2023 62nd IEEE Conference on Decision and Control (CDC)
December 13-15, 2023. Marina Bay Sands, Singapore

A Distributed Linear Quadratic Discrete-Time Game Approach to Formation Control with Collision Avoidance

Prima Aditya and Herbert Werner

Abstract— **Formation control problems can be expressed as linear quadratic discrete-time games (LQDTG) for which Nash equilibrium solutions are sought. However, solving such problems requires solving coupled Riccati equations, which cannot be done in a distributed manner. A recent study showed that a distributed implementation is possible for a consensus problem when fictitious agents are associated with edges in the network graph rather than nodes. This paper proposes an extension of this approach to formation control with collision avoidance, where collision is precluded by including appropriate penalty terms on the edges. To address the problem, a state-dependent Riccati equation needs to be solved since the collision avoidance term in the cost function leads to a state-dependent weight matrix. This solution provides relative control inputs associated with the edges of the network graph. These relative inputs then need to be mapped to the physical control inputs applied at the nodes; this can be done in a distributed manner by iterating over a gradient descent search between neighbors in each sampling interval. Unlike inter-sample iteration frequently used in distributed MPC, only a matrix-vector multiplication is needed for each iteration step here, instead of an optimization problem to be solved. This approach can be implemented in a receding horizon manner, this is demonstrated through a numerical example.**

I. INTRODUCTION

Distributed control of multi-agent (in the sense of multi-vehicle) systems has been extensively studied over the last two decades with potential applications in many areas. Formation control is one such problem that has received significant attention. In formation control, all agents in a multi-agent system must move from arbitrary initial states to attain a pre-determined geometric shape [1]. To attain and maintain the formation, the agents in the team exchange information about their positions and velocities.

When formation control schemes are implemented in a distributed manner, then in situations that involve e.g. collision avoidance, agents may have conflicting interests, and achieving their individual objectives may take precedence over cooperation. Such situations reflect non-cooperative game behavior, as agents strive to meet their goals without collaboration. The solution to this type of game is to find a Nash equilibrium, where individual agents cannot improve their payoff by changing their strategy unilaterally. Linear quadratic differential games (LQDG) have been proposed as a means of addressing this problem, where the cost of each agent is quadratic, and agent dynamics are assumed to be linear.

Prima Aditya and Herbert Werner are with Institute of Control Systems, Hamburg University of Technology, 21073, Hamburg, Germany {prima.aditya, h.werner}@tuhh.de

A formation control problem modeled as LQDG has been discussed in [2]. There, a coupled Riccati differential equation is solved to find a Nash equilibrium. A discrete-time version of LQDG, referred to as linear quadratic discrete-time game (LQDTG), is more appropriate for receding horizon implementations, and has been proposed in [3]. However, solving coupled Riccati differential or difference equations is likely to be intractable for large networks; moreover, the solution cannot be implemented in a distributed manner.

Based on an idea proposed in [4], it was shown in [5] that one can avoid solving coupled Riccati difference equations by relocating the coupling terms that initially appear in the cost function to the system dynamics. Consequently, the modified problem can be reformulated as a fictitious multi-agent system evolving on the edges of the network graph instead of the nodes, allowing for a distributed solution to the decoupled Riccati difference equations. The resulting relative control inputs associated with each edge can then be mapped back to the physical control inputs in a distributed manner by employing a distributed steepest descent iteration between agents over two sampling instants. Such intersample iteration is frequently used in distributed MPC [6]. However, unlike in distributed MPC, the approach proposed in [5] does not require solving an optimization problem at each iteration step but only involves performing a matrix-vector multiplication.

Whereas the distributed scheme proposed in [5] considers an unconstrained consensus control problem, our contribution in this article is to extend this approach to a formation control problem that includes collision avoidance among agents. We begin by formulating the problem on the graph nodes, considering the desired formation displacements together with relative constraints for collision avoidance which are represented as soft constraints in the cost function. These state-dependent collision avoidance terms in the cost lead to coupled state-dependent Riccati difference equations (SDRDE). To decouple these, we use the same idea as in [5] by relocating the coupling term from the cost function to the system dynamics on the edges of the graph. This results in a decoupled cost that still incorporates the collision avoidance term. The reformulated problem involves solving a set of decoupled SDRDEs. This can be achieved using a receding horizon technique proposed in [7], and can be implemented in a distributed manner.

The paper is organized as follows: Section II provides a review of graph theory and the formation control problem with collision avoidance. Our proposed distributed solution is outlined in Section III. Section IV showcases simulation results, and finally, Section V concludes this article.

II. PRELIMINARIES

A. Graph Theory

A graph $\mathcal{G} := (\mathcal{V}, \mathcal{E})$ consists of a set of nodes $\mathcal{V} = \{\nu^1, ..., \nu^N\}$, and a set of edges $\mathcal{E} = \{(\nu^i, \nu^j) \in \mathcal{V} \times \mathcal{V}, \nu^j \neq \nu^i\}$ which contains ordered pairs of distinct nodes. N is the number of nodes, and M is the number of edges. $\mathcal{G}$ is called undirected if $(\nu^i, \nu^j) \in \mathcal{E} \iff (\nu^j, \nu^i) \in \mathcal{E}$. An edge, denoted as $e^m := (\nu^i, \nu^j)$, indicates that agent i receives information from agent j, where m represents the number of edge (ν^i, ν^j). Let us enumerate the edge set as $\mathcal{E} = \{e^1, ..., e^M\}$, where $e^m \in \mathcal{E}$ represents the m-th edge. For $m \in \{1, ..., M\}$, let $\alpha^m \in \mathbb{R}$ be a positive scalar denoting the edge weight corresponding to the m-th edge.

