2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

# Fully Decentralized ADMM for Coordination and Collision Avoidance

Felix Rey, Zhoudan Pan, Adrian Hauswirth, and John Lygeros

***Abstract*— We utilize the alternating direction method of multipliers (ADMM) to devise a communication and control protocol for fully decentralized coordination of moving agents. In particular, we consider a model predictive control (MPC) framework for a group of agents. Each agent has linear dynamics with convex state and input constraints. Nonconvex collision avoidance constraints constitute inter-agent coupling. We develop an algorithm that, if applied by all agents, mediates individual objectives while satisfying constraints. The resulting procedure exhibits several attractive features, including $(i)$ fully decentralized, parallel, and aggregator-free operation, where each agent is only aware of its closest neighbors; $(ii)$ adaptive linearization for handling the nonconvex collision avoidance constraints; and $(iii)$ the treatment of uncooperative agents.**

## I. Introduction

A rising level of automation in multi-agent systems constitutes the demand for scalable, versatile, and fast coordination schemes. Prominent application areas are autonomous driving, marine navigation, process line automation, and coordination of flight vehicles, such as quadcopters and drones. In this paper, we advocate an optimization-based coordination method that uses MPC [1] and ADMM [2]. The agents have linear dynamics and follow a position target, while nonconvex constraints prevent collisions. We avoid using a central coordination facility which would establish a single point of failure, limit the number of agents by its computational capabilities, and require all agents' blind trust in assigning collision-free trajectories. Instead, we devise a communication and control protocol where all agents share the coordination effort. Our method stands out by evolving around the agents' point of view, i.e., each agent navigates in a local coordinate system and communicates with neighboring agents only. The agents participate equally and simultaneously through frequent communication. This fully decentralized paradigm leads to a high level of agent-autonomy, tolerates frequent setup changes, and promotes resilience against failures. We handle the nonconvex collision avoidance constraint through first-order Taylor approximation at each ADMM iteration. While this linearization curtails the solution space of the nonconvex problem, it results in a lightweight and therefore fast formulation that guarantees collision avoidance upon convergence. Since we linearize at each ADMM iteration, we achieve a high adaptation rate, reducing the negative influence of the solution space curtailment. We show that our method handles static obstacles and formation flight, as well as uncooperative agents that do not communicate. Fig. 1 illustrates the setup.

F. Rey rey⋆, Z. Pan panz@student.ethz.ch, A. Hauswirth hadrian⋆ and J. Lygeros lygeros⋆ are with the Automatic Control Laboratory at ETH Zurich, Switzerland, ⋆@control.ee.ethz.ch.

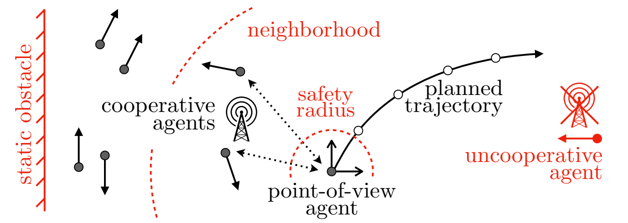


Fig. 1. Illustration of the coordination setup in two dimensions, centered around the point of view of a single agent.

### *Related Literature*

The review focuses on optimization-based coordination; an excellent overview of the greater field is presented in [3]. In [4], collisions are avoided by partitioning the space into convex safe-regions, which leads to a mixed-integer optimization problem. Using the same mixed-integer models, [5] advocates a distributed scheme where agents take turns in modifying the planned trajectories. Compared to [4], [5], we linearize the collision avoidance constraint, avoiding the substantial computational burden of binary decision variables. In [6]–[8], a hierarchical approach is pursued, where collision avoidance is separated from other tasks, such as satisfying local constraints. We follow a different route by handling all coordination aspects simultaneously. Being closer related to our approach, [9] develops an efficient ADMM-based coordination method, however still uses partially centralized elements. The same is true for [3], [10], which devise a coordination procedure that requires a central facility. Further, in [3], [9], [10], the linearization of the avoidance constraint is performed only once per MPC call, resulting in less frequent adaptation and therefore potentially less reactive agents. On the other hand, [3], [9], [10] go beyond our setup by including nonlinear agent dynamics. Another related line of research is [11]–[13], where an agent navigates through an environment with moving obstacles, using spline-based trajectory plans. This setup is extended in [14]–[16] to include multiple agents and inter-agent collision avoidance. The most recent work [16] contains an ADMM-based procedure that is related to our approach. The difference lies in the convexification of collision avoidance, where our approach adheres more closely to the standard ADMM framework. Another prominent difference is that we use discrete-time systems instead of splines. The work in [16] goes beyond our setup with thoughts on recursive feasibility and suboptimal algorithm termination. Conversely, we complement [16] with the use of local coordinate systems, a deadlock-protection mechanism, and a more rigorous discussion of convergence guarantees. The entire line [11]–[16] contains excellent application examples.

978-3-9524-2699-9 ©2018 EUCA

## II. CENTRALIZED SETUP AND CONVEX FORMULATION

In this section, we define the coordination problem, and we present the linearization of the collision avoidance constraint.

