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.**

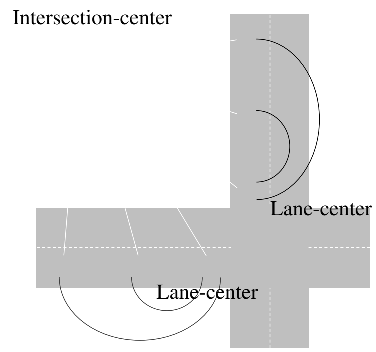


Fig. 1: Illustration of distribution structure

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 in 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]–[14] 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]–[13]. 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.

We have studied the intersection problem in earlier work. In [15], we introduced a Sequential Quadratic Programming (SQP) algorithm based on a primal decomposition of the fixed-order coordination problem, where most computations are parallelized and performed on-board the vehicles. We considered the receding horizon application of the SQP algorithm in [16], and presented experimental results. We considered the extension to non-linear motion models and economic objective functions in [17] and to scenarios with turning vehicles in [18]. In [19], we proposed an OC-based heuristic for crossing order selection, and compared the performance of to traffic-lights and other algorithms in [20].

However, the algorithm in [15] 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 [15], [16], this approach is partly centralized, and relies on central units for some 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.

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. While similar distribution schemes can be found in the literature (see e.g. [21], [22]), the application to the intersection problem is novel.

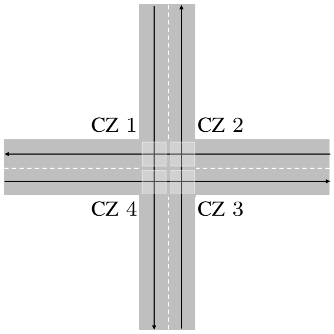


Fig. 2: Illustration of the scenarios considered, Assumption 2 (black lines) and the Conflict Zones (red boxes).

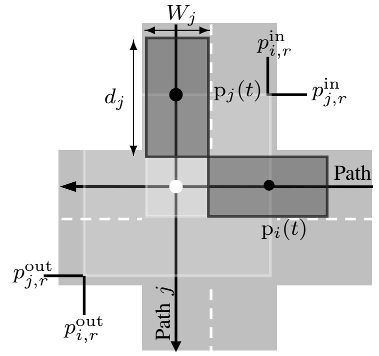


Fig. 3: Illustration of the elements used in the side collision avoidance conditions. d_j and W_j denotes the length and width of the vehicle, respectively.

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 parallelized. In Section IV we show how the KKT system can be solved with computation at the vehicle, lane and intersection-centers. In Section V we show how to select the step-size in a distributed fashion. In Section VI we state a rudimentary practical algorithm and provide a numerical example. In Section VII we analyze the communication requirements and propose an approximate representation of the RECA constraints, using which the data sent per iterate can be reduced. The paper is concluded in Section VIII

II. Optimal Coordination at Intersections

We consider intersection scenarios as shown in Fig. 2, where N vehicles approach an intersection with L lanes, 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

$$\dot{\mathrm{x}}_i(t) = f_i(\mathrm{x}_i(t), \mathrm{u}_i(t)), \tag{1a}$$
$$0 \geq c_i(\mathrm{x}_i(t), \mathrm{u}_i(t)), \tag{1b}$$

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 and c_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 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 through auxiliary variables for the time of entry and departure of each CZ, implicitly defined through

$$\mathrm{p}_i(t_{i,r}^{\text{in}}) = p_{r,i}^{\text{in}}, \qquad \forall r \in \mathcal{Z}_i \tag{2}$$
$$\mathrm{p}_i(t_{i,r}^{\text{out}}) = p_{r,i}^{\text{out}}, \qquad \forall r \in \mathcal{Z}_i \tag{3}$$

where $\mathcal{Z}_i$ collects the indices of the CZ crossed by vehicle i, and $p_{r,i}^{\text{in}}$ and $p_{r,i}^{\text{out}}$ are defined as shown in Fig. 3[1].

A condition for Side Collision Avoidance (SICA) is then

$$t_{r,i}^{\text{out}} \leq t_{r,j}^{\text{in}}, \quad (i, j, r) \in \mathcal{S}, \tag{4}$$

where $\mathcal{S}$ collects all vehicle pairs (i, j) and CZ r where side collisions can occur, and encodes the crossing order.

C. Rear-End Collision Avoidance (RECA)

Due to Assumption 2, rear-end collisions can only occur between two adjacent vehicles on the same lane. We collect all vehicle pairs (i, j) such that i is immediately behind j on lane j in $\mathcal{R}_j$, and state the necessary condition for Rear-End Collision Avoidance (RECA)

$$\mathrm{p}_i(t) + \delta_{iq} \leq \mathrm{p}_q(t), \quad (i, q) \in \mathcal{R}_j, \tag{5}$$

where $\delta_{ij} = d_i/2 + d_q/2$. All but the first and last vehicle on a lane has a forward and rearward-facing RECA constraint.

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

D. Discretization

We employ a direct formulation of the optimal coordination problem, assuming a piece-wise constant parameterization 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

$$x_{i,0} = \hat{x}_{i,0} \tag{6}$$

$$x_{i,k+1} = F_i(x_{i,k}, u_{i,k}, \Delta t), \qquad k = 0, \dots, K-1, \tag{7}$$

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 $u_i = (u_{i,0}, \dots, u_{i,K})$, which we collect as $w_{v_i} = (x_i, u_i)$. Moreover, we express the position $\mathrm{p}_i(t)$ at time t as a function of w_i

$$p_i(t, w_{v_i}) = F_{i,p}(x_{i,k}, u_{i,k}, t - t_k), \ k = \lfloor t/\Delta t \rfloor, \tag{8}$$

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 $t^{\text{in}}_{i,r}$, $t^{\text{in}}_{i,r}$ are well-defined, continuous functions of w_i through

$$p_i(t^{\text{in}}_{i,r}, w_{v_i}) = p^{\text{in}}_{r,i}, \qquad \forall r \in \mathcal{Z}_i, \tag{9}$$

$$p_i(t^{\text{out}}_{i,r}, w_{v_i}) = p^{\text{out}}_{r,i}, \qquad \forall r \in \mathcal{Z}_i, \tag{10}$$

when (6) holds.

Finally, we only consider enforcement of constraints (1b) and the RECA constraints (5) at times t_k so that

$$c_i(x_{i,k}, u_{i,k}) \leq 0, \qquad k = 0, \dots, K, \tag{11}$$

$$p_{i,k} + \delta_{iq} \leq p_{q,k} \quad k = 0, \dots, K, \quad (i, q) \in \mathcal{R}_j. \tag{12}$$

E. Optimal Control Formulation

We define

- $\mathcal{N} = \{1, \dots, N\}$ The set of all vehicles
- $\mathcal{N}_L = \{1, \dots, L\}$ The set of all lanes
- $\mathcal{N}_{l_j}$ The set of all vehicles on lane j

and let

- T_{v_i} collect $T_{r,i} = (t^{\text{in}}_{r,i}, t^{\text{out}}_{r,i})$, $\forall r \in \mathcal{Z}_i$
- $T = (T_{v_1}, \dots, T_{v_N})$
- $y_{v_i} = (w_{v_i}, T_{v_i})$
- $y = (y_{v_i}, \dots, y_{v_N})$
- $p_{v_i} = (p_{i,0}, \dots, p_{i,K})$
- p_{l_j} collect p_{v_i}, $\forall i \in \mathcal{N}_{l_j}$
- p collect p_{v_i}, $\forall i \in \mathcal{N}$
- $g_{v_i}(y_i) = 0$ collect constraints (6),(9)
- $h_{v_i}(y_i) \leq 0$ collect constraints (11)
- $h_{l_j}(p_{l_j}) \leq 0$ collect constraints (12)
- $h_T(T) \leq 0$ collect constraints (4).

The objective functions consider are on the form

$$J_{v_i}(y_{v_i}) = V^{\text{f}}_i(x_{i,N}) + \sum_{k=0}^{K-1} \ell_i(x_{i,k}, u_{i,k}), \tag{13}$$

with the continuously differentiable terminal cost $V^{\text{f}}_i : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ and stage cost $\ell_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \mapsto \mathbb{R}$.

