Quadratic approximation based heuristic for optimization-based coordination of automated vehicles in confined areas

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

(article starts on next page)

Quadratic approximation based heuristic for optimization-based coordination of automated vehicles in confined areas

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

Abstract— We investigate the problem of coordinating multiple automated vehicles (AVs) in confined areas. This problem can be formulated as an optimal control problem (OCP) where the motion of the AVs is optimized such that collisions are avoided in cross-intersections, merge crossings, and narrow roads. The problem is combinatorial and solving it to optimality is prohibitively difficult for all but trivial instances. For this reason, we propose a heuristic method to obtain approximate solutions. The heuristic comprises two stages: In the first stage, a Mixed Integer Quadratic Program (MIQP), similar in construction to the Quadratic Programming (QP) sub-problems in Sequential Quadratic Programming (SQP), is solved for the combinatorial part of the solution. In the second stage, the combinatorial part of the solution is held fixed, and the optimal state and control trajectories for the vehicles are obtained by solving a Nonlinear Program (NLP). The performance of the algorithm is demonstrated by a simulation of a non-trivial problem instance.

I. INTRODUCTION

The idea of fully automated vehicles (AV) is receiving substantial attention from both the public and scientific world as significant progress towards deploying automated vehicles has been made during the last decade. Unfortunately, many barriers between the current state-of-the-art and large-scale commercial application exits, especially for deployment of automated vehicles on public roads [1]. However, in confined areas, such as ports, mines, and logistic centers, some of the hard challenges of public road driving are absent. In particular, such areas are typically void of unpredictable non-controlled actors, which dramatically reduces safety concerns. Therefore, it is believed that confined areas present a good opportunity for early, large-scale deployment of automated vehicles, as part of larger commercial transport solutions for material flow.

One of the challenges in confined areas is the safe and efficient coordination of AVs in mutually exclusive (MUTEX) zones, such as intersections, work-stations (e.g. crushers, loading/unloading spots, etc.), narrow roads, and, in the case of electrified AVs, charging-stations. Adequate coordination can lead to improved energy-efficiency and considerable increases in productivity.

A. Related Work

Automating and coordinating intersections for fully automated vehicles is a frequently discussed control problem, see [2] for a comprehensive survey. The problem has been formally shown to be NP-hard [3], and such problems are in general difficult to solve. For this reason a number of methods have been proposed, leveraging results from, e.g., solutions based on hybrid system theory [4], reinforcement learning [5], scheduling [6], model predictive control (MPC) [7], [8] or direct optimal control (DOC) [9], [10].

In contrast to intersection scenarios often found in the literature, confined areas have a number of distinguishing features. For example, in the case of intersection coordination, an approach that is often considered is to have vehicles arriving at speed in a cutout around the intersection area [7], [8]. This is often motivated by practical considerations; neither the intent of the automated vehicles nor the state of the uncertain environment can be accurately predicted over long time-horizons. For confined areas, however, it is possible, and desirable, to plan the motion of each vehicle from the start of a transport mission to its end. Moreover, a number of works on public-road applications focus on distributed and decentralized schemes, sometimes with intermittent or corrupted communication [17]. For applications at confined sites, a central computational unit and good wireless coverage can often be assumed. Thus, for these use cases, we believe that a centralized approach that provides high level control actions is favorable. A low level controller that tracks the obtained optimal state and control trajectories will typically be also deployed in practice, however, it is not covered in this work.

B. Main Contribution and Outline

In this paper, we formulate the site-coordination problem as an optimal control problem, which after transcription results in a Mixed Integer Nonlinear Program, and propose a two-staged heuristic method for its approximate solution. In the first stage of the heuristic, an MIQP problem is solved to obtain the combinatorial part of the solution, i.e., the order in which the vehicles utilize the MUTEX-zones. In the second stage, the combinatorial part is fixed and a continuous NLP is solved for the optimal vehicle trajectories. The MIQP of the first stage is formed as an approximation of the original MINLP, in a manner similar to how the Quadratic Programming (QP) sub-problems are formed in Sequential Quadratic Programming (SQP) [11]. The idea of using SQP-like methods in approximate solution of MINLPs has been used in other works [13], [14], however, to the authors' best

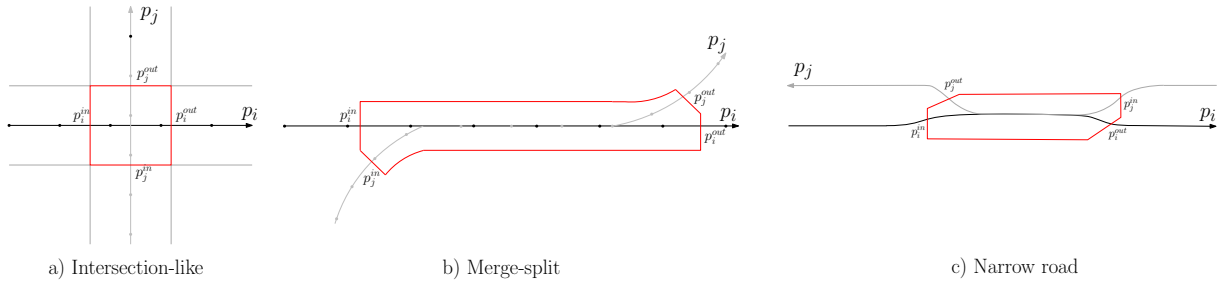**a) Intersection-like** **b) Merge-split** **c) Narrow road**

