Optimal Coordination of Automated Vehicles at Intersections: Theory and Experiments

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

***Abstract*—With the introduction of Cooperative Automated Vehicles, traffic lights can be replaced by coordination algorithms. In this paper, we present a bi-level, model predictive controller for coordination of automated vehicles at intersection. The bi-level controller consists of a coordination level, where intersection occupancy timeslots are allocated, and vehicle-level controllers, where the control commands for the vehicles are computed. We establish persistent feasibility and stability of the bi-level 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 Programming (SQP) 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 proposed controller, and validates that safety can be ensured for large positioning uncertainties.**

***Index Terms*—Intersection Coordination, Networked Mobile Systems, Model Predictive Control, Distributed Optimization**

I. Introduction

The recent years have seen a rapid development in the field of automated driving (AD), 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 passenger safety 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, such as intersections, roundabouts and highway on-ramps 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 EU [31] and similar numbers have been found in the US.

The risks associated with these scenarios have required a strict regulation of the involved vehicles by means of traffic lights, signs and right-of-way rules. In part due to the regulation, intersections, roundabouts and on-ramps form bottlenecks in the traffic system and are commonly the cause of congested traffic. As a consequence, these locations cause unnecessary energy waste (e.g., through deceleration/acceleration and idling [23]) and improvements to traffic flow typically requires an expansion of the infrastructure.

In these scenarios the introduction of CAVs enables a potential remedy: instead of using the current regulation, the CAVs could be controlled using automated coordination algorithms, which would guarantee collision avoidance, increase energy efficiency and optimize the traffic flow. The vision is one where the CAVs travel through the intersections, roundabouts and on-ramps at normal speeds in tightly packed, interlocking streams. Such behavior requires a coordination algorithm to act on the individual vehicles, as opposed to the macro perspective found in schemes focused on traffic flow control which utilizes, e.g., adaptive traffic signal timings.

The problem of coordinating CAVs through intersections, roundabouts and on-ramps poses a number of challenges [14], [34]. For instance, a coordination algorithm must rely on potentially lossy wireless communication [32], utilize imperfect measurements (in particular position) and handle various forms of perturbations at the vehicle level. Such an algorithm must therefore be executed in closed-loop, and the control of the vehicles continuously adjusted to incorporate up-to-date information about the vehicle states and the surroundings.

In this paper, we utilize an optimal control formulation of the coordination problem at intersections, first presented in [13], and propose its application to closed-loop, receding horizon control. In particular, we propose a bi-level controller consisting of: 1) An intersection-level control loop which allocates and updates optimal and collision-free timeslots during which each vehicle is allowed to be in the intersection, and 2) lower level control loops which provide optimal actuation commands for all vehicles given the allocated timeslots. We provide a study of the nominal closed-loop system, and present conditions under which it is persistently feasible and stable. We also propose a modification of the controller which allows it to retain persistent feasibility in the presence of bounded perturbations, and present conditions under which robust collision avoidance can be ensured. Furthermore, we describe an implementation of the proposed controller on an experimental test bed consisting of three communicating, automated vehicles. The implementation uses the Sequential Quadratic Programming (SQP) algorithm proposed in [17] and [35], which is solved in a semi-distributed 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.

*Corresponding author. The authors are with the department of Electrical Engineering at Chalmers University of Technology, SE-412 96, Gothenburg, Sweden. e-mail: {hultr, mzanon, grosse, paolo.falcone}@chalmers.se

A. Contributions

The contributions of this paper are as follows: 1) A formulation of bi-level Model Predictive Controller (MPC) for intersection coordination. 2) An analysis of the nominal close-loop systems properties. 3) A modification to the controller which allows robustness against perturbations and enables practical implementations. 4) A description of a practical implementation of the bi-level controller using distributed computations. 5) An experimental validation of the algorithm and an analysis of experimental results.

Although experimental results on intersection coordination and collision avoidance algorithms have been reported before (see, e.g., [12], [25] or [15]), this is to the best of our knowledge the first time an MPC-based coordination scheme has been validated on real vehicles.

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 [17] and [35].

B. 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 [6], [7] a system was presented where the oncoming vehicles request a reservation of timeslots, during 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 which case the vehicle slows down and sends new reservation requests. A different and provably safe timeslot-reservation based IM was proposed in [21], where the timeslot reservations are improved incrementally as the system evolves with the purpose of minimizing travel time. In [22], 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 [24] an algorithm is proposed where a central entity first decides the crossing order and timeslots of the vehicles using results from polling system theory, an thereafter computes their respective control commands. The authors of [11] proposes 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 [33] that the possibility of efficient coordination diminishes 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 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 [3], 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 Optimal Control (OC) and Model Predictive Control (MPC) as frameworks in which the coordination problem can be addressed. For instance, in [4] an optimal control 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 which 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 [5]. Another distributed MPC scheme was proposed in [19], 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 optimal control problem where the previously predicted positions of higher priority vehicles are used as constraints to ensure safety. Similarly, in [20], the authors 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. The authors of [27] propose an MPC scheme which assumes a given crossing order and include constraints which restrict the control inputs of each vehicle to lie in an order-preserving set of inputs. This is shown to guarantee nominal collision avoidance. A MPC-scheme for intersection crossing is proposed in [18] which includes risk minimization objective function of the problem. In particular, the risk minimization leads to solutions where vehicles cross the intersection with large margins w.r.t. collisions when possible, and small margins only when necessary. A continuous time optimal controller for intersection coordination based on indirect methods is proposed in [36], where the crossing order is assumed given by a First-In-First-Out (FIFO) heuristic. Other related works include [26] where the problem is posed using a spatial rather than temporal OC formulation and [30] which uses a robust MPC formulation of the problem in the context of collision avoidance between automated and non-automated 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 [2],[29] and the references therein.

C. Outline

The remainder of the paper is organized as follows: In Section II we state the intersection coordination optimal control 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 nominal controller is both persistently feasible and stabilizes

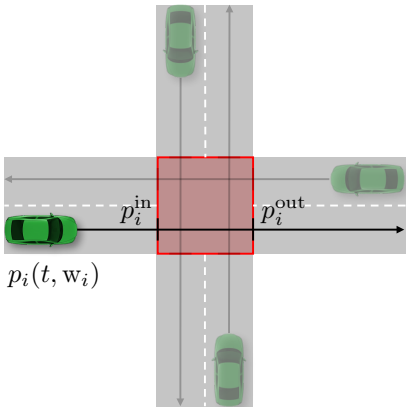


Fig. 1: Schematic illustration of the scenarios considered in this paper. The arrows shows the fixed paths of the vehicles, and the red square illustrates the zone inside the intersection where collisions can occur.

the system. In Section IV we present a modification to the controller that 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, the paper is concluded in Section VI.

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 non-cooperative entities (vehicles, pedestrians etc.) are present.

A. System model

We assume that the vehicles move along predefined and fixed paths, and model the motion of the vehicles along their paths in discrete time as

$$x_{i,k+1} = f_i(x_{i,k}, u_{i,k}, t_\text{s}), \tag{1}$$

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 $t_k = kt_\text{s}$, respectively, and t_s is the sampling time. The state update function $f_i(x_{i,k}, u_{i,k}, t_\text{s})$ is the numerical integration of the continuous time dynamics $\dot{x}_i^\text{c}(t) = f_i^\text{c}(x_i^\text{c}(t), u_i^\text{c}(t))$ over t_s, using $u_i^\text{c} = u_{i,k}$ and starting from $x_{i,k}$. Without loss of generality, we assume that the position $p_{i,k}$ of the vehicle along its path is a state, so that $x_{i,k} = (p_{i,k}, z_{i,k})$, where $z_{i,k} \in \mathbb{R}^{n_i-1}$ collects all non-position states. With $\text{w}_i = (x_{i,0}, u_{i,0}, x_{i,1}, u_{i,1}, \ldots)$, we define a continuous time representation of the position using the discrete time state and control as

$$p_i(t, \text{w}_i) := [1, \mathbf{0}^{1\times n_i-1}] f_i(x_{i,k}, u_{i,k}, t - kt_\text{s}),\ k = \lfloor t/t_\text{s} \rfloor. \tag{2}$$

That is, the position at $t \in]kt_\text{s}, (k+1)t_\text{s}[$ is obtained through integration of the continuous time dynamics. 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^in, t^out at which the vehicle enters and exits the intersection through

$$p_i(t_i^\text{in}, \text{w}_i) = p_i^\text{in} \quad \text{and} \quad p_i(t_i^\text{out}, \text{w}_i) = p_i^\text{out}. \tag{3}$$

A sufficient condition for collision avoidance is that

$$t_i^\text{out} \le t_j^\text{in} \quad \text{or} \quad t_j^\text{out} \le t_i^\text{in}, \quad \forall i, j,\ i \neq j. \tag{4}$$

The states and controls are subject to constraints $h(x_{i,k}, u_{i,k}) \le 0$ capturing, e.g., actuator saturation and speed limits. In particular, we assume that (1) and $h(x_{i,k}, u_{i,k}) \le 0$ are such that $p_i(t, \text{w}_i)$ is monotonically increasing in t. This assumption is standard in intersection coordination, see e.g., [3] or [5], and means that no vehicle ever reverses.

We clarify the above definitions by the following example.

Example 1 (Double integrator dynamics:). If the vehicle is modeled as a point on the path coordinate, its position can be described as the double integration of its acceleration. The discrete time representation is

$$x_{i,k+1} = \begin{bmatrix} 1 & t_\text{s} \\ 0 & 1 \end{bmatrix} x_{i,k} + \begin{bmatrix} \frac{1}{2}t_\text{s}^2 \\ t_\text{s} \end{bmatrix} u_{i,k}, \tag{5}$$

where $x_{i,k} = [p_{i,k}, v_{i,k}]^\top$, $v_{i,k}$ being the velocity along the path. In this case, using $k = \lfloor t/t_\text{s} \rfloor$, (2) becomes

$$p_i(t, \text{w}_i) = p_{i,k} + (t - t_k)v_{i,k} + \frac{1}{2}(t - t_k)^2 u_{i,k}. \tag{6}$$

B. Open loop optimal coordination

In this section we introduce the optimal control formulation of the intersection coordination problem, first presented in [13]. Here, and in the remainder of the paper, we use $x_{i,k}$ to denote the actual state of the system at time $t_k = kt_\text{s}$, and let $\bar{x}_{i,k+n}$ be the open loop prediction of the state at time $(k+n)t_\text{s}$ and denote $T_i = (t_i^\text{in}, t_i^\text{out})$ the predicted intersection occupancy timeslot. Additionally, we denote the stacked state of all vehicles, $X_k = (x_{1,k}, \ldots, x_{N_a,k})$, and the stacked timeslots as $\boldsymbol{T} := (T_1, \ldots, T_{N_\text{a}})$.

