Taylor & Francis
Taylor & Francis Group

Vehicle System Dynamics

International Journal of Vehicle Mechanics and Mobility

ISSN: 0042-3114 (Print) 1744-5159 (Online) Journal homepage: https://www.tandfonline.com/loi/nvsd20

Optimisation-based coordination of connected, automated vehicles at intersections

Robert Hult, Mario Zanon, Sébastien Gros, Henk Wymeersch & Paolo Falcone

To cite this article: Robert Hult, Mario Zanon, Sébastien Gros, Henk Wymeersch & Paolo Falcone (2020) Optimisation-based coordination of connected, automated vehicles at intersections, Vehicle System Dynamics, 58:5, 726-747, DOI: 10.1080/00423114.2020.1755446

To link to this article: https://doi.org/10.1080/00423114.2020.1755446

Published online: 27 Apr 2020.

VEHICLE SYSTEM DYNAMICS
2020, VOL. 58, NO. 5, 726–747
https://doi.org/10.1080/00423114.2020.1755446

Optimisation-based coordination of connected, automated vehicles at intersections

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

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

ABSTRACT

In this paper, we analyse the performance of a model predictive controller for coordination of connected, automated vehicles at intersections. The problem has combinatorial complexity, and we propose to solve it approximately by using a two stage procedure where (1) the vehicle crossing 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 persistently safe and compare its performance against traffic lights and two simpler optimisation-based coordination schemes. The results show that our approach outperforms the considered alternatives in terms of both energy consumption and travel-time delay, especially for medium to high traffic loads.

ARTICLE HISTORY
Received 29 March 2019
Revised 7 April 2020
Accepted 8 April 2020

KEYWORDS
Connected and autonomous vehicles; model predictive control; intelligent transportation systems

1. Introduction

By combining Autonomous Driving (AD) technologies with communication [1,2], cooperative strategies can be implemented, which augment the capabilities of automated vehicles. In this paper, we consider such a strategy for coordinating connected and automated vehicles (CAV) at intersections, where the traffic system is burdened with high accident rates [3], congestion and energy-waste [4]. Currently, vehicles are coordinated at intersections with combinations of traffic lights, road signs and right-of-way rules, but with the introduction of CAVs, the coordination could instead be automated.

Unfortunately, the design of algorithms and controllers for this application is challenging for several reasons. First, the general problem of finding collision-free trajectories for vehicles crossing intersections has been shown to be NP-hard [5]. Additionally, imperfect prediction models, non-ideal sensors and impairments of the wireless communication channel [6] lead to uncertainties that necessitate closed-loop control. That is, the coordination must be updated repeatedly by accounting for the most up-to-date information from

CONTACT Robert Hult robert.hult@chalmers.se
*Present address: Engineering Department "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy

the vehicles. Establishing that such closed-loop systems are stable and persistently safe is in general a difficult task.

1.1. Solving the intersection coordination problem

The problem of coordinating CAVs at intersections has been surveyed in [7,8]. Most existing contributions are focused on scenarios where all vehicles are automated, and disregard 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 [9–11]. However, the problem is naturally formulated as a constrained optimal control problem (OCP), as it involves the optimisation of trajectories generated by dynamical systems, subject to (at least) collision avoidance constraints. A number of contributions have, therefore, been proposed by the control community [12–35]. However, solving the coordination OCP includes finding the order in which the vehicles cross the intersection and the space of possible solutions grows rapidly with the number of vehicles involved. Most existing approaches, therefore, avoid treating the coordination OCP directly, and are often based on a mix of heuristics and simpler OCPs. These schemes can roughly be categorised as *Sequential/Parallel* or *Simultaneous*, depending on what type of OCPs they involve.

In *Sequential/Parallel* schemes, a priority ranking of the vehicles is typically decided first. 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 [12,13] or the so-called MPC^* [14], 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 instead, the vehicle OCPs use predictions of higher priority vehicles' trajectories. Along these lines, Kim and Kumar [15] propose to use conservative estimates, based on predicted trajectories resulting from maximum braking manoeuvres. With the so-called MPC_0 solution, Qian et al. [14] suggest constant velocity predictions, whereas constant acceleration predictions are considered in [16]. 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. [17–20] and the so-called MPC_1 [14]). 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 [21]. There, a crossing time schedule is first constructed sequentially based on an FCFS policy, followed by the parallel solution of OCPs to compute the state and control trajectories.

Due to the reliance on priority relations, *Sequential/Parallel* schemes are often 'greedy', in the sense that a high priority vehicle does not take decisions that are beneficial to lower priority vehicles, if that decision is detrimental to the high priority vehicle itself. As a consequence, the effort required to resolve difficult conflicts is typically pushed 'downwards' in the priority hierarchy. Although sub-optimal by design, these schemes are often suitable for decentralised implementations with low and accurately predictable requirements on computation and information exchange.

In *Simultaneous* methods, on the other hand, the solution is found through joint optimisation of several vehicles' trajectories. However, to manage the combinatorial complexity, the crossing order 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 optimising the trajectories of the vehicles for the given crossing order. Such *fixed-order* joint optimisation was considered in [24–31]. Alternative approaches (e.g. [32,33]) apply local continuous optimisation methods to the full coordination OCP directly. The crossing order is thus selected by the optimiser, but depends on the initial-guess provided to the solver. A few contributions propose to solve the full coordination OCP directly and simultaneously optimise 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. Although such approaches are able to find optimal solutions, they typically scale poorly with the problem size (i.e. number of vehicles) and are therefore practically limited to small problem instances.

While *Simultaneous* approaches, in general, optimise 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 centralised fashion [24–26], or with iterative, distributed optimisation algorithms [27–31] which rely 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.

1.2. Contribution

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 optimisation 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 for the constraints and objective of the full problem. We extend the work of Hult et al. [27,36–39], by introducing the possibility to add and remove vehicles from the intersection scenario, thus simulating the arrival of new vehicles and the departures of vehicles which have already crossed the intersection. Since our approach is applicable to a vast class of vehicle models, we first present it in a general form, and afterwards we compare it to other approaches on a specific example in simulations.

We evaluate the closed-loop performance of the receding horizon application of the controller on a simulated scenario and compare it with the performance of (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 optimised. 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 optimised jointly, where only the trajectories are optimised jointly or where both the trajectories and the crossing order are optimised jointly.

The remainder of the paper is organised as follows: in Section 2, we introduce the proposed coordination algorithm. In Section 3, we introduce the scenario on which we evaluate the performance and detail the benchmarks considered. We discuss the results in Section 4 and conclude the paper in Section 5.

2. Optimal coordination at intersections

In this section, we model the intersection scenario and introduce the two-stage approximation procedure. Both the scenario modelling and the problem formulation are based on the following fundamental assumption.

Assumption 2.1: No non-cooperative entities are present in the scenario.

That is, we do not consider scenarios with, for example, legacy vehicles, pedestrians or bicyclists. Assumption 2.1, though standard in the literature on vehicle coordination problems, see, e.g. [11,12,21,40,41], is restrictive and may limit the applicability of coordination algorithms. We remark that our approach can be extended to accommodate the presence of non-cooperative road users by introducing appropriate models describing their behaviour in traffic. This is the subject of ongoing research.

