Optimization based coordination of autonomous vehicles in confined areas

N.B. When citing this work, cite the original published paper.

(article starts on next page)

Optimization based coordination of autonomous vehicles in confined areas

Stefan Kojchev[1], Robert Hult[2] and Jonas Fredriksson[3]

Abstract— **Confined areas present an opportunity for early deployment of autonomous vehicles (AV) due to the absence of non-controlled traffic participants. In this paper, we present an approach for coordination of multiple AVs in confined sites. The method computes speed-profiles for the AVs such that collisions are avoided in cross-intersection and merge crossings. Specifically, this is done through the solution of an optimal control problem where the motion of all vehicles is optimized jointly. The order in which the vehicles pass the crossings is determined through the solution of a Mixed Integer Quadratic Program (MIQP). Through simulation results, we demonstrate the capability of the algorithm in terms of performance and satisfaction of collision avoidance constraints.**

I. INTRODUCTION

It is believed that fully automated vehicles (AV) have the potential to drastically change the transport industry, both in terms of increased safety and efficiency [1]. The most drastic improvements are expected when a substantial part of the vehicles on public roads are fully automated, as in e.g., "robo-taxis" and hub-to-hub transports on highways. Unfortunately, managing the unpredictable conditions on public roads in a reliably safe manner has proven to be harder than initially expected, and the current state-of-the-art exhibits a lack of production-level maturity.

However, confined areas, such as mines, ports, and logistic centers, lack many of the difficult aspects of public road driving, and present use cases for near-future, large-scale deployment of automated vehicles. Within this context, the AVs form a component in transport solutions for commercial operations, where for example material-flow can be handled without human involvement.

One of the challenges in such systems is the efficient coordination of multiple AVs use of mutually exclusive resources (MUTEX), such as intersections, narrow roads, work-stations (e.g. crushers, loading/unloading spots, etc.) and, in the case of electrified AVs, charging-stations. Poor coordination can lead to substantial decreases in productivity and energy-efficiency, reducing the benefits of automation.

The problem of handling mutual exclusive resources has been addressed for industrial robots [16], [17], where dif-

ferent scheduling algorithms have been explored. The coordination of multiple AVs, however, adds a different aspect to the challenge, where for example, the dynamics of the vehicles and the road topography play a significant part in the optimization problem. The coordination of automated vehicles at intersections has been widely discussed in the literature recently, see [2] for a comprehensive survey. In general, the problem is difficult to solve and has been formally shown to be NP-hard in [15]. Often relying on simplifying assumptions and heuristics, a number of methods has been presented that solve the problem using, e.g., hybrid system theory [3], reinforcement learning [4], scheduling [5], model predictive control (MPC) [6], [7] or direct optimal control (DOC) [8], [9].

Coordination of AVs in confined areas has some distinct differences compared to the intersection scenarios often found in the literature. For instance, the full site-layout of confined areas is typically known at the planning stage, and it can often be expected that no non-controlled actors will disturb the execution of a plan once it's formed. The motion of each vehicle can therefore be planned from the start of a transport mission to its end. Planning for confined sites is thus benefited by methods that can handle long planning horizons. This is in contrast to the intersection coordination context found in the literature, where a cutout around the intersection proper is most often considered, with the vehicles arriving at speed [6], [7].

In this paper, we formulate the MUTEX-coordination problem as an an optimal control problem. We adapt the two-stage heuristic procedure proposed in [9] to the confined-site context and employ it to solve the problem. Besides cross-intersections, we consider the merge-split MUTEX-zones, where the vehicles first join in on a common patch of road which after some distance separate. The approach is capable to optimize the vehicle trajectories over their full path and there are no limitations on the model that is used for the vehicles. Although the approach focuses on confined sites, the method of handling the mutual exclusion zones can be extendable to other scenarios as well (e.g., public road applications).

From similar scenarios that have been considered, the authors in [10] and [11] propose an optimization approach for handling merge scenarios, which is a subset of the merge-split collision zone, and [12] uses a game-theoretical strategy for optimizing traffic flow through multiple intersection collision zones. The handling of multiple intersections and zones of different types were also identified in the survey [2] as topics for further work in this field.

The remainder of the paper is organized as follows: Section II formulates the problem that is solved in this paper. In Section III the method for solving the stated problem is presented, followed by Section IV where simulation results illustrate the coordination algorithm. Section V concludes the work and provides some possible extensions.

II. PROBLEM FORMULATION

We consider N_a fully automated vehicles on a road network with cross-intersection, path merges and path splits. The road network is assumed to be fully in a confined area, such that non-controlled traffic participants (e.g. manually operated vehicles, pedestrians, bicyclists etc.) are absent. We further assume that the paths of all vehicles, i.e., their routes through the road network are known, that no vehicle reverses, and that overtakes are prohibited.

A. Vehicle modelling

The motion of the vehicles along their path is described by

$$\dot{p}_i(t) = v_i(t) \tag{1}$$
$$\dot{x}_i(t) = f_i(p_i(t), x_i(t), u_i(t)) \tag{2}$$
$$0 \leq h_i(p_i(t), x_i(t), u_i(t)). \tag{3}$$

where $p_i(t) \in \mathbb{R}$ is the position, $x_i(t) \in \mathbb{R}^n$ the vehicle state, $u_i(t) \in \mathbb{R}^m$ the control input, with $i \in \{1, \ldots, N_a\}$. The state is subdivided as $x_i(t) = (v_i(t), z_i(t))$, with the speed along the path $v_i(t) \in \mathbb{R}$ and $z_i(t) \in \mathbb{R}^{n-1}$ collecting possible other states. The functions f_i and h_i, both assumed smooth, describes the dynamics and constraints that capture, e.g., actuator and speed limits respectively.