The problem of optimal intersection coordination is then

$$\min_{y} \quad \sum_{i=1}^{N} J_{v_i}(y_{v_i}) \tag{14a}$$

$$\text{s.t.} \quad g_{v_i}(y_i) = 0, \qquad i \in \mathcal{N} \tag{14b}$$

$$h_{v_i}(y_i) = 0, \qquad i \in \mathcal{N} \tag{14c}$$

$$h_{l_j}(p_{l_j}) \leq 0, \qquad j \in \mathcal{N}_L, \tag{14d}$$

$$h_T(T) \leq 0. \tag{14e}$$

Note that in its "full" form, problem (14) includes selection of the crossing order $\mathcal{S}$. This is a difficult combinatorial problem, and finding exact solutions is in general intractable. In this paper, we assume that the crossing order $\mathcal{S}$ is provided by a heuristic. Problem (14) is consequently denoted the *fixed-order* problem, and is a continuous Nonlinear Program (NLP).

Remark 1. *We present Problem* (14) *without turning vehicles for simplicity, but note that the same formalism [18]. Moreover, both* (4) *and* (5) *would in practice be defined with state-dependent margins 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-dual variables on the perturbed first-order necessary conditions for optimality (FONC). By simultaneously driving the perturbation to zero, a sequence of primal-dual solution candidates results. This sequence converge to local a minimum of the NLP under some conditions [23].

A. A PDIP formulation of the fixed order problem

Collecting (14a) in $J(w)$, (14b) in $g(y)$ and (14c)-(14e) in $h(y)$, the perturbed FONC of Problem (14) are

$$\nabla_y \mathcal{L} = 0, \tag{15a}$$

$$g(y) = 0, \tag{15b}$$

$$h(y) + s \leq 0, \tag{15c}$$

$$D(s)\mu - \mathbf{1}\tau = 0, \tag{15d}$$

$$\mu \geq 0, \tag{15e}$$

$$s \geq 0, \tag{15f}$$

Here, s is the slack variable associated with h, $D(s)$ is the matrix with the elements of 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

$$\mathcal{L}(y, \lambda, \mu) = J(w) + \lambda^\top g(y) + \mu^\top h(y), \tag{16}$$

where λ and μ are the Lagrange multipliers associated with constraints g and h respectively. Collecting y, λ, μ, s in z, we write (15a)-(15d) as $r_\tau(z) = 0$.

Starting from $z^{[0]}$ strictly satisfying (15e),(15f), the sequence of primal-dual solution candidates is generated through the Newton-iteration

$$z^{[k+1]} = z^{[k]} + \alpha^{[k]} \Delta z^{[k]}, \tag{17}$$

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

$$M(z^{[k]})\Delta z^{[k]} = -r_{\tau^{[k]}}(z^{[k]}). \tag{18}$$

where the matrix $M(z)$ is known as the KKT-matrix, which is constructed from $\frac{\partial r_\tau}{\partial z}$, evaluated at $z^{[k]}$. The step-size is selected such that the updated solution candidate $z^{[k+1]}$ strictly satisfies (15e), (15f) and provides sufficient decrease on the merit function $\phi(z)$ and strictly satisfies (15e),(15f). Finally, as the PDIP algorithm progresses, the barrier parameter is $\tau^{[k]}$ is updated, i.e.,

$$\tau^{[k+1]} = \xi(\tau^{[k]}, z^{[k]}), \tag{19}$$

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 (18) is solved using a standard linear algebra routine. The information needed to assemble $M(z^{[k]})$ and $r_{\tau^{[k]}}(z^{[k]})$ must thus be made available centrally before the search-direction $\Delta z^{[k]}$ can be found. However, the structure of Problem (14) admits (18) 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. Solving the KKT-system

In this section, we construct $r_{\tau^{[k]}}(z^{[k]})$ and $M(z^{[k]})$ for Problem (14). For brevity, we omit the iteration index and dependence on τ in what follows, and only include the arguments of functions to highlight specific dependencies.

We first note that the problem consists of three levels.

1) *The Vehicle-Level*:, consisting of the objective function (14a)and constraints (14b),(14c). The functions can be evaluated independently for each vehicle.
2) *The Lane-Level*: The RECA constraints (14d) couple vehicles on the same lane. The function $h_{l_j}(p_{l_j})$ can be evaluated independently for each lane.
3) *The Intersection-Level*: The SICA constraints (14e) couple vehicles on different lanes. The function $h_T(T)$ require information from all vehicles in the scenario.

In the sequel, the use of "Vehicle-level", "Lane-level" and "Intersection-level" refers to the functions as discussed here.

A. Variable Definitions

We use the following notation:

- λ_{v_i} multiplier for constraint (14b) of vehicle i
- μ_{v_i} multiplier for constraint (14c) of vehicle i
- s_{v_i} slack variable for constraint (14c) of vehicle i
- $\mu^{\mathrm{f}}_{v_i}$ multiplier for forward RECA constraint of vehicle i
- $\mu^{\mathrm{r}}_{v_i}$ multiplier for rearward RECA constraint of vehicle i
- $\mu_{T(i)}$ multiplier for SICA constraints of vehicle i.
- μ_{l_j} multiplier for constraint (14d) on lane j. Collects $\mu^{\mathrm{f}}_{v_i}$,$\mu^{\mathrm{r}}_{v_i}$ $\forall i \in \mathcal{N}_{l_j}$
- s_{l_j} slack variable for constraint (14d) on lane j
- $s_l = (s_{l_1}, \dots, s_{l_L})$
- $\mu_l = (s_{\mu_1}, \dots, s_{\mu_L})$
- μ_T multiplier for constraint (14e) on lane j. Collects $\mu_{T(i)}$ $\forall i \in \mathcal{N}$.
- s_T slack variable for constraint (14e) on lane j.

and collect the primal-dual variables as

- $z_{v_i} = (y_{v_i}, \lambda_{v_i}, \mu_{v_i}, s_{v_i})$
- $z_v = (z_{v_1}, \dots, z_{v_N})$
- $z_{l_j} = (\mu_{s_j}, s_{l_j})$
- $z_l = (z_{l_1}, \dots, z_{l_1})$
- $z_T = (\mu_T, s_T)$
- $z = (z_v, z_l, z_T)$

Note that since the vehicles are coupled through the SICA constraints (14e), each element in μ_T is found in both $\mu_{T(i)}$ and $\mu_{T(j)}$ when $(i, j, r) \in \mathcal{S}$. Similarly, due to the RECA constraints (14d), $\mu^{\mathrm{f}}_{v_i} = -\mu^{\mathrm{r}}_{v_q}$ when $(i, q) \in \mathcal{R}_j$.

Additionally, we use the notion $z_{v(j)}$ to collect z_{v_i} $\forall i \in \mathcal{N}_{l_j}$, and denote the lane of vehicle i as $l(i)$. For ease of presentation, we also assume that the vehicles are ordered as they appear on each lane, so that $z_v = (z_{v(1)}, \dots, z_{v(L)})$.

B. KKT-System

Note that all primal-variables y are at the vehicle level of the problem. The elements of (15a) relating to the primal variables of vehicle i, y_{v_i} are

$$\nabla_{y_{v_i}}\mathcal{L} = \nabla_{y_{v_i}} J_{v_i} + \nabla_{y_{v_i}} g_{v_i}\lambda_{v_i} + \nabla_{y_{v_i}} g_{v_i}\mu_{v_i} + \frac{\partial y_{v_i}}{\partial p_{v_i}}(\mu^{\mathrm{r}}_{v_i} - \mu^{\mathrm{f}}_{v_i}) + \frac{\partial y_{v_i}}{\partial T_{v_i}}\frac{\partial h_T}{\partial T_{v_i}}^\top \mu_{T(i)}. \tag{20}$$

