Distributed Game Strategy Design with Application to Multi-Agent Formation Control

Wei Lin, *Member, IEEE,* Zhihua Qu, *Fellow, IEEE,*, and Marwan A. Simaan, *Life Fellow, IEEE*

Abstract

In this paper, we consider a multi-agent formation control problem from a game theory point of view. It is well known that a major difficulty in a communication network based formation control problem is that each agent is only able to exchange information with other agents according to the communication topology. This information constraint prevents many game strategy design approaches that require individual agents to have global information from being implemented in many cases. We formulate the formation control problem in such a way that individual agents try to minimize their locally measured formation errors and to solve it as a differential game problem. We consider two cases of non-cooperative and cooperative games and propose a novel distributed design approach that utilizes the relationship between the initial and terminal state variables. This approach is applied to an illustrative formation control example among three agents and the formation errors under various scenarios are compared and analyzed.

I. Introduction

An important application of cooperative control theory [?], [?] is the multi-agent formation control problems [?], [?], [?]. This problem revolves around designing local information constrained control inputs for the agents (mobile robots, unmanned vehicles, etc) and making their positions follow a prescribed formation. There is considerable work done in this field and the comprehensive review paper [?] on multi-agent distributed control systems is a good reference on recent progress in cooperative control theory including the formation control problem. Recently, multiple attempts [?], [?], [?] have been made to incorporate differential game theory [?], [?] into the formation control design problem. A fundamental connection between these two problems can be explained as follows: in a multi-agent formation control problem, all the agents are basically trying to minimize the formation errors between each other. If every agent is able to observe all the other agents, then the problem can be essentially viewed as an optimal control problem where a common performance index of global formation errors can be assigned for all the agents. However, in most of realistic applications, since the agents are usually connected through a communication network with a certain topology, they are only able to exchange information with each other in a distributed manner (i.e., to share information locally). In this case, instead of minimizing a globally defined performance index, it is more realistic for each agent to only minimize a performance index of its locally measured formation errors. This scenario is exactly equivalent to that of a differential game where each player is trying to minimize its own performance index. Therefore, most multi-agent formation control design problems can be viewed and solved as differential game problems. The game solutions including Nash equilibria for non-cooperative agents and Pareto optimality for cooperative agents can be well established. However, solving for the Nash and Pareto solutions under distributed information is not a trivial task in the multi-agent environment because of the following two issues: (1) the strategy for each agent has to conform to its local information topology constraint and (2) local strategy computing within each agent is preferred rather than centralized strategy computing where strategies are designed by a centralized designer and assigned to individual agents. In [?], [?], the first issue has been addressed and the corresponding Nash and Pareto strategies are considered for the formation control problem as a differential game. In this present paper, we will focus more on the second aspect. By investigating the structure of the game solution in more details, a strategy design approach as well as an implementation algorithm are proposed for each agent to carry out its optimal control strategy design in a fully distributed manner. The remainder of this paper is organized as follows. A brief setup of the game problem is presented in Section II. The main results of the distributed Nash and Pareto strategy designs are presented in Section V and VI respectively. An illustrative example with three agents under noncooperative and cooperative cases is presented and analyzed in Section VII.

II. Problem Formulation

We consider the finite-time formation control problem between N agents in the n-dimensional Euclidean space where the motion dynamics of each agent is described as the following double integrator model:

$$\begin{bmatrix} \dot{p}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} 0 & I_n \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_i \\ v_i \end{bmatrix} + \begin{bmatrix} 0 \\ I_n \end{bmatrix} u_i \tag{1}$$

This work is supported in part by US National Science Foundation under grants ECCS-1308928 and CCF-0956501, by US Department of Energys award DE-EE0006340, by US Department of Transportations award DTRT13-G-UTC51, by L-3 Communications contract 11013I2034, and by Leidos contract P010161530.

Wei Lin received his PhD degree from University of Central Florida in Dec. 2013 and is currently working in Western Digital Corporation, Irvine, CA, 92612, USA. E-mail: weilin0929@gmail.com.

Zhihua Qu and Marwan A. Simaan are with the Department of EECS, University of Central Florida, Orlando, Florida, 32816, USA, e-mails: qu@eecs.ucf.edu, simaan@eecs.ucf.edu.

for $i = 1, \cdots, N$. Vector $p_i \in \mathbb{R}^n$ and $v_i \in \mathbb{R}^n$ are the position and velocity of agent i respectively. Vector $u_i \in \mathbb{R}^n$ is the acceleration control of agent i. We assume that the agents are able to exchange information with each other through a communication network whose topology can be described by a set $\mathcal{E}$ in graph theory. An edge $e_{ij} \in \mathcal{E}$ represents information being transmitted from agent j to agent i. The objective of the agents is to form a prescribed formation over a finite time interval, which is equivalent to individual agents minimizing the following performance indices:

$$\begin{aligned} J_i = \sum_{e_{ij} \in \mathcal{E}} \frac{1}{2} \bigg[& \|p_i(t_f) - p_j(t_f) - \mu_{ij}\|^2 \\ & + \|v_i(t_f) - v_j(t_f)\|^2 \bigg] \\ & + \frac{r_i}{2} \int_0^{t_f} [\|p_i(t) - p_j(t) - \mu_{ij}\|^2 \\ & + \|v_i(t) - v_j(t)\|^2 + \|u_i\|^2] \mathrm{d}t \end{aligned} \tag{2}$$

for $i = 1, \cdots, N$, where $\| \cdot \|$ is the Euclidean norm, μ_{ij} is the desired displacement vector pointing from agent j to agent i, and r_i is a positive scalar. Minimizing the performance index J_i in (2) means that every agent will try to minimize the total terminal formation error and terminal velocity error according to the graph while at the same time minimizing its control effort during the time interval. Since every agent is only aware of its local communication pattern and its own performance index, this formation problem can be regarded as a differential game problem where each player has its own objectives to pursue over a time interval. In this paper, we consider two types of open-loop (where the control inputs are functions of initial states and time only) formation control approaches through differential game strategy design. Specifically, if the control inputs $u_1, \cdots, u_N$ can be looked as strategies, then for the formulated problem, strategies $u_1^N, \cdots, u_N^N$ are the Nash strategies and form a Nash equilibrium if the inequality