The set of neighbors of agent i is denoted by $\mathcal{N}^i$. The (oriented) incidence matrix $D \in \mathbb{R}^{N \times M}$ of the graph $\mathcal{G}$ is defined component-wise by

$$D_{im} = \begin{cases} +1, & \text{if node } i \text{ is the source node of edge } e^m, \\ -1, & \text{if node } i \text{ is the sink node of edge } e^m, \\ 0, & \text{otherwise,} \end{cases}$$

where for undirected graphs the orientation in the incidence matrix can be chosen arbitrarily.

The weighted Laplacian of a graph $\mathcal{G}$ can be defined as

$$L = DWD^T,$$

where $W = \text{diag}(\alpha^1, ..., \alpha^M) \in \mathbb{R}^{M \times M}$ is a diagonal matrix of edge weights. The Laplacian matrix is symmetric and positive semi-definite. In the context of game theory, we define a local Laplacian for agent / player i as

$$L^i = DW^iD^T,$$

and let $W^i \in \mathbb{R}^{M \times M}$ be a diagonal matrix such that the m-th diagonal entry of W^i is equal to α^m if $e^m \in \mathcal{E}^i$ and zero otherwise, where $\mathcal{E}^i = \{e^i, ..., e^i_{deg_i}\} \subset \mathcal{E}$ be the set of edges incident at node $\nu^i \in V$. In this paper we assume $\alpha^m = 1$, for all $\forall m \in M$.

Assumption 1: Graph $\mathcal{G}$ is connected, i.e. there exists an undirected path between every two vertices $\nu^i,\ \nu^j \in \mathcal{V}$, $j \neq i$.

From now on, we assume that the graph used in this paper is undirected.

B. σ-Norms

The σ-norm of a vector is a map $\mathbb{R}^n \to \mathbb{R}_{\geq 0}$ *(not a norm)* defined as [8]

$$||y||_\sigma = \frac{1}{\epsilon}\left[\sqrt{1 + \epsilon||y||^2} - 1\right], \tag{1}$$

where $||\cdot||$ is an Euclidian norm in $\mathbb{R}^n$, $\epsilon > 0$ is a small scalar value, and the gradient $\sigma_\epsilon(y) = \nabla||y||_\sigma$ is

$$\sigma_\epsilon(y) = \frac{y}{\sqrt{1 + \epsilon||y||^2}} = \frac{y}{1 + \epsilon||y||_\sigma}.$$

The map $||y||_\sigma$ is differentiable everywhere. This property of the σ-norm will be used when dealing with the norm in the state-dependent weight matrix.

C. Agent Dynamics

In this article, we consider a homogeneous multi-agent system where each agent is modeled as a zero-order hold discretisation of a double integrator. Each agent is assumed to be moving in an n-dimensional plane. In the context of game theory, each agent acts as a player in the game. The single-agent discrete-time dynamics is

$$x^i_{k+1} = fx^i_k + gu^i_k, \qquad \text{for } i = 1, ..., N, \tag{2}$$

where the state vector for agent i is $x^i_k = \left[p^i_k, v^i_k\right]^T \in \mathbb{R}^{2n}$, and contains position p^i_k and velocity v^i_k at time k, with

$$f = \begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \otimes I_n \in \mathbb{R}^{2n \times 2n}, \quad g = \begin{bmatrix} \frac{\delta^2}{2} \\ \delta \end{bmatrix} \otimes I_n \in \mathbb{R}^{2n \times n}.$$

Here, u^i_k is the (acceleration) control input of agent i, and δ is the sampling time. To define the state vector for the multi-agent system, we select $x_k = \left[p^1_k, \cdots, p^N_k, 1, v^1_k, \cdots, v^N_k\right]^T \in \mathbb{R}^{2Nn+1}$. Having an entry with value 1 between the positions and velocities allows the inclusion of a formation offset term, as explained below. The multi-agent dynamics can then be represented as

$$x_{k+1} = Fx_k + \sum_{i=1}^{N} G^i u^i_k, \tag{3}$$

where

$$F = \begin{bmatrix} I_{Nn} & 0_{Nn \times 1} & \delta I_{Nn} \\ 0_{1 \times Nn} & 1 & 0_{1 \times Nn} \\ 0_{Nn \times Nn} & 0_{Nn \times 1} & I_{Nn} \end{bmatrix} \in \mathbb{R}^{(2Nn+1) \times (2Nn+1)},$$

$$G^i = \begin{bmatrix} \frac{\delta^2}{2}\hat{g}^i \\ 0_{1 \times n} \\ \delta\hat{g}^i \end{bmatrix} \in \mathbb{R}^{(2Nn+1) \times n},$$

with $\hat{g}^i = \hat{c}^i \otimes I_n \in \mathbb{R}^{Nn \times n}$, where $\hat{c}^i$ is the i-th column of the identity matrix of size N. $I_{Nn} \in \mathbb{R}^{Nn \times Nn}$ is an identity matrix. The scalar value of 1 in the matrix F corresponds to a formation offset term, which will be explained in the next subsection.