B. Vehicle modelling in the spatial domain

For confined site optimization, it is beneficial to optimize the trajectories of the vehicles over their full paths. However, the time it takes a vehicle to traverse a path is dependent on the solution, and not known *a-priori*. Consequently, it is inappropriate to plan the vehicle's motion with time as the independent variable. Due to this, the problem is reformulated in the spatial domain, using that $\frac{dp_i}{dt} = v_i(t)$ and $dt = dp_i/v_i(t)$. The formulation of the vehicle dynamics (1) in the spatial domain is

$$\frac{dt_i}{dp_i} = \frac{1}{v_i(p_i)} \tag{4}$$
$$\frac{dx_i}{dp_i} = \frac{1}{v_i(p_i)} f_i(p_i, x_i(p_i), u_i(p_i)) \tag{5}$$
$$0 \leq h(p_i, x_i, u_i). \tag{6}$$

where the position p_i is the independent variable.

C. Conflict zone modelling

A *conflict zone* (CZ) is described by the entry and exit position $[p_i^{in}, p_i^{out}]$ on the path of each vehicle. From the known positions, the time of entry and exit of vehicle i is $t_i^{in} = t_i(p_i^{in})$, $t_i^{out} = t_i(p_i^{out})$, respectively. In this paper, two types of conflict zones are considered, as depicted in Figure 1: the "intersection-like" and the "merge-split". We let

$\mathcal{I} = \{I_1, I_2, \ldots, I_{r_0}\}$ denote the set of all intersections in the confined site, with r_0 being the total number of intersection CZs, and let $\mathcal{Q}_r = \{q_{r,1}, q_{r,2}, \ldots, q_{r,l}\}$ denote the set of vehicles that cross an intersection I_r. In the intersection-like CZ, it is desired to only have one vehicle inside the CZ, i.e., not allowing the vehicle j to enter the CZ before vehicle $i \neq j$ exits the CZ, or vice-versa. The order in which the vehicles cross the intersection I_r is denoted $\mathcal{O}_r^{\mathcal{I}} = (s_{r,1}, s_{r,2}, \ldots, s_{r,|\mathcal{Q}_r|})$, where $s_{r,1}, s_{r,2}, \ldots$ are vehicle indices and we let $\mathcal{O}^{\mathcal{I}} = \{\mathcal{O}_1^{\mathcal{I}}, \ldots, \mathcal{O}_r^{\mathcal{I}}\}$. A sufficient condition for collision avoidance for the r-th intersection CZ can be formulated as

$$t_{s_{r,i}}(p_{s_{r,i}}^{out}) \leq t_{s_{r,i+1}}(p_{s_{r,i+1}}^{in}), \ i \in \mathbb{I}_{[1,|\mathcal{Q}_r|-1]}, \tag{7}$$

where t is determined from (4). In the merge-split CZ case, let $\mathcal{M} = \{M_1, M_2, \ldots, M_{w_0}\}$ denote a set of all merge-split zones, with w_0 being the total number of merge-split CZs in the site and $\mathcal{Z}_w = \{z_{w,1}, z_{w,2}, \ldots, z_{w,h}\}$ denote the set of vehicles that cross the merge-split CZ M_w. For efficiency, it is desirable to have several vehicles in the zone at the same time, instead of blocking the whole zone. This requires having rear-end collision constraints once the vehicles have entered the CZ safely. In this case, the order in which the vehicles enter the zone is denoted as $\mathcal{O}_w^{\mathcal{M}} = (s_{w,1}, s_{w,2}, \ldots, s_{w,|\mathcal{Z}_w|})$, and we let $\mathcal{O}^{\mathcal{M}} = \{\mathcal{O}_1^{\mathcal{M}}, \ldots, \mathcal{O}_w^{\mathcal{M}}\}$. The collision avoidance requirement for this w-th CZ is described with the following constraints:

$$t_{s_{w,i}}(p_{s_{w,i}}^{in}) + \Delta t \leq t_{s_{w,i+1}}(p_{s_{w,i+1}}^{in} + c) \tag{8a}$$
$$t_{s_{w,i},k_i} + \Delta t \leq t_{s_{w,i+1}}(p_{s_{w,i},k_i} - p_{s_{w,i}}^{in} + p_{s_{w,i+1}}^{in} + c),$$
$$k_{s_{w,i}}^{in} \leq k_i \leq k_{s_{w,i}}^{out} \tag{8b}$$
$$t_{s_{w,i}}(p_{s_{w,i}}^{out}) + \Delta t \leq t_{s_{w,i+1}}(p_{s_{w,i+1}}^{out} + c), \tag{8c}$$
$$i \in \mathbb{I}_{[1,|\mathcal{Z}_w|-1]}.$$

That is, while in the CZ, the vehicles must be separated by at least a time-period Δt_i and a distance c_i or by Δt_j and c_j, depending on if vehicle j is in front of vehicle i or vice versa. This is equivalent to the standard offset and time-headway formulation often used in automotive adaptive cruise controllers.