### A. Coordination Problem

We consider a setup where $M$ agents navigate in an $n$-dimensional space under the presence of nonconvex collision avoidance constraints. We approach this problem in a discrete-time MPC framework, where an optimization routine updates the planned trajectories at each time step. At first, we focus on a centralized formulation that coordinates all agents. The nonconvex coordination problem is

$$\min_{\{x_i, v_i, a_i\}_i} \quad \sum_{i=1}^{M} J_i(x_i, a_i) \tag{1a}$$
$$\text{s.t.} \quad (x_i, v_i, a_i) \in \mathcal{C}_i \quad \forall i \tag{1b}$$
$$h_{ij}(x_i, x_j) \geq 0 \quad \forall i, j \text{ and } j \neq i, \tag{1c}$$

where we write $\{x_i, v_i, a_i\}_i$ for $\{(x_i, v_i, a_i) \,|\, i = 1, \ldots, M\}$. We use the position $x_i$, velocity $v_i$, and acceleration $a_i$ along the planned trajectory for agent $i$. Below, we define the parametrization $(x_i, v_i, a_i)$, the objectives $J_i$, the dynamics $\mathcal{C}_i$, and the nonconvex avoidance constraints $h_{ij}$.

### B. Agent Modeling

*1) Trajectories and Dynamics:* Each agent $i$ has a current position $x_i^0 \in \mathbb{R}^n$, velocity $v_i^0 \in \mathbb{R}^n$, and acceleration $a_i^0 \in \mathbb{R}^n$. Initialized with these values, we predict its movement with constrained double-integrator dynamics

$$\left.\begin{aligned} x_i^{k+1} &= x_i^k + T v_i^k \\ v_i^{k+1} &= v_i^k + T a_i^k \end{aligned}\right\} \quad k = 0, 1, \ldots, N+1 \tag{2a}$$
$$(x_i^{k+2}, v_i^{k+1}, a_i^k) \in (\mathcal{X}_i \times \mathcal{V}_i \times \mathcal{A}_i) \quad k = 1, \ldots, N, \tag{2b}$$

where $\times$ is the Cartesian product and $T > 0$ is the discretization interval. The sets $\mathcal{X}_i$, $\mathcal{V}_i$, and $\mathcal{A}_i$ model static obstacles, as well as limits for velocity and acceleration. We require these sets to be convex, noting that they can be changed between MPC calls, e.g., depending on the agent's position and orientation. We collect $x_i = (x_i^3, \ldots, x_i^{N+2})$, $v_i = (v_i^2, \ldots, v_i^{N+1})$, and $a_i = (a_i^1, \ldots, a_i^N)$, which are the free decision variables as illustrated in Fig. 2. For (1b), we then use $\mathcal{C}_i = \{(x_i, v_i, a_i) \,|\, (2)\}$.

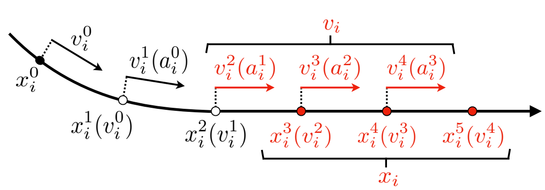


Fig. 2. Trajectory for $N = 3$. The free decision variables, which are not determined by the initial conditions $(x_i^0, v_i^0, a_i^0)$, are indicated in red.

*2) Individual Objectives:* Each agent has the objective

$$J_i(x_i, a_i) = \|x_i - r_i\|_Q^2 + \|a_i\|_R^2, \tag{3}$$

where $\|x_i - r_i\|_Q^2$ is equal to $(x_i - r_i)^\top Q (x_i - r_i)$, $r_i$ is the target trajectory, and $Q, R$ are positive semi-definite weight matrices. In combination with the term $\|a_i\|_R^2$, the matrix $R$ adjusts the acceleration intensity. It is possible to choose $r_i$ relative to other agents, making them move in formations.

*3) Collision Avoidance:* Agents prevent collisions by placing Euclidean spheres with safety radius $\Delta_i$ around themselves. For each agent pair $(i, j)$, this is encoded with

$$h_{ij}(x_i, x_j) = \min_{k=3,\ldots,N+2} \|x_i^k - x_j^k\|_2 - \Delta_i. \tag{4a}$$

*4) Neighborhoods:* Towards a decentralized formulation, we introduce a concept of neighborhoods. We define

$$\mathcal{N}_i = \left\{ j \in \{1, 2, \ldots, M\} \backslash \{i\} \,\middle|\, \|x_i^0 - x_j^0\|_2 \leq \Delta^{dect} \right\}, \tag{5}$$

where $\Delta^{dect} > 0$ is the detection distance that all agents have in common. We assume that $\Delta^{dect}$ is large enough, such that agents detect each other before they need coordination. In this case, we can enforce the collision avoidance constraint (1c) for $j \in \mathcal{N}_i$ only, without changing the problem.

### C. Convex Problem Formulation

We perform a first-order Taylor approximation of the avoidance constraint around nominal trajectories $(\bar{x}_i, \bar{x}_j)$. The strategy to choose and update these trajectories is described in Section III-C. For the moment we assume them to be given. Similar to [17], we use

$$g_{ij}^k(x_i^k, x_j^k) = {\eta_{ij}^k}^\top \left[ (x_i^k - x_j^k) - (\bar{x}_i^k - \bar{x}_j^k) \right] - \Delta_i \tag{6a}$$
$$\bar{h}_{ij}(x_i, x_j) = \min_{k=3,\ldots,N+2} g_{ij}^k(x_i^k, x_j^k), \tag{6b}$$

where $\eta_i^k = (\bar{x}_i^k - \bar{x}_j^k)/\|\bar{x}_i^k - \bar{x}_j^k\|_2$. As illustrated in Fig. 3, the linear functions $g_{ij}^k$ are used to avoid collisions at each step along the given nominal trajectories. The construction results in the convex approximation $\bar{h}_{ij}$ of the original nonconvex constraint $h_{ij}$.