D. Formation with Collision Avoidance on the Nodes System

The problem considered in this article is formation control, i.e. all agents in a multi-agent system are supposed to move from arbitrary initial states to attain a formation (specified in terms of desired displacements d^{ij} between agents i and j), while minimizing a performance index over a finite time horizon $[0, T]$

$$J^i(U^i) = \frac{1}{2}\Big(X^T_k \mathcal{Q}^i(x_k) X_k + U^{i^T}_k \mathcal{R}^{ii} U^i_k\Big), \tag{4}$$

with the stacked state vector for the whole horizon $X_k = \left[x_{k+1}, x_{k+2}, ..., x_{k+T}\right]^T \in \mathbb{R}^{(2Nn+1)T}$ and the stacked control inputs vector $U^i_k = \left[u^i_k, u^i_{k+1}, ..., u^i_{k+T-1}\right]^T \in \mathbb{R}^{NnT}$. The state weighting matrix for each agent i is given by $\mathcal{Q}^i(x_k) = \text{blkdiag}(Q^i(x_k), ..., Q^i(x_k), Q^i_T(x_T)) \in \mathbb{R}^{(2Nn+1)T \times (2Nn+1)T}$, where $Q^i(x_k) = (Q^i_\alpha + Q^i_\beta(x_k)) \in \mathbb{R}^{(2Nn+1) \times (2Nn+1)}$ is a positive semi definite matrix, with

Q^i_α and $Q^i_\beta(x_k)$ represent the weighting matrices for formation and collision avoidance terms, respectively. The terminal weighting matrix $Q^i_T(x_T)$ has the same pattern as $Q^i(x_k)$ and can be defined by choosing arbitrary scalar weights of $\beta^i > 0$.

The control weighting matrix is $\mathcal{R}^{ii} = \text{blkdiag}(R^{ii}) \in \mathbb{R}^{NnT\times NnT}$, where $R^{ii} \in \mathbb{R}^{Nn\times Nn}$ is a positive definite matrix. Here, we assume there is no cross coupling in the input, i.e., $\mathcal{R}^{ij} = 0$, where $j \neq i$. Next, the rest of this subsection is dedicated to discussing the formulation of the first term of the cost in (4). The formation error of each agent i with collision avoidance can be expressed as

$$\Psi^i_k = \sum_{j\in\mathcal{N}^i} \Big\{ \big(||p^i_k - p^j_k - d^{ij}||^2 + ||v^i_k - v^j_k||^2\big) + \beta^i \Big(\frac{||p^i_k - p^j_k - d^{ij}||^2 + ||v^i_k - v^j_k||^2}{||p^i_k - p^j_k||^2 - r^{i^2}} \Big) \Big\}, \quad (5)$$

where r^i is the safety radius of agent i that is assumed to be the same for all $i \in N$ homogeneous agent, i.e $r^i = r$, and $\beta^i > 0$ is a tuning parameter for agent i. By the property of sum-of-squares, (5) can be transformed into a matrix form

$$\begin{aligned}
&\sum_{j\in\mathcal{N}^i} \Big\{ ||p^i_k - p^j_k||^2 - 2(p^i_k - p^j_k)^T d^{ij} + ||d^{ij}||^2 + ||v^i_k - v^j_k||^2 \\
&\quad + \frac{\beta^i ||p^i_k - p^j_k||^2}{||p^i_k - p^j_k||^2 - r^2} - \frac{2\beta^i (p^i_k - p^j_k)^T d^{ij}}{||p^i_k - p^j_k||^2 - r^2} \\
&\quad + \frac{\beta^i ||d^{ij}||^2}{||p^i_k - p^j_k||^2 - r^2} + \frac{\beta^i ||v^i_k - v^j_k||^2}{||p^i_k - p^j_k||^2 - r^2} \Big\} = \\
&p^T_k \mathcal{L}^i_\alpha p_k - 2p^T_k \mathcal{D}\mathcal{W}^i_\alpha d + d^T \mathcal{W}^i_\alpha d + v^T_k \mathcal{L}^i_\alpha v_k + p^T_k \mathcal{L}^i_\beta(x_k) p_k \\
&\quad - 2p^T_k \mathcal{D}\mathcal{W}^i_\beta(x_k) d + d^T \mathcal{W}^i_\beta(x_k) d + v^T_k \mathcal{L}^i_\beta(x_k) v_k \\
&\quad = x^T_k \big(Q^i_\alpha + Q^i_\beta(x_k)\big) x_k = x^T_k Q^i(x_k) x_k
\end{aligned}$$

where

$$Q^i_\alpha = \delta \begin{bmatrix} \mathcal{L}^i_\alpha & -\mathcal{D}\mathcal{W}^i_\alpha d & 0 \\ -(\mathcal{D}\mathcal{W}^i_\alpha d)^T & d^T \mathcal{W}^i_\alpha d & 0 \\ 0 & 0 & \mathcal{L}^i_\alpha \end{bmatrix}$$