Fig. 1.   Types of conflict zones.

knowledge, it has not been adapted and applied to vehicle coordination problems. A structurally similar heuristic was presented in [9], where a scheduling problem is derived from the original MINLP, with the introduction of a number of approximations, and solved as an MIQP. While sharing the same two-staged structure, the method presented in this paper avoids some of the approximations and shortcomings of [9], without expanding the combinatorial solution space. In particular, the heuristic presented herein enables easy inclusion of rear-end collision constraints, which were previously neglected, and avoids the expensive computation of parametric sensitivities.

In addition to the cross-intersections, we consider merge-split and narrow road MUTEX zones. In the merge-split MUTEX zones, the vehicles first join in on a common patch of road which after some distance separate, and in the narrow-roads the vehicles that are coming from different directions join in on a common patch of road. Merge-split and narrow roads are often found in confined sites and occur due to the construction of the site. 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).

The remainder of the paper is organized as follows: Section II presents a formulation of 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

In this paper, we consider a fully confined area, meaning that non-controlled traffic participants such as pedestrians, manually operated vehicles, bicycles, etc., are absent. Furthermore, the confined road network consists of N_a fully automated vehicles with cross-intersection, path merges, path splits, and narrow roads. In addition, we assume that the paths of all vehicles, i.e., their routes through the road network are known, that overtakes are prohibited, and that no vehicle reverses.

A. Optimal Coordination Problem

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

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

$$\min_{x_i, u_i, \mathcal{O}^{\mathcal{I}}, \mathcal{O}^{\mathcal{M}}} \sum_{i=1}^{N_a} J_i \left(x_i, u_i \right) \tag{1a}$$

$$\text{s.t} \quad \text{initial states} \ x_{i,0} = \hat{x}_{i,0}, \forall i \tag{1b}$$

$$\text{system dynamics} \ \forall i \tag{1c}$$

$$\text{state and input constraints} \ \forall i \tag{1d}$$

$$\text{safety constraints} \ \forall i \tag{1e}$$

where N_a is the number of vehicles, $J_i(x_i, u_i)$ is the cost function, $\mathcal{O}^{\mathcal{I}}, \mathcal{O}^{\mathcal{M}}$ are the order in which the vehicles enter the MUTEX zones and will be formally stated in this section.

The problem is formulated in the spatial domain as it is beneficial to optimize the trajectories of the vehicles over their full paths. The rationale for using spatial dynamics is that the time it takes for the vehicle to traverse a path is not known *a-priori*. Thus, it is inappropriate to plan the vehicle's motion with time as the independent variable.

B. System dynamics and state and input constraints

The system dynamics for vehicle $i \in 1, ..., N_a$ in the spatial domain can be formed using that $\frac{dp_i}{dt} = v_i(t)$ and $dt = dp_i / v_i(t)$ and stated as

$$\frac{dt_i}{dp_i} = \frac{1}{v_i(p_i)} \tag{2}$$

$$\frac{dx_i}{dp_i} = \frac{1}{v_i(p_i)} f_i(p_i, x_i(p_i), u_i(p_i)) \tag{3}$$

$$0 \le h(p_i, x_i, u_i). \tag{4}$$

where the position p_i is the independent variable, the time t_i and x_i are the vehicle state variables, where $x_i \in \mathbb{R}^{n-1}$ collects the remaining vehicle states, and $u_i \in \mathbb{R}^m$ the control input, with $i \in \{1, \ldots, N_a\}$. Note that what the remaining state variable (x_i) are, depends on what model is used for the system dynamics. We assume that the functions f_i and h_i, that describe the vehicle system's dynamics and constraints, are smooth.

C. Safety constraints

The safety constraints should ensure a collision-free crossing of the *conflict zone* (CZ) that the vehicles encounter.

