

## Nash equilibrium-based geometric pattern formation control for nonholonomic mobile robots

Seung-Mok Lee, Hanguen Kim, Serin Lee and Hyun Myung<sup>\*</sup>

*Urban Robotics Laboratory, KAIST,  
291 Daehak-ro, Yuseong-gu, Daejeon 305-701, Korea*

*(Received March 6, 2013, Revised August 21, 2013, Accepted October 3, 2013)*

**Abstract.** This paper deals with the problem of steering a group of mobile robots along a reference path while maintaining a desired geometric formation. To solve this problem, the overall formation is decomposed into numerous geometric patterns composed of pairs of robots, and the state of the geometric patterns is defined. A control algorithm for the problem is proposed based on the Nash equilibrium strategies incorporating receding horizon control (RHC), also known as model predictive control (MPC). Each robot calculates a control input over a finite prediction horizon and transmits this control input to its neighbor. Considering the motion of the other robots in the prediction horizon, each robot calculates the optimal control strategy to achieve its goals: tracking a reference path and maintaining a desired formation. The performance of the proposed algorithm is validated using numerical simulations.

**Keywords:** nash equilibrium; geometric pattern formation (GPF); formation control; nonholonomic mobile robots; receding horizon control (RHC); model predictive control (MPC)

### 1. Introduction

A multi-robot coordination system is a promising alternative to a single robot system because it provides a higher level of robustness as a result of its redundancy and the potential for simpler functionality in each robot. Moreover, the possibility of conducting work in parallel allows various applications, e.g., cooperative transport, reconnaissance, coverage, and exploration tasks. In order to perform these tasks, robots must be capable of tracking a reference path while maintaining a desired formation.

Several approaches to formation control have been investigated in previous researches. The classical approaches can be categorized into three types: behavioral approaches, leader-follower approaches, and virtual structure approaches. The behavioral approach (Balch and Arkin 1998) consists of some low-level behaviors (e.g., collision/obstacle avoidance, formation keeping, and goal seeking) to accomplish the group behavior. This is one of the simplest methods to implement, but the main problem is that it is difficult to analyze the approach theoretically and guarantee the convergence of the formation to a desired configuration. In the leader-follower approach (Desai *et al.* 2001, Tanner *et al.* 2004, Consolini *et al.* 2008, Chen *et al.* 2010), one of the robots is

---

<sup>\*</sup>Corresponding author, Professor, E-mail: [hmyung@kaist.ac.kr](mailto:hmyung@kaist.ac.kr)

considered to be a leader robot, while the rest are considered to be followers, which follow the leader by maintaining their range and/or bearing relative to the leader. The main drawback of this approach is that it depends only on the leader to form a formation. Hence, if the leader moves in an unexpected trajectory (e.g., turning sharply), then tracking the desired trajectory produced by the leader's movement is not easy for followers in nonholonomic condition. This will be shown in Section 4. The virtual structure approach (Lewis and Tan 1997) regards the whole formation as a single rigid structure, where each robot tracks the relative trajectory formed by the movement of the structure. The conventional method for this approach, however, cannot be applied to formations that are time-varying, and it requires that the full states of the robots be shared with each other. Time-varying formations (Egerstedt and Hu 2001, Tabuada *et al.* 2005, Broek *et al.* 2009, Sun *et al.* 2009) and the problem of limited information exchange between the robots in a team (Do and Pan 2007) have been considered. Recently, a multi-robot formation control scheme (Ou *et al.* 2012) based on Lyapunov function approach is proposed to form desired geometric patterns within a finite-time. This approach requires the communication topology graph to be connected and undirected. A neural network (NN)-based nonlinear optimal control scheme (Dierks *et al.* 2013) is proposed for mobile robot formations in the presence obstacles, and it shows better performance compared with that of a nonoptimal NN controller.

Receding horizon control (RHC) (Fontes 2001, Gu and Hu 2005), also known as model predictive control (MPC), has been studied for application to multi-robot formation control in a distributed way (Dunbar and Murray 2006), modeled by fully actuated systems. RHC incorporating a linear quadratic Nash differential game, called a receding horizon (RH) Nash controller, was developed by Gu (2008) through the use of graph theory. The robots participating in the game have different cost functions from each other, and they calculate control policies depending on their initial state. Most of the cited researches on distributed RHC for multi-robot formation, however, have focused on fully actuated systems modeled by second order dynamics. For nonholonomic mobile robots, distributed formation control laws were developed by Dong and Farrell (2008) and Moshtagh *et al.* (2009), but the desired formation was very simple, e.g., maintaining a constant distance or a consensus for all headings. More recently, a leader-follower formation incorporating an RHC scheme was proposed to achieve various formation requirements such as formation forming, maintaining, and switching (Chen *et al.* 2010).

The purpose of this paper is to provide a solution to the problem of steering a group of nonholonomic mobile robots along a reference path while maintaining a desired formation. To deal with this problem, this paper proposes a predictive control strategy based on the Nash equilibria (Jank and Abou-Kandil 2003, Engwerda 2005), which is a set of the best strategies of players involving a game in which each player is assumed to know the strategies of the others. The overall formation is decomposed into numerous pairs of robots, and then its state is defined as a set of parameters for a geometric pattern. Each robot calculates the local optimal control strategy to minimize its predefined cost function. Therefore, the formation will converge to a desired geometric configuration.

The advantages of the proposed method can be summarized as follows:

- This paper can deal with a nonholonomic type of robot. The distributed control algorithms (Dunbar and Murray 2006, Gu 2008) were proposed for fully actuated systems modeled by double integrator. Therefore, they can only deal with holonomic systems. The proposed method in this paper overcomes this limitation.
- The control algorithm proposed in this paper is superior to the leader-follower approach. In the proposed algorithm, considering the other robot's motion over finite horizon, each robot

maintains the balance between tracking a trajectory and maintaining geometric formations. Thus, the robots show good performance when turning sharply while maintaining desired formations. The formation control laws (Desai *et al.* 2001, Tanner *et al.* 2004, Consolini *et al.* 2008, Chen *et al.* 2010, Gu 2008), based on leader-follower approach, however, do not show good performance on the reference paths that needs dramatic rotations. In this paper, the proposed algorithm was compared with the leader-follower approach intensively and statistically.

The rest of this paper is organized as follows. Section 2 presents a system description with a group of nonholonomic mobile robots. Section 3 proposes a Nash equilibrium-based predictive control algorithm. The simulation results for the proposed algorithm are provided and discussed in Section 4. Finally, a conclusion is presented in Section 5.

