An Interior Point Algorithm for Optimal Coordination of Automated Vehicles at Intersections

Robert Hult, Mario Zanon, Sébastien Gros, Paolo Falcone

***Abstract*—In this paper, we consider the optimal coordination of automated vehicles at intersections under fixed crossing-orders. We state the problem as a Direct Optimal Control problem, and propose a line-search Primal-Dual Interior Point algorithm with which it can be solved. We show that the problem structure is such that most computations required to construct the search-direction and step-size can be performed in parallel on-board the vehicles. This is realized through the Schur-complement of blocks in the KKT-matrix in two steps and a merit-function with separable components. We analyze the communication requirements of the algorithm, and propose a conservative approximation scheme which can reduce the data exchange. We demonstrate that in hard but realistic scenarios, reductions of almost 99% are achieved, at the expense of less than 1% sub-optimality.**

I. Introduction

The last decade has sen rapid development of technologies for Automated Vehicles (AV). Simultaneously, several standards have been adopted for vehicle-to-vehicle communication, and use of next generation cellular communication in automotive applications is under investigation. Due to this, the interest for applications where the AVs share information and cooperate is increasing, and it is commonly held that such Cooperative Automated Vehicles (CAV) would have positive effects on the traffic system.

One such application is the coordination of CAVs at intersections. The idea is to let the CAVs jointly decide how to cross the intersection safely and efficiently, rather than relying on traffic-lights, road signs and traffic rules. In this paper, we study a numerical algorithm for the optimal control formulation of such scenarios.

The literature on algorithms for coordination of CAVs at intersections was surveyed in [1]–[3], and even though most work is recent, the number of publications is growing rapidly. While a substantial part of the literature rely completely on heuristics [4]–[6], a number of contributions that employ Optimal Control (OC) tools [7]–[15] have been proposed recently. However, most OC-based algorithms rely on heuristics to some extent. This is largely due to the difficult combinatorial nature of the problem, which stems from the need to determine the order in which the vehicles cross the intersection. In a number of contributions the problem is solved in two stages where 1) the crossing order is found through a heuristic (typically variations of “first-come-first-served”) and 2) the control commands are found using OC-tools [11]–[14]. The algorithm in this paper is intended for such applications, and deals with the problem of finding the optimal control commands for a fixed crossing order.

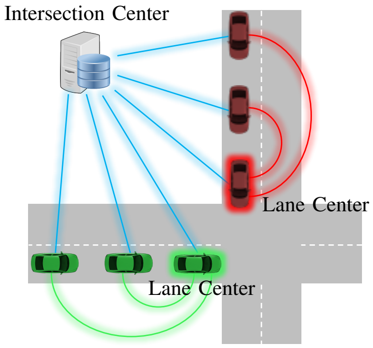


Fig. 1: Illustration of distribution structure

We have studied the intersection problem in earlier work. In particular, we introduced a Sequential Quadratic Programming (SQP) algorithm based on a primal decomposition of the fixed-order coordination problem in [16], where most computations are parallelized and performed on-board the vehicles. We considered a receding horizon application of the SQP-based algorithm in [17], where we also presented experimental results. We considered the extension to non-linear motion models and economic objective functions in [18] and to scenarios where vehicles turns inside the intersection in [19]. In [20], we proposed an OC-based heuristic for crossing order selection, and compared the performance of to traffic-lights and other coordination algorithms in [21].

However, the algorithm in [16] did not account for rear-end collisions between vehicles on the same lane, and required the solution of a non-smooth Nonlinear Program (NLP).

In this paper, we solve the fixed-order coordination problem with a Primal-Dual Interior Point (PDIP) algorithm which resolves both these issues. The algorithm relies on distributed computation of both the search-direction and step-size. As in [16], [17], this approach is partly centralized, and relies on the presence of central units for a smaller part of the computations. In particular, the algorithm uses one intersection-wide central unit and one central unit for each lane, with communication flows as illustrated in Fig. 1.

We use the Schur-complement of sub-matrices in the KKT-matrix to distribute the solution of the KKT-system for the search-direction, and a merit-function with separately computable components to distributed the step-size selection. This enables parallelization so that most computations can be performed on-board the vehicles, a smaller part on the lane-centers and the smallest part on the intersection center.

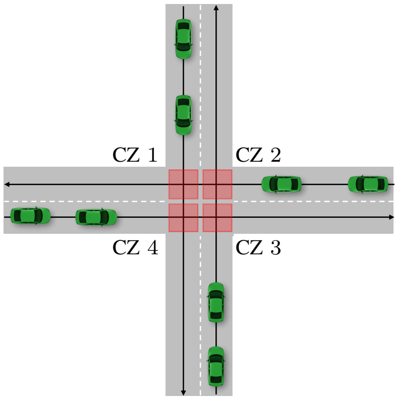


Fig. 2: Illustration of the scenarios considered, Assumption 2 (black lines) and the Conflict Zones (red boxes).

A. Contributions

The contributions in this paper are 1) The application of a distributed PDIP algorithm to the intersection problem, 2) the analysis of the communication requirements in a practical setting 3) a method with which the communication requirements can be reduced at the cost of sub-optimality.

We stress that while similar distribution schemes can be found in the literature (see e.g. [22], [23] and the references therein), the application to the intersection problem is novel.

B. Outline

The remainder of the Paper is organized as follows. In Section II we model and state the intersection problem using an Optimal Control formalism. In Section III we review Primal-Dual Interior Point methods and outline how the computations involved can be distributed. In Section IV, we construct the KKT-system (11) for a variable order which makes the problem structure apparent. In Section V we show how the solution of the KKT system can be distributed with computation at the vehicle, lane and intersection centers. In Section VI we show how to select the step-size in a distributed fashion. In Section VII we state a rudimentary practical algorithm and provide a numerical example. In Section VIII we analyze the communication requirements and propose an approximate representation of the RECA constraints, using which the data sent per iterate can be reduced. Finally, the paper is concluded in Section IX

II. Optimal Coordination at Intersections

We consider intersection scenarios as shown in Fig. 2, where N vehicles approach a four-way intersection, and make the following assumptions:

Assumption 1 (Full automation and cooperation)**.** *There are no non-cooperative entities present in the scenario.*

Assumption 2 (Vehicles on rails)**.** *The vehicles do not change lanes and move along fixed and known paths along the road. All vehicles on the same lane uses the same path.*

Assumption 1 means that we do not consider scenarios with, e.g. legacy vehicles, pedestrians or bicyclists. The assumption is restrictive and limits the applicability to traffic scenarios in a distant future. Assumption 2, however, is not restrictive, since vehicles at intersections in general follow the centerline of the lane that they are on. Both assumptions are standard in the literature (see e.g. [4]–[6], [10]).

A. Motion Models

Assumption 2 enables simple motion models that describe the one-dimensional movement of vehicles along their paths. We consider constrained ODE motion-models such that

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

$$0 \geq h_i(\mathrm{x}_i(t), \mathrm{u}_i(t)), \tag{1b}$$

where i is the vehicle index, $\mathrm{x}_i(t) \in \mathbb{R}^{n_i}$ and $\mathrm{u}_i(t) \in \mathbb{R}^{m_i}$ are the vehicle state and control. In particular, $\mathrm{x}_i(t) = (\mathrm{p}_i(t), \mathrm{v}_i(t), \tilde{\mathrm{x}}_i(t))$, where $\mathrm{p}_i(t)$ is the position of the vehicle's geometrical center on its path, $\mathrm{v}_i(t)$ is the velocity along the path and, if applicable, $\tilde{\mathrm{x}}_i(t)$ collects all remaining states (e.g. acceleration and/or internal states of the power-train). Both $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \mapsto \mathbb{R}^{n_i}$ and $h_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \mapsto \mathbb{R}^{q_i}$ are continuously differentiable.

B. Side Collision Avoidance (SICA)

Side collisions can only occur between vehicles on different lanes, when these are inside an area around the points where the vehicles' paths intersect. We denote these areas *Conflict Zones* (CZ), and note that more than one vehicle pair (i, j) can have potential collisions at a particular CZ. Collision avoidance consequently amounts to ensuring that vehicles on different lanes occupy each CZ in a mutually exclusive fashion. We enforce this condition using auxiliary variables that describe the time of entry ($t^{\text{in}}_{r,i}$) and departure ($t^{\text{out}}_{r,i}$) of the r:th CZ, implicitly defined through

$$\mathrm{p}_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}, \quad \text{and} \quad \mathrm{p}_i(t^{\text{out}}_{i,r}) = p^{\text{out}}_{r,i}.^{1} \tag{2}$$

Here, $p^{\text{in}}_{r,i}$ and $p^{\text{out}}_{r,i}$ are defined as shown in Fig. 3, using the vehicle length L_i and width W_i. A condition for Side Collision Avoidance (SICA) is then

$$t^{\text{out}}_{r,i} \leq t^{\text{in}}_{r,j}, \quad (i, j, r) \in \mathcal{S}. \tag{3}$$

The set $\mathcal{S}$ collects all vehicle pairs (i, j) and CZ where side collisions can occur, and encodes the crossing order (which vehicle crosses the intersection first).

C. Rear-End Collision Avoidance (RECA)

Due to Assumption 2, rear-end collisions can only occur between two adjacent vehicles on the same path. We collect all vehicle pairs (i, j) such that i is immediately behind j on

[1] In the event that $\mathrm{v}_i(t^{\text{in}}_{i,r}) = 0$, $t^{\text{in}}_{i,r}$ is not uniquely defined by $\mathrm{p}_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}$. A practical remedy is to instead use the slightly more complex definition $t^{\text{in}}_{r,i} = \min t$ s.t. $\mathrm{p}_i(t^{\text{in}}_{i,r}) = p^{\text{in}}_{r,i}$. Since $\dot{\mathrm{p}}(t^{\text{in}}_{i,r}) = 0$ rarely will be encountered in practice, this is avoided for ease of presentation.

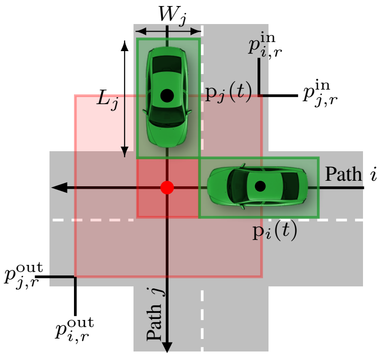


Fig. 3: Illustration of the elements used in the side collision avoidance conditions.

the same lane in $\mathcal{C}_R$, and state the necessary condition for Rear-End Collision Avoidance (RECA)

$$\mathrm{p}_i(t) + \delta_{ij} \leq \mathrm{p}_j(t), \quad (i,j) \in \mathcal{C}_R \tag{4}$$

where $\delta_{ij} = L_i/2 + L_j/2$.

D. Optimal Control Formulation

We employ a direct optimal control formulation of the coordination problem, assuming a piece-wise constant parametrization of the inputs $\mathrm{u}_i(t)$. That is, $\mathrm{u}_i(t) = u_{i,k}$, $u_{i,k} \in \mathbb{R}^{m_i}$, $t \in [t_k, t_{k+1})$, $k = 1, \dots, K-1$,$K \in \mathbb{N}$ where $t_k = k\Delta t$ and Δt is the time-discretization size. We introduce the vectors $x_i = (x_{i,0}, \dots, x_{i,K})$, $x_{i,k} \in \mathbb{R}^{n_i}$ and consider a multiple shooting discretization of the dynamics (1a), enforcing

$$x_{i,k+1} = x_i(t_{k+1}) = F_i(x_{i,k}, u_{i,k}, \Delta t), \tag{5}$$

and $x_{i,0} = \hat{x}_{i,0}$, where $\hat{x}_{i,0}$ is the initial state of vehicle i. Here, $F_i(x_{i,k}, u_{i,k}, \Delta t)$, $F_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \times \mathbb{R} \mapsto \mathbb{R}^{n_i}$ denotes the solution to (1a) at $t = t_k + \Delta t$, when $\mathrm{x}_i(t_k) = x_{i,k}$ and $\mathrm{u}_i(t) = u_{i,k}$. The state and control trajectories $\mathrm{x}_i(t)$ and $\mathrm{u}_i(t)$ are thereby described by x_i and $u_i = (u_{i,0}, \dots, u_{i,K_1})$, which we collect as $w_i = (x_i, u_i)$. Moreover, we express the position $\mathrm{p}_i(t)$ at time t as a function of w_i

$$p_i(t, w_i) = F_{i,p}(x_{i,k}, u_{i,k}, t - t_k),\ k = \lfloor t/\Delta t \rfloor, \tag{6}$$

where $F_{i,p} : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \times \mathbb{R} \mapsto \mathbb{R}$ denotes the position component of F_i. Consequently, all $t^{\mathrm{in}}_{i,r}$, $t^{\mathrm{in}}_{i,r}$ are well defined, continuous functions of w_i through (2) when (5) holds. Note that while (6) describes the position at an arbitrary time t, the position at t_k is a part of $x_{i,k}$, which we denote $p_{i,k}$. Finally, we only enforce the inequality constraints (1b) at times t_k.