6157

A conflict zone 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 $t_i^{in} = t_i(p_i^{in})$ and $t_i^{out} = t_i(p_i^{out})$, respectively. In this paper, we consider three types of conflict zones, the "intersection-like", "narrow road" and the "merge-split", depicted in Figure 1. The narrow road CZ is when two vehicles that are coming from opposite directions have to share a common patch of road. In the intersection-like CZ and narrow road CZ, it is necessary 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. We let $\mathcal{I} = \{I_1, I_2, ..., I_{r_0}\}$ denote the set of all intersections and narrow roads in the confined site, with r_0 being the total number of intersection and narrow road CZs, and $\mathcal{Q}_r = \{q_{r,1}, q_{r,2}, ..., q_{r,l}\}$ denote the set of vehicles that cross an intersection or narrow road I_r. The order in which the vehicles cross the intersection I_r is denoted $\mathcal{O}_r^{\mathcal{I}} = (s_{r,1}, s_{r,2}, ..., s_{r,|\mathcal{Q}_r|})$, where $s_{r,1}, s_{r,2}, ...$ are vehicle indices and we let $\mathcal{O}^{\mathcal{I}} = \{\mathcal{O}_1^{\mathcal{I}}, ..., \mathcal{O}_r^{\mathcal{I}}\}$. A sufficient condition for collision avoidance for the r-th intersection or narrow road CZ can be formulated as

$$t_{s_{r,i}}(p_{s_{r,i}}^{out}) \leq t_{s_{r,i+1}}(p_{s_{r,i+1}}^{in}), i \in \mathbb{I}_{[1,|\mathcal{Q}_r|-1]}, \qquad (5)$$

where t is determined from (2).

In the merge-split CZ case, let $\mathcal{M} = \{M_1, M_2, ..., M_{w_0}\}$ denote a set of all merge-split zones, with w_0 being the total number of merge-split CZs in the site and $\mathcal{Z}_w = \{z_{w,1}, z_{w,2}, ..., z_{w,h}\}$ 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 merge-split CZ. In this case, the order in which the vehicles enter the zone is denoted as $\mathcal{O}_w^{\mathcal{M}} = (s_{w,1}, s_{w,2}, ..., s_{w,|\mathcal{Z}_w|})$, and we let $\mathcal{O}^{\mathcal{M}} = \{\mathcal{O}_1^{\mathcal{M}}, ..., \mathcal{O}_w^{\mathcal{M}}\}$. The collision avoidance requirement for the w-th merge-split CZ is described with the following constraints:

$$t_{s_{w,i}}(p_{s_{w,i}}^{in}) + \Delta t \leq t_{s_{w,i+1}}(p_{s_{w,i+1}}^{in} + c) \qquad (6a)$$

$$t_{s_{w,i},k_i} + \Delta t \leq t_{s_{w,i+1}}(p_{s_{w,i},k_i} - p_{s_{w,i}}^{in} + p_{s_{w,i+1}}^{in} + c),$$
$$k_{s_{w,i}}^{in} \leq k_i \leq k_{s_{w,i}}^{out} \qquad (6b)$$

$$t_{s_{w,i}}(p_{s_{w,i}}^{out}) + \Delta t \leq t_{s_{w,i+1}}(p_{s_{w,i+1}}^{out} + c), \qquad (6c)$$
$$i \in \mathbb{I}_{[1,|\mathcal{Z}_w|-1]}.$$

That is, while in the CZ, the vehicles must be separated by at least a time-period Δt and a distance c, 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. A practical reformulation of the collision constraints

A common way to handle constraints such as (5) and (6) is to introduce auxiliary binary variables and use the "big-M" technique [12]. For example, an equivalent representation to the constraint (5), with $b_{s_{r,i,i+1}} \in \{0, 1\}$, $i \in \mathbb{I}_{[1,|\mathcal{Q}_r|-1]}$

and a sufficiently large M is

$$t_{s_{r,i}}(p_{s_{r,i}}^{out}) - t_{s_{r,i+1}}(p_{s_{r,i+1}}^{in}) \leq b_{s_{r,i,i+1}}M, \qquad (7a)$$

$$t_{s_{r,i+1}}(p_{s_{r,i+1}}^{out}) - t_{s_{r,i}}(p_{s_{r,i}}^{in}) \leq (1 - b_{s_{r,i,i+1}})M. \qquad (7b)$$

In the case where $b_{s_{r,i,i+1}} = 0$, the vehicle $i + 1$ is constrained to cross the CZ after the vehicle i, with the opposite being true if $b_{s_{r,i,i+1}} = 1$. We collect all integer variables for all CZs in $b \in \mathbb{Z}_2^{r_o+w_0}$.

E. Discretization

The independent variable is discretized as $p_i = (p_{i,0}, ..., p_{i,N_i})$, 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 (2), (3) are (numerically) integrated on this grid, giving the "discretized" state transition relation

$$\begin{bmatrix} t_{i,k+1} \\ x_{i,k+1} \end{bmatrix} = F(x_{i,k}, u_{i,k}, p_{i,k}, p_{i,k+1}) \qquad (8)$$

where F denotes the integration of (2), (3) from $p_{i,k}$ to $p_{i,k+1}$.