## 2. System description

### 2.1 Formation state

Consider a differential drive mobile robot  $i$  in a group of  $n$  robots that have nonholonomic constraint ( $i=1, \dots, n$ ). The motion state defined by  $p_i=[x_i, y_i, \theta_i]^T$  can be described in discrete-time form as

$$\begin{bmatrix} x_i(t_{k+1}) \\ y_i(t_{k+1}) \\ \theta_i(t_{k+1}) \end{bmatrix} = \begin{bmatrix} x_i(t_k) + \Delta T \cos \theta_i(t_k) v_i(t_k) \\ y_i(t_k) + \Delta T \sin \theta_i(t_k) v_i(t_k) \\ \theta_i(t_k) + \Delta T \omega_i(t_k) \end{bmatrix} \quad (1)$$

where  $p_i$  is described by its position  $(x_i, y_i)$  and orientation  $(\theta_i)$ ,  $v_i$  and  $\omega_i$  are the linear and angular velocities of each robot  $i$ , respectively,  $t_k$  is a time step, and  $\Delta T$  is a sample time. The interconnections between the robots can be topologically described by a directed graph. In this paper, we only deal with a graph associated with unidirectional information flow, referred to as a directed cyclic graph (Gross and Yellen 2004), for simplicity in dealing with a geometrically complex formation system involving nonholonomic mobile robots. Examples of directed cyclic graphs are shown in Fig. 1. Let us consider a couple of nodes (robots) and the orientation of the edge between the nodes. In a given graph, the orientation of the edge indicates the information flow from a source node to a sink node. Thus, regarding the sink node as robot  $i$ , the source node can be considered to be an upstream neighbor of robot  $i$ , which is denoted by  $-i$ . In addition, a robot that receives state information from  $i$ , can be defined as a downstream neighbor of robot  $i$ , denoted by  $+i$ . The following assumption is made throughout this paper.

**Assumption 1 (Connectivity)** *The interconnection between the robots is a directed cyclic graph with all the edges being oriented in the same direction.*

The tracking error of a pair of robots  $i$  and  $-i$  is now derived considering global frame  $G$ , coordinate frame  $O_i$ , and rotation frame  $R_i$ , as shown in Fig. 2. We introduce a state vector,  $p_{c,i}^G$ , in global frame  $G$  as follows

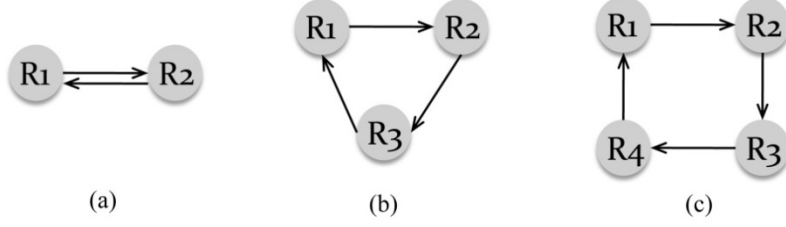


Fig. 1 Examples of directed cyclic graph. In each case, the relationship of robots is as follows: (a)  $-2/+2=1$ ,  $-1/+1=2$ . (b)  $-2/+3=1$ ,  $-3/+1=2$ ,  $-1/+2=3$ . (c)  $-2/+4=1$ ,  $-3/+1=2$ ,  $-4/+2=3$ ,  $-1/+3=4$ .  $-i=j$  indicates that robot  $i$  receives its information from robot  $j$ ;  $+i=j$  indicates that robot  $i$  delivers the information to robot  $j$ .

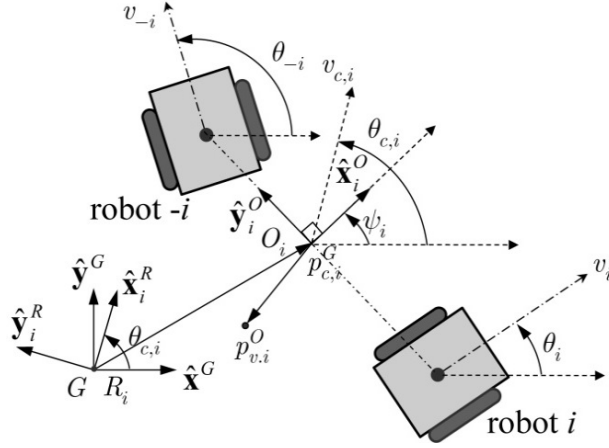


Fig. 2 State description of a geometric pattern with a pair of robots

$$p_{c,i}^G = \begin{bmatrix} x_{c,i} \\ y_{c,i} \\ \theta_{c,i} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} x_i + x_{-i} \\ y_i + y_{-i} \\ \theta_i + \theta_{-i} \end{bmatrix}$$

where  $(x_{c,i}, y_{c,i})$  denotes the center position of the pair of robots and  $\theta_{c,i}$  denotes its orientation defined by the average of  $\theta_i$  and  $\theta_{-i}$ . In frame  $O_i$ ,  $\hat{\mathbf{y}}_i^O$  is a unit vector pointing toward the position of robot  $-i$  from  $p_{c,i}^G$ , and  $\hat{\mathbf{x}}_i^O$  is an orthogonal axis to  $\hat{\mathbf{y}}_i^O$ . Frame  $R_i$  can be obtained by rotating global frame  $G$  by  $\theta_{c,i}$ . Here, we introduce a virtual center,  $p_{v,i}^O$ , fixed at coordinate frame  $O_i$ , which is a vector pointing the formation center inferred by the pair of robots  $i$  and  $-i$  such that the formation center follows a given reference path,  $p_r^G$ . The tracking error,  $p_{e,i}^R$ , between reference path  $p_r^G$  and  $p_{v,i}^G$  in rotation frame  $R_i$  can be represented as

$$\begin{aligned} p_{e,i}^R &= T(\theta_{c,i}) [p_r^G - p_{v,i}^G] \\ &= T(\theta_{c,i}) [p_r^G - (p_{c,i}^G + T(-\psi_i) p_{v,i}^O)], \end{aligned} \quad (2)$$

where  $T(\cdot)$  is a rotational transformation matrix, i.e.,

$$T(\cdot) = \begin{bmatrix} \cos(\cdot) & \sin(\cdot) & 0 \\ -\sin(\cdot) & \cos(\cdot) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the reference path at time  $t_k$  is given by  $p_r^G(t_k) = [x_r(t_k) \ y_r(t_k) \ \theta_r(t_k)]^T$ , where  $\theta_r(t_k) = \arctan 2(y_r(t_k) - y_r(t_{k-1}), x_r(t_k) - x_r(t_{k-1}))$ , and it is assumed that  $p_r^G$  is continuously differentiable at any time.