Here, $\frac{\partial h_T}{\partial T_{v_i}}$, is a square matrix with 1 and -1 on the main diagonal where the sign of a particular element is determined by $\mathcal{S}$ through (4). We define

$$r_{v_i}(z_{v_i}, \mu^{\mathrm{f}}_{v_i}, \mu^{\mathrm{r}}_{v_i}, \mu_{T(i)}) = \begin{bmatrix} \nabla_{y_i}\mathcal{L}(y_{v_i}, \mu^{\mathrm{f}}_{v_i}, \mu^{\mathrm{r}}_{v_i}, \mu_{T(i)}) \\ g_{v_i}(y_{v_i}) \\ h_{v_i}(y_{v_i}) + s_{v_i} \\ \mu_{v_i} + D(s_{v_i})^{-1}\mathbf{1}\tau \end{bmatrix}, \tag{21}$$

$$r_{l_j}(p_{l_j}, z_{l_j}) = \begin{bmatrix} h_{l_j}(p_{l_j}) + s_{l_j} \\ \mu_{l_j} + D(s_{l_j})^{-1}\mathbf{1}\tau \end{bmatrix}, \tag{22}$$

$$r_l(p, z_l) = (r_{l_1}, \dots, r_{l_L}), \tag{23}$$

$$r_T(T, z_T) = \begin{bmatrix} h_T(T) + s_T \\ \mu_T + D(s_T)^{-1}\mathbf{1}\tau \end{bmatrix}, \tag{24}$$

$$M_{v_i}(z_{v_i}) = \begin{bmatrix} \nabla^2_{y_{v_i}}\mathcal{L}_i & \nabla_{y_{v_i}} g_{v_i} & \nabla_{y_{v_i}} h_{v_i} & \\ \nabla_{y_{v_i}} g_{v_i}^\top & & & \\ \nabla_{y_{v_i}} h_{v_i}^\top & & & I \\ & & I & D(s_{v_i})^{-1}D(\mu_{v_i}) \end{bmatrix}, \tag{25}$$

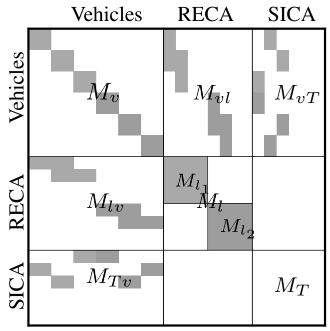


Fig. 4: Illustration of the KKT-matrix of (14). Blue and green differentiate vehicles on different lanes.

$$M_v = \mathrm{blockdiag}(M_{v_1}, \ldots, M_{v_N}), \tag{26}$$

$$M_{l_j} = \begin{bmatrix} 0 & I \\ I & D(s_{l_j})^{-1}D(\mu_{l_j}) \end{bmatrix}, \tag{27}$$

$$M_l = \mathrm{blockdiag}(M_{l_1}, \ldots, M_{l_L}), \tag{28}$$

$$M_T = \begin{bmatrix} 0 & I \\ I & D(s_T)^{-1}D(\mu_T) \end{bmatrix}, \tag{29}$$

and let

$$M_{ab} = \frac{\partial r_a}{\partial z_b}, \tag{30}$$

so that, e.g., $M_{v_iT} = \frac{\partial r_{v_i}}{\partial z_T}$. The KKT-system (18) can then be written

$$\underbrace{\begin{bmatrix} M_v & M_{vl} & M_{vT} \\ M_{lv} & M_l & \\ M_{Tv} & & M_T \end{bmatrix}}_{M(z)} \underbrace{\begin{bmatrix} \Delta z_v \\ \Delta z_l \\ \Delta z_T \end{bmatrix}}_{\Delta z} = - \underbrace{\begin{bmatrix} r_v \\ r_l \\ r_T \end{bmatrix}}_{r(z)}. \tag{31}$$

C. Solving the KKT-system

An illustration of M is given in Fig. 4, and we note the following

- M is symmetric and $M_{lv} = M_{vl}^\top$, $M_{Tv} = M_{vT}^\top$
- The variables z_{v_i} are local to the vehicles, i.e., all functions are separable between the vehicles. Due to this $M_v^{-1} = \mathrm{blockdiag}(M_{v_1}^{-1}, \ldots, M_{v_N}^{-1})$ and M_{lv}, M_{Tv} are constant.
- The RECA constraints only couple vehicle on the same, so that

$$M_{lv} = \mathrm{blockdiag}(M_{l_1}, \ldots, M_{l_L}) \tag{32a}$$

$$M_{lv} = \mathrm{blockdiag}(M_{l_1v(1)}, \ldots, M_{l_Lv(L)}) \tag{32b}$$

where we recall that subscript $v(j)$ collects v_i, $\forall i \in \mathcal{N}_L$.

We show next that the computations involved in solving (31) can be split between the vehicle, lane and intersection levels of the problem.

Proposition 1. *The KKT-system* (31) *can be solved as the following sets of equations*

$$\Theta \Delta z_T = \theta, \tag{33a}$$

$$\Gamma_j \Delta z_{l_j} = \gamma_j - \Lambda_j^\top \Delta z_T, \qquad \forall j \in \mathcal{N}_L, \tag{33b}$$

$$M_{v_i}\Delta z_{v_i} = - r_{v_i} - \begin{bmatrix} M_{v_il} & M_{v_iT} \end{bmatrix} \begin{bmatrix} \Delta z_l \\ \Delta z_T \end{bmatrix}, \quad \forall i \in \mathcal{N}, \tag{33c}$$

where

$$\Theta = \left(M_T - \sum_{i=1}^N M_{Tv_i} M_{v_i}^{-1} M_{v_iT} - \sum_{j=1}^L \Lambda_j \Gamma_j^{-1} \Lambda_j^\top \right), \tag{34a}$$

$$\theta = -r_T + \sum_{i=1}^N M_{Tv_i} M_{v_i}^{-1} r_{v_i} - \sum_{j=1}^L \Lambda_j \Gamma_j^{-1} \gamma_j, \tag{34b}$$

$$\Gamma_j = M_{l_j} - M_{l_jv(j)} M_{v(j)}^{-1} M_{v(j)l_j}, \tag{34c}$$

$$\gamma_j = -r_{l_j} + M_{l_jv(j)} M_{v(j)}^{-1} r_{v(j)}, \tag{34d}$$

$$\Lambda_j = -M_{Tv(j)} M_{v(j)}^{-1} M_{l_jv(j)}^\top. \tag{34e}$$

Proof. Using the Schur complement we have that

$$\begin{bmatrix} \Gamma & \Lambda^\top \\ \Lambda & \Psi \end{bmatrix} \begin{bmatrix} \Delta z_l \\ \Delta z_T \end{bmatrix} = \begin{bmatrix} \gamma \\ \psi \end{bmatrix}, \tag{35}$$

$$M_v \Delta z_v = r_v - \begin{bmatrix} M_{vl} & M_{vT} \end{bmatrix} \begin{bmatrix} \Delta z_l \\ \Delta z_T \end{bmatrix}, \tag{36}$$

where

$$\Gamma = M_l - M_{lv} M_v^{-1} M_{vl}, \tag{37}$$

$$\Lambda = -M_{Tv} M_v^{-1} M_{vl}, \tag{38}$$

$$\Psi = M_T - M_{Tv} M_v^{-1} M_{vT}, \tag{39}$$

$$\gamma = -r_l + M_{lv} M_v^{-1} r_v, \tag{40}$$

$$\psi = -r_T + M_{Tv}^\top M_v^{-1} r_v. \tag{41}$$

A second use of the Schur complement gives that

$$\Theta \Delta z_T = \theta, \tag{42}$$

$$\Gamma \Delta z_l = \gamma - \Lambda^\top \Delta z_T, \tag{43}$$

where

$$\Theta = \Psi - \Lambda \Gamma^{-1} \Lambda^\top, \tag{44}$$

$$\theta = \psi - \Lambda \Gamma^{-1} \gamma. \tag{45}$$

Due to (32) and (28) we have that

$$\Gamma = \mathrm{blockdiag}(\Gamma_1, \ldots, \Gamma_L), \tag{46}$$

$$\Gamma_j = M_{l_j} - M_{l_jv(j)} M_{v(j)}^{-1} M_{v(j)l_j}, \tag{47}$$

$$\gamma = (\gamma_1, \ldots, \gamma_L), \tag{48}$$

$$\gamma_j = -r_{l_j} + M_{l_jv(j)} M_{v(j)}^{-1} r_{v(j)}, \tag{49}$$

which gives

$$\Lambda = \begin{bmatrix} \Lambda_1, & \ldots, & \Lambda_L \end{bmatrix}, \tag{50}$$

$$\Lambda_j = -M_{Tv(j)} M_{v(j)}^{-1} M_{l_jv(j)}^\top, \tag{51}$$

so that

$$\Lambda \Gamma^{-1} \Lambda^\top = \sum_{j=1}^L \Lambda_j \Gamma_j^{-1} \Lambda_j^\top, \quad \Lambda \Gamma^{-1} \gamma = \sum_{j=1}^L \Lambda_j \Gamma_j^{-1} \gamma_j^\top, \tag{52}$$

and (43) can be solved as the L separate sets of equations

$$\Gamma_j \Delta z_{l_j} = \gamma_j - \Lambda_j^\top \Delta z_T, \quad \forall j \in \mathcal{N}_L. \tag{53}$$

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 v_i denotes vehicle i. All vehicles are assumed to hold copies of $\mu^{\mathrm{f}}_{v_i}$, $\mu^{\mathrm{r}}_{v_i}$ and $\mu_{T(i)}$.