Under these assumptions, the coordination problem is informally stated as ‘Find the control commands for all CAVs in the vicinity of the intersection which result in collision-free state trajectories that are consistent with the vehicle dynamics, satisfy all physical and design constraints and maximise performance.’

For ease of presentation and interpretation of the results, we consider only scenarios like the one shown in Figure 1(a), with two intersecting single-lane roads, where no turns

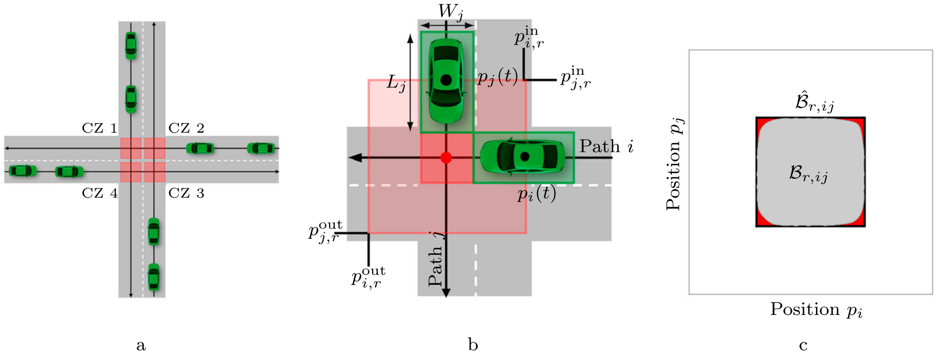

Figure 1. Illustrations of collision avoidance conditions. (a) Illustration of Assumption 2.2 and the Conflict Zones (red boxes); (b) scheme of the SICA condition (3): $\hat{\mathcal{G}}_i(p_i(t))$ shown as boxes around the vehicles and (c) conservativeness induced by rectangular outer approximations.

are allowed. We remark that the formalism presented in this paper can be applied to more general settings including multiple lanes [37] and turning vehicles [39].

2.1. Scenario modeling

While one can model the motion of the CAVs in the intersection with arbitrary accuracy, the following assumption, widely accepted in the literature [12,14,17,18,21,32], is convenient for our problem formulation.

Assumption 2.2: The CAVs move along fixed and known paths and do not reverse.

Assumption 2.2 is not restrictive, as it is reasonable to assume that the CAVs only move forward in existing lanes, and when the CAVs follow the lane centreline, the assumption is satisfied. We consider models of the form

$$\dot{x}_i(t) = f_i(x_i(t), u_i(t)), \tag{1a}$$

$$0 \geq g_i(x_i(t), u_i(t)), \tag{1b}$$

where $i \in \mathcal{N}(t) \subset \mathbb{N}$ is the CAV index, $N(t) = |\mathcal{N}(t)|$ is the number of CAVs that are coordinated at time t, $x_i(t) \in \mathbb{R}^{n_i}$ and $u_i(t) \in \mathbb{R}^{m_i}$ are the vehicle state and control, and both $f_i : \mathbb{R}^{n_i \times m_i} \mapsto \mathbb{R}^{n_i}$ and $g_i : \mathbb{R}^{n_i \times m_i} \mapsto \mathbb{R}^{q_i}$ are continuously differentiable. Additionally, the states are defined such that $x_i(t) = (p_i(t), v_i(t), \tilde{x}_i(t))$, where $p_i(t) \in \mathbb{R}$ is the position of the CAV's geometrical centre on its path, $v_i(t) \in \mathbb{R}_+$ is the velocity along the path and $\tilde{x}_i(t) \in \mathbb{R}^{n_i-2}$ collects all remaining states, if any.

2.1.1. Side collision avoidance conditions

As illustrated in Figure 1(a), side-collisions can occur between vehicles on different lanes when they are inside an area around the intersection of the vehicle paths, which we denote *Conflict Zone* (CZ). Collisions between vehicles i and j in CZ r are avoided at all time t if

$$(x_i(t), x_j(t)) \notin \mathcal{B}_{r,ij} = \left\{(x_i, x_j) \mid \mathcal{G}_i(p_i) \cap \mathcal{G}_j(p_j) \neq \emptyset\right\}, \quad \forall t, \tag{2}$$

where $\mathcal{G}_i(p_i)$ is the area occupied by vehicle i in the horizontal plane when the path coordinate is p_i. As illustrated in Figure 1(b), a slightly conservative but much simpler condition can be obtained using rectangular outer approximations $\hat{\mathcal{G}}_i(p_i) \supseteq \mathcal{G}_i(p_i)$, such that (2) is formulated as

$$(x_i(t), x_j(t)) \notin \hat{\mathcal{B}}_{r,ij} = \left\{(x_i, x_j) \mid p_i \in [p^{\text{in}}_{r,i}, p^{\text{out}}_{r,i}],\ p_j \in [p^{\text{in}}_{r,j}, p^{\text{out}}_{r,j}]\right\}, \tag{3}$$

where $p^{\text{in}}_{r,i}$ and $p^{\text{out}}_{r,i}$ are the first and last position on the path of vehicle i for $\hat{\mathcal{G}}_i(p_i) \cap \hat{\mathcal{G}}_j(p_j) \neq \emptyset$ for all p_j at CZ r. The conservative distance thus introduced is typically very small (see Figure 1(c)).

This approach is adopted in several works on intersection coordination (e.g. [12,18,21]), 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, defined implicitly through

$$p_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}, \quad \text{and} \quad p_i(t^{\text{out}}_{i,r}) = p^{\text{out}}_{r,i}, \quad \forall\, r \in \mathcal{R}_i, \tag{4}$$

where[1] $\mathcal{R}_i \subset \mathbb{N}$ collects the CZ crossed by CAV i. Since by Assumption 2.2 $v_i(t) \geq 0$, we have $p^{\text{in}}_{r,i} < p^{\text{out}}_{r,i} \Rightarrow t^{\text{in}}_{r,i} < t^{\text{out}}_{r,i}$ and side collision avoidance (SICA) (3) reads as

$$\left(t^{\text{out}}_{r,i} \leq t^{\text{in}}_{r,j}\right) \vee \left(t^{\text{out}}_{r,j} \leq t^{\text{in}}_{r,i}\right), \tag{5}$$

i.e. either vehicle j must leave CZ r before vehicle i enters or vice versa. Note that, in order to formulate SICA conditions, the actual value of the times is not relevant and only the difference between the times is. Therefore, any convention for defining $t = 0$ can be adopted, as long as all vehicles abide by the same convention.

2.1.2. Rear-end collision avoidance conditions

Under Assumption 2.2, rear-end collisions can only occur between two adjacent vehicles on the same path. By denoting the length of vehicle i as L_i and $\delta_{ij} = L_i/2 + L_j/2$, rear-end collision avoidance (RECA) is enforced by

$$p_i(t) + \delta_{ij} \leq p_j(t), \tag{6}$$

when vehicle i is behind vehicle j. Condition (6) can be extended to include conservative (and more practical) distance keeping policies, e.g. spacing policies with $\delta_{ij} = \epsilon_{ij} + L_i/2 + L_j/2$, for some $\epsilon_{ij} > 0$, either fixed or velocity dependent to include a time headway. We note that the convention used to fix $p_i(t) = 0$ is irrelevant, as long as all vehicles adopt the same convention, since only relative distances are required to formulate RECA conditions.