With the index set $\mathcal{I}_a = \{0, \dots, a\}$, for an integer $a > 0$, the problem of optimal intersection coordination is

$$\begin{aligned}
\min_{w,T} \quad & \sum_{i=1}^{N} J_i(w_i) && && \text{(7a)}\\
\text{s.t.} \quad & x_{i,k} = \hat{x}_{i,0}, && i \in \mathcal{N}, && \text{(7b)}\\
& x_{i,k+1} = F_i(x_{i,k}, u_{i,k}, \Delta t), && i \in \mathcal{N},\ k \in \mathcal{I}_{K-1}, && \text{(7c)}\\
& p_i(t^{\mathrm{in}}_{i,r}, w_i) = p^{\mathrm{in}}_{r,i} && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(7d)}\\
& p_i(t^{\mathrm{out}}_{i,r}, w_i) = p^{\mathrm{out}}_{r,i}, && i \in \mathcal{N},\ r \in \mathcal{R}_i, && \text{(7e)}\\
& h_i(x_{i,k}, u_{i,k}) \leq 0 && i \in \mathcal{N},\ k \in \mathcal{I}_{K-1}, && \text{(7f)}\\
& p_{i,k} + \delta_{ij} \leq p_{j,k}(t) && (i,j) \in \mathcal{C}_R,\ k \in \mathcal{I}_K && \text{(7g)}\\
& t^{\mathrm{out}}_{r,i} \leq t^{\mathrm{in}}_{r,j} && (i,j,r) \in \mathcal{S}. && \text{(7h)}
\end{aligned}$$

where $\mathcal{N} = (1, \dots, N)$, $w = (w_1, \dots, w_N)$, $T = (T_1, \dots, T_N)$ and T_i collects $T_{r,i} = (t^{\mathrm{in}}_{r,i}, t^{\mathrm{out}}_{r,i})$, $\forall r \in \mathcal{R}_i$, with $\mathcal{R}_i$ collecting the CZ crossed by vehicle i.

The objective functions J_i is on the form

$$J_i(w_i) = V^{\mathrm{f}}_i(x_{i,N}) + \sum_{k=0}^{K-1} \ell_i(x_{i,k}, u_{i,k}), \tag{8}$$

with the continuously differentiable terminal cost $V^{\mathrm{f}}_i : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ and stage cost $\ell_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \mapsto \mathbb{R}$. The latter can be defined directly by $x_{i,k}$, $u_{i,k}$ or indirectly through the integration of a function of $\mathrm{x}_i(t), \mathrm{u}_i(t)$ over $[t_k, t_{k+1}]$.

Note that in its "full" form, problem (7) includes selection of the crossing order $\mathcal{S}$. This is a notoriously difficult combinatorial problem, and obtaining exact solutions is in general intractable. In this paper, we assume that the crossing order $\mathcal{S}$ is provided externally through a heuristic and fixed. Problem (7) is consequently denoted the *fixed-order* problem, and is a continuous Nonlinear Program (NLP).

Remark 1. *We present Problem* (7) *for four-way intersections with one lane in each direction and no turning vehicles. However, we emphasize that this is not restrictive and that general geometries could be considered using the same formalism. In particular, we refer to [19] for details on how turning vehicles can be included. Moreover, we note that both* (3) *and* (4) *in practice would be defined with state-dependent margins (e.g. time-headways). Such details are omitted for brevity.*

III. Solution with an Interior Point Algorithm

The primal-dual interior point (PDIP) algorithms are iterative procedures devised to find (local) minimizers of inequality constrained NLPs. They operate by taking Newton-type steps in the primal and dual variables on a perturbed representation of the first-order optimality conditions. By simultaneously driving the perturbation to zero, a sequence of primal-dual solution candidates results. This sequence converge to local minima of the NLP under some conditions [25].

A. A PDIP formulation of the fixed order problem

Collecting $y = (w, T)$, (7a) in $J(w)$, (7b)-(7e) in $g(y)$ and (7f)-(7h) in $h(y)$, the perturbed first order optimality conditions of Problem (7) are

$$\nabla_y \mathcal{L} = 0 \tag{9a}$$
$$g(y) = 0 \tag{9b}$$
$$h(y) + s \leq 0 \tag{9c}$$
$$D(s)\mu - \mathbf{1}\tau = 0 \tag{9d}$$
$$\mu \geq 0 \tag{9e}$$
$$s \geq 0 \tag{9f}$$

Here, s are the slack variables associated with h, $D(s)$ is a the matrix with s on the main diagonal, $\tau \in \mathbb{R}_+$ is the barrier parameter and $\nabla_y \mathcal{L}$ is the gradient of the Lagrange function

$$\mathcal{L}(y, \lambda, \mu) = J(w) + \lambda^\top g(y) + \mu^\top h(y), \tag{10}$$

where λ and μ are the Lagrange multipliers associated with constraints g and h respectively. Collecting y, λ, μ, s in z, we write (9a)-(9d) as $r_\tau(z) = 0$.

Starting from a $z^{[0]}$ strictly satisfying (9e),(9f), the sequence of primal-dual solution candidates is generated through the Newton-iteration

$$z^{[k+1]} = z^{[k]} + \alpha^{[k]} \Delta z^{[k]} \tag{11}$$

where k is the iteration index, $\alpha^{[k]}$ the *step size* and $\Delta z^{[k]}$ the *search-direction*. The latter is obtained as the solution of the KKT-system

$$M(z^{[k]})\Delta z^{[k]} = -r_{\tau^{[k]}}(z^{[k]}). \tag{12}$$

where the matrix $M(z)$ is known as the KKT-matrix, typically constructed as a modification of $\frac{\partial r_\tau}{\partial z}$, evaluated at $z^{[k]}$. The step-size is selected such that the updated solution candidate $z^{[k+1]}$ strictly satisfies (9e), (9f) and provides sufficient decrease on the merit function $\phi(z)$ and strictly satisfies (9e),(9f). Finally, as the PDIP algorithm progresses, the barrier parameter is $\tau^{[k]}$ is updated, i.e.,

$$\tau^{[k+1]} = \xi(\tau^{[k]}, z^{[k]}), \tag{13}$$

using some update strategy ξ, so that eventually $\tau^{[k]} \to 0$.

B. Distribution Strategy

If the PDIP algorithm is applied in a fully centralized setting, the linear system (12) is solved using a standard linear algebra routine. This means that all information needed to assemble $M(z^{[k]})$ and $r_{\tau^{[k]}}(z^{[k]})$ must be made available to the central node before the search-direction $\Delta z^{[k]}$ can be found. However, the structure of Problem (7) admits (12) to be solved with most computations performed separately for each vehicle and for each lane. Due to this, $\Delta z^{[k]}$ can be computed without constructing and solving the full KKT-system centrally. Moreover, the evaluation of the merit function can be split along the same lines, whereby the step size $\alpha^{[k]}$ can be selected in a distributed fashion. In the following sections, we detail how these distributions are made and the information-exchange required between the vehicles, lane centers and intersection center.

IV. Construction of the KKT-system

In this section, we construct $r_{\tau^{[k]}}(z^{[k]})$ and $M(z^{[k]})$ for the fixed-order coordination problem (7), and select a variable ordering which makes the problem structure visible. For brevity, we omit the iteration index and dependence on τ in what follows, and only include the arguments of r and M to highlight specific dependencies.

A. Ordering of the primal-dual variables

We first note that problem (7) is such that the primal variables (i.e. the trajectories w_i and timeslots T_i, $\forall i \in \mathcal{N}$) are associated with a specific vehicle, and that couplings between the primal variables of different vehicles are due to the inequality constraints (7g) and (7h). We also note (7b)-(7e) and the objective function are separable between the vehicles. The problem can thereby be seen as consisting of three levels

1) *The Vehicle Level*, consisting of the objective function and constraints (7b)-(7e). Due to the absence of inter-vehicle couplings, the functions on the vehicle level are separable between the vehicles.
2) *The Lane Level*, consisting of all RECA constraints (7g) which couples vehicles on the same lane. Due to the absence of RECA constraints between vehicles on different lanes, (7g) are separable between the lanes.
3) *The Intersection Level*, consisting of all SICA constraints (7h) which couples vehicles on different lanes.

We denote the number of lanes L, $\mathcal{N}_L = \{1, \dots, L\}$, collect the set of vehicles on lane j in $\mathcal{N}_{l_j}$ and make the following sub-division of the primal-dual variables of NLP (7)

- z_i: collects the primal variables for vehicle i with the slacks and multipliers associated with that vehicle's constraints (7g) and (7h).
- $z_{v(j)}$: collects z_i for all vehicles i on lane j.
- z_v: collects z_i, $\forall i \in \mathcal{N}$.
- $z_{l(i)}$: are the slacks and multipliers for the RECA constraints (7g) of vehicle i
- z_{l_j}: collects the slacks and multipliers for all RECA constraints (7g) involving vehicles on lane j.
- z_T: collects the slacks and multipliers of all SICA constrints

Without loss of generality we assume that the vehicles are ordered as they appear on each lane, i.e., vehicles $i, \dots, n_1$ are on lane 1, vehicles $n_1 + 1, \dots, n_1 + n_2$ are on lane 2 and so on. and order the primal-dual variables as follows

$$z = (z_v, z_c) \tag{14}$$
$$z_v = (z_1, \dots, z_N) \tag{15}$$
$$z_c = (z_l, z_T) \tag{16}$$
$$z_l = (z_{l_1}, \dots, z_{l_L}) \tag{17}$$

In the remainder of this section we construct the KKT system for this ordering of the variables. We perform the construction in steps, and first introduce the equations related to the individual vehicles (7a)-(7e), followed by the equations associated with the coupling constraints.