$$J_i(u_1^N, \cdots, u_i^N, \cdots, u_N^N) \leq J_i(u_1^N, \cdots, u_i, \cdots, u_N^N) \tag{3}$$

holds for all $u_i \in U_i$ and for all $i = 1, \cdots, N$, where U_i is the open-loop control set for agent i. The Nash equilibrium is an equilibrium where it is not possible for any agent to lower its performance index value by unilaterally deviating from its Nash strategy. The Nash equilibrium is a widely used solution in the situation where the players do not cooperate with each other. On the other hand, The strategies $u_0^P, u_1^P, \cdots, u_N^P$ are the Pareto strategies and form a Pareto optimality if the inequality

$$J_i(u_1^P, \cdots, u_N^P) < J_i(u_1, \cdots, u_N) \tag{4}$$

holds for at least one $i \in \{1, \cdots, N\}$. The Pareto optimality can be interpreted as a solution in which any changes made do not help lower every agent's performance index value. The Pareto optimality is a widely used solution in the situation where the players can cooperate with each other. In the following sections, two formation control approaches will be proposed based on the open-loop Nash strategy (assuming that the agents do not cooperate) and the open-loop Pareto strategy (assuming that the agents cooperate).

III. No State Variable inside Integral

Assuming PI be

$$J_i = \frac{1}{2} \sum_{e_{ij} \in \mathcal{E}} [\|p_i(t_f) - p_j(t_f) - \mu_{ij}\|^2 + \|v_i(t_f) - v_j(t_f)\|^2] + \frac{1}{2} \int_0^{t_f} \|u_i\|^2 \mathrm{d}t \tag{5}$$

We define the Hamiltonian for agent i as

$$H_i = \frac{1}{2} \|u_i\|^2 + \lambda_{pi}^T v_i + \lambda_{vi}^T u_i \tag{6}$$

where vectors λ_{pi} and λ_{vi} are the Lagrangian multipliers. According to the well known Pontryagin's minimum principle [**?**], the necessary conditions for optimality are

$$\dot{p}_i = \frac{\partial H_i}{\partial \lambda_{pi}} = v_i, \tag{7a}$$

$$\dot{v}_i = \frac{\partial H_i}{\partial \lambda_{vi}} = u_i \tag{7b}$$

$$\dot{\lambda}_{pi} = -\frac{\partial H_i}{\partial p_i} = 0, \tag{7c}$$

$$\dot{\lambda}_{vi} = -\frac{\partial H_i}{\partial v_i} = -\lambda_{pi}, \tag{7d}$$

$$\lambda_{pi}(t_f) = \sum_{e_{ij}\in\mathcal{E}} [p_i(t_f) - p_j(t_f) - \mu_{ij}], \tag{7e}$$

$$\lambda_{vi}(t_f) = \sum_{e_{ij}\in\mathcal{E}} [v_i(t_f) - v_j(t_f)], \tag{7f}$$

$$\frac{\partial H_i}{\partial u_i} = u_i + \lambda_{vi} = 0, \tag{7g}$$

$$\frac{\partial^2 H_i}{\partial u_i^2} = r_i > 0. \tag{7h}$$

The optimal control can be obtained easily as

$$u = -\lambda_{vi}$$

Denoting $p = [p_1, \cdots, p_2]^T$, $v = [v_1, \cdots, v_2]^T$, $\lambda_p = [\lambda_{p1}, \cdots, \lambda_{p2}]^T$, $\lambda_v = [\lambda_{v1}, \cdots, \lambda_{v2}]^T$, and $\mu = [\mu_1, \cdots, \mu_2]^T$, the following state and co-state equations can be formed

$$\begin{bmatrix} \dot{p} \\ \dot{v} \\ \dot{\lambda}_p \\ \dot{\lambda}_v \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & 0 & -I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \end{bmatrix} \begin{bmatrix} p \\ v \\ \lambda_p \\ \lambda_v \end{bmatrix}$$

Solving the above equation yields

$$\begin{bmatrix} p(0) \\ v(0) \\ \lambda_p(0) \\ \lambda_v(0) \end{bmatrix} = \Phi(0, t_f) \begin{bmatrix} p(t_f) \\ v(t_f) \\ \lambda_p(t_f) \\ \lambda_v(t_f) \end{bmatrix}$$

where state-transition matrix is

$$\begin{aligned} \Phi(0, t_f) =& e^{-At_f} \\ =& \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} + \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & 0 & -I \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -I & 0 \end{bmatrix} (-t_f) + \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -I \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} (-t_f)^2 + \frac{1}{6} \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} (-t_f)^3 \\ =& \begin{bmatrix} I & -t_f I & -1/6t_f^3 & -1/2t_f^2 I \\ 0 & I & 1/2t_f^2 I & t_f I \\ 0 & 0 & I & 0 \\ 0 & 0 & t_f I & I \end{bmatrix} \end{aligned}$$

Therefore, we have

$$
\begin{aligned}
\begin{bmatrix} p(0) \\ v(0) \\ \lambda_p(0) \\ \lambda_v(0) \end{bmatrix} &= \begin{bmatrix} I & -t_f I & -1/6t_f^3 & -1/2t_f^2 I \\ 0 & I & 1/2t_f^2 I & t_f I \\ 0 & 0 & I & 0 \\ 0 & 0 & t_f I & I \end{bmatrix} \begin{bmatrix} p(t_f) \\ v(t_f) \\ \lambda_p(t_f) \\ \lambda_v(t_f) \end{bmatrix} \\
&= \begin{bmatrix} I & -t_f I & -1/6t_f^3 & -1/2t_f^2 I \\ 0 & I & 1/2t_f^2 I & t_f I \\ 0 & 0 & I & 0 \\ 0 & 0 & t_f I & I \end{bmatrix} \left(\begin{bmatrix} I & 0 \\ 0 & I \\ L & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} p(t_f) \\ v(t_f) \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} \mu \right) \\
&= \begin{bmatrix} I - \frac{1}{6}t_f^3 L & -t_f I - \frac{1}{2}t_f^2 L \\ \frac{1}{2}t_f^2 L & I + t_f L \\ L & 0 \\ t_f L & L \end{bmatrix} \begin{bmatrix} p(t_f) \\ v(t_f) \end{bmatrix} - \begin{bmatrix} -\frac{1}{6}t_f^3 \\ \frac{1}{2}t_f^2 I \\ I \\ t_f I \end{bmatrix} \mu
\end{aligned}
$$