1: **procedure** SEARCHDIRECTION(z,τ)
2: $\quad\forall v_i$: Compute $\mathcal{D}_{v_i\to C}$,$\mathcal{D}_{v_i\to l(i)}$, and pass to C and $l(i)$
3: $\quad\forall l_j$: Compute $\mathcal{D}_{l_j\to C}$ and pass to C
4: $\quad C$: Assemble and solve (33a), pass the appropriate parts of $\Delta\mu_T$ to all l_j, and all v_i.
5: $\quad\forall l_j$: Solve (33b) for Δz_{l_j}, using the received components of $\Delta\mu_T$, pass the appropriate parts of $\Delta\mu_{l_j}$ to all v_i on lane j
6: $\quad\forall v_i$: Solve (33c) for Δz_{v_i}, using the received components of $\Delta\mu_T$ and $\Delta\mu_{l_j}$
7: **end procedure**

Similarly, due to (26)

$$M_{Tv}M_v^{-1}M_{vT} = \sum_{i=1}^{N} M_{Tv_i}M_{v_i}^{-1}M_{v_iT}, \tag{54}$$

$$M_{Tv}M_v^{-1}r_v = \sum_{i=1}^{N} M_{Tv_i}M_{v_i}^{-1}r_{v_i}, \tag{55}$$

and (36) can be solved as the N sets of equations

$$M_{v_i}\Delta z_{v_i} = r_{v_i} - \begin{bmatrix} M_{v_il} & M_{v_iT}\end{bmatrix}\begin{bmatrix}\Delta z_l \\ \Delta z_T\end{bmatrix}, \quad \forall i\in\mathcal{N}. \tag{56}$$

□

Proposition 1 gives that (31) can be solved in a dynamic programming-like fashion, with an upwards and a downwards pass through the problem levels. Algorithm 1 summarize this procedure in a setting where the vehicle-level computations are performed on the vehicles, the lane-level computations are performed on lane-centers and the intersection-level computations are performed at the intersection-center (c.f. Fig. 1). Here, $\mathcal{D}_{v_i\to C}$, $\mathcal{D}_{v_i\to l(i)}$ collects the data sent from the vehicle to the lane and intersection-centers, and $\mathcal{D}_{l_j\to C}$ the data sent from a lane-center to the intersection-center, to be detailed in the next sub-section.

Note that this enables parallelization, since Lines 2 and 3 are separable between the vehicles, and Lines 3 and 5 are separable between the lanes. Note also that the factors of matrices M_{v_i}, $\forall i\in\mathcal{N}$ and Γ_j $\forall j\in\mathcal{N}_L$ computed on Lines 2 and 3 can be stored and reused on Lines 6 and 5, respectively.

D. Message-passing between the problem levels

The vehicles need to supply $\mathcal{D}_{v_i\to l(i)}$ to the lane-centers and $\mathcal{D}_{v_i\to C}$ to the intersection-center to enable construction of (33b) and (33a). Similarly, the lane-centers need to supply $\mathcal{D}_{l_j\to C}$ to the intersection-center. These matrices are sparse and contain the following data.

Content of $\mathcal{D}_{l_j\to C}$: The information passed from lane-center j to the intersection-center consists of $\Lambda_j\Gamma_j^{-1}\Lambda_j^\top$, $\Lambda_j\Gamma_j^{-1}\gamma_j$.

Content of $\mathcal{D}_{v_i\to l(i)}$: The lane-centers need to assemble (34c)-(34e) to evaluate $\Lambda_j\Gamma_j^{-1}\Lambda_j^\top$ and $\Lambda_j\Gamma_j^{-1}\gamma_j^\top$. Since the RECA and SICA constraints are separable between the vehicles, we have that

$$M_{l_jv(j)}M_{v(j)}^{-1}M_{l_jv(j)}^\top = \sum_{i\in\mathcal{N}_{l_j}} M_{l_jv_i}M_{v_i}^{-1}M_{v_il_j} \tag{57}$$

$$M_{l_jv(j)}M_{v(j)}^{-1}r_{v(j)} = \sum_{i\in\mathcal{N}_{l_j}} M_{l_jv_i}M_{v_i}^{-1}r_{v_i} \tag{58}$$

$$M_{Tv(j)}M_{v(j)}^{-1}r_{v(j)l_j}^\top = \sum_{i\in\mathcal{N}_{l_j}} M_{Tv_i}M_{v_i}^{-1}r_{v_i} \tag{59}$$

The information sent from vehicle i to its lane-center is therefore $M_{l_jv_i}M_{v_i}^{-1}M_{v_iT}$ (to build Γ_j), $M_{l_jv_i}M_{v_i}^{-1}r_{v_i}$ and p_{v_i} (to build γ_j), $M_{Tv_i}M_{v_i}^{-1}M_{v_il_j}$ (to build Λ_j). Due to (12) and (4), the non-zero elements of M_{Tv_i}, M_{l_j} are in the columns corresponding to p_{v_j},T_{v_i}, so that $\mathcal{D}_{v_i\to l(i)}$ contains

$$\frac{\partial z_{v_i}}{\partial p_{v_i}}^\top M_{v_i}^{-1}\frac{\partial z_{v_i}}{\partial p_{v_i}}, \qquad \frac{\partial z_{v_i}}{\partial T_{v_i}}^\top M_{v_i}^{-1}\frac{\partial z_{v_i}}{\partial p_{v_i}}, \tag{60}$$

$$\frac{\partial z_{v_i}}{\partial p_{v_i}}^\top M_{v_i}^{-1}r_{v_i}, \qquad p_{v_i}. \tag{61}$$

Content of $\mathcal{D}_{v_i\to C}$: The information sent from vehicle i to the intersection-center is $M_{Tv_i}M_{v_i}^{-1}r_{v_i}$ (to build Θ), $M_{Tv_i}M_{v_i}^{-1}M_{v_iT}$ and T_{v_i} (to build θ). Due to (4) the non-zero elements of M_{Tv_i} are in the columns corresponding to T_{v_i}, so that $\mathcal{D}_{v_i\to C}$ contains

$$\frac{\partial z_{v_i}}{\partial T_{v_i}}^\top M_{v_i}^{-1}\frac{\partial z_{v_i}}{\partial T_{v_i}}, \qquad \frac{\partial z_{v_i}}{\partial T_{v_i}}^\top M_{v_i}^{-1}r_{v_i}, \qquad T_{v_i}. \tag{62}$$

Communication from C to l_j: The intersection-center needs to pass $\Delta\mu_{T(i)}$ $\forall i\in\mathcal{N}_{l_j}$ to construct the right-hand side of (33b) for lane j.

Communication from l_j and C to v_i: The dependencies on $\Delta\mu_T$ and $\Delta\mu_{l_j}$ in (33c) reads as

$$\begin{bmatrix} M_{v_il} & M_{v_iT}\end{bmatrix}\begin{bmatrix}\Delta z_l \\ \Delta z_T\end{bmatrix} = \frac{\partial z_{v_i}}{\partial p_{v_i}}(\Delta\mu^{\mathrm{r}}_{v_i} - \Delta\mu^{\mathrm{f}}_{v_i}) + \frac{\partial z_{v_i}}{\partial T_{v_i}}\frac{\partial h_T}{\partial T_{v_i}}\Delta\mu_{T(i)}. \tag{63}$$

That is, the intersection-center must pass $\Delta\mu_{T(i)}$, and the lane-center for vehicle i need to send $\Delta\mu^{\mathrm{r}}_{v_i} - \Delta\mu^{\mathrm{f}}_{v_i}$ to construct the right-hand side of (33c) for vehicle i.

V. 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 separated between, and computed in parallel within, the problem levels.

