Optimization-based Coordination of Connected, Automated Vehicles at Intersections

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 February 4, 2020

ABSTRACT
In this paper, we analyze the performance 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 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

By combining Autonomous Driving (AD) technologies with communication [1,2], 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 in these respects was recognized over a decade ago, in the discussion following the 2007 DARPA Urban Challenge [3].

In this paper, we discuss the problem of coordinating connected and automated vehicles at intersections. Such problem is traditionally motivated by safety [4] and traffic efficiency [5] issues. In fact, with the introduction of connected automated vehicles the intersections could be managed completely by coordination algorithms rather than relying on traffic lights, road signs and right-of-way rules. However, the design of such coordination algorithms is challenging for several reasons [6]. Uncertainties in the perception of the surroundings, as well as impairments of the wireless communication channel [7], must be handled. Moreover, even in presence of perfect communication and sensing, the search of a solution to the coordination problem is in itself hard [8], where

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

one particular difficulty is finding the order in which the vehicles should cross the intersection. Finally, to compensate for uncertainties 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 been proposed by the control community [14–35]. However, due to its combinatorial complexity, the problem is rarely solved exactly 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 considered objective function, 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 as a whole, if that decision is detrimental to the vehicle itself. As a consequence, the effort required to

resolve difficult conflicts 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 with the problem size (i.e., number fo vehicles) and are therefore practically limited 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 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.

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 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. In order to be able to use this scheme, we extend the work of [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.

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 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.

The remainder of the paper is organized 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 state both a general optimal control formulation of the coordination problem and 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.*

That is, we do not consider scenarios with, e.g., legacy vehicles, pedestrians or bicyclists. This assumption, though standard in the literature on vehicle coordination problems [13,14,23,40,41], is restrictive and limits the applicability to traffic scenarios in a distant future. We stress that our approach can be straightforwardly extended to accommodate for non-cooperative entities. This will be the subject of future research.

In this paper we consider the nominal situation, i.e., we assume that the model is prefectly describing the real system. While this is clearly not possible, we have observed in experiments that, even when relatively large measurement errors are introduced, the real system has relatively small deviations form the nominal prediction [38]. A thorough discussion on robustness is therefore avoided for simplicity in this paper and the interested reader is referred to [38].

2.1. Introduction to the optimal coordination problem

We consider cross-intersection scenarios, such as the one shown in Figure 1, consisting of four incoming lanes with continuously arriving vehicles. The problem of finding the control commands that optimize a given performance metric, reads conceptually 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 formal mathematical definition of (1) is provided in [38]. Problem (1) gives the control commands for all vehicles which satisfy all physical and design constraints (1c) (e.g., actuator limitations, comfort bounds, speed limits), result in collision free trajectories (1b),(1d), consistent with the vehicle dynamics and maximize the performance (1a) which can be, e.g., minimization of energy consumption, minimization of travel-time and maximization of intersection throughput.

Note that (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 [42]. 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 [42].

2.2. Scenario Modeling

In this subsection a model of the intersection scenario is presented, which is used in the formulation of the coordination problem. While one can model the motion of the vehicles in the intersection with arbitrary accuracy, the following assumption, widely accepted in literature [14,16,19,20,23,32], is convenient for our problem formulation and does not limit the generality of our approach.

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

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

where $i \in \{1,\ 2, \ldots, N(t)\} \subseteq \mathbb{N}$ 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 $g_i(\cdot)$ 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 makes a turn inside the intersection, but remark that such problems can be tackled as detailed in [39]. Additionally, we consider single-lane roads, but the formalism can handle more general layouts.

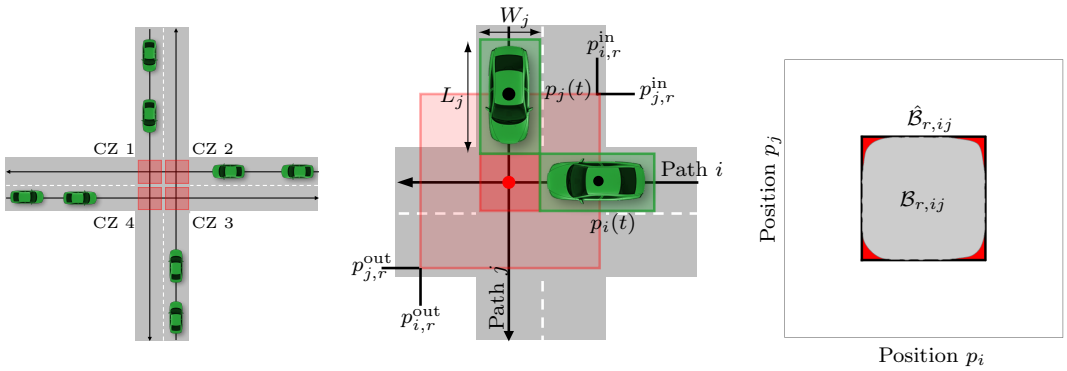
Side Collision Avoidance (SICA) Conditions. As illustrated in Figure 1a, side-collisions can occur between vehicles on different lanes when they are inside an area around the intersection of the vehicle paths, which we denote *Conflict Zone* (CZ). Collisions between vehicles i, and j in CZ r are avoided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