The relationship between $p(0), v(0)$ and $p(t_f), v(t_f)$ is

$$
\begin{bmatrix} p(0) \\ v(0) \end{bmatrix} = \begin{bmatrix} I - \frac{1}{6}t_f^3 L & -t_f I - \frac{1}{2}t_f^2 L \\ \frac{1}{2}t_f^2 L & I + t_f L \end{bmatrix} \begin{bmatrix} p(t_f) \\ v(t_f) \end{bmatrix} - \begin{bmatrix} -\frac{1}{6}t_f^3 \\ \frac{1}{2}t_f^2 I \end{bmatrix} \mu
$$

where

$$
\begin{bmatrix} I - \frac{1}{6}t_f^3 L & -t_f I - \frac{1}{2}t_f^2 L \\ \frac{1}{2}t_f^2 L & I + t_f L \end{bmatrix} = \begin{bmatrix} I & -t_f I \\ 0 & I \end{bmatrix} \left(I + \begin{bmatrix} \frac{1}{3}t_f^3 L & \frac{1}{2}t_f^2 L \\ \frac{1}{2}t_f^2 L & t_f L \end{bmatrix} \right)
$$

The left hand side is obviously a Hurwitz matrix.

IV. State Variable inside Integral

Assuming PI be

$$
\begin{aligned}
J_i =& \frac{1}{2}[\|p_i(t_f) - p_j(t_f) - \mu_{ij}\|^2 + \|v_i(t_f) - v_j(t_f)\|^2] \\
&+ \frac{1}{2}\int_0^{t_f} [\|p_i(t) - p_j(t) - \mu_{ij}\|^2 + \|v_i(t) - v_j(t)\|^2 + \|u_i\|^2] \mathrm{d}t
\end{aligned} \tag{8}
$$

with state penalty within the integral. We define the Hamiltonian for agent i as

$$
H_i = \frac{1}{2}[\|p_i(t) - p_j(t) - \mu_{ij}\|^2 + \|v_i(t) - v_j(t)\|^2 + \|u_i\|^2] + \lambda_{pi}^T v_i + \lambda_{vi}^T u_i \tag{9}
$$

where vectors λ_{pi} and λ_{vi} are the Lagrangian multipliers. According to the well known Pontryagin's minimum principle [**?**], the necessary conditions for optimality are

$$
\dot{p}_i = \frac{\partial H_i}{\partial \lambda_{pi}} = v_i, \tag{10a}
$$

$$
\dot{v}_i = \frac{\partial H_i}{\partial \lambda_{vi}} = u_i \tag{10b}
$$

$$
\dot{\lambda}_{pi} = -\frac{\partial H_i}{\partial p_i} = -(p_i - p_j - \mu_{ij}), \tag{10c}
$$

$$
\dot{\lambda}_{vi} = -\frac{\partial H_i}{\partial v_i} = -(v_i - v_j + \lambda_{pi}), \tag{10d}
$$

$$
\lambda_{pi}(t_f) = p_i(t_f) - p_j(t_f) - \mu_{ij}, \tag{10e}
$$

$$
\lambda_{vi}(t_f) = v_i(t_f) - v_j(t_f), \tag{10f}
$$

$$
\frac{\partial H_i}{\partial u_i} = u_i + \lambda_{vi} = 0, \tag{10g}
$$

$$
\frac{\partial^2 H_i}{\partial u_i^2} = r_i > 0. \tag{10h}
$$

The optimal control can be obtained easily as

$$
u = -\lambda_{vi}
$$

Denoting $p = [p_1, \cdots, p_2]^T$, $v = [v_1, \cdots, v_2]^T$, $\lambda_p = [\lambda_{p1}, \cdots, \lambda_{p2}]^T$, $\lambda_v = [\lambda_{v1}, \cdots, \lambda_{v2}]^T$, and $\mu = [\mu_1, \cdots, \mu_2]^T$ the following state and co-state equations can be formed

$$\begin{bmatrix} \dot{p}_i \\ \dot{v}_i \\ \dot{\lambda}_{pi} \\ \dot{\lambda}_{vi} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & 0 & -I \\ -L & 0 & 0 & 0 \\ 0 & -L & -I & 0 \end{bmatrix} \begin{bmatrix} p_i \\ v_i \\ \lambda_{pi} \\ \lambda_{vi} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} \mu$$

Solving the above equation yields

$$\begin{bmatrix} p(0) \\ v(0) \\ \lambda_p(0) \\ \lambda_v(0) \end{bmatrix} = \Phi(0, t_f) \begin{bmatrix} p(t_f) \\ v(t_f) \\ \lambda_p(t_f) \\ \lambda_v(t_f) \end{bmatrix} + \int_{t_f}^{0} \Phi(0, \tau) \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} \mu \mathrm{d}\tau$$

where state-transition matrix is

$$\begin{aligned} \Phi(0, t_f) =& e^{-At_f} \\ =& \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & I \end{bmatrix} + \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & 0 & -I \\ -L & 0 & 0 & 0 \\ 0 & -L & -I & 0 \end{bmatrix} (-t_f) + \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & -I \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} (-t_f)^2 + \frac{1}{6} \begin{bmatrix} 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} (-t_f)^3 \\ =& \begin{bmatrix} I & -t_f I & -1/6 t_f^3 & -1/2 t_f^2 I \\ 0 & I & 1/2 t_f^2 I & t_f I \\ 0 & 0 & I & 0 \\ 0 & 0 & t_f I & I \end{bmatrix} \end{aligned}$$

