feasible and infeasible. For the former case, the following holds.

Theorem 12: Provided that $(\partial/\partial s)p(0) \geq \lambda_{\max}$ and NLP (31) has a solution, the solution of NLP (32) yields $\sigma = 0$ and $w = w^*$, which is also optimal for NLP (31).

Proof: A proof of this theorem can be found in, e.g., [31, Th. 14.3.1], but we provide one here for the reader's convenience. We first note that the KKT conditions of (31) and (32), they differ only in that

$$h(w) \geq 0, \quad \lambda_i h_i(w) = 0, \quad \lambda \geq 0 \tag{33}$$

hold at optimality for (31), whereas

$$h(w) + \sigma \geq 0, \quad \lambda_i(h_i(w) + \sigma_i) = 0, \quad \lambda \geq 0 \tag{34a}$$
$$\sigma \geq 0, \quad s_i\mu_i = 0, \quad \mu \geq 0, \tag{34b}$$

and

$$\frac{\partial}{\partial\sigma}p(\sigma) - \lambda - \mu = 0 \tag{34c}$$

hold at optimality for (32), where μ is the Lagrange multiplier of the constraint (32c). A primal-dual solution to (31) and $\sigma = 0$ is, thus, a solution to (32) if $(\partial/\partial s)p(0) \geq \lambda_{\max}$ since

$$\mu = \frac{\partial}{\partial s}p(s) - \lambda \geq 0 \tag{35}$$

implies that (34b) holds. Second-order optimality conditions are then easily proven by noting that any perturbation of the primal solution induces an increase in the cost function. □

To examine the case where (31) is infeasible, we further introduce the infeasibility minimizing NLP

$$\min_{w\in\mathcal{W},\sigma} \quad \|\sigma\|_\infty \tag{36a}$$
$$\text{s.t. } h(w) + \sigma \geq 0 \tag{36b}$$
$$\sigma \geq 0 \tag{36c}$$

and the auxiliary NLP

$$\min_{w\in\mathcal{W},\sigma} \quad f(w) \tag{37a}$$
$$\text{s.t. } h(w) + \sigma^* \geq 0 \tag{37b}$$

where σ^* is a solution to (36). We denote $\nu \geq 0$ as the Lagrange multiplier associated with constraint (37b), and let $\nu_{\max}$ be the maximum value it can attain.

Theorem 13: Provided that $\nabla_\sigma p(\sigma^*) \geq \nu_{\max}$, NLP (32) yields a solution $\sigma = \sigma^*$ and $w = w^*$, which is also optimal for (36).

Proof: We first note that any solution σ to (32) can be written $\sigma = \sigma^* + \sigma$ for some $\sigma \geq 0$. Therefore, (32) is equivalent to

$$\min_{w\in\mathcal{W},\sigma} \quad f(w) + p(\sigma^* + \sigma) \tag{38a}$$
$$\text{s.t. } h(w) + \sigma^* + \sigma \geq 0 \quad \sigma \geq 0. \tag{38b}$$

Since (38) is also a relaxation of (37), Theorem 12 gives that $\sigma = 0$ when $\nabla_\sigma p(\sigma^*) \geq \nu_{\max}$ and a solution to (37) exists. By construction, (36) and, therefore, (37) have a solution when (32) has a solution. Therefore, the solution to (32) is $\sigma = \sigma^*$ and $w = w^*$, which also is a solution to (36). □

ACKNOWLEDGMENT

The authors would like to thank G. Frison for supporting them with HPMPC, A. Severinson for the help with the V2V communication equipment, and the following individuals for the help with the equipment for the experimental validation: A. Karsolia and F. von Corswant at the Chalmers REVERE Lab, W. Mostovski at Halmstad University, H. Lind at Volvo Cars, and A. Colombo at Politechnico di Milano. They would also like to thank Fengco Real Time Control and Leica Geosystems for technical assistance with the hardware.

REFERENCES

[1] M. C. Simon, T. Hermitte, and Y. Page, "Intersection road accident causation: A European view," in *Proc. 21st Int. Tech. Conf. Enhanced Saf. Vehicles*, 2009, pp. 1–10.

