Optimization-based Coordination of Connected, Automated Vehicles at Intersections

Robert Hult[a], Mario Zanon[b], Sébastien Gros[a,c], Henk Wymeersch[a], Paolo Falcone[a]

[a] Department of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden; [b]IMT Institute for Advanced Studies, Lucca, Italy; [c] Norwegian University of Science and Technology, Trondheim, Norway

ARTICLE HISTORY
Compiled March 28, 2019

ABSTRACT
In this paper, we introduce a model predictive controller for coordination of connected, automated vehicles at intersections. The problem has combinatorial complexity, and we propose to solve it approximately using a two stage procedure where 1) the order in which the vehicles cross the intersection is found by solving a mixed integer quadratic program and 2) the control commands are subsequently found by solving a nonlinear program. We show that the controller is perpetually safe and compare its performance to that of traffic lights and two simpler coordination controllers that share central characteristics with most existing work on the topic. The results show that our approach by far outperforms the considered alternatives in terms of both energy consumption and travel-time delay, especially for medium to high traffic loads.

KEYWORDS
Connected and Autonomous Vehicles, Model Predictive Control, Intelligent Transportation Systems

1. Introduction

Recent years have seen a rapid development in the field of automated driving (AD) following a surge of interest from automotive OEMs, software companies and the academic research community. In parallel, technologies for vehicular communication, via both direct radio links and the cellular network, continue to be improved and several standards have been adopted [1,2]. By combining AD technologies with communication, cooperative strategies can be implemented to augment the capabilities of automated vehicles, allowing them to both perform better and increase safety. The potential of such strategies was recognized over a decade ago, in the discussion following the 2007 DARPA Urban Challenge [3].

In this paper, we discuss a problem where cooperative strategies could play an important role: the coordination of connected and automated vehicles at intersections. The problem is motivated by the fact that intersections alone are the location of more than 20% of the traffic related fatalities and more than 40% of the injuries [4]. Furthermore, they tend to form bottlenecks in the traffic system and cause congestion, in-

CONTACT Robert Hult. Email: robert.hult@chalmers.se

creased emissions and energy-waste (e.g., through deceleration/acceleration and idling [5]). To alleviate the situation, a common strategy is to expand the infrastructure by e.g, adding more lanes, tunnels or overpasses. However, the road-traffic system already today claims a significant part of the ground surface in most urban areas. Hence, such expansions are at best undesirable and in many cases impossible.

The introduction of connected automated vehicles presents a potential remedy: rather than relying on traffic lights, road signs and right-of-way rules, the intersections could be managed completely by coordination algorithms to ensure that the vehicles can cross efficiently and without collisions. The central idea is to find control commands for the *individual vehicles* such that they cross the intersection without colliding, rather than trying to control the flows on the incoming roads. In the ideal case, cooperative AD has penetrated the marked completely and the vehicles could be coordinated to cross the intersection in virtually uninterrupted, tightly packed streams. Would this succeed, more vehicles could use the intersection simultaneously, traveling at higher velocities while spending less energy, thus increasing the capacity of the existing infrastructure and reducing the environmental footprint of the overall traffic system.

However, the design of such coordination algorithms is challenging for several reasons [6]. Uncertainties in the perception of the surroundings, must be handled, as well as impairments of the wireless communication channel [7]. Moreover, even if the sensors and communication system are ideal, the process of finding a solution to the coordination problem is in itself hard [8], where one particular difficulty is finding the order in which the vehicles should cross the intersection. This is particularly true if more than a few vehicles are involved and the algorithm is designed to be optimal in some metric, which necessitates exploration of the solution space. Finally, to account for uncertainties and to counter unexpected events, a coordination algorithm must be executed in closed-loop. That is, the coordination should be repeatedly updated with the most up-to-date information from the vehicles. Establishing that such closed-loop systems are stable and persistently safe is in general a difficult task.

Solving the intersection coordination problem

The problem of coordinating connected automated vehicles at intersections has been surveyed in [9,10]. Most existing contributions are focused on scenarios where all vehicles are automated, and disregards non-cooperative entities such as legacy vehicles or pedestrians. A large part of this work has been performed outside the control community, and has relied heavily on tailored heuristics [11–13]. However, the problem of coordinating vehicles at an intersection is fundamentally a constrained optimal control problem (OCP), as it involves the optimization of trajectories generated by dynamical systems, subject to (at least) collision avoidance constraints. A number of contributions have therefore come from the control community [14–35]. However, due to its combinatorial complexity, the problem is rarely solved in its full form, and most existing approaches are based on a mix of heuristics and optimal control formulations of smaller sub-problems. These heuristic schemes can roughly be categorized as *Sequential/Parallel* or *Simultaneous*, depending on what type of OCPs they involve.

In *Sequential/Parallel* schemes, a priority ranking of the vehicles is first decided. The solution is thereafter obtained by solving a number of smaller OCPs, commonly one per vehicle, where constraints are imposed to avoid collisions with higher priority vehicles. The ranking itself is typically the result of a heuristic, where common choices are variations of *first-come-first-served* (FCFS) policies.

In purely *Sequential* schemes such as [14,15] or the so-called *"MPC^*"* alternative of [16], the vehicles compute their solution in sequence based on a *decision order*, which implicitly reflects the priority. That is, each vehicle solves an OCP, constrained to avoid collisions with respect to the (already decided and available) solutions from the OCPs of vehicles preceding it in the decision order.

In *Parallel* schemes, the vehicle OCPs instead use predictions of the actually planned trajectories of higher priority vehicles. Along these lines, [17] proposes to use conservative estimates, based on predicted trajectories resulting from maximum braking maneuvers. With the so-called "MPC_0" solution, [16] instead suggests constant velocity predictions, whereas constant acceleration predictions are considered in [18]. Another alternative, suitable for receding horizon implementations, is to use the predicted trajectories of the higher priority vehicles from the previous time instant (see e.g. [19–22] and the so-called "MPC_1" alternative in [16]). If the priority ordering is constant between two time instants, this corresponds to a sequential solution with delayed information exchange. A scheme which uses both sequential and parallel components was suggested in [23]. There, a crossing time schedule is first constructed sequentially based on a FCFS policy, followed by the parallel solution of the vehicle OCPs for the state and control trajectories.

While they differ in aspects such as the objective function considered, the motion models and the formulation of collision avoidance conditions, the contributions in the *Sequential/Parallel* category are all "greedy". In particular, no vehicle ever takes decisions that are beneficial to the performance of the intersection scenario as a whole, if that decision is detrimental to the vehicle itself. As a consequence, the effort required to resolve difficult conflict is pushed "downwards" to lower priority vehicles. Although sub-optimal by design, these schemes can often be easily implemented in an almost completely decentralized fashion with low and accurately predictable requirements on both computation and information exchange.

In *Simultaneous* methods on the other hand, the solution is found through joint optimization of several vehicles' trajectories. However, to avoid the combinatorial complexity of the full coordination OCP, parts of the solution are typically still found using heuristics. In most schemes, this is done by first selecting the crossing order using a heuristic (often variations of FCFS policies), and thereafter jointly optimizing the trajectories of the vehicles for the given crossing order. Such *fixed-order* joint optimization was considered in [24–31]. Alternative approaches, e.g. [32,33] apply local continuous optimization methods to the full coordination OCP directly. The crossing order is thus selected by the optimizer, but dependent on the initial-guess provided to the solver. A few contributions propose to solve the full coordination OCP directly, and simultaneously optimize all aspects of the problem. For instance, both [34] and the benchmark discussed in [35] consider mixed integer quadratic programming (MIQP) formulations of the problem, returning both the optimal trajectories and crossing order. While such approaches are able to find globally optimal solutions, they typically scale poorly and can therefore only be applied to small problem instances.

While *Simultaneous* approaches in general optimize over a larger set of solutions than their *Sequential/Parallel* counterparts, their application is significantly more involved. In particular, since the joint problems must be solved iteratively, the solution is either computed with standard tools in a completely centralized fashion [24–26], or with iterative, distributed optimization algorithms [27–31] which relies on repeated communication between the vehicles and a central network node. As a result, the computational and communication requirements of *Simultaneous* approaches are in general higher and harder to predict accurately than those of *Sequential/Parallel* approaches.

A two-stage approximation scheme for simultaneous optimization

In this paper we evaluate the performance of a closed-loop algorithm directly derived from the full OCP formulation of the intersection problem, where the optimal solution is obtained by joint optimization of all parts of the problem, but performed in two stages. Similar to most other *Simultaneous* schemes, we first find the crossing order and thereafter solve a fixed-order OCP for the vehicle trajectories. However, contrary to the methods described above, the crossing order is found by solving an approximate, lower-dimensional representation of the full problem in the form of an MIQP, which approximately accounts the constraints and objective of the full problem.

We evaluate the closed-loop performance of the receding horizon application of the controller on a 4-way intersection by simulating the scenario under arrival rates ranging from 1000 to 2500 vehicles per lane and per hour. We compare the results to those obtained from for 1) an overpass solution where the roads are physically separated, 2) a traffic light controller, 3) a controller based on the sequential solution of OCPs, and 4) a controller where the crossing order is obtained through a *first-come-first-serve* heuristic and the trajectories are jointly optimized. The purpose of the comparisons is to establish 1) the loss induced by the proposed controller with respect to the overpass solution, 2) the gain with respect to the traffic light controller, and 3) the performance difference between the cases where nothing is optimized jointly, where only the trajectories are optimized jointly or where both the trajectories and the crossing order are optimized jointly.

This paper builds upon the problem formulation in [35], the MIQP based heuristic in [36] and uses the properties of the closed-loop controller established in [37]. The novel contributions of this paper are 1) The two-staged algorithm for closed-loop coordination, 2) The performance comparison with current coordination controllers (traffic lights and overpass), and 3) The performance comparison between coordination controllers of different complexity, illustrating the differences between sequential, partly simultaneous and fully simultaneous optimization.

We highlight in particular the comparison between sequential and joint optimization, which to the best of our knowledge has not been considered elsewhere. Details on how the method would be implemented in a practical setting and details on tailored optimization algorithms are beyond the scope of the paper, and we refer the interested reader to [27–30,36,37].

The remainder of the paper is organized as follows: In Section 2, we introduce and model the intersection scenario and present both a general optimal control formulation of the problem as well as a formulation suitable for receding horizon control. In Section 3 we detail the two-staged, closed-loop coordination algorithm, which approximately solves the receding horizon problem. In Section 4, we introduce the scenario on which we evaluate the performance, and detail the benchmarks considered. We discuss the results in Section 5, and conclude the paper in Section 6.

2. Optimal Coordination at Intersections

In this section we model the intersection scenario and state both a general optimal control formulation of the coordination problem as well as a discrete-time, finite horizon formulation, suitable for receding horizon control. Both the scenario modeling and the problem formulation is based on the following fundamental assumption

Assumption 1. *There are no non-cooperative entities present in the scenario.*

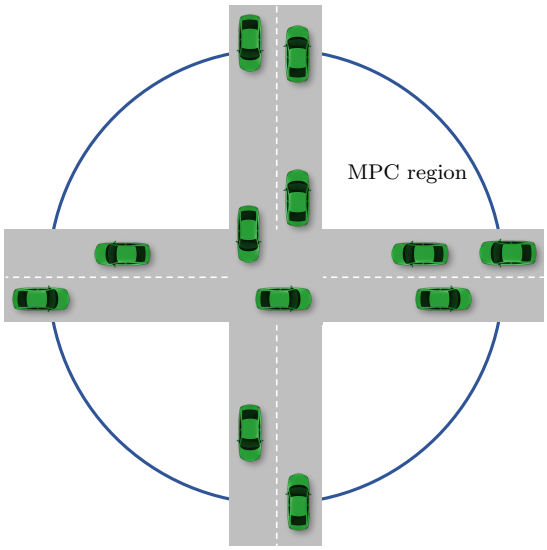


Figure 1.: Illustration of the intersection scenarios considered. The round circle illustrates that region in which a closed-loop coordination controller is applied.

That is, we do not consider scenarios with e.g. legacy vehicles, pedestrians or bicyclists. The assumption is restrictive and limits the applicability to traffic scenarios in a distant future, but is standard in the literature on vehicle coordination problems (see e.g. [13,14,23,38,39]).

2.1. Introduction to the optimal coordination problem

We consider cross-intersection scenarios, such as that shown in Figure 1, consisting of four incoming lanes with continuously arriving vehicles. The problem of finding the control commands that optimize a specified performance metric, reads as

$$\begin{aligned} \max_{\text{Control commands}} \quad & \text{Performance} && \text{(1a)} \\ \text{subject to} \quad & \text{Vehicle Dynamics initialized at Initial State} && \text{(1b)} \\ & \text{Physical and Design Constraints} && \text{(1c)} \\ & \text{Collision Avoidance.} && \text{(1d)} \end{aligned}$$

The solution to this problem are the control commands for all vehicles which (1c) satisfy all physical and design constraints (e.g., actuator limitations, comfort bounds, speed limits), (1b),(1d) result in collision free trajectories consistent with the vehicle dynamics and (1a) maximize the performance. Clearly, different performance metrics are possible, including minimization of energy consumption, minimization of travel-time and maximization of intersection throughput.

While (1) provides the open-loop optimal solution for a static scenario, a closed-loop, model predictive controller (MPC) can be obtained by solving a discrete-time, finite horizon approximation to (1) in a receding horizon fashion. In this setting, the approximate problem is solved periodically, based on the measured current state of all vehicles in the scenario, whereafter the first part of the optimal control is applied [40]. In a practical setting, the MPC coordination algorithm will only be applied to vehicles

that are sufficiently close to the intersection, as illustrated by the "MPC region" in Figure 1. However, we note that a receding horizon formulation naturally handles both vehicle arrivals and departures to this region, where a vehicle that is sufficiently far beyond the intersection simply is removed from the problem, and a vehicle that comes sufficiently close is added.

2.2. Scenario Modeling

In this subsection we model the intersection scenario, with the purpose of using the result in an optimal control formulation of the problem. We keep the description general, and return to the specifics used in the comparison in Section 5.