has size $\mathbb{R}^{(2Nn+1)\times(2Nn+1)}$, with a diagonal matrix with the edge weight $\mathcal{W}^i_\alpha = W^i \otimes I_n \in \mathbb{R}^{Mn\times Mn}$. A lifted local Laplacian matrix is defined as $\mathcal{L}^i_\alpha = \mathcal{D}\mathcal{W}^i_\alpha \mathcal{D}^T \in \mathbb{R}^{Nn\times Nn}$ with $\mathcal{D} = D \otimes I_n \in \mathbb{R}^{Nn\times Mn}$ being the incidence matrix lifted to dimension n of the space in which agents are moving, and $d = \text{col}(d^{ij}) \in \mathbb{R}^{Mn}$ the column vector of desired displacements vector $d^{ij} \in \mathbb{R}^n$. The state-dependent weighting matrix is then

$$Q^i_\beta(x_k) = \delta \begin{bmatrix} \mathcal{L}^i_\beta(x_k) & -\mathcal{D}\mathcal{W}^i_\beta(x_k) d & 0 \\ -(\mathcal{D}\mathcal{W}^i_\beta(x_k) d)^T & d^T \mathcal{W}^i_\beta(x_k) d & 0 \\ 0 & 0 & \mathcal{L}^i_\beta(x_k) \end{bmatrix}$$

of size $\mathbb{R}^{(2Nn+1)\times(2Nn+1)}$, with the state-dependent Laplacian matrix defined as $\mathcal{L}^i_\beta(x_k) = \mathcal{D}\mathcal{W}^i_\beta(x_k)\mathcal{D}^T \in \mathbb{R}^{Nn\times Nn}$. The state-dependent edge weight matrix is $\mathcal{W}^i_\beta(x_k) = W^i_\beta(x_k) \otimes I_n \in \mathbb{R}^{Mn\times Mn}$, where now the m-th diagonal entry of $W^i_\beta(x_k) \in \mathbb{R}^{M\times M}$ is equal to $\dfrac{\beta^i}{||p^i_k - p^j_k||^2 - r^2}$ if $e^i \in \mathcal{E}^i$ and zero otherwise. The Laplacian matrix $\mathcal{L}^i_\beta(x_k)$ depends on the state since the diagonal edge matrix $\mathcal{W}^i_\beta(x_k)$ contains collision terms between agents i and j.

The formulation of the state vector $x_k \in \mathbb{R}^{2Nn+1}$ has been confirmed, and as a result, the state matrix F matches the dimensions of the state weighting matrix $Q^i(x_k) \in \mathbb{R}^{(2Nn+1)\times(2Nn+1)}$.

Assumption 2: The initial positions of the agents satisfy $||p^i_0 - p^j_0|| > r^i + r^j$, for all $i, j \in N$, $j \neq i$.

By adopting the same reasoning as outlined in [9], assumption 1 ensures that the term $x^T_0 Q^i(x_0) x_0$ in (4), for all $i \in N$, remains bounded. It follows that the agents operate without entering the avoidance region.

E. Nash Equilibrium and Coupled State-Dependent Riccati Equation (CSDRDE)

The formulation of the formation control problem with dynamics (3) and cost functions (4) as a game reflects the non-cooperative behavior, where each player is searching for a Nash equilibrium corresponding to its own local cost function.

Definition 1: A collection of strategies $U^{i\star}$ constitutes a Nash equilibrium if and only if the inequalities

$$J^i(U^{1\star}, ..., U^{N\star}) \leq J^i(U^{1\star}, ..., U^{i-1\star}, U^i, U^{i+1\star}, ..., U^{N\star})$$

hold for $i = 1, ..., N$.

We now formulate the first problem (for the multi-agent system running on the nodes) as follows.

Problem 1: Find local control sequences that achieve a Nash equilibrium corresponding to the local cost functions (4) over the control input sequences u^i subject to (3).

Theorem 1: An open-loop Nash equilibrium for the game defined by Problem 1 is achieved by the control sequences

$$u^{i\star}_k(x_k) = K^i_k(x_k) x_k, \quad (6)$$

where

$$K^i_k(x_k) = -R^{ii^{-1}} G^{i^T} P^i_{k+1}(x_{k+1}) \Lambda^{-1}_k F, \quad (7)$$

and $P^i_{k+1}(x_{k+1})$ is the solution to the coupled state-dependent Riccati difference equation

$$\begin{aligned} P^i_k(x_k) &= F^T P^i_{k+1}(x_{k+1}) \Lambda^{-1}_{k+1} F + Q^i(x_k) \\ &\quad + (I_N \otimes x^T_k) \Big[x^T_k \tfrac{\partial Q^i(x_k)}{\partial x^1_k}, \quad ..., \quad x^T_k \tfrac{\partial Q^i(x_k)}{\partial x^N_k} \Big]^T, \end{aligned} \quad (8)$$

which can be solved backward with $P^i_T(x_T) = Q^i_T(x_T)$. The corresponding closed-loop state trajectory is

$$x^\star_{k+1} = \Lambda^{-1}_k F x^\star_k, \quad (9)$$

where

$$\Lambda_k = \Big(I + \sum_{j=1}^{N} G^j R^{jj^{-1}} G^{j^T} P^j_{k+1}(x_{k+1}) \Big). \quad (10)$$

Proof: See appendix in [10]. ■

By looking at (8), solving for $P^i_k(x_k)$ requires information about $P^j_{k+1}(x_{k+1})$ for all $j \in N$, and thus the Riccati equation cannot be solved in a distributed way. Furthermore, it should be noted that once the Riccati equation is solved, the state feedback gains obtained in (7) are fully populated and require knowledge of all states across the network.