2.2. Optimal control formulation

The optimal control trajectories for all CAVs to be coordinated at time t_a are obtained as the solution of the following OCP:

$$\max_{x(t),u(t),T} \quad \sum_{i\in\mathcal{N}(t_a)} \int_{t_a}^{\infty} \ell_i(x_i(t), u_i(t))\mathrm{d}t, \tag{7a}$$

$$\text{s.t.} \quad (1),\ (4),\ x_i(t_a) = \hat{x}_i(t_a)\ \forall i \in \mathcal{N}(t_a), \tag{7b}$$

$$(6)\ \forall (i,j) \in \mathcal{C}_R(t_a),\ (5)\ \forall (i,j,r) \in \mathcal{C}_S(t_a). \tag{7c}$$

Here, $\hat{x}_i(t_a)$ is a state of CAV i at time t_a. Set $\mathcal{C}_R(t_a)$ includes all pairs (i,j) of subsequent CAVs on the same lane, i.e. all CAVs for which rear-end collisions might occur. Set $\mathcal{C}_S(t_a)$ includes all triplets (i,j,r) for which CAV pairs (i,j) have paths that cross in conflict zone r, i.e. set $\mathcal{C}_S(t_a)$ enumerates all vehicles and conflict zones for which side collisions are possible. Vector T collects all $t^{\text{in}}_{r,i}, t^{\text{out}}_{r,i}, \forall r \in \mathcal{R}_i, \forall i \in \mathcal{N}(t_a)$, and $x(t)$, $u(t)$ collect all $x_i(t)$ and $u_i(t)$, respectively. Finally, $\ell_i : \mathbb{R}^{n_i \times m_i} \mapsto \mathbb{R}$ is a continuously differentiable performance metric. We emphasise that OCP (7) optimises over all possible crossing orders through (5), and thereby, it is a combinatorial optimisation problem.

Problem (7) provides the open-loop optimal solution for the scenario at time t_a. If models (7b) perfectly describe the actual CAVs and $\hat{x}(t_a)$ is known exactly, this *is* the optimal solution until time $t_b > t_a$ when one additional vehicle approaches the intersection and must be accounted for. Moreover, since (1) does not perfectly describe the CAVs, and $\hat{x}(t_a)$ is typically known with some uncertainty, the actual state-trajectory obtained by applying the solution $u_i^*(t)$, $t \geq t_a$ will in general differ from $x_i^*(t)$. It could, therefore, be the case that constraints are actually violated by the vehicles, and collisions are caused as a result. For these reasons, closed-loop coordination is required in practice, where feedback is introduced and all $u_i^*(t)$ are recomputed periodically based on the most up-to-date information.

2.3. Closed-loop optimal coordination

A (closed-loop) model predictive controller (MPC) can be obtained by solving a discrete-time, finite-horizon approximation of (7) in a receding horizon fashion [42]. The finite-horizon problem is solved periodically, based on the current state of all vehicles in the scenario, whereafter the first part of the optimal control is applied. We employ a piecewise constant input parametrisation $u_i(t) = u_k$, $t \in [t_k, t_{k+1}]$, $t_k = k\Delta t$, $\Delta t \in \mathbb{R}_+$, $k \in \mathbb{N}$, horizon length $K \in \mathbb{N}$, and a multiple-shooting discretisation of the system dynamics [43]:

$$x_{i,k+r+1} - F_i(x_{i,k+r}, u_{i,k+r}) = 0, \quad r = 0, 1, \dots, K-1, \quad x_{i,k} = \hat{x}_i(t_k), \tag{8a}$$

$$g(x_{i,k+r}, u_{i,k+r}) \leq 0, \quad r = 0, 1, \dots, K, \tag{8b}$$

where $F_i : \mathbb{R}^{n_i \times m_i} \mapsto \mathbb{R}^{n_i}$ is the (numerical) integration of (1a), such that $F_i(x_i(t_k), u_{i,k}) = x_i(t_{k+1})$. Collecting $x_{i,k+r}$, $u_{i,k+r}$, $r = 0, \dots, K$ in $\mathbf{x}_i$, $\mathbf{u}_i$, respectively, the finite-horizon performance metric reads

$$J_i(\mathbf{x}_i, \mathbf{u}_i) = V_i^{\mathrm{f}}(x_{i,k+K}) + \sum_{r=0}^{K-1} \int_{t_{k+r}}^{t_{k+r+1}} \ell_i(x_i(t), u_{i,k+r})\, \mathrm{d}t, \tag{9}$$

where the terminal cost $V_i^{\mathrm{f}} : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ is continuously differentiable. Collecting $\mathbf{x}_i$, $\mathbf{u}_i$, $\forall i \in \mathcal{N}(t_k)$, in X, U, respectively, the approximation of (7) reads

$$\max_{X,U,T} \quad \sum_{i \in \mathcal{N}(t_k)} J_i(\mathbf{x}_i, \mathbf{u}_i) \tag{10a}$$

$$\text{s.t.} \quad (8),\ (4),\ \forall i \in \mathcal{N}(t_k), \tag{10b}$$

$$p_{i,k+r} + \delta_{ij} \leq p_{j,k+r},\ r = 0, \dots, K, \quad \forall (i,j) \in \mathcal{C}_R(t_k), \tag{10c}$$

$$(5)\ \forall (i,j,r) \in \mathcal{C}_S(t_k). \tag{10d}$$

Further details on the transcription of (7)–(10) are given in [38].

2.4. A two-stage approximation scheme

Problem (10) can be solved as a Mixed Integer NLP, but in a real-time MPC setting this is in general not tractable. However, an approximate solution can be obtained by the following two-stage procedure:

(1) Find a crossing order through a heuristic.

(2) Find the control commands and open-loop optimal state sequences by solving the *fixed-order problem*

$$\max_{X,U,T} \quad \sum_{i\in\mathcal{N}(t_k)} J_i(x_i,u_i) \tag{11a}$$

$$\text{s.t.} \quad (8),\ (4),\ \forall i \in \mathcal{N}(t_k), \tag{11b}$$

$$p_{i,k} + \delta_{ij} \le p_{j,k},\ k = a,\ldots,a+K, \quad \forall (i,j) \in \mathcal{C}_R(t_k), \tag{11c}$$

$$t^{\text{out}}_{i,r} \le t^{\text{in}}_{i,r},\ (i,j,r) \in \mathcal{S}(t_k), \tag{11d}$$

where $\mathcal{S}(t_k)$ is the ordered version of $\mathcal{C}_S(t_k)$, encoding the crossing order given by the selected heuristic.