Next, in order to describe the geometric shape of the formation, a geometric pattern is defined as  $q_i^G = [l_i, \psi_i]^T$ , where  $l_i = \sqrt{(x_i - x_{-i})^2 + (y_i - y_{-i})^2}$  and  $\psi_i = \arctan 2(x_i - x_{-i}, y_i - y_{-i})$ , which are invariant under the transformation of the reference frame, i.e.,  $q_i^G = q_i^O = q_i^R$ . With the given desired geometric pattern,  $q_{r,i}^R = [l_{r,i}, \psi_{r,i}]^T$ , the geometric pattern error is defined as  $q_{e,i}^R = q_{r,i}^R - q_i^R$ . Note that a formation can be described by geometric patterns; thus, the number of geometric patterns is equal to the number of edges on their network graph.

The formation state error,  $z_{e,i}$ , consisting of tracking error  $p_{e,i}^R = [x_{e,i}, y_{e,i}, \theta_{e,i}]^T$  and geometric pattern error  $q_{e,i}^R = [l_{e,i}, \psi_{e,i}]^T$ , can be described for each robot  $i$ , as  $z_{e,i} = [x_{e,i}, y_{e,i}, \theta_{e,i}, l_{e,i}, \psi_{e,i}]^T$  where  $x_{e,i} = \cos \theta_{c,i}(x_r - x_{c,i}) + \sin \theta_{c,i}(y_r - y_{c,i}) - \cos(\theta_{c,i} - \psi_i)p_x - \sin(\theta_{c,i} - \psi_i)p_y$ ,  $y_{e,i} = -\sin \theta_{c,i}(x_r - x_{c,i}) + \cos \theta_{c,i}(y_r - y_{c,i}) + \sin(\theta_{c,i} - \psi_i)p_x - \cos(\theta_{c,i} - \psi_i)p_y$ ,  $\theta_{e,i} = \theta_r - \theta_{c,i}$ ,  $l_{e,i} = l_{r,i} - l_i$ , and  $\psi_{e,i} = \psi_{r,i} - \psi_i$ . By differentiating  $z_{e,i}$  with respect to time, the error state equation of the formation state for robot  $i$  can be represented as

$$z_{e,i}(t_{k+1}) = f_i(z_{e,i}(t_k), u_i(t_k), u_{-i}(t_k)) \quad (3)$$

where  $u_i = [v_i, \omega_i]^T$ ,  $u_{-i} = [v_{-i}, \omega_{-i}]^T$ , and  $f_i: \mathbb{R}^5 \times \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^5$ . The elements of the obtained error state difference equation  $z_{e,i}(t_{k+1})$  are as follows

$$\begin{aligned} x_{e,i}(t_{k+1}) &= x_{e,i}(t_k) + \Delta T [\omega_{c,i}(t_k)(y_{e,i}(t_k) + \xi_{1,i}(t_k)) + v_r(t_k) \cos \theta_{e,i}(t_k) - v_{c,i}(t_k)] \\ &\quad - \Delta T \frac{\xi_{1,i}(t_k)}{l_i(t_k)} [v_i(t_k) \cos(\psi_i(t_k) - \theta_i(t_k)) - v_{-i}(t_k) \cos(\psi_i(t_k) - \theta_{-i}(t_k))], \\ y_{e,i}(t_{k+1}) &= y_{e,i}(t_k) - \Delta T [\omega_{c,i}(t_k)(x_{e,i}(t_k) - \xi_{2,i}(t_k)) - v_r(t_k) \sin \theta_{e,i}(t_k)] \\ &\quad - \Delta T \frac{\xi_{2,i}(t_k)}{l_i(t_k)} [v_i(t_k) \cos(\psi_i(t_k) - \theta_i(t_k)) - v_{-i}(t_k) \cos(\psi_i(t_k) - \theta_{-i}(t_k))], \\ \theta_{e,i}(t_{k+1}) &= \theta_{e,i}(t_k) + \Delta T (\omega_r(t_k) - \omega_{c,i}(t_k)), \\ l_{e,i}(t_{k+1}) &= l_{e,i}(t_k) + \Delta T \dot{l}_r(t_k) - \Delta T (v_i(t_k) \sin(\psi_i(t_k) - \theta_i(t_k))) + \Delta T v_{-i}(t_k) \sin(\psi_i(t_k) - \theta_{-i}(t_k)), \\ \psi_{e,i}(t_{k+1}) &= \psi_{e,i}(t_k) + \Delta T \omega_r(t_k) - \frac{\Delta T}{l_i(t_k)} [(v_i(t_k) \cos(\psi_i(t_k) - \theta_i(t_k))) - (v_{-i}(t_k) \cos(\psi_i(t_k) - \theta_{-i}(t_k)))] \end{aligned}$$

where  $v_{c,i} = (v_i + v_{-i})/2$ ,  $\omega_{c,i} = (\omega_i + \omega_{-i})/2$ ,  $\xi_{1,i} = p_x \sin(\theta_{c,i} - \psi_i) - p_y \cos(\theta_{c,i} - \psi_i)$ ,  $\xi_{2,i} = p_x \cos(\theta_{c,i} - \psi_i) + p_y \sin(\theta_{c,i} - \psi_i)$ , and  $v_r$  and  $\omega_r$  are the desired linear and angular velocities of the center of the formation, which can be derived by differentiating  $p_r$ .

## 2.2 Control objectives

The control objective is to achieve two goals at the same time: tracking a reference path and

maintaining a desired formation in a cooperative and distributed way. Suppose that robot  $-i$  (i.e., an upstream neighbor of  $i$ ) transmits its state information predicted over a receding horizon to robot  $i$ . Based on the future state information of its upstream neighbor, robot  $i$  makes a decision about where it should move. This problem is called the geometric pattern formation (GPF) problem in this paper and is stated formally in the following.

**Problem Definition 1 (GPF)** *Consider a group of nonholonomic mobile robots, given a reference path  $p_r(t_k)$  and desired geometric patterns  $q_{r,i}(t_k)$ , which are differentiable in time with bounded time-derivatives. For each robot  $i$ , find a predictive controller such that the total formation state error,  $\|z_{e,i}(t_k)\|$ , converges to zero as  $t_k \rightarrow \infty$ .*

Note that  $p_{v,i}$  for  $i = 1, \dots, n$  would be different from each other with respect to global frame  $G$  initially. However, if  $\|z_{e,i}\|$  converges to zero,  $p_{v,i}$  indicate the same position with the others. Therefore, as viewed from an overall perspective, the formation will converge to a desired state.

### 3. Nash equilibrium-based local predictive control strategy

In this section, a local predictive control algorithm based on Nash equilibrium is proposed for a GPF problem in a distributed way. The algorithm provides the optimal control strategy to achieve the goals of reference path tracking and formation keeping.