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 optimal control problem

$$\begin{aligned}
&V_i(x_{i,k}, T_i) := \\
&\min_{w_i} \quad \sum_{n=0}^{N_i-1} \ell_i(\bar{x}_{i,k+n}, \bar{u}_{i,k+n}) + V_i^\text{f}(\bar{x}_{k+N_i}) && \text{(7a)}\\
&\text{s.t.} \quad \bar{x}_{i,k} = x_{i,k} && \text{(7b)}\\
&\qquad \bar{x}_{i,k+n+1} = f(\bar{x}_{i,k+n}, \bar{u}_{i,k+n}), \qquad n \in \mathbb{I}_{[0,N_i-1]}, && \text{(7c)}\\
&\qquad h_i(\bar{x}_{i,k+n}, \bar{u}_{i,k+n}) \le 0, \qquad n \in \mathbb{I}_{[1,N_i]}, && \text{(7d)}\\
&\qquad p_i(t_i^\text{in}, w_i) - p_i^\text{in} \le 0, && \text{(7e)}\\
&\qquad p_i^\text{out} - p_i(t_i^\text{out}, w_i) \le 0. && \text{(7f)}
\end{aligned}$$

where $N_i \in \mathbb{N}$ 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, b.

We let $\mathcal{T}_i(x_{i,k})$ be the set of timeslots T_i for which the parametric optimization problem (7) 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 $\{1, \ldots, N_a\}$ such that vehicle s_i crosses the intersection

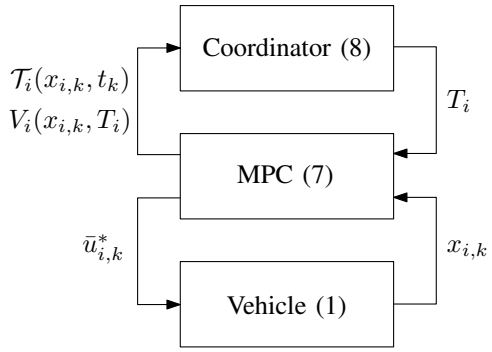


Fig. 2: Schematic illustration of the bi-level control structure for one vehicle. The coordinator is in closed-loop with all vehicles in the same way.

before vehicle s_{i+1}. The optimal collision-free timeslots at t_k are obtained as the solution to

$$V(X_k) := \min_{\boldsymbol{T},S} \quad \sum_{i=1}^{N_\mathrm{a}} V_i(x_{i,k}, T_i) \tag{8a}$$

$$\text{s.t.} \quad T_i \in \mathcal{T}_i(x_{i,k}), \qquad i \in \mathbb{I}_{[1,N_\mathrm{a}]}, \tag{8b}$$

$$t^{\mathrm{out}}_{s_i} \leq t^{\mathrm{in}}_{s_{i+1}} \tag{8c}$$

$$S \in \mathrm{perm}(\mathbb{I}_{[1,N_a]}), \tag{8d}$$

where $\mathrm{perm}(\mathbb{I}_{[1,N_a]})$ denotes the set of all permutations of the index set $\mathbb{I}_{[1,N_a]}$. The timeslot allocation problem (8) 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 (8d), the problem (8) is combinatorial.

C. Receding horizon optimal coordination

For closed-loop control, we apply (7), (8) in a receding horizon fashion. Consequently, at each time instant k, the optimization problems (7) and (8) 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 bi-level 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 (8) at X_k as $\boldsymbol{T}^*(X_k)$, the closed-loop dynamics for vehicle i is

$$x_{i,k+1} = f_i\left(x_{i,k}, \bar{u}^*_{i,0}\left(x_{i,k}, T^*_i\left(X_k\right)\right)\right). \tag{9}$$

In the following, we refer to the receding horizon application of (7) as the *vehicle-level* control-loop, differentiating it from the *intersection-level* control-loop, consisting of the calculation of $\boldsymbol{T}^*\left(X_k\right)$ through (8).

In the next section, we study the properties of the nominal bi-level controller and establish persistent feasibility and stability for the closed loop system (9).

As the closed-loop system evolves all vehicles will eventually first pass p^{in}_i and thereafter p^{out}_i. To enable (7) to remain feasible, we adopt the convention that if $x_{i,k}$ is such that the vehicle has passed p^{in}_i or p^{out}_i, the corresponding constraints (7e) and (7e) are removed from (7).

III. Closed Loop Stability and Persistent Feasiblity

We first introduce the optimal control problem

$$V^u_i(x_{i,k}) := \min_{w_i} \ \text{(7a)}, \quad \text{s.t. (7b)} - \text{(7d)}, \tag{10}$$

which is the vehicle-level optimization problem (7) without the position constraints (7e) and (7f). This corresponds to the optimal uncoordinated case, where collisions are not explicitly avoided. Furthermore, we denote $\mathcal{N}_i := \{x_{i,k} \mid z_{i,k} = z^{\mathrm{ref}}_i\}$, where z^{ref}_i is a reference such that $\mathcal{N}_i \subset \mathcal{H}_i$ and $\mathcal{H}_i := \{x_{i,k} \mid \exists u_{i,k} : h_i(x_{i,k}, u_{i,k}) \leq 0\}$. Additionally, we define the distance from a point a to a set $\mathcal{A}$ as

$$|a|_{\mathcal{A}} = \min_{b \in \mathcal{A}} \ ||a - b||. \tag{11}$$

Finally, we make the following assumptions:

Assumption 2 (Stability and persistent feasibility of vehicle-level MPC)**.** *The vehicle-level MPC formulated based on* (10)*, satisfies $\ell_i(x_{i,k}, u_{i,k}) \geq \alpha_{1,i}(|x_{i,k}|_{\mathcal{N}_i})$ for all feasible $x_{i,k}$ and $u_{i,k}$ and $V^f_i(x_{i,k}) \leq \alpha_{2,i}(|x_{i,k}|_{\mathcal{N}_i})$ for $x_{i,k} \in \mathcal{H}_i$, where $\alpha_{1,i}, \alpha_{2,i}$ are $\mathcal{K}_\infty$-functions. Moreover, there exists a controller $\kappa^{\mathrm{f}}(x_{i,N})$ such that $V^f_i(f(x_{i,N}, \kappa^{\mathrm{f}}(x_{i,N}))) - V^f_i(x_{i,N}) \leq -\ell_i(x_{i,N}, \kappa^{\mathrm{f}}(x_{i,N}))$ and $f_i(x_{i,k}, \kappa^{\mathrm{f}}(x_{i,k})) \in \mathcal{H}_i$ for all $x_{i,N} \in \mathcal{H}_i$.*

Assumption 3 (Regularity of vehicle-level optimal control problems)**.** *For all $x_{i,k}$ such that Problem* (10) *is feasible, Linear Independence Constraint Qualifications (LICQ) and the Second Order Sufficient Conditions (SOSC) hold at the solution.*

Consequently, the receding horizon application (10) is persistently feasible and $V^u_i(\cdot)$ is a Lyapunov function for the closed-loop system [28]. Moreover, the MPC problem (10) is well posed.

A. Persistent feasibility

While a receding horizon controller based on (10) is persistently feasible by Assumption 2, this is not necessarily the case for the receding-horizon application of (7). This is the case since the feasibility set of (7) is a subset of the feasibility set of (10) due to the presence of the position constraints (7e) and (7f) in (7). We note in particular that due to (2), (7e) and (7f) are formulated using the state and control at the predicted stage $k^{\mathrm{in}}_i = \lfloor (t^{\mathrm{in}}_i - t_k)/t_\mathrm{s} \rfloor$ and $k^{\mathrm{out}}_i = \lfloor (t^{\mathrm{out}}_i - t_k)/t_\mathrm{s} \rfloor$ respectively. Consequently, as the system evolves and t_k increases, k^{in}_i and k^{out}_i decrease, and the constraints (7e) and (7f) are "shifted" towards the beginning of the prediction horizon.

These issues are addressed in the following proposition.

Proposition 4 (Persistent feasibilty)**.** *Suppose that Assumption 2 holds and that the vehicle states X_0 are such that the timeslot schedule $\boldsymbol{T}_0$ is feasible in* (8)*, then the nominal bi-level closed-loop system is persistently feasible.*

Proof. Since the intersection level problem (8) has a solution $\boldsymbol{T}^*$ for X_0 at t_0, each of the vehicle-level problems (7) are

feasible by construction for parameters $(x_{i,0}, T_i^*)$, with solution $(x_{i,0}, \bar{u}_{i,0}^*, \bar{x}_{i,1}^*, \ldots, \bar{x}_{i,N_i}^*)$. For vehicle i the position constraint (7e) is formulated using $(\bar{x}_{i,k_i^{\text{in}}}, \bar{u}_{i,k_i^{\text{in}}})$, and if T_i^* is held constant, (7e) is formulated using $(\bar{x}_{i,k_i^{\text{in}}-1}, \bar{u}_{i,k_i^{\text{in}}-1})$ at t_1. Together with Assumption 2 this means that for a constant T_i^*, $w_i = (\bar{x}_{i,1}^*, \bar{u}_{i,1}^*, \bar{x}_{i,2}^*, \ldots, \bar{x}_{i,N}^*, \kappa^{\text{f}}(\bar{x}_{i,N}), f(\bar{x}_{i,N}, \kappa^{\text{f}}(\bar{x}_{i,N})))$ is a feasible solution to (7) for parameters $(x_{i,1}, T_i^*)$ at t_1, where $x_{i,1} = \bar{x}_{i,1}^*$. Hence, if $\boldsymbol{T}^*$ is feasible in (8) at X_0, it is also feasible for $X_k, k > 0$, and consequently, at least one solution to the intersection-level problem (8) exists for all closed-loop states. Consequently, the intersection level control loop is persistently feasible, and by construction therefore also the vehicle-level control loop.