B. Components relating to the vehicles

For ease of presentation, we introduce the *vehicle problem*

$$\min_{y_i} \quad J_i(w_i) \tag{18a}$$

$$\text{s.t.} \quad C_i^E(w_i, T_i) = 0 \tag{18b}$$

$$C_i^I(w_i) \leq 0 \tag{18c}$$

where $y_i = (w_i, T_i)$ and written constraints (7b)-(7e) for vehicle i as $C_i^E(y_i) = 0$ and constraints (7f) as $C_i^I(w_i) \leq 0$. That is, the solution to (18) is the "greedy" solution of vehicle i, where all couplings to other vehicles are ignored. The Lagrange function of (18) is

$$\mathcal{L}_i(z_i) = J_i(w_i) + \lambda_i^\top C_i^E(w_i, T_i) + \mu_i^\top C_i^I(w_i) \tag{19}$$

where λ_i and μ_i are the multipliers associated with (18b) and (18c), respectively. Defining $z_i = (y_i, \lambda_i, \mu_i, s_i)$, where s_i are the slack variables associated with constraint (18c), the KKT system for (18) reads $M_i \Delta z_i = -r_i(z_i)$, where

$$r_i(z_i) = \begin{bmatrix} \nabla_{y_i} \mathcal{L} \\ C_i^E(w_i, T_i) \\ C_i^I(w_i) + s_i \\ \mu_i - D(s_i)^{-1} \mathbf{1} \tau \end{bmatrix}, \tag{20}$$

and

$$M_i(z_i) = \begin{bmatrix} \nabla_{y_i}^2 \mathcal{L}_i & \nabla_{y_i} C_i^E & \nabla_{y_i} C_i^I & \\ \nabla_{y_i} {C_i^E}^\top & & & \\ \nabla_{y_i} {C_i^I}^\top & & & I \\ & & I & D(s_i)^{-1} D(\mu_i) \end{bmatrix} \tag{21}$$

C. Components relating to the coupling constraints

Collecting the position of vehicle i in $p_i = (p_{i,0}, \ldots, p_{i,K})$, we write RECA constraint (7g) between vehicle i and $i+1$ as

$$C_{l,(i,i+1)}(p_i, p_{i+1}) = p_{i+1} - p_i + \mathbf{1} \delta_{i+1,i} \leq 0 \tag{22}$$

and denote the corresponding Lagrange multipliers and slack variables as $\mu_{l,(i,i+1)}$, $s_{l,(i,i+1)}$ respectively. All rear-end collision avoidance constraints (22) for the vehicles on lane j and the associated multipliers and slacks are collected in

$$C_{l_j}(\bar{p}_j) = \begin{bmatrix} C_{l,(i,i+1)}(p_i, p_{i+1}) \\ \vdots \\ C_{l,(i+n-1,i+n)}(p_{i+n_j-1}, p_{i+n_j}) \end{bmatrix} \leq 0 \tag{23}$$

$$\mu_{l_j} = (\mu_{l,(i,i+1)}, \ldots, \mu_{l,(i+n_j-1,i+n_j)}) \tag{24}$$

$$s_{l_j} = (s_{l,(i,i+1)}, \ldots, s_{l,(i+n_j-1,i+n_j)}) \tag{25}$$

where $\bar{p}_j = (p_i, \ldots, p_{i+n_j})$ and $\{i, \ldots, i+n_j\} = \mathcal{N}_{l_j}$ are the indices of the n_j vehicles on lane j. Similarly, with $\bar{p} = (\bar{p}_1, \ldots, \bar{p}_L)$, we collect all RECA constraints with the associated slacks and multipliers in

$$C_l(\bar{p}) = \begin{bmatrix} C_{l_1}(\bar{p}_1) \\ \vdots \\ C_{l_L}(\bar{p}_L) \end{bmatrix} \tag{26}$$

$$\mu_l = (\mu_{l_1}, \ldots, \mu_{l_L}) \tag{27}$$

$$s_l = (s_{l_1}, \ldots, s_{l_L}). \tag{28}$$

Finally, we collect all SICA constraints (7h) in

$$C_T(T) \leq 0 \tag{29}$$

and denote the associated Lagrange multipliers and slacks in μ_T and s_T, respectively. With this, we let

$$z_{l_j} = (\mu_{l_j}, s_{l_j}) \tag{30}$$

$$z_l = (z_{l_1}, \ldots, z_{l_L}) \tag{31}$$

$$z_T = (\mu_T, s_T) \tag{32}$$

D. Centralized KKT System

The Lagrange function of NLP (10) can thus be written as

$$\mathcal{L}(z) = \mathcal{L}_v(z_v) + \mathcal{L}_c(T, \bar{p}, z_c), \tag{33}$$

$$\mathcal{L}_v(z_v) = \sum_i^N \mathcal{L}_i(z_i), \tag{34}$$

$$\mathcal{L}_c(T, \bar{p}, z_c) = \mathcal{L}_T(T, z_T) + \mathcal{L}_l(\bar{p}, z_l), \tag{35}$$

$$\mathcal{L}_T(T, z_T) = \mu_T^\top C_T(T), \tag{36}$$

$$\mathcal{L}_l(\bar{p}, z_l) = \sum_{j=1}^L \mathcal{L}_{l_j}(\bar{p}_j, z_{l_j}), \tag{37}$$

$$\mathcal{L}_l(\bar{p}_j, z_{l_j}) = \mu_{l_j}^\top C_{l_j}(\bar{p}_j), \tag{38}$$

and we re-state the KKT-system of Problem (7) as

$$M(z) \Delta z = -r(z), \tag{39}$$

where

$$r(z) = \begin{bmatrix} r_v(z_v, \mu_l, \mu_T) \\ r_c(T, \bar{p}, z_c) \end{bmatrix}, \tag{40}$$

with

$$r_v(z_v, \mu_l, \mu_T) = \begin{bmatrix} r_1(z_1) + \nabla_{z_1} \mathcal{L}_c \\ \vdots \\ r_N(z_N) + \nabla_{z_N} \mathcal{L}_c \end{bmatrix}, \tag{41}$$

and

$$r_c(T, \bar{p}, z_c) = \begin{bmatrix} r_l(\bar{p}, z_l) \\ r_T(T, z_T) \end{bmatrix}, \tag{42}$$

$$r_l(\bar{p}, z_l) = \begin{bmatrix} r_{l_j}(\bar{p}_1, z_{l_1}) \\ \vdots \\ r_{l_L}(\bar{p}_L, z_{l_L}) \end{bmatrix}, \tag{43}$$

$$r_{l_j}(\bar{p}_j, z_{l_j}) = \begin{bmatrix} C_{l_j}(\bar{p}_j) + s_{l_j} \\ \mu_{l_j} - D(s_{l_j})^{-1} \mathbf{1} \tau \end{bmatrix}, \tag{44}$$

$$r_T(T, z_T) = \begin{bmatrix} C_T(T) + s_T \\ \mu_T - D(s_T)^{-1} \mathbf{1} \tau \end{bmatrix}. \tag{45}$$

In the next section, we detail the structure of $M(z)$ and show how the solution to (39) can be distributed.

V. Distributed solution of the KKT-system

The distributed solution of (39) are made possible by the structure of the KKT-matrix, and uses the Shur-complement

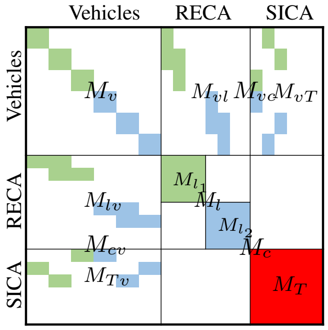
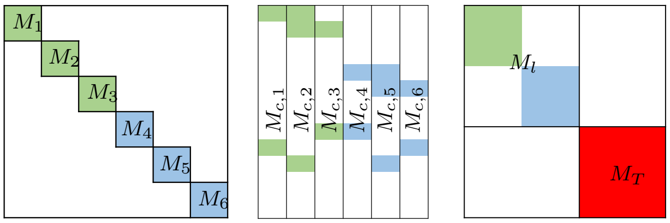


Fig. 4: Illustration of the KKTMatrix of (7).



(a) M_v

(b) M_{cv}

(c) M_c

Fig. 5: Illustration of the sub-matrices involved in the first decomposition level from the KKT matrix shown in Fig. 4

of different sub-matrices of $M(z)$ in two steps. In particular, the structure of $M(z)$ is such that the following hold

$$\text{Centralized KKT System (39)} \Leftrightarrow \begin{array}{l}\text{Coupling Sub-system (54a)}\\ \text{Vehicle Sub-systems (54b)}\end{array}$$

$$\text{Coupling Sub-system (54a)} \Leftrightarrow \begin{array}{l}\text{SICA Sub-system (60a)}\\ \text{RECA Sub-systems (60b)}\end{array}$$

An illustration of the KKT matrix structure is given in Fig. 4, where we use the notation

$$M_v(z_v) = \text{blockdiag}(M_1, \dots, M_N), \tag{46}$$
$$M_c(z_c) = \text{blockdiag}(M_l, M_T), \tag{47}$$
$$M_l(z_l) = \text{blockdiag}(M_{l_1}, \dots, M_{l_L}), \tag{48}$$
$$M_{l_j}(z_{l_j}) = \begin{bmatrix} 0 & I \\ I & D(s_{l_j})^{-1}D(\mu_{l_j}) \end{bmatrix}, \tag{49}$$
$$M_T(z_T) = \begin{bmatrix} 0 & I \\ I & D(s_T)^{-1}D(\mu_T) \end{bmatrix}, \tag{50}$$

for the diagonal blocks, and

$$M_{vc} = \begin{bmatrix} M_{vl} & M_{vT} \end{bmatrix}^\top, \tag{51}$$
$$M_{vl} = \nabla_{z_v} C_l(\bar{p}), \qquad M_{vT} = \nabla_{z_v} C_T(T), \tag{52}$$

for the cross terms, and define $M_{cv} = M_{vc}^\top$, $M_{Tv} = M_{vT}^\top$ and $M_{lv} = M_{vl}^\top$ We also note that due to the linearity of C_l and C_T, M_{cv} are constant matrices.

A. Separation between the Vehicles and Couplings

Note that the KKT matrix $M(z)$ can be sub-divided as

$$M(z) = \begin{bmatrix} M_v & M_{cv}^\top \\ M_{cv} & M_c \end{bmatrix} \tag{53}$$

With the Schur-complement of M_c in M, (39) can be solved as

$$\left(M_c - M_{cv}M_v^{-1}M_{cv}^\top\right)\Delta z_c = -r_c + M_{cv}M_v^{-1}r_v \tag{54a}$$
$$M_v \Delta z_v = -r_v - M_{cv}^\top \Delta z_c. \tag{54b}$$

That is, we can separate computation of the search direction in the coupling variables, Δz_c, from that in the variables associated with the vehicles, Δz_v, by solving (54a) and (54b) in sequence. We show next that $M_{cv}M_v^{-1}M_{cv}$ and $M_{cv}M_v^{-1}r_v$ consist of independent components from each vehicle.

Distributed solution of the KKT-System, first level: An illustration of the the sub-matrices M_v, M_{cv} and M_c are shown in Fig. 5c, and we note that M_{cv} can be divided into components associated with each vehicle

$$M_{cv} = \left[M_{c,1}, \dots, M_{c,N}\right], \qquad M_{c,i} = \frac{\partial r_c}{\partial z_i}. \tag{55}$$

Since M_v is block diagonal with one block M_i for each vehicle, we have that

$$M_{cv}M_v^{-1}M_{cv}^\top = \sum_{i=1}^{N} M_{c,i}M_i^{-1}M_{c,i}^\top \tag{56}$$
$$M_{cv}M_v^{-1}r_v = \sum_{i=1}^{N} M_{c,i}M_i^{-1}\left(r_i + \nabla_{z_i}\mathcal{L}_c\right). \tag{57}$$

Since $M_{c,i}$ are constant matrices, the terms required to assemble $M_{cv}M_v^{-1}M_{cv}^\top$ and $M_{cv}M_v^{-1}r_v$ can be computed separately for each vehicle. Due to the structure of M_v and M_{cv}, (54b) can be solved independently for each i as

$$M_i \Delta z_i = -r_i - \nabla_{z_i}\mathcal{L}_c - M_{c,i}^\top \Delta z_c \tag{58}$$

where we stress that $M_{c,i}^\top \Delta z_c$ only requires components of Δz_c relating to couplings involving vehicle i.

Since (44) and (45) are such that M_c is independent of z_v, the solution to (39) can be found in three steps: 1) Compute $M_{c,i}M_i^{-1}M_{c,i}^\top$ and $M_{c,i}M_i^{-1}r_i$ $\forall i \in \mathcal{N}$, 2) assemble and solve (54b), 3) solve (58) $\forall i \in \mathcal{N}$.

B. Separation between RECA and SICA couplings

We define

$$\begin{bmatrix} \Gamma & \Lambda^\top \\ \Lambda & \Psi \end{bmatrix} = \begin{bmatrix} M_l - M_{lv}M_v^{-1}M_{lv}^\top & -M_{lv}M_v^{-1}M_{Tv}^\top \\ -M_{Tv}M_v^{-1}M_{lv}^\top & M_T - M_{Tv}M^{-1}M_{Tv}^\top \end{bmatrix}, \tag{59a}$$
$$\gamma = -r_l + M_{lv}M_v^{-1}r_v, \tag{59b}$$
$$\psi = -r_T + M_{Tv}^\top M_v^{-1}r_v, \tag{59c}$$

so that matrix (59a) and vectors (59b),(59c) are the left- and right hand side of (54a), respectively. With the Schur complement of Ψ in (59a), (60a) can be solved as

$$\left(\Psi - \Lambda\Gamma^{-1}\Lambda^\top\right)\Delta z_T = \psi - \Lambda\Gamma^{-1}\gamma \tag{60a}$$
$$\Gamma \Delta z_l = \gamma - \Lambda^\top \Delta z_T. \tag{60b}$$

That is, the direction in the variables associated with the SICA constraints, Δz_T can be separated from that of the variables associated with the RECA constraints, Δz_l, by solving (60a) and (60b) in sequence. We show next that $\Lambda\Gamma^{-1}\Lambda^\top$ and $\Lambda\Gamma^{-1}\gamma$ consists of independent components from each lane.