Substituting (18) into (10b) and integrating both sides from 0 to t_f yield

$$\begin{aligned} & s_{pi}(t_f) + \frac{w_{pp}}{r_i} \sum_{e_{ij} \in \mathcal{E}} l_{ij} [s_{pi}(t_f) - s_{pj}(t_f)] + \frac{w_{pv}}{r_i} \sum_{e_{ij} \in \mathcal{E}} l_{ij} [s_{vi}(t_f) - s_{vj}(t_f)] \\ =& s_{pi}(0) + \frac{w_{pp}}{r_i} \alpha_i \end{aligned} \tag{11}$$

and

$$\begin{aligned} & s_{vi}(t_f) + \frac{w_{vp}}{r_i} \sum_{e_{ij} \in \mathcal{E}} l_{ij} [s_{pi}(t_f) - s_{pj}(t_f)] + \frac{w_{vv}}{r_i} \sum_{e_{ij} \in \mathcal{E}} l_{ij} [s_{vi}(t_f) - s_{vj}(t_f)] \\ =& s_{vi}(0) + \frac{w_{vp}}{r_i} \alpha_i \end{aligned} \tag{12}$$

where scalars $w_{pp}, w_{pv}, w_{vp}, w_{vv}$ are defined in (**??**)-(**??**) and α_i is defined in (**??**). Combining (11) and (12) and stacking them from $i = 1$ to $i = N$ yield

$$\begin{bmatrix} s_p(t_f) \\ s_v(t_f) \end{bmatrix} = M^{-1} \left\{ \begin{bmatrix} s_p(0) \\ s_v(0) \end{bmatrix} + \left(\begin{bmatrix} w_{pp} \\ w_{vp} \end{bmatrix} \otimes R^{-1} \otimes I_3 \right) \alpha \right\} \tag{13}$$

where vectors s_p and s_v are defined in (**??**), matrix M is defined in (24) and invertible according to Lemma **??**, and vector α is defined in (**??**). Therefore, rewriting (18) as

$$u_i = -\frac{1}{r_i} F_i \begin{bmatrix} s_p(t_f) \\ s_v(t_f) \end{bmatrix} + \frac{1}{r_i} \tilde{B}_p^T \alpha_i \tag{14}$$

where F_i is defined in (**??**) and substituting (23) into (14) yields (**??**). Since $s_p(0), s_v(0)$ are in fact functions of the initial state $z(0)$ through (15), strategies $u_1^*, \cdots, u_N^*$ in (**??**) form an open-loop Nash equilibrium.

V. Distributed Nash Strategy Design

In this section, assuming that the agents do not corporate, we first consider the formation control through distributed open-loop Nash strategy design. We introduce the following new state variable:

$$s_i(t) = p_i(t) + (t_f - t)v_i(t) \tag{15}$$

for agent i. Then the performance index (2) becomes

$$J_i = \sum_{e_{ij}\in\mathcal{E}} \frac{1}{2}[\|s_i(t_f) - s_j(t_f) - \mu_{ij}\|^2 \\ + \|v_i(t_f) - v_j(t_f)\|^2] + \frac{r_i}{2}\int_0^{t_f} \|u_i\|^2 \mathrm{d}t. \tag{16}$$

for $i = 1, \cdots, N$. The dynamics of the new state variables is given by

$$\dot{s}_i = (t_f - t)u_i \quad \text{and} \quad \dot{v}_i = u_i. \tag{17}$$

Based on the above performance indices and state dynamics, we can solve inequality (3) for $i = 1, \cdots, N$ as N simultaneous open-loop optimal control problems (by using the approach of calculus of variation [**?**]) and obtain the following open-loop Nash strategies:

$$u_i^N = -\frac{t_f - t}{r_i}\sum_{e_{ij}\in\mathcal{E}}[s_i(t_f) - s_j(t_f) - \mu_{ij}] \\ -\frac{1}{r_i}\sum_{e_{ij}\in\mathcal{E}}[v_i(t_f) - v_j(t_f)] \quad \forall i = 1, \cdots, N \tag{18}$$

which is analogous to the open-loop Nask solutions provided in [**?**]. Theoretically, the terminal variables $s_i(t_f)$ and $v_i(t_f)$ can be obtained by substituting the open-loop control (18) back into the system dynamics (17) and solving the equations for all $i = 1, \cdots, N$ simultaneously. However, solving these equations is beyond the capability of each agent because the agent is only aware of its own performance index and its local communication topology. To overcome this difficulty, one approach is to let all the agents exchange certain information across the communication network so that the terminal variable values $s_i(t_f)$ and $v_i(t_f)$ can be obtained by agent i in a distributed manner for all $i = 1, \cdots, N$. Toward this end, the following theorem is proposed.

Theorem 1: If agent i updates vectors h_{si} and h_{vi} continuously according to

$$\begin{bmatrix}\dot{h}_{si}\\ \dot{h}_{vi}\end{bmatrix} = k_i\Bigg\{\begin{bmatrix}s_i(0)\\ v_i(0)\end{bmatrix} + \frac{1}{r_i}\begin{bmatrix}(t_f^3/3)\rho_i\\ (t_f^2/2)\rho_i\end{bmatrix} - \begin{bmatrix}h_{si}\\ h_{vi}\end{bmatrix} \\ -\frac{1}{r_i}(W\otimes I_n)\sum_{e_{ij}\in\mathcal{E}}\left(\begin{bmatrix}h_{si}\\ h_{vi}\end{bmatrix} - \begin{bmatrix}h_{sj}\\ h_{vj}\end{bmatrix}\right)\Bigg\} \tag{19}$$

where k_i is a positive scalar,

$$W = \begin{bmatrix}t_f^3/3 & t_f^2/2\\ t_f^2/2 & t_f\end{bmatrix}, \tag{20}$$

$$\rho_i = \sum_{e_{ij}\in\mathcal{E}} \mu_{ij} \quad \forall i = 1, \cdots, N, \tag{21}$$

then

$$\lim_{\tau\to\infty}\begin{bmatrix}h_{si}(\tau)\\ h_{vi}(\tau)\end{bmatrix} = \begin{bmatrix}s_i(t_f)\\ v_i(t_f)\end{bmatrix} \tag{22}$$

for $i = 1, \cdots, N$, where $s_i(t_f)$ and $v_i(t_f)$ are the terminal variables in Nash strategies (18).