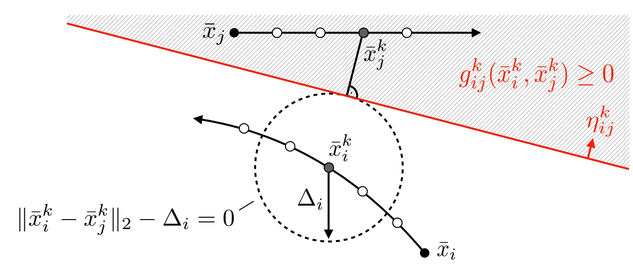


Fig. 3. Linearization around nominal trajectories at one time instance.

The problem formulation with the linearized avoidance constraint, including the concept of neighborhoods, is

$$\min_{\{x_i, v_i, a_i\}_i} \quad \sum_{i=1}^{M} J_i(x_i, a_i) \tag{7a}$$
$$\text{s.t.} \quad (x_i, v_i, a_i) \in \mathcal{C}_i \quad \forall i \tag{7b}$$
$$\bar{h}_{ij}(x_i, x_j) \geq 0 \quad \forall i \text{ and } j \in \mathcal{N}_i. \tag{7c}$$

The feasible set of the linearized problem (7) is an inner approximation of the nonconvex problem's feasible set. This curtailment causes the trajectories obtained from solving the linear problem to be more conservative. For this reason, the agent reactivity depends on the quality of the chosen linearization trajectories $(\bar{x}_i, \bar{x}_j)$.

## III. Decentralized Optimization with ADMM

In this section, we show the decentralized approach for agent coordination and we discuss several extensions.

### A. General ADMM Formulation

Following [18], ADMM is a first-order optimization method that solves problems of the form

$$\min_{\boldsymbol{z},\boldsymbol{w}} \{\, f(\boldsymbol{z}) + g(\boldsymbol{w}) \;\text{ s.t. }\; \boldsymbol{z} - \boldsymbol{w} = 0 \,\}\,, \tag{8}$$

where $\boldsymbol{z}$ is the original decision variable and $\boldsymbol{w}$ is a duplicate, allowing for the separate treatment of the objectives $f$ and $g$. As shown in Alg. 1, ADMM solves (8) by alternatingly searching for a saddle point of the augmented Lagrangian

$$\mathcal{L}_\rho(\boldsymbol{z},\boldsymbol{w},\boldsymbol{\lambda}) = f(\boldsymbol{z}) + g(\boldsymbol{w}) + \boldsymbol{\lambda}^\top \boldsymbol{\epsilon} + \tfrac{\rho}{2}\|\boldsymbol{\epsilon}\|_2^2\,, \tag{9}$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier, $\boldsymbol{\epsilon} = \boldsymbol{z} - \boldsymbol{w}$ is a residual, and $\rho$ is a user-chosen penalty parameter. For any initialization of $(\boldsymbol{z},\boldsymbol{w},\boldsymbol{\lambda})$, and any positive value of $\rho$, ADMM converges to a solution of (8), given that a solution exists and $f, g$ have nonempty, closed, and convex epigraph [18].

**Algorithm 1** ADMM

**repeat**
$\boldsymbol{z} \leftarrow \arg\min_{\boldsymbol{z}} \mathcal{L}_\rho(\boldsymbol{z},\boldsymbol{w},\boldsymbol{\lambda})$
$\boldsymbol{w} \leftarrow \arg\min_{\boldsymbol{w}} \mathcal{L}_\rho(\boldsymbol{z},\boldsymbol{w},\boldsymbol{\lambda})$
$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \rho \frac{\partial}{\partial \boldsymbol{\lambda}} \mathcal{L}_\rho(\boldsymbol{z},\boldsymbol{w},\boldsymbol{\lambda})$
**until** satisfaction of a stopping criterion

### B. ADMM for Coordination of Moving Agents

In Alg. 1, the two-group partition in (8) leads to a separate treatment of $\boldsymbol{z}$ and its duplicate $\boldsymbol{w}$. We utilize this mechanism for isolating the collision avoidance constraint (7c). Similar to [14], we achieve this by introducing duplicates of the trajectory plans. We use $w_i$, which is an agent-local trajectory duplicate, and $w_{i\to j}$, which is the $i$-th agent's duplicate of the $j$-th agent's trajectory. We also call $w_{i\to j}$ a trajectory *proposal* from $i$ to $j$. The resulting equivalent reformulation of (7) is

$$\begin{aligned}
\min_{\substack{\{x_i, w_i, v_i, a_i\}_i \\ \{w_{i\to j}\}_{i,j}}} \quad & \textstyle\sum_{i=1}^{M} J_i(x_i, a_i) && && \text{(10a)}\\
\text{s.t.} \quad & (x_i, v_i, a_i) \in \mathcal{C}_i && \forall i && \text{(10b)}\\
& \bar{h}_{ij}(w_i, w_{i\to j}) \ge 0 && \forall i \text{ and } j \in \mathcal{N}_i && \text{(10c)}\\
& w_i = x_i;\; w_{i\to j} = x_j && \forall i \text{ and } j \in \mathcal{N}_i, && \text{(10d)}
\end{aligned}$$

where the collision avoidance constraint does not depend on the original trajectories $x_i$ anymore. We write (10) in the form of (8) by associating $\boldsymbol{z}$ with the agent-individual variables $\{x_i, v_i, a_i\}_i$, and $\boldsymbol{w}$ with $\{w_i, w_{i\to j}\}_{i,j}$.

