# Distributed Navigation with Dynamic Obstacles

Riemens, E.H.J.; Rajan, R.T.

**DOI**
10.1109/ICASSP49660.2025.10888973

**Publication date**
2025

**Document Version**
Final published version

**Published in**
Proceedings of the ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888973

# Distributed Navigation with Dynamic Obstacles

Ellen H. J. Riemens
*Signal Processing Systems, Faculty of EEMCS*
*Delft University of Technology*, Delft, The Netherlands
e.h.j.riemens@tudelft.nl

Raj Thilak Rajan
*Signal Processing Systems, Faculty of EEMCS*
*Delft University of Technology*, Delft, The Netherlands
r.t.rajan@tudelft.nl

***Abstract*—One of the key challenges for multi-agent systems is collision free navigation in an unknown environment. In this work, we propose a unified framework for joint localization, control, and collision avoidance of multi-agent systems navigating in an unknown environment in the presence of dynamic obstacles. The cooperative agents rely on information from immediate neighboring agents within their communication neighborhood, and the dynamic obstacles are modelled as non-cooperative agents. The agents achieve localization by exploiting the individual agent dynamics, and pairwise distance measurements with agents in the sensing neighborhood of each cooperative agent. To ensure collision-free navigation, we exploit a Model Predictive Control (MPC) for each agent, with avoidance constraints using safety radius between pairwise agents. Futhermore, to avoid single point of failure, we propose Cooperative Positioning, Control and Collision Avoidance (CPCCA), which is based on distributed Method of Multipliers methods. We validate our framework and algorithms through simulations, demonstrating its effectiveness in real world scenarios, and propose directions for future work.**

***Index Terms*—Multi-agent systems, Collision avoidance, ADMM, MPC, Distributed optimization.**

## I. Introduction

The past decade has seen a rise in the application of multi-agent systems (MAS) in various sectors, including aerospace, robotics, automotive, and aviation, to name a few. The positioning of a network of MAS have been well studied for both anchored [1], [2], and anchorless scenarios [3]. These localization algorithms typically take in measurements such as Received Signal Strength (RSS), Time of Arrival (ToA), Time Difference of Arrival (TDoA), and may use information from on-board sensors e.g., IMU [4], [5]. Maximum likelihood and convex optimization approaches to estimate stationary agent positions from pairwise distance measurements (e.g., from ToA) with full or time-varying (nonsymmetric) connectivity have been proposed [6], [7].

In case of mobile MAS, there are various strategies explored in literature to track the state of the mobile agent [8], including the exploitation of the agent dynamics, which leads to adaptive parametric filtering methods e.g., distributed Kalman filtering [9]. Alternatively, non-parametric methods such as particle filtering or MCMC methods [10], [11] could be used for non-linear models or when no information on the underlying statistical noise is available. Multi-target tracking uses message-passing and belief propagation algorithms suited to nonlinear and non-Gaussian models, for both known and unknown numbers of targets have been studied [12], [13]. More recently, relative kinematics of mobile MAS were modeled based on time-varying pairwise distance measurements, inferring positions and higher-order kinematics in conjunction with accelerometer information available onboard [14], [15].

Collision-free navigation in MAS have been modeled in various ways [16], from simple swarming behavior [17] to optimal behaviors such a using a model predictive control (MPC) with collision avoidance constraints [18]. Along similar lines, dynamic obstacles have recently been included as non-cooperative agents and solved using ADMM-based algorithms [19]. However, to the best of our knowledge, the joint estimation of agent positions, the dynamic prediction of optimal future positions for both cooperative and non-cooperative agents, in combination with collision avoidance has not been addressed before, which is our key contribution in this work.

In this article, we propose a novel framework for localization, control, and collision avoidance of multi-agent systems navigating in an unknown environment in the presence of dynamic obstacles. Our framework models dynamic obstacles as non-cooperative agents and uses MPC in conjunction with various constraints to ensure safe navigation. Our proposed Cooperative, Positioning, Control and Collision Avoidance (CPCCA) algorithm is decentralized and scalable, even with increasing number of agents or multiple obstacles. We demonstrate the robustness of our solution through simulations, and propose directions for future work.