□

B. Nominal stability

By Assumption 2, a model predictive controller based on (10) stabilizes the uncoordinated vehicles to the sets $\mathcal{N}_i$. However, the assumption does not directly imply stability for the coordinated vehicles using (7), due to the introduction of the constraints (7e) and (7f). 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 2, the uncoordinated closed-loop system will remain in $\mathcal{N}_i$. However, for a given timeslot t, the coordinated closed-loop system must deviate from $\mathcal{N}_i$ to satisfy (7e) and (7f). Consequently, for collision-free stabilization to $\mathcal{N}_i$, both $V_i^{\text{u}}(x_{i,k})$ 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 $\mathcal{P}_i = \{x_{i,k} \mid p_{i,k} \geq p_i^{\text{out}}\}$. This means that stability of the closed loop system means that all vehicles both pass the intersection and reach $\mathcal{N}_i$. We formalize the stability result in the following Theorem.

Theorem 5 (Stability of the bi-level controller)**.** *Suppose that the timeslots are updated through the solution of* (8) *and Assumptions 2 and 3 hold. Then, the bi-level controller stabilizes X_k to $\mathcal{Q} = \prod_{i=1}^{N_a} \mathcal{Q}_i$, where $\prod$ denotes the Cartesian product. Moreover, the bi-level controller is stabilizing if a) only $\boldsymbol{T}$ is optimized and S is fixed b) $\boldsymbol{T}$ and/or S optimized at a lower rate than the vehicle-level problems or c) both $\boldsymbol{T}$ and S are fixed.*

Proof. The proof is given in Appendix A. □

IV. Robustness with respect to perturbations

The vehicle-level MPC Problem (7) differs from standard MPC problem formulations in that the position constraints (7e) and (7f) 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 by the real system is typically difficult, and the persistent feasibility guarantees of the nominal case no longer hold. Since violations of the position constraints (7e) and (7f) by the vehicles implies a risk of collisions inside the intersection, the issue must be resolved for the bi-level controller to be useful in practice. In this section, we present a relaxation of the bi-level control formulation which ensures persistent feasibility of the optimization problems (7) and (8). Furthermore, we also state conditions under which the closed-loop system is collision free.

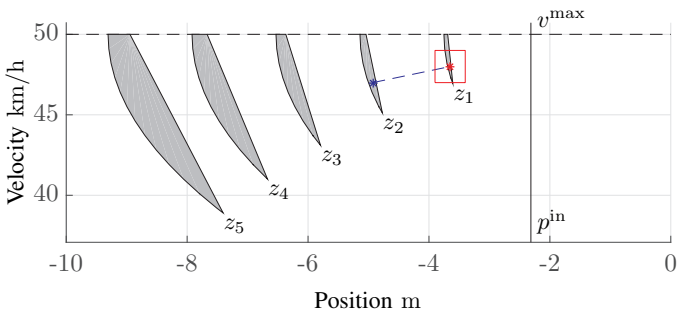


Fig. 3: Illustration of the feasible sets of (7) for a given T_i using the double integrator system of Example 1 and $t_{\text{s}} = 0.1$. The set z_j is the set of states $x_{i,k}$ for which (7) has a solution when $t_i^{\text{in}} - t_k = jt_{\text{s}}$. The red star is a one-step prediction from the blue star and the red box the bounds of a modest uniform uncertainty ($\pm 1\text{km/h}$ and ± 0.25 m), stemming from, e.g., model mismatch, measurement errors or external perturbations. The figure illustrates that since z_j is shrinking as the vehicle approach the intersection, infeasibility of (7) is more likely.

A. Ensuring optimization problem feasibility

As described in Section III-A, the position constraints (7e) and (7f) move closer to the first prediction stage as the vehicle approach 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 (7e) and (7f) decreases. As illustrated in Fig. 3, this causes the set of states for which the vehicle-level MPC problem (7) 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 (7) 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 (7) and (8) along the closed-loop system trajectories, we propose the following relaxation of the constraints of the vehicle MPC-problem (7):

$$
\begin{aligned}
V_i^r(x_{i,k}, T_i) := \min_{w_i, \sigma_i} \quad & \rho_i(\sigma_i) + (7\text{a}) && \text{(12a)}\\
\text{s.t.} \quad & (7\text{b}), (7\text{c}), (7\text{d}) && \text{(12b)}\\
& p_i(t_i^{\text{in}}, w_i) - p_i^{\text{in}} \leq \sigma_i^{\text{in}}, && \text{(12c)}\\
& p_i^{\text{out}} - p_i(t_i^{\text{out}}, w_i) \leq \sigma_i^{\text{out}}, && \text{(12d)}\\
& \sigma_i^{\text{out}} \geq 0,\ \sigma_i^{\text{in}} \geq 0, && \text{(12e)}
\end{aligned}
$$

where $\sigma_i = (\sigma_i^{\text{in}}, \sigma_i^{\text{out}})$ are slack variables and $\rho_i(\cdot)$ is an exact penalty function. The solution to (12) has the following properties:

Proposition 6 (Characteristics of solutions to the relaxed vehicle problem)**.** *Provided that $\nabla \rho_i(0)$ is large enough, the relaxed problem has the following properties:*

(a) whenever there exists a feasible solution for the original problem (7)*, the relaxed problem* (12) *yields the same primal solution and*
(b) otherwise, the relaxed problem (12) *yields a solution which minimizes* $||\sigma_i||_\infty$.

Proof. This result of (a) is well-known, and can be found in, e.g., [9, Theorem 14.3.1]. However, we could not find a proof for (b) in the literature, and therefore provide a proof in Appendix B, where we formalize the result in Theorem 12. □

We note that with the relaxation of the constraints, the set of $x_{i,k}$ for which (12) is feasible is the same as for the uncoordinated problem (10), 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 (8) can be written as

$$\min_{\boldsymbol{T},S} \quad \sum_{i=1}^{N_a} V_i^r(x_{i,k}, T_i) \quad \text{s.t.} \quad (8c), (8d). \tag{13}$$

Consequently, the use of (12) instead of (7) relaxes (8), and the following result holds for the characteristics of its solutions.

Proposition 7 (Characteristics of solutions to the relaxed timeslot allocation problem)**.** *Provided that* $\forall i$*,* ρ_i *is chosen large enough, the solution to the relaxed timeslot allocation problem* (13) *coincides with the solution to* (8)*, whenever a solution to* (8) *exist. If* $\rho_i(\cdot) \in \mathbb{I}_{[1,N_a]}$ *are chosen so that* $\nabla\rho_i(0)$ *is large enough for all* i*, the solution* $(S, \boldsymbol{T})$ *is such that the largest constraint violation in* (12) *among all vehicles is minimized.*

Proof. The intersection level problem (8) can equivalently be stated as

$$\min_{\boldsymbol{T},S,\sigma,W} \quad \sum_{i=1}^{N_a} \rho_i(\sigma_i) + \sum_{i=1}^{N_a} (7a) \tag{14a}$$
$$\text{s.t.} \quad (12b) - (12e), \quad i \in \mathbb{I}_{[1,N_a]} \tag{14b}$$
$$t_{s_i}^{\text{out}} \leq t_{s_{i+1}}^{\text{in}}, \tag{14c}$$
$$S \in \text{perm}(\mathbb{I}_{[1,N_a]}), \tag{14d}$$

where $\sigma = (\sigma_1, \ldots, \sigma_{N_a})$ and $W = (w_1, \ldots, w_{N_a})$. The result follow from Proposition 6. □

The relaxation thus enables the bi-level controller to act, and transition between, two modes. If, for instance, a previously feasible timeslot schedule $\boldsymbol{T}$ has become infeasible for one or more vehicles due to e.g. perturbations or model-mismatches, the timeslots are recomputed. In particular, the timeslot allocation controller (13) utilizes the authority it has over *all* vehicles to, if necessary, find the $(\boldsymbol{T}, S)$ which results in the smallest constraint violation $\|\sigma\|_\infty$.

B. Ensuring closed-loop collision avoidance

Although the modification introduced in (12) ensures persistent feasibility, the closed-loop system is not guaranteed to satisfy the collision avoidance constraints. However, from Proposition 6, we know that the modified bi-level controller minimizes violations of the position constraints (7e) and (7f). If an upper bound can be found to the smallest violation of (7e) and (7e) by the closed-loop system, (12) can be modified to ensure collision avoidance of the closed-loop system. In particular, consider the following modification of (12)

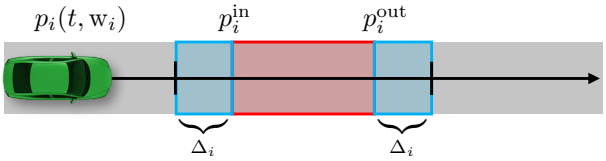


Fig. 4: Illustration of the enlargement of the intersection resulting in a tightening of constraints (12c) and (12d)

.

$$V_i^t(x_{i,k}, T_i, \Delta_i) :=$$
$$\min_{w_i} \quad (7a) \tag{15a}$$
$$\text{s.t.} \quad (7b), (7c), (7d), (12e) \tag{15b}$$
$$p_i(t_i^{\text{in}}, w_i) - p_i^{\text{in}} + \Delta_i \leq \sigma_i^{\text{in}}, \tag{15c}$$
$$p_i^{\text{out}} - \Delta_i - p_i(t_i^{\text{out}}-, w_i) \leq \sigma_i^{\text{out}}, \tag{15d}$$

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

Proposition 8. *Suppose that* $\boldsymbol{T}$ *satisfies* (8c) *for some* S *and that* $\forall i$*, at* t_0*,* (15) *is feasible for* $(x_{i,k}, T_i, \Delta_i)$*. Then the closed-loop application of* (15) *ensures that no collisions occur if* Δ_i *is chosen so that,* $\Delta_i \geq \sigma_i^{\text{in}}$ *and* $\Delta_i \geq \sigma_i^{\text{out}}$*,* $\forall k$ *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 $\bar{p}_i(t_i^{\text{in}}) \leq p_i^{\text{in}} - \Delta_i + \sigma_i^{\text{in}}$, which due to $\Delta_i \geq \sigma_i^{\text{in}}$, gives that $\bar{p}_i(t_i^{\text{in}}) \leq p_i^{\text{in}}$. Similarly, due to $\Delta_i \geq \sigma_i^{\text{out}}$, we have that $\bar{p}_i(t_i^{\text{out}}) \geq p_i^{\text{out}}$. 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 $[t_i^{\text{in}}, t_i^{\text{out}}] \cap [t_j^{\text{in}}, t_j^{\text{out}}] = \emptyset, j \neq i$ due to (8c), $\bar{p}_i(t) \in [p_i^{\text{in}}, p_i^{\text{out}}] \Rightarrow \bar{p}_j(t) \notin [p_j^{\text{in}}, p_j^{\text{out}}]$, $i \neq j$ ensuring that no collisions can occur for the closed-loop system. □

Remark 9. 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 to model-mismatch, measurement errors and external perturbations, and in particular, the effects on constraint satisfaction.