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

$$s + \alpha^{\max}\Delta s \geq \kappa s, \tag{64a}$$

$$\mu + \alpha^{\max}\Delta\mu \geq \kappa s, \tag{64b}$$

where $\kappa > 0$ is a parameter [23]. Due to the problem structure, (64) can be evaluated separately for the vehicles, giving $\alpha_{v_i}^{\max}$, $\forall i \in \mathcal{N}$, 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

$$\alpha^{\max} = \min(\alpha_{v_1}^{\max}, \dots, \alpha_{v_N}^{\max}, \alpha_{l_1}^{\max}, \dots, \alpha_{l_L}^{\max}, \alpha_T^{\max}). \quad (65)$$

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 [23], which reads

$$\phi(y, s) = \sum_{i=1}^{N} \phi_{v_i}(y_{v_i}, s_{v_i}) + \sum_{j=1}^{L} \phi_{l_j}(p_{l_j}, s_{l_j}) + \phi_T(T, s_T), \quad (66)$$

where

$$\begin{aligned} \phi_{v_i}(y_{v_i}, s_{v_i}) &= J_{v_i}(y_{v_i}) + \nu \left(||g_{v_i}(y_{v_i})||_1 + ||h_{v_i}(y_{v_i}) + s_{v_i}||_1 \right) \\ &\quad - \tau \mathbf{1}^\top \log(s_{v_i}), \\ \phi_{l_j}(p_j, s_{l_j}) &= \nu ||h_{l_j}(p_{l_j}) + s_{l_j}||_1 - \tau \mathbf{1}^\top \log(s_{l_j}), \\ \phi_T(T, s_T) &= \nu ||h_T(T) + s_T||_1 - \tau \mathbf{1}^\top \log(s_T), \end{aligned}$$

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

$$\phi(y + \alpha \Delta y, s + \alpha \Delta s) \leq \phi(y, s) + \zeta \phi'(y, s) \alpha, \quad (67)$$

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

$$\begin{aligned} \phi'(y, s) &= \left. \frac{\mathrm{d}\phi(y + \alpha \Delta y, s + \alpha \Delta s)}{\mathrm{d}\alpha} \right|_{0 \leftarrow \alpha} = \frac{\partial \phi}{\partial y} \Delta y + \frac{\partial \phi}{\partial s} \Delta s \\ &= \sum_{i=1}^{N} \phi'_{v_i}(y_{v_i}, s_{v_i}) + \sum_{j=1}^{L} \phi'_{l_j}(p_{l_j}, s_{l_j}) + \phi'_T(T, s_T). \quad (68) \end{aligned}$$

Evaluation of $\phi(y, s)$, $\phi'(y, s)$ can be separated between the vehicles ($\phi_{v_i}(y_{v_i}, s_{v_i})$, $\phi'_{v_i}(y_{v_i}, s_{v_i})$), lanes ($\phi_{l_j}(p_{l_j}, s_{l_j})$, $\phi'_{l_j}(p_{l_j}, s_{l_j})$) and the intersection ($\phi_T(T, s_T)$, $\phi'_T(T, s_T)$), Algorithm 2 summarizes the procedure.

C. Handling non-convexity

It is known [23] that $(\Delta y, \Delta s)$ is a descent direction on ϕ, if

$$v^\top \begin{bmatrix} \nabla_y^2 \mathcal{L} & \\ & D(s)^{-1} D(\mu) \end{bmatrix} v > 0, \ \forall v \ : \ \begin{bmatrix} \nabla_y g^\top & \\ \nabla_y h^\top & I \end{bmatrix} v = 0. \quad (69)$$

Since $D(s)^{-1} D(\mu) \succ 0$ by construction, this is determined by $\nabla_y^2 \mathcal{L}$. If (69) 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 $\nabla_y^2 \mathcal{L} = \mathrm{blockdiag}(\nabla_{y_{v_1}}^2 \mathcal{L}, \dots, \nabla_{y_{v_N}}^2 \mathcal{L})$. That is, Δz is a descent direction on $\phi(y, s)$ if each vehicle uses a positive definite modification of $\nabla_{y_{v_i}}^2 \mathcal{L}$ in (27) when necessary.

VI. 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.

Algorithm 2 Distributed selection of step-size α, first level.

procedure StepSizeSelection(z,Δz,τ)
  $\forall v_i$: Find $\alpha_{v_i}^{\max}$, assemble $\phi_{v_i}(y_{v_i}, s_{v_i})$ and $\phi'_{v_i}(y_{v_i}, s_{v_i})$, pass to C together with ΔT_{v_i}, pass Δp_{v_i} to $l(i)$.
  $\forall l_j$: Find $\alpha_{l_j}^{\max}$, assemble $\phi_{l_j}(p_{l_j}, s_{l_j})$ and $\phi'_{l_j}(p_j, s_{l_j})$
  C: Find $\alpha_T^{\max}$ and determine $\alpha^{\max}$ with (65).
  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
    $\forall v_i$: Pass $\phi_{v_i}(y_{v_i} + \alpha \Delta y_{v_i}, s_{v_i} + \alpha \Delta s_{v_i})$ to C
    $\forall l_j$: Pass $\phi_{l_j}(p_{l_j} + \alpha \Delta p_{l_j}, s_{l_j} + \alpha \Delta s_{l_j})$ 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 (66)
    if $\phi(y + \alpha \Delta y, s + \alpha \Delta s) < \phi(y, s) + \alpha \gamma \phi'(y, s)$ **then**
      return α, $\alpha^{\max}$ and accept-notice to all v_i, l_j
    else
      $\alpha \leftarrow \beta \alpha$
      Pass α to all v_i, l_j
    end if
  end loop
end procedure

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

$$||r_{\tau^{[k]}}(z^{[k+1]})|| < \varepsilon \quad \text{and} \quad \tau^{[k]} < \varepsilon, \quad (70)$$

for some tolerance ε. Termination must thus be decided centrally, and all r_{v_i}, r_{l_j} must be sent to the intersection-center.

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

$$\dot{\mathrm{p}}_i(t) = \mathrm{v}_i(t), \quad (71a)$$

$$\dot{\mathrm{v}}_i(t) = \frac{1}{m_i} (c^{\mathrm{Torque}} \mathrm{M}_i(t) - \mathrm{F}_i^{\mathrm{b}} - c^{\mathrm{drag}} \mathrm{v}_i(t)^2 - c^{rr}), \quad (71b)$$

$$\mathrm{M}(t) \leq \min(\mathrm{M}^{\max}, P_i^{\max} / \omega_i(t)) \quad (71c)$$

$$0 \leq \omega_i(t) \leq \omega_i^{\max}, \quad (71d)$$

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 $P_i^{\max}$ [17], and we use $K = 100$ and Explicit 4th order

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 v_i and l_j
  $\forall l_j$: Initialize $z_{l_j}^{[0]}$ and send $\mu_{l_j}^{[0]}$ to $v(j)$
  $\forall v_i$: Initialize $z_{v_i}^{[0]}$ and send $T_{v_i}^{[0]}$ to C, $p_{v_i}^{[0]}$ to l_j
  loop
    $k \leftarrow k+1$
    C: Send iteration start and $\tau^{[k]}$ to all v_i, l_j.
    $\forall v_i$: Compute M_{v_i}, r_{v_i} , modify if necessary $\nabla^2_{y_{v_i}}\mathcal{L}$.
    $\Delta z^{[k]} \leftarrow$SEARCHDIRECTION($z^{[k]}$,$\tau^{[k]}$)
    $\alpha^{[k]} \leftarrow$STEPSIZESELECTIONz,Δz,τ
    C: Update $z_T^{[k+1]} \leftarrow z_T^{[k]} + \alpha^{[k]}\Delta z_T^{[k]}$
    $\forall l_j$: Update $z_{l_j}^{[k+1]} \leftarrow z_{l_j}^{[k]} + \alpha^{[k]}\Delta z_{l_j}^{[k]}$
    $\forall v_i$: Update $z_{v_i}^{[k+1]} \leftarrow z_{v_i}^{[k]} + \alpha^{[k]}\Delta z_{v_i}^{[k]}$
    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

Runge-Kutta integrators from ACADO [24] with $\Delta t = 0.2$. The objective function is

$$J_{v_i}(y_{v_i}) = Q_i^{\mathrm{f}}(v_{i,K} - v_r)^2 + \sum_{k=0}^{K} Q_i(v_{i,k} - v_r)^2 + (u_{i,k} - u_i^r)^\top R_i(u_{i,k} - u_i^r), \quad (72)$$

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 (71b) 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 $w_i^{[0]}, T_{v_i}^{[0]}$ is that where all vehicles drive at v_r for $k = 0, \ldots, K$, $\lambda^{[0]} = 0$, $\mu^{[0]} = s^{[0]} = \mathbf{1}$ and $\tau^{[0]} = 1$.