#### 3.1 Local predictive control based on nash equilibrium

As described in Problem Definition 1, two control objectives should be accomplished. Considering the system in Eq. (3) with an upstream neighbor  $-i$ , the cost function,  $J_i$ , for robot  $i$  over finite horizon  $N$  corresponding to update time step  $k$  can be expressed as follows

$$J_i \equiv \frac{1}{2} z_{e,i}(t_{k+N})^2_Q + \frac{1}{2} \sum_{j=0}^{N-1} [z_{e,i}(t_{k+j})^2_Q + u_i(t_{k+j})^2_R + u_{-i}(t_{k+j})^2_R] \quad (4)$$

where  $Q = Q^T \geq 0$ ,  $R = R^T \geq 0$ , and  $x_Q^2$  represents  $x^T Q x$ . Each robot  $i$  tries to minimize its own cost function Eq. (4) subject to Eq. (3) by using the state information of robot  $-i$  over the prediction horizon to make a correct decision according to the local predictive control strategy, i.e., depending on the future trajectory of robot  $-i$ , robot  $i$  computes its optimal policy for the prediction horizon. In the literature on dynamic games, this problem is known as the open-loop Nash nonzero-sum linear quadratic differential game (Jank and Abou-Kandil 2003, Engwerda 2005), and solving the problem gives the optimal control strategy, that is the Nash equilibrium strategy. In order to apply the Nash equilibrium strategy to system Eq. (3), we restrict the game to a two-player case, and Eq. (3) is modeled by a linear discrete-time process model as follows

$$\hat{z}_{e,i}(t_{k+1}) = A_i(t_k) \hat{z}_{e,i}(t_k) + B_i(t_k) u_i(t_k) + B_{-i}(t_k) u_{-i}(t_k) \quad (5)$$

where  $\hat{z}_{e,i}$  is the linearized state error, and  $A_i$ ,  $B_i$ , and  $B_{-i}$  are Jacobians of Eq. (3) given by  $A_i = \partial f_i / \partial z_{e,i}$ ,  $B_i = \partial f_i / \partial u_i$ , and  $B_{-i} = \partial f_i / \partial u_{-i}$ . The details of the Jacobian are provided in Appendix A. To obtain optimal control inputs, a new performance index  $H_i$  considering the constraints in Eq. (5) is defined as follows

$$H_i = \hat{J}_i + \sum_{k=0}^{N-1} \lambda_i^T(t_{k+1}) [A_i(t_k) \hat{z}_{e,i}(t_k) + B_i(t_k) u_i(t_k) + B_{-i}(t_k) u_{-i}(t_k) - \hat{z}_{e,i}(t_{k+1})] \quad (6)$$

where  $\lambda_i(k) = K_i(k) \hat{z}_{e,i}(k)$  is the Lagrangian multiplier for  $k=1, \dots, N$ , and  $\hat{J}_i$  is defined in terms of  $\hat{z}_{e,i}$  as follows

$$\hat{J}_i \equiv \frac{1}{2} \|\hat{z}_{e,i}(t_{k+N})\|_Q^2 + \frac{1}{2} \sum_{j=0}^{N-1} [\|\hat{z}_{e,i}(t_{k+j})\|_Q^2 + \|u_i(t_{k+j})\|_R^2 + \|u_{-i}(t_{k+j})\|_R^2].$$

Setting the partial differentiation of Eq. (6) with respect to  $\hat{z}_{e,i}$  to zero,  $\lambda_i$  can be rewritten as

$$\lambda_i(t_k) = A_i^T(t_k) K_i(t_{k+1}) \hat{z}_{e,i}(t_{k+1}) + Q^T \hat{z}_{e,i}(t_k). \quad (7)$$

The Nash equilibrium strategies  $(u_i^*, u_{-i}^*)$  are defined by the condition  $J_i(u_i^*, u_{-i}^*) \leq J_i(u_i, u_{-i}^*)$  for  $i = 1, \dots, n$ . Using the Nash strategy, each robot cooperates with the others in order to move toward an equilibrium state. Minimization of Eq. (6) with respect to  $u_i$  and  $u_{-i}$  yields the open-loop Nash strategies (Jank and Abou-Kandil 2003, Engwerda 2005)

$$u_i(t_k) = -R^{-T} B_i^T(t_k) \lambda_i(t_{k+1}), \quad (8)$$

$$u_{-i}(t_k) = -R^{-T} B_{-i}^T(t_k) \lambda_i(t_{k+1}). \quad (9)$$

Substituting Eqs. (8) and (9) into Eq. (5) and again substituting it into Eq. (7) yields the Riccati difference equation

$$\begin{aligned} K_i(t_k) &= A_i^T(t_k) K_i(t_{k+1}) W^{-1}(t_k) A_i(t_k) + Q^T, \\ K_i(t_N) &= Q, k = 1, \dots, N-1, \end{aligned} \quad (10)$$

where  $W(t_k) = I + B_i(t_k) R^{-1} B_i^T(t_k) K_i(t_{k+1}) + B_{-i}(t_k) R^{-1} B_{-i}^T(t_k) K_i(t_{k+1})$ . By substituting Eq. (5) into Eq. (8), the control function,  $u_i(t_k)$ , is given by

$$u_i(t_k) = -[I + R^{-T} B_i^T K_i(t_{k+1}) B_i(t_k)]^{-1} R^{-T} B_i^T(t_k) K_i(t_{k+1}) [A_i(t_k) \hat{z}_{e,i}(t_k) + B_{-i}(t_k) u_{-i}(t_k)]. \quad (11)$$

Once  $K_i(t_k)$  is obtained through a backward calculation, as in Eq. (10),  $u_i(t_k)$  can be computed through a forward calculation using Eq. (11). Thus, the local predictive strategy for robot  $i$  can be computed via Riccati iterations. According to Theorem 7.2 in Engwerda (2005), if the Riccati difference equation (10) has a solution, then the linear quadratic differential game has a unique open-loop Nash equilibrium for every initial state. Based on the theorem, the existence of the solution to Eqs. (10) and (11) is proved in the following theorems.