In this way, the selection of the crossing order is removed from the MPC problem, and the optimisation problem solved for the control commands is reduced to an NLP. Unfortunately, all heuristics will in general generate sub-optimal $\mathcal{S}(t_k)$. To reduce this inevitable sub-optimality, the heuristic can be designed to take the objective function and constraints of the original MINLP (10) into account to some extent, for example, as proposed in [37]. There $\mathcal{S}(t_k)$ is obtained by solving a low-dimensional, Mixed-Integer Quadratic Program (MIQP) approximation of (10), defined as in (12), where T_i collects $t^{\text{in}}_{i,r}$ and $t^{\text{out}}_{i,r}$, $\forall r \in \mathcal{R}_i$; function $\tilde{V}(T_i)$ is the second-order Taylor expansion of $V_i(T_i)$, defined as

$$V_i(T_i) = \max_{x_i,u_i} \quad J_i(x_i,u_i) \tag{12a}$$

$$\text{s.t.} \quad (8),\ (4), \tag{12b}$$

and $\tilde{h}(T_i)$ is a polytopic approximation of the domain of $V_i(T_i)$:

$$\max_{T_i} \quad \sum_{i\in\mathcal{N}(t_k)} \tilde{V}_i(T_i), \tag{13a}$$

$$\text{s.t.} \quad \tilde{h}(T_i) \le 0, \quad \forall i \in \mathcal{N}(t_k), \tag{13b}$$

$$(5)\ \forall (i,j,r) \in \mathcal{C}_S(t_k), \tag{13c}$$

Function $V_i(T_i)$ is the value function of vehicle i, *given* the time of entry and departure from the intersection T_i. Constraints (1b) impose limitations on T_i, such that the domain of $V_i(T_i)$ is bounded. MIQP (12) finds the times T_i, $\forall i \in \mathcal{N}(t_k)$, that minimise the sum of quadratic approximations of all $V_i(T_i)$ over polytopic approximations of their domains, subject to SICA constraints (5), but ignoring RECA constraints (6). The crossing order $\mathcal{S}(t_k)$ can thereafter be extracted from the optimal T_i. Details on the derivation of the MIQP heuristic and its implementation are given in [37].

Besides sub-optimality, with the proposed two-stage procedure, it is, in general, not possible to guarantee (a) that a feasible solution to (11) exists for the crossing order $\mathcal{S}(t_k)$; nor (b) that the heuristic will find an order if one exists. This is due to the reliance on approximations of the domain of $V_i(T_i)$ and the removal of the RECA constraints in the heuristic. However, due to the safety-critical nature of the application, it is paramount that the closed-loop coordination is persistently safe, i.e. that the controller does not bring the

CAVs to configurations where collisions are unavoidable. In the next section, we detail how this issue can be tackled, if a non-restrictive assumption is imposed on the CAVs arriving at the intersection.

2.5. Persistent safety

Persistent safety for constant $\mathcal{N}(t)$ was established in [38, Proposition 5], in order to extend such result to a time-varying set $\mathcal{N}(t)$, the following assumption is introduced.

Assumption 2.3: When a vehicle i is added to $\mathcal{N}(t)$ at time t_k, its state $x_{i,k}$ is such that feasible control commands exist such that collisions are avoided with all other vehicles already present in $\mathcal{N}(t)$ at all times.

Assumption 2.3 is necessary to guarantee the existence of a solution to the coordination problem. This assumption holds if, for example, vehicles added to the coordination problem are able (i) to brake so as to avoid collision with the vehicle immediately in front of the same lane, even if this brakes to its full capacity and (ii) to stop before the intersection. We can now state the following result.

Proposition 2.1 ('Nominal' Persistent Safety): *If Assumption* 2.3 *holds and the approximate time slot allocation problem and the fixed-order problem are feasible at all times, the system is persistently safe.*

Proof: If a vehicle is removed from $\mathcal{N}(t)$, the set of feasible solutions for the remaining vehicles cannot be smaller than in the case of constant $\mathcal{N}(t)$ and does therefore not jeopardise persistent safety. When a vehicle is added to $\mathcal{N}(t)$, Assumption 2.3 ensures that it can execute at least one collision-free trajectory. Therefore, the closed-loop application of the two-staged controller is persistently safe. ■

Due to Proposition 2.1, we can conclude that if fixed-order problem (11) 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 $\mathcal{N}(t)$ is constant or a vehicle is removed; (b) the order $\mathcal{S}(t)$ is updated by adding a vehicle so that it crosses the intersection after all other vehicles in $\mathcal{N}(t)$. The latter option is always feasible since the added vehicle can stop before the intersection due to Assumption 2.3. Therefore, if at t_{k+1} either the calculation of a new crossing order fails or the crossing order is updated such that fixed-order problem is infeasible, a safe-guard mechanism can be implemented where the fixed-order problem is re-solved using the crossing order from t_k. This entails the following theorem.

Theorem 2.2 (Persistent Safety): *If Assumption* 2.3 *holds and the safe-guard mechanism is employed, the system is persistently safe.*

A note on stability: Asymptotic stability for the coordination of CAVs at intersections for fixed $\mathcal{N}(t)$ has been established in [38, Theorem 6] under the assumption that V_i^{f} and ℓ_i are selected such that they provide asymptotic stability guarantees for vehicle i in the absence of RECA and SICA constraints. The extension to time-varying $\mathcal{N}(t)$ is straightforward and out of the scope of this paper.

3. 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 2, with one lane per direction. 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 neighbouring vehicles on each lane. We divide the area around the intersection in two zones: the Scenario Zone (SZ), where vehicles are added and removed from the scenario; and the Intersection Zone (IZ), where coordination is performed. All vehicles travel between their initial position and the border of the IZ without performing any coordinating action, and the different coordination controllers are applied once the vehicles enter the IZ. We consider symmetric SZ and IZ, where we denote the entry and departure positions of the SZ as $p^{\mathrm{e}}_{\mathrm{SZ}}$ and $p^{\mathrm{d}}_{\mathrm{SZ}}$, respectively, and similarly denote the counterparts for the IZ as $p^{\mathrm{e}}_{\mathrm{IZ}}$ and $p^{\mathrm{d}}_{\mathrm{IZ}} = p^{\mathrm{d}}_{\mathrm{SZ}}$. Note that the SZ is introduced to enable arrival of vehicles in the IZ such that Assumption 2.3 holds. Moreover, vehicles which have left the IZ can still influence the solution, since they might force vehicles still in the IZ to slow down to avoid collisions.

Vehicle arrival and removal: The arrival of new vehicles to the scenario is modelled as a Poisson Point Process (PPP) and we let the time interval d between the arrivals of two consecutive vehicles on a lane be drawn from the exponential distribution $d \sim \lambda e^{-\lambda d}$, with rate parameter $\lambda \in [0, d^{\max}]$. Vehicle i is added to the scenario at a time t^{e}_i with initial velocity v^{e} at position $p_i(t^{\mathrm{e}}_i) = \min(p_{\mathrm{SZ}}, p_j(t^{\mathrm{e}}_i) - \delta p^{\mathrm{safe}})$, 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}. Finally, the scenario includes both passenger *Cars* and *Trucks*, where the vehicle type is drawn randomly on generation with probabilities p^{Car}, p^{Truck}, respectively, and a vehicle is removed from the scenario when it leaves the SZ. We denote the time of generation, type, and position at generation for all vehicles introduced to 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 given λ.