The development of $||r(z)^{[k]}||_2$ and $\tau^{[k]}$ is shown in Fig. 5a, and the step sizes used is shown in Fig. 5b.

To illustrate how the algorithm progresses, the velocity profiles of one vehicle is provided in Fig. 5c. Note that the final 15 iterates are indistinguishable. In a practical context, little would be lost by stopping after iteration 16.

For illustration, the sparsity-pattern of M is given in Fig. 6. 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_{v_i}, roughly sized 2010×2010 [2], and the lane blocks M_{l_j}, sized 404×404. Since the factors for all M_{v_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

$$t^{\mathrm{direction}} \approx \max_{i \in \mathcal{N}}(\mathrm{timeToFactorize}(M_{v_i})) + \max_{j \in \mathcal{N}_L}(\mathrm{timeToFactorize}(\Gamma_j)) \quad (73)$$

However, computational time will likely not dominate the time it takes to find a solution. Following the lessons learned from the experiments reported in [16], 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.

VII. 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 VII-A, and demonstrate that the proposed algorithm requires an unrealistic data exchange. We discuss how the requirements can be reduced in Sections VII-B and VII-C.

A. Analysis of Communication Requirements

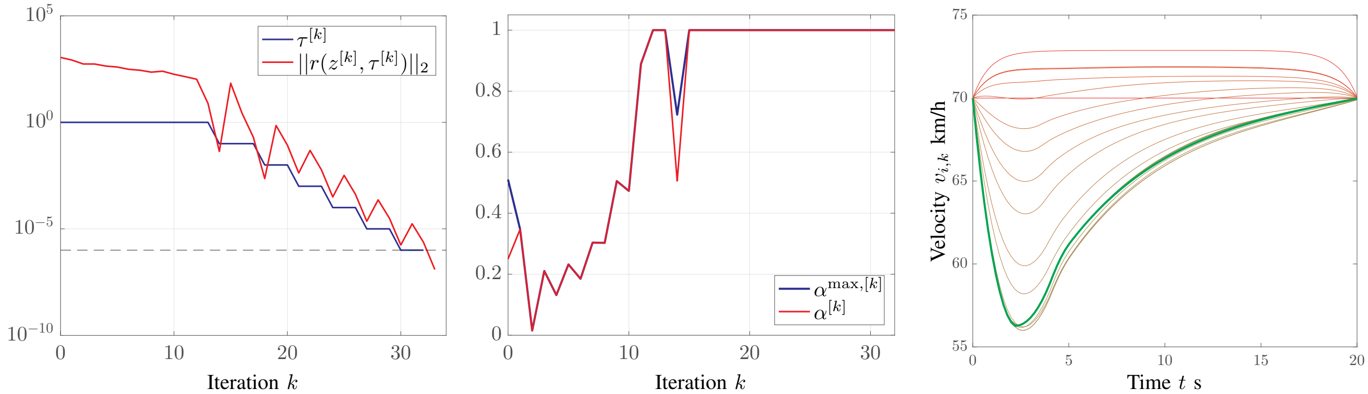
Most data is exchanged during solution of the KKT-system in Algorithm 1 and the selection of the step-size in Algorithm 2. Descriptions of the data involved as well as the number of floats communicated are summarized in Table I.

Most often $K \gg n_{T_{v_i}}$, whereby most communication occurs during Line 2 of Algorithm 1 when the vehicle sends $\mathcal{D}_{v_i \mapsto l(i)}$. Besides the communication between the lane-centers and the intersection-center and an initial round of communication where the initial guess $z_T^{[0]}$, $z_l^{[0]}$ 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. 7, these can be sent together with the search-direction and step-size results.

Communication in the Example: In the example $K = 100$, whereby each vehicle sends more than 5000 floats *per iterate* (more than 320000 bits) to their respective lane-center. Even if all vehicles communicate in parallel, the physical transmission will take at least 58.7 ms using the 802.11p protocol; the current standard for vehicular communications[3]. During 33 iterations, at least 1.94 s would be spent communicating to construct Δz, much too high for practical applications. Next, we discuss how the data exchange can be reduced.

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

[3]The time per bit is computed using the formula $50 + 8\mathrm{ceil}((n_{\text{data bits}} + 22)/48)$ μs, reported in [25]. 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 $z_{v_i}^{[k]}$.

Fig. 5: Data 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 in hues in-between.. In (a), the increases in $||r(z)||$ that follow decreases in τ, is due to the nature of the simple IP-method deployed.

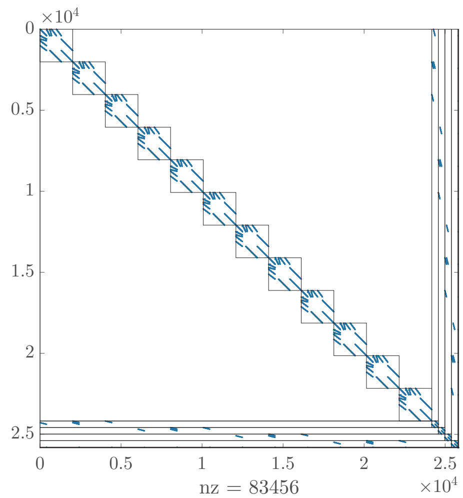


Fig. 6: KKT-matrix $M(z)$ from a 12-vehicle scenario. The large upper left hand block is M_v, consisting of the sub-blocks M_{v_i}, $i \in \mathcal{N}$. The smaller blocks in the lower right corner are M_{l_j}, while M_T is barely visible. The lines demarcate the sections of M_{ca} and M_{cv} associated with the RECA constraints on each lane and (barely visible) the SICA constraints

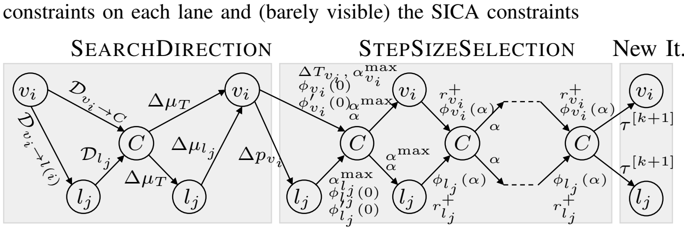


Fig. 7: Illustration of the communication flows in the problem. The horizontal direction indicates sequence whereas the vertical differentiates the vehicle, lane and intersection levels. Here, $\mathcal{D}_{l_j}$ collects $\Lambda_j \Gamma_j^{-1} \Lambda_j^\top$ and $\Lambda_j \Gamma_j^{-1} \gamma_j$, 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 $r^+ = r(z + \alpha \Delta z)$.

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 due to use of second order information, and enforcement of the RECA at every time instant k. One obvious remedy is 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 (12) with constraints on the form

$$p_{i,k} + \delta_{ij}/2 \leq B_{ij}(k, \theta_{ij}), \quad k = 1, \dots, K \tag{74a}$$
$$B_{ij}(k, \theta_{ij}) + \delta_{ij}/2 \leq p_{j,k}, \quad k = 1, \dots, K, \tag{74b}$$

where $B_{ij}(k, \theta_{ij})$ is a function of k, parametrized with $\theta_{ij} \in \mathbb{R}^q$, and introduce θ_{ij} as additional decision variables in the fixed order problem (14). Rather than enforcing the RECA constraints directly using p_{v_i} and p_{v_i}, (74) 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, and enables practical schemes which relies on data exchange that scales as q^2. The price paid 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. In the following discussion, we use notation as shown in (8), where $\theta_{v_i}^{\mathrm{f}}$ and $\theta_{v_i}^{\mathrm{r}}$ are coupling parameters of vehicle i's forward (74a) and rearward (74b) facing RECA conditions, respectively. That is, if vehicle j is in front of vehicle i, we have that $\theta_{v_i}^{\mathrm{f}} = \theta_{v_j}^{\mathrm{r}} = \theta_{ij}$. Similarly, we let $B_{v_i}^{\mathrm{f}}(\theta_{v_i}^{\mathrm{f}})$ and $B_{v_i}^{\mathrm{r}}(\theta_i^{\mathrm{r}})$ collect the function $B_{ij}(k, \theta_{ij})$ for $k = 1, \dots, K$ in the forward- and rearward-facing constraints for vehicle i, respectively, and denote the corresponding multipliers and slacks $(\nu_{v_i}^{\mathrm{f}}, s_{v_i}^{\mathrm{f}})$ and $(\nu_{v_i}^{\mathrm{r}}, s_{v_i}^{\mathrm{r}})$. We also let $\theta_{v_i} = (\theta_{v_i}^{\mathrm{f}}, \theta_{v_i}^{\mathrm{r}})$, $B_{v_i}(\theta_{v_i}) = (B_{v_i}^{\mathrm{f}}(\theta_{v_i}^{\mathrm{f}}), B_{v_i}^{\mathrm{r}}(\theta_{v_i}^{\mathrm{r}}))$, $\nu_{v_i} = (\nu_{v_i}^{\mathrm{f}}, \nu_{v_i}^{\mathrm{r}})$ and $s_{v_i}^{\mathrm{RECA}} = (s_{v_i}^{\mathrm{f}}, s_{v_i}^{\mathrm{r}})$ for each vehicle, collect θ_{ij} $\forall (i,j) \in \mathcal{R}_j$ in θ_{l_j}, and collect θ_{l_j} $\forall j \in \mathcal{N}_L$ in θ.