**Theorem 1** *If  $K_i(t_{k+1})$  is positive semidefinite, then  $W(t_k)$  is positive definite and invertible, where  $W(t_k) = I + B_i(t_k) R^{-1} B_i^T(t_k) K_i(t_{k+1}) + B_{-i}(t_k) R^{-1} B_{-i}^T(t_k) K_i(t_{k+1})$ .*

**Proof.** Since the matrix,  $B_i(t_k) R^{-1} B_i^T(t_k)$ , in  $W(t_k)$  is symmetric and  $R \geq 0$  is diagonal, it can be rewritten as

$$\begin{aligned} B_i(t_k)R^{-1}B_i^T(t_k) &= B_i(t_k)R^{-1/2}R^{-1/2}B_i^T(t_k) \\ &= (B_i(t_k)R^{-1/2})(B_i(t_k)R^{-1/2})^T. \end{aligned}$$

Then,

$$\begin{aligned} v^T B_i(t_k)R^{-1}B_i^T(t_k)v &= v^T (B_i(t_k)R^{-1/2})(B_i(t_k)R^{-1/2})^T v \\ &= \|R^{-1/2}B_i^T(t_k)v\|_2^2 \geq 0 \end{aligned}$$

for any nonzero vector  $v$ . Thus,  $B_i(t_k)R^{-1}B_i^T(t_k) \geq 0$ . As indicated in Theorem 3 in Meenakshi and Rajian (1999), a product of positive semidefinite matrices is also positive semidefinite. If  $K_i(t_{k+1}) \geq 0$ , then  $B_i(t_k)R^{-1}B_i^T(t_k)K_i(t_{k+1}) \geq 0$ . In the similar manner, it can be also found that  $B_{-i}(t_k)R^{-1}B_{-i}^T(t_k)K_i(t_{k+1}) \geq 0$ . The fact that  $B_i(t_k)R^{-1}B_i^T(t_k)K_i(t_{k+1}) \geq 0$ ,  $B_{-i}(t_k)R^{-1}B_{-i}^T(t_k)K_i(t_{k+1}) \geq 0$ , and  $I > 0$ , implies that  $W(t_k) > 0$  and it is also invertible.

**Theorem 2** *If  $K_i(t_{k+1})$  is symmetric, then  $K_i(t_{k+1})W^{-1}(t_k)$  is symmetric.*

**Proof.** With the fact that  $K_i(t_{k+1}) = K_i^T(t_{k+1})$ , the following result can be obtained.

$$\begin{aligned} K_i(t_{k+1})W^{-1}(t_k) &= K_i(t_{k+1})[I + B_i(t_k)R^{-1}B_i^T(t_k)K_i(t_{k+1}) + B_{-i}(t_k)R^{-1}B_{-i}^T(t_k)K_i(t_{k+1})]^{-1} \\ &= K_i(t_{k+1})[K_i^{-1}(t_{k+1})(I + K_i(t_{k+1})B_i(t_k)R^{-1}B_i^T(t_k) + K_i(t_{k+1})B_{-i}(t_k)R^{-1}B_{-i}^T(t_k))K_i(t_{k+1})]^{-1} \\ &= K_i(t_{k+1})K_i^{-1}(t_{k+1})[I + K_i(t_{k+1})B_i(t_k)R^{-1}B_i^T(t_k) + K_i(t_{k+1})B_{-i}(t_k)R^{-1}B_{-i}^T(t_k)]^{-1}K_i(t_{k+1}) \\ &= [ (I + B_i(t_k)R^{-1}B_i^T(t_k)K_i^T(t_{k+1}) + B_{-i}(t_k)R^{-1}B_{-i}^T(t_k)K_i^T(t_{k+1}))^T ]^{-1}K_i(t_{k+1}) \\ &= [I + B_i(t_k)R^{-1}B_i^T(t_k)K_i(t_{k+1}) + B_{-i}(t_k)R^{-1}B_{-i}^T(t_k)K_i(t_{k+1})]^{-T}K_i(t_{k+1}) \\ &= W^{-T}(t_k)K_i(t_{k+1}). \end{aligned}$$

**Theorem 3 (Existence)** *If  $K_i(t_N)$  is positive definite (positive semidefinite) and  $A_i(t_k)$  is a nonsingular matrix, then there exists a solution of  $u_i(t_k)$  for  $k=1, \dots, N-1$ .*

**Proof.** For  $k \in \{1, \dots, N-1\}$ , let's begin with  $k = N-1$ . Since  $K(t_N) = Q \geq 0$ ,  $W(t_{N-1}) > 0$  is invertible, and its inverse is also positive definite. As appeared in Theorem 3.6 in Chen (1999), every real symmetric matrix can be decomposed into a product of a matrix and its transpose, i.e., Theorem 2 implies that there exists a nonsingular matrix  $M(t_k)$  such that  $K_i(t_{k+1})W^{-1}(t_k) = M^T(t_k)M(t_k)$ . Thus

$$\begin{aligned} K_i(t_{N-1}) &= A_i^T(t_{N-1})K_i(t_N)W^{-1}(t_{N-1})A_i(t_{N-1}) + Q^T \\ &= A_i^T(t_{N-1})M^T(t_{N-1})M(t_{N-1})A_i(t_{N-1}) + Q^T \geq 0 \end{aligned}$$

since  $Q \geq 0$ . Repeating the above process for  $k$  from  $N-2$  to 1, we can obtained that  $K_i(t_k) \geq 0$  for all  $k$ . Using Theorem 1,  $W(t_k)$  is invertible over  $k=1, \dots, N-1$ . Therefore,  $u_i(t_k)$  has a solution given by Eq. (11) for  $k=1, \dots, N-1$ .



### 3.2 Distributed predictive control algorithm for GPF

Let  $\delta t$  denote the control time interval and  $\Delta T$  denote the prediction time interval, where  $\delta t \leq \Delta T$ . Using the received state error from its upstream neighbor  $-i$ , robot  $i$  predicts the future formation state error,  $z_{e,i}$ , with time interval  $\Delta T$  at update time step  $k$ . Thus, based on the roughly predicted motion of robot  $-i$ , robot  $i$  moves to reduce the cost function in Eq. (4). For each robot  $i$  at update time step  $t_k$ , we define:

- $\bar{p}_i(t_{k+j} | t_k)$ : the predicted motion state
- $\bar{z}_{e,i}(t_{k+j} | t_k)$ : the predicted formation state error
- $\bar{u}_i(t_{k+j} | t_k)$ : the predicted control input

where  $j$  denotes a time step for the prediction horizon.

Using these notations, the procedure of the distributed local predictive control algorithm for GPF is described as follows.