While one can model the motion of the vehicles in the intersection to an arbitrary accuracy, we employ the following fundamental assumption, which is standard in the literature [14,16,19,20,23,32]

Assumption 2. *The vehicles move along fixed and known paths along the road.*

Assumption 2 is not restrictive, as vehicles at intersections move along predefined lanes, and enables models which only describe the one-dimensional motion of a vehicle along its path. In particular, we consider models

$$\dot{x}_i(t) = f_i(x_i(t), u_i(t)), \tag{2a}$$
$$0 \geq h_i(x_i(t), u_i(t)), \tag{2b}$$

where i is the vehicle index, $x_i(t) \in \mathbb{R}^{n_i}$ and $u_i(t) \in \mathbb{R}^{m_i}$ are the vehicle state and control and both $f_i(\cdot)$ and $h_i(\cdot)$ are continuously differentiable. In particular, the models are such that $x_i(t) = (p_i(t), v_i(t), \tilde{x}_i(t))$, where $p_i(t)$ is the position of the vehicle's geometrical center on its path, $v_i(t)$ is the velocity along the path and, if applicable, $\tilde{x}_i(t)$ collects all remaining states (e.g. acceleration and/or internal states of the power-train).

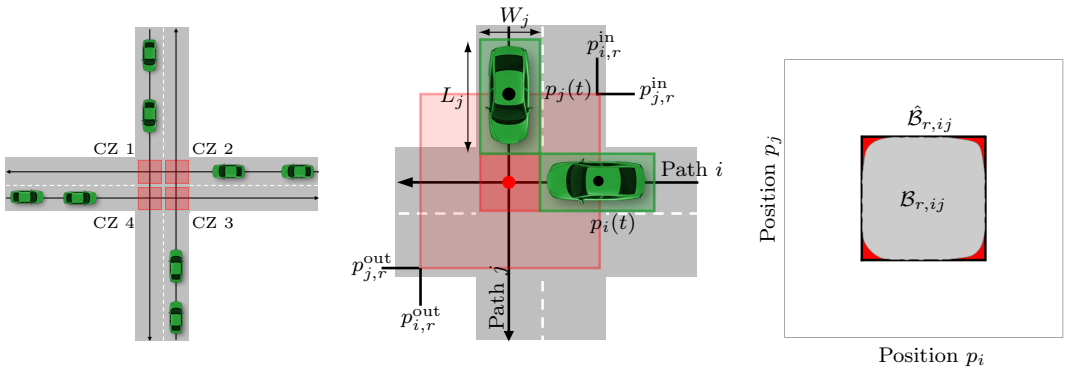
For convenience, we assume that no vehicle is allowed to make a turn inside the intersection, but remark that such problems can be easily tackled [41]. Similarly, the material is presented for single-lane roads, but we emphasize that the formalism can handle more general layouts.

2.2.1. Side Collision Avoidance (SICA) Conditions

As illustrated in Figure 2a, side-collisions can only occur between vehicles on different lanes, when these are inside an area around the points where the vehicle paths intersect. We denote these areas *Conflict Zones* (CZ), and note that more than one vehicle pair i, j can have potential conflicts at a particular CZ. In general, collisions between two vehicles i,j are avoided if

$$(x_i(t), x_j(t)) \notin \mathcal{B}_{r,ij} = \{(x_i, x_j) \mid \mathcal{G}_i(p_i) \cap \mathcal{G}_j(p_j) = \emptyset\}, \ \forall t \tag{3}$$

where $\mathcal{G}_i(p_i)$ is the area occupied by vehicle i in the horizontal plane when the path coordinate is p_i. However, rather than deriving a CA condition from the complicated vehicle geometry directly, a slightly conservative condition can be obtained using rectangular outer approximations $\hat{\mathcal{G}}_i(p_i) \supseteq \mathcal{G}_i(p_i)$. A requirement which ensures that



a Illustration of Assumption 2 and the Conflict Zones (red boxes).

b Illustration of the elements used in the SICA condition (4). The rectangular outer approximations of $\mathcal{G}_i(p_i(t))$ are the green boxes.

c Illustration of the conservativeness induced (red) when rectangular outer approximations are used.

Figure 2.: Illustrations of collision avoidance conditions.

$\mathcal{G}_i(p_i) \cap \mathcal{G}_j(p_j) = \emptyset$ at the r-th CZ can be formulated as

$$(x_i(t), x_j(t)) \notin \hat{\mathcal{B}}_{r,ij} = \left\{(x_i, x_j) \mid p_i \in [p^{\text{in}}_{r,i}, p^{\text{out}}_{r,i}],\ p_j \in [p^{\text{in}}_{r,j}, p^{\text{out}}_{r,j}]\right\}. \tag{4}$$

Here, $p^{\text{in}}_{r,i}$ and $p^{\text{out}}_{r,i}$ are the first and last position on the path of vehicle i for $\hat{\mathcal{G}}_i(p_i) \cap \hat{\mathcal{G}}_j(p_j) = \emptyset$ for all p_j at CZ r. An illustration of the components of (4) is provided in Figure 2b and a visualization of the conservativeness induced when the tightest rectangular outer approximation is used is given in Figure 2c. As the latter highlights, event hough $\mathcal{B}_{r,ij} \subseteq \hat{\mathcal{B}}_{r,ij}$, very little is normally lost by using approximate representations of the vehicle geometry, as most vehicles have an almost rectangular shape. Moreover, in a practical setting one would often strive to satisfy (4) with a margin, whereby the approximation would be of no consequence. This approach is taken in several works on intersection coordination (e.g. [14,20,23]), but is often formulated using auxiliary variables that describe the time of entry ($t^{\text{in}}_{r,i}$) and departure ($t^{\text{out}}_{r,i}$) of CZ r, which are defined implicitly through

$$p_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}, \quad \text{and} \quad p_i(t^{\text{out}}_{i,r}) = p^{\text{out}}_{r,i}.^{1} \tag{5}$$

Since by assumption $v_i(t) \geq 0$, $p^{\text{in}}_{r,i} < p^{\text{out}}_{r,i} \Rightarrow t^{\text{in}}_{r,i} < t^{\text{out}}_{r,i}$ and therefore $\left(t^{\text{out}}_{r,i} \leq t^{\text{in}}_{r,j}\right) \Rightarrow \neg\left(t^{\text{out}}_{r,j} \leq t^{\text{in}}_{r,i}\right)$, the following statement is equivalent to (4) and assures Side Collision Avoidance (SICA):

$$\left(t^{\text{out}}_{r,i} \leq t^{\text{in}}_{r,j}\right) \vee \left(t^{\text{out}}_{r,j} \leq t^{\text{in}}_{r,i}\right). \tag{6}$$

In words, (6) requires that either vehicle j must leave CZ r before vehicle i enters or vice-versa

[1] In the event that $v_i(t^{\text{in}}_{i,r}) = 0$, $t^{\text{in}}_{i,r}$ is not uniquely defined by $p_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}$. A practical remedy is to instead use the slightly more complex definition $t^{\text{in}}_{r,i} = \min t$ s.t. $p_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}$. Alternatively, one can modify (2) so that $\dot{p} \geq \epsilon$, for some small $\epsilon > 0$. Since $\dot{p}(t^{\text{in}}_{i,r}) = 0$ rarely will be encountered in practice, this is avoided in the main text for ease of presentation.

2.2.2. Rear-End Collision Avoidance (RECA) Conditions

Under Assumption 2, rear-end collisions can only occur between two adjacent vehicles on the same path, which translates to simple conditions on the vehicle positions. Denoting the length of vehicle i as L_i and $\delta_{ij} = L_i/2 + L_j/2$, a necessary condition for Rear-End Collision Avoidance (RECA) is

$$p_i(t) + \delta_{ij} \leq p_j(t), \tag{7}$$

when vehicle i is behind vehicle j. Condition (7) could be extended to include conservative (and more practical) distance keeping policies, e.g., fixed spacing policies with $\delta_{ij} = \epsilon_{ij} + L_i/2 + L_i/2$, for some $\epsilon_{ij} > 0$, or velocity dependent policies including a time headway.

2.3. Optimal Control Formulation

Using the modeling and notation introduced so far, we consider N vehicles that are to cross an intersection as in Figure 1. Starting at states $\hat{x}_{i,0}$, $\forall i \in \mathcal{N}$, $\mathcal{N} = \{1, \ldots, N\}$, the optimal state and control trajectories can then be computed by solving

$$\begin{aligned}
\min_{x(t),u(t),T} \quad & \sum_{i=1}^{N} \int_0^{\infty} \ell_i(x_i(t), u_i(t)) \mathrm{d}t, && && \text{(8a)}\\
\text{s.t.} \quad & x_i(0) = \hat{x}_{i,0}, && i \in \mathcal{N}, && \text{(8b)}\\
& \dot{x}_i(t) = f_i(x_i(t), u_i(t)), && i \in \mathcal{N}, && \text{(8c)}\\
& h_i(x_i(t), u_i(t)) \leq 0, && i \in \mathcal{N}, && \text{(8d)}\\
& p_i(t_{i,r}^{\text{in}}) = p_{r,i}^{\text{in}},\ p_i(t_{i,r}^{\text{out}}) = p_{r,i}^{\text{out}}, && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(8e)}\\
& p_i(t) + \delta_{ij} \leq p_j(t), && (i,j) \in \mathcal{C}_R. && \text{(8f)}\\
& \left(t_{r,i}^{\text{out}} \leq t_{r,j}^{\text{in}}\right) \vee \left(t_{r,j}^{\text{out}} \leq t_{r,i}^{\text{in}}\right), && (i,j,r) \in \mathcal{C}_S, && \text{(8g)}
\end{aligned}$$

Here, the continuously differentiable function $\ell_i(x_i(t), u_i(t))$ describes the performance metric to optimize, $x(t) = (x_1(t), \ldots, x_N(t))$, $u(t) = (u_1(t), \ldots, u_N(t))$, $T = (T_1, \ldots, T_N)$ and T_i collects $T_{r,i} = (t_{r,i}^{\text{in}}, t_{r,i}^{\text{out}})$, $\forall r \in \mathcal{R}_i$, with $\mathcal{R}_i$ being the CZ crossed by vehicle i. Moreover, $\mathcal{C}_S$ collects all vehicle-pairs and conflict zones (i, j, r) where side-collisions are possible and $\mathcal{C}_R$ collects all vehicle pairs (i, j) where rear-end collisions are possible.

We note that solving OCP (8) involves deciding the *crossing order* for all conflict zones, i.e., whether vehicle i should cross CZ r before vehicle j or the other way around for all $(i, j, r) \in \mathcal{C}_S$, which renders the problem combinatorial. Finally, we emphasize that the solution to (8) is *the* optimal open-loop solution of the coordination problem in terms of $\ell_i(x_i(t), u_i(t))$ under the stated assumptions, and that all subsequent developments in this paper and elsewhere introduce approximations.

2.4. Closed-Loop Control

A closed-loop-controller can be realized by iteratively solving a finite horizon approximation of (8) in a receding horizon-fashion [40]. To render the problem finite-dimensional, we consider control inputs that are constant over time intervals Δt, so

that $u_i(t) = u_{i,k}$, $t \in [t_k, t_{k+1}[$, for $t_k = k\Delta t$ and $u_{i,k} \in \mathbb{R}^{m_i}$. Letting the finite time horizon be $t_f = K\Delta t$, for some integer K, we note that $u(t)$, $t \in [0, t_f]$ thereby is described by $u_i = (u_{i,0}, \dots, u_{i,K-1})$. Starting from an initial condition $x_i(0) = \hat{x}_{i,0}$, this enables the recursion

$$x_{i,k+1} := x_i(t_{k+1}) = F_i(x_{i,k}, u_{i,k}, \Delta t), \tag{9}$$

where $F_i(x_{i,k}, u_{i,k}, \Delta t)$ denotes the solution to $\dot{x}_i(t) = f_i(x_i(t), u_i(t))$ at $t = \Delta t$, with $x_i(0) = x_{i,k}$ and $u_i(t) = u_{i,k}$. The state at an arbitrary time t can thus be written as $x_i(t) = F_i(x_{i,k}, u_{i,k}, \delta t)$, with $\delta t = t - t_k$ and $k = \lfloor t/\Delta t \rfloor$, whereby $x_i(t)$ is completely determined for all $t \in [0, t_f]$ by the vectors $x_i = (x_{i,0}, \dots, x_{i,K})$ and $u_i = (u_{i,0}, \dots, u_{i,K-1})$. In particular, letting $w_i = (x_i, u_i)$, we express the position $p_i(t)$ at an arbitrary time t as

$$p_i(t, w_i) = F_{i,p}(x_{i,k}, u_{i,k}, t - t_k), \ k = \lfloor t/\Delta t \rfloor, \tag{10}$$

where $F_{i,p}(\cdot)$ denotes the position component of the vector function $F_i(\cdot)$. Consequently, the time of entry, $t^{\text{in}}_{i,r}$, and time of exit, $t^{\text{in}}_{i,r}$, of all CZ are well defined functions of w_i through (5) and are not necessarily integer multiples of Δt. Finally, we only enforce the inequality constraints (8d) at the times $t_0, \dots, t_K$.