The "Primal" Approach: The first alternative is to handle the coupling parameters θ_{ij} at the lane-centers, and constraints (74a), (74b) on-board the vehicles. The lane-center variables are in this case $z_{l_j} = \theta_{l_j}$, so that $r_{l_j} = \nabla_{\theta_{l_j}} \mathcal{L} = \sum_{i \in \mathcal{N}_{l_j}} \nabla_{\theta_{l_j}} B_{v_i} \nu_{v_i}$, and the vehicle variables z_{v_i} include $(\nu_{v_i}, s_{v_i}^{\mathrm{RECA}})$. The information sent to the lane-center is

Link	Location	Data per Communication Round	#Floats
		SEARCHDIRECTION	
v_i to lj	A.1,L.2	$\underbrace{\frac{\partial z_{v_i}}{\partial p_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial p_{v_i}}}_{(K+1)^2}, \underbrace{\frac{\partial z_{v_i}}{\partial T_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial p_i}}_{n_{T_{v_i}}(K+1)}, \underbrace{\frac{\partial z_{v_i}}{\partial p_{v_i}}^\top M_{v_i}^{-1}(r_{v_i} + \nabla_{z_{v_i}} \mathcal{L}_c)}_{K+1}, \underbrace{p_{v_i}}_{K+1}$	$\frac{1}{2}K^2 + \left(n_{T_{v_i}} + \frac{9}{2}\right) K + 3 + n_{T_{v_i}}$
v_i to C	A.1, L.2	$\underbrace{\frac{\partial z_{v_i}}{\partial T_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial T_{v_i}}}_{n_{T_{v_i}}^2}, \underbrace{\frac{\partial z_i}{\partial T_i}^\top M_i^{-1}(r_i + \nabla_{z_i} \mathcal{L}_c)}_{n_{T_i}}, \underbrace{T_i}_{n_{T_i}}$	$\frac{1}{2}n_{T_i}^2 + \frac{5}{2}n_{T_i}$
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 v_i	A.1, L.4	$\underbrace{\Delta\mu_{T,i}}_{n_{T_i}}$	n_{T_i}
l_j to v_i	A.1, L.5	$\underbrace{\Delta\mu_{v_i}^{\mathrm{f}}}_{K+1}, \underbrace{\Delta\mu_{v_i}^{\mathrm{r}}}_{K+1}$	$2K + 2$
		STEPSIZESELECTION	
v_i to l_j	A.2, L.2	$\underbrace{\Delta p_{v_i}}_{K+1}$	$K + 1$
v_i to C	A.2, L.2	$\underbrace{\Delta T_{v_i}}_{n_{T_{v_i}}}, \alpha_{v_i}^{\max}, \phi_{v_i}(y_{v_i}, s_{v_i}), \phi'_{v_i}(y_{v_i}, s_{v_i})$	$n_{T_{v_i}} + 3$
l_j to C	A.2, L.3	$\alpha_{l_j}^{\max}, \phi_{l_j}(p_{l_j}, s_{l_j}), \phi'_{l_j}(p_{l_j}, s_{l_j})$	3
C to v_i, l_j	A.2, L.6/15	$\alpha, \alpha^{\max}$	1
v_i to C	A.2, L.8	$\phi_{v_i}(y_{v_i} + \alpha\Delta y_{v_i}, s_{v_i} + \alpha\Delta s_{v_i})$	1
l_j to C	A.2, L.9	$\phi_{l_j}(p_{l_j} + \alpha\Delta p_{l_j}, s_{l_j} + \alpha\Delta s_{l_j})$	1

TABLE I: Summary of the communication between the vehicles (v_i), lane-centers (l_j) and intersection-center (C). The second column states in which Algorithm (A) and Line (L) the communication occurs. The numbers under braces denotes the size of the object, where $n_{T_{v_i}}$ sizeof($\mathrm{T_{v_i}}$). $T_{v(j)}$ collects T_{v_i} $\forall i \in \mathcal{N}_{l_j}$, $n_{T_j^l} = \mathrm{sizeof}(T_j^l)$ and $\Delta\mu_{T_j^l}$ concernes the multipliers of the SICA constraints that involve $T_{v(j)}$. The three final rows is the line-search.

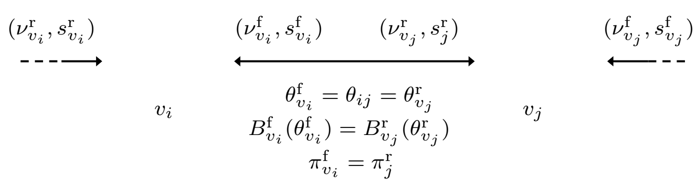


Fig. 8: Illustration of the relationship between the variables introduced to reduce the communication demands.

$$\nabla_{\theta_{v_i}} B_{v_i} \frac{\partial z_{v_i}}{\partial \nu_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial \nu_{v_i}} \nabla_{\theta_i} B_{v_i}^\top, \qquad \text{size } (2q)^2, \tag{75a}$$

$$\nabla_{\theta_{v_i}} B_{v_i} \frac{\partial z_{v_i}}{\partial \nu_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial T_{v_i}}, \qquad \text{size } 2qT_{v_i}, \tag{75b}$$

$$\nabla_{\theta_{v_i}} B_{v_i} \nu_{v_i}, \qquad \text{size } 2q. \tag{75c}$$

Assuming that both $\theta_{v_i}^{\mathrm{f}}$ and $\theta_{v_i}^{\mathrm{r}}$ 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 $\Delta\theta_{v_i}$ ($2q$ floats). Since p_{v_i} and p_{v_j} only are coupled indirectly, the term $\frac{\partial z_{v_i}}{\partial p_{v_i}}\left(\mu_{v_i}^{\mathrm{r}} - \mu_{v_i}^{\mathrm{f}}\right)$ is not present in $\nabla_{y_{v_i}} \mathcal{L}$, and $\Delta\mu_{v_i}^{\mathrm{r}}$ or $\Delta\mu_{v_i}^{\mathrm{f}}$ is not sent by the lane-centers.