Before adapting Alg. 1 to this setting, we shift our point of view from the multi-agent problem to the perspective of agent $i$. We drop the index $i$ from $x_i$, $J_i$, $\bar{h}_{ij}$, $\mathcal{C}_i$, and $\mathcal{N}_i$. Further, we write $w_{j\to} = w_{j\to i}$ and $w_{\to j} = w_{i\to j}$. Consistent with the previous interpretation, $w_{j\to}$ is a proposal that the point-of-view agent receives from agent $j$. Conversely, $w_{\to j}$ is sent to agent $j$. Alg. 2 shows the coordination method and Prop. 1 constitutes its use for solving (7). We show details of the algorithm formulation and the proof of the proposition in Appendix V-A.

**Algorithm 2** Coordination procedure for each agent.

**update** $\mathcal{N}$ and **initialize** $\lambda$, $w$, $\{\lambda_{\to j}, \lambda_{j\to}, w_{j\to}\}_{j\in\mathcal{N}}$
**repeat**
*prediction*: update $(x, v, a)$ with
$$\begin{aligned}\arg\min_{x,v,a}\; & J(x,a) + \lambda^\top(x - w) + \tfrac{\rho}{2}\|x - w\|_2^2 \\ & + \sum_{j\in\mathcal{N}} \big(\lambda_{j\to}^\top (x - w_{j\to}) + \tfrac{\rho}{2}\|x - w_{j\to}\|_2^2\big) \\ \text{s.t.}\; & (x, v, a) \in \mathcal{C}\end{aligned}$$
⋆ *communication*: send $x$ to $j \in \mathcal{N}$; receive $\{x_j\}_{j\in\mathcal{N}}$
*coordination*: update $(w, \{w_{\to j}\}_{j\in\mathcal{N}})$ with
$$\begin{aligned}\arg\min_{w,\{w_{\to j}\}_{j\in\mathcal{N}}}\; & \lambda^\top(x - w) + \tfrac{\rho}{2}\|x - w\|_2^2 \\ & + \sum_{j\in\mathcal{N}} \big(\lambda_{\to j}^\top (x_j - w_{\to j}) + \tfrac{\rho}{2}\|x_j - w_{\to j}\|_2^2\big) \\ \text{s.t.}\; & \bar{h}_j(w, w_{\to j}) \ge 0 \quad \forall j \in \mathcal{N}\end{aligned}$$
*mediation*: update $(\lambda, \{\lambda_{\to j}\}_{j\in\mathcal{N}})$ with
$\lambda \leftarrow \lambda + \rho(x - w)$
$\lambda_{\to j} \leftarrow \lambda_{\to j} + \rho(x_j - w_{\to j}) \quad \forall j \in \mathcal{N}$
⋆ *communication*:
send $(\lambda_{\to j}, w_{\to j})$ to $j \in \mathcal{N}$; receive $\{\lambda_{j\to}, w_{j\to}\}_{j\in\mathcal{N}}$
**until** satisfaction of a stopping criterion

**Proposition 1.** *If all agents simultaneously use Alg. 2, then they jointly converge to an optimal solution of the linearized problem* (7)*, as long as a solution exists.*

Alg. 2 contains steps for *prediction*, *coordination*, and *mediation*, as well as two rounds of communication. In the prediction step, the agent plans a trajectory $x$ that satisfies its dynamics and pursues its objective. While doing so, it remains close to its collision-free trajectory $w$, as well as the trajectories $\{w_{j\to}\}_j$ that were proposed by its neighbors. The plans $x, \{x_j\}_j$ are then exchanged. In the coordination step, the agent develops strategies $w, \{w_{\to j}\}_j$ that orchestrate the neighborhood to become collision-free, while remaining close to the previous plans $x, \{x_j\}_j$. We can execute the associated optimization efficiently since it separates along the prediction horizon. In the mediation step, Lagrange multipliers $\lambda, \{\lambda_{\to j}\}_j$ accumulate the gap between plans $x, \{x_j\}_j$ and collision-avoiding trajectories $w, \{w_{\to j}\}_j$. By influencing the prediction and coordination steps, the multipliers promote a consensus between all trajectories and agents. In the final communication step, the agents exchange the Lagrange multipliers and trajectory proposals. Fig. 4 illustrates the communication procedure. For the initialization of Alg. 2, we use time-shifted trajectories from the previous

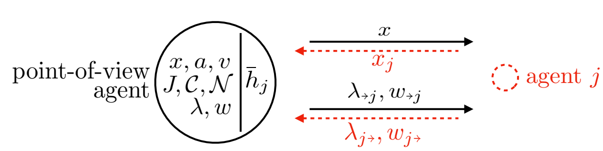


Fig. 4. Communication of the point-of-view agent with agent $j$ during one iteration of Alg. 2.

MPC call, as shown in [16]. This warm starting strategy potentially reduces the computational burden, especially in slow changing environments.