III. METHOD

The optimal coordination problem, Problem 1, can be stated as a Mixed Integer Nonlinear Program (MINLP), where the crossing order correspond to the combinatorial ("integer part") and the state and control trajectories correspond to the "NLP part". In essence, we can state Problem 1 as

$$\min_{\mathcal{W}, b} \quad J(\mathcal{W}) \qquad (9a)$$

$$\text{s.t.} \quad g(\mathcal{W}) = 0 \qquad (9b)$$

$$h(\mathcal{W}) \leq 0 \qquad (9c)$$

$$c(\mathcal{W}, b) \leq 0, \qquad (9d)$$

where $\mathcal{W} = \{\mathcal{X}, \mathcal{U}\}$, $J(\mathcal{W}) = \sum_{i=1}^{N_a} J_i(w_i)$, $g(\mathcal{W}), h(\mathcal{W})$ gather all equality and inequality constraints, and $c(\mathcal{W}, b) = c_w(\mathcal{W}) + Cb$ are the integer constraints for the combinatorial part of the problem with C being a matrix that captures the influence of the integer variables.

Since finding a solution to MINLP problems is known to be difficult, it is common to obtain approximate solutions with heuristics. One heuristic approach used in e.g. [9], is to use a two-staged procedure, where the integer part of the solution first is found with a heuristic, and the continuous part is found by solving the NLP obtained by eliminating all other integer options from the MINLP. In this paper, we follow this decomposition idea and propose an alternative heuristic for the integer part than [9]. In the confined area coordination context, the integer part of the solution corresponds to the crossing order of the vehicles at the MUTEX zones.

A. Crossing order heuristic

The crossing order heuristic is based on solving an MIQP that is assembled from a quadratic approximation of (9). The quadratic approximation is formed similarly to how the QP sub-problems are formed in SQP methods [11]. In essence, (9) can be reformulated as:

$$\min_{\Delta\mathcal{W},b} \frac{1}{2} \begin{bmatrix} \Delta\mathcal{W} \\ b \end{bmatrix}^T \mathbf{H}(\mathcal{W},\lambda,\mu) \begin{bmatrix} \Delta\mathcal{W} \\ b \end{bmatrix} +$$

$$\nabla_{\mathcal{W}} J(\mathcal{W})^T \begin{bmatrix} \Delta\mathcal{W} \\ b \end{bmatrix} + J(\mathcal{W}^{**}) \tag{10a}$$

$$\text{s.t.} \quad g(\mathcal{W}^{**}) + \nabla_{\mathcal{W}} g(\mathcal{W}^{**})^T \begin{bmatrix} \Delta\mathcal{W} \\ b \end{bmatrix} = 0 \tag{10b}$$

$$h(\mathcal{W}^{**}) + \nabla_{\mathcal{W}} h(\mathcal{W}^{**})^T \begin{bmatrix} \Delta\mathcal{W} \\ b \end{bmatrix} \leq 0 \tag{10c}$$

$$c_w(\mathcal{W}^{**}) + \nabla_{\mathcal{W}} c_w(\mathcal{W}^{**})^T \begin{bmatrix} \Delta\mathcal{W} \\ b \end{bmatrix} + Cb \leq 0, \tag{10d}$$

where $\mathbf{H}(\mathcal{W},\lambda,\mu) = \text{blkdiag}(\{H_i\}_{i=1}^{N_a}, \mathbf{0}_{r_0+w_0,r_0+w_0})$ is a block diagonal matrix with positive definite $H_i(w_i,\lambda_i,\mu_i) = \nabla^2_{w_i}\mathcal{L}(w_i,\lambda_i,\mu_i) = \nabla^2_{w_i}J_i(w_i) - \nabla^2_{w_i}\lambda_i^T g(w_i) - \nabla^2_{w_i}\mu_i^T h(w_i)$, where λ_i,μ_i are the dual variables and $\mathbf{0}_{r_0+w_0,r_0+w_0}$ zeros of appropriate size for the integer variables, and $\Delta\mathcal{W} = \mathcal{W} - \mathcal{W}^{**}$, with a solution guess $\mathcal{W}^{**}$.

For the heuristic used in this paper, we make the simplification that the dual variables (λ_i,μ_i) are equal to zero. This results in that H_i only includes the second order expansion of the cost function, i.e., $H_i(w_i) = \nabla^2_{w_i}J_i(w_i)$. The solution guess $\mathcal{W}^{**}$ can be obtained, for example, by solving the optimization problem (1) without safety constraints (1e), or a forward simulation of the vehicles with, for example, a simple feedback controller.