D. Discretization

The independent variable is discretized as $p_i = (p_{i,1}, \ldots, p_{i,N_i})$, where p_{i,N_i} indicates the end position for vehicle i, and the input is approximated using zero order hold such that $u(p) = u_{i,k}, p \in [p_{i,k}, p_{i,k+1}[$. The equations (4), (5) are (numerically) integrated on this grid, giving the "discretized" state transition relation

$$\begin{bmatrix} t_{i,k+1} \\ x_{i,k+1} \end{bmatrix} = F(x_{i,k}, u_{i,k}, p_{i,k}, p_{i,k+1}) \tag{9}$$

where F denotes the integration of (4), (5) from $p_{i,k}$ to $p_{i,k+1}$.
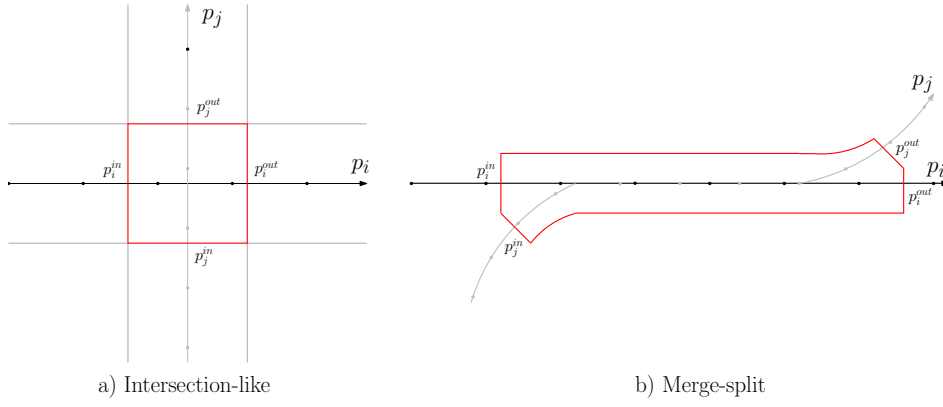
Fig. 1. Types of conflict zones.

a) Intersection-like   b) Merge-split

E. Optimal coordination problem

The problem of finding the optimal vehicle trajectories that avoid collisions can be formalized as:

Problem 1: (Optimal coordination problem) Obtain the optimal state and control trajectories $\mathcal{X}^* = \{x_1^*, ..., x_{N_a}^*\}$, $\mathcal{U}^* = \{u_1^*, ..., u_{N_a}^*\}$, given the initial state $\mathcal{X}_0 = \{x_{1,0}, ..., x_{N_a,0}\}$, by solving the optimization problem

$$\min_{x_{i,k}, u_{i,k}, \mathcal{O}^{\mathcal{I}}, \mathcal{O}^{\mathcal{M}}} \quad \sum_{i=1}^{N_a} J_i(x_{i,k}, u_{i,k}) \tag{10a}$$

$$\text{s.t} \quad \text{initial states } x_{i,0} = \hat{x}_{i,0}, \forall i \tag{10b}$$

$$\text{system dynamics } (9), \ \forall i, \forall k \tag{10c}$$

$$\text{state and input constraints } (6), \ \forall i, \forall k \tag{10d}$$

$$\text{safety constraints } (7),(8), \ \forall i, \forall k \tag{10e}$$

Note in particular that this involves finding the crossing orders $\mathcal{O}^{\mathcal{I}}$, $\mathcal{O}^{\mathcal{M}}$, which makes the problem combinatorial and difficult to solve.

III. METHOD

Problem 1 can be stated as a Mixed Integer Nonlinear Program (MINLP), where the crossing order correspond to the "integer part" and the state and control trajectories corresponds to the "NLP part". However, finding a solution to MINLP problems is known to be difficult, especially when the constraints or the objective function are non-convex [13]. Therefore, a common procedure is to apply an approach where the integer part of the solution is obtained first using a heuristic, and the continuous part of the solution thereafter is obtained by solving the nonlinear program (NLP) that results from fixing the integers to the values found with the heuristic. In this paper, we follow this approach. The heuristic that is used is similar to that of [9] which approximates the integer part of the solution of (10) by solving a Mixed Integer Quadratic Problem (MIQP). With the integer solution given, the state and control trajectories are obtained by solving the "fixed-order coordination" NLP, i.e., Problem (10) with fixed crossing orders $\mathcal{O}^{\mathcal{I}}$, $\mathcal{O}^{\mathcal{M}}$.

A. Crossing order heuristic

The crossing order heuristic follows the procedure proposed in [9] where the coordination problem is approximated as time-slot scheduling MINLP in $\mathbf{T}$, where $\mathbf{T} = (T_i, ..., T_{N_a})$ and T_i is the set of entry and exit times for vehicle i for all the CZ it encounters. With this formulation, the integer part of the solution, i.e., the crossing orders, are treated in $\mathbf{T}$ space. The problem can be formulated as

$$\min_{\mathbf{T}, \mathcal{O}^{\mathcal{I}}, \mathcal{O}^{\mathcal{M}}} \quad \sum_{i=1}^{N_a} V_i(T_i) \tag{11a}$$

$$\text{s.t. } T_i \in \text{dom}(V_i) \ \forall i, 1, \dots, N_a, (7), (8a), (8c), \tag{11b}$$

where V_i is the value function and $\text{dom}(V_i)$ is the domain of it. The definition of $V_i(T_i)$ follows next.

Note that the solution to (11) is an approximation of the solution to (10). In particular, the rear-end collision avoidance constraints (8b) are not enforced and could therefore be violated.