Distributed solution of the KKT-System, second level: Note that Λ is defined by sums of terms from each vehicle due to (56). However, constraints (23) only couple vehicles on the same lane and constraints (29) only couple vehicles on different lanes, which can be used to separate the computations required to construct Λ between the lanes. Recalling that the vehicles are ordered as they appear on their lanes, we let

- $r_{v(j)}(z_{v(j)}, z_{l_j}, z_T)$ collect $r_i(r_i) - \nabla_{z_i}\mathcal{L}_c\ \forall i \in \mathcal{N}_{l_j}$,
- $M_{v(j)}(z_{v(j)})$ be the block diagonal matrix with blocks M_i, $i \in \mathcal{N}_{l_j}$,

so that

$$M_{lv} = \text{blockdiag}\left(\frac{\partial r_{l_1}}{\partial z_{v(1)}}, \dots, \frac{\partial r_{l_L}}{\partial z_{v(L)}}\right), \tag{61}$$

$$M_{Tv} = \left[\frac{\partial r_T}{\partial z_{v(1)}}, \dots, \frac{\partial r_T}{\partial z_{v(L)}}\right], \tag{62}$$

$$M_v = \text{blockdiag}(M_{v(1)}, \dots, M_{v(L)}), \tag{63}$$

where

$$\frac{\partial r_T}{\partial z_{v(j)}} = \begin{bmatrix} \frac{\partial C_T}{\partial z_{v(j)}} \\ 0 \end{bmatrix}, \quad \frac{\partial r_{l_j}}{\partial z_{v(j)}} = \begin{bmatrix} \frac{\partial C_{l_j}}{\partial z_{v(j)}} \\ 0 \end{bmatrix}, \tag{64}$$

and

$$-M_{Tv}M_v^{-1}M_{lv}^\top = \Lambda = [\Lambda_1, \dots, \Lambda_L], \tag{65}$$

where

$$\Lambda_j(z_{v(j)}) = \begin{bmatrix} -\frac{\partial C_T}{\partial z_{v(j)}} M_{v(j)}^{-1} \left(\frac{\partial C_{l_j}}{\partial z_{v(j)}}\right)^\top & 0 \\ 0 & 0 \end{bmatrix}, \tag{66}$$

and

$$M_{lv}M_v^{-1}M_{lv}^\top = \text{blockdiag}\left(\frac{\partial r_{l_1}}{\partial z_{v(1)}} M_{v(1)}^{-1} \left(\frac{\partial r_{l_1}}{\partial z_{v(1)}}\right)^\top, \dots, \frac{\partial r_{l_L}}{\partial z_{v(L)}} M_{v(L)}^{-1} \left(\frac{\partial r_{l_L}}{\partial z_{v(L)}}\right)^\top\right), \tag{67}$$

where

$$\frac{\partial r_{l_j}}{\partial z_{v(j)}} M_{v(j)}^{-1} \left(\frac{\partial r_{l_j}}{\partial z_{v(j)}}\right)^\top = \begin{bmatrix} \frac{\partial C_{l_j}}{\partial z_{v(j)}} M_{v(1)}^{-1} \left(\frac{\partial C_{l_j}}{\partial z_{v(j)}}\right)^\top & 0 \\ 0 & 0 \end{bmatrix}. \tag{68}$$

Since $M_l(z_l) = \text{blockdiag}(M_{l_1}, \dots, M_{l_L})$, we have

$$\Gamma(z_v, z_l) = \text{blockdiag}(\Gamma_1(z_{v(1)}, z_{l_1}), \dots, \Gamma_L(z_{v(L)}, z_{l_L})), \tag{69}$$

where

$$\Gamma_j(z_{v(j)}, z_{l_j}) = M_{l_j} - \frac{\partial r_{l_j}}{\partial z_{v(j)}} M_{v(j)}^{-1} \left(\frac{\partial r_{l_j}}{\partial z_{v(j)}}\right)^\top, \tag{70}$$

and $\gamma = (\gamma_1, \dots, \gamma_L)$, where

$$\gamma_j = -r_{l_j} + \frac{\partial r_{l_j}}{\partial z_{v(j)}} M_{v(j)} r_{v(j)} \tag{71}$$

Since all SICA and RECA constraints are linear, Λ_j, the blocks of (67) and the product in (71) are thereby defined solely by the primal-dual variables of vehicles on lane j (through $M_{v(j)}$ and $r_{v(j)}$) and the multipliers of the couplings in which they take part (through the $\nabla_{z_i}\mathcal{L}_c$ terms in $r_{v(j)}$). Due to this, the left- and right hand side of (60a) are

$$\Lambda\Gamma^{-1}\Lambda^\top = \sum_{j=1}^{L} \Lambda_j \Gamma_j^{-1} \Lambda_j^\top \tag{72a}$$

$$\Lambda\Gamma^{-1}\gamma = \sum_{j=1}^{L} \Lambda_j \Gamma_j^{-1} \gamma_j, \tag{72b}$$

i.e., the terms in $\Lambda\Gamma^{-1}\Lambda^\top$ and $\Lambda\Gamma^{-1}\gamma$ can be computed independently for each lane. Finally, Λ and Γ are such that (60b) can be solved as the L independent sets of equations

$$\Gamma_j \Delta z_{l_j} = \gamma_j - \Lambda_j^\top \Delta z_T, \qquad j \in \mathcal{N}_L, \tag{73}$$

i.e., one set of equations for each lane.

Consequently, (39) can be solved in the following steps: 1) compute the components in (59a), (59b) and (59c) for each vehicle, 2) compute the terms in (72) for each lane, 3) solve (60a) for Δz_T, 4) solve (73) for Δz_{l_j} (using Δz_T), 5) solve (58) for Δz_i (using Δz_T and $\Delta z_l(i)$).

C. Algorithm

We first note that $\frac{\partial r_{l_j}}{\partial z_{v(j)}} M_{v(j)}^{-1} \left(\frac{\partial r_{l_j}}{\partial z_{v(j)}}\right)^\top$ in Γ_j is sparse, where all non-zero elements are in the blocks

$$\frac{\partial C_{l_1}}{\partial z_{v(1)}} M_{v(1)}^{-1} \left(\frac{\partial C_{l_1}}{\partial z_{v(1)}}\right)^\top = \begin{bmatrix} c_1 + c_2 & -c_2 & & & & \\ -c_2 & c_2 + c_3 & -c_3 & & & \\ & -c_3 & c_3 + c_4 & -c_4 & & \\ & & \ddots & \ddots & \ddots & \\ & & & & & -c_{n_1-1} \\ & & & & -c_{n_1-1} & c_{n_1} \end{bmatrix} \tag{74}$$

where

$$c_i = \frac{\partial z_i}{\partial p_i}^\top M_i^{-1} \frac{\partial z_i}{\partial p_i}. \tag{75}$$

Similarly, Λ_j is also sparse, with all non-zero elements in the blocks $\frac{\partial C_T}{\partial z_{v(j)}} M_{v(j)}^{-1} \frac{\partial C_{l_j}}{\partial z_{v(j)}}$, respectively. The latter are in turn built up by the vehicle specific blocks

$$I_i^- \frac{\partial z_i}{\partial T_i}^\top M_i^{-1} \frac{\partial z_i}{\partial p_i} \tag{76}$$

$$\frac{\partial z_i}{\partial p_i}^\top M_i^{-1} \frac{\partial z_i}{\partial T_i} I_i^- \tag{77}$$

where I_i^- is a square matrix of size $\text{sizeof}(T_i)$, with 1 or -1 on the diagonal and zero otherwise, where the sign of the individual diagonal entires depend on the corresponding T_i's entry in constraint C_T.

Finally, the term $M_{Tv}M_v^{-1}M_{Tv}$ in Ψ is also sparse, with all non-zero elements in the block $\frac{\partial C_T}{\partial z_v} M_v^{-1} \frac{\partial C_T}{\partial z_v}^\top$, built up by the following elements from the vehicles

$$I_i^- \frac{\partial z_i}{\partial T_i}^\top M_i^{-1} \frac{\partial z_i}{\partial T_i} I_i^- \tag{78}$$

Algorithm 1 Distributed solution of KKT system. Here, C denotes the central unit for the intersection, L_j denotes the central unit for lane j, and A_i denotes vehicle i. All vehicles are assumed to hold local copies of the corresponding μ_i^f, μ_i^r and μ_i^T.

```
1: procedure SearchDirection($z$,$\tau$)
2:     $\forall A_i$: Compute $\mathcal{D}_{A_i \to C}$,$\mathcal{D}_{A_i \to l(i)}$, and pass to $C$,$L_j$
       where $j$ is the lane of vehicle $i$
3:     $\forall L_j$: Compute $\Lambda_j \Gamma_j^{-1} \Lambda_j^\top$, $\Lambda_j \Gamma_j^{-1} \gamma_j$ and pass to $C$
4:     $C_T$ : Assemble and solve (60a), pass the appropriate
       parts of $\Delta\mu_T$ to all $C_{l_j}$, and all $A_i$.
5:     $\forall L_j$: Solve (73) for $\Delta z_{l_j}$, using the received components
       of $\Delta\mu_T$, pass the appropriate parts of $\Delta\mu_{l_j}$ to all
       $A_i$ on lane $j$
6:     $\forall A_i$: Solve (58) for $\Delta z_i$, using the received components
       of $\Delta\mu_T$ and $\Delta\mu_{l_j}$
7: end procedure
```

in a manner similar to (74). However, unlike the RECA constraints (23), we there is in general is no "chain-structure" in the SICA constraints (29), due to which the resulting matrix $\frac{\partial C_T}{\partial z_v} M_v^{-1} \frac{\partial C_T}{\partial z_v}^\top$ is not block-banded as (74).

With this we note that each vehicle needs to pass the following information to its lane center

$$\frac{\partial z_i}{\partial p_i}^\top M_i^{-1} \frac{\partial z_i}{\partial p_i}, \qquad \frac{\partial z_i}{\partial T_i}^\top M_i^{-1} \frac{\partial z_i}{\partial p_i}, \tag{79a}$$

$$\frac{\partial z_i}{\partial p_i}^\top M_i^{-1} (r_i + \nabla_{z_i} \mathcal{L}_c), \qquad p_i, \tag{79b}$$

and the following information to the intersection center

$$\frac{\partial z_i}{\partial T_i}^\top M_i^{-1} \frac{\partial z_i}{\partial T_i}, \quad \frac{\partial z_i}{\partial T_i}^\top M_i^{-1} (r_i + \nabla_{z_i} \mathcal{L}_c), \quad T_i. \tag{80}$$

Each lane center on the other hand, needs to assemble and solve $\Lambda_j \Gamma_j^{-1} \Lambda_j^\top$, $\Lambda_j \Gamma_j^{-1} \gamma_j$, and pass the results to the intersection center.

Moreover, $\nabla_{z_i} \mathcal{L}_c$ is a sparse vector, with the only non-zero elements being on the positions corresponding to p_i and T_i. These consist of the (signed) multipliers of the corresponding constraints, so that

$$\nabla_{z_i} \mathcal{L}_c = \frac{\partial z_i}{\partial p_i} (\mu_{i,r} - \mu_{i,f}) + \frac{\mu z_i}{\partial T_i} I_i^- \mu_{i,T}, \tag{81}$$

where $\mu_{i,f}$, $\mu_{i,r}$ denotes the multipliers for the RECA constraints (elements of μ_{l_j} for the vehicle's lane) where the vehicle is in front and to the rear, respectively, and where $\mu_{i,T}$ are the multipliers for the SICA constraints in which the vehicle is involved (elements of μ_T). Collecting the matrices and vectors of (79) in $\mathcal{D}_{A_i \to l(i)}$ and those of (80) in $\mathcal{D}_{A_i \to C}$, we summarize the distributed solution of (39) in Algorithm 1.