The MIQP problem (10) can be compactly written as

$$\min_{\mathcal{W},b} \frac{1}{2} \begin{bmatrix} \mathcal{W} \\ b \end{bmatrix}^T \mathbf{H} \begin{bmatrix} \mathcal{W} \\ b \end{bmatrix} + \mathbf{J}^T \begin{bmatrix} \mathcal{W} \\ b \end{bmatrix} + \alpha \tag{11a}$$

$$\text{s.t.} \quad A_{\text{eq}} \begin{bmatrix} \mathcal{W} \\ b \end{bmatrix} = b_{\text{eq}} \tag{11b}$$

$$A_{\text{ineq}} \begin{bmatrix} \mathcal{W} \\ b \end{bmatrix} \leq b_{\text{ineq}}, \tag{11c}$$

where $\mathbf{J}$ now contains all the first order terms, α contains the linear terms and where the constraints (10b)-(10d) are grouped into equality constraints $A_{\text{eq}}, b_{\text{eq}}$ and inequality constraints $A_{\text{ineq}}, b_{\text{ineq}}$, respectively. The solution to the MIQP problem provides the "approximately optimal" crossing orders $\hat{\mathcal{O}}^{\mathcal{I}}, \hat{\mathcal{O}}^{\mathcal{M}}$ that is obtained from the values of the integer variables b.

B. Fixed-order NLP

Removing all other integer solutions than that found by the heuristic, (1) is reduced to an NLP. Obtaining the optimal state and control trajectories is thus found through solving the *fixed-order coordination problem*

$$\min_{x_{i,k},u_{i,k}} \sum_{i=1}^{N_a} J_i(x_{i,k}, u_{i,k}) \tag{12a}$$

$$\text{s.t} \quad (1b)-(1e), \; \forall i, \forall k \tag{12b}$$

$$\mathcal{O}^{\mathcal{I}} = \hat{\mathcal{O}}^{\mathcal{I}}, \; \mathcal{O}^{\mathcal{M}} = \hat{\mathcal{O}}^{\mathcal{M}} \tag{12c}$$

The two stage approximation approach is summarized in Algorithm 1.

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 a solution guess w_i^{**} by, e.g., solving NLP (1) w/o safety constraints (5).
2: Calculate and form the approximation terms $\mathbf{H}, \mathbf{J}, \alpha$.
3: Solve the MIQP (11) to get "approximately optimal" crossing orders $\hat{\mathcal{O}}^{\mathcal{I}}, \hat{\mathcal{O}}^{\mathcal{M}}$.
4: Solve the fixed-order NLP (12) using $\hat{\mathcal{O}}^{\mathcal{I}}, \hat{\mathcal{O}}^{\mathcal{M}}$ to 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:

$$\begin{bmatrix} \frac{dt_i}{dp_i} \\ \frac{dv_i}{dp_i} \\ \frac{da_i}{dp_i} \end{bmatrix} = \begin{bmatrix} \frac{1}{v_i} \\ \frac{a_i}{v_i} \\ \frac{u_i}{v_i} \end{bmatrix}, \tag{13}$$

where a_i is the acceleration and u_i is the jerk j_i. The problem is transcribed using multiple shooting (with N shooting points) and an Explicit Runge-Kutta-4 (ERK4) integrator.

The state 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 vehicles are expected to manoeuvre on curved roads, thus, it is necessary to limit the lateral forces to avoid vehicle stability problems, like roll over. 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],

$$\left(\frac{a_{i,k}}{a_{i,lon}}\right)^2 + \left(\frac{\kappa_i(s_{i,k})v_{i,k}}{\overline{a}_{i,lat}}\right)^2 \leq 1, \tag{14}$$

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 a trade-off between minimization of the total travel time and the

6159

squares of the longitudinal acceleration and longitudinal jerk:

$$J(x_i(p_i), u_i(p_i)) = \qquad (15)$$
$$\int_{p_{i,0}}^{p_{i,N_i}} \left(P_i a_i(p_i)^2 + Q_i j_i(p_i)^2 \right) \frac{1}{v_i(p_i)} \mathrm{d}p_i + R_i t_i(p_{i,N_i}),$$

where P_i, Q_i, R_i are the appropriate weights. The inclusion of the acceleration and jerk in the objective can be interpreted as a drivability measure. The objective function is integrated with Forward Euler, leading to the "discretized" objective function

$$J_i(x_{i,k}, u_{i,k}) = \qquad (16)$$
$$\sum_{k=1}^{N} \left(\left(P_i a_{i,k}^2 + Q_i j_{i,k}^2 \right) \frac{\Delta p_{i,k}}{v_{i,k}} \right) + R_i t_{i,N},$$