1) The vehicle problem: The function $V_i(T_i)$ is the optimal value function of the following parametric optimal control problem, denoted as the "Vehicle Problem":

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

$$\text{s.t. } (10b), (9), (6) \tag{12b}$$

$$\xi_i^{in} = t_i(p_i^{in}) \tag{12c}$$

$$\xi_i^{out} = t_i(p_i^{out}), \tag{12d}$$

with ξ_i being the parameter over which we optimize i.e. the in- and out-times of the CZs. Note that $\text{dom}(V_i)$ thereby is the set of parameters T_i for which (12) has a solution.

2) An MIQP-based heuristic: The time-slot scheduling problem (11) is still a difficult problem, with non-convex objective function and constraints. Motivated by the availability of efficient solvers of MIQPs, we therefore consider the following approximation of (11) using the first and second order Taylor expansion of the objective function and

constraints:

$$\min_{\mathbf{T}} \quad \sum_{i=1}^{N_a} \frac{1}{2} T_i^T \nabla^2 V_i T_i + \left(\nabla V_i - \nabla^2 V_i T_i^{[0]} \right)^T T_i \quad (13a)$$

$$\text{s.t.} \quad (7), (8a), (8c), \quad (13b)$$

where the derivatives are evaluated at $T_i^{[0]}$. Note that besides that the problem now is an MIQP, the constraint $\mathrm{dom}(V_i)$ is removed. As discussed in [9], the boundary of this constraint are all timeslots where the control is fully saturated prior to the intersection. This typically occur only in scenarios with a large number of vehicles that are forced to resolve conflicts over short distances, at high speeds. As such situations can be avoided in the target application, the constraint is removed.

The solution to the MIQP problem provides an approximately optimal time-slot schedule $\mathbf{T}$ and therefore approximate crossing orders $\hat{\mathcal{O}}^{\mathcal{I}}$, $\hat{\mathcal{O}}^{\mathcal{M}}$.

A comment on differentiation of $V_i(T_i)$: From the solution of the parametric NLP (12), the first and second order derivatives w.r.t. the parameter can be calculated using parametric sensitivity analysis [14]. In particular, this can be done at a low additional cost when the optimal problem is solved. Note that for a solution to problem (12) feasible entry and exit times, i.e., feasible values for the parameters, for each CZ that vehicle i encounters are necessary. These times can, for example, be computed by solving the optimization problem (10) without safety constraints (7), or a forward simulation of the vehicles with, for example, an LQR controller.

B. Fixed-order NLP

With the found crossing order, the integer part of the solution is obtained, making (10) an NLP. Obtaining the optimal state and control trajectories is thus found through solving the fixed-order coordination problem

$$\min_{x_{i,k}, u_{i,k}} \quad \sum_{i=1}^{N_a} J_i \left(x_{i,k}, u_{i,k} \right) \quad (14a)$$

$$\text{s.t} \quad (10b) - (10e), \forall i, \forall k \quad (14b)$$

$$\mathcal{O}^{\mathcal{I}} = \hat{\mathcal{O}}^{\mathcal{I}}, \quad \mathcal{O}^{\mathcal{M}} = \hat{\mathcal{O}}^{\mathcal{M}} \quad (14c)$$

The two stage approximation approach is summarized in Algorithm 1 and depicted in Figure 2.

Algorithm 1 Two stage approximation algorithm

Input: $N_a, \mathcal{I}, \mathcal{Q}_r, \mathcal{M}, \mathcal{Z}_w$, vehicle paths
Output: $\mathcal{X}^*, \mathcal{U}^*$

1: $\forall i$: Obtain feasible entry and exit times $T_i^{[0]}$ by, e.g., solving NLP (10) w/o safety constraints (7).
2: $\forall i$: Solve NLP (12) and calculate the first and second order sensitivities ($\nabla V_i, \nabla^2 V_i$).
3: Solve MIQP (13) for the approximate crossing orders $\hat{\mathcal{O}}^{\mathcal{I}}, \hat{\mathcal{O}}^{\mathcal{M}}$.
4: Solve the fixed-order NLP (14) using $\hat{\mathcal{O}}^{\mathcal{I}}, \hat{\mathcal{O}}^{\mathcal{M}}$ and obtain $\mathcal{X}^*, \mathcal{U}^*$.

IV. SIMULATION RESULTS

In this section, we present a simulation example showing the operation and performance of the coordination algorithm.

A. Simulation setup

The vehicles are modelled as a triple integrator $\dddot{x} = u$, whereby the spatial model is:

$$\begin{bmatrix} \frac{\mathrm{d}t_i}{\mathrm{d}p_i} \\ \frac{\mathrm{d}v_i}{\mathrm{d}p_i} \\ \frac{\mathrm{d}a_i}{\mathrm{d}p_i} \end{bmatrix} = \begin{bmatrix} \frac{1}{v_i} \\ \frac{a_i}{v_i} \\ \frac{u_i}{v_i} \end{bmatrix}, \quad (15)$$

where a_i is the acceleration and u_i is the jerk. The dynamics are discretized using multiple shooting (with N shooting points) and an Explicit Runge-Kutta-4 (ERK4).