Proof: Firstly, substituting the open-loop strategies (18) back into system dynamics (17) and integrating the equations yields

$$M\begin{bmatrix}s(t_f)\\ v(t_f)\end{bmatrix} = \begin{bmatrix}s(0)\\ v(0)\end{bmatrix} + \begin{bmatrix}(t_f^3/3)(R^{-1}\otimes I_n)\\ (t_f^2/2)(R^{-1}\otimes I_n)\end{bmatrix}\mu \tag{23}$$

where $s = [s_1^T \ \cdots \ s_N^T]^T$, $v = [v_1^T \ \cdots \ v_N^T]^T$, $R = \mathrm{diag}\{r_1, \cdots, r_N\}$, $\mu = [\mu_1^T \ \cdots \ \mu_N^T]^T$, and

$$M = [I_{2N} + W\otimes(R^{-1}\mathcal{L})]\otimes I_n, \tag{24}$$

where matrix $\mathcal{L} = [l_{ij}] \in \mathbb{R}^{N\times N}$ is the Laplacian matrix associated with the information graph among the agents and

$$l_{ij} = \begin{cases} -1 & \text{if } e_{ij} \in \mathcal{E} \text{ for } j \neq i \\ 0 & \text{if } e_{ij} \notin \mathcal{E} \text{ for } j \neq i \\ -\sum_{q=1,q\neq i}^{N} l_{iq} & \text{if } j = i \end{cases}, \tag{25}$$

Since all the eigenvalues of a Laplacian matrix have nonnegative real parts, all the eigenvalues of matrix M in (24) have positive real parts and matrix M is hence invertible.

Secondly, combining equations (19) for $i = 1, \cdots, N$ in a more compact vector form yields

$$\begin{bmatrix} \dot{h}_s \\ \dot{h}_v \end{bmatrix} = K\left\{ \begin{bmatrix} s(0) \\ v(0) \end{bmatrix} - M \begin{bmatrix} h_s \\ h_v \end{bmatrix} + \begin{bmatrix} (t_f^3/3)(R^{-1} \otimes I_n) \\ (t_f^2/2)(R^{-1} \otimes I_n) \end{bmatrix} \mu \right\}, \tag{26}$$

where $h_s = [h_{s1}^T \ \cdots h_{sN}^T]^T$, $h_v = [h_{v1}^T \ \cdots h_{vN}^T]^T$, $K = \text{blkdiag}\{k_1(I_{2n}), \cdots, k_N(I_{2n})\}$. Since all the eigenvalues of matrix M has positive real parts, system (26) is stable and the final convergent values of h_s and h_v are identical to the solution of $s(t_f)$ and $v(t_f)$ in equation (23). Therefore,

$$\lim_{\tau\to\infty} \begin{bmatrix} h_s(\tau) \\ h_v(\tau) \end{bmatrix} = \begin{bmatrix} s(t_f) \\ v(t_f) \end{bmatrix}.$$

and hence equation (22) holds for all $i = 1, \cdots, N$. ■

Based on the above theorem, all the agents can exchange information with each other through the communication network and estimate the unknown terminal values in (18). Therefore, the following implementation algorithm is proposed.

Algorithm 1:

1. All the agents initialize the state $s_i(0) = p_i(0) + t_f v_i(0)$ for all $i = 1, \cdots, N$.
2. For a certain period of time (Δt), all the agents communicate with each other through the network and update the states h_{si} and h_{vi} according to (19) and obtain the estimate of $s_i(t_f)$ and $v_i(t_f)$ for all $i = 1, \cdots, N$.
3. At $t = \Delta t$, all the agents implement the open-loop Nash strategies in (18) with $s_i(t_f)$ and $v_i(t_f)$ replaced by the estimated terminal states $h_{si}(\Delta t)$ and $h_{vi}(\Delta t)$.

The above algorithm is fully distributed (except for the knowledge of time instants 0 and Δt) because agent i only need to send out its own state information h_{si} and h_{vi} to the agents that have direct information injection links from agent i and receive the state variables from the agents that have direct information links to agent i.

Note that the terminal state estimate will approximate to the ideal Nash equilibrium strategies since the convergence of the linear system (19) is asymptotic in nature. Therefore, the strategies obtained in the above algorithm are also known to form an ϵ-Nash equilibrium. A quantitative analysis is provided in [**?**].

VI. Distributed Pareto Strategy Design

In this section, assuming that the agents cooperate, we consider the formation control through distributed open-loop Pareto strategy design. The Pareto strategies can be derived through solving the following convex combination of all the agents' performance indices

$$J = \sum_{j=1}^{N} \alpha_j J_j \tag{27}$$

where $0 < \alpha_j < 1$ and $\sum_{j=1}^{N} \alpha_j = 1$. In this paper, the value of α_j has to lie strictly within the interval between 0 and 1 for all $j = 1, \cdots, N$, which basically means that all agents have to be taken into account in (27). This minimization is essentially an optimal control problem with any fixed parameters $\alpha_1, \cdots, \alpha_N$. Adopting the new state representation in (15), the convex combination in (27) with J_i defined in (16) can be expressed as

$$\begin{aligned} J =& \sum_{j=1}^{N} \frac{\alpha_j}{2} \Big\{ \sum_{e_{jk}\in\mathcal{E}} [\|s_j(t_f) - s_k(t_f) - \mu_{jk}\|^2 \\ & + \|v_j(t_f) - v_k(t_f)\|^2] + r_j \int_0^{t_f} \|u_j\|^2 \mathrm{d}t \Big\} \end{aligned} \tag{28}$$

for $i = 1, \cdots, N$. Now, since all the agents will minimize the common performance index above, it is necessary to make several assumption in order to have a distributed design approach.