## II. Problem formulation

Consider a set $\mathcal{K} = \{1, 2, \ldots, K\}$ of $K$ agents in a $D$ dimensional space, which consist of a set of cooperative mobile agents, and non-cooperative agents or obstacles. Thus, $\mathcal{K} \triangleq \mathcal{K}^c \cup \mathcal{K}^{nc}$, where $\mathcal{K}^c$ and $\mathcal{K}^{nc}$ denote cooperative and non-cooperative agents respectively. The cooperative agents are capable of bidirectional communication with other agents, which is represented by the communication graph $\mathcal{G}^c(\mathcal{K}^c, \mathcal{E}^c)$, where $\mathcal{E}^c$ denote the set of communication edges between the agents. A communication edge between an agent pair $(k, j) \in (\mathcal{K}^c \times \mathcal{K}^c)$ exists, if and only if the pairwise distance is less than communication radius $r_c$. In addition, a directed sensing graph $\mathcal{G}^s(\mathcal{K}, \mathcal{E}^s)$ is established between all agents, where $\mathcal{E}^s$ is the set of sensing edges. A sensing edge between an agent pair $(k, j) \in (\mathcal{K}^c \times \mathcal{K})$ exists, if an only if the

This work is partially funded by TUD CRANES (Cooperative Relative Navigation of Multi-agent Systems) and by the European Commission HORIZON-KDT-JU-2023-2-RIA ShapeFuture projects.

pairwise agent distance is less than the sensing radius $r_s$. The sensing graph enables measurements of nearby agents, while the communication graph allows for cooperative distributed computation. The end goal of the agents is to reach their respective individual targets or way points, while avoiding obstacles on their way.

### A. Dynamics and measurement model

Let $\mathbf{p}_k^t \in \mathbb{R}^{D\times 1}$ denote the position at $t > 0$ of the $k$th agent $\forall k \in \mathcal{K}$, we then consider a scenario where the agents follow single integrator dynamics with underlying zero-mean Gaussian noise, and subsequently the posterior probability distributions with discrete state transitions are given by

$$p(\mathbf{p}_k^t|\mathbf{p}_k^{t-1}, \mathbf{u}_k^{t-1}) \sim \mathrm{N}(\mathbf{F}\mathbf{p}_k^{t-1} + \mathbf{G}\mathbf{u}_k^{t-1}, \mathbf{\Sigma}_k)\forall k \in \mathcal{K}^c, \quad (1a)$$

$$p(\mathbf{p}_k^t|\mathbf{p}_k^{t-1}, \mathbf{v}_k^{t-1}) \sim \mathrm{N}(\bar{\mathbf{F}}\mathbf{p}_k^{t-1} + \bar{\mathbf{G}}\mathbf{v}_k^{t-1}, \mathbf{\Sigma}_k)\forall k \in \mathcal{K}^{nc}, \quad (1b)$$

where the $D$-dimensional vectors $\mathbf{u}_k^t$ and $\mathbf{v}_k^t$ denote the control input of the cooperative agent and the velocity of a non-cooperative agent, respectively. The state transition matrices are assumed to be known, along with the noise covariance $\mathbf{\Sigma}_k \forall\ k \in \mathcal{K}$. Observe that the control inputs of cooperative agents $\mathbf{u}_k^t \forall k \in \mathcal{K}^c$ need to be estimated, however, the initial velocity of non-cooperative agents can be estimated from their respective previous positions. For a given time instant $t$, we collect all the positions and control inputs of the agents in the matrices $\mathbf{P}^t = [\mathbf{p}_1^t, \dots, \mathbf{p}_{|\mathcal{K}|}^t] \in \mathbb{R}^{D\times K}$ and $\mathbf{U}^t = [\mathbf{u}_1^t, \dots, \mathbf{u}_{|\mathcal{K}^c|}^t] \in \mathbb{R}^{D\times|\mathcal{K}^c|}$ respectively. In addition to the dynamical model, we measure the noisy relative distance measurements between the agents i.e.,

$$w_{kj}^t = \left\|\mathbf{p}_k^t - \mathbf{p}_j^t\right\|^2 + \xi_{kj} \quad \forall\, (k,j) \in \mathcal{E}^s, \quad (2)$$

where the underlying noise is Gaussian i.e., $\xi_{kj} \sim \mathrm{N}(0, \sigma_k^2)$, where the agent dependent variance $\sigma_k^2$ is known.

Given the initial positions $\mathbf{P}^0$ and the relative distance measurements to their neighboring agents, the goal of all cooperative agents is to navigate to their respective individual target position $\mathbf{p}_k^* \forall k \in \mathcal{K}^c$, by estimating their respective positions $\mathbf{P}^t$ and their control input $\mathbf{U}^t$ at a discrete time $t > 0$, and avoid (non-) cooperative obstacles during their course. In addition to the system model, we assume the sensing and communication radii $\{r_s, r_c\}$ are known to all agents.