A discretized, finite horizon reformulation of OCP (8) then reads

$$\begin{aligned}
\min_{w,T} \quad & \sum_{i=1}^{N} J_i(w_i) && && \text{(11a)}\\
\text{s.t.} \quad & x_{i,k} = \hat{x}_{i,0}, && i \in \mathcal{N}, && \text{(11b)}\\
& x_{i,k+1} = F_i(x_{i,k}, u_{i,k}, \Delta t), && i \in \mathcal{N},\ k \in \mathcal{I}_{K-1}, && \text{(11c)}\\
& h_i(x_{i,k}, u_{i,k}) \le 0, && i \in \mathcal{N},\ k \in \mathcal{I}_{K-1}, && \text{(11d)}\\
& p_i(t^{\text{in}}_{i,r}, w_i) = p^{\text{in}}_{r,i}, && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(11e)}\\
& p_i(t^{\text{out}}_{i,r}, w_i) = p^{\text{out}}_{r,i}, && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(11f)}\\
& p_{i,k} + \delta_{ij} \le p_{j,k}, && (i,j) \in \mathcal{C}_R,\ k \in \mathcal{I}_K, && \text{(11g)}\\
& \left(t^{\text{out}}_{r,i} \le t^{\text{in}}_{r,j}\right) \vee \left(t^{\text{out}}_{r,j} \le t^{\text{in}}_{r,i}\right), && (i,j,r) \in \mathcal{C}_S, && \text{(11h)}
\end{aligned}$$

where we have gathered $w = (w_1, \dots, w_N)$ and used the notation $\mathcal{I}_a = \{0, \dots, a\}$, for an integer $a > 0$, and where

$$J_i(w_i) = V^{\text{f}}_i(x_{i,N}) + \sum_{k=0}^{K-1} \int_{t_k}^{t_{k+1}} \ell_i(x_i(t), u_{i,k}) \mathrm{d}t. \tag{12}$$

The terminal cost $V^{\text{f}}_i(x_{i,N})$ is assumed to be continuously differentiable and chosen so that an MPC based on (11) is stabilizing [37].

We note that as (8), Problem (11) is a combinatorial optimization problem, and while it can be cast and solved as a mixed integer nonlinear program (MINLP), finding exact solutions is in general intractable. Motivated by this fact, we detail next a method which gives approximate solutions to (11).

3. An Approximate Two-Stage Coordination Controller

In this section, we describe a closed-loop, receding horizon controller based on approximate solutions to (11). The approximation consists of a two-stage procedure where at each times instant

(1) The crossing order is found using an optimization-based heuristic,
(2) The state and control trajectories w that are optimal under that crossing order are found by solving a nonlinear program (NLP).

In particular, we employ an optimization-based heuristic in Stage 1, which approximately accounts for the objective and constraints in (11) when the order is selected.

In the remainder of this section, we detail the two stages and the closed-loop controller, starting with the NLP solved in Stage 2.

3.1. The fixed-order coordination problem

The *fixed-order* simplification of (11) reads

$$\begin{aligned}
\min_{w,T} \quad & \sum_{i=1}^{N} J_i(w_i) && && \text{(13a)}\\
\text{s.t.} \quad & x_{i,k} = \hat{x}_{i,0}, && i \in \mathcal{N}, && \text{(13b)}\\
& x_{i,k+1} = F_i(x_{i,k}, u_{i,k}, \Delta t) && i \in \mathcal{N},\ k \in \mathcal{I}_{K-1}, && \text{(13c)}\\
& h_i(x_{i,k}, u_{i,k}) \leq 0, && i \in \mathcal{N},\ k \in \mathcal{I}_{K-1}, && \text{(13d)}\\
& p_i(t^{\text{in}}_{i,r}, w_i) = p^{\text{in}}_{r,i} && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(13e)}\\
& p_i(t^{\text{out}}_{i,r}, w_i) = p^{\text{out}}_{r,i}, && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(13f)}\\
& p_{i,k} + \delta_{ij} \leq p_{j,k} && (i,j) \in \mathcal{C}_R,\ k \in \mathcal{I}_K. && \text{(13g)}\\
& t^{\text{out}}_{r,i} \leq t^{\text{in}}_{r,j} && (i,j,r) \in \mathcal{S}, && \text{(13h)}
\end{aligned}$$

where the difference from (11) is the replacement of constraint (11h) with the simple inequality (13h). Here, the (given) crossing order is denoted by $\mathcal{S}$, and $(i,j,r) \in \mathcal{S}$ means that vehicle i crosses CZ r before vehicle j.

We emphasize that the removal of constraints (11h), render (13) a continuous NLP. However, even though the crossing order is provided externally, the exact time of entry $t^{\text{in}}_{i,r}$ and exit $t^{\text{out}}_{i,r}$ for each vehicle i and each CZ r must still be determined. Since $t^{\text{in}}_{i,r}$ and $t^{\text{out}}_{i,r}$ depend nonlinearly on w_i through (13e) and (13f), (13) is a non-convex NLP, even when the dynamics are linear, the path constraints affine and the objective function convex. It is also worth noting that due to the non-convex nature of (13), (11) is a non-convex MINLP even in the simplest case.

3.2. An MIQP Based Crossing Order Heuristic

In order for the proposed two-stage scheme to perform well, the crossing order $\mathcal{S}$ must be selected so that the solution to (13) in Stage 2 is good in terms of (13a). This means that $\mathcal{S}$ should be found in a manner that accounts for the objective (11a) and constraints (11b)-(11h), ideally exactly. However, this essentially amounts to solving (11), which is what we aim to avoid in the first place.

To circumvent this conundrum we introduce a reduced formulation of (11), where all couplings between the vehicles outside the intersection are removed, but those relevant *inside* the intersection are formulated in a lower-dimensional space. Since this problem also takes a form of a non-convex MINLP, we construct an approximating MIQP to retrieve the crossing order $\mathcal{S}$.

While the solution space of the MIQP suffers from the same combinatorial growth as the original MINLP (11), efficient solvers are available which make it possible to solve problems of practically relevant size fast enough for the application.

3.2.1. A reduced OCP formulation

The reduced formulation of the problem is founded on two simplifications of (11). First, rather than considering the RECA constraint (11g) at all times $t_k, k = 1, \dots, K$, we enforce it only at a single time. Second, we remove the possibility of choosing all $t_{r,i}^{\text{in}}, t_{r,i}^{\text{out}}$ freely, and instead fix the relationship between the components of T_i, whereby the degrees of freedom in the problem are reduced. The purpose is to obtain a decomposed problem formulation, where the entire trajectory of each vehicle is determined by a scalar parameter t_i^{p}, and solve that problem for $T^{\text{p}} = (t_i^{\text{p}}, \dots, t_i^{\text{p}}) \in \mathbb{R}^N$.

To this end, we introduce the *vehicle-problem*

$$
\begin{aligned}
V_i(t_i^{\text{p}}) = \min_{w_i} \quad & J_i(w_i) && && \text{(14a)}\\
\text{s.t.} \quad & x_{i,k} = \hat{x}_{i,0}, && && \text{(14b)}\\
& x_{i,k+1} = F_i(x_{i,k}, u_{i,k}, \Delta t), && k \in \mathcal{I}_{K-1}, && \text{(14c)}\\
& h_i(x_{i,k}, u_{i,k}) \leq 0, && k \in \mathcal{I}_{K-1}, && \text{(14d)}\\
& p_i(t_i^{\text{p}}, w_i) = p_i^{\text{p}}, && && \text{(14e)}
\end{aligned}
$$

where, $p_i^{\text{p}} \in \mathbb{R}$ is a position on the vehicles coordinate inside the intersection, and the parameter $t_i^{\text{p}} \in \mathbb{R}$ is a time. The solution to (14) is consequently the optimal solution for one vehicle, given that it is at position p_i^{p} at time t_i^{p}. We note that the entry and departure time constraints of (11e),(11e) are absent from (14) but that (14e) is a constraint of the same type.

We denote the minimizer of (14) $w_i^*(t_i^{\text{p}})$, and note that it is a continuous function of t_i^{p} under some mild assumptions (see [27] for more details). The time of entry and departure from each CZ $r \in \mathcal{R}_i$ associated with $w_i^*(t_i^{\text{p}})$ are therefore implicit functions of t_i^{p}, defined by the solution to

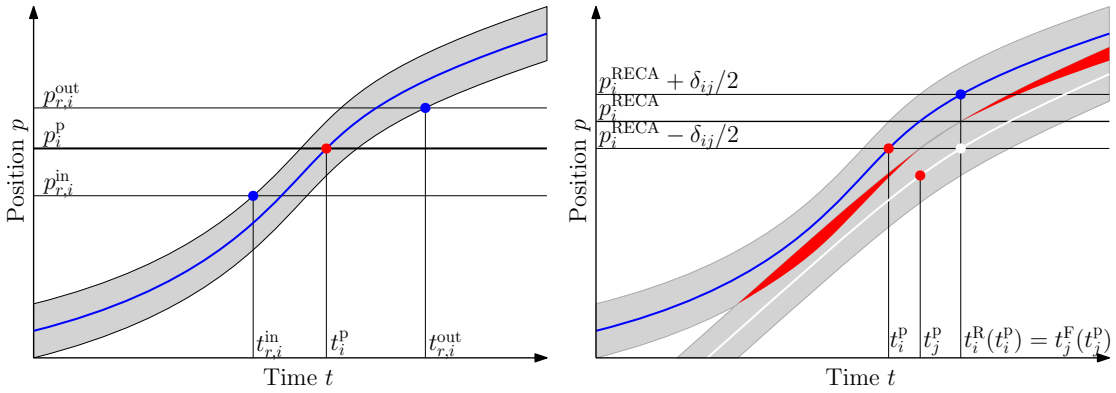
$$
p_i(t_{r,i}^{\text{x}}, w_i^*(t_i^{\text{p}})) = p_{r,i}^{\text{x}}, \tag{15}
$$

with $\text{x} \in \{\text{in}, \text{out}\}$. Figure 3b illustrates this relationship, where the vehicle is forced to be at p_i^{p} at t_i^{p} (red dot) and the time of entry and departure can be retrieved from the result (blue dots).

Similarly, we implicitly define the functions $t_i^{\text{R}}(t_i^{\text{p}})$ and $t_i^{\text{F}}(t_i^{\text{p}})$ as solutions to

$$
\begin{aligned}
p_i(t_i^{\text{R}}, w_i^*(t_i^{\text{p}})) &= p_i^{\text{RECA}} + L_i/2, && \text{(16a)}\\
p_i(t_i^{\text{F}}, w_i^*(t_i^{\text{p}})) &= p_i^{\text{RECA}} - L_i/2, && \text{(16b)}
\end{aligned}
$$

so that t_i^{R} is the time when the rear bumper of vehicle i is at p_i^{RECA} and t_i^{F} is the time when the front bumper of vehicle i is at p_i^{RECA}, associated with the solution to



a Illustration of the implicit dependence of T_i on $w_i^*(t_i^{\text{p}})$. The blue line is $p_i(t, w_i^*(t_i^{\text{p}}))$ and the gray area includes the length of $\hat{\mathcal{G}}_i(p_i)$, corresponding to the vehicle size.

b Illustration of constraint (17). While rear-end collisions are avoided at p_i^{RECA}, the solutions to (14) that satisfies (17) can correspond to collisions at other positions, shown as the red fields in the figure.

Figure 3.: Illustration of approximations made in the reduced problem formulation.

(14) using the parameter t_i^{p}. Choosing $p_i^{\text{RECA}} = p_j^{\text{RECA}}$ for vehicles i and j, where j is directly in front of i and on the same path, the condition

$$t_i^{\text{F}}(t_i^{\text{p}}) - t_j^{\text{R}}(t_j^{\text{p}}) \leq 0 \tag{17}$$

thereby ensures that there is no rear-end collision (c.f. RECA condition (7)) at the position p_i^{RECA}. However, selecting t_i^{p} and t_j^{p} satisfying (17) does not ensure that the RECA condition (7) holds at other positions, since trajectories resulting from (14) are computed without considering (7). This is illustrated in Figure 3b, which shows solutions to (14) for vehicles i and j, with t_i^{p} and t_j^{p} selected in satisfaction of (17), but where (7) is violated at other times (areas marked in red).

Finally, we note that not all t_i^{p} are feasible in (14). In particular, as reported in [36], the domain of $V_i(t_i^{\text{p}})$ is an interval $[t_i^{\text{p,min}}, t_i^{\text{p,max}}]$ corresponding to maximally accelerating and decelerating input commands respectively.

Collecting all implicit functions defined by (15) for vehicle i in $q_i(t_i^{\text{p}})$, we introduce the *timeslot allocation* problem

$$\begin{aligned}
\min_{T, T^{\text{P}}} \quad & \sum_{i=1}^{N} V_i(t_i^{\text{p}}) && && \text{(18a)}\\
\text{s.t.} \quad & t_i^{\text{p}} \in [t_i^{\text{p,min}}, t_i^{\text{p,max}}], && i \in \mathcal{N}, && \text{(18b)}\\
& T_i = q_i(t_i^{\text{p}}), && i \in \mathcal{N}, && \text{(18c)}\\
& \left(t_{r,i}^{\text{out}} \leq t_{r,j}^{\text{in}}\right) \vee \left(t_{r,j}^{\text{out}} \leq t_{r,i}^{\text{in}}\right), && (i,j,r) \in \mathcal{C}_S, && \text{(18d)}\\
& t_i^{\text{F}}(t_i^{\text{p}}) \leq t_j^{\text{R}}(t_j^{\text{p}}), && (i,j) \in \mathcal{C}_R. && \text{(18e)}
\end{aligned}$$

The solution to (18) consists of occupancy timeslots that minimize (18a), satisfy all SICA constraints and all RECA constraints at positions p_i^{RECA}, $\forall i \in \mathcal{N}$. A crossing order which could be used for the solution of the fixed-order problem (13) can consequently be extracted from the solution to (18). However, (18) is a non-convex MINLP and not less difficult to solve than (11). For this reason, we propose to instead utilize

an MIQP approximation of MINLP (18).