[2] M. Li, K. Boriboonsomsin, G. Wu, W. Zhang, and M. J. Barth, "Traffic energy and emission reductions at signalized intersections: A study of the benefits of advanced driver information," *Int. J. Intell. Transp. Syst. Res.*, vol. 7, no. 1, pp. 49–58, Jun. 2009.

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

[4] H. Wymeersch, G. R. de Campos, P. Falcone, L. Svensson, and E. G. Ström, "Challenges for cooperative ITS: Improving road safety through the integration of wireless communications, control, and positioning," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 573–578.

[5] E. Steinmetz, R. Hult, G. R. de Campos, M. Wildemeersch, P. Falcone, and H. Wymeersch, "Communication analysis for centralized intersection crossing coordination," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 813–818.

[6] R. Hult, G. R. de Campos, P. Falcone, and H. Wymeersch, "An approximate solution to the optimal coordination problem for autonomous vehicles at intersections," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2015, pp. 763–768.

[7] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Primal decomposition of the optimal coordination of vehicles at traffic intersections," in *Proc. 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 2567–2573.

[8] M. Zanon, S. Gros, H. Wymeersch, and P. Falcone, "An asynchronous algorithm for optimal vehicle coordination at traffic intersections," in *Proc. 20th IFAC World Congr.*, 2017, pp. 12008–12014.

[9] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.

[10] H. Kowshik, D. Caveney, and P. R. Kumar, "Provable systemwide safety in intelligent intersections," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 804–818, Mar. 2011.

[11] J. Lee and B. Park, "Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 81–90, Mar. 2012.

[12] D. Miculescu and S. Karaman, "Polling-systems-based control of high-performance provably-safe autonomous intersections," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 1417–1423.

[13] J. Gregoire and E. Frazzoli, "Hybrid centralized/distributed autonomous intersection control: Using a job scheduler as a planner and inheriting its efficiency guarantees," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 2549–2554.

[14] P. Tallapragada and J. Cortés, "Coordinated intersection traffic management," *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 233–239, 2015.

[15] A. Colombo and D. Del Vecchio, "Efficient algorithms for collision avoidance at intersections," in *Proc. 15th ACM Int. Conf. Hybrid Syst., Comput. Control*, New York, NY, USA, 2012, pp. 145–154.

[16] G. R. de Campos, P. Falcone, and J. Sjöberg, "Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1456–1461.

[17] G. R. de Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg, "Cooperative receding horizon conflict resolution at traffic intersections," in *Proc. 53rd IEEE Conf. Decis. Control (CDC)*, Dec. 2014, pp. 2932–2937.

[18] A. Katriniok, P. Kleibaum, and M. Joševski, "Distributed model predictive control for intersection automation using a parallelized optimization approach," in *Proc. 20th IFAC World Congr.*, 2017, pp. 5940–5946.

[19] K.-D. Kim and P. R. Kumar, "An MPC-based approach to provable system-wide safety and liveness of autonomous ground traffic," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3341–3356, Dec. 2014.

[20] X. Qian, J. Gregoire, A. De La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *Proc. Eur. Control Conf.*, Jul. 2015, pp. 3452–3458.

[21] M. A. S. Kamal, J.-I. Imura, T. Hayakawa, A. Ohata, and K. Aihara, "A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 3, pp. 1136–1147, Jun. 2015.

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

[23] N. Murgovski, G. R. de Campos, and J. Sjöberg, "Convex modeling of conflict resolution at traffic intersections," in *Proc. 54th IEEE Conf. Decis. Control (CDC)*, Dec. 2015, pp. 4708–4713.

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

[25] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.

[26] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.

[27] M. R. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio, "Cooperative collision avoidance at intersections: Algorithms and experiments," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 3, pp. 1162–1175, Sep. 2013.

[28] V. Milanes, J. Alonso, L. Bouraoui, and J. Ploeg, "Cooperative maneuvering in close environments among cybercars and dual-mode cars," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 15–24, Mar. 2011.