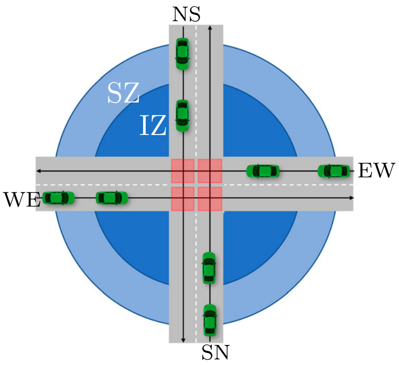

Figure 2. Scenario for the performance evaluation.

RECA before the IZ: To enforce Assumption 2.3, all vehicles outside the IZ are controlled to enforce RECA even if the vehicle in front applies maximal braking. If the velocity drops inside the IZ, vehicles outside the IZ will be forced to slow down.

Simulation Termination: The simulation is terminated when the end time is reached or the scenario is considered congested, i.e. when a new vehicle cannot be added to the scenario without the risk of violating the RECA conditions. This is the case 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.

3.1. Evaluated controllers

We denote the two-stage procedure introduced in Section 2.3 as the *MIQP/Fixed Order* (MIQP/FO) controller and compare it to the following four strategies.

Overpass: This scheme corresponds to a physical separation of the roads, and it is used to remove any cost of coordination. Hence, the *Overpass* 'controller' does not issue any coordinating action when the vehicles are inside the IZ (and side collisions can occur). All vehicles travel with the initial velocity v^{e}, which is generated at a safe inter-vehicle distance, until the end of the SZ.

Traffic light: In this scheme, the red and green phases of the two roads alternate with cycle time C, without an intermediate yellow phase. The vehicles are assumed to know the times for both phase-shifts and the intended trajectory of the preceding vehicle. As a consequence, all vehicles move synchronously from standstill after a red-light phase has passed, but in a manner which minimises $J_i(w_i)$ and satisfies $g_i(x_{i,k}, u_{i,k})$. However, no vehicle takes any action to favour other vehicles.

Sequential controller: In the *Sequential* controller, the vehicles decide how to cross the intersection in sequence based on a priority ranking. 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 minimises 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.

FCFS/FO controller: In the *First-Come-First-Served/Fixed-Order (FCFS/FO)* coordination controller, the Fixed Order (FO) Problem is solved in a receding horizon fashion for all vehicles in the IZ, using a crossing order selected through an FCFS heuristic: any vehicle entering the IZ is required to yield to all vehicles already in the IZ. If more than one vehicle enters the IZ simultaneously, they are sorted based on their expected arrival to the intersection when the SICA constraints are ignored and added to the crossing order accordingly. Similar ordering policies are used in the FCFS/FO controller and the Sequential controller. However, as opposed to the latter and similarly to the MIQP/FO case, the control commands by the FCFS/FO are found through simultaneous optimisation of all vehicles trajectories. As a result, some vehicles might take actions that increase their own objective functions, but yield a decrease for the objective of the scenario as a whole.

3.2. Motion models and control objectives

All controllers use the following prediction model:

$$F_i(x_{i,k}, u_{i,k}) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_{i,k} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} u_{i,k}, \quad a_i^{\min} \leq u_{i,k} \leq a_i^{\max}; \tag{14}$$

no model-plant mismatch is introduced and the vehicles' objective functions are

$$J_i(x_i, u_i) = m_i \left(Q_i^{\mathrm{f}}(v_{i,K} - v^r)^2 + \sum_{k=0}^{K} Q_i(v_{i,k} - v^r)^2 + R_i u_{i,k}^2 \right), \tag{15}$$

where $Q_i^{\mathrm{f}}, Q_i, R_i \in \mathbb{R}$ are weighting factors, $v^{\mathrm{r}} = v^{\mathrm{e}}$ is a reference velocity and m_i is the vehicle mass. Although the prediction model is simple, experimental results indicate that it can be sufficiently accurate for this application [38].

3.3. Secondary performance objectives

Two often cited reason for using coordination controllers is the reduction of energy consumption and travel time [12,18,21]. While not explicitly optimised through control objective (15), we also assess the performance of the coordination controllers w.r.t. these quantities.

Note that even though quadratic objective (15) does not explicitly describe the secondary objectives, the velocity deviation term $(v_{i,k} - v^{\mathrm{r}})^2$ penalises low velocities and will indirectly force the travel time delay δt_i to be small. Furthermore, the acceleration term $u_{i,k}^2$ is proportional to the square of the forces applied to the vehicles which relates to the energy supplied by the propulsion system. Keeping the acceleration term small will consequently yield an energy consumption close to E_i^{OP}.

Travel time delay: The 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} (the Overpass Case), i.e.

$$\delta t_i = t_i^{\mathrm{d}} - t_i^{\mathrm{e}} - \frac{p_i(t_i^{\mathrm{d}}) - p_i(t_i^{\mathrm{e}})}{v^{\mathrm{e}}}, \tag{16}$$

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

Energy consumption: The energy cost of the coordination is assessed by introducing an Electric Vehicle (EV) modelled as

$$\dot{v}_i(t) = \frac{1}{m_i} \left(\frac{G_i}{r_i^{\mathrm{w}}} M_i(t) - b_i(t) - \frac{1}{2} \rho a_i c_i^{\mathrm{d}} v_i(t)^2 - m_i g c_i^{\mathrm{rr}} \right), \tag{17}$$

where $M_i(t)$ is the electric motor torque 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 $c_i^{\mathrm{d}}, c_i^{\mathrm{rr}}$. The energy consumption associated with the closed-loop trajectory of vehicle i is calculated as

$$E_i = \sum_{k \in \mathcal{K}_i} \int_{k\Delta t}^{(k+1)\Delta t} \frac{\omega_i(t) M_{i,k}}{\eta_i(\omega_i(t), M_{i,k})} \, \mathrm{d}t, \tag{18}$$

where $M_{i,k}$ is the electric motor torque applied between $k\Delta t$ and $(k+1)\Delta t$, $\eta_i : \mathbb{R}^2 \mapsto]0,1]$ is the electric motor's efficiency map, $\omega_i(t) = G_i/r_i^{\mathrm{w}} v_i(t)$ is the electric motor speed and $\mathcal{K}_i$ collects the time instants when vehicle i is in the SZ. We define the *cost of coordination (CoC)* for vehicle i as the energy consumption increase with respect to the energy E_i^{OP} consumed by the Overpass controller, i.e.

$$E_i^{\mathrm{CoC}} = E_i - E_i^{\mathrm{OP}}. \tag{19}$$

4. 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 $R = 10{,}000$ vehicles/hour (1000–2500 vehicles/hour/lane). The parameters used in the simulations are summarised 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 emphasise that a fully centralised solution is not a necessity and that one could employ the distributed methods of [27–30] to solve the fixed-order problem. Animations showing the results can be found at [44]. Videos showing how the proposed coordination scheme works with real vehicles can also be found at [45], containing the material from the experimental validation presented in [38].

Table 1. General parameters.

Type	Symbol	Value	Unit
SZ Start	$p^{\mathrm{e}}_{\mathrm{SZ}}$	−350	m
SZ Stop	$p^{\mathrm{d}}_{\mathrm{SZ}}$	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
Discretisation 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 2. Vehicle parameters.