### C. Adaptive Linearization

We consider the curtailed solution space of (1) and the resulting conservative trajectory choice as the most pressing limitation of linearization-based coordination methods. We follow the rationale that the more often we update the linearization, the higher the potential for reducing conservatism. Many existing approaches [3], [9], [10] linearize around the results of the previous MPC call. For obtaining a higher adaptation rate, we instead utilize the iterative nature of ADMM to adapt the nominal trajectories in each algorithm iteration. Mod. 1 and Prop. 2 constitute the procedure. The proposition is true since the modification is without effect once $\bar{h}_j$ reaches a fixed point.

**Modification 1** Adaptive linearization for Alg. 2.

- To be added before the *coordination* step:
  update $\{\bar{h}_j\}_{j\in\mathcal{N}}$ based on (6) and $x, \{x_j\}_{j\in\mathcal{N}}$

**Proposition 2.** *If all agents use Alg. 2 with Mod. 1, and the iterations of all agents converge, then the solution is optimal for the successively adapted linearized problem* (7).

Prop. 2, as opposed to Prop. 1, does not guarantee convergence. In fact, by adapting $\bar{h}_j$ during the ADMM iteration, we change (7) while solving it. The regularization terms in the prediction and coordination updates curb the adaption process, smoothening the change. Since all of our simulations behave well, we tolerate the absence of a convergence guarantee for potentially reduced conservatism. In case a procedure is required that combines both, we use Mod. 1 for the first $K$ iterations only, which results in partially adaptive linearization while inheriting the guarantees of Prop. 1.

### D. Deadlock Protection

Even with adaptive linearization, the options of an agent can become much more limited than with a nonconvex approach. A prominent example is a head-to-head encounter of two agents, which results in a deadlock where both agents slow down and stop. Fig. 5 illustrates the situation.

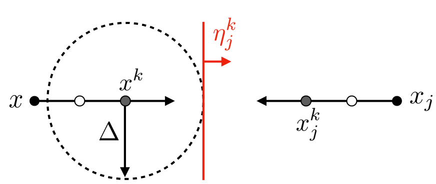


Fig. 5. Frontal deadlock situation. Due to the linearized collision avoidance constraint, there is no option for moving above or below.

We prevent a deadlock situation by modifying each local linearization $g_i^k$ through rotation of $\eta_j^k$, which preserves the integrity of the collision avoidance constraint. We use

$$\eta_j^k \leftarrow R_\varphi \eta_j^k \quad \forall k = 3, \ldots, N+2, \tag{11}$$

where the matrix $R_\varphi \in \mathbb{R}^{n\times n}$ performs a rotation by the positive angle $\varphi$. Mod. 2 describes the deadlock protection.

**Modification 2** Deadlock protection for Alg. 2 with Mod. 1.

- To be added after the *adaptive linearization* step:
  **if** $\bar{h}_j$ unchanged **and** $(x^k - x^{k-1}) \odot \eta_j^k = 0 \ \forall k$
  **then** rotate all $\eta_j^k$ by $\varphi$

In Mod. 2, we first detect whether $\bar{h}_j$ remained unchanged in the last algorithm iteration. We then use the cross product $\odot$ to determine whether the normal vectors $\{\eta_j^k\}_k$ are in parallel with the current trajectory. If this is the case, we are in a locked situation, and we break the tie as noted in (11). The released trajectories then converge around each other by themselves; therefore, the actual value of $\varphi > 0$ does not affect the result. Fig. 6 shows the release of the deadlock.

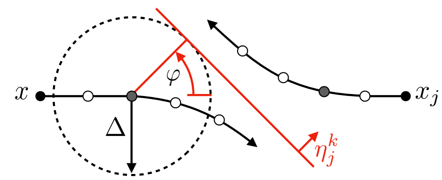


Fig. 6. Release of the locked situation through rotation of $\eta_j^k$ by $\varphi$.

### E. Local Coordinate Systems

So far, we have considered that all agents navigate in the same coordinate system, which conflicts with our notion of decentralization. Instead, it is desirable that each agent operates and communicates in local coordinates. Towards this, we assume that all agents have a common notion of orientation, e.g., they orient towards the magnetic north or any other distant fixed point. Moreover, we assume that agents can determine the relative position of their neighbors. Fig. 7 illustrates the situation.

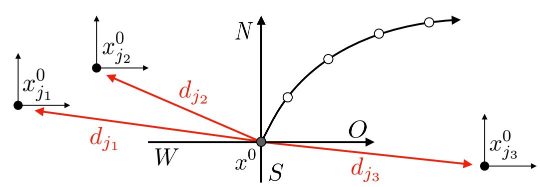


Fig. 7. Local coordinate system at the position of the point-of-view agent. The relative positions of the neighbors are shown in red.

We denote the relative position of the $j$-th neighbor as $d_j$. For parsing incoming communication to the local coordinates, we use Mod. 3. The Lagrange multipliers are translation-invariant and therefore unaffected.