Assumption 1: The communication link between any two agents is assumed to be bidirectional, that is, the information graph $\mathcal{G}$ is undirected.

Assumption 2: Agent i has the knowledge of the convex parameter α_j of agent j for all $e_{ij} \in \mathcal{E}$.

Based on the above assumptions, the following Pareto strategies can be obtained by solving inequality (4) as an open-loop optimal control problem using the calculus of variation approach again:

$$\begin{aligned} u_i^P = & -\frac{t_f - t}{\alpha_i r_i} \sum_{e_{ij} \in \mathcal{E}} (\alpha_i + \alpha_j)[s_i(t_f) - s_j(t_f) - \mu_{ij}] \\ & - \frac{1}{\alpha_i r_i} \sum_{e_{ij} \in \mathcal{E}} (\alpha_i + \alpha_j)[v_i(t_f) - v_j(t_f)]. \end{aligned} \tag{29}$$

which is analogous to the non-inferior solutions provided in [**?**]. Since the value of α_i lies strictly within the interval between 0 and 1 for all $i = 1, \cdots, N$, α_i can be safely placed in the denominator in (29). The terminal states can be obtained by substituting the strategies (29) into the system dynamics (17) and solving the N equations. Due to the local information constraint, we propose the following theorem similar to Theorem 1 to overcome agent i's inability to solve these equations.

Theorem 2: If agent i updates vectors h_{si} and h_{vi} continuously according to

$$\begin{aligned} \begin{bmatrix} \dot{h}_{si} \\ \dot{h}_{vi} \end{bmatrix} = k_i \Bigg\{ & \begin{bmatrix} s_i(0) \\ v_i(0) \end{bmatrix} + \frac{1}{\alpha_i r_i} \begin{bmatrix} (t_f^3/3)\rho_i \\ (t_f^2/2)\rho_i \end{bmatrix} - \begin{bmatrix} h_{si} \\ h_{vi} \end{bmatrix} \\ & - \frac{1}{\alpha_i r_i} (W \otimes I_3) \sum_{e_{ij} \in \mathcal{E}} (\alpha_i + \alpha_j) \left(\begin{bmatrix} h_{si} \\ h_{vi} \end{bmatrix} - \begin{bmatrix} h_{sj} \\ h_{vj} \end{bmatrix} \right) \Bigg\} \end{aligned} \tag{30}$$

where k_i is a positive scalar, matrix W is defined in (20), and

$$\rho_i = \sum_{e_{ij} \in \mathcal{E}} (\alpha_i + \alpha_j)\mu_{ij},$$

then the limit (22) holds for $i = 1, \cdots, N$, where $s_i(t_f)$ and $v_i(t_f)$ are the terminal variables in Pareto strategies (29).

Proof: Similar to the proof of Theorem 1, substituting the open-loop strategies (29) into system dynamics (17) yields equation (23) where matrix M is defined in (24) with

$$l_{ij} = \begin{cases} -(\alpha_i + \alpha_j) & \text{if } e_{ij} \in \mathcal{E} \text{ for } j \neq i \\ 0 & \text{if } e_{ij} \notin \mathcal{E} \text{ for } j \neq i \\ -\sum_{q=1, q \neq i}^{N} l_{iq} & \text{if } j = i \end{cases}, \tag{31}$$

for the Laplacian matrix $\mathcal{L}$. Stacking equation (30) from $i = 1$ to $i = N$ yields an equation similar to (26). Since matrix $(-M)$ is Hurwitz. Therefore, the estimator is asymptotically convergent as shown in (22). ■

Based on the above theorem, the same algorithm as Algorithm 1 can be carried out to obtain Pareto strategies in practice.

A special case to consider is when the agent does not receive any information from the other agents. In this situation, according to the expressions in (18) and (29), either the Nash or Pareto strategy becomes zero and hence the agent will keep a constant velocity, which is still a reasonable choice in practice.

VII. Illustrative Example

In this section, we consider an illustrative formation control example in two-dimensional space and apply the proposed distributed Nash and Pareto strategy design approaches. The dynamics of the agents are given by (1) with $n = 2$. For the ease of illustration, the horizontal axis is called the x-axis and the vertical axis is called the y-axis. Various scenarios are considered as follows.

Scenario 1: We start with a simple case, where there are three agents with directed communication topology as shown in Figure 1, where agent 1 as the leader does not have information injection from the other agents but only sends its information out to the other agents. Hence, the Laplacian matrix defined in (25) is given by

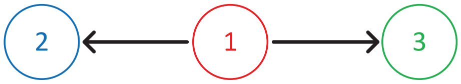

Fig. 1: Three agents with undirected graph

$$\mathcal{L} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}.$$

The initial states of the three agents are given by

$$p_1 = [0\ \ 0]^T, p_2 = [-3\ \ 0]^T, p_3 = [3\ \ 0]^T,$$
$$v_1 = [0\ \ 2]^T, v_2 = [0\ \ 0]^T, v_3 = [0\ \ 0]^T,$$

which means their initial positions are equally spaced and agent 1's initial velocity is upward and agents 2 and 3's initial velocities are equal to zero. We assume that the desired formation among the agents is a triangle with the displacement vectors given by

$$\mu_{12} = [2\ \ 2]^T, \quad \mu_{13} = [-2\ \ 2]^T.$$

In performance indices (2), the parameter r_i is given by $r_1 = r_2 = r_3 = 1$ and $t_f = 3$ for the time interval of the formation control. Since the information graph is directed, we only consider the formation control using the distributed Nash strategy design in this scenario. For comparison purpose, we solve for the actual Nash strategies by directly solving the equations (17) with strategies in (18) substituted in. The Nash strategy components on x-axis and y-axis are shown as blue trajectories in Figure 2 and the agents' terminal states under the Nash strategies are

$$s_1(t_f) = \begin{bmatrix} 0 \\ 6 \end{bmatrix}, \quad v_1(t_f) = \begin{bmatrix} 0 \\ 2 \end{bmatrix},$$
$$s_2(t_f) = \begin{bmatrix} -2.20 \\ 3.65 \end{bmatrix}, \quad v_2(t_f) = \begin{bmatrix} 0.23 \\ 1.90 \end{bmatrix},$$
$$s_3(t_f) = \begin{bmatrix} 2.20 \\ 3.65 \end{bmatrix}, \quad v_3(t_f) = \begin{bmatrix} -0.23 \\ 1.90 \end{bmatrix}.$$