The constraints are chosen as bounds on the speed and longitudinal acceleration ($\underline{v}_i \leq v_{i,k} \leq \overline{v}_i, \ a_{i,k} \leq \overline{a}_{i,lon}$) in order to obey speed limits and physical constraints. The lower bound on the speed is necessary to be non-zero in order to avoid singularity in the vehicle model. The vehicles are expected to manoeuvre on curved roads, thus, it is needed to consider the lateral forces they experience. As the one-dimensional model that is used in this paper does not account for lateral motion, the following constraint is enforced, as similarly proposed in [8].

$$\left(\frac{a_{i,k}}{a_{i,lon}} \right)^2 + \left(\frac{\kappa_i(p_{i,k})v_{i,k}}{\overline{a}_{i,lat}} \right)^2 \leq 1, \quad (16)$$

where $\overline{a}_{i,lat}$ is the lateral acceleration limit and $\kappa_i(p_{i,k})$ is the road curvature, that is assumed to be available at every point along the path.

While other objectives could be used, we consider the following trade-off between minimization of the total travel time and the squares of the longitudinal acceleration and longitudinal jerk:

$$J(x_i(p_i), u_i(p_i)) = \quad (17)$$
$$\int_{p_{i,0}}^{p_{i,N_i}} \left(P_i a_i(p_i)^2 + Q_i j_i(p_i)^2 \right) \frac{1}{v_i(p_i)} \mathrm{d}p_i + R_i t_i(p_{i,N_i}),$$

where P_i, Q_i, R_i are the appropriate weights. This objective is integrated with Forward Euler, leading to the "discretized" objective expression

$$J_i \left(x_{i,k}, u_{i,k} \right) = \quad (18)$$
$$\sum_{k=1}^{N} \left(\left(P_i a_{i,k}^2 + Q_i j_{i,k}^2 \right) \frac{\Delta p_{i,k}}{v_{i,k}} \right) + R_i t_{i,N},$$

with $\Delta p_{i,k} = p_{i,k+1} - p_{i,k}$.

For evaluation purposes, we are investigating a mock-up confined area with four vehicles as depicted in Figure 3. For this scenario, the vehicle paths are permanent throughout the simulation. There are in total two merge-split CZs and five intersection CZs. Every vehicle starts from an initial velocity of 50 km/h and the chosen bounds are $\underline{v}_i = 3.6$ km/h, $\overline{v}_i = 90$ km/h, $\overline{a}_{i,lon} = 4$ [m/s^2], $\overline{a}_{i,lat} = 2$ [m/s^2]. The weight coefficients for the optimal problem are: $P_i = 1, \ Q_i = 1, \ R_i = 10$, and are a suitable trade-off between
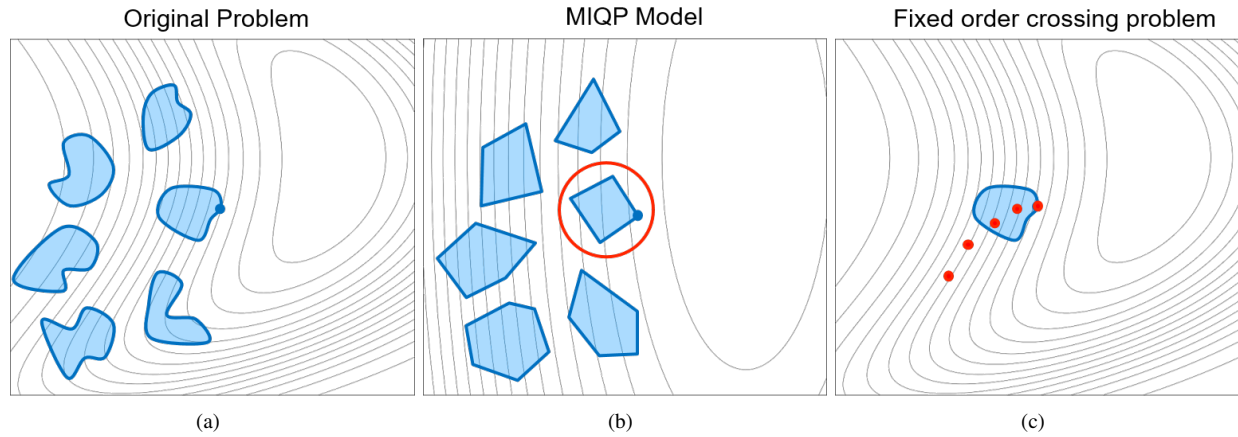
Fig. 2. Illustration of the heuristic approach. As seen, in (a) the original problem is a non-convex problem with a disconnected solution space (each compact subset corresponding to an integer combination), whose solution is, typically, difficult to obtain. The MIQP approximation (b) of the original problem allows for finding a region where the solution of (a) could be. Finally, through the fixed-order NLP (c), the approximate optimal solution is acquired.

the objectives for this scenario. The total shooting points for this scenario is $N = 100$ and is same for all vehicles. In the intersection-like scenario the CZ is created with 5 meter margin ahead of and behind the collision point, where as in the merge-split scenario the margin is 15 meters for both the entry and exit collision point. The margin should account for the length of a vehicle. For the merge-split CZ it is also desired to keep at least 0.5s margin between the vehicles. We utilize the CasADi [18] toolkit with the IPOPT [19] solver for solving the optimization problems (12), (14) and Gurobi for solving the MIQP problem (13).

B. Discussion of results

In the following, the results for this simulation scenario are presented. In particular, we demonstrate the uncoordinated and coordinated results for one merge-split zone and one intersection zone, both circled over in Figure 3. The unco-ordinated results are obtained as the individual optimum of each vehicle, i.e., the speed profiles are obtained by solving the optimization problem without any safety constraints.