		Value		
Type	Symbol	Car	Truck	Unit
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		$\mathrm{m/s^2}$
Acceleration U.B.	$a_i^{\max}$	3		$\mathrm{m/s^2}$
Gear ratio	G_i	7.94	15	
Wheel radius	r_i^{w}	0.35		m
Projected Front Area	a_i	2.3	4	$\mathrm{m^2}$
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

4.1. Performance metrics

The performance scores for all controllers are computed as the average overall vehicles that have both entered and left the scenario during the simulation time. For objective (15), we define the average closed-loop cost associated to velocity and acceleration

$$\hat{J}^{v} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|}\sum_{i\in\mathcal{N}^{\mathrm{c}}}\sum_{k\in\mathcal{K}_i} m_i Q_i (v_{i,k} - v^{\mathrm{r}})^2, \quad \hat{J}^{u} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|}\sum_{i\in\mathcal{N}^{\mathrm{c}}}\sum_{k\in\mathcal{K}_i} m_i R_i u_{i,k}^2, \tag{20}$$

where $\mathcal{N}^{\mathrm{c}}$ collects the indices of all vehicles that cross the SZ within S. Similarly, for the secondary objectives, we define the average total cost of coordination and travel time delay induced

$$\hat{E}^{\mathrm{CoC}} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|}\sum_{i\in\mathcal{N}^{\mathrm{c}}} E_i^{\mathrm{CoC}}(x_i), \quad \hat{\delta t} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|}\sum_{i\in\mathcal{N}^{\mathrm{c}}} \delta t_i, \tag{21}$$

with E_i^{CoC} and δt_i given by (19) and (16). For comparison, we also consider the percentage change in energy consumption, with respect to the Overpass solution:

$$\hat{E}^{\%} = \frac{\sum_{i\in\mathcal{N}^{\mathrm{c}}} E_i}{\sum_{i\in\mathcal{N}^{\mathrm{c}}} E_i^{\mathrm{OP}}} \times 100. \tag{22}$$

The efficiency map η 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.

4.2. Performance results

Figures 3–5 summarise the main results of the performance evaluation. The simulation termination discussed further in Section 4.3 results in 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.

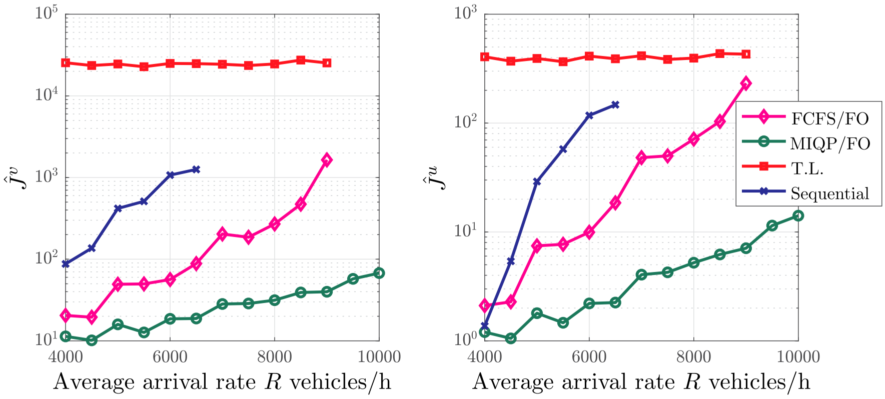

Figure 3. Components of the quadratic objective. T.L. denotes the traffic light controller.

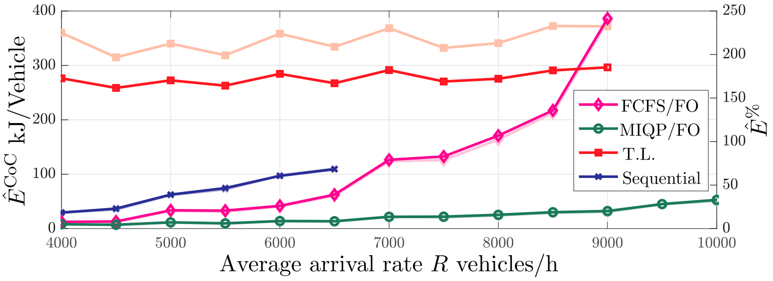

Figure 4. Energy cost of coordination: percentage increase $\hat{E}^{\%}$ (solid line, right axis); and CoC increase $\hat{E}^{\text{CoC}}$ (pale lines, left axis). Colour coding as in Figure 3.

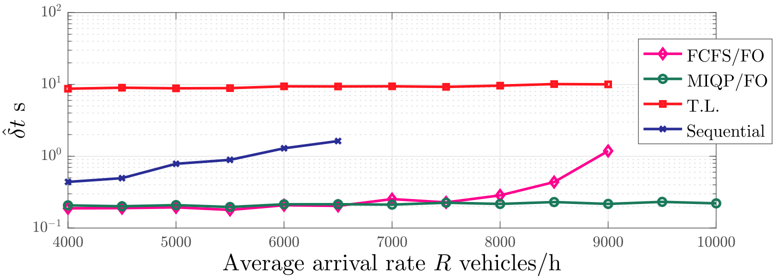

Figure 5. Travel time delay compared to the Overpass solution.

Traffic light vs. automated controllers: The difference 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. We highlight in particular the performance of the proposed MIQP/FO controller in terms of energy consumption: it can handle very high traffic intensities ($R =$ 10,000) with an energy increase not exceeding 40% of the Overpass controller energy. This is mainly due to the automated controllers' ability to coordinate the vehicles without forcing them to stop.

Effect of joint optimisation: The performance of the different automated controllers increases with added complexity. FCFS/FO outperforms the sequential controller in all performance metrics. Additionally, MIQP/FO outperforms FCFS/FO in all performance metrics except the travel-time delay, where the FCFS/FO and MIQP/FO perform closely for $R \leq 8000$ vehicles/h, with close to zero average delay. This is a consequence of the joint optimisation of the trajectories, which increases the velocity of some vehicles (resulting in 'negative' delays), and decreases velocities of other vehicles (resulting in positive delays), with an average close to zero. This is further illustrated in Figure 6, which shows estimates of the distribution of travel time delays under the three automated controllers.

A closer look at the vehicle velocities: The results in Figures 4 and 5 are explained by the velocity profiles in Figure 7. In particular, smaller velocity variations (decelerations) lead

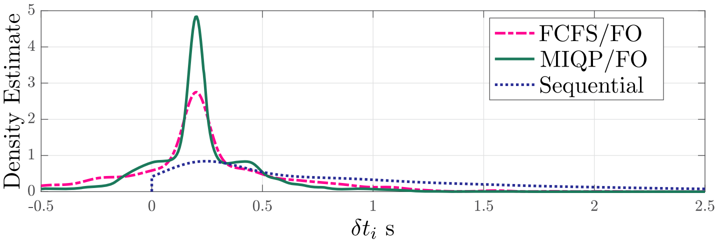

Figure 6. Estimate of the distribution of the observed δt_i for $R = 6000$ vehicles/h.