The "Dual" Approach: The second alternative is to consider $\theta_{v_i}^{\mathrm{r}}$ and $\theta_{v_j}^{\mathrm{r}}$ as separate variables, and couple the vehicles through the constraint

$$\theta_{v_i}^{\mathrm{r}} - \theta_{v_j}^{\mathrm{f}} = 0. \tag{76}$$

In this approach, $(\theta_{v_i}, \nu_{v_i}, s_{v_i}^{\mathrm{RECA}})$ is included in the vehicle variables y_{v_i} while $z_{l_j} = \pi_{l_j}$ collects the Lagrange multipliers π_{ij} of the constraints (76), for all couplings on lane j. Correspondingly, $r_{l_j}(z_{v(j)}, z_{l_j})$ consists of $\theta_{v_i}^{\mathrm{r}} - \theta_{v_j}^{\mathrm{f}}$ for all couplings on lane j, and the information sent from vehicle i on Line 5 of Algorithm 1 is

$$\frac{\partial z_{v_i}}{\partial \theta_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial \theta_{v_i}}, \qquad \text{size } (2q)^2, \tag{77a}$$

$$\frac{\partial z_{v_i}}{\partial \theta_{v_i}}^\top M_{v_i}^{-1} \frac{\partial z_{v_i}}{\partial T_{v_i}}, \qquad \text{size } 2qT_{v_i}, \tag{77b}$$

$$\theta_{v_i}, \qquad \text{size } 2q. \tag{77c}$$

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

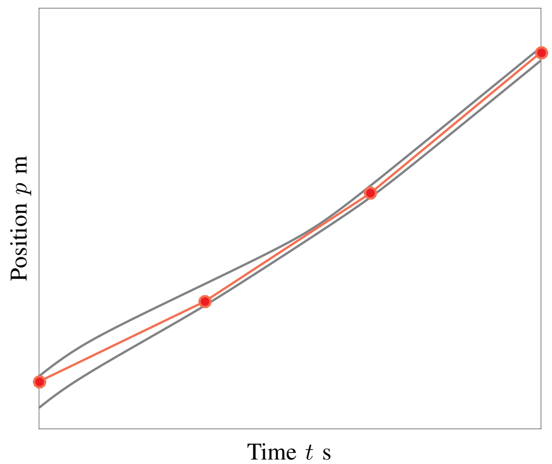


Fig. 9: Illustration with $B_{ij}(k,\theta_{ij})$ piece-wise linear with three linear segments. The trajectories of two vehicles are drawn in gray and the parameterized function in red, with the round markers being the parameters.

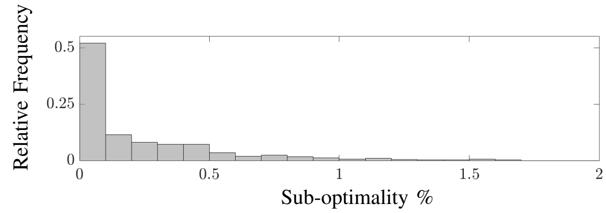


Fig. 10: Distribution of the suboptimality resulting from the use of approximate RECA constraints with a piece-wise linear $B_{ij}(k,\theta_{ij})$

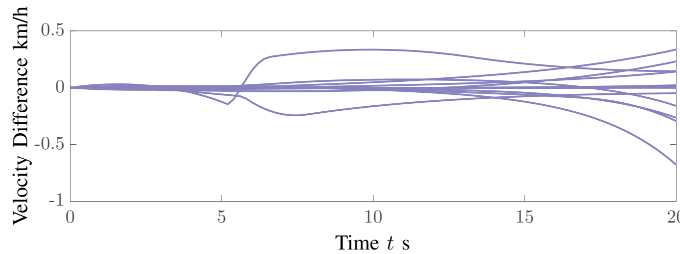


Fig. 11: Difference between the optimal velocity profiles and those obtained using the parametrized RECA constraints (74) for a 16 vehicle scenario.

$$\nabla_{y_{v_i}}\mathcal{L} = \nabla_{y_{v_i}} J_{v_i} + \nabla_{y_{v_i}} g_{v_i}\lambda_{v_i} + \nabla_{y_{v_i}} g_{v_i}\mu_{v_i} + \frac{\partial y_{v_i}}{\partial \theta^{\mathrm{f}}_{v_i}}\pi^{\mathrm{f}}_{v_i} - \frac{\partial y_{v_i}}{\partial \theta^{\mathrm{r}}_{v_i}}\pi^{\mathrm{r}}_{v_i} + \frac{\partial y_{v_i}}{\partial T_{v_i}}\mu_{i,T}. \quad (78)$$

Consequently, the lane-center-to-vehicle communication on Line 5 of Algorithm 1 consists of $\Delta\pi^{\mathrm{f}}_{v_i}$, $\Delta\pi^{\mathrm{r}}_{v_i}$, 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 $\nabla^2_y\mathcal{L}$. Due to this, positive definiteness of $\nabla_y\mathcal{L}$ can not be ensured by modifying $\nabla^2_{y_{v_i}}\mathcal{L}$ (c.f. the discussion in Section V-C). In Dual approach, 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. 9, where $B_{ij}(k,\theta_{ij})$ is the piece-wise linear function

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

where the superscript on θ_{ij} indicates the element (i.e. $q=4$). When $n_{T_{v_i}}=4$, 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 8), 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 VI-A. The vehicles were initialized at randomly drawn distances between 50 and 150 meters from the intersection, and the crossing order was computed with the heuristic of [19]. As Fig. 10 demonstrates, the sub-optimality induced by the parametrized constraints is below 0.1 % in more than 50% of the cases. The small impact is illustrated in Fig. 11, which shows the difference in the optimal velocity profiles for the scenario corresponding to the median sub-optimality. Interestingly, the difference between the optimal control commands at $k=0$ in the two solutions is smaller than 0.013 % of the input range for all vehicles. This is below the quantization error of many actuators, whereby the difference might not be noticed in practice. Finally, more "flexible" parametrization of B_{ij} could be used to reduce sub-optimality, e.g. by including additional linear segments or polynomials.

C. Reduction of the number of communication rounds

Algorithm 3 is rudimentary and could be augmented to converge in fewer iterations. For instance, instead of the simple update rule for τ, adaptive or a Predictor-Corrector strategies could be employed [23].

To reduce the number of iterations, the problem to a sufficiently high accuracy for larger τ. In this case, all equality constraints are satisfied to the set tolerance, and the inequality constraints are satisfied with a margin. While convergence to is reached in fewer iterations, sub-optimality is introduced. However, as remarked in the discussion on Fig. 5c, practically acceptable solutions can be obtained for τ larger than relevant tolerances on $||r(z)||$. It is therefore expected that the practical implications of stopping Algorithm 3 at τ significantly larger than ε is small. As an example, Fig. 12 shows results from the scenario considered in Section VI-A, where τ is prevented from being smaller than $\tau^{\min}$, for $\tau^{\min}$ between 1 and 10^{-6}, with $\varepsilon = 10^{-6}$ for all cases. The sub-optimality induced and the number iterations required to reach $||r(z)|| \leq \varepsilon$ is shown in Fig. 12. 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 % sub-optimality. The optimal velocity profiles for one vehicle for the different values of $\tau^{\min}$ is shown in Fig. 13 (c.f. Fig. 5c). The difference with

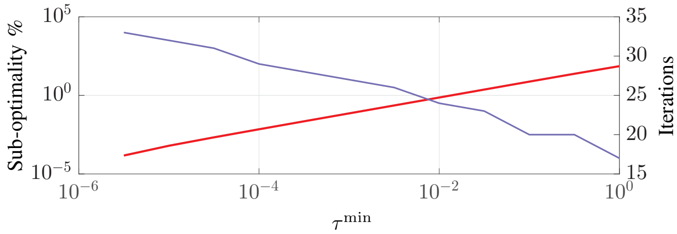


Fig. 12: Sub-optimality (red) and number of iterations (blue) required to reach $||r(z)|| \leq \varepsilon$ for different value of $\tau^{\min}$.

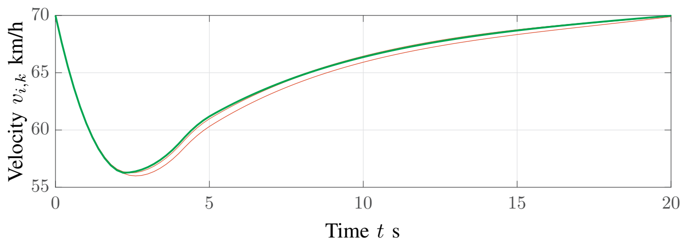


Fig. 13: Velocity profiles for a vehicle corresponding to the solution satisfying $||r(z)||_2 \leq 10^{-6}$ for different values of $\tau^{\min}$. For the red trajectory $\tau^{\min} = 1$ and for the green trajectory $\tau^{\min} = 10^{-6}$. Hues between red and green correspond to intermediate values.

respect to the optimal solution is small enough to be practically irrelevant for all but the highest value of $\tau^{\min}$.

VIII. 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 parameterized 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 *et al.*, "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 *et al.*, "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.

[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.

[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.

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

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

[14] C. Bali and A. Richards, "Merging vehicles at junctions using mixed-integer model predictive control," in *European Control Conference*, 06 2018, pp. 1740–1745.

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

[16] 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.

[17] ——, "Energy-optimal coordination of autonomous vehicles at intersections," in *European Control Conference*, June 2018, pp. 602–607.

[18] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections with turns," in *European Control Conference*, 2019.

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

[20] 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.

[21] 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.

[22] 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.

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

[24] 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

[25] 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.