### B. Localization

Given the dynamical model (1) and relative distance measurements (2), we propose to estimate the discrete positions of the agents at time $t > 0$ by minimizing the sum of the corresponding negative log posterior distributions i.e.,

$$J_1(\mathbf{P}^t) := -\Bigg\{ \sum_{k\in\mathcal{K}^c} \ln p(\mathbf{p}_k^t|\mathbf{p}_k^{t-1}, \mathbf{u}_k^{t-1}) + \\ + \sum_{k\in\mathcal{K}^{nc}} \ln p(\mathbf{p}_k^t|\mathbf{p}_k^{t-1}, \mathbf{v}_k^{t-1}) + \sum_{(k,j)\in\mathcal{E}^s} \ln p(\mathbf{p}_k^t, \mathbf{p}_j^t|\mathbf{w}_{kj}^t) \Bigg\}, \quad (3)$$

Now, since each individual log posterior is quadratic and convex, $J_1(\mathbf{P}^t)$ is also convex and hence can be readily solved.

### C. Model Predictive Control with Collision avoidance

In addition to the agent positions, we also need to estimate the optimal control actions for cooperative agents, while for non-cooperative agents, a constant velocity is assumed to predict future positions. The control inputs are typically obtained using finite-horizon Model Predictive Control (MPC), where the optimal actions are calculated based on the projected future states over a given control horizon [20]. Let $\mathcal{T} = \{1, \dots, T\}$ be a set of discrete time indices over control horizon $T$, then our objective is to estimate all control inputs $\tilde{\mathcal{U}}^t := \{\mathbf{U}^{t+\tau-1}\ \forall\ \tau\ \in \mathcal{T}\}$ at $t$. Since the positions over the horizon are fully defined by the position at $t$ and these control inputs, we have the following objective function

$$J_2(\tilde{\mathcal{U}}^t) := \frac{1}{T} \sum_{k\in\mathcal{K}^c} \sum_{\tau\in\mathcal{T}} ||\mathbf{p}_k^{t+\tau} - \mathbf{p}_k^*||^2 \quad \forall k \in \mathcal{K}^c, \quad (4)$$

where $\mathbf{p}_k^*$ is the target location for each cooperative agent, and $\mathbf{p}_k^{t+\tau}$ is given by the linear state transition (1a) i.e.,

$$\mathbf{p}_k^{t+\tau} = \mathbf{F}\mathbf{p}_k^{t+\tau-1} + \mathbf{G}\mathbf{u}_k^{t+\tau-1} \quad \forall\, k \in \mathcal{K}^c, \tau \in \mathcal{T}. \quad (5)$$

To ensure a feasible solution, we bound the control effort for each agent $k$ i.e.,

$$||\mathbf{u}_k^{t+\tau-1}|| \le u_{\max} \quad \forall\, k \in \mathcal{K}^c,\, \tau \in \mathcal{T}, \quad (6)$$

where $u_{\max}$ is the maximum control effort. In addition, to ensure a collision-free path, we aim to maintain a safety radius of $r_{\text{safe}}$ for all agents. Let $\mathbf{s}_{kj}^\tau = \mathbf{p}_j^{t+\tau} - \mathbf{p}_k^{t+\tau}$ be the relative position at $\tau$ between an agent pair $(k, j)$, then we have the following constraint i.e., $||\mathbf{s}_{kj}^\tau|| \ge r_{\text{safe}}$, $\forall (k,j) \in \mathcal{E}, \tau \in \mathcal{T}$. Unlike (6), since this constraint is nonconvex [21], we introduce a first-order approximation of $||\mathbf{s}_{kj}^\tau||$, which linearises the constraint while guaranteeing the trajectories to be collision-free [19], [22] i.e.,

$$||\bar{\mathbf{s}}_{kj}^\tau|| + \left(\frac{\bar{\mathbf{s}}_{kj}^\tau}{||\bar{\mathbf{s}}_{kj}^\tau||}\right)^\top (\mathbf{s}_{kj}^\tau - \bar{\mathbf{s}}_{kj}^\tau) \ge r_{\text{safe}}, \quad (7)$$

where the linearization points $\bar{\mathbf{s}}_{kj}^\tau$ are relative positions corresponding to the approximated individual trajectory points $\bar{\mathbf{p}}_k^{t+\tau}\ \forall\ k \in \mathcal{K}, \tau \in \mathcal{T}$. The approximation of the trajectory points can be done in several ways, such as by assuming constant velocity or by predicting over the horizon at time $t-1$. Due to underlying state transition noise, the safety radius $r_{\text{safe}}$ may not be guaranteed, and thus must be chosen conservatively depending on application. In summary, to achieve collision-free movement towards a target, given their position at $t$, the cost function (4) must be minimized under constraints (5), (6) and (7), which gives $\tilde{\mathcal{U}}^t$ and subsequently $\tilde{\mathcal{P}}^t := \{\mathbf{P}^{t+\tau-1}\ \forall\ \tau\ \in \mathcal{T}\}$.