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

where $\mathcal{G}_i(p_i)$ is the area occupied by vehicle i in the horizontal plane when the path coordinate is p_i. As illustrated in Figure 1b, 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 (3) is formulated as

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

where $p_{r,i}^{\text{in}}$ and $p_{r,i}^{\text{out}}$ 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 introduced is very small (see Figure 1c) and is likely much smaller than the safety margins used in a real-world



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

b Scheme of the SICA condition (4): $\hat{\mathcal{G}}_i(p_i(t))$ shown as green boxes.

c Conservativeness induced by rectangular outer approximations (red).

Figure 1.: Illustrations of collision avoidance conditions.

setting. This approach is adopted 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 $v_i(t) \geq 0$, we have $p^{\text{in}}_{r,i} < p^{\text{out}}_{r,i} \Rightarrow t^{\text{in}}_{r,i} < t^{\text{out}}_{r,i}$ and SICA (4) reads as

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

i.e., either vehicle j must leave CZ r before vehicle i enters or vice-versa.

Rear-End Collision Avoidance (RECA) Conditions. Under Assumption 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 $\delta_{ij} = L_i/2 + L_j/2$, RECA is enforced by

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

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 $\epsilon_{ij} > 0$, either fixed or velocity dependent.

2.3. Closed-loop optimal coordination

We consider a scenario with N vehicles as in Figure 1a. The optimal coordination problem obtained by enforcing the system dynamics (2) and the SICA and RECA conditions (5)-(7), respectively, in problem (1) is a constrained finite-time optimal control problem in continuous time. The derivation of a discrete-time equivalent problem, that is necessary in order to be solved by a numerical solver, is detailed in [38].

[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 (2) 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.

The cost function in (1a) can be chosen as the sum of local costs

$$J_i(w_i) = V_i^{\mathrm{f}}(x_{i,N}) + \sum_{k=0}^{K-1} \int_{t_k}^{t_{k+1}} \ell_i(x_i(t), u_{i,k}) \mathrm{d}t. \tag{8}$$

If Problem (1) is to be solved in a receding-horizon fashion [42], the terminal cost $V_i^{\mathrm{f}}(x_{i,N})$ should be selected as suggested in [38] to enforce closed-loop stability. While solving the mixed-integer and nonconvex Problem (1) can be prohibitive, approximated solutions can be sought by resorting the two-stage procedure:

(1) The crossing order is found using an optimization-based heuristic [37],
(2) The state and control trajectories w that are optimal under that crossing order are found by solving a nonlinear program (NLP) [38,43].

We employ an optimization-based heuristic in Stage 1, which approximately accounts for the objective and constraints in Problem (1) when the order is selected [37]. We stress that, as highlighted in [37,38], this approach is not more computationally demanding than computing the schedule by some MIQP heuristic and controlling each vehicle to abide by the schedule.

2.4. Persistent Safety

Due to the safety-critical nature of the application, it is paramount that the closed-loop is persistently safe, i.e., the controller does not bring the vehicles to configurations where collisions are unavoidable. However, we note that the two-stage procedure can fail when a) the procedure to find a crossing order [37] fails, or b) when although a crossing order $\mathcal{S}$ can be computed, no solution exists to the fixed-order problem for $\mathcal{S}$. Although neither of those cases has been observed in simulations, a safe-guard mechanism handling such cases can be implemented. Denote by $\mathcal{C}$ the set of vehicles to be coordinated. Persistent safety when $\mathcal{C}$ is constant was established in [Proposition 5, [38]]. In order to extend such result to a time-varying set $\mathcal{C}$, the following assumption is needed.

Assumption 3.