V. Experimental Validation

In this section we present experimental results from a 4-way intersection scenario, obtained from an implementation of the bi-level controller on a test setup consisting of three automated vehicles equipped with GPS positioning and vehicle-to-vehicle communication systems.

For ease of implementation, the intersection level problem (8) was solved for a fixed order $S = (1, \ldots, N_a)$ 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 bi-level, closed-loop controller was performed at the Asta Zero test-track outside Gothenburg, Sweden on a heterogenous test setup consisting of three different Volvo vehicles. In all experiment runs, the vehicles were initialized in configurations which would lead to collisions if no action was taken. From these initial configurations, the bi-level 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. 5a, second, in an actual intersection, shown in Fig. 5b. In particular, the parallel configuration was used to safely carry out the validation at at higher speeds. Due to collision risks, the controller was only validated at speeds up to 50 $\mathrm{km/h}$ and with added safety margins in the crossing configuration. We emphasize that there were no differences in the controller between the two configurations.

Test platform: The test setup consisted of a central computer, acting as a coordinator, and three Volvo Cars: one Volvo S60 D5 turbo-diesel sedan, one Volvo S60 T6 petrol-turbo sedan and one Volvo XC90 T6 petrol-turbo SUV. The vehicles were equipped with Real Time Kinematic (RTK) GPS receivers, wheel encoders and inertial measurement units and vehicle-to-vehicle communication devices from RENDITS [1]. Furthermore, to improve the positioning accuracy, the measurement data was fused using Extended Kalman Filters (EKF), based on [15].

The one-dimensional positions $p_{i,k}$ was constructed by first projecting the current estimate of the vehicle's global-position onto a reference path along the road. As illustrated in Fig. 6, $p_{i,k}$ was thereafter taken as the distance along the path from the projected point to the start of the intersection. Finally, the clocks in the three vehicles were synchronized through the GPS receivers.

Algorithmic implementation: The vehicle control commands were computed using the relaxed vehicle-level problem (12), where the solution was obtained using the dedicated MPC-QP-solver HPMPC [10].

To solve the fixed-order timeslot allocation problem (8), we employed the semi-distributed SQP method developed in [17] and [35], which we briefly describe here. As noted in [17], the constraints (8b) can be written as $g_i(x_{i,k}, T_i) \leq 0, \forall i$, and evaluated by solving 4 linear programs.

With $\mathcal{L}_i(x_{i,k}, T_i, \lambda_i) = V_i(x_{i,k}, T_i) + \lambda_i^\top g_i(x_{i,k}, T_i)$, the Lagrange function of (8) is

$$\mathcal{L}(\boldsymbol{T}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{i=1}^{N_a} \mathcal{L}_i(x_{i,k}, T_i, \lambda_i) + \boldsymbol{\mu}^\top M \boldsymbol{T} \tag{16}$$

where the constraint (8c) is written $M\boldsymbol{T} \leq 0$ and $\boldsymbol{\mu}$ is the corresponding Lagrange multiplier. Moreover, $\forall i$, λ_i is the Lagrange multiplier corresponding to (8b) and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{N_a})$. To solve the intersection level problem, (8),

Algorithm 1 Schematic description of the Distributed SQP. The arguments of the involved functions have been removed for brevity.

1: Coordinator initializes the problem.
2: **while** exit conditions not fulfilled **do**
3: Coordinator broadcasts $\boldsymbol{T}$.
4: Each vehicle solves (7)
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 (18).
7: Coordinator and vehicles compute α.
8: Coordinator takes step (17).

the iteration

$$\boldsymbol{T}^+ \leftarrow \boldsymbol{T} + \alpha \Delta \boldsymbol{T} \tag{17a}$$

$$\boldsymbol{\lambda}^+ \leftarrow \boldsymbol{\lambda}(1-\alpha) + \alpha \boldsymbol{\lambda}_{QP}, \tag{17b}$$

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 quadratic program

$$\min_{\Delta \boldsymbol{T}} \quad \sum_{i=1}^{N_a} \frac{1}{2} \Delta T_i^\top H_i \Delta T_i^\top + \nabla_{T_i} V_i^r(x_{i,k}, T_i)^\top \Delta T_i \tag{18a}$$

$$\text{s.t.} \quad g_i(x_{i,k}, T_i) + \nabla g_i(x_{i,k}, T_i)^\top \Delta T_i \leq 0, \ i \in \mathbb{I}_{[1,N_a]}, \tag{18b}$$

$$t_{s_i}^{\text{out}} \leq t_{s_{i+1}}^{\text{in}}, \quad i \in \mathbb{I}_{[1,N_a-1]}, \tag{18c}$$

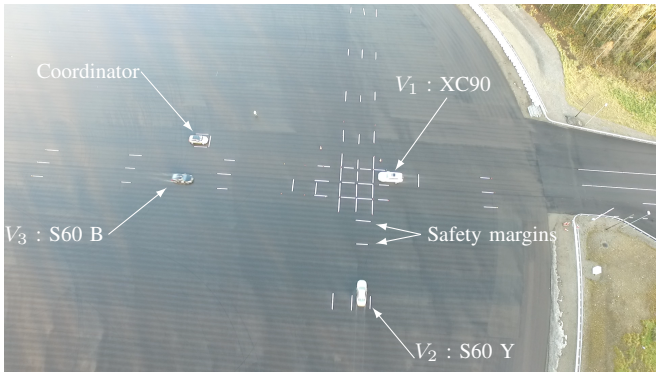
where $\lambda_{i,QP}$ is the Lagrange multipliers of the constraint (18b). Furthermore, the step-size α is in each iteration chosen through the line search procedure described in [35], and H_i is the possibly regularized Lagrange function Hessian $\nabla^2_{T_i} \mathcal{L}_i(x_{i,k}, T_i, \lambda_i)$. 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 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 computation performed by the coordinator is the solution of the small SQP sub-problems (18).

Prediction model, objective function and parameters: All vehicles were commanded through a low-level interface, which tracks commanded acceleration inputs. We therefore used the double-integrator dynamics introduced in Example 1 as the prediction model in (7). 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 (7d) were $0 \leq v_{i,k}$, $u_{i,k} \geq u_i^{\min} = -3\ \mathrm{m/s^2}$ and $u_{i,k} \leq u_i^{\max} = 1.6\ \mathrm{m/s^2}$, $\forall i, k$, where the input bounds reflect bounds present in the vehicles' actuation interfaces. Finally, the objective was taken to be

$$J_i(w_i) = (v_{i,N} - v_i^{\text{ref}})^2 Q_i^f + \sum_{i=0}^{N_i-1} (v_{i,k} - v_i^{\text{ref}})^2 Q_i + u_{i,k}^2 R_i, \tag{19}$$

(a) Photo 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 photo of the crossing configuration used in the experimental validation. The white lines before the square representing the intersection illustrates the different safety margins employed. Video material from the experimental campaign can be found in [16].

Fig. 5: Photos of the two different configurations used in the experimental validation.

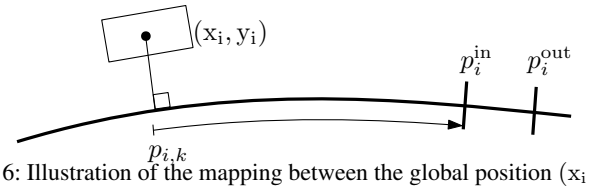


Fig. 6: Illustration of the mapping between the global position $(\mathrm{x_i}, \mathrm{y_i})$ to the position $p_{i,k}$ on the road-bound coordinate.

where R_i, Q_i and Q_i^{f} are positive constants and v_i^{ref} is a reference speed, and were all varied between different experiment runs. The horizon and sampling time were set to $N = 200, t_{\text{s}} = 0.1$ 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 m and 12 m, corresponding to two different safety margins (cf. Fig. 5b) [1].

Due to 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 (7), and (8) was solved every $t_{\text{s}}^{\text{c}} = 3$ seconds. In particular, in most cases the time required by Algorithm 1 ranged between 0.5 s 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 (8) between $(k-1)t_{\text{s}}^{\text{c}}$ and kt_{s}^{c}, using a prediction of the system state X at kt_{s}^{c}. To ensure that the vehicle-level controller remained feasible throughout the experiments, the relaxed formulation (12) was used to compute the control commands $u_{i,k}$. The exact penalty function

$$\rho_i(\sigma_i) = \frac{1}{2}\sigma_i^\top \sigma_i \phi_i^q + \phi_i \mathbf{1}^\top \sigma_i, \tag{20}$$

[1]The GPS antennas were mounted on the center of each vehicle, and the beginning and end of the intersection used in (7e) and (7e) was 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 (7e) consequently meant that the front of the vehicle were before p_i^{in} at t_i^{in}, and satisfaction of (7f) meant that the rear of the vehicle had passed p_i^{out} at t_i^{out}. This can be observed in Fig. 5a.

was used, where ϕ_i and ϕ_i^q are positive weights. 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) 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 bi-level 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, vehicle 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. 7b and Fig. 7c, 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. 7a, illustrates that the bi-level level controller thereby satisfied the precedence constraints (8c). As emphasized by the cutout of Fig. 7a, 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 (19), 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 $kt_{\mathrm{s}}^{\mathrm{c}}$ and $(k+1)t_{\mathrm{s}}^{\mathrm{c}}$ were 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. 7d, 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 bi-level controller had been initialized, the driver of the first vehicle pressed the brake pedal, and thereby suspended the MPCs authority to control that vehicle between $t = 1.8$ s and $t = 3.8$ s. The response of the system response is shown in Fig. 8. Fig. 8d, illustrates the reaction of the intersection level control loop to the disturbance, and shows the issuing of larger adjustments to the timeslots of all vehicles compared to the unperturbed case (cf. Fig. 7d). Recall that since the time required to solve the intersection level problem (8) is non negligible, the timeslot applied at $kt_{\mathrm{s}}^{\mathrm{c}}$ is computed using information known at $(k-1)t_{\mathrm{s}}^{\mathrm{c}}$ (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. 8c. 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 re-computed 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 increases in vehicle 3. This illustrates the ability of the intersection level control loop to distribute the effort needed to counter the perturbation between the vehicles according to the objective (19).