1. Based on a desired formation of robots, construct feasible desired geometric patterns,  $q_{r,i}(t_k)$ , for each robot  $i$ .
2. Given  $z_{e,i}(t_{k+j})$  at initial time  $j = 0$ , the predicted control input,  $\bar{u}_i(t_{k+j} | t_k)$ , over the prediction horizon (i.e.,  $j = 0, 1, \dots, N-1$ ) is set to zero, and then each robot transmits the predicted control function,  $\bar{u}_i(t_{k+j} | t_k)$ , and the current motion state,  $p_i(t_k)$ , to its downstream neighbor robot  $+i$ .
3. Based on the received  $\bar{u}_{-i}(t_{k+j} | t_k)$  and  $p_{-i}(t_k)$  from robot  $-i$ , each robot  $i$  calculates  $\bar{p}_{-i}(t_{k+j} | t_k)$  for the prediction horizon.
4. Using  $\bar{u}_{-i}(t_{k+j} | t_k)$ ,  $\bar{p}_{-i}(t_{k+j} | t_k)$ ,  $v_x(t_{k+j})$ , and  $\omega_r(t_{k+j})$ , calculate the error state,  $\bar{z}_{e,i}(t_{k+j} | t_k)$ , and the matrices,  $\bar{A}_i(t_{k+j} | t_k)$ ,  $\bar{B}_i(t_{k+j} | t_k)$ , and  $\bar{B}_{-i}(t_{k+j} | t_k)$ , corresponding to the prediction horizon.
5. Given the boundary condition,  $K_i(t_N) = Q$ , perform backward calculations of  $K_i(t_k)$  in Eq. (10).
6. Perform forward calculations to obtain  $\bar{u}_i(t_{k+j} | t_k)$  by using Eq. (11) with the error model in Eq. (5).
7. The control input is applied to robot  $i$  only over period  $[t_k, t_k + \delta t)$  where  $t_k$  is the time at step  $k$  and  $\delta t \leq \Delta T$ , and then robot  $i$  transmits  $\bar{u}_i(t_{k+j} | t_k)$  and  $p_i(t_k)$  to robot  $+i$ .
8. Update the time step as  $t_{k+1} \leftarrow t_k + \delta t$ .
9. Repeat the procedure from step 3 until a certain termination condition is satisfied.

When calculating  $\bar{A}_i(t_{k+j} | t_k)$ ,  $\bar{B}_i(t_{k+j} | t_k)$ , and  $\bar{B}_{-i}(t_{k+j} | t_k)$  in step 4, we use  $\bar{u}_i(t_{k+j} | t_{k-1})$  instead of  $\bar{u}_i(t_{k+j} | t_k)$  because  $\bar{u}_i(t_{k+j} | t_k)$  is not accessible at update step  $t_k$ . However, if  $\delta t \ll \Delta T$  (about 10 times), then the state predicted at  $t_{k-1}$  would be similar to the state predicted at step  $t_k$ . Therefore, the assumption that  $\bar{u}(t_{k+j} | t_k) \cong \bar{u}(t_{k+j} | t_{k-1})$  is acceptable.

## 4. Simulation results

In this section, we present several simulation results to validate the proposed algorithm. The

number of prediction horizon steps is selected as  $N = 20$ , while the control time interval and prediction time interval are selected to be  $\delta t = 0.01$  s and  $\Delta T = 0.1$  s, respectively. Thus, the prediction horizon is 2 s.

In the first test, we present a simulation in which two robots maintain an in-line formation. The two robots use the network graph described in Fig. 1(a). This implies that the upstream neighbor of robot 1 is robot 2, i.e.,  $-1=2$  and  $-2=1$ . The reference path for the first test is Ref. path 1 given by Table 1, which starts from the origin with  $\theta_r(t_0) = \pi/6$  rad. The initial poses of the robots are set to  $p_1 = [-0.05, -0.055, \pi/10]^T$  and  $p_2 = [-0.05, 0.055, \pi/10]^T$ , respectively. The desired geometric patterns for the two robots are  $q_r^1 = [0.1, \theta_r]^T$  and  $q_r^2 = [0.1, -\theta_r]^T$ , which requires them to move along the reference path with a parallel formation and keep themselves aligned with a direction perpendicular to the reference path. To track the reference path by the center of the formation, the virtual center is selected to be  $p_{v,i} = [0, 0, 0]^T$ . The weight matrices are selected as  $Q = \text{diag}[0.1, 0.1, 0.1, 0.5, 0.001]$  and  $R = \text{diag}[0.1, 0.1]$ .

To show the performance of the proposed algorithm, the leader-follower approach is also tested. The pose and control input of the leader are defined as  $[x_l, y_l, \theta_l]^T$  and  $[v_l, \omega_l]^T$ , respectively. Given the follower's pose,  $[x_f, y_f, \theta_f]^T$ , the control input for follower  $[v_f, \omega_f]^T$  is calculated by a Lyapunov-based controller (Chen *et al.* 2010)

$$\begin{aligned} v_f &= v_l \cos e_\theta + k_x e_x, \\ \omega_f &= \omega_l + k_\theta e_\theta + v_l e_y \frac{\sin e_\theta}{e_\theta}, \end{aligned}$$

where  $e_x = \cos \theta_f (x_d - x_f) + \sin \theta_f (y_d - y_f)$ ,  $e_y = -\sin \theta_f (x_d - x_f) + \cos \theta_f (y_d - y_f)$ ,  $e_\theta = \theta_l - \theta_f$ ,  $x_d = x_l + l_r \cos(\psi_r + \theta_l)$ ,  $y_d = y_l + l_r \sin(\psi_r + \theta_l)$ , and  $k_x$  and  $k_\theta$  are fixed control gains set to 0.1. To maintain the desired in-line formation,  $l_r$  and  $\psi_r$  are set to 0.1 and  $\pi/2$ , respectively.

The trajectories generated by the proposed algorithm and the leader-follower approach are shown in Fig. 3. The squares, triangles, and circles denote the trajectories for robot 1, robot 2, and the formation center, respectively, by the proposed method. The dashed-line denotes the follower's trajectory, in which robot 1 is regarded as its leader. From the trajectories of robot 1 and robot 2, it is shown that the two control objectives of trajectory tracking and formation keeping can be optimally achieved using the proposed method. When the robots perform turns to track the reference path, the robot located on the outside turns quickly while the other robot turns slowly to maintain the in-line formation. This can also be seen from the control input computed by the proposed algorithm in Fig. 4. This work cannot be accomplished by the leader-follower approach in the case of nonholonomic mobile robots, as shown in Fig. 3. As can be seen from Fig. 5, the follower generates a large bearing error when its leader (i.e., robot 1) turns abruptly because of nonholonomic constraint. However, the trajectories generated by the proposed algorithm produce only small geometric pattern errors in formation keeping compared to the leader-follower approach.