Note that Lines 2 and 3 are separable between the vehicles, where the main effort consist of the (parallel) factorization of matrices M_i. Similarly, Lines 3 and 5 are separable between the lanes, and the main effort consists of the (parallel) factorization of matrices Γ_j. The factors of M_i and Γ_j can be stored and reused on Lines 6 and 5, respectively.

An interpretation of Algorithm 1, is that the search direction in the vehicle variables z_i is constructed as

$$\Delta z_i = \Delta z_i^{\text{Greedy}} + \Delta z_i^{\text{RECA}} + \Delta z_i^{\text{SICA}} \tag{82}$$

where

$$\Delta z_i^{\text{Greedy}} = -M_i^{-1} r_i \tag{83}$$

is the "greedy" step, ignoring couplings to other vehicles,

$$\Delta z_i^{\text{RECA}} = -M_i^{-1} \frac{\partial z_i}{\partial p_i} (\mu_{i,r} + \Delta\mu_{i,r} - \mu_{i,f} - \Delta\mu_{i,f}) \tag{84}$$

corrects for couplings to vehicles on the same lane,

$$\Delta z_i^{\text{SICA}} = -M_i^{-1} \frac{\partial z_i}{\partial T_i} I_i^- (\mu_{i,T} + \Delta\mu_{i,T}) \tag{85}$$

corrects for couplings to vehicles on other lanes.

VI. Distributed computation of the step-size

In this section, we discuss selection of the step size α through a back-tracking line-search on a merit function where most computations can be distributed.

A. Feasibility enforcing step-size selection

To ensure that α is chosen so that $s^{[k+1]} > 0$, $\mu^{[k+1]} > 0$, we employ the *fraction from the boundary* rule

$$s + \alpha^{\max} \Delta s \geq \kappa s \tag{86a}$$

$$\mu + \alpha^{\max} \Delta\mu \geq \kappa s \tag{86b}$$

where $\kappa > 0$ is a parameter [25]. Due to the problem structure, (86) can be evaluated separately for the vehicles, giving $\alpha_i^{\max}$, $\forall i \in \mathcal{N}$, for the RECA constraints on a lane, giving $\alpha_{l_j}^{\max}$, $\forall j \in \mathcal{N}_L$ and for the SICA constraints $\alpha_T^{\max}$. The maximum allowed step size for the search direction Δz is thereby

$$\alpha^{\max} = \min(\alpha_1^{\max}, \dots, \alpha_N^{\max}, \alpha_{l_1}^{\max}, \dots, \alpha_{l_L}^{\max}, \alpha_T^{\max}) \tag{87}$$

B. Solution-improving step-size selection

We find $\alpha \leq \alpha^{\max}$ which improves the solution by back-tracking on the merit function suggested in [25], which reads

$$\phi(y, s) = \sum_{i=1}^{N} \phi_i(y_i, s_i) + \sum_{j=1}^{L} \phi_{l_j}(\bar{p}_j, s_{l_j}) + \phi_T(T, s_T) \tag{88}$$

where

$$\begin{aligned} \phi_i(y_i, s_i) &= J_i(w_i) + \nu \left(||C_i^E(w_i, T_i)||_1 + ||C_i^I(w_i) + s_i||_1 \right) \\ &\quad - \tau \mathbf{1}^\top \log(s_i), \\ \phi_{l_j}(\bar{p}_j, s_{l_j}) &= \nu ||C_{l_j}(\bar{p}_j) + s_{l_j}||_1 - \tau \mathbf{1}^\top \log(s_{l_j}), \\ \phi_T(T, s_T) &= \nu ||C_T(T) + s_T||_1 - \tau \mathbf{1}^\top \log(s_T), \end{aligned}$$

with the logarithm taken element-wise and parameter ν. We use the Armijo condition, and accept a step α when

$$\phi(y + \alpha \Delta y, s + \alpha \Delta s) \leq \phi(y, s) + \zeta \phi'(y, s) \alpha, \tag{89}$$

Algorithm 2 Distributed selection of step-size α, first level. C denotes the central unit, L_j denotes the central unit for lane j, and A_i denotes vehicle i. Parameters: $\gamma \in]0, 0.5]$, $\beta \in]0, 1]$

```
procedure StepSizeSelection($z$,$\Delta z$,$\tau$)
    $\forall A_i$: Find $\alpha_i^{\max}$, assemble $\phi_i(y_i, s_i)$ and $\phi_i'(y_i, s_i)$, pass to $C$ together with $\Delta T_i$, pass $\Delta p_i$ to $L_j$.
    $\forall L_j$: Find $\alpha_{l_j}^{\max}$, assemble $\phi_{l_j}(\bar{p}_j, s_{l_j})$ and $\phi_i'(\bar{p}_j, s_{l_j})$
    $C$: Find $\alpha_T^{\max}$ and determine $\alpha^{\max}$ with (87).
    $C$: Find $\phi_T(T, s_T)$,$\phi_T'(T, s_T)$ ass. $\phi(y, s)$,$\phi'(y, s)$
    $C$: Set $\alpha = \alpha^{\max}$
    loop
        $\forall A_i$: Pass $\phi_i(y_i + \alpha\Delta y_i, s_i + \alpha\Delta s_i)$ to $C$
        $\forall L_j$: Pass $\phi_{l_j}(\bar{p}_j + \alpha\Delta\bar{p}_j, s_{l_j} + \alpha\Delta s_{l_j})$ to $C$
        $C$: Compute $\phi_T(T + \alpha\Delta T, s_T + \alpha\Delta s_T)$, assemble $\phi(y + \alpha\Delta y, s + \alpha\Delta s)$ through (88)
        if $\phi(y + \alpha\Delta y, s + \alpha\Delta s) < \phi(y, s) + \alpha\gamma\phi'(y, s)$ then
            return $\alpha$, $\alpha^{\max}$ and accept notice to all $A_i$, $L_j$
        else
            $\alpha \leftarrow \beta\alpha$
            Pass $\alpha$ to all $A_i$, $L_j$
        end if
    end loop
end procedure
```

Algorithm 3 A Basic Distributed Primal-Dual Interior Point algorithm for the fixed order intersection problem.

```
procedure FixedOrderPDIP($\tau^{[0]}$)
    $k \leftarrow 0$
    $C$ : Initialize $z_T^{[0]}$ and send $\mu^{[0]}$ to all $A_i$ and $L_j$
    $\forall L_j$: Initialize $z_{l_j}^{[0]}$ and send $\mu_{l_j}^{[0]}$ to $A_i$ on lane $j$.
    $\forall A_i$: Initialize $z_i^{[0]}$ and send $T_i^{[0]}$ to $C$, $p_i^{[0]}$ to $L_j$
    loop
        $k \leftarrow k + 1$
        $C$: Send iteration start and $\tau^{[k]}$ to all $A_i$, $L_j$.
        $\forall A_i$: Compute $M_i, r_i$ , modify if necessary $\nabla_{y_i}^2\mathcal{L}_i$.
        $\Delta z^{[k]}$ ←SearchDirection($z^{[k]}$,$\tau^{[k]}$)
        $\alpha^{[k]}$ ←StepSizeSelection$z$,$\Delta z$,$\tau$
        $C$: Update $z_T^{[k+1]} \leftarrow z_T^{[k]} + \alpha^{[k]}\Delta z_T^{[k]}$
        $\forall L_j$: Update $z_{l_j}^{[k+1]} \leftarrow z_{l_j}^{[k]} + \alpha^{[k]}\Delta z_{l_j}^{[k]}$
        $\forall A_i$: Update $z_i^{[k+1]} \leftarrow z_i^{[k]} + \alpha^{[k]}\Delta z_i^{[k]}$
        if Terminate($z^{[k+1]}$,$\Delta z^{[k]}$,$\tau^{[k]}$) then
            return Solution found
        else
            $\tau^{[k+1]} \leftarrow$ UpdateBarrierParameter()
        end if
    end loop
end procedure
```

where $\zeta \in]0, 0.5]$ is a parameter, and

$$\phi'(y, s) = \left.\frac{\mathrm{d}\phi(y + \alpha\Delta y, s + \alpha\Delta s)}{\mathrm{d}\alpha}\right|_{0 \leftarrow \alpha} = \frac{\partial\phi}{\partial y}\Delta y + \frac{\partial\phi}{\partial s}\Delta s$$
$$= \sum_{i=1}^{N} \phi_i'(y_i, s_i) + \sum_{j=1}^{L} \phi_{l_j}'(\bar{p}_j, s_{l_j}) + \phi_T'(T, s_T). \quad (90)$$

Noting that computation of $\phi(y, s)$, $\phi'(y, s)$ can be separated between the vehicles $(\phi_i(y_i, s_i)$,$\phi_i'(y_i, s_i))$, the lanes $(\phi_{l_j}(\bar{p}_j, s_{l_j})$,$\phi_{l_j}'(\bar{p}_j, s_{l_j}))$ and the intersection $(\phi_T(T, s_T)$,$\phi_T'(T, s_T))$, Algorithm 2 summarizes the selection of α.

C. Handling non-convexity

It is known [25] that $(\Delta y, \Delta s)$ is a descent direction on ϕ, if

$$v^\top \begin{bmatrix} \nabla_y^2\mathcal{L} & \\ & D(s)^{-1}D(\mu) \end{bmatrix} v > 0, \ \forall v \ : \ \begin{bmatrix} \nabla_y g^\top & \\ \nabla_y h^\top & I \end{bmatrix} v = 0. \quad (91)$$

Since $D(s)^{-1}D(\mu) \succ 0$ by construction, this is determined by $\nabla_y^2\mathcal{L}$. If (91) does not hold, a modification of $\nabla_y^2\mathcal{L}$ can be used. One particular (and likely conservative) alternative, is to find $U \succeq 0$ such that $H = \nabla_y^2\mathcal{L} + U \succ 0$, and use H in place of $\nabla_y\mathcal{L}$. Importantly, such modification could be applied independently for all vehicles, since $\nabla_y^2\mathcal{L} = \mathrm{blockdiag}(\nabla_{y_1}^2\mathcal{L}_1, \ldots, \nabla_{y_N}^2\mathcal{L}_N)$. That is, Δz is a descent direction on $\phi(y, s)$ if each vehicle uses a positive definite modification of $\nabla_{y_i}^2\mathcal{L}_i$ when necessary.

VII. A practical Algorithm

A basic procedure which uses Algorithms 1 and 2 is summarized in Algorithm 3. Note that Algorithm 3 gives exactly the same iterates and has the same convergence properties as a fully centralized algorithm.

1) Termination Criteria: We use the norm of r as termination criteria, such algorithm terminates when

$$||r_{\tau^{[k]}}(z^{[k+1]})|| < \varepsilon \quad \text{and} \quad \tau^{[k]} < \varepsilon, \quad (92)$$

for some tolerance ε. Termination must thus be decided centrally, and the components of r that relate to the lane couplings and vehicles must be available at the central node.

2) Barrier Parameter Update: While elaborate schemes are possible for updates of τ, we employ the Fiacco-McCormick rule for simplicity. In particular, we update $\tau^{[k+1]} \leftarrow \eta\tau^{[k]}$, where the parameter $\eta \in]0, 1[$, when $||r_{\tau^{[k]}}(z^{[k+1]})|| < \tau^{[k]}$. The barrier parameter must thus be decided centrally.