III. DISTRIBUTED FRAMEWORK

Building upon the method from [5], this section outlines a distributed strategy to address the issue. The strategy involves an associated fictitious multi-agent system that evolves on the edges of the communication graph, departing from the conventional node-based approach.

A. The Edge System

Inspired by [4], we associate a fictitious agent with each edge (ν^i, ν^j) of the communication graph with dynamics

$$\begin{bmatrix} q_k^m \\ w_k^m \end{bmatrix} = \begin{bmatrix} p_k^i - p_k^j - d^{ij} \\ v_k^i - v_k^j \end{bmatrix} \quad \text{and} \quad a_k^m = u_k^i - u_k^j, \tag{11}$$

for $m = 1, ..., M$. The state vector for edge agent m is $z_k^m = [q_k^m, w_k^m] \in \mathbb{R}^{2n}$. Then, the relative dynamics for edge agent m is

$$z_{k+1}^m = f z_k^m + g a_k^m, \quad \text{for } m = 1, ., ., M.$$

The state vector for the whole edge system can be arranged as $\tilde{z}_k = [z_k^1, ..., z_k^M]^T \in \mathbb{R}^{2Mn}$. We rearrange the states by a permutation

$$z_k = \Pi \tilde{z}_k,$$

with permutation matrix

$$\Pi = \begin{bmatrix} I_M \otimes \begin{bmatrix} 1 & 0 \end{bmatrix} \\ I_M \otimes \begin{bmatrix} 0 & 1 \end{bmatrix} \end{bmatrix} \otimes I_n \in \mathbb{R}^{2Mn \times 2Mn}.$$

Therefore, the whole edge dynamics can be written as

$$z_{k+1} = \bar{F} z_k + \sum_{m=1}^{M} \bar{G}^m a_k^m, \tag{12}$$

where $z_k = [q_k^1, ..., q_k^M, w_k^1, ..., w_k^M]^T \in \mathbb{R}^{2Mn}$ and

$$\bar{F} = \left(\begin{bmatrix} 1 & \delta \\ 0 & 1 \end{bmatrix} \otimes I_M \otimes I_n \right) \in \mathbb{R}^{2Mn \times 2Mn},$$

$$\bar{G}^m = \begin{bmatrix} \frac{\delta^2}{2} \bar{g}^m \\ \delta \bar{g}^m \end{bmatrix} \in \mathbb{R}^{2Mn \times n},$$

with $\bar{g}^m = \bar{c}^m \otimes I_n \in \mathbb{R}^{Mn \times n}$, where $\bar{c}^m$ is the m-th column of identity matrix of size M.

B. Formation with Collision Avoidance on the Edge System

Since we relocated the coupling terms that were initially in the cost function to the system dynamics, the local error for an edge agent m at time instance k to be minimized is

$$\begin{aligned} \bar{\Psi}_k^m &= \alpha^m \left(||q_k^m||^2 + ||w_k^m||^2 \right) + \beta^m \left(\frac{||q_k^m||^2 + ||w_k^m||^2}{||q_k^m + d^{ij}||^2 - r^2} \right) \\ &= z_k^T \left(\bar{Q}_\alpha^m + \bar{Q}_\beta^m(z_k^m) \right) z_k \\ &= z_k^T \bar{Q}^m(z_k^m) z_k \end{aligned}$$

where

$$\bar{Q}_\alpha^m = \delta(I_2 \otimes \bar{W}_\alpha^m \otimes I_n) \in \mathbb{R}^{2Mn \times 2Mn},$$
$$\bar{Q}_\beta^m(z_k^m) = \delta(I_2 \otimes \bar{W}_\beta^m(z_k^m) \otimes I_n) \in \mathbb{R}^{2Mn \times 2Mn},$$

with $\bar{W}_\alpha^m \in \mathbb{R}^{M \times M}$ is a diagonal matrix such that the m-th diagonal entry of $\bar{W}_\alpha^m$ is equal to α^m and zero otherwise and let $\bar{W}_\beta^m(z_k^m) \in \mathbb{R}^{M \times M}$ be a diagonal matrix such that the m-th diagonal entry of $\bar{W}_\beta^m(z_k^m)$ is equal to $\dfrac{\beta^m}{||q_k^m + d^{ij}||^2 - r^2}$ and zero otherwise.

Note that in contrast to $Q^i(x_k)$ from the first problem, $\bar{Q}^m(z_k^m)$ here is a block diagonal matrix where edge dynamics are decoupled. Therefore, we can arrange the decoupled cost function for the m-th edge as

$$\bar{J}^m(A^m) = \frac{1}{2} \Big(Z_k^T \bar{\mathcal{Q}}^m(z_k^m) Z_k + A_k^{m^T} \bar{\mathcal{R}}^{mm} A_k^m \Big), \tag{13}$$

where the stacked edge state vector now is arranged as $Z_k = [z_{k+1}, z_{k+2}, ..., z_{k+T}]^T \in \mathbb{R}^{2MnT}$ and the stacked relative control inputs vector is $A_k^m = [a_k^m, a_{k+1}^m, ..., a_{k+T-1}^m]^T \in \mathbb{R}^{MnT}$.