## III. Joint localization, control and avoidance

We now propose a multi-objective joint localization, MPC optimization and collision avoidance formulation by combining the cost functions of localization (3) and MPC (4) i.e.,

$$\min_{\mathbf{P}^t, \tilde{\mathcal{U}}^t} \alpha J_1(\mathbf{P}^t) + (1-\alpha) J_2(\tilde{\mathcal{U}}^t) \text{ s.t. } (5), (6), (7), \quad (8)$$

where we introduce the hyperparameter $\alpha \in \mathbb{R}$ s.t. $0 < \alpha < 1$ which weighs the trade-off between the two objectives (3) and (4). Observe that the combined optimization problem consists of quadratic objectives, quadratic constraints, and linear constraints, which is readily solvable by a Second Order Cone Programming (SOCP) under nominal conditions [23], [24].

### A. Distributed formulation

To avoid a single point of failure in a multi-agent network, we now present a distributed formulation to solve (8). Observe that the cost functions have both vertex dependent variables for each agent $k \in \mathcal{K}^c$, and edge dependent variables for their neighbors. Now consider a shared vector $\mathbf{z}_{kj}^t \in \mathbb{R}^{(2(T+1))D}$, defined for each pair $(k,j) \in \mathcal{E}^s$ i.e.,

$$\mathbf{z}_{kj}^t := [(\mathbf{p}_k^t)^\top, \dots, (\mathbf{p}_k^{t+T})^\top, (\mathbf{p}_j^t)^\top, \dots, (\mathbf{p}_j^{t+T})^\top]^\top, \quad (9)$$

and let the local copies of this shared vector $\mathbf{z}_{kj}^t$ at agent $k$ and $j$ be given by $\mathbf{y}_{kj,k}^t$ and $\mathbf{y}_{kj,j}^t$, respectively. If a cooperative agent does not receive the necessary $\mathbf{y}_{kj,j}^t$, e.g., from a non-cooperative neighboring agent $j \in \mathcal{K}^{nc}$, then the agent implements an augmentation strategy by substituting its own local estimate i.e., $\mathbf{y}_{kj,j}^t = \mathbf{y}_{kj,k}^t$. Now, for a given agent $k \in \mathcal{K}^c$, we define

$$\mathbf{Y}_k^t := [\mathbf{y}_{k1,k}^t, \dots, \mathbf{y}_{k|\mathcal{N}_k|,k}^t], \; \tilde{\mathbf{U}}_k^t := [\mathbf{u}_k^t, \dots, \mathbf{u}_k^{t+T-1}] \quad (10)$$

where $\mathbf{Y}_k^t \in \mathbb{R}^{(2(T+1))D \times \mathcal{N}_k}$ denotes $\mathbf{y}_{kj,k}$ w.r.t. all the agents in the neighbourhood of agent $k$ and $\tilde{\mathbf{U}}_k^t$ are the control inputs for the agent over the horizon.

Given the definitions (9) and (10), the centralized objective in (8) can be rewritten for the $k$th agent as

$$\begin{aligned} J_k(\tilde{\mathbf{U}}_k^t, \mathbf{Y}_k^t) &:= -\alpha \Bigg\{ \sum_{j \in \mathcal{N}_k^c} \ln p(\mathbf{p}_j^t | \mathbf{p}_j^{t-1}, \mathbf{u}_j^{t-1}) \\ &+ \sum_{j \in \mathcal{N}_k^{nc}} \ln p(\mathbf{p}_j^t | \mathbf{p}_j^{t-1}, \mathbf{v}_j^{t-1}) + \sum_{j \in \mathcal{N}_k} \ln p(\mathbf{p}_k^t, \mathbf{p}_j^t | \mathbf{w}_{kj}^t) \Bigg\} \\ &+ (1-\alpha) \frac{1}{T} \sum_{\tau \in \mathcal{T}} ||\mathbf{p}_k^{t+\tau} - \mathbf{p}_k^*||^2 \quad \forall k \in \mathcal{K}^c, \end{aligned} \quad (11)$$

where $\mathcal{N}_k^c$ and $\mathcal{N}_k^{nc}$ denote the set of cooperative and non-cooperative neighbors respectively, and the overall distributed optimization problem is