with $\Delta p_{i,k} = p_{i,k+1} - p_{i,k}$. We consider a confined site with layout shown in Figure 2. There are in total ten vehicles, two merge-split CZs, two narrow road CZs and sixteen intersection CZs. Every vehicle starts from an initial velocity of 50 km/h and vehicles 5 and 10 start from a nonzero initial time to ensure that a collision occurs if no coordinating action is taken. The respective initial times for those vehicles are $t_{0,5} = 56$ and $t_{0,10} = 11.3$ seconds. The chosen velocity and acceleration bounds are $\underline{v}_i = 3.6$ km/h, $\overline{v}_i = 90$ km/h, $\overline{a}_{i,lon} = 4$ m/s², $\overline{a}_{i,lat} = 2$ m/s². The weight coefficients for the objective are: $P_i = 1$, $Q_i = 1$, $R_i = 10$ and the number of shooting points is $N = 100$ for each vehicle. The intersection CZ is created with 5 meter margin ahead of and behind the collision point, where as in the merge-split and narrow road CZ the margin is 15 meters for both the entry and exit collision point. For the merge-split CZ it is desirable to keep at least 0.5s margin between the vehicles, i.e. $\Delta t = 0.5$ in (6).

We utilize the CasADi [15] toolkit and IPOPT [16] to formulate and solve the optimization problem (12) and use Gurobi for the MIQP (11).

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, one intersection zone, and one narrow road, all circled over in Figure 2. The uncoordinated results are obtained as the individual optimum of each vehicle, i.e., the speed profiles are obtained by solving the optimization problem without any MUTEX zone constraints.

Figure 3 and Figure 4 show the position vs. time trajectories of the two vehicles at the merge-split CZ in the uncoordinated and coordinated case, respectively. The trajectories are shifted such that position zero is the entry position of the CZ for both vehicles. The gray rectangle depicts the CZ and time spent inside the CZ for the 1ˢᵗ vehicle. The interpretation of constraints (6) is that a collision occurs if the trajectories intersect while both vehicles are in the CZ. Figure 3 shows that without coordination, the second vehicle would run into the first just after $t = 20$ s (the trajectories intersect). In the coordinated case the vehicles are instead controlled to avoid
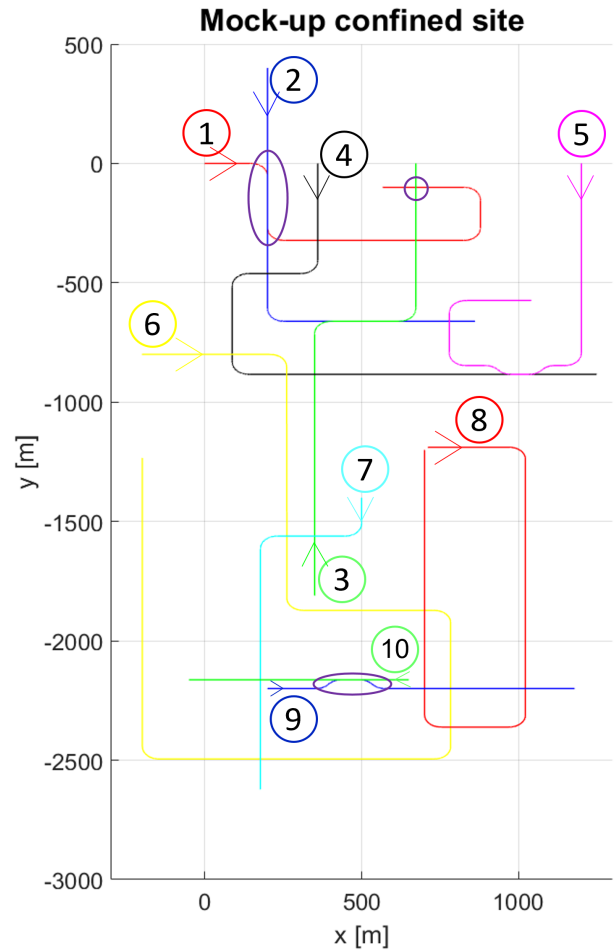

Fig. 2. Mock-up confined site area. The CZs that are investigated in the simulation scenario are circled over in this figure.

a collision (the trajectories do not intersect) and keep at least the specified distance at all times. Note that an intersection between the trajectories outside the CZ is not relevant as they are no longer on a shared road. By allowing both vehicles to be in the CZ, and not blocking the whole zone for one vehicle, the throughput is increased.

In the scenarios where the paths of two (or more) vehicles intersect, it is necessary to only have one vehicle in the zone at any given time. 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. Figure 5 shows the position-vs-position trajectory of the 1ˢᵗ and 3ʳᵈ vehicle in the uncoordinated and coordinated case, respectively. The red area represents the CZ, but in this case, a collision occurs when two vehicles are inside the CZ at the same time. This is equivalent to all configurations where the trajectory is inside the gray area. As the figures show, a collision occurs in the uncoordinated case (the trajectory passes through the red area), the coordinated vehicles satisfy the collision constraints and avoid being in the CZ at the same time.