(1) When a vehicle i is added to $\mathcal{C}$ at time t_k, its state $x_{i,k}$ is such that the vehicle 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.
(2) The vehicles that have not yet been included in the coordination do not collide with the vehicle ahead. Since this is a general requirement for all (automated) vehicles, it is not restrictive.

We can now state the following result.

Proposition 2.1 ("Nominal" Perpetual Safety)**.** *If Assumption 3 holds and the approximate timeslot allocation problem and the fixed order problem are feasible at all times, such that the system is perpetually safe.*

Proof. 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

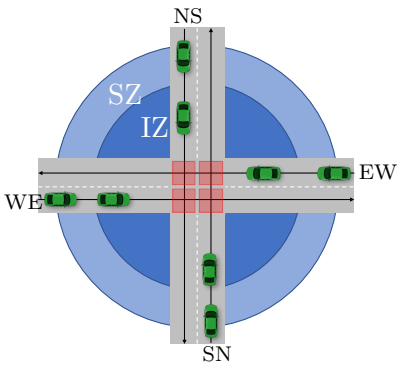


Figure 2.: Scenario for the performance evaluation.

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 2.1 we can conclude that if the fixed-order problem 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 is removed; b) the order $\mathcal{C}$ is updated by adding a vehicle 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 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 (Perpetual Safety)**.** *If Assumption 3 holds and the safe-guard mechanism is employed, the system is perpetually safe.*

A note on stability Conditions for stability for problems where $\mathcal{C}$ is constant was established in [Theorem 6, [38]]. While we haven't observed any issues, the extension to problems with changing $\mathcal{C}$ is the subject of current research.

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 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. 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 as follows. 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^{e}_{SZ} and p^{d}_{SZ}, respectively, and similarly denote the counterparts for the IZ as $p^{\mathrm{e}}_{\mathrm{IZ}}$ and $p^{\mathrm{d}}_{\mathrm{IZ}} = p^{\mathrm{d}}_{SZ}$. Note that the SZ is introduced to allow

us to generate vehicles in configurations which are safe, i.e., such that Assumption 3 is satisfied. Moreover, vehicles which have left the IZ can still influence the solution, since they might force preceding vehicles to slow down to avoid collisions.

Vehicle Arrival and Removal The arrival of new vehicles to the scenario is roughly modeled 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_i^{e} with initial velocity v^{e} at position $p_i(t_i^{\mathrm{e}}) = \min(p_{SZ}, p_j(t_i^{\mathrm{e}}) - \delta p^{\mathrm{safe}})$, where $p_j(t_i^{\mathrm{e}})$ 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. A vehicle is removed from the scenario when it leaves the 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 given λ.

Simulation Termination. The simulation is terminated when the simulation end time is reached or the scenario is considered congested. The latter is the case when a new vehicle cannot be added to the scenario. 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 the 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 directions (NS/SN and EW/WE) alternate with cycle time C, without an intermediate yellow phase. The vehicles are assumed to know both times for all phase-shifts and the intended trajectory of the preceding vehicle. 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 $g_i(x_{i,k}, u_{i,k})$. However, no vehicle takes any action to favor 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 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.

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 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. 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 optimization of all vehicles trajectories. As a result, some vehicles might take actions that increase their own objective functions, but yields a decrease for the scenario as a whole.

3.2. Motion Models and Control Objectives

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

$$F_i(x_{i,k}, u_{i,k}, \Delta t) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_{i,k} + \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} u_{i,k}, \qquad a_i^{\min} \leq u_{i,k} \leq a_i^{\max}. \tag{9}$$

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

$$J_i(w_i) = m_i \left(Q_i^{\mathrm{f}}(v_{i,K} - v^r)^2 + \sum_{k=0}^{K} Q_i(v_{i,k} - v^r)^2 + R_i u_{i,k}^2 \right), \tag{10}$$

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 sufficiently accurate for the application [38].

3.3. 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 optimized through the control objective (10), we also assess the performance of the coordination controllers w.r.t. these quantities.