$$\begin{aligned} \min_{\{\tilde{\mathbf{U}}_k^t, \mathbf{Y}_k^t\}_{k \in \mathcal{K}^c}} \quad & \sum_{k \in \mathcal{K}} J_k(\tilde{\mathbf{U}}_k^t, \mathbf{Y}_k^t) \\ \text{s.t.} \quad & \mathbf{y}_{kj}^k - \mathbf{z}_{kj} = 0, \mathbf{y}_{kj}^j - \mathbf{z}_{kj} = 0 \\ & \tilde{\mathbf{U}}_k^t \text{ satisfies (5), (6)} \quad \forall\, k \in \mathcal{K}^c \\ & \mathbf{z}_{kj} \text{ satisfies (7)} \quad \forall\, (k,j) \in \mathcal{E}_s. \end{aligned} \quad (12)$$

### B. CPCCA

Note that (12) is separable since only local and neighborhood information is required, and can be solved using conventional distributed algorithms e.g., ADMM [25]. Let $\rho > 0$ be a known penalty, and let $\boldsymbol{\lambda}_{kj,k}^t$ and $\boldsymbol{\lambda}_{kj,j}^t$ be the

**Algorithm 1** CPCCA (for agent $k \in \mathcal{K}^c$ at $t > 0$)

**Input:** $w_{k,j} \forall j \in \mathcal{N}_k$
**Initialize:** $\mathbf{z}_{kj}^0 \leftarrow \mathbf{0}$, $\boldsymbol{\lambda}_{kj}^{k(0)} \leftarrow \mathbf{0}$, $\boldsymbol{\lambda}_{kj}^{j(0)} \leftarrow \mathbf{0}$, $i \leftarrow 0$
**repeat**
    $(\mathbf{Y}_k^t, \tilde{\mathbf{U}}_k^t)^{i+1} \leftarrow \min \mathcal{L}_k^t$ in (13)
    Communicate $(\mathbf{y}_{kj}^k)^{i+1}$ to all neighbors $j \in \mathcal{N}_k$
    **if** $j \in \mathcal{N}_k^c$ **then**
        Receive $(\mathbf{y}_{kj,j}^t)^{i+1}$
    **else**
        $(\mathbf{y}_{kj,j}^t)^{i+1} = (\mathbf{y}_{kj,k}^t)^{i+1}$
    **end if**
    Calculate $\mathbf{z}_{kj}^{(i+1)} = 0.5\big[(\mathbf{y}_{kj,k}^t)^{i+1} + (\mathbf{y}_{kj,j}^t)^{i+1}\big]$
    Update $(\boldsymbol{\lambda}_{kj,k}^t)^{i+1} = (\boldsymbol{\lambda}_{kj,k})^i + \rho\big[(\mathbf{y}_{kj,k}^t)^{i+1} - \mathbf{z}_{kj}^{i+1}\big]$
    Update $(\boldsymbol{\lambda}_{kj,j})^{i+1} = (\boldsymbol{\lambda}_{kj,j}^t)^i + \rho\big[(\mathbf{y}_{kj,j}^t)^{i+1} - \mathbf{z}_{kj}^{i+1}\big]$
    $i \leftarrow i+1$
**until** convergence
**Output:** $\mathbf{Y}_k^t, \tilde{\mathbf{U}}_k^t$

Lagrange multipliers at agent $k \in \mathcal{K}^c$ and $j \in \mathcal{N}_k$ respectively, then the Augmented Lagrangian for each agent $k$ at time $t > 0$ can be expressed as

$$\begin{aligned} \mathcal{L}_k^t &:= J_k(\tilde{\mathbf{U}}_k^t, \mathbf{Y}_k^t) \\ &+ \sum_{j \in \mathcal{N}_k} (\boldsymbol{\lambda}_{kj,k}^t)^\top (\mathbf{y}_{kj,k}^t - \mathbf{z}_{kj}^t) + (\boldsymbol{\lambda}_{kj,j}^t)^\top (\mathbf{y}_{kj,j}^t - \mathbf{z}_{kj}^t) \\ &+ 0.5\rho(||\mathbf{y}_{kj,k}^t - \mathbf{z}_{kj}^t||^2 + ||\mathbf{y}_{kj,j}^t - \mathbf{z}_{kj}^t||^2), \end{aligned} \quad (13)$$

which can be solved iteratively until convergence is reached, as summarized in Algorithm 1, which we call Cooperative Positioning, Control and Collision Avoidance (CPCCA). The algorithm converges to a feasible solution, unless in the occurrence of a deadlock [26] [27], which is a scarce phenomenon, due to underlying measurement noise in practise.