To validate the superiority of the proposed algorithm compared to the conventional leader-follower approach, we also performed a statistical analysis of the formation errors from all the trajectories with four reference paths. The reference paths used in the analysis are summarized in Table 1. The geometric pattern formation error distributions,  $q_{e,i}$ , which are generated by the leader-follower approach and the proposed algorithm, are shown in box and whisker plots as in Figs. 6 and 7. The results show that the errors in the proposed algorithm are very small compared to the leader-follower approach. Furthermore, the error deviations of the proposed algorithm are

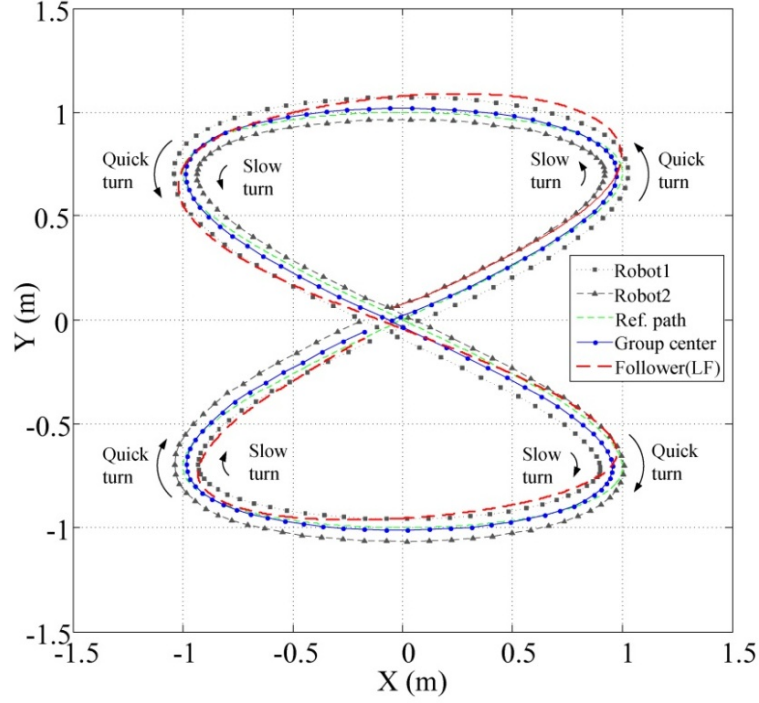


Fig. 3 In-line formation tracking '8'-shaped path. When turning along the trajectory, the inside robot turns slowly while the outside robot turns quickly to maintain the formation

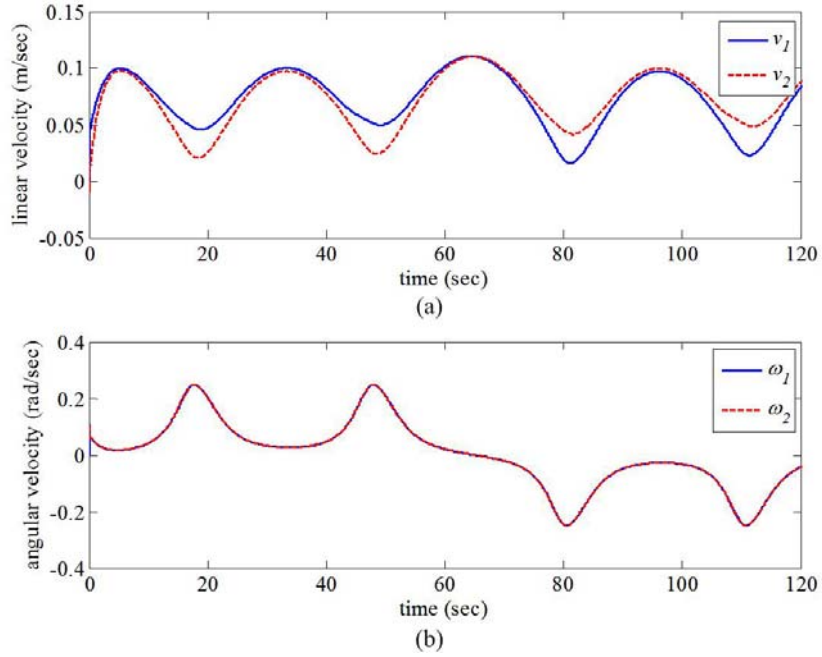


Fig. 4 Control inputs computed by the proposed algorithm for tracking '8'-shaped path shown in Fig. 3

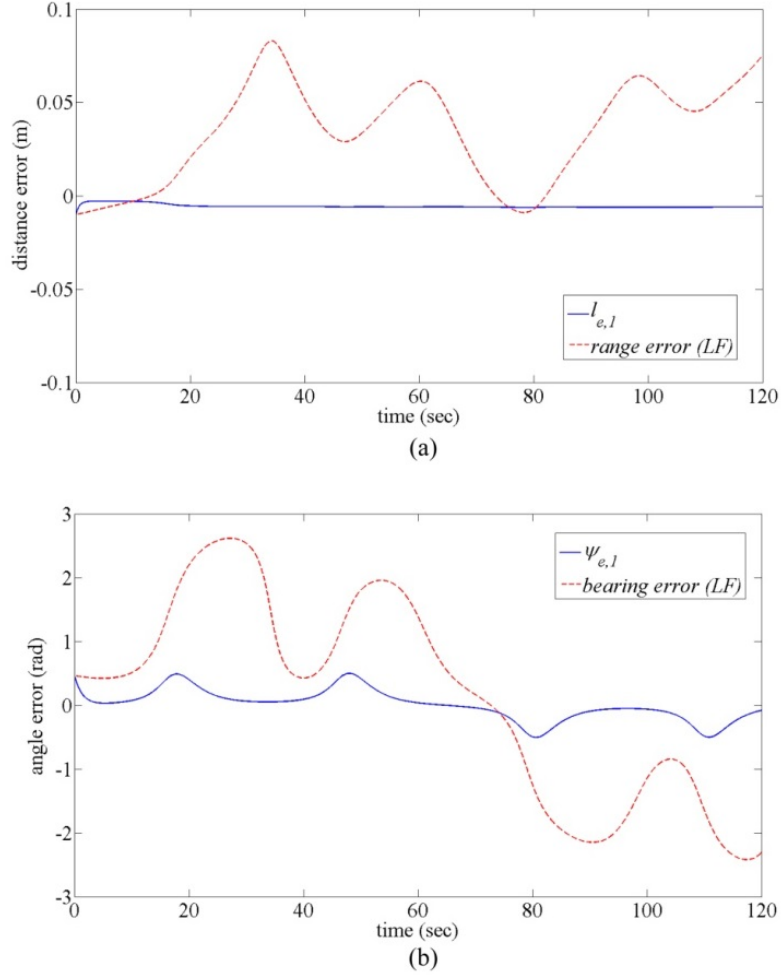


Fig. 5 Plot of the geometric pattern errors during tracking '8'-shaped path. (a)  $l_{e,1}(=l_{e,2})$ , and range error of leader-follower (LF). (b)  $\psi_{e,1}(=\psi_{e,2})$ , and bearing error of LF