**Modification 3** Local coordinates for Alg. 2.

- To be added after the first *communication* step:
  $x_j \leftarrow x_j + (d_j, d_j, \ldots, d_j) \quad \forall j \in \mathcal{N}$
- To be added after the second *communication* step:
  $w_{\to j} \leftarrow w_{\to j} + (d_j, d_j, \ldots, d_j) \quad \forall j \in \mathcal{N}$

### F. Uncooperative Agents

Agents are uncooperative if they neither send nor accept trajectory proposals. The point-of-view agent collects unresponsive neighbors in $\mathcal{N}^u$, leaving cooperative neighbors in $\mathcal{N}^c = \mathcal{N} \backslash \mathcal{N}^u$. The agent does not receive $(x_j, \lambda_{j\rightarrow}, w_{j\rightarrow})$ from its uncooperative neighbors, and it expects that the coordination request $(\lambda_{\rightarrow j}, w_{\rightarrow j})$ is not heard. For being able to handle this situation, we require that the point-of-view agent detects the current position and velocity $(x_j^0, v_j^0)$ of uncooperative agents. Assuming constant velocity, the agent then constructs an expected trajectory $x_j$ with (2a). Further, it uses $w_{j\rightarrow} = x$, i.e., it pretends that the uncooperative agent does not propose trajectory changes. Also, when coordinating its neighborhood, the point-of-view agent adds the constraint $w_{\rightarrow j} = x_j$, i.e., it assumes that it cannot affect the uncooperative agent. Fig. 8 illustrates the resulting procedure, and Mod. 4 shows the necessary changes to our algorithm.

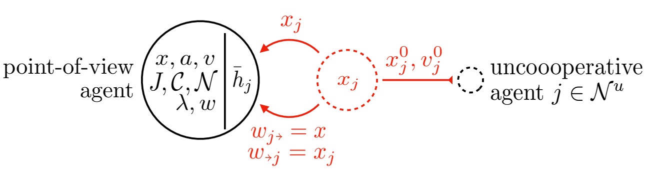


Fig. 8. Handling of an uncooperative agent as noted in Mod. 4.

**Modification 4** Information augmentation for Alg. 2.

- $+$ We execute both *communication* steps for $\mathcal{N}^c$ only
- $+$ To be added after the first *communication* step: measure $\{x_j^0, v_j^0\}_{j\in\mathcal{N}^u}$; estimate $\{x_j\}_{j\in\mathcal{N}^u}$
- $+$ To be added after the second *communication* step: $w_{j\rightarrow} \leftarrow x \quad \forall j \in \mathcal{N}^u$
- $+$ We augment the *coordination* step with the constraint: $w_{\rightarrow j} = x_j \quad \forall j \in \mathcal{N}^u$

Through Mod. 4, the Lagrange multipliers $\{\lambda_{j\rightarrow}\}_{j\in\mathcal{N}^u}$ become irrelevant. Prop. 3, proven in Appendix V-B, constitutes the use of Mod. 4.

**Proposition 3.** *We augment* (7) *with additional constraints that fix immovable trajectories of uncooperative agents. If all uncooperative agents travel at constant velocity, then Alg. 2 with Mod. 4 converges to a solution of the augmented problem, as long as a solution exists.*

### G. Algorithm Summary

We obtain the final coordination method by applying all modifications to Alg. 2. We show the result in Alg. 3, where the adaptive linearization does not hinder convergence since it only affects the first $K$ iterations. Hence, the statement of Prop. 3 also applies for Alg. 3.

## IV. Simulation Study

We consider four agents in $\mathbb{R}^2$ that approach a common intersection point. One of the agents is uncooperative. Fig. 9 shows the closed-loop trajectories where the cooperative

**Algorithm 3** Coordination procedure, including Mod. 1–4.

**update** $\mathcal{N} = \mathcal{N}^c \cup \mathcal{N}^u$, $d$, $\{d_j\}_{j\in\mathcal{N}}$
**initialize** $\lambda$, $w$, $\{\lambda_{\rightarrow j}, \lambda_{j\rightarrow}, w_{j\rightarrow}\}_{j\in\mathcal{N}}$
**repeat**

1: *prediction*: update $(x, v, a)$ with
$$\begin{aligned}\operatorname*{arg\,min}_{x,v,a}\; & J(x,a) + \lambda^\top(x-w) + \tfrac{\rho}{2}\|x-w\|_2^2 \\ & + \sum_{j\in\mathcal{N}} \left(\lambda_{j\rightarrow}^\top(x - w_{j\rightarrow}) + \tfrac{\rho}{2}\|x - w_{j\rightarrow}\|_2^2\right) \\ \text{s.t. } & (x, v, a) \in \mathcal{C}\end{aligned}$$

$\star$ *communication*: send $x$ to $j \in \mathcal{N}^c$; receive $\{x_j\}_{j\in\mathcal{N}^c}$
$x_j \leftarrow x_j + (d_j, d_j, \ldots, d_j) \quad \forall j \in \mathcal{N}^c$

$+$ *information augmentation*:
measure $\{x_j^0, v_j^0\}_{j\in\mathcal{N}^u}$; estimate $\{x_j\}_{j\in\mathcal{N}^u}$

$+$ *adaptive linearization*: (only in first $K$ iterations)
update $\{\bar{h}_j\}_{j\in\mathcal{N}}$ based on (6) and $x, \{x_j\}_{j\in\mathcal{N}}$

$+$ *deadlock protection*:
**if** $\bar{h}_j$ unchanged **and** $(x^k - x^{k-1}) \odot \eta_j^k = 0 \;\forall k$
**then** rotate all $\eta_j^k$ by $\varphi$