The computational complexity for solving the centralized problem is typically cubic in the number of variables: $\mathcal{O}((K \times D \times T)^3)$. In the distributed setting, this reduces to $\mathcal{O}(((1+|\mathcal{N}_k|) \times D \times T)^3)$ computed in parallel by each agent, at every iteration.

## IV. Simulations

We conduct experiments to evaluate the performance of our proposed distributed solution for joint positioning, control, and collision avoidance. The code is available in https://github.com/asil-lab/EHJR_CPCCA. We consider a network of $K = 8$ agents, in an intersection scenario. The objective is for all agents to reach their respective target locations $\mathbf{P}^*$, marked with a star from their initial positions $\mathbf{P}^0$, marked with a cross in Figure 1a.

We employ Algorithm 1 for all agents individually, for 100 steps. After every step $t$, the resulting control input is given to the system and executed. The trade-off parameter to weigh $J_1$ and $J_2$ is chosen as $\alpha = 0.99$, and the collision radius is set to $r_{\text{safe}} = 7$. The sensing and communication radius are equal: $r_s = r_c = 30$. The maximum permitted control effort is set to $u_{\max} = 1.5$, the control horizon is $T = 15$. The state transition

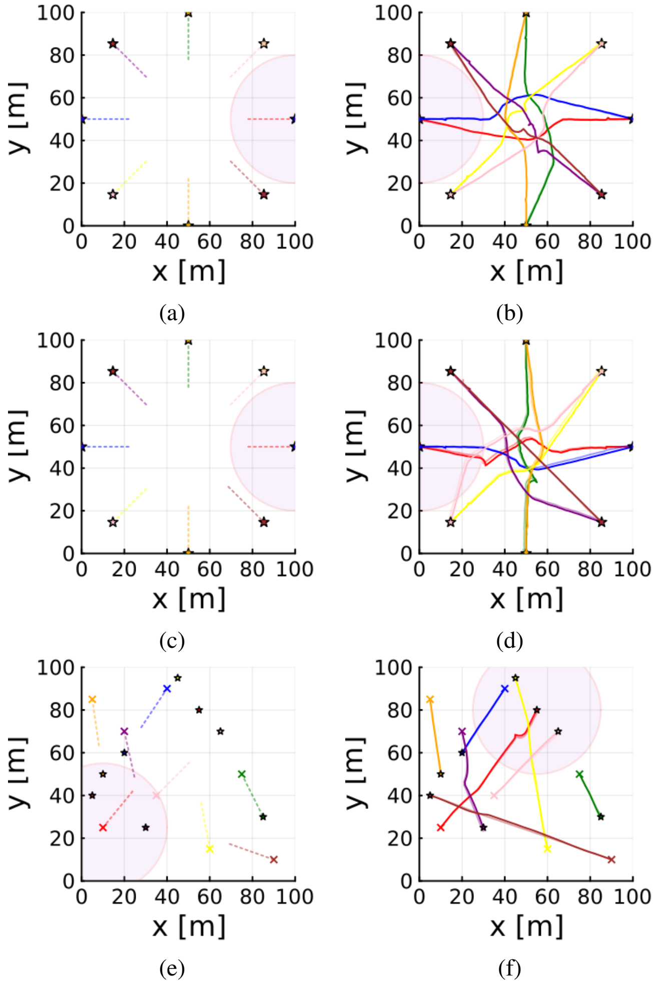


Fig. 1: Behavior of $K = 8$ agents in different scenarios, from initial (left) to final positions (right). Dotted lines indicate the planned trajectory over the horizon. (a) Shows the $K = 8$ cooperative agents at their initial positions, aiming to cross the intersection. (b) shows the agents safely past the intersection to their respective goals. (c) and (d) show a similar configuration, but with one agent (in brown) being non-cooperative. The non-cooperative agent crosses the intersection in a straight line without adapting to others. Again the agents reach their respective goals safely. In (e), the agents are spread out at random initial points with random goals. In (f) we see all agents reaching their goals, including the non-cooperative agent.