The state weighting matrix for the new cost evolving on edges is defined as $\bar{\mathcal{Q}}^m(z_k^m) = \text{blkdiag}\big(\bar{Q}^m(z_k^m), ..., \bar{Q}^m(z_k^m), \bar{Q}_T^m(z_T^m)\big) \in \mathbb{R}^{2MnT \times 2MnT}$, where the terminal cost $\bar{Q}_T^m(z_T^m) \in \mathbb{R}^{2MnT \times 2MnT}$ has the same pattern as $\bar{Q}^m(z_k^m)$ with arbitrary choices of scalar weights instead of $\alpha^m, \beta^m > 0$. The control weight is $\bar{\mathcal{R}}^{mm} = \text{blkdiag}(\bar{R}^{mm}) \in \mathbb{R}^{MnT \times MnT}$ with a positive definite matrix $\bar{R}^{mm} \in \mathbb{R}^{Mn \times Mn}$. Finally, we formulate the new problem for the edge dynamics (3) as follows.

Problem 2: Minimize the local cost function (13) over the relative acceleration control input sequences a^i subject to dynamics (12).

Theorem 2: The optimal solution to Problem 2 is

$$a_k^{m\star}(z_k^m) = \bar{K}_k^m(z_k^m) z_k, \quad \text{for } m = 1, ..., M, \tag{14}$$

where

$$\begin{aligned} \bar{K}_k^m(z_k^m) = -(\bar{R}^{mm} + \bar{G}^{m^T} \bar{P}_{k+1}^m(z_{k+1}^m) \bar{G}^m)^{-1} \times \\ \bar{G}^{m^T} \bar{P}_{k+1}^m(z_{k+1}^m) \bar{F}, \end{aligned} \tag{15}$$

and $\bar{P}_{k+1}^m(z_{k+1}^m)$ is the solution to the decoupled state-dependent Riccati difference equation

$$\begin{aligned} \bar{P}_k^m(z_k^m) &= \bar{F}^T \bar{P}_{k+1}^m(z_{k+1}^m) \bar{F} + \bar{F}^T \bar{P}_{k+1}^m(z_{k+1}^m) \bar{G}^m \bar{K}_k^m(z_k^m) \\ &+ \bar{Q}^m(z_k^m) + (I_M \otimes z_k^{m^T}) \left[z_k^{m^T} \frac{\partial \bar{Q}^m(z_k^m)}{\partial z_k^1} \; \dots \; z_k^{m^T} \frac{\partial \bar{Q}^m(z_k^m)}{\partial z_k^M} \right]^T \end{aligned} \tag{16}$$

with $\bar{P}_T^m(z_T^m) = \bar{Q}_T^m(z_T^m)$.

Proof: Can be shown similarly to Section 3 in [11]. ■

Note that because both $\bar{Q}^m(z_k^m)$ and its derivative in (16) involve the norm of a variable, the σ-norm defined in (1) is employed to ensure differentiability throughout. The feedback gains $\bar{K}_k^m(z_k^m)$ in (15) are now decoupled from each other. This decoupling principally permits a distributed implementation, in contrast to $K_k^i(x_k)$ in (7).

However, solving the decoupled SDRDE in (16) is challenging due to its state-dependency. To address this challenge, we embrace the receding horizon technique for solving SDRDE presented in [7]. The approach involving the decoupled SDRDE entails the following steps:

1. Utilize the state-feedback gains $\bar{K}_k^m$ from (15) computed in the previous iteration. Let z_k^p be the prediction

of the dynamics, commencing from the current state z_k.
2. Work backwards in time to compute the Riccati solution, yielding $\bar{P}^m_{k+T}, ..., \bar{P}^m_{k+1}$ along the predicted state trajectory.
3. Employ this information to update the state feedback gains $\bar{K}^m_k, ..., \bar{K}^m_{k+T-1}$. Implement the first gain $\bar{K}^m_k$ for control purposes.
4. At the subsequent sampling instant, repeat this process, and make use of the remaining gains $\bar{K}^m_{k+1}, ..., \bar{K}^m_{k+T-1}$.
5. Determine the terminal gain $\bar{K}^m_{k+T}$ required for the next iteration by solving the decoupled SDRDE along the predicted states. This approach facilitates a receding horizon strategy.

The detailed steps to evaluate the decoupled SDRDE approach are provided in Algorithm 1 in [10].

C. Distributed Implementation

In this subsection, we show how to obtain the optimal control inputs of the physical vertex agents from the relative control inputs $a^{m\star}_k$ of the fictitious (edge) agents in a distributed fashion. We will use the symbol $\hat{u}^{i\star}_k$ to denote the physical control inputs corresponding to the fictitious relative control inputs $a^{m\star}_k$. Recall that from (11), we can express the relation between $a^\star_k$ and $\hat{u}^\star_k$ as

$$\Phi \hat{u}^\star_k = a^\star_k,$$

where $\Phi = D^T \otimes I_n$ and $a^\star_k = [a^{1\star^T}_k, ..., a^{M\star^T}_k]^T$.