3.2.2. A timeslot allocation MIQP

By performing first and second order expansions of the functions in the timeslot allocation problem (18) around some $\tilde{T}^{\mathrm{p}} = (\tilde{t}_1^{\mathrm{p}}, \dots, \tilde{t}_N^{\mathrm{p}})$, the following *approximate timeslot allocation* problem can be constructed

$$\min_{T, T^{\mathrm{p}}} \quad \sum_{i=1}^{N} \frac{1}{2} \frac{\mathrm{d}^2 V_i}{\mathrm{d}t_i^{\mathrm{p}2}} (t_i^{\mathrm{p}})^2 + \left(\frac{\mathrm{d}V_i}{\mathrm{d}t_i^{\mathrm{p}}} - \frac{\mathrm{d}^2 V_i}{\mathrm{d}t_i^{\mathrm{p}2}} \tilde{t}_i^{\mathrm{p}} \right) t_i^{\mathrm{p}} \tag{19a}$$

$$\text{s.t.} \quad t_i^{\mathrm{p}} \in [t_i^{\mathrm{p,min}}, t_i^{\mathrm{p,max}}], \qquad i \in \mathcal{N}, \tag{19b}$$

$$T_i = q_i(\tilde{t}_i^{\mathrm{p}}) + \frac{\mathrm{d}q_i}{\mathrm{d}t_i^{\mathrm{p}}} (t_i^{\mathrm{p}} - \tilde{t}_i^{\mathrm{p}}), \qquad i \in \mathcal{N}, \tag{19c}$$

$$\left(t_{r,i}^{\mathrm{out}} \leq t_{r,j}^{\mathrm{in}}\right) \vee \left(t_{r,j}^{\mathrm{out}} \leq t_{r,i}^{\mathrm{in}}\right), \qquad (i,j,r) \in \mathcal{C}_S \tag{19d}$$

$$t_i^{\mathrm{F}}(t_i^{\mathrm{p}}) + \frac{\mathrm{d}t_i^{\mathrm{F}}}{\mathrm{d}t_i^{\mathrm{p}}} (t_i^{\mathrm{p}} - \tilde{t}_i^{\mathrm{p}}) \leq t_j^{\mathrm{R}}(t_j^{\mathrm{p}}) + \frac{\mathrm{d}t_j^{\mathrm{R}}}{\mathrm{d}t_j^{\mathrm{p}}} (t_j^{\mathrm{p}} - \tilde{t}_j^{\mathrm{p}}), \qquad (i,j) \in \mathcal{C}_R, \tag{19e}$$

where all derivatives for vehicle i are evaluated at $\tilde{t}_i^{\mathrm{p}}$.

Problem (19) and is thus an approximation of (18), constructed in a fashion equivalent to how the QP sub-problems are formed in sequential quadratic programming [42], and can be solved as an MIQP.

If it exists, the solution to (19) consists of a non-overlapping timeslot schedule T^* which encodes a crossing order $\mathcal{S}$. The solution is optimal in the quadratic approximation of (18a), which in turn approximately represents the objective function in (11), and belongs to a polytopic approximation of the feasible set of (18a), which is a lower dimensional approximate representation of the feasible set of (11). The corresponding solutions $w_i^*(t_i^{\mathrm{p}*})$ to the vehicle problems (14) using the minimizer $T^{\mathrm{p}*} = (t_1^{\mathrm{p}*}, \dots, t_N^{\mathrm{p}*})$ of (19) gives an approximate (and possibly infeasible) solution w^* to (11). Specifically, it is approximate since $w_i^*(t_i^{\mathrm{p}*})$ must be optimal solutions to the vehicle problems (14) (which excludes possible solutions to (18)) and since the RECA constraints (7) only are enforced approximately and at a single position (which allows solutions that are infeasible in (18), see Figure 3b).

However, while $T^{\mathrm{p}*}$ is such that $w_i^*(t_i^{\mathrm{p}*})$ likely is infeasible in (11), this is of less importance. The important outcome of (19) is instead the crossing order $\mathcal{S}$ extracted from T^*, since feasibility can be restored through the solution of the fixed order problem (13). In fact, if the crossing order $\mathcal{S}$ is the optimal order for the original problem (11), solving NLP (13) to global optimality gives the optimal solution to MINLP (11).

A note on derivative calculation The objective function (18) consists of the optimal value functions V_i from the NLP (14), and its derivatives can be obtained using tools from parametric sensitivity analysis under some mild assumptions [43]. The first-order derivatives of the implicit functions defined through (15), (16) are obtained through the implicit function theorem and have the form

$$\frac{\mathrm{d}t_i^{\mathrm{R}}}{\mathrm{d}t_i^{\mathrm{p}}} = -\left(\frac{\partial p(t_i^{\mathrm{R}}, w_i^*)}{\partial t_i^{\mathrm{R}}} \right)^{-1} \frac{p(t_i^{\mathrm{R}}, w_i^*)}{\partial w} \frac{\mathrm{d}w^*}{\mathrm{d}t_i^{\mathrm{p}}}, \tag{20}$$

Algorithm 1 Execution of coordination controller at time t_k. The controller is applied to the vehicles listed in $\mathcal{C}$.

procedure CoordinationController
  for all vehicles i **do**
    Measure current state $\hat{x}_{i,0}$
  end for
  Remove crossed vehicles from $\mathcal{C}$.
  Add new vehicles in IZ to $\mathcal{C}$
  for vehicle $i \in \mathcal{C}$ **do**
    Solve (14) using $\hat{x}_{i,0}, t_i^{\mathrm{p}} = \tilde{t}_i^{\mathrm{p}}$
    Compute derivatives used to build MIQP
  end for
  Solve MIQP (19), extract order $\mathcal{S}$
  Solve fixed-order problem (13) using $\mathcal{S}$
  Apply optimal controls $u_{i,0}$, $\forall i \in \mathcal{C}$
end procedure

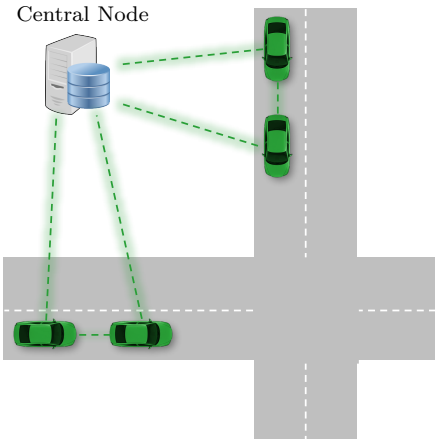


Figure 4.: Illustration of the communication pattern of a practical implementation of Algorithm 1.

where $\frac{\mathrm{d}w^*}{\mathrm{d}t_i^{\mathrm{p}}}$ is the first-order sensitivity of the solution w_i^* w.r.t. variations in t_i^{p}. Provided that a second-order method is used to solve (14), these sensitivities are available at little additional computational expense. For further details on the sensitivity calculations required to construct (19), see [27,36].

3.3. Closed Loop Controller

The closed-loop application of the proposed procedure is summarized in Algorithm 1 and executed as follows. At each time t_k, the vehicles' states are measured, and only those that are sufficiently close to the intersection are selected for inclusion in the coordination procedure (c.f. the MPC Zone of Figure 1). The MIQP is then built (by solving the vehicle problem (14) for each considered vehicle), and solved for the crossing order $\mathcal{S}$. This is followed by the solution of the fixed order problem (13), which gives the control commands $u_{i,0}$ that subsequently are applied to the vehicles.

3.3.1. Perpetual Safety

Due to the safety critical nature of the application, its paramount that the closed-loop controller does not bring the vehicles to configurations from which collisions are unavoidable, i.e., that it is perpetually safe. However, we note that the two-stage procedure can fail when a) MIQP (19) is infeasible, or b) when the MIQP is feasible and the crossing order $\mathcal{S}$ can be computed, but where no solution exists to the fixed-order problem (13) for $\mathcal{S}$. While we observed neither in simulation, a safe-guard mechanism for such failures can be implemented.

Below, we formalize the perpetual safety of the two-stage controller including the safe-guard mechanism. The results hold under the following assumptions, where $\mathcal{C}$ denotes the set of vehicles for which the coordination controller is applied

Assumption 3. *When a vehicle i is added to $\mathcal{C}$ at time t_k, its state $x_{i,k}$ is such that*

(1) It can stop before the intersection. That is

$$\exists\{u_{i,k+l}\}_{l=0}^{\infty} \ : \ p_{i,k+l} \leq \min_{r\in\mathcal{R}_i} p_{r,i}^{\text{in}}, \ \forall l, \tag{21}$$

(2) It can avoid collisions to the vehicle directly in front on the same lane. That is

$$\exists\{u_{i,k+l}\}_{l=0}^{\infty} \ : \ p_{i,k+l} + \delta_{ij} \leq p_{j,k+l}, \ \forall l, \tag{22}$$

$\forall\{p_{j,k+l}\}_{l=0}^{\infty}$ *such that* $x_{j,k+l+1} = F_j(x_{j,k+l}, u_{j,k+l}, \Delta t)$ *and* $h_j(x_{j,k+l}, u_{j,k+l}) \leq 0$,
In both cases, $x_{i,k+l+1} = F_i(x_{i,k+l}, u_{i,k+l}, \Delta t)$ *and* $h_i(x_{i,k+l}, u_{i,k+l}) \leq 0$.

Assumption 3 part (1) requires a vehicle to execute a local control law which ensures that it can stop before the intersection or, alternatively, that the zone in which the coordination controller is applied is large enough for a given vehicle velocity and control authority. Neither is restrictive. Assumption 3 part (2) requires that the vehicles that have not yet been included in the coordination execute local controllers to ensure that they remain safe with respect to the vehicle directly in front. Since this is a general requirement for all (automated) vehicles, it is not restrictive. Furthermore, we assume that the vehicles that are removed from $\mathcal{C}$ simply can be ignored, but note that this would not be possible in a practical setting (the removed vehicles can take subsequent actions that constrain the vehicles remaining in $\mathcal{C}$). While this assumption could be relaxed by introducing appropriately chosen constraints for the vehicles in $\mathcal{C}$, this is outside the scope of this paper. With this, we state the following result

Proposition 3.1 ("Nominal" Perpetual Safety)**.** *If Assumption 3 holds and the approximate timeslot allocation problem* (19) *and the fixed order problem* (13) *is feasible at all times, the system is perpetually safe.*

Proof. Perpetual safety when $\mathcal{C}$ is constant was established in [Proposition 5, [37]]. If a vehicle is removed from $\mathcal{C}$, the set of feasible solutions for the remaining vehicles cannot be smaller than in the case of constant $\mathcal{C}$, and does therefore not jeopardize perpetual safety. When a vehicle is added to $\mathcal{C}$, Assumption 3 ensures that it can execute at least one collision free trajectory. Therefore, the closed-loop application of the two-staged controller is perpetually safe. □

Due to Proposition 3.1 we can conclude that if the fixed-order problem (13) is feasible for an order $\mathcal{S}$ at time t_k, it will be feasible at t_{k+1} for a) the same crossing order used at t_k if $\mathcal{C}$ is constant or a vehicle removed b) a vehicle is added to $\mathcal{C}$ and the order updated so that it crosses the intersection after all other vehicles in $\mathcal{S}$. The latter option is always feasible since the added vehicle can stop before the intersection due to Assumption 3. Therefore, if at t_{k+1} either the calculation of a new crossing order fails (infeasible MIQP (19)) or if the crossing order is updated such that fixed-order problem (13) is infeasible, a safe-guard mechanism can be implemented where the fixed-order problem (13) is re-solved using the crossing order from t_k.

We therefore state the following result, where the proof follows from Proposition 3.1 and the reasoning above:

Theorem 3.2 (Perpetual Safety)**.** *If Assumption 3 holds and the safe-guard mechanism is employed, the system is perpetually safe.*

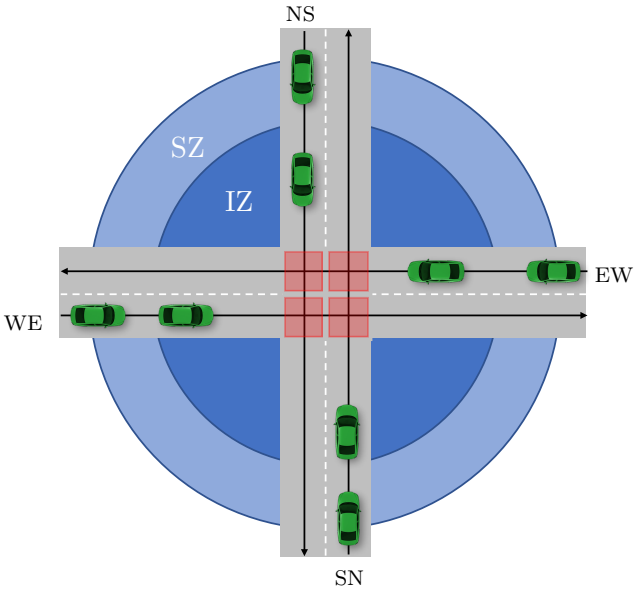


Figure 5.: Scenario for the performance evaluation.

A note on stability We note that conditions for stability for problems where $\mathcal{C}$ is constant was established in [Theorem 6, [37]]. While we haven't observed any issues, the extension to problems with changing $\mathcal{C}$ is the subject of current research.

3.3.2. A note on practical application

In a practical setting, much of the computations involved can be performed in a parallel and distributed fashion. In particular, while the MIQP is solved at a central node, the problem data (Solution of (14) on Line 11 of Algorithm 1) can be computed on-board the vehicles, and be made available using communication. Moreover, the subsequent solution of the fixed-order problem on Line 12 of Algorithm 1 can also be distributed, using one of several existing methods [27,28,30]. These consist of iterative procedures where data is communicated between the vehicles and a central node, following the pattern shown in Figure 4. Communication is also required on Lines 2-6 to determine the set of vehicles $\mathcal{C}$.

4. Evaluation Scenario and Compared Controllers

In this section, we describe the scenario in which the proposed controller is evaluated, and introduce the alternative coordination controllers used in the comparisons.

The scenario consists of the two-road intersection shown in Figure 5, with one lane in each direction (East-West (EW), West-East (WE), North-South (NS) and South-North (SN)). There are consequently 4 Collision Zones in the problem, where side-collisions can occur between vehicles on crossing paths, and rear-end collisions can occur between neighboring vehicles on each lane.