Note that even though the quadratic objective (10) does not explicitly describe secondary objectives, 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. Furthermore, the acceleration term $u_{i,k}^2$ is proportional to 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: 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.,

$$\delta t_i = t_i^{\mathrm{d}} - t_i^{\mathrm{e}} - \frac{p_i(t_i^{\mathrm{d}}) - p_i(t_i^{\mathrm{e}})}{v^{\mathrm{e}}} \tag{11}$$

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) modeled as

$$\dot{v}_i(t) = \frac{1}{m_i}\left(\frac{G_i}{r_i^{\mathrm{w}}}M_i(t) - b_i(t) - \frac{1}{2}\rho a_i c_i^{\mathrm{d}} v_i(t)^2 - m_i g c_i^{\mathrm{rr}}\right), \tag{12}$$

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 state trajectory x_i is calculated as

$$E_i(x_i) = \sum_{k=k_i^{\mathrm{e}}}^{k_i^{\mathrm{d}}-1} \int_{k\Delta t}^{(k+1)\Delta t} \frac{\omega_i(t) M_{i,k}}{\eta_i(\omega_i(t), M_{i,k})} \mathrm{d}t. \tag{13}$$

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. We define the *cost of coordination (CoC)* for vehicle i as the energy consumption increase with respect to the energy $E_i^{\mathrm{OP}}(x_i)$ consumed by the Overpass controller, i.e.,

$$E_i^{\mathrm{CoC}}(x_i) = E_i(x_i) - E_i^{\mathrm{OP}}(x_i). \tag{14}$$

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 $R = 10000$ vehicles/hour (1000 to 2500 vehicles/hour/lane). The parameters used in the simulations 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. 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].

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.

4.1. 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.

For the objective (10) we define the average closed-loop cost associated to velocity and acceleration

$$\hat{J}^v = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} \sum_{k=k_i^{\mathrm{e}}}^{k_i^{\mathrm{d}}} m_i Q_i (v_{i,k} - v^{\mathrm{r}})^2, \qquad \hat{J}^u = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} \sum_{k=k_i^{\mathrm{e}}}^{k_i^{\mathrm{d}}} m_i R_i u_{i,k}^2, \tag{15}$$

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.

Similarly, for the secondary objectives, we define the average total cost of coordination and travel time delay induced

$$\hat{E}^{\mathrm{CoC}} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} E_i^{\mathrm{CoC}}(x_i), \qquad \hat{\delta t} = \frac{1}{|\mathcal{N}^{\mathrm{c}}|} \sum_{i \in \mathcal{N}^{\mathrm{c}}} \delta t_i, \tag{16}$$

with $E_i^{\mathrm{CoC}}(x_i)$ and δt_i given by (14) and (11). For comparison, we also consider the percentage change in energy consumption, with respect to the Overpass solution:

$$\hat{E}^{\%} = \frac{\sum_{i \in \mathcal{N}^{\mathrm{c}}} E_i(x_i)}{\sum_{i \in \mathcal{N}^{\mathrm{c}}} E_i^{\mathrm{OP}}(x_i)} \times 100 \tag{17}$$

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.

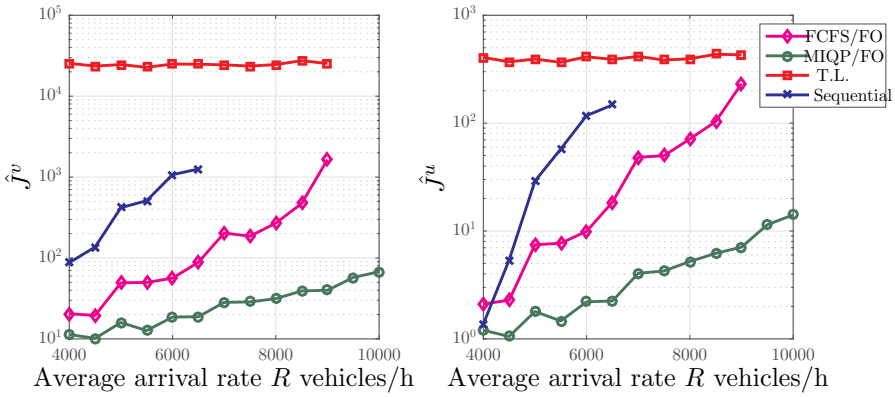


Figure 3.: Components of the quadratic objective. T.L. denotes the traffic light controller.

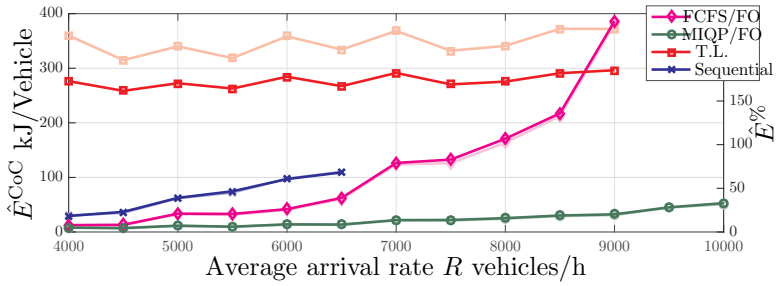


Figure 4.: Energy cost of coordination: percentage increase $\hat{E}^{\%}$ (solid line, right axis); and CoC increase $\hat{E}^{\text{CoC}}$ (pale lines, left axis). Color coding as in Figure 3.