To further demonstrate the capabilities of the bi-level controller, results from a simulation of the perturbed scenario where the intersection level control loop includes re-computing the order s is given in Fig. 9. 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 $\dot{a}_i(t) = \frac{1}{\tau}(u_i(t) - a_i(t))$, where $\tau = 0.5$ s for the two S60s and 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-A.

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 $\boldsymbol{T}$ that is significantly larger than in the experiment. The effect of the large delays present during the experiment can be seen by comparing Fig. 9b and Fig. 8d: 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. 8a, the introduction of the large perturbation did not significantly degrade the ability of the closed-loop system to satisfy the precedence constraints (8c). During the experimental campaign, more than 75 runs where performed, which involved speeds ranging from 30 to 90 $\mathrm{km/h}$, more than 20 different sets of parameters (objective function weights, starting configurations, etc.) and in several cases, larger perturbations.

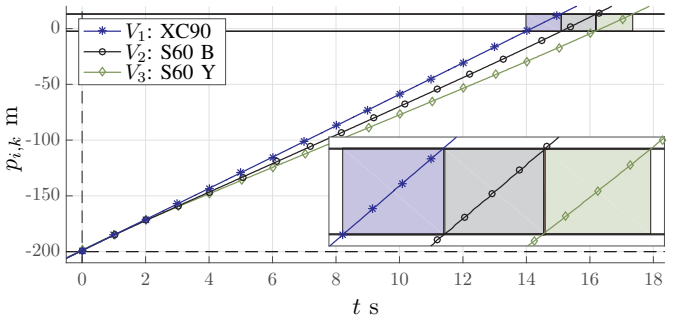
With a few exceptions, the bi-level controller consistently managed to command the vehicles so that little or no overlap of the timeslots occurred: In 56.7%[2] of the evaluations, the position constraints (7e) and (7f) were satisfied by the closed-loop system trajectories, while in the remaining 40.2%, 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 was expected.

The distribution of the constraint violations are shown in Fig. 10, 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.

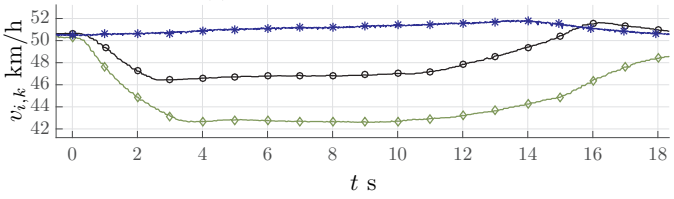
In Fig. 11 we illustrate the consistency of the controller by reporting the position trajectories of the closed-loop system in 34 of the successful experimental runs performed in the parallel configuration. The figure shows an overlay of the system trajectories in the configuration spaces of the vehicle 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 the position of vehicle 2. The intersection is represented as the rectangle $\mathcal{R} = \left([p_1^{\mathrm{in}}, p_1^{\mathrm{out}}] \times [p_2^{\mathrm{in}}, p_2^{\mathrm{out}}]\right) \cup \left([p_3^{\mathrm{in}}, p_3^{\mathrm{out}}] \times [p_2^{\mathrm{in}}, p_2^{\mathrm{out}}]\right)$, 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. 10, are noticeable in Fig. 11 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. 10 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, discussed in Section IV-B. In particular, the red square in the cutout of Fig. 11 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 (15) with $\Delta = 0.7$ m. As can be seen in the figure, the smaller intersection would in this case have been completely mutually exclusive for all experimental runs,

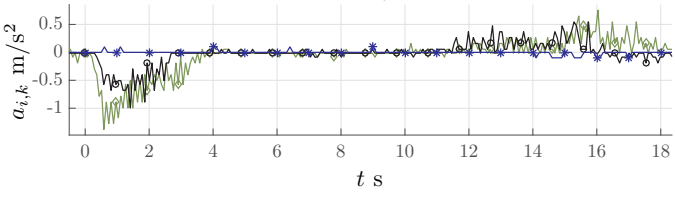
[2]The data was collected during 75 experimental runs, each with three involved vehicles and 2 position constraint evaluations per vehicle. Of the 450 total evaluations, 255 satisfied the constraints, 181 had a violations smaller than 1 m and 14 a violation larger than 1 m.



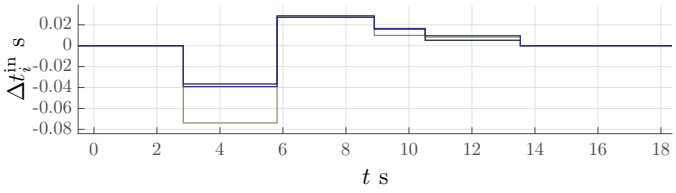
(a) Distance to intersection



(b) Velocity

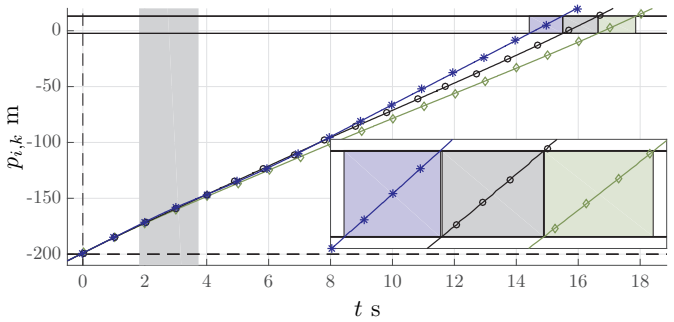


(c) Acceleration

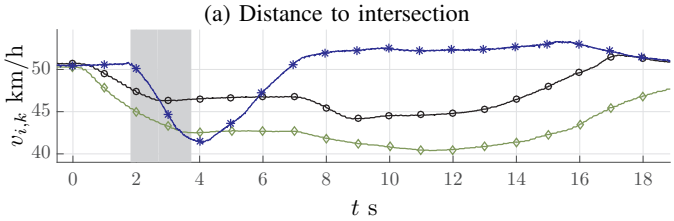


(d) Timeslot changes

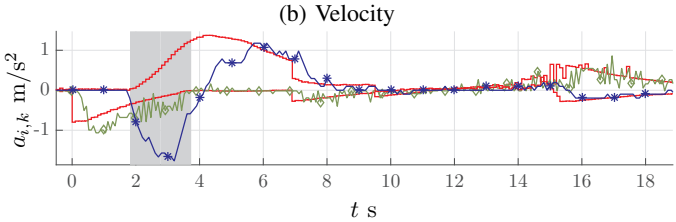
Fig. 7: Example from non-perturbed scenario. In (a), the solid horizontal lines indicates the beginning and end of the intersection, and the dashed lines the scenario starting time and position. The colored blocks illustrate the timeslot used by each vehicle. Subfigure (d) shows the difference in the optimal t_i^{in} between two executions of the intersection level control loop, as sent out by the coordinator.



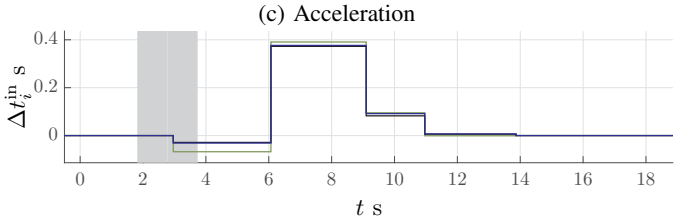
(a) Distance to intersection



(b) Velocity

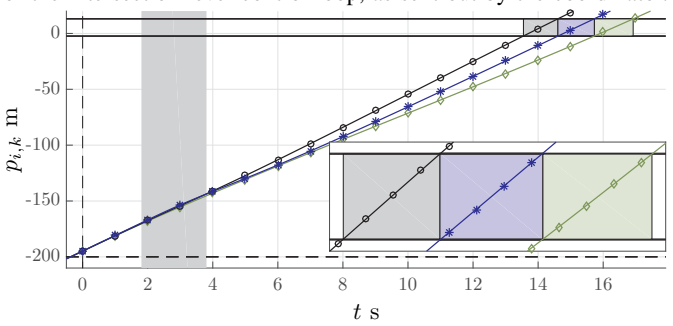


(c) Acceleration

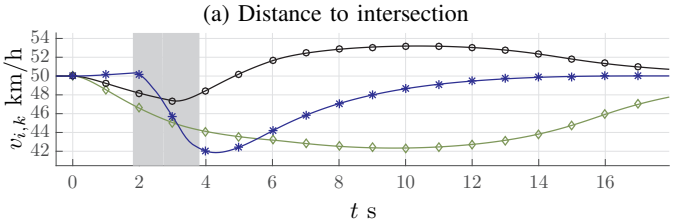


(d) Timeslot changes

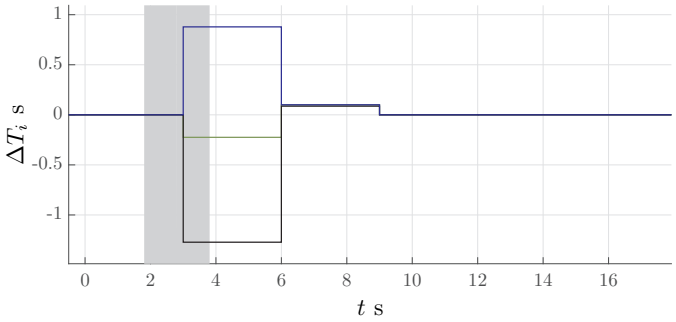
Fig. 8: Example from a perturbed scenario. In (a)-(d), the gray bar illustrates the time during which the MPC controller lost authority of vehicle 1due to that the driver overrode the system by pressing the brake pedal. The difference between the actual $a_{i,k}$ acceleration and the MPC command $u_{i,k}$ for vehicle 1 and 3 is illustrated in (c).



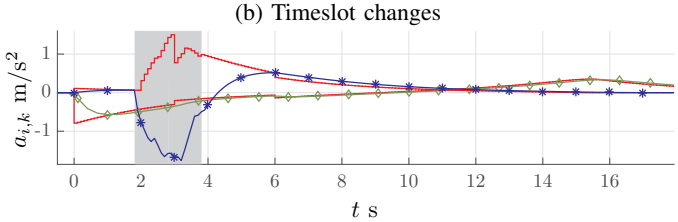
(a) Distance to intersection



(c) Velocity



(b) Timeslot changes



(d) Acceleration

Fig. 9: Example from perturbed simulated scenario including recalculation of the order S. The disturbance from Fig. 8 is applied to the simulated vehicle 1.