A. Example

As an example, we consider a scenario with three vehicles on each lane. Assuming that all vehicles are electric, their motion is described by

$$\dot{\mathrm{p}}_i(t) = \mathrm{v}_i(t), \quad (93a)$$
$$\dot{\mathrm{v}}_i(t) = \frac{1}{m_i}(c^{\mathrm{Torque}}\mathrm{M}_i(t) - \mathrm{F}_i^{\mathrm{b}} - c^{\mathrm{drag}}\mathrm{v}_i(t)^2 - c^{rr}), \quad (93b)$$
$$\mathrm{M}(t) \leq \min(\mathrm{M}^{\max}, P_i^{\max}/\omega_i(t)) \quad (93c)$$
$$0 \leq \omega_i(t) \leq \omega_i^{\max}, \quad (93d)$$

where $\mathrm{M}_i(t)$ is the motor torque, $\mathrm{F}_i^{\mathrm{b}}(t)$ the friction brake force, $\omega_i(t) = c^\omega\mathrm{v}_i(t)$ the motor speed and $\mathrm{x}_i(t) =$

$(\mathrm{p}_i(t), \mathrm{v}_i(t))$, $\mathrm{u}_i(t) = (\mathrm{M}_i(t), \mathrm{F}_i^{\mathrm{b}}(t))$. The parameters $c^{Torque}, c^{\omega}, c^{\mathrm{drag}}, c^{\mathrm{rr}}, \omega_i^{\max}, \mathrm{M}^{\max}$ and are selected as in $P_i^{\max}$ [18], and we use $K = 100$ and Explicit 4th order Runge-Kutta integrators from ACADO [26] with $\Delta t = 0.2$. The objective function is

$$J_i(w_i) = Q_i^f (v_{i,K} - v_r)^2 + \sum_{k=0}^{K} Q_i (v_{i,k} - v_r)^2 + (u_{i,k} - u_i^r)^\top R_i (u_{i,k} - u_i^r), \quad (94)$$

where v^r is a reference speed, and u_i^r is an input which maintains v^r. Here, the weights are selected as $Q_i = 1/(v_i^r)^2$ and $R_i = \mathrm{diag}((1/T_{i,m}^{\max})^2, 1/F_i^{b,\max})^2)$, and Q_i^f is the cost-to-go associated with the LQR controller computed with Q_i, R_i and the linearization of (93b) around v_i^r.

The vehicles are initialized randomly between 80 and 120 meters before the intersection, with $v_{i,0} = v_r = 70$ km/h. The initial solution candidate $w_i^{[0]}, T_i^{[0]}$ is the solution to (18), where all vehicles drive at v_r for $k = 0, \dots, K$, $\lambda^{[0]} = 0$, $\mu^{[0]} = s^{[0]} = \mathbf{1}$ and $\tau^{[0]} = 1$.

The development of $||r(z)^{[k]}||_2$ and $\tau^{[k]}$ is shown in Fig. 6a, and the step sizes used is shown in Fig. 6b. Note that the increases $||r(z)||$ that follows decreases in τ, is a consequence of the somewhat rudimentary IP method deployed.

To illustrate how the algorithm progresses in the primal variables, the velocity profiles of one vehicle is provided in Fig. 6c. Note that the final 15 iterates are virtually indistinguishable from each other. In a practical context, little would likely be lost by stopping after iteration 16.

For illustration, the sparsity-pattern of M is given in Fig. 7. The size is 25832×25832, where M_v is of size 24176×24176, M_{l_j} of size 404×404 and M_T of size 40×40. Besides evaluating the involved functions and their derivatives, the main computational effort is therefore the factorization of the vehicle blocks M_i, roughly sized 2010×2010 [2], and the lane blocks M_{l_j}, sized 404×404. Since the factors for all M_i's can be computed in parallel between the vehicles (Line 2 in Algorithm 1), and the Γ_j's can be factorized in parallel between the lanes (Line 3), the computational time required to solve the KKT system $t^{\mathrm{direction}}$ will roughly be

$$t^{\mathrm{direction}} \approx \max_{i \in \mathcal{N}} (\mathrm{timeToFactorize}(M_i)) + \max_{j \in \mathcal{N}_L} (\mathrm{timeToFactorize}(\Gamma_j)) \quad (95)$$

However, computational time will likely not dominate the time it takes to find a solution. Following the lessons learned from the experiments reported in [17], the time required to communicate between the vehicles, lane centers and intersection center is likely larger. In the next section, we analyze the communication requirements, and discuss some modifications to the scheme which decrease both the number of transmissions and the amount of data communicated.

VIII. Communication Requirements

In this section, we discuss the communication requirements of Algorithm 3. We first analyze the data flow between the vehicles, lane centers and intersection center in Section VIII-A, and demonstrate that the proposed algorithm requires an unrealistic data exchange. We discuss how the requirements can be reduced in Sections VIII-B and VIII-C.

A. Analysis of Communication Requirements

Most data is exchanged during solution of the KKT-system in Algorithm 1 and the selection of the step-size in Algorithm 2. Descriptions of the data involved as well as the number of floats communicated are summarized in Table I.

Most often $K \gg n_{T_i}$, whereby most communication occurs during Line 2 of Algorithm 1 when the vehicle sends $\mathcal{D}_i^l$. Besides the communication between the lane-centers and the intersection center and an initial round of communication where the initial guess $z_c^{[0]}$ is sent to the vehicles, the communication required for the remaining parts (i.e., the indication of a new iteration, the current barrier parameter value, termination of line-search or algorithm completion) consists of single floats and logicals. As illustrated in Fig. 8, these can therefore be sent together with the search-direction and step-size results. In particular, we highlight in the figure where the various components of $||r(z^{[k]})||$ can be passed to the intersection center, for use in termination criteria and barrier parameter update rule.

Communication in the Example: As summarized in Table I, the most data is sent between the vehicles and their lane centers during the first part of the search-direction computation (Line 2 of Algorithm 1). In the example studied above $K = 100$ whereby each vehicle is required to send more than 5000 floats *per iterate* (more than 320000 bits) to their respective lane leader. Even if all vehicles can communicate in parallel the physical transmission will take at least 58.7 ms using the 802.11p protocol, which is the current standard for vehicular communications[3]. During 33 iterations, at least 1.94 would be spent communicating to construct Δz, which is much too high for practical applications. In the next subsection, we discuss how the data exchange can be reduced.

B. Reduction of Data exchange per iterate

The main issue is the K^2 growth in the number of communicated floats on Line 2 of Algorithm 1. This is comes from the use of second order information, and that the RECA constraints are enforced at every time instant k. One obvious remedy is therefore to reduce the time horizon length K significantly. However, long horizons K are desirable, and an alternative approach is needed.

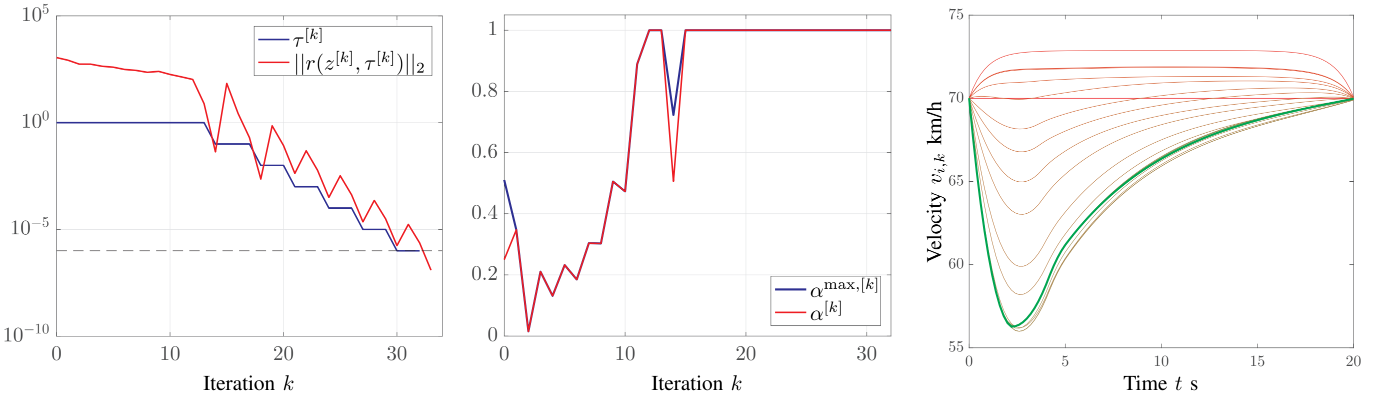
To this end, we propose to replace all RECA constraints

$$p_{i,k} + \delta_{ij} \leq p_{j,k}(t), \quad k \in \mathcal{I}_K \quad (96)$$

with constraints on the form

[2] This implementation only includes the elements of T_i needed for the SICA constraints. The size of T_i therefore vary between different i.

[3] The time per bit is computed using the formula $50 + 8\mathrm{ceil}((n_{\text{data bits}} + 22)/48)\ \mu s$, reported in [27]. Double precision is assumed.



(a) Progress Metrics. Dashed line is tolerance.

(b) Heterogeneous. The velocity profile of the heavy vehicle is drawn in bold.

(c) Velocity profiles for a vehicle corresponding to the solution candidates $z_i^{[k]}$.

Fig. 6: Algorithm progress from a 12-Vehicle Example. In (c), the initial guess is drawn in red and the optimal solution in thick green, with the solution at intermediate iterates ranging between red and green.

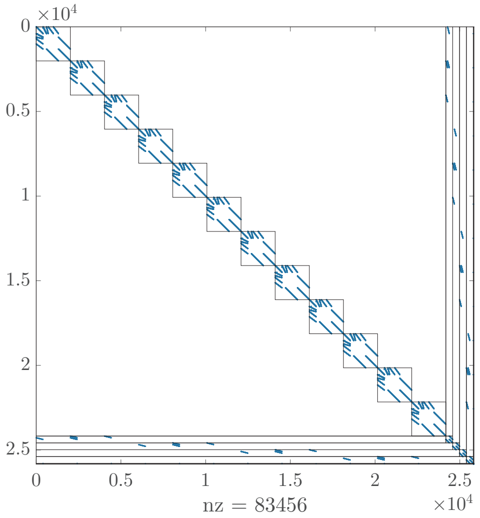


Fig. 7: KKT Matrix $M(z)$ from a 12-vehicle scenario. The large upper left hand block is M_v, consisting of the sub-blocks M_i, $i \in \mathcal{N}$. The smaller blocks in the lower right corner are M_{l_j}, while M_T is barely visible. The lines demarcates the sections of M_{ca} and M_{cv} associated with the RECA constraints on each lane and (barely visible) the SICA constraints

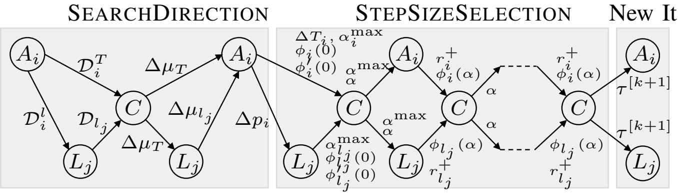


Fig. 8: Illustration of the communication flows in the problem. The horizontal direction indicates sequence whereas the vertical differentiates the vehicle, lane and intersection levels. Here, $\mathcal{D}_{l_j}$ collects $\Lambda_j\Gamma_j^{-1}\Lambda_j^\top$ and $\Lambda_j\Gamma_j^{-1}\gamma_j$, and with slight abuse of notation we write $\phi(\alpha) = \phi(y + \alpha\Delta y, s + \alpha\Delta s)$ and similarly for $\phi'(\alpha)$, and denote $r^+ = r(z + \alpha\Delta z)$.

$$p_{i,k} + \delta_{ij}/2 \le B_{ij}(k,\theta_{ij}) \qquad k \in \mathcal{I}_K \tag{97a}$$
$$B_{ij}(k,\theta_{ij}) + \delta_{ij}/2 \le p_{j,k} \qquad k \in \mathcal{I}_K, \tag{97b}$$

where $B_{ij}(k,\theta_{ij})$ is a function of k, parametrized with $\theta_{ij} \in \mathbb{R}^q$, and introduce θ_{ij}, $\forall(i,j) \in \mathcal{C}_R$ as additional decision variables in the fixed order problem (7). Rather than enforcing the RECA constraints directly using p_i and p_j, (97) requires that $B_{ij}(k,\theta_{ij})$ is between $p_{i,k}$ and $p_{j,k}$ at all k, whereby RECA is ensured by selection of the *coupling parameter* θ_{ij}. This circumvents the K^2 growth in the number of floats communicated, and enables practical schemes which relies on a more realistic data exchange that scales as q^2. The price payed is conservativeness and sub-optimality, as the set of feasible trajectories is reduced when $q < K$.