4.2. Performance results

Figures 3, 4 and 5 summarize 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.

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 ($R = 10000$) with an energy increase not exceeding 40% of the Overpass controller energy.

Effect of joint optimization: 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 optimization of the trajectories, which increases the velocity of some vehicles (resulting in "negative" delays), and decreases velocities of other

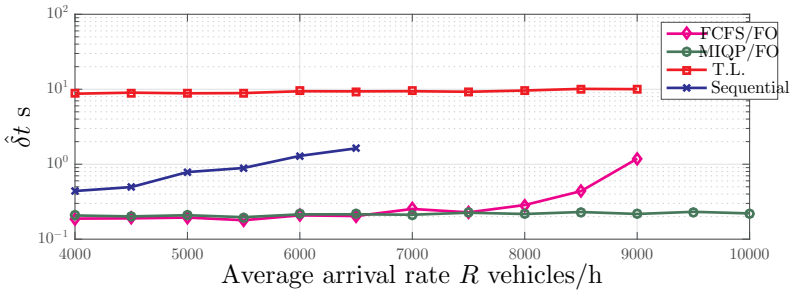


Figure 5.: Travel time delay compared to the Overpass solution.

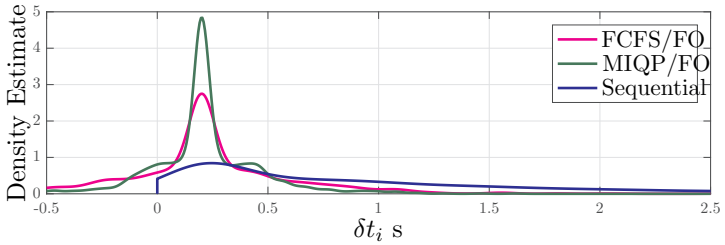


Figure 6.: Estimate of the distribution of the observed δt_i for $R = 6000$ vehicles/h.

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, 5 are explained by the velocity profiles in Figure 7. In particular, smaller velocity variations (decelerations) lead to more efficient coordination policies.

A small detail in Figure Figure 9 helps understanding the potential of MIQP/FO-like techniques. 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 9, 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 manuver 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, i.e., such that Assumption 3 holds. 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

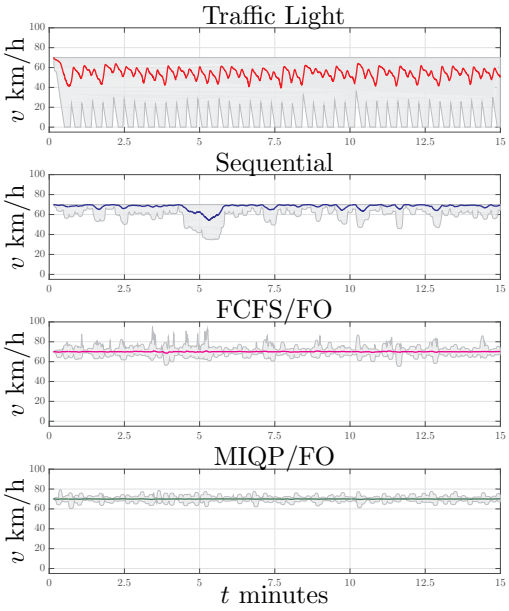


Figure 7.: Average velocity (colored lines) and velocity intervals (gray surface) in a scenario with $R = 6000$ vehicles/hour.

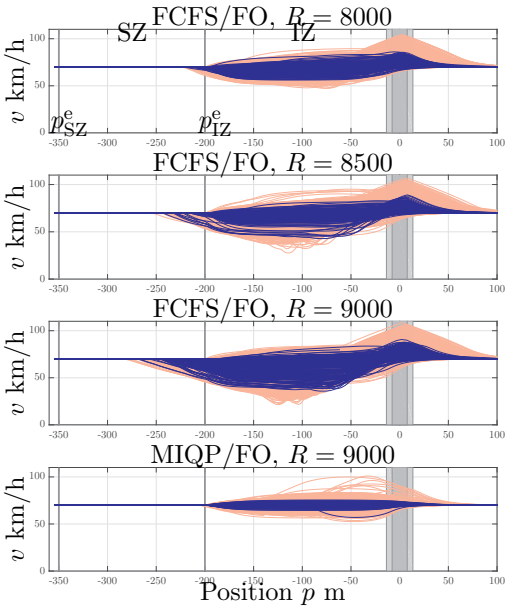


Figure 8.: Speed profiles of Trucks (blue) and Cars (pale red). The gray bars and the two vertical lines mark the intersection for Cars (dark) and Trucks (light) and the start of the SZ and IZ, respectively.