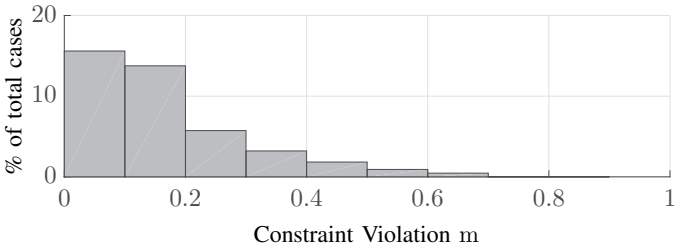


Fig. 10: Constraint violation statistics from 75 experiment runs, corresponding to 450 evaluations of the position constraints (7e) and (7f) by the closed-loop system. In 255 out of 450 evaluations, the constraints were satisfied by the closed-loop system. A histogram of the constraint violations in the remaining cases is given in the figure. The plot only shows data from successful runs where the violation was smaller than 1 m.

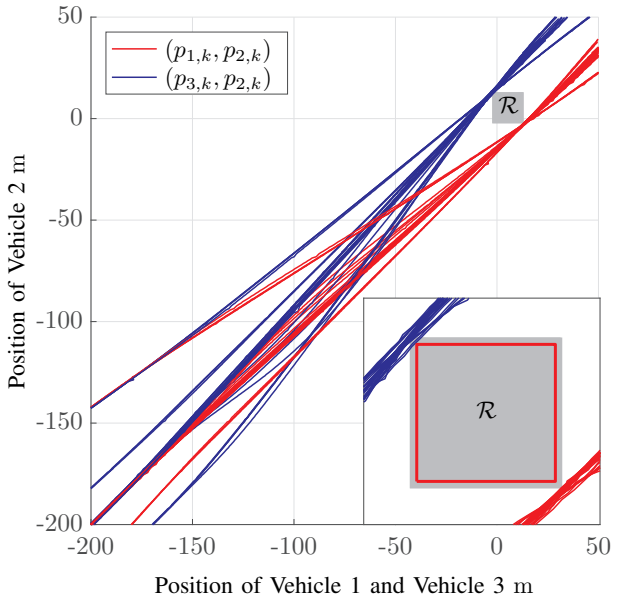
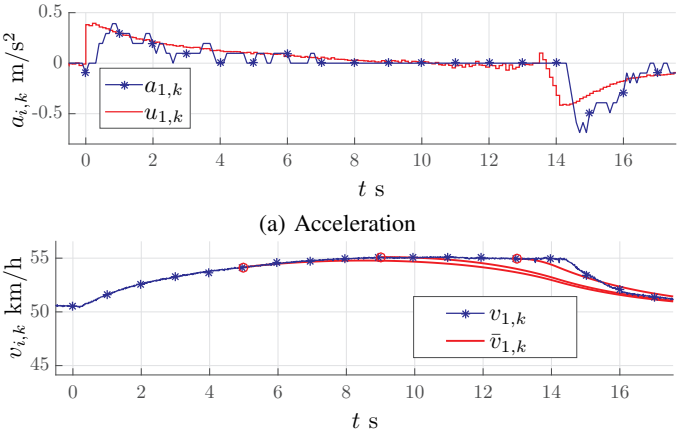


Fig. 11: Position trajectories from 34 experimental runs with 3 vehicles. The trajectories of the system in the position space of vehicles 1 and 2 is shown in red, whereas the corresponding trajectories for vehicles 2 and 3 are shown in blue. The gray rectangle $\mathcal{R} = \left([p_1^{\text{in}}, p_1^{\text{out}}] \times [p_2^{\text{in}}, p_2^{\text{out}}]\right) \cup \left([p_3^{\text{in}}, p_3^{\text{out}}] \times [p_2^{\text{in}}, p_2^{\text{out}}]\right)$ corresponds to the intersection, and consequently, when a trajectory is inside it, at least one involved vehicle violates one of the position constraints (7e) and (7f). The red square illustrates an intersection where the width is 1.4 m less than $\mathcal{R}$ in all dimensions.

and the closed-loop system would have satisfied all position constraints (7e) and (7e).

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. 12a, which displays the commanded, $u_{i,k}$, and actual, $a_{i,k}$, acceleration for a successful run. Note in



(a) Acceleration

(b) Velocity

Fig. 12: Illustration of the model-plant mismatch. The demanded, $u_{i,k}$, and actual, $a_{i,k}$ acceleration is shown in (b), where the difference between the two highlight non-modeled dynamics. The corresponding velocity profile is given in (a) where the prediction of $\bar{v}_{i,k}$ at $t = 5, 9, 13$ is displayed.

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.12b, where an open loop prediction is shown for comparison. The effects of the non-modeled dynamics are clearly visible in the difference between the open- and closed-loop trajectories. Moreover, the upper bound on performable 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 (7). Due to these model-plant mismatches, the nominal prediction could satisfy the position constraints (7e) and (7f), 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. 13, where the position estimate used by the controller is overlayed on an off-line, re-construction 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 (7e) and (7f) for vehicle 1, the violations were smaller than 0.18 m, compared to 0.38 m 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 occurred with a frequency of 1 Hz, which is visible around times $t = 15.5$ s and $t = 16.5$ s in Fig. 13. 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 (7e) and (7f),

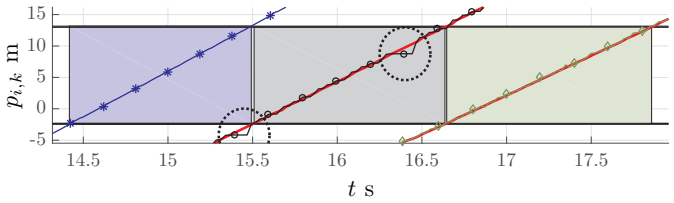


Fig. 13: Position trajectories of one experimental run illustrating the positioning issues in vehicle 2 (black) and 3 (green). Vehicle 2 suffered recurring large position errors, which are encircled in the figure. For reference, more accurate position trajectories are given in red, which have been reconstructed off-line based on raw-measurements acquired during the experimental run.

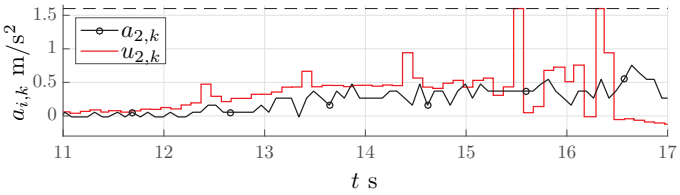


Fig. 14: Desired and actual acceleration of Vehicle 2 corresponding to the position trajectories in Fig. 13. The effects of the faulty position estimates are visible as spikes in the demanded acceleration between $t = 12$ s and $t = 17$ s.

the controller attempted to compensate the faulty position by commanding higher acceleration from the vehicle. The behavior can be seen in Fig. 14, 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 (7e) far exceeds the upper bound on the input, and the MPC problem (7) 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 (12) was used, the infeasibility issue is avoided. Instead, Fig. 14 demonstrates the constraint-violation minimization mode of the controller which was discussed in Section IV-A; i.e., while no input exists such that the position constraints (7e) and (7f) 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 $\nabla\rho_i(\sigma_i) = \phi_i\mathbf{1} + \phi_i^q\sigma_i$ (cf. (20)) 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 Fig. 15 and Fig. 16 respectively. In both cases, the bi-level 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. 15b, 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. 16b. As a result, the closed-loop controller with small ϕ_i causes significant violations of the collision avoidance constraints, as illustrated in Fig. 15a, while an adequately chosen ϕ_i results in only minor constraint violations, as shown in Fig. 16a.

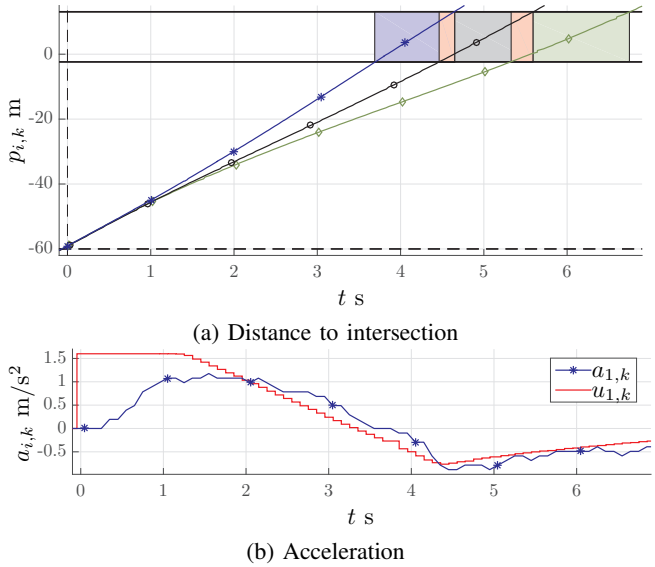
VI. Discussion, Conclusions and Future Research

In this paper, we first proposed, analyzed and established properties of a bi-level, closed-loop controller for coordination of automated vehicles at intersections. In particular, we proved that the bi-level 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 controller applicability through experimental validation. In particular, we showed that the controller performance was consistent, that it managed to counteract both large and small perturbations and that 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.

Furthermore, as opposed to many other intersection coordination schemes, the controller presented in this paper does not depend on a specific dynamic equation (cf., (7c)), specific constraints (cf. (7d)) nor a specific objective function (cf. (7a)). 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 emphasis on energy efficiency in low intensity situations.

Comments on the experimental results: The precision observed in the enforcement of the position constraints (7e) and (7f) 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.