As shown in Figure 5, we divide the area around the intersection in two zones, where vehicles are added and removed from the scenario around the beginning and end of the Scenario Zone (SZ), and the coordination is performed for vehicles inside the Intersection Zone (IZ). That is, the scenario is such that all vehicles travel between

their generation point to the border of the IZ without performing any coordinating action, and the different coordination controllers are only applied once the vehicles enters the IZ. In particular, we consider symmetric SZ and IZ, where we denote the entry and departure positions of the SZ as p^{e}_{SZ} and p^{d}_{SZ}, respectively, and similarly denote the counterparts for the IZ as $p^{\mathrm{e}}_{\mathrm{IZ}}$ and $p^{\mathrm{d}}_{\mathrm{IZ}} = p^{\mathrm{d}}_{SZ}$.

4.1. Vehicle Arrival and Removal

The arrival of new vehicles to the scenario is roughly modeled as a Poisson Point Process (PPP)[2] . In particular, we let the time d between each added vehicle on a lane be drawn from the exponential distribution

$$d \sim \lambda e^{-\lambda d}, \tag{23}$$

with rate parameter λ, for practical reasons truncated to the interval $[0, d^{\max}]$. That is, new vehicles are added randomly according to (23), but with a time spacing of at most $d^{\max}$ s. When a vehicle is added to the scenario at a time t^{e}_i, it is added with initial velocity v^{e} at the position

$$p_i(t^{\mathrm{e}}_i) = \min(p_{SZ}, p_j(t^{\mathrm{e}}_i) - \delta p^{\mathrm{safe}}), \tag{24}$$

where $p_j(t^{\mathrm{e}}_i)$ is the position of the vehicle directly in front of vehicle i on the same lane, and δp^{safe} is the smallest distance such that a rear-end collision can be avoided if vehicle j brakes to its fullest capacity when at v^{e}. That is, the vehicles are always added to the scenario in positions at which they can avoid rear-end collisions with the forward vehicle on the same lane, provided that that vehicle travels with velocity v^{e}. Finally, the scenario is symmetric, with the same λ for all lanes, and includes both passenger *Cars* and *Trucks*, where the vehicle type is drawn randomly on generation with probabilities p^{Car}, p^{Truck}, respectively. A vehicle is removed from the scenario when it passes the departure position of the SZ, $p^{\mathrm{d}}_{\mathrm{SZ}}$.

We denote the time of generation, type, and position of generation for all vehicles introduced in the scenario over a simulation length S as the *generation pattern*. To enable a fair comparison, the same generation pattern is used on all controllers when the performance is evaluated for a specific value of the rate parameter λ.

4.2. Evaluated Controllers

We denote the two-stage procedure introduced in Section 3 as the *MIQP/Fixed Order* (MIQP/FO) controller and compare it to the following four strategies.

4.2.1. Overpass

The *Overpass* "controller" gives the solution where no coordinating action is taken when the vehicles are inside the IZ (and side collisions can occur). Since the vehicles are generated at a safe inter-vehicle distance, all vehicles instead travel with the initial velocity v^{e} until the end of the SZ. This corresponds to a physical separation of the roads, and constitutes the first benchmark for the coordination controller comparison

[2]Due to the truncation of the distribution and the minimum-safe distance in (24) the time between arrivals will not be drawn from the standard exponential distribution. Consequently, the vehicle arrivals are not really a PPP. We emphasize that the exact distribution of arrivals is of less importance.

4.2.2. Traffic Light

The fixed-cycle *Traffic Light* controller serves as a surrogate for an actual traffic light for scenarios with human or non-cooperative autonomous vehicles, and forms the second benchmark for the comparison. In this scheme, the red and green phases of the two directions (NS/SN and EW/WE) alternate with cycle time C, without an intermediate yellow phase. The vehicles are assumed to know both the times for all phase-shifts as well as the intended trajectory of the vehicle directly in front of it. As a consequence, all vehicles move synchronously from stand still after a red-light phase has passed, but in a manner which minimizes $J_i(w_i)$ and satisfies $h_i(x_{i,k}, u_{i,k})$. However, no vehicle takes any action to favor other vehicles, i.e. there are no accelerations in order to allow rearward vehicles to cross the intersection within a green phase.

4.2.3. The Sequential Controller

In the *Sequential* controller case, the vehicles decide how to cross the intersection in sequence (c.f. the discussion on sequential schemes in the introduction) based on a priority ranking. In this context, a decision consists of the full state trajectory of a vehicle from the first state such that the vehicle position $p_{i,k} > p_{\mathrm{IZ}}^{\mathrm{e}}$ until the end of the SZ is reached. The controller is executed as follows: when a single vehicle enters the IZ at time t_k, it forms its decision by finding the dynamically feasible state trajectory which minimizes the objective function, satisfies the path constraints and avoids collisions w.r.t. the (already formed) decisions of higher priority vehicles. If more than one vehicle enters the IZ at t_k, the decisions are formed in sequence based on the estimated time of arrival to the intersection when all SICA constraints are ignored. Note that as in the Traffic Light controller case, the vehicles do not perform any action for the benefit of other vehicles.

4.2.4. The FCFS/FO Controller

In the *First-Come-First-Served/Fixed-Order (FCFS/FO)* coordination controller, the Fixed Order (FO) Problem (13) is solved in a receding horizon fashion for all vehicles in the IZ, using a crossing order selected through a FCFS heuristic. In particular, if a single vehicle enters the IZ at time t_k it is required to yield to all vehicles already in the IZ. If more than one vehicle enters the IZ at time t_k, they are sorted based on their expected arrival to the intersection when the SICA constraints are ignored, and added to the crossing order accordingly.

The FCFS/FO controller has similarities with the Sequential controller in that a similar ordering policy is used. However, as opposed to the latter, the control commands for each vehicle are found through simultaneous optimization of all vehicles in the scenario, similar to the MIQP/FO case. As a result, some vehicles might take actions that increase their own objective functions, but yields a decrease for the scenario as a whole.

4.3. Motion Models and Control Objectives

All controllers use the double integrator longitudinal dynamics with input bounds as prediction model, such that

$$F_i(x_{i,k}, u_{i,k}, \Delta t) = \underbrace{\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}}_{A_i} x_{i,k} + \underbrace{\begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix}}_{B_i} u_{i,k}, \quad a_i^{\min} \leq u_{i,k} \leq a_i^{\max}. \tag{25}$$

Moreover, the vehicles are assumed to follow the control command perfectly, i.e. there is no model-plant mismatch. The control objective is taken to be

$$J_i(w_i) = m_i \left(Q_i^{\mathrm{f}}(v_{i,K} - v^r)^2 + \sum_{k=0}^{K} Q_i(v_{i,k} - v^r)^2 + R_i u_{i,k}^2 \right) \tag{26}$$

where K is the prediction horizon length, $Q_i^{\mathrm{f}}, Q_i, R_i$ are objective weights, $v^{\mathrm{r}} = v^{\mathrm{e}}$ is a reference velocity and m_i is the vehicle mass.

Although the prediction model is simple, experimental results indicate that it is sufficient for the application [37].

4.4. Secondary Performance Objectives

Two often cited reason for using coordination controllers is the reduction of energy consumption and travel time [14,20,23]. While not explicitly stated in the control objective (26), we also assess the performance of the coordination controllers w.r.t these quantities.

Travel Time Delay: Travel time delay is evaluated by comparing the time required for a vehicle to leave the SZ using a solution resulting from the coordination controllers, to the time required to cover the same distance by keeping the initial velocity v^{e}, i.e., the Overpass Case. In particular, the travel-time delay for vehicle i is

$$\delta t_i = t_i^{\mathrm{d}} - t_i^{\mathrm{e}} - \frac{p_i(t_i^{\mathrm{d}}) - p_i(t_i^{\mathrm{e}})}{v^{\mathrm{e}}} \tag{27}$$

where t^{d} is the time of departure from the SZ.

Energy Consumption: Since the state and input sequences for the simple kinematic model (25) does not contain enough information to compute the energy supplied to a vehicle, we assess the total energy consumption indirectly. In particular, for the state sequence $x_i = (x_{i,k_i^{\mathrm{e}}}, \ldots, x_{i,k_i^{\mathrm{d}}})$, generated by a coordination controller using the model (25) between the discrete times of SZ entry k_i^{e} and departure k_i^{d}, we investigate the energy consumed by a vehicle with an electric power-train when it follows x_i exactly. To this end, we introduce the simple Electric Vehicle (EV) model

$$\dot{p}_i(t) = v_i(t), \tag{28a}$$

$$\dot{v}_i(t) = \frac{1}{m_i} \left(\frac{G_i}{r_i^{\mathrm{w}}} M_i(t) - b_i(t) - \frac{1}{2}\rho a_i c_i^{\mathrm{d}} v_i(t)^2 - m_i g c_i^{\mathrm{rr}} \right), \tag{28b}$$

where $M_i(t)$ is torque delivered by the electric motor and $b_i(t)$ the friction-brake force. The model parameters are: the fixed gear-ratio G_i, the wheel radius r_i^{w}, the air density ρ, the projected frontal surface area a_i, the acceleration due to gravity g and the air-drag and rolling resistance coefficients $c_i^{\mathrm{d}}, c_i^{\mathrm{rr}}$. We assume that the EV input $u_i^{\mathrm{EV}}(t) = (M_i(t), b_i(t))$ is piece-wise constant over time-intervals Δt, and determine the EV input sequence $u_i^{\mathrm{EV}} = (u_{i,k_i^{\mathrm{e}}}^{\mathrm{EV}}, \ldots, u_{i,k_i^{\mathrm{d}}-1}^{\mathrm{EV}})$ that gives x_i through the integration of (28). Noting that u_i^{EV} thereby is a function of x_i, the energy consumption associated with the state trajectory x_i is taken to be

$$E_i(x_i) = \sum_{k=k_i^{\mathrm{e}}}^{k_i^{\mathrm{d}}-1} \int_{k\Delta t}^{(k+1)\Delta t} \frac{\omega_i(t) M_{i,k}}{\eta_i(\omega_i(t), M_{i,k})} \mathrm{d}t. \tag{29}$$

where $M_{i,k}$ is the electric motor torque applied between $k\Delta t$ and $(k+1)\Delta t$, $\eta_i(\cdot)$ is the electric motor's efficiency map and $\omega_i(t) = G_i / r_i^{\mathrm{w}} v_i(t)$ is the electric motor speed.

Since traveling at a constant velocity is associated with a certain energy consumption due to various resistive forces, of particular interest is the *cost of coordination (CoC)*, which we define as the increase in energy consumption compared to the Overpass controller. For vehicle i, we define the cost of coordination as

$$E_i^{\mathrm{CoC}}(x_i) = E_i(x_i) - E_i^{\mathrm{OP}}(x_i) \tag{30}$$

where $E_i^{\mathrm{OP}}(x_i)$ is the energy consumed by a vehicle that travels between $p_{i,k_i^{\mathrm{e}}}$ and $p_{i,k_i^{\mathrm{d}}}$ at the constant velocity v^{e}.

Note that even though the quadratic objective (26) does not explicitly describe the, arguable more relevant, secondary objectives, it consists of related quantities. In particular, the velocity deviation term $(v_{i,k} - v^{\mathrm{r}})^2$ penalizes low velocities and will indirectly force the travel time delay δt_i to be small. Similarly, the acceleration term $u_{i,k}^2$ is proportional to the forces applied to the vehicles which relates to the energy supplied to the propulsion system. Keeping the acceleration term small will consequently yield an energy consumption close to that of E_i^{OP}.

4.5. Safety-Enforcing Vehicle Controllers

To ensure that Assumption 3 is satisfied, the vehicles that are in the SZ but not yet in the IZ execute safety enforcing controllers. In particular, all such vehicles apply

$$u_{i,k} = \min(u_i^{\mathrm{LQR}}(x_{i,k}), u_i^{\mathrm{Safe}}(x_{i,k}, x_{j,k})) \tag{31}$$

where $u_i^{\mathrm{LQR}} = \max(a_i^{\min}, \min(a_i^{\max}, K_i^{\mathrm{LQR}}(v_{i,k} - v^{\mathrm{r}})))$ and where K_i^{LQR} is the LQR gain computed from A_i, B_i, Q_i, R_i. Moreover, $u_{i,k}^{\mathrm{Safe}}(x_{i,k}, x_{j,k})$ is such that

$$\begin{aligned}
u_i^{\mathrm{Safe}}(x_{i,k}, x_{j,k}) \leq \max_{w_i} \quad & u_{i,k} && && \text{(32a)}\\
\text{s.t.} \quad & x_{i,k} = \hat{x}_{i,k}, && && \text{(32b)}\\
& x_{i,k+l+1} = A_i x_{i,k+l} + B_i u_{i,k+l}, && l \in \mathcal{I}_{K-1}, && \text{(32c)}\\
& a_i^{\min} \leq u_{i,k+l} \leq a_i^{\max}, && k \in \mathcal{I}_{K-1}, && \text{(32d)}\\
& p_{i,k+l} + \delta_{ij} \leq \bar{p}_{j,k+l}(x_{j,k}), && k \in \mathcal{I}_{k_j^{\mathrm{s}}}, && \text{(32e)}
\end{aligned}$$

where $\{p_{j,k+l}(x_{j,k})\}_{l=0}^{k_j^{\mathrm{s}}}$ is the position sequence resulting from a maximum brake maneuver from vehicle j, starting from state $x_{i,j}$ and reaching 0 velocity at k_j^{s}.

While the use of (31) ensures that all optimization problems solved in the IZ are feasible, any velocity reduction performed by the vehicles inside the IZ can propagate backwards, leading to lower velocities between the scenario starting point and the start of the IZ.

4.6. Termination due to Congestion

The simulation is terminated when the simulation end time is reached or the scenario is considered congested. The latter is the case when no safe control command $u_{i,k}^{\mathrm{Safe}}(x_{i,k}, x_{j,k})$ exists for vehicle i when it is added according to the scenario's generation pattern. This only occurs when the velocity in the IZ has dropped due to the action of the coordination controller, and significant velocity reductions have propagated to the start of the SZ.