The parametrized RECA coupling can be included in the distributed scheme in two different ways that give the same solution. In the following discussion, we notation as shown in (9), where $\theta_{i,f}$ and $\theta_{i,r}$ are coupling parameters of vehicle i's forward (97a) and rearward (97b) facing RECA conditions, respectively. That is, if vehicle j is in front of vehicle i, we have that $\theta_{i,f} = \theta_{j,r} = \theta_{ij}$. Similarly, we let $B_{i,f}(\theta_{i,f})$ and $B_{i,r}(\theta_{i,r})$ collect the function $B_{ij}(k,\theta_{ij})$ for $\forall k \in \mathcal{I}_K$ in the forward and rearward facing constraints for vehicle i, respectively, and denote the corresponding multipliers and slacks $(\nu_{i,f}, s_i^f)$ and $(\nu_{i,r}, s_i^r)$. We also let $\theta_i = (\theta_{i,f}, \theta_{j,r})$, $B_i(\theta_i) = (B_{i,f}(\theta_{i,f}), B_{i,r}(\theta_{i,r}))$, $\nu_i = (\nu_{i,f}, \nu_{i,f})$ and $s_i^{\text{RECA}} = (s_{i,f}, s_{i,r})$ for each vehicle, collect θ_{ij} for all vehicle pairs (i,j) on lane j in θ_{l_j}, and collect θ_{ij} for all $(i,j) \in \mathcal{C}_R$ in θ.

The "Primal" Approach: The first alternative is to handle the coupling parameters θ_{ij} at the lane centers, and constraints (97a), (97b) on-board the vehicles. The lane center variables are in this case $z_{l_j} = \theta_{l_j}$, so that $r_{l_j} = \nabla_{\theta_{l_j}}\mathcal{L} = \sum_{i\in\mathcal{N}_{L_j}} \nabla_{\theta_{l_j}} B_i\nu_i$, and the vehicle variables z_i include $(\nu_i, s_i^{\text{RECA}})$. The information sent to the lane center is

$$\nabla_{\theta_i} B_i \frac{\partial z_i}{\partial \nu_i}^\top M_i^{-1} \frac{\partial z_i}{\partial \nu_i} \nabla_{\theta_i} B_i^\top, \qquad \text{size } (2q)^2, \tag{98a}$$
$$\nabla_{\theta_i} B_i \frac{\partial z_i}{\partial \nu_i}^\top M_i^{-1} \frac{\partial z_i}{\partial T_i}, \qquad \text{size } 2qT_i, \tag{98b}$$
$$\nabla_{\theta_i} B_i \nu_i, \qquad \text{size } 2q. \tag{98c}$$

Assuming that both $\theta_{i,f}$ and $\theta_{i,r}$ are of the same size q, this amounts to $2q^2 + (3 + 2n_{T_i})q$ floats on Line 2 of Algorithm 1. Moreover, the information sent from the lane center to a vehicle on Line 5 of Algorithm 1 consists of $\Delta\theta_i$

Link	Location	Data per Communication Round	#Floats
		SEARCHDIRECTION	
A_i to Lj	A.1,L.2	$\underbrace{\frac{\partial z_i}{\partial p_i}^{\top} M_i^{-1} \frac{\partial z_i}{\partial p_i}}_{(K+1)^2}, \underbrace{\frac{\partial z_i}{\partial T_i}^{\top} M_i^{-1} \frac{\partial z_i}{\partial p_i}}_{n_{T_i}(K+1)}, \underbrace{\frac{\partial z_i}{\partial p_i}^{\top} M_i^{-1}(r_i + \nabla_{z_i}\mathcal{L}_c)}_{K+1}, \underbrace{p_i}_{K+1}$	$\frac{1}{2}K^2 + \left(n_{T_i} + \frac{9}{2}\right)K + 3 + n_{T_i}$
A_i to C	A.1, L.2	$\underbrace{\frac{\partial z_i}{\partial T_i}^{\top} M_i^{-1} \frac{\partial z_i}{\partial T_i}}_{n_{T_i}^2}, \underbrace{\frac{\partial z_i}{\partial T_i}^{\top} M_i^{-1}(r_i + \nabla_{z_i}\mathcal{L}_c)}_{n_{T_i}}, \underbrace{T_i}_{n_{T_i}}$	$\frac{1}{2}n_{T_i}^2 + \frac{5}{2}n_{T_i}$
L_j to C	A.1, L.3	$\underbrace{\Lambda_j \Gamma_j^{-1} \Lambda_j^{\top}}_{n_{T_j^l}^2}, \underbrace{\Lambda_j \Gamma_j^{-1} \gamma_j}_{n_{T_j^l}}$	$\frac{1}{2}n_{T_j^l}^2 + \frac{3}{2}n_{T_j^l}$
C to L_j	A.1, L.4	$\underbrace{\Delta\mu_{T_j^l}}_{n_{T_j^l}}$	$n_{T_j^l}$
C to A_i	A.1, L.4	$\underbrace{\Delta\mu_{T,i}}_{n_{T_i}}$	n_{T_i}
L_j to A_i	A.1, L.5	$\underbrace{\Delta\mu_{i,f}}_{K+1}, \underbrace{\Delta\mu_{i,r}}_{K+1}$	$2K+2$
		STEPSIZESELECTION	
A_i to L_j	A.2, L.2	$\underbrace{\Delta p_i}_{K+1}$	$K+1$
A_i to C	A.2, L.2	$\underbrace{\Delta T_i}_{n_{T_i}}, \alpha_i^{\max}, \phi_i(y_i, s_i), \phi_i'(y_i, s_i)$	$n_{T_i} + 3$
L_j to C	A.2, L.3	$\alpha_{l_j}^{\max}, \phi_{l_j}(\bar{p}_j, s_{l_j}), \phi_{l_j}'(\bar{p}_j, s_{l_j})$	3
C to A_i, L_j	A.2, L.6/15	$\alpha, \alpha^{\max}$	1
A_i to C	A.2, L.8	$\phi_i(y_i + \alpha\Delta y_i, s_i + \alpha\Delta s_i)$	1
L_j to C	A.2, L.9	$\phi_{l_j}(\bar{p}_j + \alpha\Delta\bar{p}_j, s_{l_j} + \alpha\Delta s_{l_j})$	1

TABLE I: Summary of the data in the separate communication instances between the vehicles (A_i), the lane centers (L_j) and the intersection center (C). The location columns states in which Algorithm (A) and Line (L) the communication takes place. The numbers under the braces denotes the size of the matrices and vectors involved. Here, n_{T_i} is the number of time variables of vehicle i, all assumed to be part of SICA couplings. T_j^l collects all time variables for vehicles on lane j, $n_{T_j^l} = \text{sizeof}(T_j^l)$ and $\Delta\mu_{T_j^l}$ is the update direction for the Lagrange multipliers corresponding to the SICA constraints which involve T_j^l. The three final rows consists of the line-search iteration, and can consequently be performed more than once.

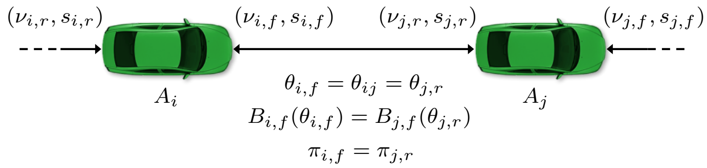


Fig. 9: Illustration of the relationship between the variables introduced to reduce the communication demands.

($2q$ floats). Since p_i and p_j only are coupled indirectly, the term $\frac{\partial z_i}{\partial p_i}(\mu_{i,r} - \mu_{i,f})$ is not present in $\nabla_{z_i}\mathcal{L}_c$, and $\Delta\mu_{i,r}$ or $\Delta\mu_{i,f}$ (c.f. (41)) is not sent by the lane-centers.

The "Dual" Approach: The second alternative is to consider $\theta_{i,f}$ and $\theta_{j,r}$ as separate variables, and couple the vehicles through the constraint

$$\theta_{i,r} - \theta_{j,f} = 0. \tag{99}$$

In this approach, $(\theta_i, \nu_i, s_i^{\text{RECA}})$ is included in the vehicle variables z_i while $z_{l_j} = \pi_{l_j}$ collects the Lagrange multipliers π_{ij} of the constraints (99), for all couplings on lane j. Correspondingly, $r_{l_j}(z_{v(j)}, z_{l_j})$ consists of $\theta_{i,r} - \theta_{j,f}$ for all couplings on lane j, and the information sent from vehicle i on Line 5 of Algorithm 1 is

$$\frac{\partial z_i}{\partial \theta_i}^{\top} M_i^{-1} \frac{\partial z_i}{\partial \theta_i}, \qquad \text{size } (2q)^2, \tag{100a}$$

$$\frac{\partial z_i}{\partial \theta_i}^{\top} M_i^{-1} \frac{\partial z_i}{\partial T_i}, \qquad \text{size } 2qT_i, \tag{100b}$$

$$\theta_i \qquad \text{size } 2q. \tag{100c}$$

i.e., the same amount of data as the Primal approach. Moreover, denoting the multiplier associated with vehicle i's forward and rearward-facing couplings as $\pi_{i,f}$ and $\pi_{i,r}$, respectively, we have

$$\nabla_{z_i}\mathcal{L}_c(z) = \frac{\partial z_i}{\partial \theta_{i,f}}\pi_{i,f} - \frac{\partial z_i}{\partial \theta_{i,r}}\pi_{i,r} + \frac{\partial z_i}{\partial T_i}\mu_{i,T}. \tag{101}$$

Consequently, the lane center-to-vehicle communication on Line 5 of Algorithm 1 consists of $\Delta\pi_{i,f}$, $\Delta\pi_{i,r}$, i.e. $2q$ floats.

Since both approaches have the same communication footprint, they are both useful alternatives. However, when $B_{ij}(k, \theta_{ij})$ is nonlinear in θ_{ij} terms appear off the main block-diagonal in $\nabla^2_{(y,\theta)}\mathcal{L}$. Due to this, positive definiteness of $\nabla_y\mathcal{L}$ can not be ensured by simply modifying $\nabla^2_{y_i}\mathcal{L}_i$ (c.f. the discussion in Section VI-C). In Dual approach on the other

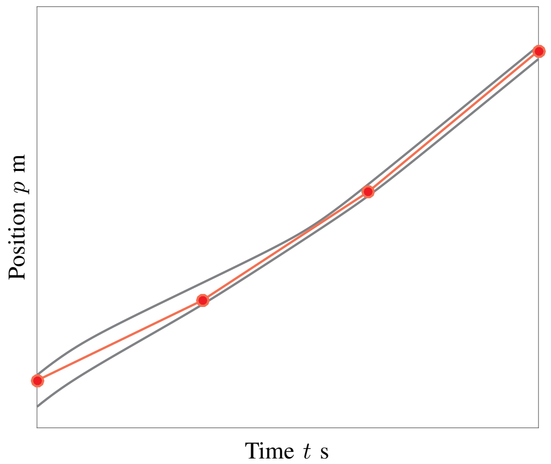


Fig. 10: Illustration with $B_{ij}(k, \theta_{ij})$ piece-wise linear with three linear segments. The trajectories of two vehicles are drawn in gray and the parameterized function in red, with the round markers being the parameters.

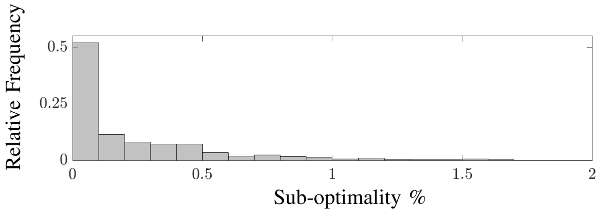


Fig. 11: Distribution of the suboptimality resulting from the use of approximate RECA constraints with a piece-wise linear $B_{ij}(k, \theta_{ij})$

hand, all primal variables are at the vehicle level, and block-wise regularization is possible.