If we implement the distributed Nash strategy design algorithm described in Algorithm 1 with $\Delta t = 0.1$ and $k_1 = k_2 = k_3 = 20$ in (19), then the final terminal state estimate values are

$$h_{s1}(\Delta t) = \begin{bmatrix} 0 \\ 5.51 \end{bmatrix}, \quad h_{v1}(\Delta t) = \begin{bmatrix} 0 \\ 1.84 \end{bmatrix},$$
$$h_{s2}(\Delta t) = \begin{bmatrix} -2.19 \\ 3.16 \end{bmatrix}, \quad h_{v2}(\Delta t) = \begin{bmatrix} 0.21 \\ 1.72 \end{bmatrix},$$
$$h_{s3}(\Delta t) = \begin{bmatrix} 2.19 \\ 3.16 \end{bmatrix}, \quad h_{v3}(\Delta t) = \begin{bmatrix} -0.21 \\ 1.72 \end{bmatrix}$$

and the corresponding control strategies (18) with $s_i(t_f)$ and $v_i(t_f)$ replaced by the estimated terminal states $h_{si}(\Delta t)$ and $h_{vi}(\Delta t)$ for $i = 1, 2, 3$ are shown as red circled trajectories in figure 2. We can see that the largest terminal state estimation error is 13.4% and the resulting control inputs slightly deviate from the original Nash strategies. This estimation error can be dramatically reduced with larger values of k_1, k_2, and k_3. Therefore, if we implement the same algorithm with $\Delta t = 0.1$ and $k_1 = k_2 = k_3 = 40$, then the final estimate values becomes

$$h_{s1}(\Delta t) = \begin{bmatrix} 0 \\ 5.89 \end{bmatrix}, \quad h_{v1}(\Delta t) = \begin{bmatrix} 0 \\ 1.96 \end{bmatrix},$$
$$h_{s2}(\Delta t) = \begin{bmatrix} -2.20 \\ 3.54 \end{bmatrix}, \quad h_{v2}(\Delta t) = \begin{bmatrix} 0.23 \\ 1.86 \end{bmatrix},$$
$$h_{s3}(\Delta t) = \begin{bmatrix} 2.20 \\ 3.54 \end{bmatrix}, \quad h_{v3}(\Delta t) = \begin{bmatrix} -0.23 \\ 1.86 \end{bmatrix}.$$

and the largest terminal state estimation error is 3%. In this case, the corresponding control strategies (18) with $s_i(t_f)$ and $v_i(t_f)$ replaced by the estimated terminal states $h_{si}(\Delta t)$ and $h_{vi}(\Delta t)$ for $i = 1, 2, 3$ are shown as green crossed trajectories in figure 2, which are almost identical to original Nash strategies. The motion trajectories under various control inputs are shown in Figure 3. Graphically, there does not exist much difference between the trajectories under the original Nash strategies and the ones under designed strategies. Quantitatively, we are able to calculate the total terminal formation error with respect to the desired displaced among the agents, which is