Figure 4 and Figure 5 illustrate the uncoordinated and coordinated crossing of the merge-split CZ, respectively. In the figures, the path trajectories of the involved vehicles are depicted. The path trajectories have an offset of their entry position to the merge-split CZ, meaning that, for both vehicles, a position of zero indicates the entry position to the CZ. The gray rectangle depicts the CZ, with entry and exit times for the red vehicle, since that is the first vehicle that enters the zone in both cases. As mentioned, in this type of CZ it is desirable to have multiple vehicles inside the zone whilst keeping a minimum gap between them. This means that while it is allowed for both trajectories to be inside the zone, it is not allowed for the trajectories to intersect whilst in the zone. In the uncoordinated case, it can be noticed that the blue vehicle enters the CZ later than the red vehicle, however, intersects the red vehicle and exits the zone first. The interpretation of this behaviour is that inside the CZ the blue vehicle "overruns" the red vehicle, meaning that a
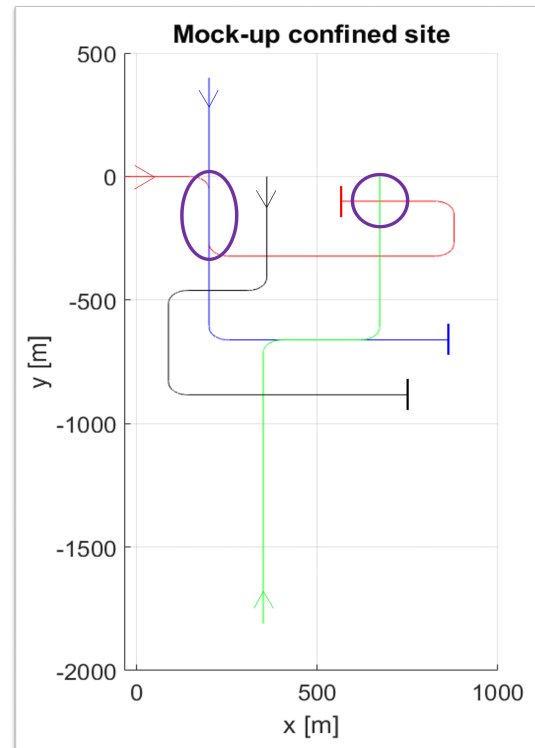

Fig. 3. Mock-up confined site area. The lines indicate the vehicle's path through the site with different color per vehicle. The arrow is the direction of movement for the vehicles and the solid line is indicates the path end. The CZs that are investigated in the simulation scenario are circled over in this figure.

collision occurs. In the coordinated case, the vehicles are capable of being inside the CZ at the same time and keeping at least a minimum designated gap between each other. Note that an intersection of the trajectories outside the CZ is not relevant as they are no longer on a shared road. By allowing the vehicles to be both in the CZ, and not blocking the whole zone for one vehicle, the throughput is increased meaning
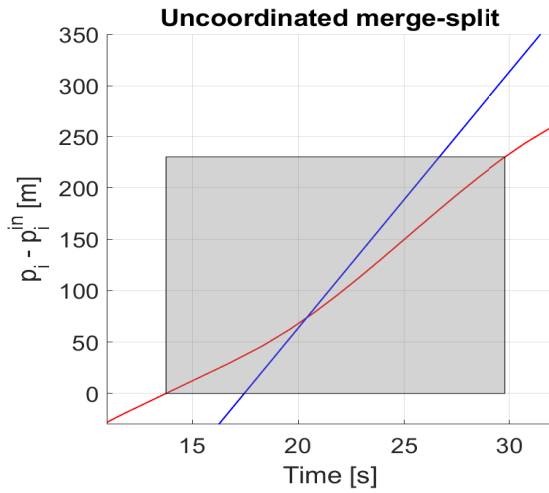
Fig. 4. Uncoordinated crossing for the merge-split CZ for the red and blue vehicle. The gray box is the occupancy time in the zone for the red vehicle. While it is allowed for the blue vehicle to enter the CZ whilst the red vehicle is in it, an intersection between the vehicles translates to a collision between the vehicles.
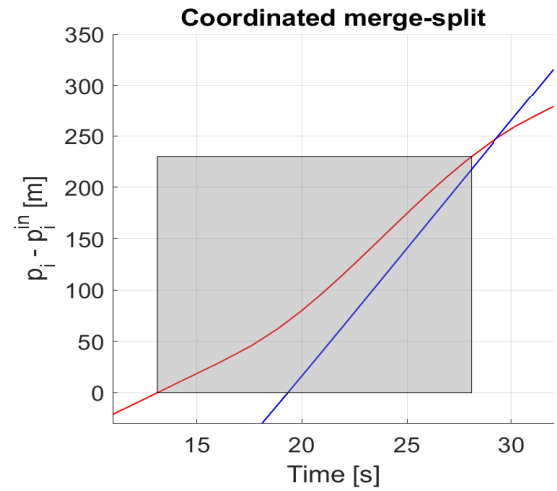