Evaluation and Example: By selecting θ_{ij} such that q is small, significant reductions in the amount of data communicated are achieved at the cost of sub-optimality. To evaluate the cost-benefit trade-off we consider the case shown in Fig. 10, where $B_{ij}(k, \theta_{ij})$ is the piece-wise linear function

$$B_{ij}(k,\theta_{ij}) = \begin{cases} \theta_{ij}^{(1)} + \frac{\theta_{ij}^{(2)}-\theta_{ij}^{(1)}}{\lfloor K/3\rfloor}k & k\in[0,\lfloor K/3\rfloor] \\ \theta_{ij}^{(2)} + \frac{\theta_{ij}^{(3)}-\theta_{ij}^{(2)}}{\lfloor K/3\rfloor}(k-\lfloor K/3\rfloor) & k\in[\lfloor K/3\rfloor+1, 2\lfloor K/3\rfloor] \\ \theta_{ij}^{(3)} + \frac{\theta_{ij}^{(4)}-\theta_{ij}^{(1)}}{\lceil K/3\rceil}(k-2\lfloor K/3\rfloor) & k\in[2\lfloor K/3\rfloor+1, K+1] \end{cases} \tag{102}$$

where the superscript on θ_{ij} indicates the element (i.e. $q = 4$). When $n_{T_i} = 4$, no more than 60 floats are sent from a vehicle to the lane-center on Line 5 of Algorithm 1, which theoretically will take at least 0.7 ms (see footnote on Page 10), a reduction of almost 99 %.

To assess the sub-optimality induced, we evaluated 500 scenarios with 4 vehicles per lane (16 in total), using the models and objective functions of Section VII-A. In all scenarios the vehicles had an initial speed of 70 km/h but were initialized at randomly drawn distances between 50 and 150 meters from the intersection and the crossing order was computed with the heuristic of [20]. As Fig. 11 demonstrates, the sub-optimality induced by the parametrized constraints is below 0.1 % in more than half of the cases, with a few worst case instances just over 1%. The small impact is illustrated in Fig. 12, which shows the difference in the optimal velocity profiles for a scenario corresponding to the median sub-optimality (at 0.091%). Interestingly, the difference between the optimal control commands at $k = 0$ in the two solutions is not greater than about 0.013 % of the input range for any vehicle in this scenario. It is worth noting that this would be below the quantization error of many actuators, an indication that the difference might not be noticed in practice. Finally, we emphasize that these scenarios are "hard" in the sense that they mandates harsh accelerations for some vehicles. Even though such scenarios are rare, one can use a more "flexible" parametrization of B_{ij} to reduce sub-optimality, e.g. by including additional linear segments or (piece-wise) polynomials.

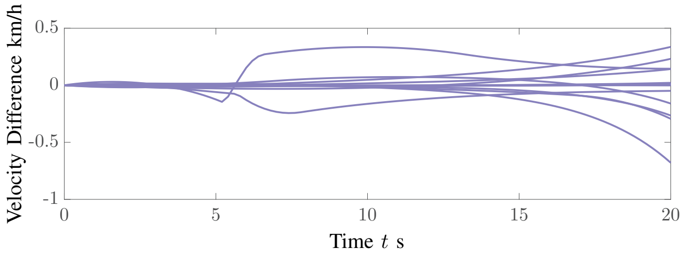


Fig. 12: Difference between the optimal velocity profiles and those obtained using the parametrized RECA constraints (97) for a 16 vehicle scenario.

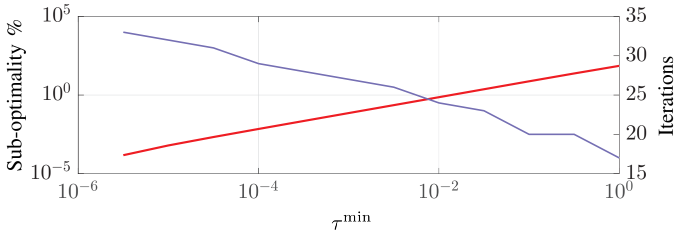


Fig. 13: Sub-optimality (red) and number of iterations (blue) required to reach $||r(z)|| \leq \varepsilon$ for different value of $\tau^{\min}$.

C. Reduction of the number of communication rounds

Algorithm 3 is rudimentary and could be augmented in several ways to converge in fewer iterations. For instance, instead of the simplistic update rule for τ, one can employ adaptive strategies or a Predictor-Corrector strategy [25].

Another approach is to prevent τ from becoming too small but otherwise solve the problem to a sufficiently high accuracy. In this case, all equality constraints are satisfied to the set tolerance, and the inequality constraints are satisfied with a margin. While convergence to the desired tolerance occurs in fewer iterations, sub-optimality is introduced. However, as we remarked in the discussion around Fig. 6c, practically acceptable solutions can be obtained well before τ comes close to relevant tolerances for $||r(z)||$. It is therefore expected that the practical implications of using $\tau^{\min}$ significantly larger than the residual tolerance ε is small. As an example, Fig. 13 shows results from the scenario considered in Section VII-A, where τ is prevented from being smaller than $\tau^{\min}$, for τmin between 1 and 10^{-6}, with $\varepsilon = 10^{-6}$ for all cases. The sub-optimality induced and the number iterations required to reach

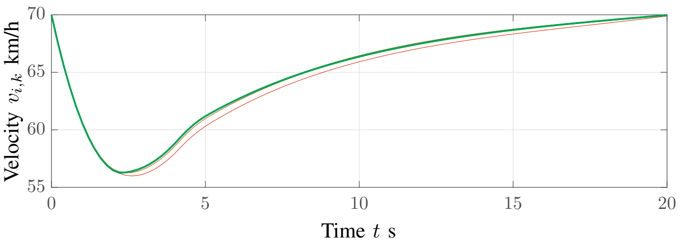


Fig. 14: Velocity profiles for a vehicle corresponding to the solution satisfying $||r(z)||_2 \leq 10^{-6}$ for different values of $\tau^{\min}$. For the red trajectory $\tau^{\min} = 1$ and for the green trajectory $\tau^{\min} = 10^{-6}$. Hues between red and green correspond to intermediate values.

$||r(z)|| \leq \varepsilon$ is shown in Fig. 13. Note for instance that 23 iterations, are required for $\tau^{\min} = 10^{-2}$, compared to 33 in case of $\tau^{\min} = 10^{-6}$, at the expense of less than 1 % suboptimality. The optimal velocity profiles for one vehicle for the different values of $\tau^{\min}$ is shown in Fig. 14 (c.f. Fig. 6c). We emphasize that the difference with respect to the optimal solution is small enough to be practically irrelevant for all but the highest value of $\tau^{\min}$.

IX. Conclusion

In this paper we presented a Primal-Dual Interior Point algorithm for the optimal coordination of automated vehicles at intersections under a fixed crossing order. The algorithm is motivated by deficiencies in earlier work, and enables inclusion of complicating rear-end collision avoidance constraints. We showed that the problem is structured so that the KKT-system can be solved in steps, where most operations are parallelized and solved separately for all vehicles and for all lanes. We demonstrated that step-size selection through backtracking on a merit function can be distributed under the same pattern. To reduce the data exchange, we proposed a parametrized and slightly conservative re-formulation of the rear-end collision avoidance constraints, and demonstrated its merits through randomized evaluation.

We are currently investigating formulations of the coordination problem that allows removal of the restrictive assumption of full CAV penetration. We also aim at extending our approach to scenarios with several connected intersections.

References

[1] C. Englund, L. Chen, J. Ploeg, E. Semsar-Kazerooni, A. Voronov, H. H. Bengtsson, and J. Didoff, “The grand cooperative driving challenge 2016: boosting the introduction of cooperative automated vehicles,” *IEEE Wireless Communications*, vol. 23, no. 4, pp. 146–152, August 2016.

[2] J. Rios-Torres and A. A. Malikopoulos, “A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 1066–1077, 2017.

[3] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymeersch, “Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation,” *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 74–84, Nov 2016.

[4] K. Dresner and P. Stone, “A Multiagent Approach to Autonomous Intersection Management,” *Journal of Artificial Intelligence Research*, vol. 31, no. 1, pp. 591–656, Mar. 2008.

[5] H. Kowshik, D. Caveney, and P. R. Kumar, “Provable systemwide safety in intelligent intersections,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 3, pp. 804–818, March 2011.

[6] J. Lee and B. Park, “Development and evaluation of a cooperative vehicle intersection control algorithm under the connected vehicles environment,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 81–90, March 2012.

[7] K. Kim and P. R. Kumar, “An mpc-based approach to provable system-wide safety and liveness of autonomous ground traffic,” *IEEE Transactions on Automatic Control*, vol. 59, no. 12, pp. 3341–3356, Dec 2014.

[8] A. Katriniok, P. Kleibaum, and M. Josevski, “Distributed model predictive control for intersection automation using a parallelized optimization approach,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 5940 – 5946, 2017, 20th IFAC World Congress.

[9] A. Britzelmeier and M. Gerdts, “Non-linear model predictive control of connected, automatic cars in a road network using optimal control methods,” *IFAC-PapersOnLine*, vol. 51, no. 2, pp. 168 – 173, 2018, 9th Vienna International Conference on Mathematical Modelling.

[10] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang, “A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections,” *Automatica*, vol. 93, pp. 244 – 256, 2018.

[11] P. Tallapragada and J. Cortés, “Coordinated intersection traffic management,” *IFAC-PapersOnLine*, vol. 48, no. 22, pp. 233 – 239, 2015, 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015.

[12] L. Riegger, M. Carlander, N. Lidander, N. Murgovski, and J. Sjöberg, “Centralized mpc for autonomous intersection crossing,” in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1372–1377.

[13] J. Shi, Y. Zheng, Y. Jiang, M. Zanon, R. Hult, and B. Houskal, “Distributed control algorithm for vehicle coordination at traffic intersections,” in *2018 European Control Conference (ECC)*, June 2018, pp. 1166–1171.

[14] M. Kneissl, A. Molin, H. Esen, and S. Hirche, “A feasible mpc-based negotiation algorithm for automated intersection crossing *,” in *European Control Conference (ECC)*, 06 2018, pp. 1282–1288.

[15] C. Bali and A. Richards, “Merging vehicles at junctions using mixed-integer model predictive control,” in *European Control Conference (ECC)*, 06 2018, pp. 1740–1745.

[16] R. Hult, M. Zanon, S. Gros, and P. Falcone, “Primal decomposition of the optimal coordination of vehicles at traffic intersections,” in *2016 IEEE 55th Conference on Decision and Control (CDC)*, Dec 2016, pp. 2567–2573.

[17] R. Hult, M. Zanon, S. Gros, and P. Falcone, “Optimal coordination of automated vehicles at intersections: Theory and experiments,” *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2018.

[18] ——, “Energy-optimal coordination of autonomous vehicles at intersections,” in *2018 European Control Conference (ECC)*, June 2018, pp. 602–607.

[19] R. Hult, M. Zanon, S. Gros, and P. Falcone, “Optimal coordination of automated vehicles at intersections with turns,” in *To be presented at the European Control Conference (ECC)*, 2019.

[20] R. Hult, M. Zanon, S. Gras, and P. Falcone, “An miqp-based heuristic for optimal coordination of vehicles at intersections,” in *2018 IEEE Conference on Decision and Control (CDC)*, Dec 2018, pp. 2783–2790.

[21] R. Hult, M. Zanon, S. Gros, H. Wymeersch, and P. Falcone, “Optimization-based coordination of connected, automated vehicles at intersections,” *submitted to Vehicle System Dynamics*, 2019.

[22] S. K. Pakazad, A. Hansson, M. S. Andersen, and I. Nielsen, “Distributed primal-dual interior-point methods for solving tree-structured coupled convex problems using message-passing,” *Optimization Methods and Software*, vol. 32, no. 3, pp. 401–435, 2017.

[23] J. Gondzio and A. Grothey, “Parallel interior-point solver for structured quadratic programs: Application to financial planning problems,” *Annals of Operations Research*, vol. 152, no. 1, pp. 319–339, Jul 2007.

[24] K. Dresner and P. Stone, “Multiagent traffic management: a reservation-based intersection control mechanism,” in *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.*, July 2004, pp. 530–537.

[25] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. New York, NY, USA: Springer, 2006.

[26] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, “Autogenerating microsecond solvers for nonlinear mpc: A tutorial using acado integrators,” *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 685–704, 2015. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/oca.2152

[27] J. A. Fernandez, K. Borries, L. Cheng, B. V. K. V. Kumar, D. D. Stancil, and F. Bai, “Performance of the 802.11p physical layer in vehicle-to-vehicle environments,” *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 3–14, Jan 2012.