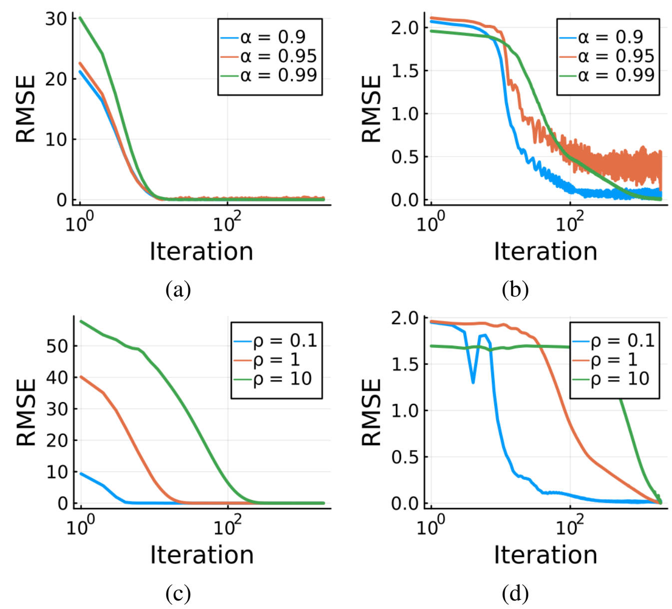


Fig. 2: Convergence rate of Algorithm 1 for different hyper-parameter values. (a) and (b) show the convergence of $\mathbf{P}^t$ and $\mathbf{U}^t$ respectively for different values of $\alpha$ at $\rho = 0.5$. (c) and (d) show the convergence for different values of $\rho$ with $\alpha = 0.99$.

matrices are taken as $\mathbf{F} = \bar{\mathbf{F}} = \mathbf{I}$ and $\mathbf{G} = \bar{\mathbf{G}} = \begin{bmatrix}1\\1\end{bmatrix}$. The noise variance levels are $\sigma_k^2 = 0.2u_{\max}\mathbf{I}, \mathbf{\Sigma}_k = 0.1$, and the penalty parameter for ADMM is $\rho = 0.1$. The optimization problems are solved with Ipopt.jl [28] in `JuMP` [29].

First, we show the trajectories obtained using the proposed algorithm with cooperative agents in Figure 1b. The shaded circle visualizes the communication radius for one agent. Then, one of the agents is non-cooperative, in the same initial setting displayed in 1c. In 1d, we see that the non-cooperative agent in brown crossing the intersection, while other agents avoid collision. In the next scenario, we initialize the agents at random locations in the environment, with corresponding random goals in 1e. Again, the agent in brown is non-cooperative and all agents reach the targets in 1f.

We investigate the influence of hyperparameters $\alpha$ and $\rho$ on the convergence rate. In Figure 2a and 2b, we see the convergence rate separately of $\mathbf{P}^t$ and $\mathbf{U}^T$. With the increase of $\alpha$, the value of $\mathbf{P}^t$ converges marginally slower. $\mathbf{U}^t$ on the other hand, converges slower at higher values of $\alpha$, but shows more stable behavior. We show the convergence rate for different values of $\rho$ in Figure 2c and 2d. For a value of $\rho = 0.1$, the algorithm converges quickly to the final value. With the increase of $\rho$, the convergence rate decreases drastically, for both $\mathbf{P}^t$ and $\mathbf{U}^T$. From this, we conclude that low values of $\rho$ are preferred. Benefits of higher values of $\rho$ could include more stable convergence behavior.

## V. Conclusion

This paper introduced a novel distributed framework for enhancing the navigation of multi-agent systems in dynamic environments with obstacles. The proposed Cooperative Positioning, Control and Collision Avoidance (CPCCA) algorithm uses Model Predictive Control alongside distributed optimization techniques to achieve collision-free navigation in the presence of non-cooperative agents. Future work will focus on a deeper analytical exploration of the CPCCA framework. Furthermore, we aim to investigate alternative optimization methods to ADMM for distributed environments, and explore the incorporation of non-linear state space models.

## References


[1] F. Yin, Z. Lin, Q. Kong, Y. Xu, D. Li, S. Theodoridis, and S. R. Cui, "FedLoc: Federated Learning Framework for Data-Driven Cooperative Localization and Location Data Processing," *IEEE Open Journal of Signal Processing*, vol. 1, pp. 187–215, 2020.

[2] R. M. Vaghefi and R. M. Buehrer, "Cooperative Joint Synchronization and Localization in Wireless Sensor Networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 14, pp. 3615–3627, Jul. 2015.

[3] U. Ferner, H. Wymeersch, and M. Z. Win, "Cooperative anchor-less localization for large dynamic networks," in *2008 IEEE International Conference on Ultra-Wideband*, vol. 2. IEEE, 2008, pp. 181–185.

[4] N. Patwari, A. Hero, M. Perkins, N. Correal, and R. O'Dea, "Relative location estimation in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[5] V. Sneha and M. Nagarajan, "Localization in wireless sensor networks: a review," *Cybernetics and Information Technologies*, vol. 20, no. 4, pp. 3–26, 2020.