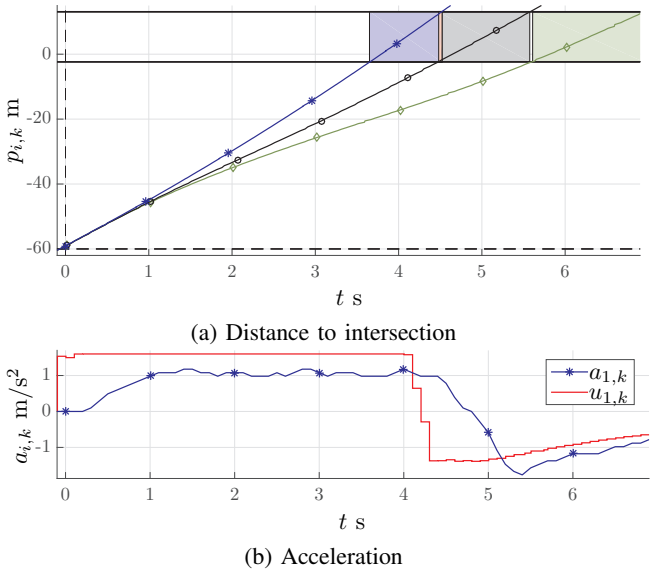
While access to accurate positioning systems is a strong assumption outside the experimental domain, we emphasize that the controller presented in this paper could handle large uncertainties by appropriately choosing the constraint tightening parameter Δ_i. Most likely, a practical implementation of the controller would even utilize margins well beyond the constraint satisfaction accuracy, if nothing else than for the confidence of the passengers. While it is hard to estimate, a



(a) Distance to intersection

(b) Acceleration

Fig. 15: Illustration of controller behavior for Vehicle 1 with a penalty function where $\nabla\rho_i(\sigma_i)$ is too small. In (a), the light-red colored blocks illustrates the times during which more than one vehicle is inside the intersection (i.e., where the collision avoidance constraints are violated).



(a) Distance to intersection

(b) Acceleration

Fig. 16: Illustration of controller behavior for Vehicle 1 with a penalty function where $\nabla\rho_i(\sigma_i)$ is large enough. Note the discrepancy between the actual and modeled input saturation level $u_1^{\max}$ in (b), discussed in Section V-B3.

practical implementation of the controller could start from a large Δ_i, and use the data obtained from passing vehicles to adapt it on-line through, e.g., machine learning techniques.

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 which 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.

Future research: 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 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.

Moreover, a large-scale performance evaluation of the proposed controller, including comparisons with both current regulatory mechanisms (i.e., traffic lights) and other coordination schemes, is lacking. The benefits of directly optimizing e.g. fuel consumption or travel delay should be compared to heuristic methods which attempts to achieve the same indirectly.

Due to the safety-critical reliance on vehicle-to-vehicle 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 (8), also how the problem is solved. Therefore, an adaptation of the algorithm that explicitly accounts for limited-capacity communication channels is highly desirable.

Moreover, while the coordination controller presented in this paper does not rely on a specific objective function, Assumption 2 imposes some restrictions on its characteristics. It is our intention to incorporate more general, economic objectives to the coordination controller in future work.

Finally, while we did not discuss algorithms for the solution of (8) 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 NLP solver to address (8), including finding the order S.

Acknowledgments

This research was funded by The Swedish Research Council under Grant No. 2012-4038, Vinnova under grants 2015-04849 (Copplar project) and 2015-03075 (AstaZero program). The Authors would like to thank Gianluca Frison for supporting us with HPMPC and Albin Severinson for help with the V2V communication equipment, and express our gratitude towards he following individuals for help with equipment for the experimental validation: Arpit Karsolia and Fredrik von Corswant at Chalmers REVERE-lab, Wojciech Mostovski at Halmstad University, Henrik Lind at Volvo Cars and Alessandro Colombo at Politechnico di Milano. We also want to thank our colleagues Giuseppe Giordano, Ankit Gupta and Johan Karlsson as well as Marco di Vaio and Gabriel de Campos, who acted as drivers and helpers during the experiments. Finally, we want to thank Fengco Real Time Control and Leica Geosystems for technical assistance with the hardware.

APPENDIX

A. Proof of Theorem 5

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

Proposition 10 (Stability of one vehicle under a fixed timeslot)**.** *Suppose that Assumption 2 and 3 holds and that a feasible solution to* (7) *exist for* $(x_{i,k}, T_i)$*. Then the MPC based on* (7) *stabilizes the vehicle to* $\mathcal{Q}_i = \mathcal{N}_i \cap \mathcal{P}_i$*.*

Proof. Consider the candidate Lyapunov function

$$\bar{V}_{i,k}(x_{i,k}, T_i) = V_i(x_{i,k}, T_i) + \alpha_{1,i}(\max(0, p_i^{\text{out}} - p_{i,k})). \quad (21)$$

Since the feasibility set of (7) is a subset of the feasibility set of (10) we obtain

$$V_i(x_{i,k}, T_i) \geq V_i^u(x_{i,k}) \geq \alpha_{1,i}(|x_{i,k}|_{\mathcal{N}_i}), \quad (22)$$

where the last inequality is due to Assumption 2. Then there exists a $\mathcal{K}_\infty$-function $\bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) \leq \bar{V}_i(x_{i,k}, T_i)$:

$$\begin{aligned}
\bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) &:= \alpha_{1,i}\left(\frac{1}{2}|x_{i,k}|_{\mathcal{Q}_i}\right) && \text{(23a)}\\
&\leq \alpha_{1,i}\left(\frac{1}{2}\left(|x_{i,k}|_{\mathcal{N}_i} + \max(0, p_i^{\text{out}} - p_{i,k})\right)\right) && \text{(23b)}\\
&\leq \alpha_{1,i}(|x_{i,k}|_{\mathcal{N}_i}) + \alpha_{1,i}(\max(0, p_i^{\text{out}} - p_{i,k})) && \text{(23c)}\\
&\leq \bar{V}_i(x_{i,k}, T_i) && \text{(23d)}
\end{aligned}$$

Here, (23c) follows from the property $\alpha(a+b) \leq \alpha(2a) + \alpha(2b)$ of $\mathcal{K}_\infty$-functions, and (23b) 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 $V_i^{\text{u}}(x_{i,k}) = V_i(x_{i,k}, T_i)$ for $x_{i,k} \in \mathcal{P}_i$. Due to Assumption 2, $V_i^{\text{u}}(x_{i,k}) = 0, \forall x_{i,k} \in \mathcal{N}_i$, 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 $\bar{V}_i(x_{i,k}, T_i) = V_i(x_{i,k}, T_i) = V_i^{\text{u}}(x_{i,k})$, and that continuity follows from Assumption 2. For case 2, (7e) is removed and (7f) 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 (7f) 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 $V_i(x_{i,k})$ due to Assumption 2.

We now turn to directions where $p_{i,k} < p_i^{\text{out}}$ and t_i^{out} is such that (7) is feasible. Standard results from parametric programming establish that the optimal value function of a parametric NLP is continuous in the problem parameters if the Mangasarian-Fromovitz Constraint Qualification (MFCQ) holds [Theorem 4.2, [8]]. Unfortunately, MFCQ fails at $\partial\mathcal{P}_i$, i.e., continuity does not directly follow from the formulation (7). However, MFCQ does hold if the position component of the initial condition constraint (7b) 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 11 this relaxed formulation is equivalent to (7), 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., [28]. We denote the solution to (7) 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 4 states that closed-loop system is persistently feasible, we have that

$$\begin{aligned}
&V_i(x_{i,k+1}, T_i) \leq V_i(x_{i,k}, T_i)\\
&- \ell_i(x_{i,k}, u_{i,0}^*) - V_i^{\text{f}}(x_{i,N_i}^*) + \ell_i(x_{i,N}^*, \kappa_i^{\text{f}}(x_{i,N_i}^*)) \quad (24)\\
&+ V_i^{\text{f}}(f_i(x_{i,N_i}^*, \kappa_i^{\text{f}}(x_{i,N_i}^*))
\end{aligned}$$

By Assumption 2, this gives

$$V_i(x_{i,k+1}, T_i) - V_i(x_{i,k}, T_i) \leq \ell_i(x_{i,k}, u_{i,0}^*). \quad (25)$$

By construction, $p_{i,k}$ is non-decreasing in k, and therefore $\alpha_{1,i}(\max(0, p_i^{\text{out}} - p_{i,k}))$ is non-increasing 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, which is stabilized to the set $\mathcal{Q}_i$. □

We now provide the proof for Theorem 5:

Proof of Theorem 5. Consider the Lyapunov function candidate

$$\bar{V}(X_k) = \sum_{i=1}^{N} \bar{V}_i(x_{i,k}, T_i). \quad (26)$$

First, by Proposition 10 the terms in (26) 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

$$\sum_{i=1}^{N_a} \bar{\alpha}_{1,i}(|x_{i,k}|_{\mathcal{Q}_i}) \geq \tilde{\alpha}\left(\sum_{i=1}^{N_a} |x_{i,k}|_{\mathcal{Q}_i}\right) \geq \tilde{\alpha}(|X_k|_{\mathcal{Q}}), \quad (27)$$

and since

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

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 10 ensures decrease of $\bar{V}(X_k)$, since $\forall i$, $\bar{V}_i(x_{i,k}, T_i)$ decrease. Denoting the solution to (8) at X_k as $T_{i,k}^*, i \in \mathbb{I}_{[1,N_a]}$, we have by optimality that

$$\sum_{i=1}^{N_a} V_i(x_{i,k+1}, T_{i,k+1}^*) \leq \sum_{i=1}^{N_a} V_i(x_{i,k+1}, T_i) < \sum_{i=1}^{N_a} V_i(x_{i,k}, T_i), \quad (29)$$

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 non-increasing, $\bar{V}(X_k)$ 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 bi-level closed-loop system which is stabilized to $\mathcal{Q}$.

[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 $|a|_{\mathcal{B}} + |a|_{\mathcal{C}} \geq |a|_{\mathcal{A}}$.

Besides the case where $\boldsymbol{T}$ and S are optimized, the decrease condition (29) hold 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 bi-level controller is stabilizing under these conditions as well. □

Note that, if S is updated using an external heuristic, the bi-level controller is stabilizing if for S, $\boldsymbol{T}$ can be found which ensures (29). This is particularly important as the solution of the combinatorial problem (8) might be prohibitively hard, and reliance on heuristics for closed loop control might be necessary.