2: *coordination*: update $(w, \{w_{\rightarrow j}\}_{j\in\mathcal{N}})$ with
$$\begin{aligned}\operatorname*{arg\,min}_{w,\{w_{\rightarrow j}\}_{j\in\mathcal{N}}}\; & \lambda^\top(x-w) + \tfrac{\rho}{2}\|x-w\|_2^2 \\ & + \sum_{j\in\mathcal{N}} \left(\lambda_{\rightarrow j}^\top(x_j - w_{\rightarrow j}) + \tfrac{\rho}{2}\|x_j - w_{\rightarrow j}\|_2^2\right) \\ \text{s.t. } & \bar{h}_j(w, w_{\rightarrow j}) \geq 0 \quad \forall j \in \mathcal{N} \\ & w_{\rightarrow j} = x_j \quad \forall j \in \mathcal{N}^u\end{aligned}$$

3: *mediation*: update $(\lambda, \{\lambda_{\rightarrow j}\}_{j\in\mathcal{N}})$ with
$\lambda \leftarrow \lambda + \rho(x - w)$
$\lambda_{\rightarrow j} \leftarrow \lambda_{\rightarrow j} + \rho(x_j - w_{\rightarrow j}) \quad \forall j \in \mathcal{N}$

$\star$ *communication*:
send $(\lambda_{\rightarrow j}, w_{\rightarrow j})$ to $j \in \mathcal{N}^c$; receive $\{\lambda_{j\rightarrow}, w_{j\rightarrow}\}_{j\in\mathcal{N}^c}$
$w_{\rightarrow j} \leftarrow w_{\rightarrow j} + (d_j, d_j, \ldots, d_j) \quad \forall j \in \mathcal{N}^c$

$+$ *information augmentation*: $w_{j\rightarrow} \leftarrow x \quad \forall j \in \mathcal{N}^u$

**until** satisfaction of a stopping criterion

agents use Alg. 3 to determine their trajectories. The individual optimization problems are solved with Gurobi [19].

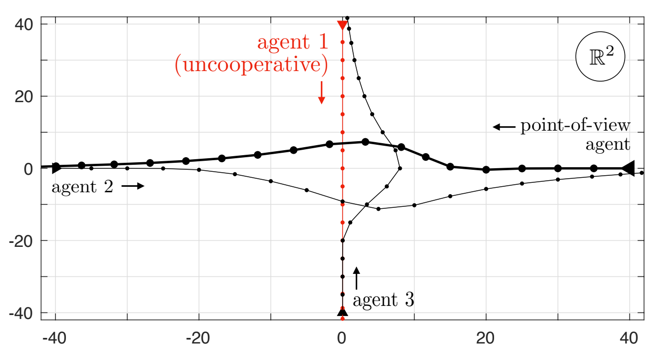


Fig. 9. Closed-loop avoidance scenario with four agents closing in around the origin. The uncooperative agent is shown in red.

In the shown situation, the agents avoid collisions while keeping their direction of travel. Fig. 10 shows one prediction from the closed loop simulation. Here, exceeding the given guarantees, Alg. 3 converges without freezing the lineariza-

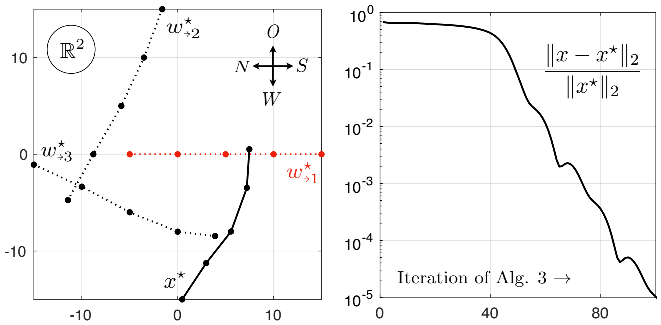


Fig. 10. Left: Converged trajectory plans in the coordinate system of the point-of-view agent when being close to the intersection point. Right: convergence of $x$ to its final value $x^\star$.

tion. Moreover, using a nonconvex solver [20], we can verify that the converged result attains a local minimum of the original problem (1). Alg. 3 reaches reasonable accuracy in 60 iterations. Surprisingly, we observe that adaptive linearization increases the average convergence speed. A reason can be that adaptation eases the linearized constraint, i.e., while moving around another agent, the constraint retreats from the planned trajectory. The resulting maneuvering room may benefit the convergence speed since ADMM is observed to be particularly slow in heavily constrained situations [21].

## V. Conclusions

We advocate a method for agent coordination that relies on adaptive linearization and results in a lightweight and therefore fast procedure. A high level of decentralization characterizes our method, which we present as a communication and action protocol from a single agent's point of view. If the adaptive linearization is frozen after a fixed number of algorithm iterations, and a feasible solution exists, then we guarantee reaching a consensus among all agents. We have discussed several extensions of our method, and we have demonstrated its viability in simulation.