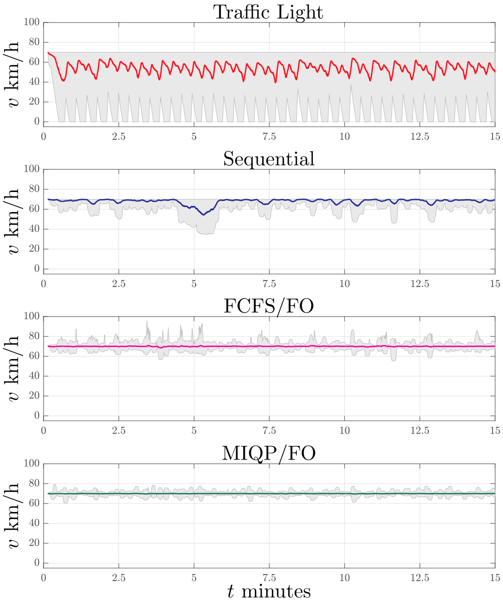

Figure 7. Average velocity (coloured lines) and velocity intervals (grey surface) in a scenario with $R = 6000$ vehicles/hour.

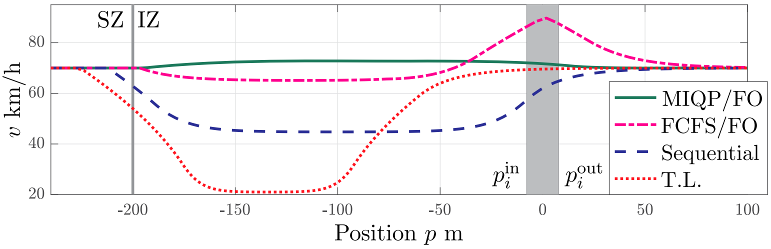

Figure 8. Position-velocity trajectories of one vehicle from the scenario with $R = 6000$ vehicles/hour. The grey bar corresponds to positions inside the intersection, 0 being the centre, whereas the grey line demarcates the beginning of the IZ.

to more efficient coordination policies. The maximum and minimum velocity profiles of the MIQP/FO are smoother than the ones of the FCFS/FO scheme (see e.g. around the 5-minute mark). A closer look at one spike is provided in Figure 8, which shows the position-velocity profiles from the same vehicle for the different controllers. As the figure illustrates, the spikes occur when the optimal solution to the fixed, FCFS-crossing order problem strongly accelerates some vehicles through the intersection. While such manoeuvre is costly, it allows other vehicles to use the intersection more efficiently. 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 reveals the MIQP's ability to select crossing orders that are convenient for the underlying fixed-order coordination problem.

4.3. 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 in order to avoid collisions and satisfy Assumption 2.3. 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 4).

4.3.1. Causes of the FCFS/FO failure

We first note that the fixed-order problem is expected to assign a relatively higher control effort and result in larger velocity deviations for Cars than Trucks due to the objective weighting (cf. objective (15) and Table 2). Regardless of how the crossing order is selected, the vehicles of the Car type are therefore expected to perform more aggressive manoeuvres in general, and attaining the maximum and minimum velocities (cf. the velocity intervals of Figure 7).

However, the magnitude of both control effort and velocity deviations depend on the selected crossing order: a Car which crosses after a Truck under the FCFS policy could be commanded to slow down significantly to decrease the total cost, whereas an alternative order inducing the Car to cross before the Truck would result in a velocity increase. 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.

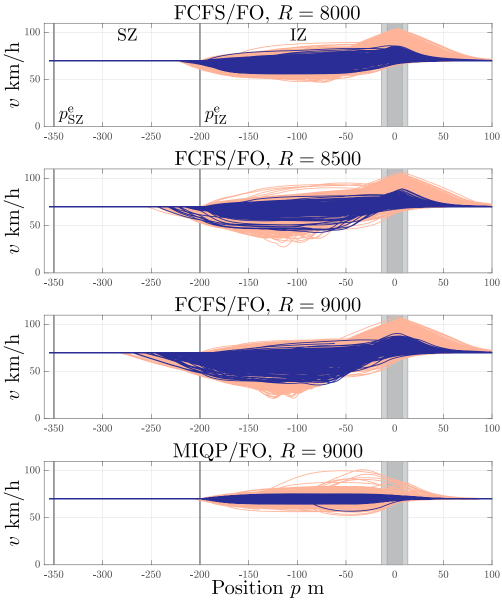

Figure 9. Speed profiles of Trucks (dark) and Cars (pale). The solid bars and the two vertical lines mark the intersection for Cars (dark) and Trucks (light) and the start of the SZ and IZ, respectively.

This effect is illustrated in Figure 9, 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. 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 behaviour is primarily seen in Cars: the optimal solution is often to first slow down to favour 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 since safety must be enforced and 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 9 illustrate 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, thus demonstrating the benefit of selecting a crossing order which takes the objective function and constraints into account at least approximately.

4.3.2. Causes of the sequential controller failure

In the sequential controller case, no vehicle increases its velocity to favour another, and collisions are avoided solely through velocity reductions. This propagates backwards on each

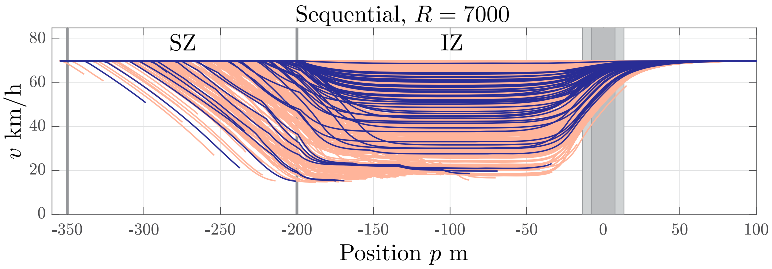

Figure 10. Velocity profiles for vehicles in the scenario with $R = 7000$, where the sequential controller failed. Colouring as in Figure 9.

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 10, where significant velocity reductions are present in the entire IZ, and propagate further backwards until they reach p^{e}_{SZ} and the simulation is terminated.

5. 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 outperform 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 optimising 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.

We also emphasise that even though the proposed MIQP/FO controller relies on approximations and is sub-optimal, the cost increase stemming from 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 large, in particular for higher traffic intensities. The improvement is obtained at the price of a higher computational demand, which, however, can be maintained within reasonable limits by a tailored distributed problem formulation [38] for the continuous part of the problem. The solution times of the MIQP have also been observed to be real-time feasible, especially when the amount of coordinated vehicles is less than 30. However, integer feasible solutions with a small optimality gap were typically quickly computed for larger scenarios. It would, therefore, be possible to stop the MIQP solver prematurely while retaining persistent safety, at the expense of possible sub-optimality. A deeper investigation of such aspects will be the subject of future research.

Finally, it is worth recalling that the results provided in this paper formally hold in the nominal case only. That is, the results on persistent safety in Section 2.5 may not hold in the presence of model uncertainties, external disturbances and measurement noise. Nevertheless, experimental results have demonstrated the resilience of the proposed closed-loop coordination algorithms. Additionally, the algorithm can be adapted to explicitly account for uncertainty and, therefore, provide robustness guarantees.