We consider minimizing the residual $f(u) = ||\Phi \hat{u}^\star_k - a^\star_k||^2$. Since the undirected graph $\mathcal{G}$ is assumed to be connected, there exists a unique solution to minimizing the residual $f(u)$, given by

$$\hat{u}^\star_k = \Phi^\dagger a^\star_k, \tag{17}$$

where $\Phi^\dagger$ is the pseudo-inverse of Φ. Since $\Phi^\dagger$ is a fully populated matrix, this will lead to a centralized solution. To compute (17) in a distributed way, a distributed steepest descent algorithm is employed, which updates the local control input at iteration step l according to

$$\hat{u}^\star_{l+1} = (I - 2\gamma \Phi^T \Phi)\hat{u}^\star_l + 2\gamma \Phi^T a^\star_k, \tag{18}$$

with γ as a learning rate that satisfies

$$2\gamma \leq \frac{2}{\lambda_{\max}(\Phi^T \Phi)}.$$

It was demonstrated in [12] that this algorithm converges to a solution $\hat{u}^\star_k$ in (17) which is unique. The key fact is that the two matrices on the right-hand side of (18) are sparse and allow a distributed computation of the updates $\hat{u}^\star_{l+1}$. The detailed steps to evaluate this approach are provided in Algorithm 2 in [10].

IV. ILLUSTRATIVE EXAMPLE

This section illustrates the proposed approach with a formation control problem where double integrator agents are moving in $n = 2$ dimensional space. We consider $N = 4$ agents and $M = 5$ edges with an undirected communication graph, as displayed in Figure 1.

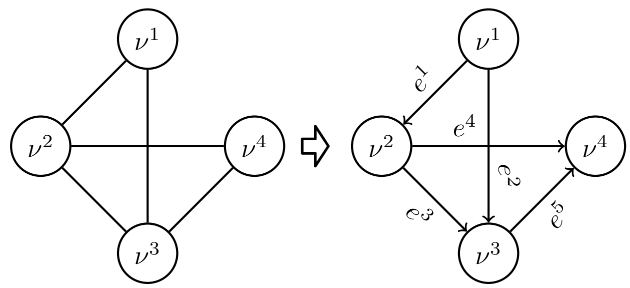

Fig. 1. Arbitrary orientation of the $M = 5$ edges of an undirected graph with $N = 4$ nodes.

The incidence matrix is

$$D = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 \\ 0 & -1 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{bmatrix} \in \mathbb{R}^{4\times 5}.$$

We assume that all agents have zero initial velocities, except agent 1 that has $v^1_0 = \begin{bmatrix}0.5, 1\end{bmatrix}^T$. The agents have initial positions

$$p^1_0 = \begin{bmatrix}3.5\\1\end{bmatrix}, p^2_0 = \begin{bmatrix}12\\1\end{bmatrix}, p^3_0 = \begin{bmatrix}0\\5\end{bmatrix}, p^4_0 = \begin{bmatrix}15\\3.5\end{bmatrix},$$

with the desired displacements vectors and safety radius

$$d^{12} = \begin{bmatrix}1.5\\1\end{bmatrix}, \quad d^{13} = \begin{bmatrix}0\\2\end{bmatrix}, \quad d^{23} = \begin{bmatrix}-1.5\\1\end{bmatrix},$$
$$d^{24} = \begin{bmatrix}-3\\0\end{bmatrix}, \quad d^{34} = \begin{bmatrix}-1.5\\-1\end{bmatrix}, \quad r = 0.5.$$

A. Simulation Results

We first show the evolution of agents' positions if collision avoidance is ignored, by taking $\bar{Q}^m_\beta(z^m_k) = 0$, for all $m \in M$.

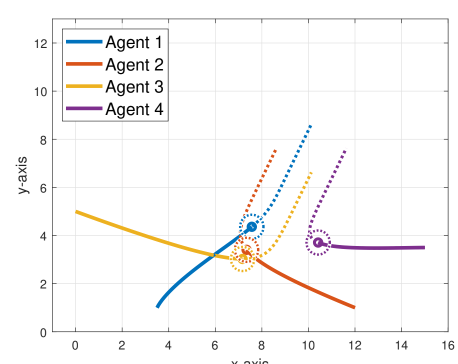

Fig. 2. Progression of four agents' position on x, y-axes in 4 seconds, without collision avoidance.

Fig. 2 depicts the four agents moving in the $x-y$-plane. A sampling time of $100ms$ is used, and the dashed lines display the final trajectories over a period of $20s$. The solid lines represent the intermediate progression run over $4s$. The figure illustrates that the collision between agents within $4s$.

For simulating formation control with collision avoidance, we construct $\bar{Q}^m_\beta(z^m_k)$ individually for each edge, wherein β^m are set to 1 for all $m \in M$. Initially, we execute the steps associated with the DSDRDE using a horizon of $T = 10$, yielding the relative control inputs $a^{m\star}_k$. Once these relative inputs are acquired, we proceed with running the distributed steepest descent method to compute the physical control inputs $\hat{u}^i_k$ and to simulate the actual dynamics.

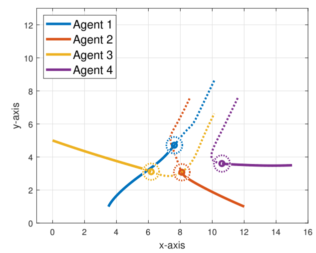

Fig. 3. Progression of four agents' position on x, y-axes with collision avoidance in 4s.