## APPENDIX

### A. Formulation of Algorithm 2 and proof of Proposition 1

We use $\boldsymbol{w}_i = (w_i, w_{i \to j}, \dots)$ for $j \in \mathcal{N}_i$, $\boldsymbol{z}_i = (x_i, v_i, a_i)$, $\boldsymbol{w} = (\boldsymbol{w}_1, \dots, \boldsymbol{w}_M)$, and $\boldsymbol{z} = (\boldsymbol{z}_1, \dots, \boldsymbol{z}_M)$. A comparison with (8) yields $f(\boldsymbol{z}) = \sum_{i=1}^M \left[ J_i(x_i, a_i) + \mathcal{I}_{\mathcal{C}_i}(x_i, v_i, a_i) \right]$ and $g(\boldsymbol{w}) = \sum_{i=1}^M \left[ \sum_{j \in \mathcal{N}_i} \mathcal{I}_{\bar{h}_{ij}(w_i, w_{i \to j}) \geq 0}(\boldsymbol{w}_i) \right]$, where $\mathcal{I}$ denotes an indicator function. The augmented Lagrangian is $\mathcal{L}_\rho(\boldsymbol{z}, \boldsymbol{w}, \boldsymbol{\lambda}) = f(\boldsymbol{z}) + g(\boldsymbol{w}) + \sum_{i=1}^M \left[ M_i + \sum_{j \in \mathcal{N}_i} M_{ij} \right]$, where $\epsilon_i = x_i - w_i$, $M_i = \lambda^\top \epsilon_i + \frac{\rho}{2} \|\epsilon_i\|_2^2$, $\epsilon_{ij} = x_j - w_{i \to j}$, and $M_{ij} = \lambda^\top \epsilon_{ij} + \frac{\rho}{2} \|\epsilon_{ij}\|_2^2$. By applying Alg. 1, we obtain a basic form of Alg. 2. Due to $j \in \mathcal{N}_i \Leftrightarrow i \in \mathcal{N}_j$, it is $\sum_j M_{ij} = \sum_j M_{ji}$. Following [16], we flip the indices $i, j$ in the first algorithm step, which is necessary to separate agents. The resulting procedure converges if $f, g$ have nonempty, closed, and convex epigraph [18]. This is the case since $J_i$, $\mathcal{C}_i$, and $\bar{h}_{ij}$ are convex. ■

### B. Proof of Proposition 3

Alg. 2 with Mod. 4 resembles a convergent standard form of ADMM. For $i \in \mathcal{N}^u$, we choose constraints in (2) such that $x_j = \bar{x}_j$, where $\bar{x}_j$ is the immovable trajectory. In (10), we add the constraint $w_{i \to j} = \bar{x}_j$ for $j \in \mathcal{N}^u$, which is then included in $g$. Further, we remove $w_{j \to i}$ for $i \in \mathcal{N}^u$. Alg. 1 then produces the same iterates as Alg. 2 with Mod. 4. ■

## References


[1] J. Maciejowski, *Pred. Control with Constraints*. Prentice Hall, 2002.

[2] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires," *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 9, no. 2, pp. 41–76, 1975.

[3] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Fast ADMM for distributed model predictive control of cooperative waterborne AGVs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 4, pp. 1406–1413, Sep 2017.

[4] A. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference (ACC)*. IEEE, May 2002, pp. 1936–1941.

[5] F. Tedesco, D. M. Raimondo, and A. Casavola, "Collision avoidance command governor for multi-vehicle unmanned systems," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 16, pp. 2309–2330, Apr 2014.

[6] G. P. Roussos, G. Chaloulos, K. J. Kyriakopoulos, and J. Lygeros, "Control of multiple non-holonomic air vehicles under wind uncertainty using model predictive control and decentralized navigation functions," in *Conference on Decision and Control (CDC)*. IEEE, 2008, pp. 1225–1230.

[7] G. Chaloulos, P. Hokayem, and J. Lygeros, "Distributed hierarchical MPC for conflict resolution in air traffic control," in *American Control Conference (ACC)*. IEEE, 2010, pp. 3945–3950.

[8] ——, "Hierarchical control with prioritized MPC for conflict resolution in air traffic control," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 1564–1569, Jan 2011.

[9] H. Y. Ong and J. C. Gerdes, "Cooperative collision avoidance via proximal message passing," in *American Control Conference (ACC)*. IEEE, Jul 2015, pp. 4124–4130.

[10] H. Zheng, R. R. Negenborn, and G. Lodewijks, "Cooperative distributed collision avoidance based on ADMM for waterborne AGVs," in *International Conference on Computational Logistics*. Springer, Oct 2015, pp. 181–194.

[11] T. Mercy, W. Van Loock, and G. Pipeleers, "Real-time motion planning in the presence of moving obstacles," in *European Control Conference (ECC)*. IEEE, Jun 2016, pp. 1586–1591.

[12] R. Van Parys and G. Pipeleers, "Spline-based motion planning in an obstructed 3d environment," in *20th IFAC World Congress*, Jul 2017.

[13] T. Mercy, R. Van Parys, and G. Pipeleers, "Spline-based motion planning for autonomous guided vehicles in a dynamic environment," *IEEE Transactions on Control Systems Technology*, Aug 2017.

[14] R. Van Parys and G. Pipeleers, "Online distributed motion planning for multi-vehicle systems," in *European Control Conference (ECC)*. IEEE, Jun 2016, pp. 1580–1585.

[15] ——, "Distributed MPC for multi-vehicle systems moving in formation," *Robotics and Autonomous Systems*, vol. 97, pp. 144–152, 2017.

[16] ——, "Distributed model predictive formation control with intervehicle collision avoidance," in *Asian Control Conference*, Dec 2017.

[17] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *International Conference on Intelligent Robots and Systems*. IEEE, Aug 2012, pp. 1917–1922.

[18] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[19] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com

[20] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[21] F. Rey, P. Hokayem, and J. Lygeros, "A tailored ADMM approach for power coordination in variable speed drives," *20th IFAC World Congress*, vol. 50, no. 1, pp. 7403–7408, 2017.