For the narrow road collision zones, the collision avoidance is defined in the same way as in the intersection zone

Uncoordinated merge-split
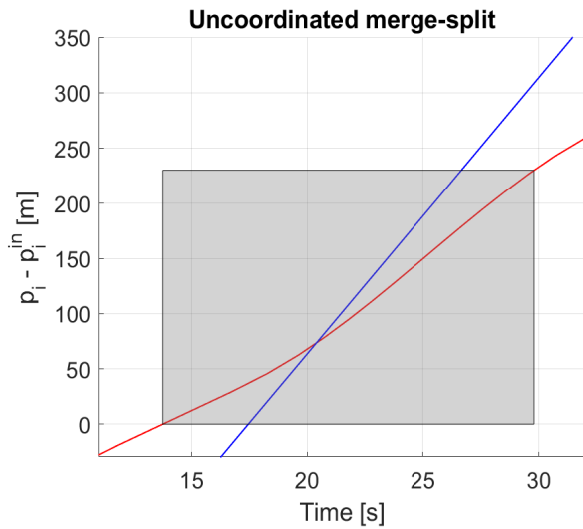
Coordinated merge-split

Fig. 3. Uncoordinated crossing for the merge-split CZ for the 1st and 2nd vehicle. The gray box is the occupancy time in the zone for the 1st vehicle. While it is allowed for the 2nd vehicle to enter the CZ whilst the 1st vehicle is in it, an intersection between the vehicles translates to a collision between the vehicles.
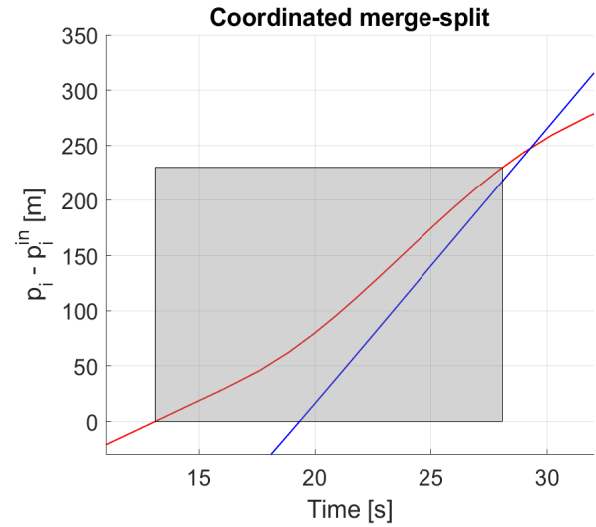
Fig. 4. Coordinated crossing for the merge-split CZ for the 1st and 2nd vehicle. The gray box is the occupancy time in the zone for the 1st vehicle. The 2nd vehicle enters the CZ whilst the 1st vehicle is in and keeps the desired time gap.

Uncoordinated vs coordinated intersection crossing
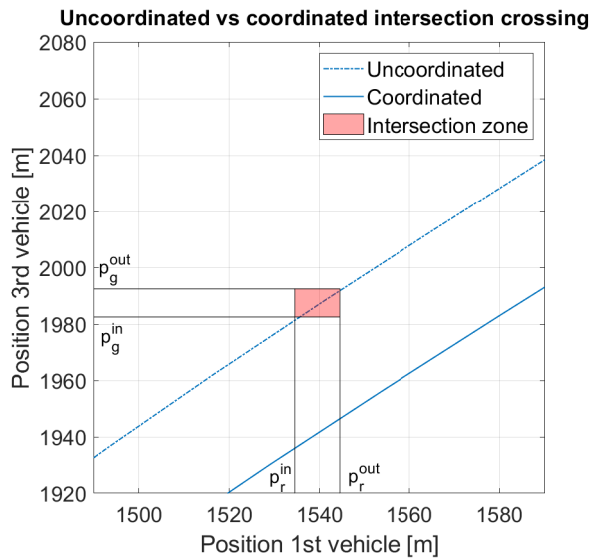
Uncoordianted vs coordianted narrow road

Fig. 5. 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.
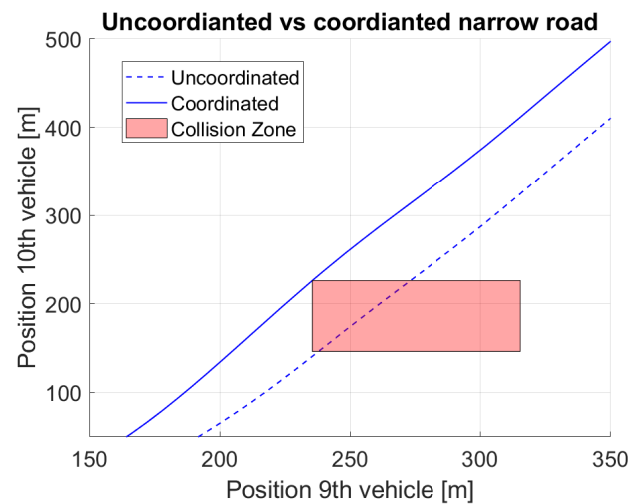
Fig. 6. Uncoordinated and coordinated crossing of a narrow road collision zone. An intersection of the depicted trajectory with the CZ is equivalent to both vehicles being in the CZ at the same time.

since the road conditions allow for only one vehicle to occupy the zone at a time. Figure 6 shows that both vehicles are inside the zone at the same time in the uncoordinated case (the trajectory passes through the red area). This is avoided in the coordinated case, although with a very small margin. The interpretation of the trajectory touching the corner of the red area is that one vehicle enters the narrow road just as the other exits.