5. Results

In this section, we present the results from the performance evaluation of the different controllers. We have considered the simulation of 15 minutes of traffic for rate parameters λ corresponding to average arrival rates ranging from $R = 4000$ to $R = 10000$ vehicles/hour (1000 to 2500 vehicles/hour/lane). The parameters used in the simulation are summarized in Tables 1 and 2.

For all controllers, the interior-point solver *fmincon* is used in MATLAB®to solve the NLPs involved, and for the MIQP/FO controller, the CPLEX®MIQP solver is employed in the first stage of the approximation procedure. We emphasize that a fully centralized solution is not a necessity, and that one could employ the distributed methods of [27–30] to solve the fixed order problem (13). Animations showing the results can be found at [44]. Videos showing how a controller like that presented here works on real vehicles can also be found at [45], containing the material from the experimental validation presented in [37].

5.1. Performance metrics

The performance scores for all controllers are computed as the average over all vehicles that have both entered and left the scenario during the simulation time. In particular, for the quadratic objective (26) we investigate

$$\hat{J}^v = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} \sum_{k=k_i^{\mathrm{e}}}^{k_i^{\mathrm{d}}} m_i Q_i (v_{i,k} - v^{\mathrm{r}})^2 \tag{33}$$

$$\hat{J}^u = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} \sum_{k=k_i^{\mathrm{e}}}^{k_i^{\mathrm{d}}} m_i R_i u_{i,k}^2 \tag{34}$$

where $\mathcal{N}^{\mathrm{c}}$ contains the indices of all vehicles that cross the SZ within S, and we recall that k_i^{e},k_i^{d} denotes the time instants of entry and departure of the SZ for vehicle i.

Type	Symbol	Value	Unit
SZ Start	p^{e}_{SZ}	-350	m
SZ Stop	p^{d}_{SZ}	250	m
IZ Start	$p^{\mathrm{e}}_{\mathrm{IZ}}$	-200	m
IZ Stop	$p^{\mathrm{d}}_{\mathrm{IZ}}$	250	m
Car Gen. Prob.	p^{Car}	0.9	
Truck Gen. Prob.	p^{Truck}	0.1	
Initial/Set Speed	v^{e}	70	km/h
Discretization size	Δt	0.2	s
Prediction Horizon	K	100	
RECA Safety distance	ϵ	1.5	m
T.L. Cycle time	C	20	s
Air density	ρ	1.225	kg/m^3
Acc. due to gravity	g	9.81	m/s^2

Table 1.: General Parameters.

Type	Symbol	Value		Unit
		Car	Truck	
Mass	m_i	1.7	20	10^3 kg
Length	L_i	4.8	16.5	m
Width	W_i	1.77	2.55	m
Speed Dev. Weight	Q_i	1		
Control Use Weight	R_i	1		
Acceleration L.B.	$a_i^{\min}$	-3		m/s^2
Acceleration U.B.	$a_i^{\max}$	3		m/s^2
Gear ratio	G_i	7.94	15	
Wheel radius	r_i^{w}	0.35		m
Projected Front Area	a_i	2.3	4	m^2
Air drag coef.	c_i^{d}	0.32	0.7	
Rolling resistance coef.	c_i^{rr}	0.015		
Max Power	$P_i^{\max}$	80	400	kW
Max Torque	$M_i^{\max}$	250	2000	kNm
Max Motor Speed	$\omega_i^{\max}$	10	15	kRPM

Table 2.: Vehicle Parameters.

That is, $\hat{J}^v$ and $\hat{J}^u$ is the closed-loop evaluation of the total velocity deviation and acceleration components of (26), averaged over all vehicles that crossed the SZ.

Similarly, for the secondary objectives, we consider

$$\hat{E}^{\mathrm{CoC}} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} E_i^{\mathrm{CoC}}(x_i), \tag{35}$$

$$\hat{\delta t} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} \delta t_i, \tag{36}$$

where $E_i^{\mathrm{CoC}}(x_i)$ and δt_i are computed through (30) and (27), respectively. That is, (35) is the total cost of coordination and travel time delay induced, averaged over all vehicles that crossed the SZ. For comparison, we also consider the percentage change in energy consumption, with respect to the Overpass solution, defined as

$$\hat{E}^{\%} = \frac{\sum_{i \in \mathcal{N}^{\mathrm{c}}} E_i(x_i)}{\sum_{i \in \mathcal{N}^{\mathrm{c}}} E_i^{\mathrm{OP}}(x_i)} \times 100 \tag{37}$$

The efficiency map $\eta(\cdot)$ used to determine the energy consumption is obtained from [46], and consists of a polynomial fit to experimental data. The map is scaled for the *Car* and *Truck* types using the parameters $M_i^{\max}$, $\omega_i^{\max}$ and $P_i^{\max}$ reported in Table 2.

5.2. Performance results

Figures 6, 7 and 8 summarize the main results of the performance evaluation. The first thing to note is the lack of data-points for the Traffic Light, and FCFS/FO controller for $R > 9000$ vehicle/h, and the lack of data-points for the Sequential scheme for $R > 6500$ vehicles/h. This is a consequence of premature simulation termination and is discussed further in Section 5.3.

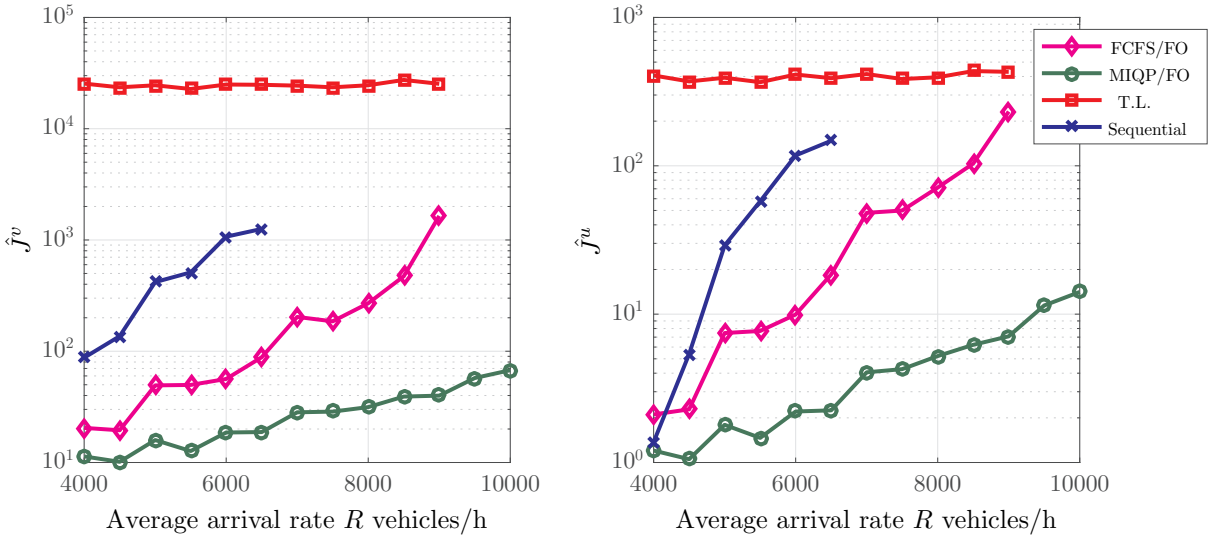


Figure 6.: Development of the components of the quadratic objective for different arrival rates. Here T.L is the traffic light controller.

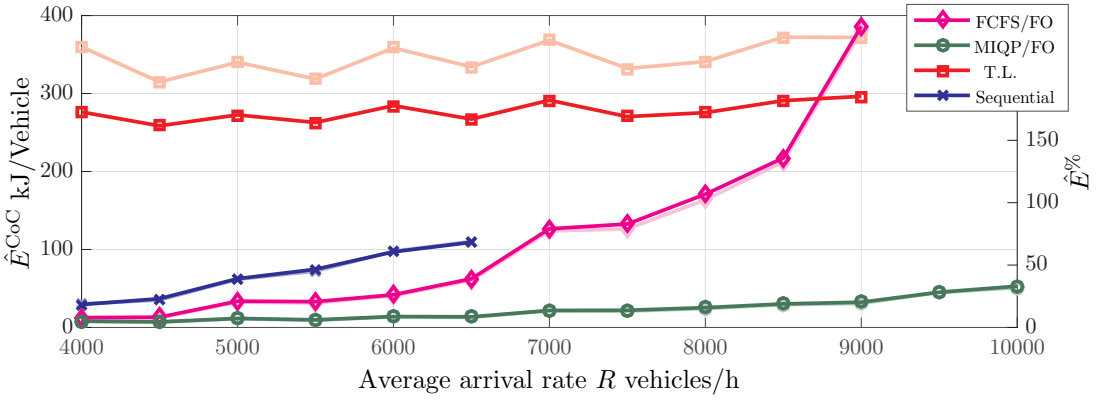


Figure 7.: The cost of coordination in terms of energy for different arrival rates, color coding as in Figure 6. The full colored lines show the percentage increase with respect to the overpass $\hat{E}^{\%}$ (right axis), whereas the pale lines show $\hat{E}^{\text{CoC}}$ (left axis).

Traffic Light vs. Automated: We note that the difference between the automated controllers and the Traffic Light solution is rather large for average arrival rates low enough to not cause congestion. For low arrival rates, all automated controllers give small increases in energy consumption compared to the overpass solution, induce a small travel time delay and are orders of magnitude better than the Traffic Light in terms of the quadratic objective. This is mainly a consequence of the different controllers' ability to coordinate the vehicles through the intersection without forcing them to stop, and thereby avoiding heavy accelerations and extended periods of low velocities observed at a traffic light. We highlight in particular the performance of the proposed MIQP/FO controller in terms of energy consumption, and note that it can handle very high traffic intensities ($R = 10000$) without "paying" more than 40% energy than simply driving at the initial velocity v^{e}.

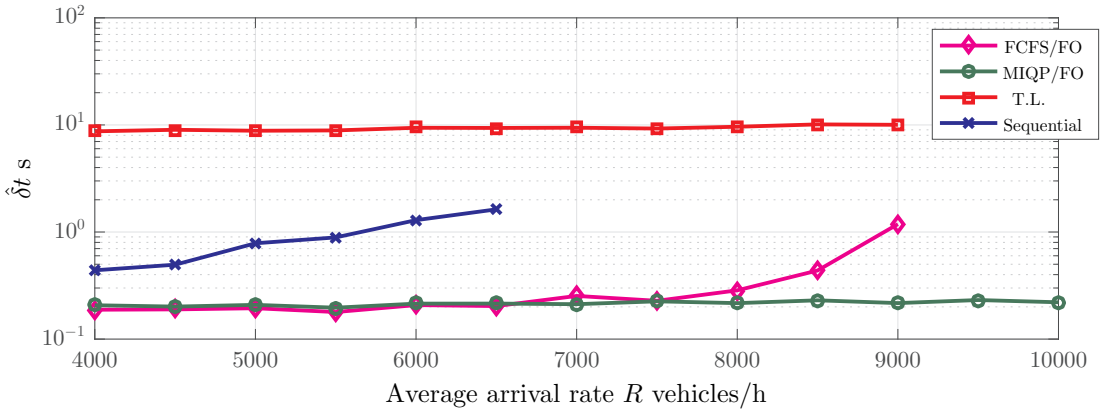


Figure 8.: Travel time delay compared to the Overpass solution.

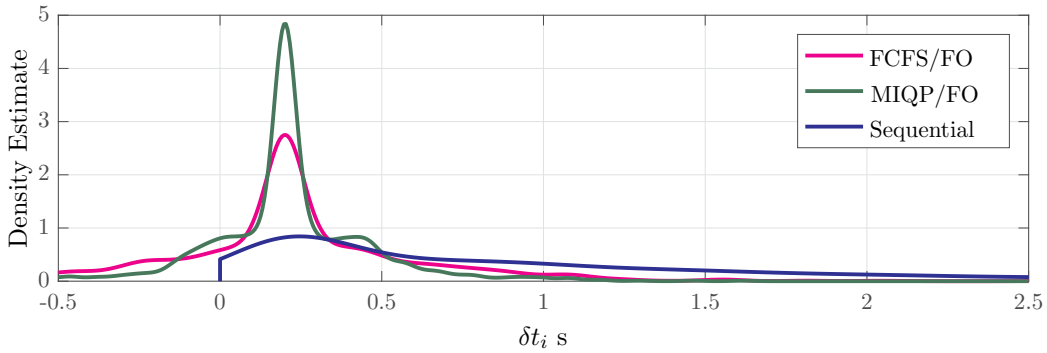


Figure 9.: Estimate of the distribution of the observed δt_i for $R = 6000$ vehicles/h.

Effect of joint optimization: We also note that the performance of the different automated controllers increases with added complexity: it is significantly better in all performance metrics to jointly optimize the trajectories under a First-Come-First Served policy (the FCFS/FO) than to decide both the order and the trajectories sequentially, and both are in turn worse than the joint optimization of both the crossing order and the trajectories (the MIQP/FO). The latter is true for all performance metrics except the travel-time delay, where the FCFS/FO and MIQP/FO perform virtually the same for $R \leq 8000$ vehicles/h, with close to zero average delay. This is a consequence of the joint optimization of the trajectories through NLP (14), which leads to increased velocity of some vehicles (resulting in "negative" delays), and decreased velocities in others (resulting in positive delays), with an average close to zero. This is further illustrated in Figure 9, which shows estimates of the distribution of travel time delays under the three automated controllers.

Finally, we remark that the results are objective-function dependent and different results would be obtained with different choices (e.g., pure optimization of δt_i).

A closer look at the vehicle velocities: An intuitive understanding of the performance differences can be obtained by examining Figure 10, which shows the development of the average, maximum and minimum velocity in the scenario. As the figure shows, the variations in velocity decrease with increasing controller complexity, where we note in particular the (expected) presence of stationary vehicles in the Traffic Light

case. Since all velocity changes ultimately lead to increased energy consumption, this gives the results shown in Figure 7. In a similar fashion, the almost constant average velocity of the FCFS/FO and MIQP/FO controllers, and varying average velocity of the Traffic Light and Sequential controllers, causes the results shown in Figure 8.