$$\eta = \sum_{i=1}^{3} \|s_1(t_f) - s_i(t_f) - \mu_{1i}\|, \tag{32}$$

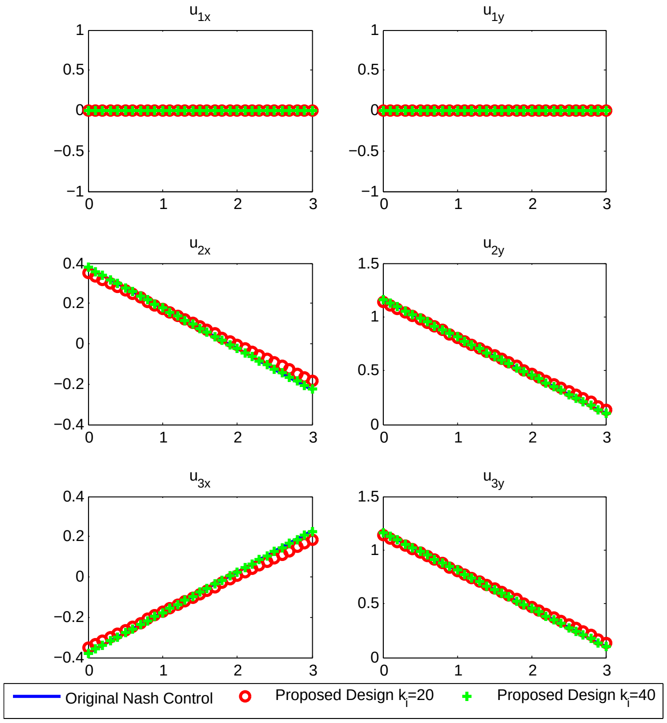

Fig. 2: Nash and proposed designed strategies

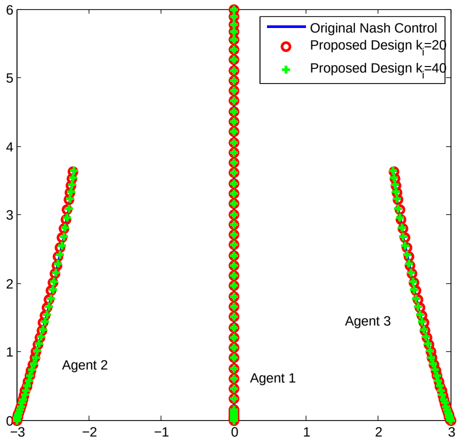

Fig. 3: Motion trajectories under different control strategies

where $s_i(t_f)$ is agent i's terminal position vector under certain control strategies. Therefore, the total terminal formation errors based on different values of $r_1 = r_2 = r_3 = r$ and $k_1 = k_2 = k_3 = k$ can be summarized into Table I. From the table, it is clear to see that the total terminal formation error decreases as k_i increase because as mentioned, larger k_i will result in more accurate terminal state estimates and more accurate control inputs (with respect to the Nash strategies). It is also clear to see that the total terminal formation error increases as r increases. This is because that if agents put higher penalties on the control effort or energy (i.e., the quadratic term of the control input in (2)), they will try to reduce more on control energy instead of the terminal formation errors in order to minimize their performance indices in (2) during the time period of the formation control. Based on more simulation data, the relationship among η, k, and r can be depicted in Figure 4. As we can see in the figure, when r is small, the variance of the total terminal formation error η among different k is small, and when r is large, the variance of the total terminal formation error η among different k becomes large. Another application of Figure 4 is to determine the lowest update gain k needs to achieve a desired terminal formation error. For instance, if we want to determine a update gain that can achieve a maximum terminal formation error of 2 given $r = 2$, then we can find that a value of k less than 6 is good choice from the figure. Therefore, this figure provides a intuitive way of choosing a proper update gain, especially to avoid choosing a high gain which is undesired in many real life applications.

	Nash	$k = 20$	$k = 40$
$\eta\ \ (r = 0.2)$	0.2162	0.2177	0.2162
$\eta\ \ (r = 1)$	0.8164	0.8513	0.8179
$\eta\ \ (r = 5)$	2.6541	2.6920	2.6560

TABLE I: Total terminal formation errors in Scenario 1

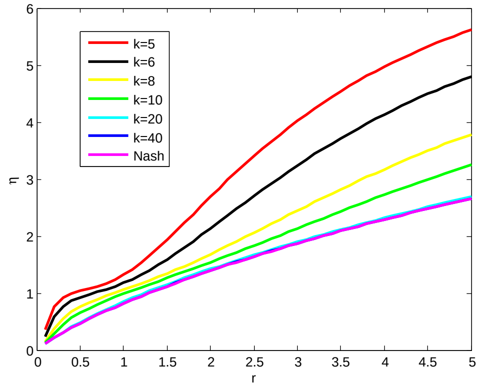

Fig. 4: Relationship among η, r, and k

Scenario 2: We now consider the scenario where there are three agents with undirected communication topology as shown in Figure 5. With this communication topology, both Nash and Pareto strategies can be derived. The initial states of the agents and the performance index parameters are the same as the ones in Scenario 1. For both Nash strategy design and Pareto strategy design, the gain k_i in the terminal state estimation equation (19) and (30) are set equal to 40 for all $i = 1, 2, 3$. For the Pareto strategies, the convex parameters in (27) are set to be $\alpha_1 = \alpha_2 = \alpha_3 = 1/3$. The Laplacian matrix for the noncooperative in (25) and the one for the cooperative case in (31) are

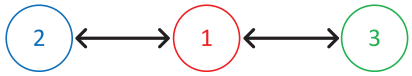

Fig. 5: Three agents with undirected graph

$$\mathcal{L} = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \mathcal{L} = \begin{bmatrix} \frac{4}{3} & -\frac{2}{3} & -\frac{2}{3} \\ -\frac{2}{3} & \frac{2}{3} & 0 \\ -\frac{2}{3} & 0 & \frac{2}{3} \end{bmatrix},$$

respectively. The resulting motion trajectories under both Nash and Pareto strategies are shown in Figure 6. The total terminal formation errors under different choices of the convex parameters are summarized in Table II. As we can clearly see in the table, the agents can actually achieve lower formation error using Pareto strategies than using Nash strategies. This is

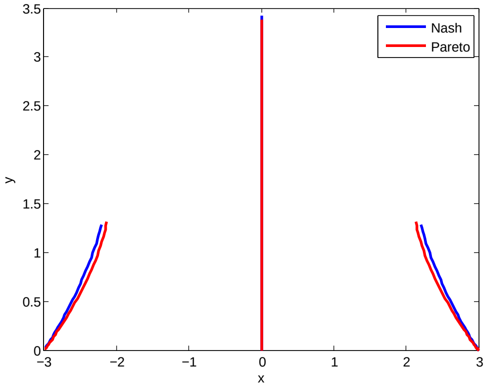

Fig. 6: Motion trajectories under Nash and Pareto strategies

	η
Nash	0.4856
Pareto ($\alpha_1 = \alpha_2 = \alpha_3 = 1/3$)	0.3033
Pareto ($\alpha_1 = 1/6, \alpha_2 = 2/3, \alpha_3 = 1/6$)	0.3128
Pareto ($\alpha_1 = 2/3, \alpha_2 = 1/6, \alpha_3 = 1/6$)	0.1791
Pareto ($\alpha_1 = 0.9, \alpha_2 = 0.05, \alpha_3 = 0.05$)	0.0602

TABLE II: Total terminal formation errors in Scenario 2

because the agents are assumed to cooperate with each other in the Pareto optimality. Also note that as we place more weight on agent 1's performance index, the total formation error can be reduced. This is because it is much easier to move agent 1 alone to form the desired formation instead of moving all the three agents together.

VIII. Conclusion

This paper considers the design of open-loop Nash and Pareto strategies for multi-agent formation control problems. The proposed approach utilizes the distributed estimation of the terminal state variables among the agents through network-based information exchange. The convergence rate of the terminal state estimation algorithm is determined by a scalar gain and can be made sufficiently fast to get a better approximation of the real Nash and Pareto strategies. An illustrative example is solved with two scenarios and various comparisons have been made.