B. Proof of Proposition 6

Consider the NLP

$$\min_{w\in\mathcal{W}} \quad f(w) \tag{30a}$$
$$\text{s.t.} \quad h(w) \geq 0, \tag{30b}$$

and the relaxed NLP

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

where $\mathcal{W} = \{w \mid g(w) \geq 0,\ g_e(w) = 0\}$. We denote $\lambda \geq 0$ the Lagrange multiplier associated with constraint (30b) and let $\lambda_{\max}$ be the largest value it can attain. In the following, we are interested in analysing the properties of NLP (31) in the two cases in which NLP (30) is feasible and infeasible. For the former case, the following holds

Theorem 11. *Provided that* $\frac{\partial}{\partial s}p(0) \geq \lambda_{\max}$ *and NLP* (30) *has a solution, the solution of NLP* (31) *yields* $\sigma = 0$ *and* $w = w^*$ *which is also optimal for NLP* (30)*.*

Proof. A proof of this theorem can be found in, e.g., [9, Theorem 14.3.1], but we provide one here for the reader's convenience. We first note that the KKT conditions of (30) and (31) they differ only in that

$$h(w) \geq 0, \qquad \lambda_i h_i(w) = 0, \qquad \lambda \geq 0, \tag{32}$$

hold at optimality for (30), whereas

$$h(w) + \sigma \geq 0, \quad \lambda_i(h_i(w) + \sigma_i) = 0, \quad \lambda \geq 0, \tag{33a}$$
$$\sigma \geq 0, \quad s_i\mu_i = 0, \quad \mu \geq 0, \tag{33b}$$

and

$$\frac{\partial}{\partial\sigma}p(\sigma) - \lambda - \mu = 0, \tag{33c}$$

hold at optimality for (31), where μ is the Lagrange multiplier of the constraint (31c). A primal-dual solution to (30) and $\sigma = 0$ is thus a solution to (31) if $\frac{\partial}{\partial s}p(0) \geq \lambda_{\max}$, since

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

implies that (33b) hold. 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 (30) is infeasible, we further introduce the infeasibility minimizing NLP

$$\min_{w\in\mathcal{W},\sigma} \quad \|\sigma\|_\infty \tag{35a}$$
$$\text{s.t.} \quad h(w) + \sigma \geq 0, \tag{35b}$$
$$\sigma \geq 0 \tag{35c}$$

and the auxiliary NLP

$$\min_{w\in\mathcal{W},\sigma} \quad f(w) \tag{36a}$$
$$\text{s.t.} \quad h(w) + \sigma^* \geq 0, \tag{36b}$$

where σ^* is a solution to (35). We denote $\nu \geq 0$ the Lagrange multiplier associated with constraint (36b), and let $\nu_{\max}$ be the maximum value it can attain.

Theorem 12. *Provided that* $\nabla_\sigma p(\sigma^*) \geq \nu_{\max}$*, NLP* (31) *yields a solution* $\sigma = \sigma^*$ *and* $w = w^*$ *which is also optimal for* (35)

Proof. We first note that any solution σ to (31) can be written $\sigma = \sigma^* + \sigma$, for some $\sigma \geq 0$. Therefore, (31) is equivalent to

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

Since (37) is also a relaxation of (36), Theorem 11 gives that $\sigma = 0$ when $\nabla_\sigma p(\sigma^*) \geq \nu_{\max}$ and a solution to (36) exists. By construction, (35) and therefore (36) has a solution when (31) has a solution. Therefore, the solution to (31) is $\sigma = \sigma^*$ and $w = w^*$ which also is a solution to (35). □

REFERENCES

[1] RENDITS. `http://www.rendits.com/`. Accessed: 2016-11-05.

[2] L. Chen and C. Englund. Cooperative intersection management: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 17(2):570–586, Feb 2016.

[3] A. Colombo and D. Del Vecchio. Efficient algorithms for collision avoidance at intersections. In *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, pages 145–154, New York, NY, USA, 2012.

[4] G. R. de Campos, P. Falcone, and J. Sjöberg. Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing. In *Intelligent Transportation Systems - (ITSC), 2013 16th International IEEE Conference on*, pages 1456–1461, 2013.

[5] G. R. de Campos, P. Falcone, H. Wymeersch, R. Hult, and J. Sjöberg. Cooperative receding horizon conflict resolution at traffic intersections. In *53rd CDC*, pages 2932–2937, Dec 2014.

[6] K. Dresner and P. Stone. Multiagent traffic management: a reservation-based intersection control mechanism. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, pages 530–537, July 2004.

[7] K. Dresner and P. Stone. A Multiagent Approach to Autonomous Intersection Management. *Journal of Artificial Intelligence Research*, 31(1):591–656, March 2008.

[8] A.V. Fiacco and Y. Ishizuka. Sensitivity and stability analysis for nonlinear programming. *Annals of Operations Research*, 27(1):215–235, Dec 1990.

[9] R. Fletcher. *Practical Methods of Optimization*. Wiley, Chichester, 2nd edition, 1987.

[10] G. Frison, H.B. Sorensen, B. Dammann, and J.B. Jorgensen. High-performance small-scale solvers for linear model predictive control. In *Proc. of the European Control Conference*, pages 128–133, 2014.

[11] J. Gregoire and E. Frazzoli. Hybrid centralized/distributed autonomous intersection control: Using a job scheduler as a planner and inheriting its efficiency guarantees. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 2549–2554, Dec 2016.

[12] M. Hafner, D. Cunningham, L. Caminiti, and D. Del Vecchio. Cooperative collision avoidance at intersections: Algorithms and experiments. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1162–1175, 2013.

[13] 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 *American Control Conference (ACC), 2015*, pages 763–768, July 2015.

[14] 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 Processing Magazine*, 33(6):74–84, Nov 2016.

[15] R. Hult, E. Sancar, M. Jalalmaab, A. Vijayan, A. Severinson, M. Di Vaio, P. Falcone, B. Fidan, and S. Santini. Design and experimental validation of a cooperative driving control architecture for the Grand Cooperative Driving Challenge 2016. *Transactions on Intelligent Transportation Systems*, 2017. to Appear.

[16] R. Hult, M. Zanon, S. Gros, and P. Falcone. Optimal coordination of three cars approaching an intersection. `https://youtu.be/nYSXvnaNRK4`. Accessed: 2017-03-03.

[17] R. Hult, M. Zanon, S. Gros, and P. Falcone. Primal Decomposition of the Optimal Coordination of Vehicles at Traffic Intersections. In *Proceedings of the Conference on Decision and Control*, 2016.

[18] 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 Transactions on Intelligent Transportation Systems*, 16(3):1136–1147, June 2015.

[19] A. Katriniok, P. Kleibaum, and M. Joševski. Distributed Model Predictive Control for Intersection Automation Using a Parallelized Optimization Approach. In *IFAC World Congress 2017*. IFAC, 2017.

[20] K. D. Kim and P. R. Kumar. An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic. *IEEE Transactions on Automatic Control*, 59(12):3341–3356, Dec 2014.

[21] H. Kowshik, D. Caveney, and P. R. Kumar. Provable systemwide safety in intelligent intersections. *IEEE Transactions on Vehicular Technology*, 60(3):804–818, March 2011.

[22] J. Lee and B. Park. Development and Evaluation of a Cooperative Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):81–90, 2012.

[23] M. Li, K. Boriboonsomsin, G. Wu, W. Zhang, and M. Barth. Traffic energy and emission reductions at signalized intersections: a study of the benefits of advanced driver information. *International Journal of Intelligent Transportation Systems Research*, 7(1):49–58, 2009.

[24] D. Miculescu and S. Karaman. Polling-systems-based control of high-performance provably-safe autonomous intersections. In *53rd IEEE Conference on Decision and Control*, pages 1417–1423, Dec 2014.

[25] V. Milanes, J. Alonso, L. Bouraoui, and J. Ploeg. Cooperative Maneuvering in Close Environments Among Cybercars and Dual-Mode Cars. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):15–24, March 2011.

[26] N. Murgovski, G. R. de Campos, and J. Sjöberg. Convex modeling of conflict resolution at traffic intersections. pages 4708–4713, Dec 2015.

[27] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In *2015 European Control Conference (ECC)*, pages 3452–3458, July 2015.

[28] J.B. Rawlings and D.Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, 2009.

[29] J. Rios-Torres and A. A. Malikopoulos. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, 18(5):1066–1077, May 2017.

[30] G. Schildbach, M. Soppert, and F. Borrelli. A collision avoidance system at intersections using Robust Model Predictive Control. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 233–238, June 2016.

[31] M.C. Simon, T. Hermitte, and Y. Page. Intersection road accident causation: A European view. In *21st International Technical Conference on the Enhanced Safety of Vehicles*, pages 1–10, 2009.

[32] E. Steinmetz, R. Hult, G. R. de Campos, M. Wildemeersch, P. Falcone, and H. Wymeersch. Communication analysis for centralized intersection crossing coordination. In *11th International Symposium on Wireless Communications Systems (ISWCS)*, pages 813–818, Aug 2014.

[33] P. Tallapragada and J. Cortés. Coordinated intersection traffic management. *IFAC-PapersOnLine*, 48(22):233 – 239, 2015. 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015.

[34] 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 *International Conference on Computing, Networking and Communications (ICNC)*, pages 573–578, Feb 2015.

[35] M. Zanon, S. Gros, H. Wymeersch, and P. Falcone. An Asynchronous Algorithm for Optimal Vehicle Coordination at Traffic Intersections. In *20th IFAC World Congress*, 2017.

[36] Y. J. Zhang, A. A. Malikopoulos, and C. G. Cassandras. Optimal control and coordination of connected and automated vehicles at urban traffic intersections. In *2016 American Control Conference (ACC)*.

Robert Hult Robert Hult received B.S. degree in Mechanical Engineering in 2011, and the M.Sc. in Systems, Control and Mechatronics in 2013, both from Chalmers University of Technology, Sweden, 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, and the Diplôme d'Ingénieur from the Ecole Centrale Paris, in 2010. After research stays at the KU Leuven, University of Bayreuth, Chalmers University, and the University of Freiburg he received the Ph.D. degree in Electrical Engineering from the KU Leuven in November 2015. He is now a Post-Doc researcher at Chalmers University. His research interests include economic MPC, optimal control and estimation of nonlinear dynamic systems, in particular for aerospace and automotive applications.

Sébastien Gros received his Ph.D degree from EPFL, Switzerland, in 2007. After a journey by bicycle from Switzerland to the Everest base camp in full autonomy, he joined a R&D group hosted at Strathclyde University focusing on wind turbine control in 2010. In 2011, he joined the university of KU Leuven, where his main research focus was on optimal control and fast NMPC for complex mechanical systems. He joined the Department of Signals and Systems at Chalmers University of Technology, Göteborg in 2013, where he is now Associate Prof. His main research interests includes numerical methods, real-time optimal control, and the optimal control of complex mechanical systems and energy applications.

Paolo Falcone received his Ph.D. degree in Information Technology in 2007 from the University of Sannio, in Benevento, Italy. He is Associate Professor at the Department of Electrical Engineering of the Chalmers University of Technology, Sweden. His research focuses on constrained optimal control applied to autonomous and semi-autonomous mobile systems, cooperative driving and intelligent vehicles. He is involved in a several projects, in cooperation with industry, focusing on autonomous driving, cooperative driving and vehicle dynamics control.