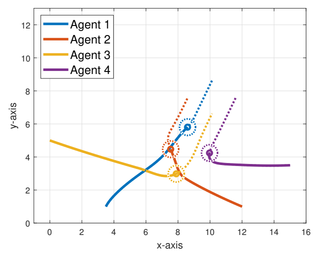

Fig. 4. Progression of four agents' position on x, y-axes with collision avoidance in 7s.

Formation with collision avoidance is visually presented in Figs. 3 and 4. It is run for 4 and $7s$, respectively. As depicted in Fig. 3, agents one, two, and three successfully avoid collisions, in contrast to the scenario shown in Fig. 2. When we extend the simulation time to $7s$, a noteworthy observation emerges: agent three closely follows agent two, who, in turn, tracks agent one, resulting in their alignment within the desired formation.

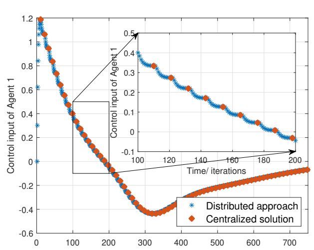

Fig. 5. Control input of agent 1; centralised and distributed solution with 10 iterations/ sampling interval.

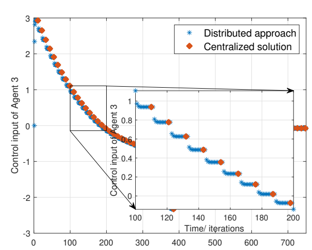

Fig. 6. Control input of agent 3; centralised and distributed solution with 10 iterations/ sampling interval.

The last two plots compare control input progression achieved through the centralized solution in (17) and the distributed approach in (18) with 10 iterations per sampling interval. Each plot displays the x-direction evolution, with blue stars indicating the distributed solution and orange diamonds representing the centralized solution. In Fig. 6, the distributed scheme quickly converges to the centralized solution, despite a small initial gap. Meanwhile, Fig. 5 reveals that agent one's control input convergence is slower due to limited interaction with only two neighbors. All visualizations were generated using the code in [13].

V. CONCLUSIONS

This article addresses the challenge of guiding a group of N agents from their initial position to a desired formation while avoiding collisions with neighboring agents. The original problem is formulated as an LQDTG with a coupled SDRDE, which cannot be solved in a distributed fashion. To address this issue, a distributed approach is proposed. This approach is based on a fictitious MAS that operates on the edges of the graph rather than the nodes. The technique incorporates relative soft constraints on the edges to prevent collisions and requires the solution of a decoupled SDRDE, using a receding horizon technique. The proposed method leverages a distributed steepest descent algorithm to map the relative control inputs to the actual physical control inputs, resulting in a simple vector-matrix multiplication per iteration, in contrast to an iterative approach often used in distributed MPC, which requires solving an optimization problem in each sampling interval. The efficacy of the proposed method is demonstrated through simulation results.

REFERENCES

[1] S. Chen, J. Dai, J.-W. Yi, and L. Chai, "An optimal design of the leader-following formation control for discrete multi-agent systems," *IFAC-PapersOnLine*, vol. 55, no. 3, pp. 201–206, 2022, 16th IFAC Symposium on Large Scale Complex Systems: Theory and Applications LSS 2022.

[2] D. Gu, "A differential game approach to formation control," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 1, pp. 85–93, 2008.

[3] J. H. Barghi and N. Vasif, "On the finite horizon nash equilibrium solution in the differential game approach to formation control," *Journal of Systems Engineering and Electronics*, vol. 30, no. 6, pp. 1233–1242, 2019.

[4] H. B. Jond and J. Platoš, "Differential game-based optimal control of autonomous vehicle convoy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 3, pp. 2903–2919, 2023.

[5] P. Aditya and H. Werner, "A distributed linear-quadratic discrete-time game approach to multi-agent consensus," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 6169–6174.

[6] G. Stomberg, H. Ebel, T. Faulwasser, and P. Eberhard, "Cooperative distributed mpc via decentralized real-time optimization: Implementation results for robot formations," 2023.

[7] A. S. Dutka, A. W. Ordys, and M. J. Grimble, "Optimized discrete-time state dependent riccati equation regulator," *Proceedings of the 2005, American Control Conference, 2005.*, pp. 2293–2298 vol. 4, 2005.

[8] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[9] T. Mylvaganam, M. Sassano, and A. Astolfi, "A differential game approach to multi-agent collision avoidance," *IEEE Transactions on Automatic Control*, vol. 62, pp. 4229–4235, 2017.

[10] P. Aditya and H. Werner, *A Distributed Linear Quadratic Discrete-Time Game Approach to Formation Control with Collision Avoidance*. arXiv, Mar 2023. [Online]. Available: https://arxiv.org/pdf/2308.12775.pdf

[11] I. Chang and J. Bentsman, "Constrained discrete-time state-dependent riccati equation technique: A model predictive control approach," in *52nd IEEE CDC*, 2013, pp. 5125–5130.

[12] F. Bullo, *Lectures on Network Systems, subsection 9.5.3, page 145*. Kindle Direct Publishing, 2022. [Online]. Available: https://fbullo.github.io/lns

[13] P. Aditya and H. Werner, *Code for paper: A Distributed LQDTG to Formation Control with Collision Avoidance*. Zenodo, Mar 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7789655