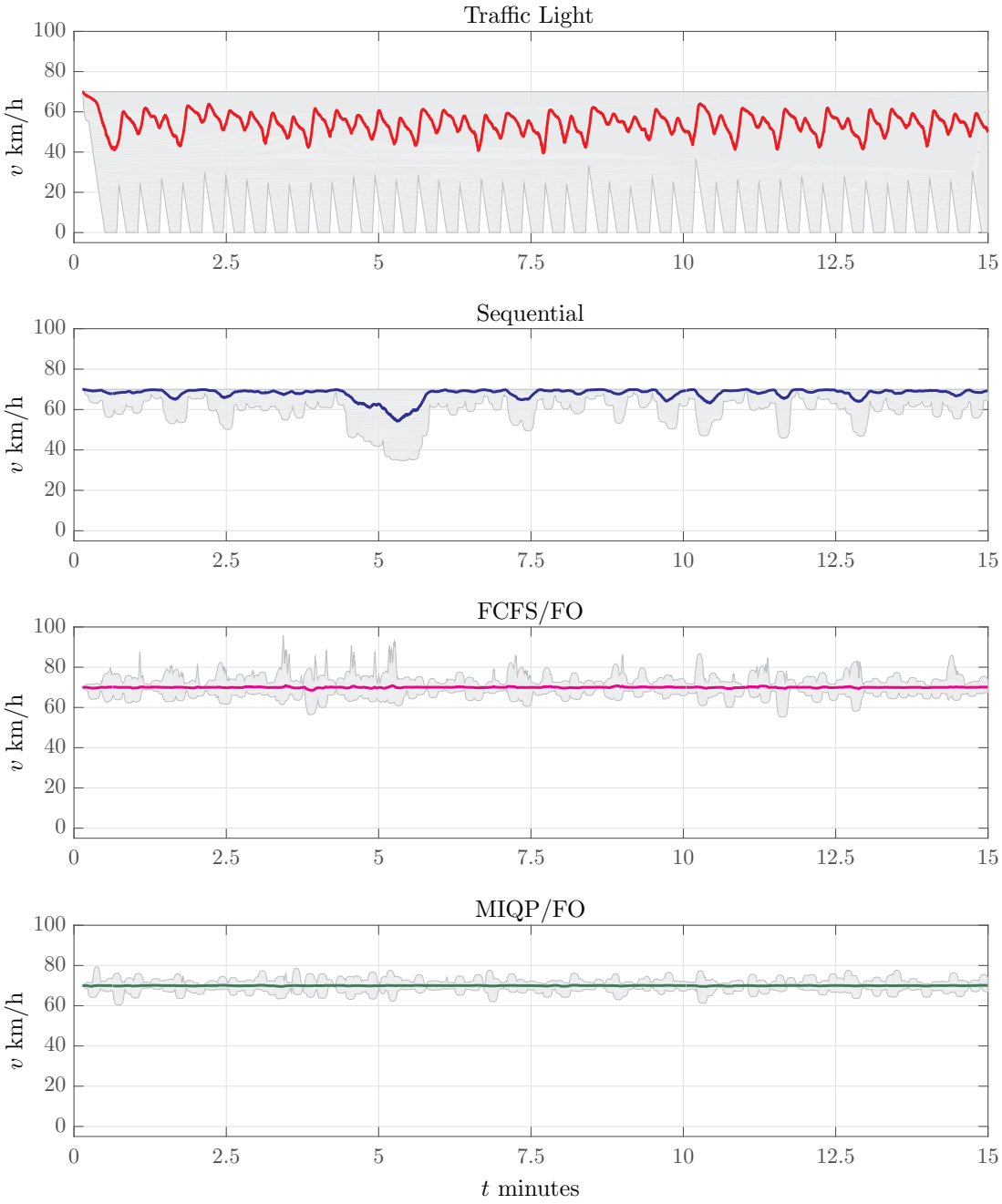


Figure 10.: Average velocity (colored lines) and velocity intervals (gray surface) in a scenario with $R = 6000$ vehicles/hour.

Note the qualitative difference between the maximum and minimum velocity intervals in the MIQP/FO and FCFS/FO cases. While the former has a smooth appearance, the latter sees spikes on a number of occasions (e.g., around the 5 minute mark). A closer look at one spike is provided in Figure 11, which shows the position-velocity

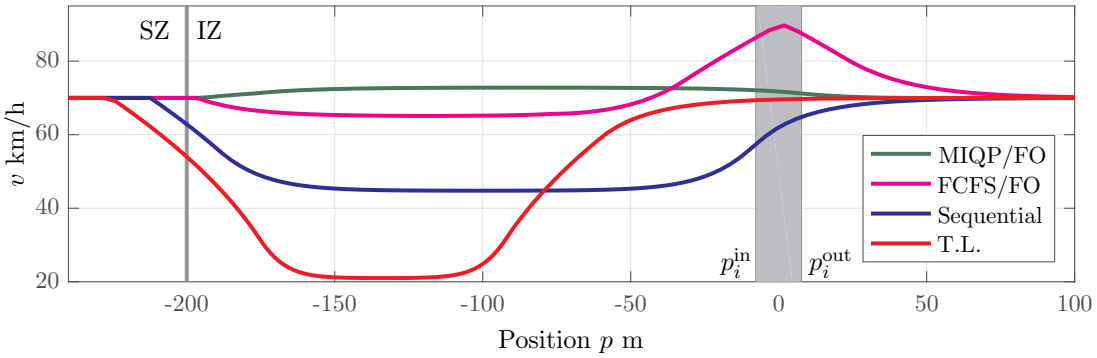


Figure 11.: Position-Velocity trajectories of one vehicle from the scenario with $R = 6000$ vehicles/hour. The gray bar corresponds to positions inside the intersection, 0 being the center, whereas the gray line demarcates the beginning of the IZ.

profiles from the same vehicle for the different controllers. As the figure illustrates, the spikes occur when the optimal solution to the fixed-order problem (13) under the FCFS-crossing order is to accelerate some vehicles heavily through the intersection. While this incurs a cost for that vehicle in terms of (26), it allows other vehicles to use the intersection in a manner that is beneficial for the scenario as a whole. Even though a slight velocity increase also results from the MIQP/FO controller, the magnitude is significantly lower and is performed well before the intersection starts. This is a testament of the MIQP's ability to select crossing orders that are good in terms of the objective (13a), which enables solutions to the fixed order problem that incur a lower total cost. Finally, we also note the effect of the feasibility-enforcing controller discussed in Section 4.5, which slows down the vehicles in the Traffic Light and Sequential cases before the IZ starts.

5.3. Failure of the FCFS/FO and Sequential controllers

In the simulation with $R = 7000$ vehicles per hour, the Sequential controller caused some vehicles inside the IZ to reduce their velocity significantly. In turn, this caused vehicles outside the IZ to slow down due to (31), in order for Assumption 3 to hold. Eventually, a significant velocity decrease propagated to the beginning of the SZ, such that the scenario was considered congested and the simulation stopped. For the FCFS/FO controller, the simulation was stopped after it performed worse than the Traffic Light (c.f. Figure 7).

5.3.1. Causes of the FCFS/FO failure

We first note that the fixed-order problem (13) is expected to assign a relatively higher control effort and result in larger velocity deviations for Cars than Trucks due the objective weighting (c.f. objective (26) and Table 2). Regardless of how the crossing order is selected, the vehicles of the Car type are therefore expected to perform more aggressive maneuvers in general, and be responsible for the maximum and minimum velocities (c.f. the velocity intervals of Figure 10).

However, the magnitude of both control effort and velocity deviations will be dependent on the selected crossing order. For instance, a Car which is ordered behind a Truck due to the FCFS policy could be commanded to slow down significantly to decrease the total cost, whereas an alternative order for which the Car crosses the

intersection before the Truck could give a small velocity increase for the same reason. When the traffic load increases, such accelerations occur farther from the intersection with lower minimum values, and the lower velocities are kept for longer periods of time.

This effect is illustrated in Figure 12, which presents the velocity as a function of distance for all vehicles resulting from the FCFS/FO controller for average arrival rates $R = 8000$, $R = 8500$ and $R = 9000$ vehicles/h, and from the MIQP/FO controller for $R = 9000$. As the figure shows, for increasing R, the FCFS/FO controller indeed results in harsher accelerations and both lower minimum velocities at greater distances from the intersection and longer periods of low velocities. This causes the performance to deteriorate. Note in particular the almost triangular velocity profiles of many Cars as the intersection is crossed, which corresponds to periods of (constant) maximum acceleration and deceleration. This behavior is primarily seen in Cars as the optimal solution many times is to first slow down in order to favor Trucks (due to the weighting of their objectives with m_i), and thereafter to cross the intersection as fast as possible to not block access for others. Moreover, we note that the velocity decreases closer to the SZ start with increasing R, due to the safety enforcing controller (31), and that the FCFS/FO is brought closer to causing congestion.

Finally, we note that while both the FCFS/FO and MIQP/FO controllers actuate Cars more than Trucks, the effect is much more pronounced in the former. In particular, the two bottom plots in Figure 12 illustrates the difference for the same generation pattern, and show that both vehicle types are actuated less under the MIQP/FO controller. This results in smoother trajectories and almost no velocity reduction outside the IZ. This demonstrates the benefit of selecting a crossing order which takes the objective function and constraints of MINLP (11) into account.

5.3.2. Causes of the Sequential failure

In the Sequential controller case, no vehicle increases its velocity to favor another, and collisions are avoided solely through velocity reductions. This propagates backwards on each lane, and can even be amplified depending on the distance between the vehicles involved. The velocity profiles from the congested $R = 7000$ vehicles/h scenario with the Sequential controller are shown in Figure 13. As can be seen, the scenario contains significant velocity reductions in the entire IZ, which eventually propagates further backwards and reaches p^{e}_{SZ}, after which the simulation is stopped.

6. Conclusion

In this paper we introduced a closed-loop controller for coordination of automated vehicles at intersections, based on a simultaneous yet approximate solution of an optimal control problem. We presented simulation results where the controller was compared against two simpler approaches for automated coordination, a traffic light coordination mechanism and a physical separation of the roads. The results demonstrate that all automated controllers out-perform the traffic light system under low to medium traffic intensities and that significant gains are achieved with increasing controller complexity. In particular, we showed that by jointly optimizing both the crossing order and the vehicle trajectories, large improvements are obtained compared to all other considered methods, both in terms of performance and capacity. This serves as a motivation for considering the more sophisticated controllers.

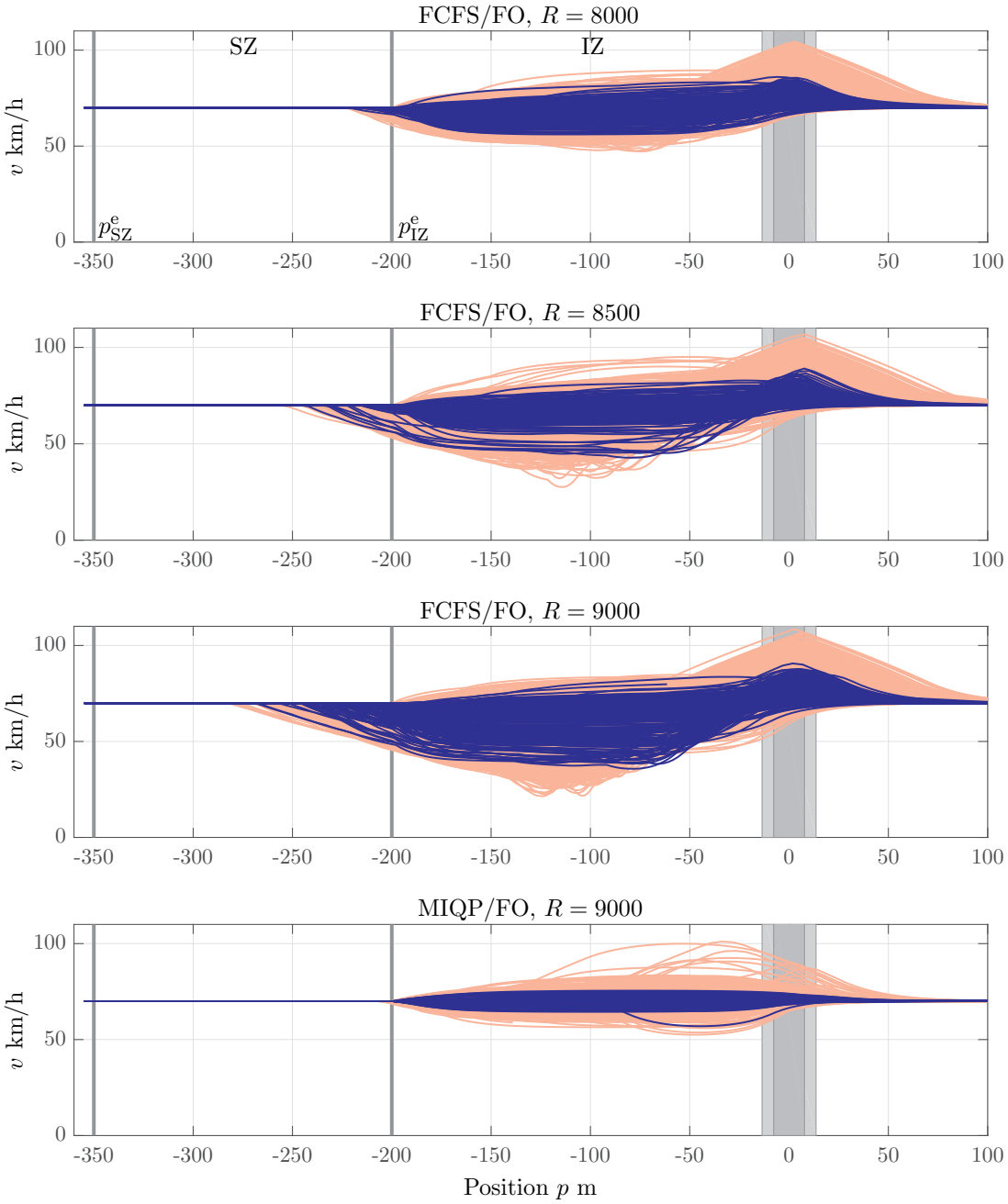


Figure 12.: Speed profiles from vehicles in different scenarios, with trajectories from Trucks in blue and from Cars in pale red. The gray bar illustrate the extent of the intersection for Cars (dark) and Trucks (light) and the two vertical lines indicate the start of the SZ and IZ respectively. We emphasize that the generation pattern in the two bottom plots are the same.