Figure 7 shows the speed profiles of the vehicles for the uncoordinated and coordinated case, where the black lines illustrate implicit bounds on speed through the acceleration

constraint (14). As the algorithm is aware of all CZs the vehicles encounters, it is able to avoid collisions with small speed changes well before the vehicle arrives at the CZ. With the objective function aiming 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 actual application. It is worth to highlighting that several collisions occur in the uncoordinated case. When the vehicles are coordinated, all collisions are avoided. For sake of brevity, we refrain from depicting these results as the critical behaviour is similar to the cases discussed in detail above.

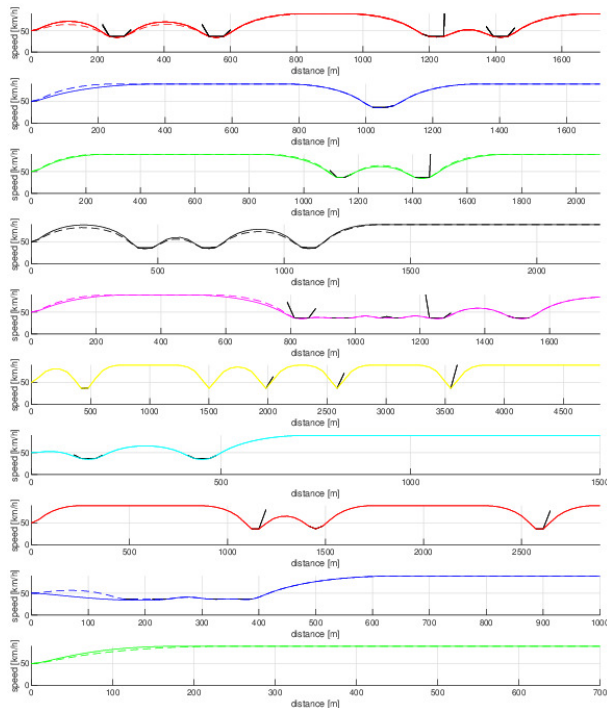The simulation scenario is implemented in MATLAB on a

Fig. 7. Speed profiles for the uncoordinated (dashed lines) and coordinated (solid lines) vehicles.

2.90GHz Intel Xeon computer with 32GB of RAM. The total solve computational time for the example is 1.973 seconds. To be precise, the MIQP requires 0.14 seconds and the main coordination optimal control problem requires 1.833 seconds.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a heuristic that finds approximate solutions to the optimal coordination problem of automated vehicles at confined sites with multiple collision zones of different types. The approach optimizes the trajectories of the vehicles over their entire path while taking all collision zones into account. In future work, we intend to investigate improvement on the NLP problem computation time, closed-loop behaviour, and coupling the approach with a safety supervisor to guarantee satisfaction of the safety constraints.

REFERENCES

[1] Fagnant, Daniel J., and Kara Kockelman. "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations." Transportation Research Part A: Policy and Practice 77 (2015): 167-181.
[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] 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.
[4] 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.
[5] Kuyer, Lior, Shimon Whiteson, Bram Bakker, and Nikos Vlassis. "Multiagent reinforcement learning for urban traffic control using coordination graphs." In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 656-671. Springer, Berlin, Heidelberg, 2008.
[6] 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.
[7] 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.
[8] 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.
[9] 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.
[10] 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.
[11] Nocedal, Jorge, and Stephen Wright. Numerical optimization. Springer Science & Business Media, 2006.
[12] Richards, Arthur, and Jonathan How. "Mixed-integer programming for control." In Proceedings of the 2005, American Control Conference, 2005., pp. 2676-2683. IEEE, 2005.
[13] Quirynen, Rien, and Stefano Di Cairano. "Sequential Quadratic Programming Algorithm for Real-Time Mixed-Integer Nonlinear MPC.
[14] Exler, Oliver, Thomas Lehmann, and Klaus Schittkowski. "A comparative study of SQP-type algorithms for nonlinear and nonconvex mixed-integer optimization." Mathematical Programming Computation 4, no. 4 (2012): 383-412.
[15] 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.
[16] 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.
[17] Zanon, Mario, Robert Hult, Sébastien Gros, and Paolo Falcone. "Experimental validation of distributed optimal vehicle coordination." In 2018 European Control Conference (ECC), pp. 1511-1516. IEEE, 2018.