[29] R. Hult *et al.*, "Design and experimental validation of a cooperative driving control architecture for the grand cooperative driving challenge 2016," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 4, pp. 1290–1301, Apr. 2018.

[30] J. B. Rawlings and D. Q. Mayne, *Model Predictive Control Theory and Design*. San Francisco, CA, USA: Nob Hill, 2009.

[31] R. Fletcher, *Practical Methods of Optimization*, 2nd ed. Chichester, U.K: Wiley, 1987.

[32] R. Hult, M. Zanon, S. Gros, and P. Falcone. *Optimal Coordination of Three Cars Approaching an Intersection*. Accessed: Mar. 3, 2017. [Online]. Available: https://youtu.be/nYSXvnaNRK4

[33] *Rendits*. Accessed: Nov. 5, 2016. [Online]. Available: http://www.rendits.com/

[34] G. Frison, H. H. B. Sørensen, B. Dammann, and J. B. Jørgensen, "High-performance small-scale solvers for linear model predictive control," in *Proc. Eur. Control Conf. (ECC)*, 2014, pp. 128–133.

[35] R. Hult, M. Zanon, S. Gros, and P. Falcone, "An MIQP-based heuristic for optimal coordination of vehicles at intersections," in *Proc. 57th Conf. Decis. Control (CDC)*, to be published.

[36] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Energy-optimal coordination of autonomous vehicles at intersections," in *Proc. Eur. Control Conf.*, 2018, pp. 602–607.

[37] A. V. Fiacco and Y. Ishizuka, "Sensitivity and stability analysis for nonlinear programming," *Ann. Oper. Res.*, vol. 27, no. 1, pp. 215–235, Dec. 1990.

Robert Hult received the B.S. degree in mechanical engineering and the M.Sc. degree in systems, control and mechatronics from the Chalmers University of Technology, Gothenburg, Sweden, in 2011 and 2013, respectively, where he is currently pursuing the Ph.D. degree.

His current research interests include distributed and cooperative predictive control, in particular with applications to cooperative vehicles and intelligent transportation systems.

Mario Zanon received the master's degree in mechatronics from the University of Trento, Trento, Italy, and the Diplôme d'Ingénieur degree from the École Centrale Paris, Châtenay-Malabry, France, in 2010, and the Ph.D. degree in electrical engineering from KU Leuven, Leuven, Belgium, in 2015.

He has held research stays at KU Leuven, the University of Bayreuth, Bayreuth, Germany, the Chalmers University of Technology, Gothenburg, Sweden, and the University of Freiburg, Freiburg, Germany. He held a post-doctoral researcher position at Chalmers University of Technology until 2017. He is currently an Assistant Professor with the IMT School for Advanced Studies Lucca, Lucca, Italy. His current research interests include the numerical methods for optimization, economic model predictive control, optimal control, and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.

Sebastien Gros received the Ph.D. degree from the École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, in 2007.

After a journey by bicycle from Switzerland to the Everest Base Camp, Khumjung, Nepal, in full autonomy, he joined the Research and Development Group hosted at Strathclyde University, Glasgow, Scotland, in 2010, where he is focusing on wind turbine control. In 2011, he joined KU Leuven, Leuven, Belgium, where his main research focus was on optimal control and fast nonlinear MPC for complex mechanical systems. In 2013, he joined the Department of Signals and Systems, Chalmers University of Technology, Gothenburg, Sweden, where he is currently an Associate Professor. His current research interests include numerical methods, real-time optimal control, and the optimal control of complex mechanical systems and energy applications.

Paolo Falcone received the Ph.D. degree in information technology from the University of Sannio, Benevento, Italy, in 2007.

He is currently an Associate Professor with the Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden. His current research interests include constrained optimal control applied to autonomous and semiautonomous mobile systems, cooperative driving, and intelligent vehicles. He is also involved in several projects, in cooperation with industry, focusing on autonomous driving, cooperative driving, and vehicle dynamics control.