We also emphasize that even though the proposed MIQP/FO controller relies on approximations and likely is sub-optimal, the price payed for coordination is remarkably small, even for high traffic intensities. At the same time, the improvement over both traffic lights and the simpler coordination mechanisms is still large, in particular for higher traffic intensities.

Considering the seemingly limited possibilities for further improvements, we there-

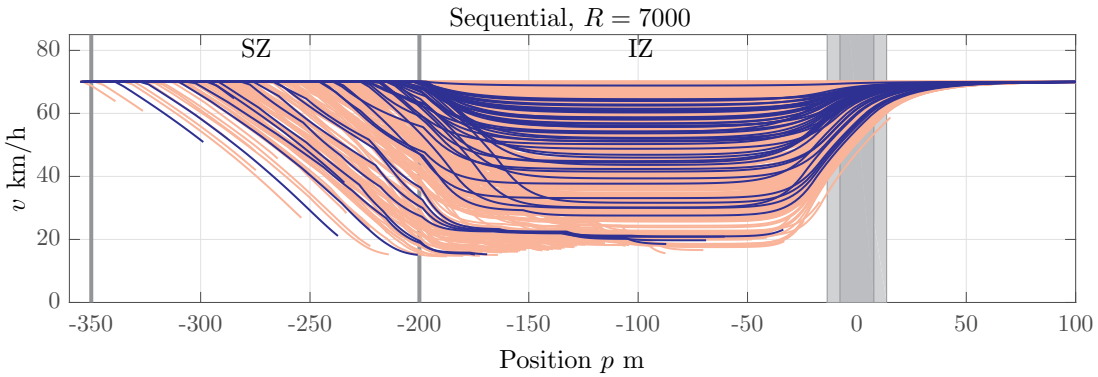


Figure 13.: Velocity profiles for vehicles in the scenario with $R = 7000$, where the sequential controller failed. Coloring as in Figure 12.

fore conjecture that the performance of the MIQP/FO controller is close to that of the true optimal solution in scales relevant for the application, both in terms of the quadratic objective (26) and the secondary performance metrics.

6.1. Comments on the motion model and objective function choice

We note that while we employ a simple linear prediction model and quadratic objective function, the proposed method as such is motion model and objective function agnostic. Higher fidelity models and general, economic objectives could be considered (see e.g. [36,45]), at the expense of solving more difficult NLPs. However, given that the potential for further improvement seems to be small, it is highly questionable if the gains of, e.g., directly optimizing energy consumption, would motivate the higher complexity optimization problems. We remark that this finding is interesting in its own right, since even though direct optimization of energy consumption using non-linear prediction models has shown to be very advantageous for smaller problems with a fixed number of vehicles [36], it does not seem to hold in larger scenarios with continuously oncoming cars.

It is also worth mentioning that in a practical setting, the coordination controller would likely belong to a tactical planning layer in the control architecture, without direct access to the actuators of the vehicles. The actual control commands would instead be generated by a lower level controller when tracking the commands produced by the coordination controller. As a consequence, the prediction model should instead describe the closed-loop response of the vehicle and the lower level control system, thus motivating the use of simpler models [47,48]. This is further strengthened by the results in [37], where a closed-loop controller based on (14), (28) and (26) was validated using real vehicles with remarkably good results.

6.2. Future research directions

The method presented in this paper, as well as most of the literature on the topic, considers scenarios where all vehicles are automated and cooperative (Assumption 1). Since such high technology penetration lies in a distant future, this assumption severely restricts the usefulness of the method. A highly relevant research direction is therefore the extension to scenarios with partial penetration, and the ability to include non-

cooperative traffic users, possibly using models such as those discussed in [49]. The extension to systems of connected intersections should also be considered, and the large-scale performance and safety of the proposed approach established.

While we solve the problem in a centralized setting in this paper, we reiterate that it can be done in a distributed fashion [27–30]. In a practical setting, the distributed optimization algorithms would be executed over the wireless network, and their performance is therefore dependent on the quality of the communication channel. Modifications to the these algorithms which ensure adequate performance under adverse communication conditions must therefore be found. As discussed in [50], a possible approach is to introduce mechanisms for communication scheduling and resource allocation, where those vehicles in most need of communication (e.g., are most likely to get into dangerous configurations) are prioritized.

Finally, as noted in [37], a constraint-tightening approach can easily be taken to make the two-stage controller robust with respect to some types of perturbations. However, further investigation is necessary on how to best robustify our approach to general perturbations.

References

[1] Sjöberg K, Andres P, Buburuzan T, et al. Cooperative intelligent transport systems in europe: Current deployment status and outlook. IEEE Vehicular Technology Magazine. 2017 June;12(2):89–97.

[2] Chen S, Hu J, Shi Y, et al. Vehicle-to-Everything (V2X) Services Supported by LTE-Based Systems and 5G. IEEE Communications Standards Magazine. 2017;1(2):70–76.

[3] Campbell ME, Egerstedt M, How JP, et al. Autonomous driving in urban environments: approaches, lessons and challenges. Philosophical transactions Series A, Mathematical, physical, and engineering sciences. 2010;368 1928:4649–72.

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

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

[6] Hult R, Campos GR, Steinmetz E, et al. Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation. IEEE Signal Processing Magazine. 2016 Nov;33(6):74–84.

[7] Wymeersch H, de Campos GR, Falcone P, et al. Challenges for cooperative its: Improving road safety through the integration of wireless communications, control, and positioning. In: 2015 International Conference on Computing, Networking and Communications (ICNC); Feb; 2015. p. 573–578.

[8] Colombo A, Vecchio DD. Least restrictive supervisors for intersection collision avoidance: A scheduling approach. IEEE Transactions on Automatic Control. 2015 June;60(6):1515–1527.

[9] Chen L, Englund C. Cooperative intersection management : A survey. IEEE transactions on intelligent transportation systems. 2016;17(2):570–586.

[10] Rios-Torres J, Malikopoulos AA. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. IEEE Transactions on Intelligent Transportation Systems. 2017;18:1066–1077.

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

[12] Milanes V, Perez J, Onieva E, et al. Controller for urban intersections based on wire-

less communications and fuzzy logic. IEEE Transactions on Intelligent Transportation Systems. 2010 March;11(1):243–248.
[13] Kowshik H, Caveney D, Kumar PR. Provable systemwide safety in intelligent intersections. IEEE Transactions on Vehicular Technology. 2011 March;60(3):804–818.
[14] de Campos GR, Falcone P, Sjöberg J. Autonomous cooperative driving: A velocity-based negotiation approach for intersection crossing. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013); Oct; 2013. p. 1456–1461.
[15] de Campos GR, Falcone P, Wymeersch H, et al. Cooperative receding horizon conflict resolution at traffic intersections. In: 53rd IEEE Conference on Decision and Control; Dec; 2014. p. 2932–2937.
[16] Qian X, Grégoire J, de La Fortelle A, et al. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In: 2015 European Control Conference (ECC); July; 2015. p. 3452–3458.
[17] Kim K, Kumar PR. An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic. IEEE Transactions on Automatic Control. 2014 Dec; 59(12):3341–3356.
[18] Molinari F, Raisch J. Automation of road intersections using consensus-based auction algorithms. In: 2018 Annual American Control Conference (ACC); June; 2018. p. 5994–6001.
[19] Makarem L, Gillet D. Model predictive coordination of autonomous vehicles crossing intersections. In: 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013); Oct; 2013. p. 1799–1804.
[20] Katriniok A, Kleibaum P, Josevski M. Distributed model predictive control for intersection automation using a parallelized optimization approach. IFAC-PapersOnLine. 2017; 50(1):5940 – 5946. 20th IFAC World Congress.
[21] Sprodowski T, Pannek J. Stability of distributed MPC in an intersection scenario. Journal of Physics: Conference Series. 2015 nov;659:012049.
[22] Britzelmeier A, Gerdts M. Non-linear model predictive control of connected, automatic cars in a road network using optimal control methods. IFAC-PapersOnLine. 2018; 51(2):168 – 173. 9th Vienna International Conference on Mathematical Modelling.
[23] Malikopoulos AA, Cassandras CG, Zhang YJ. A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections. Automatica. 2018;93:244 – 256.
[24] Murgovski N, de Campos GR, Sjöberg J. Convex modeling of conflict resolution at traffic intersections. In: 2015 54th IEEE Conference on Decision and Control (CDC); Dec; 2015. p. 4708–4713.
[25] Riegger L, Carlander M, Lidander N, et al. Centralized mpc for autonomous intersection crossing. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC); Nov; 2016. p. 1372–1377.
[26] Tallapragada P, Cortés J. Coordinated intersection traffic management. IFAC-PapersOnLine. 2015;48(22):233 – 239. 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015.
[27] Hult R, Zanon M, Gros S, et al. Primal decomposition of the optimal coordination of vehicles at traffic intersections. In: 2016 IEEE 55th Conference on Decision and Control (CDC); Dec; 2016. p. 2567–2573.
[28] Zanon M, Gros S, Wymeersch H, et al. An asynchronous algorithm for optimal vehicle coordination at traffic intersections. 2017;20th IFAC World Congress.
[29] Jiang Y, Zanon M, Hult R, et al. Distributed algorithm for optimal vehicle coordination at traffic intersections. IFAC-PapersOnLine. 2017;50(1):11577 – 11582. 20th IFAC World Congress.
[30] Shi J, Zheng Y, Jiang Y, et al. Distributed control algorithm for vehicle coordination at traffic intersections. In: 2018 European Control Conference (ECC); June; 2018. p. 1166–1171.
[31] Kneissl M, Molin A, Esen H, et al. A feasible mpc-based negotiation algorithm for au-

tomated intersection crossing *. In: European Control Conference (ECC); 06; 2018. p. 1282–1288.
[32] Kamal MAS, Imura J, Hayakawa T, et al. A vehicle-intersection coordination scheme for smooth flows of traffic without using traffic lights. IEEE Transactions on Intelligent Transportation Systems. 2015 June;16(3):1136–1147.
[33] Li B, Zhang Y, Zhang Y, et al. Near-optimal online motion planning of connected and automated vehicles at a signal-free and lane-free intersection. In: IEEE Intelligent Vehicle Symposium; 06; 2018. p. 1432–1437.
[34] Bali C, Richards A. Merging vehicles at junctions using mixed-integer model predictive control. In: European Control Conference (ECC); 06; 2018. p. 1740–1745.
[35] Hult R, Campos GR, Falcone P, et al. An approximate solution to the optimal coordination problem for autonomous vehicles at intersections. In: 2015 American Control Conference (ACC); July; 2015. p. 763–768.
[36] Hult R, Zanon M, Gras S, et al. An miqp-based heuristic for optimal coordination of vehicles at intersections. In: 2018 IEEE Conference on Decision and Control (CDC); Dec; 2018. p. 2783–2790.
[37] Hult R, Zanon M, Gros S, et al. Optimal coordination of automated vehicles at intersections: Theory and experiments. IEEE Transactions on Control Systems Technology. 2018; :1–16.
[38] Dresner K, Stone P. 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.; July; 2004. p. 530–537.
[39] Lee J, Park B. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Transactions on Intelligent Transportation Systems. 2012 March;13(1):81–90.
[40] Rawlings J, Mayne D. Model Predictive Control: Theory and Design. Nob Hill; 2009.
[41] Hult R, Zanon M, Gros S, et al. Optimal coordination of automated vehicles at intersections with turns. In: To be presented at the European Control Conference (ECC); 2019.
[42] Nocedal J, Wright SJ. Numerical optimization. 2nd ed. New York, NY, USA: Springer; 2006.
[43] Büskens C, Maurer H. Online optimization of large scale systems. Berlin, Heidelberg: Springer Berlin Heidelberg; 2001. Chapter Sensitivity Analysis and Real-Time Optimization of Parametric Nonlinear Programming Problems; p. 3–16.
[44] Hult R, Zanon M, Gros WH S, et al. Animated illustration of the MIQP/FO controller [Online `https://youtu.be/hUWQoaiqdAY`]; ???? Accessed: 2019-02-26.
[45] Hult R, Zanon M, Gros S, et al. Optimal coordination of three cars approaching an intersection [Online: `https://youtu.be/nYSXvnaNRK4`]; ???? Accessed: 2019-02-01.
[46] Murgovski N, Johannesson LM, Egardt B. Optimal Battery Dimensioning and Control of a CVT PHEV Powertrain. IEEE Transactions on Vehicular Technology. 2014 Jun; 63(5):2151–2161.
[47] Kianfar R, Augusto B, Ebadighajari A, et al. Design and experimental validation of a cooperative driving system in the grand cooperative driving challenge. IEEE Transactions on Intelligent Transportation Systems. 2012 Sep;13(3):994–1007.
[48] Hult R, Sancar FE, Jalalmaab M, et al. Design and experimental validation of a cooperative driving control architecture for the grand cooperative driving challenge 2016. IEEE Transactions on Intelligent Transportation Systems. 2018 April;19(4):1290–1301.
[49] Batkovic I, Zanon M, Lubbe N, et al. A computationally efficient model for pedestrian motion prediction. In: 2018 European Control Conference (ECC); June; 2018. p. 374–379.
[50] Steinmetz E, Hult R, Zou Z, et al. Collision-aware communication for intersection management of automated vehicles. IEEE Access. 2018;6:77359–77371.