Fig. 5. Coordinated crossing for the merge-split CZ for the red and blue vehicle. The gray box is the occupancy time in the zone for the red vehicle. The blue vehicle enters the CZ whilst the red vehicle is in and keeps the desired time gap.
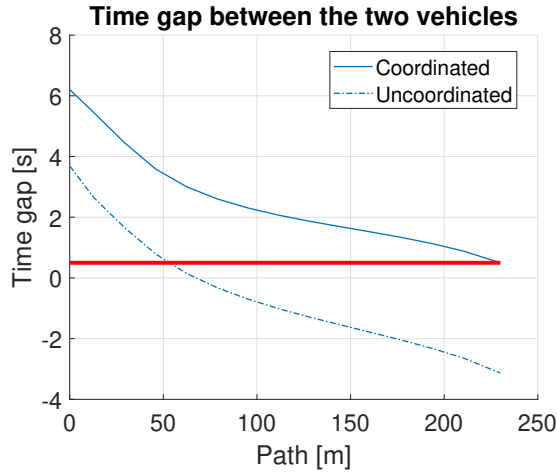

Fig. 6. The time gap between the red and blue vehicles in the merge-split CZ for the coordinated and uncoordinated case. The bold red line represents the minimum time gap that is desired to be kept between the vehicles. Crossing this border results in a collision between the vehicles.
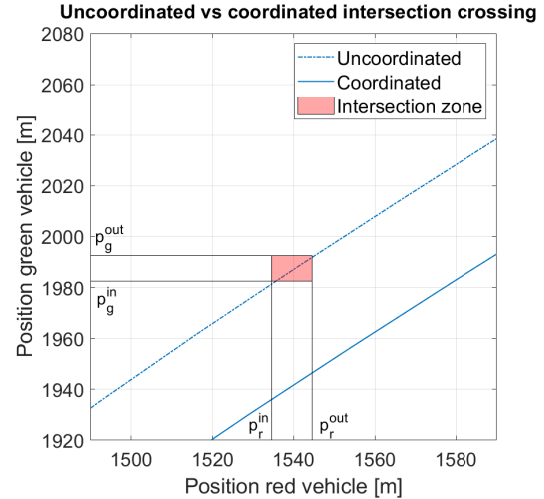

Fig. 7. Uncoordinated and coordinated crossing of the intersection zone. An intersection of the depicted trajectory with the intersection zone is equivalent to both vehicles being in the CZ at the same time.

that the vehicles could get to their end destination earlier. This could translate to increased efficiency or productivity of the site. To supplement the figures, Figure 6 depicts the time gap between the vehicles while they are in the CZ. In the figure, the moment where the blue vehicle "overruns" the red vehicle in the uncoordinated case can be noticed, as well as the fact, that the desired minimum gap is kept for the coordinated case.

In the scenarios where two (or more) vehicles intersect, it is desired to only have one vehicle in the zone. The reasoning for this decision is that the CZ in this case occupies a relatively small patch of road and blocking off the whole zone for one vehicle will not result in a major loss of throughput efficiency. Figure 7 depicts the uncoordinated and

coordinated vehicle trajectories along with the intersection zone. For this dependency, a collision is defined if the trajectory intersects the intersection zone, which is equivalent to the two vehicles being in the zone at the same time. It is noticeable that in the uncoordinated case the intersection zone is intersected, whilst the coordinated vehicles satisfy the collision constraints as they successfully avoid being in the CZ at the same time. Figure 8 depicts the vehicle's speed profile throughout the site for both the uncoordinated and coordinated results. As the objective function is to minimize the acceleration and jerk, it is noticeable that the speed profiles are smooth throughout the whole site and would not be challenging to follow in an application case. In the speed plots, the black lines represent the curvature constraints based
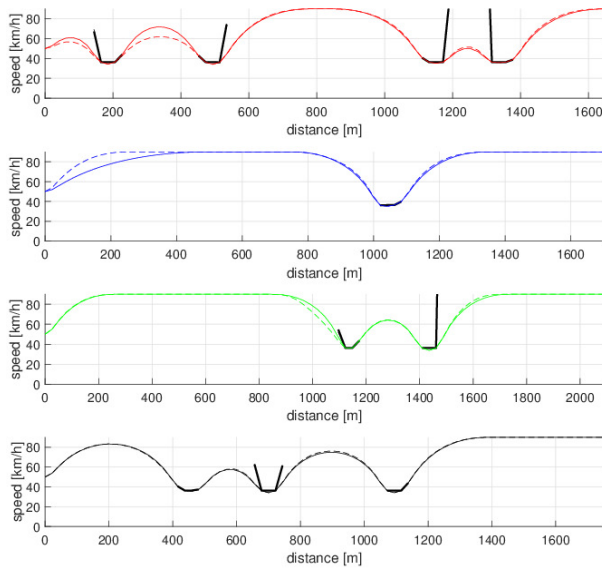
Fig. 8. Speed profiles for the uncoordinated and coordinated vehicles. The uncoordinated speed profiles are depicted with dashed lines, the coordinated speed profiles are illustrated with full lines and the curvature constraints based on (16) are depicted with bold black lines.

on (16). Since the algorithm considers all conflict zones between the vehicles from the beginning till the end of their path, undesired occupancy of the zone (i.e., collisions) can be avoided with minor changes of the speed profile over the whole route. In contrast, coordinating the vehicles in a cut-out around the CZ would require larger changes in speed in order to avoid conflicts. It is worth mentioning that in the uncoordinated case for the complete site, two more collisions occur. One in the other merge-split CZ between the blue and green vehicle and another in the intersection CZ of the blue and black vehicle. When the vehicles are coordinated, these collisions are avoided. For sake of brevity, we refrain from depicting these results as the critical behaviour is similar to the above-mentioned collision zones.