[6] A. Simonetto and G. Leus, "Distributed Maximum Likelihood Sensor Network Localization," *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1424–1437, Mar. 2014.

[7] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 2, pp. 120–136, 2016.

[8] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6019–6039, 2022.

[9] L. Lu, J. Hu, Q. Pan, C. Zhao, Z. Xu, and C. Jia, "Decentralized localization for multi-agent systems based on asynchronous communication," in *2021 40th Chinese Control Conference (CCC)*. IEEE, 2021, pp. 5489–5494.

[10] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Distributed particle filtering in agent networks: A survey, classification, and comparison," *IEEE Signal Processing Magazine*, vol. 30, no. 1, pp. 61–81, Jan. 2013.

[11] R. Tang, E. Riemens, and R. T. Rajan, "Distributed Particle Filter Based on Particle Exchanges," in *2023 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2023, pp. 1–8.

[12] F. Meyer, T. Kropfreiter, J. L. Williams, R. Lau, F. Hlawatsch, P. Braca, and M. Z. Win, "Message Passing Algorithms for Scalable Multitarget Tracking," *Proceedings of the IEEE*, vol. 106, no. 2, pp. 221–259, Feb. 2018.

[13] F. Meyer, O. Hlinka, H. Wymeersch, E. Riegler, and F. Hlawatsch, "Distributed Localization and Tracking of Mobile Networks Including Noncooperative Objects," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 2, no. 1, pp. 57–71, Mar. 2016.

[14] R. T. Rajan, G. J. T. Leus, and A. J. van der Veen, "Relative kinematics of an anchorless network," *Signal Processing*, vol. 157, 2019.

[15] A. Mishra and R. T. Rajan, "Relative Kinematics Estimation Using Accelerometer Measurements," in *2022 30th European Signal Processing Conference (EUSIPCO)*, Aug. 2022, pp. 1856–1860.

[16] F. Chen, W. Ren *et al.*, "On the control of multi-agent systems: A survey," *Foundations and Trends® in Systems and Control*, vol. 6, no. 4, pp. 339–499, 2019.

[17] R. Mach and F. Schweitzer, "Multi-agent Model of Biological Swarming," in *Advances in Artificial Life*, W. Banzhaf, J. Ziegler, T. Christaller, P. Dittrich, and J. T. Kim, Eds. Berlin, Heidelberg: Springer, 2003, pp. 810–820.

[18] L. Ferranti, L. Lyons, R. R. Negenborn, T. Keviczky, and J. Alonso-Mora, "Distributed Nonlinear Trajectory Optimization for Multi-Robot Motion Planning," *IEEE Transactions on Control Systems Technology*, pp. 1–16, 2022.

[19] F. Rey, Z. Pan, A. Hauswirth, and J. Lygeros, "Fully Decentralized ADMM for Coordination and Collision Avoidance," in *2018 European Control Conference (ECC)*, Jun. 2018, pp. 825–830.

[20] B. Alrifaee, M. G. Mamaghani, and D. Abel, "Centralized non-convex model predictive control for cooperative collision avoidance of networked vehicles," in *2014 IEEE International Symposium on Intelligent Control (ISIC)*, Oct. 2014, pp. 1583–1588.

[21] S. Boyd and L. Vandenberghe, "Semidefinite Programming Relaxations of Non-Convex Problems in Control and Combinatorial Optimization," in *Communications, Computation, Control, and Signal Processing*, A. Paulraj, V. Roychowdhury, and C. D. Schaper, Eds. Boston, MA: Springer US, 1997, pp. 279–287.

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

[23] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear Algebra and its Applications*, vol. 284, no. 1, pp. 193–228, Nov. 1998.

[24] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, Jun. 2016.

[25] 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, Jul. 2011.

[26] A. Salimi Lafmejani and S. Berman, "Nonlinear MPC for collision-free and deadlock-free navigation of multiple nonholonomic mobile robots," *Robotics and Autonomous Systems*, vol. 141, p. 103774, Jul. 2021.

[27] N. Gafur, G. Kanagalingam, A. Wagner, and M. Ruskowski, "Dynamic Collision and Deadlock Avoidance for Multiple Robotic Manipulators," *IEEE Access*, vol. 10, pp. 55 766–55 781, 2022.

[28] 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, Mar. 2006.

[29] M. Lubin, O. Dowson, J. D. Garcia, J. Huchette, B. Legat, and J. P. Vielma, "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization," Mar. 2023.