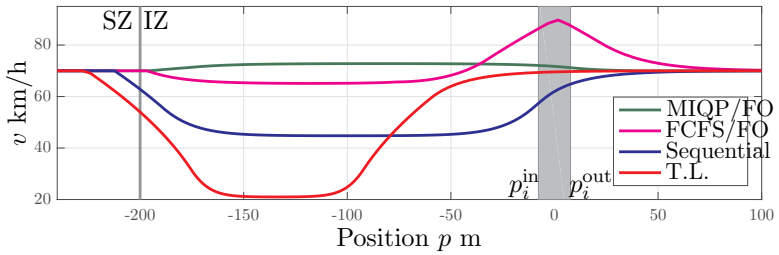


Figure 9.: Position-Velocity trajectories of one vehicle from the scenario with $R = 6000$ vehicles/hour. The gray bar corresponds to positions inside the intersection, 0 being the center, whereas the gray line demarcates the beginning of the IZ.

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 the objective weighting (c.f. objective (10) 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 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.

This effect is illustrated in Figure 8, 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 behavior is primarily seen in Cars: the optimal solution is often to first slow down 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 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 8 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 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 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.

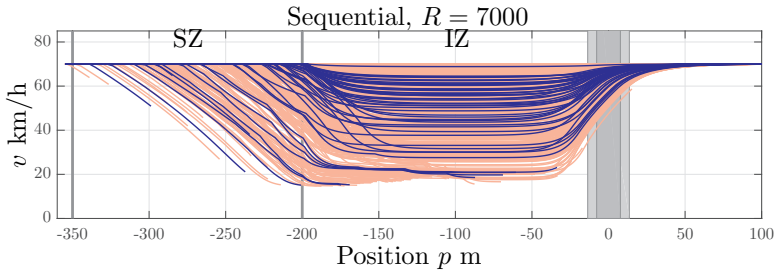


Figure 10.: Velocity profiles for vehicles in the scenario with $R = 7000$, where the sequential controller failed. Coloring as in Figure 8.

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 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.

We also emphasize that even though the proposed MIQP/FO controller relies on approximations and likely is sub-optimal, the price paid 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. The obtained improvement is obtained at the price of a higher computational demand, which, however, remains within reasonable limits, since we were able to run all simulations on a standard laptop.

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, 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, 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 wireless 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; 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: European Control Conference; 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; 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.
[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.
[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: IEEE Conference on Decision and Control; Dec; 2015. p. 4708–4713.
[25] Riegger L, Carlander M, Lidander N, et al. Centralized mpc for autonomous intersection crossing. In: IEEE 19th International Conference on Intelligent Transportation Systems; Nov; 2016. 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. 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.
[30] Shi J, et al. Distributed control algorithm for vehicle coordination at traffic intersections. In: European Control Conference; June; 2018. 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. 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; 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, Gros S, et al. Energy-optimal coordination of autonomous vehicles at intersections. In: European Control Conference; June; 2018. p. 602–607.
[37] Hult R, Zanon M, Gras S, et al. An miqp-based heuristic for optimal coordination of vehicles at intersections. In: IEEE Conference on Decision and Control; Dec; 2018. p. 2783–2790.
[38] 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.
[39] Hult R, Zanon M, Gros S, et al. Optimal coordination of automated vehicles at intersections with turns. In: European Control Conference; 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. p. 530–537.
[41] 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.
[42] Rawlings J, Mayne D. Model Predictive Control: Theory and Design. Nob Hill; 2009.
[43] Hult R, Zanon M, Frison G, et al. Experimental validation of a semi-distributed sqp method for optimal coordination of automated vehicles at intersections. Optimal Control Applications and Methods. 2020;(In press).
[44] Hult R, Zanon M, Gros WH S, et al. Animated illustration of the MIQP/FO controller [Online `https://youtu.be/hUWQoaiqdAY`]; 2019. 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`]; 2017.
[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.