The simulation scenario is implemented in MATLAB on a 2.90GHz Intel Xeon computer with 32GB of RAM. The total solver computational time for solving the example is 1.815 seconds, of which the MIQP requires 0.08 seconds and the fixed-order NLP requires 0.709 seconds and the remainder is the required time to solve the individual vehicle problems. The computation time for the MIQP scales exponentially with the number of decision variables, while the fixed-order NLP scales cubicly with the number of vehicles.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented an optimal control based approach for coordination of automated vehicles for sites where multiple collision zones of different types could occur. The approach optimizes the trajectories of the vehicles for their whole path length, taking all collision zones into account. To compute the crossing order for the collision zones, the approach relies on an MIQP-based heuristic. Simulation results demonstrate the capability to coordinate the vehicles without collisions. In future work, we intend to investigate

improvement on the solver computation time, closed-loop behaviour as well as including higher fidelity vehicle models.

REFERENCES

[1] Bagloee, Saeed Asadi, Madjid Tavana, Mohsen Asadi, and Tracey Oliver. "Autonomous vehicles: challenges, opportunities, and future implications for transportation policies." Journal of modern transportation 24, no. 4 (2016): 284-303.

[2] Khayatian, Mohammad, Mohammadreza Mehrabian, Edward Andert, Rachel Dedinsky, Sarthake Choudhary, Yingyan Lou, and Aviral Shirvastava. "A survey on intersection management of connected autonomous vehicles." ACM Transactions on Cyber-Physical Systems 4, no. 4 (2020): 1-27.

[3] Hafner, Michael R., Drew Cunningham, Lorenzo Caminiti, and Domitilla Del Vecchio. "Cooperative collision avoidance at intersections: Algorithms and experiments." IEEE Transactions on Intelligent Transportation Systems 14, no. 3 (2013): 1162-1175.

[4] Kuyer, Lior, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. "Multiagent reinforcement learning for urban traffic control using co-ordination graphs." In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 656-671. Springer, Berlin, Heidelberg, 2008.

[5] Colombo, Alessandro, and Domitilla Del Vecchio. "Least restrictive supervisors for intersection collision avoidance: A scheduling approach." IEEE Transactions on Automatic Control 60, no. 6 (2014): 1515-1527.

[6] Hult, Robert, Mario Zanon, Sebastien Gros, and Paolo Falcone. "Optimal coordination of automated vehicles at intersections: Theory and experiments." IEEE Transactions on Control Systems Technology 27, no. 6 (2018): 2510-2525.

[7] Katriniok, Alexander, Peter Kleibaum, and Martina Joševski. "Distributed model predictive control for intersection automation using a parallelized optimization approach." IFAC-PapersOnLine 50, no. 1 (2017): 5940-5946.

[8] Hult, Robert, Mario Zanon, Sébastien Gros, and Paolo Falcone. "Optimal coordination of automated vehicles at intersections with turns." In 2019 18th European Control Conference (ECC), pp. 225-230. IEEE, 2019.

[9] Hult, Robert, Mario Zanon, Sebastien Gras, and Paolo Falcone. "An miqp-based heuristic for optimal coordination of vehicles at intersections." In 2018 IEEE Conference on Decision and Control (CDC), pp. 2783-2790. IEEE, 2018.

[10] Letter, Clark, and Lily Elefteriadou. "Efficient control of fully automated connected vehicles at freeway merge segments." Transportation Research Part C: Emerging Technologies 80 (2017): 190-205.

[11] Omidvar, Aschkan, Lily Elefteriadou, Mahmoud Pourmehrab, and Clark Letter. "Optimizing freeway merge operations under conventional and automated vehicle traffic." Journal of Transportation Engineering, Part A: Systems 146, no. 7 (2020): 04020059.

[12] Bui, Khac-Hoai Nam, and Jason J. Jung. "Cooperative game-theoretic approach to traffic flow optimization for multiple intersections." Computers & Electrical Engineering 71 (2018): 1012-1024.

[13] Richards, Arthur, and Jonathan How. "Mixed-integer programming for control." In Proceedings of the 2005, American Control Conference, 2005., pp. 2676-2683. IEEE, 2005.

[14] Fiacco, Anthony V. Introduction to sensitivity and stability analysis in nonlinear programming. Vol. 165. Academic press, 1983.

[15] Colombo, Alessandro, and Domitilla Del Vecchio. "Efficient algorithms for collision avoidance at intersections." In Proceedings of the 15th ACM international conference on Hybrid Systems: Computation and Control, pp. 145-154. 2012.

[16] Kobetski, Avenir, Domenico Spensieri, and Martin Fabian. "Scheduling algorithms for optimal robot cell coordination-a comparison." In 2006 IEEE International Conference on Automation Science and Engineering, pp. 381-386. IEEE, 2006.

[17] Błażewicz, Jacek, Wolfgang Domschke, and Erwin Pesch. "The job shop scheduling problem: Conventional and new solution techniques." European journal of operational research 93, no. 1 (1996): 1-33.

[18] Andersson, Joel AE, Joris Gillis, Greg Horn, James B. Rawlings, and Moritz Diehl. "CasADi: a software framework for nonlinear optimization and optimal control." Mathematical Programming Computation 11, no. 1 (2019): 1-36.

[19] Wächter, Andreas, and Lorenz T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming." Mathematical programming 106, no. 1 (2006): 25-57.