Note

1. In case $v_i(t^{\text{in}}_{i,r}) = 0$, $t^{\text{in}}_{i,r}$ is not uniquely defined by $p_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}$, but rather by 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 (1) so that $\dot{p} \geq \epsilon$, for some small $\epsilon > 0$. Since $v_i(t^{\text{in}}_{i,r}) = 0$ will be rarely encountered in practice, we assume $v_i(t^{\text{in}}_{i,r}) \neq 0$ for ease of presentation.

Disclosure statement

No potential conflict of interest was reported by the author(s).

ORCID

Robert Hult http://orcid.org/0000-0002-1337-3880

References

[1] Sjöberg K, Andres P, Buburuzan T, et al. Cooperative intelligent transport systems in Europe: current deployment status and outlook. IEEE Veh Technol Mag. 2017;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 Commun Standards Mag. 2017;1(2):70–76.

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

[4] 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. Int J Intell Transp Syst Res. 2009;7(1):49–58.

[5] Colombo A, Del Vecchio D. Efficient algorithms for collision avoidance at intersections. In: ACM International Conference on Hybrid Systems: Computation and Control; 2012; Beijing, China. p. 145–154.

[6] 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; Anaheim, CA, USA. p. 573–578.

[7] Chen L, Englund C. Cooperative intersection management: a survey. IEEE Trans Intell Transp Syst. 2016;17(2):570–586.

[8] Rios-Torres J, Malikopoulos AA. A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. IEEE Trans Intell Transp Syst. 2017;18:1066–1077.

[9] Dresner K, Stone P. A multiagent approach to autonomous intersection management. J Artif Intell Res. 2008;31(1):591–656.

[10] Milanes V, Perez J, Onieva E, et al. Controller for urban intersections based on wireless communications and fuzzy logic. IEEE Trans Intell Transp Syst. 2010;11(1):243–248.

[11] Kowshik H, Caveney D, Kumar PR. Provable systemwide safety in intelligent intersections. IEEE Trans Veh Technol. 2011;60(3):804–818.

[12] 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; Oct; 2013; The Hague, Netherlands. p. 1456–1461.

[13] 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; Los Angeles, CA, USA. p. 2932–2937.

[14] Qian X, Grégoire J, de La Fortelle A, et al. Decentralized model predictive control for smooth coordination of automated vehicles at intersection. In: European Control Conference; July; 2015; Linz, Austria. p. 3452–3458.

[15] Kim K, Kumar PR. An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic. IEEE Trans Autom Control. 2014;59(12):3341–3356.

[16] Molinari F, Raisch J. Automation of road intersections using consensus-based auction algorithms. In: 2018 Annual American Control Conference (ACC); June; 2018; Milwaukee, WI, USA. p. 5994–6001.

[17] Makarem L, Gillet D. Model predictive coordination of autonomous vehicles crossing intersections. In: 16th International IEEE Conference on Intelligent Transportation Systems; Oct; 2013; The Hague, Netherlands. p. 1799–1804.

[18] 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.

[19] Sprodowski T, Pannek J. Stability of distributed MPC in an intersection scenario. J Phys Conf Ser. 2015;659:012049.

[20] 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.

[21] 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.

[22] Zhang Y, Cassandras CG. Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees. Automatica. 2019;109:108563.

[23] Katriniok A, Sopasakis P, Schuurmans M, et al. Nonlinear model predictive control for distributed motion planning in road intersections using PANOC. In: IEEE Conference on Decision and Control; Vol. 12; Nice, France; 2019.

[24] Murgovski N, de Campos GR, Sjöberg J. Convex modeling of conflict resolution at traffic intersections. In: IEEE Conference on Decision and Control; Dec; 2015; Osaka, Japan. p. 4708–4713.

[25] Riegger L, Carlander M, Lidander N, et al. Centralized mpc for autonomous intersection crossing. In: IEEE International Conference on Intelligent Transportation Systems; Nov; 2016; Rio de Janeiro, Brazil. p. 1372–1377.

[26] Tallapragada P, Cortés J. Coordinated intersection traffic management. IFAC-PapersOnLine. 2015;48(22):233–239.

[27] Hult R, Zanon M, Gros S, et al. Primal decomposition of the optimal coordination of vehicles at traffic intersections. In: IEEE 55th Conference on Decision and Control; Dec; 2016; Las Vegas, NV, USA. p. 2567–2573.

[28] Zanon M, Gros S, Wymeersch H, et al. An asynchronous algorithm for optimal vehicle coordination at traffic intersections. 20th IFAC World Congress; Toulouse, France; 2017.

[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.

[30] Shi J, Zheng Y, Jiang Y, et al. Distributed control algorithm for vehicle coordination at traffic intersections. In: European Control Conference; June; 2018; Limasol, Cyprus. p. 1166–1171.

[31] Kneissl M, Molin A, Esen H, et al. A feasible mpc-based negotiation algorithm for automated intersection crossing. In: European Control Conference; 06; 2018; Limasol, Cyprus. 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 Trans Intell Transp Syst. 2015;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; Changshu, China. p. 1432–1437.

[34] Bali C, Richards A. Merging vehicles at junctions using mixed-integer model predictive control. In: European Control Conference; 06; 2018; Limasol, Cyprus. 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; Chicago, IL, USA. p. 763–768.

[36] Hult R, Zanon M, Gros S, et al. Energy-optimal coordination of autonomous vehicles at intersections. In: European Control Conference; June; 2018; Limasol, Cyprus. p. 602–607.

[37] Hult R, Zanon M, Gros S, et al. An miqp-based heuristic for optimal coordination of vehicles at intersections. In: IEEE Conference on Decision and Control; Dec; 2018; Miami Beach, FL, USA. p. 2783–2790.

[38] Hult R, Zanon M, Gros S, et al. Optimal coordination of automated vehicles at intersections: theory and experiments. IEEE Trans Control Syst Technol. 2018;27(6):2510–2525.

[39] Hult R, Zanon M, Gros S, et al. Optimal coordination of automated vehicles at intersections with turns. In: European Control Conference; Naples, Italy; 2019.

[40] 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; July; 2004; New York, NY, USA. p. 530–537.

[41] Lee J, Park B. Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment. IEEE Trans Intell Transp Syst. 2012;13(1):81–90.

[42] Rawlings J, Mayne D. Model predictive control: theory and design. Madison (WI): Nob Hill; 2009.

[43] Bock H, Plitt K. A multiple shooting algorithm for direct solution of optimal control problems. IFAC Proc Volumes. 1984;17(2):1603–1608. 9th IFAC World Congress.

[44] Hult SR, Zanon M, Gros S, et al. Animated illustration of the MIQP/FO controller [Internet]; 2019. [cited 2019 Feb 26]. Available from: https://youtu.be/hUWQoaiqdAY.

[45] Hult R, Zanon M, Gros S, et al. Optimal coordination of three cars approaching an intersection [Internet]. 2017. [cited 2019 Feb 26]. Available from: https://youtu.be/nYSXvnaNRK4.

[46] Murgovski N, Johannesson LM, Egardt B. Optimal battery dimensioning and control of a CVT PHEV powertrain. IEEE Trans Veh Technol. 2014;63(5):2151–2161.