Table 1 Reference paths for statistical analysis

	Equations	Shape
Ref. path 1	$x_r(t_k) = \sin(t_k/10)$ $y_r(t_k) = \sin(t_k/20)$	'8'
Ref. path 2	$x_r(t_k) = 0.8\cos(t_k/20)$ $y_r(t_k) = 0.8\sin(t_k/20)$	circle
Ref. path 3	$x_r(t_k) = t_k/10$ $y_r(t_k) = 0.8\sin(t_k/10)$	sinusoidal
Ref. path 4	$x_r(t_k) = 0.01\cos(t_k/10)$ $y_r(t_k) = 0.01\sin(t_k/10)$	spiral

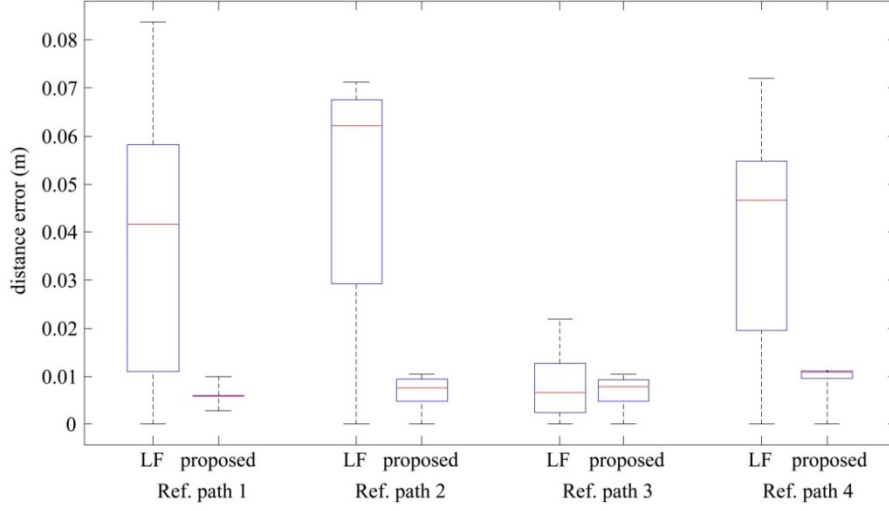


Fig. 6 Distance errors in box and whisker plots for four reference paths. Each box has three horizontal lines at the lower quartile, median, and upper quartile values. The whisker shows the range of the errors

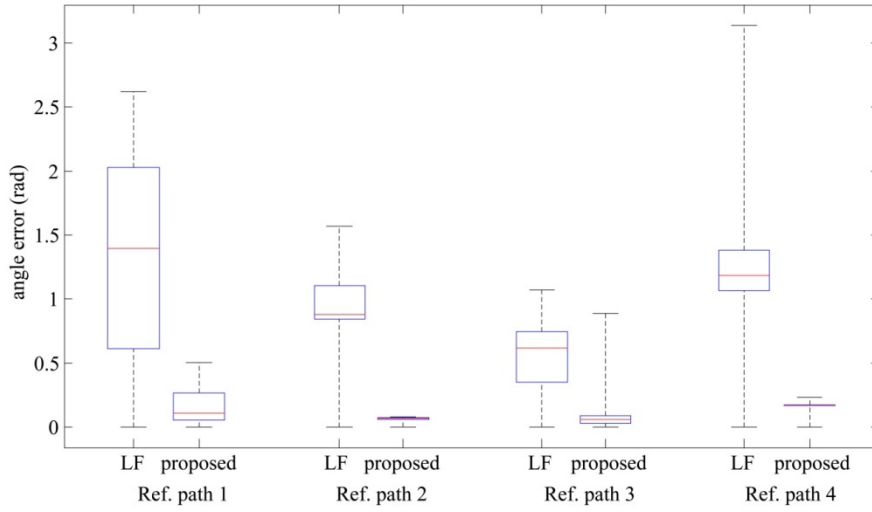


Fig. 7 Angle errors in box and whisker plots for four reference paths. Each box has three horizontal lines at the lower quartile, median, and upper quartile values. The whisker shows the range of the errors

much smaller than the leader-follower approach in all the cases.

In the next test, three mobile robots are used with the network graph described in Fig. 1(b). The reference path is linear, which is given by  $x_r(t_k) = 0$  and  $y_r(t_k) = 0.1t_k$  for  $t_k < 20$  s, and  $x_r(t_k) = 0.1(t_k - 20)$  and  $y_r(t_k) = 0.1t_k$  for  $t_k \geq 20$  s. Initially, the robots are located at  $p_1 = [0, 0.05, \pi/2]^T$ ,  $p_2 = [-0.05, -0.05, \pi/2]^T$ , and  $p_3 = [0.05, -0.05, \pi/2]^T$ , respectively. The desired geometric pattern is an equilateral triangle formation whose desired separation between the robots are  $l_{r,i} = 0.1$  and  $p_{v,i} = [(\sqrt{3}/6)l_{r,i}, (\sqrt{3}/6)l_{r,i}, 0]^T$  for  $i = 1, 2, 3$ . The weight matrices are set as  $Q = \text{diag}[0.1, 0.1, 0.1, 0.2, 0.01]$  and  $R = \text{diag}[0.1, 0.1]$ . The resulting trajectories are shown in Fig. 8. It is shown

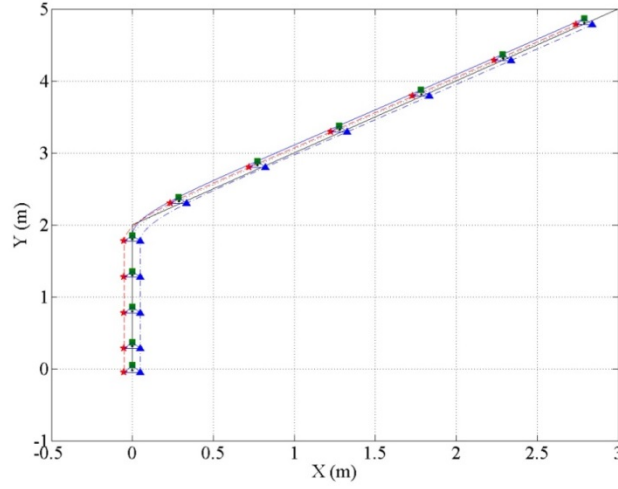


Fig. 8 Triangular formation tracking a line. The robot locations sampled at every 5s are indicated by squares, stars, and triangles for  $i = 1, 2, 3$ , and the black dots denote the center of the formation

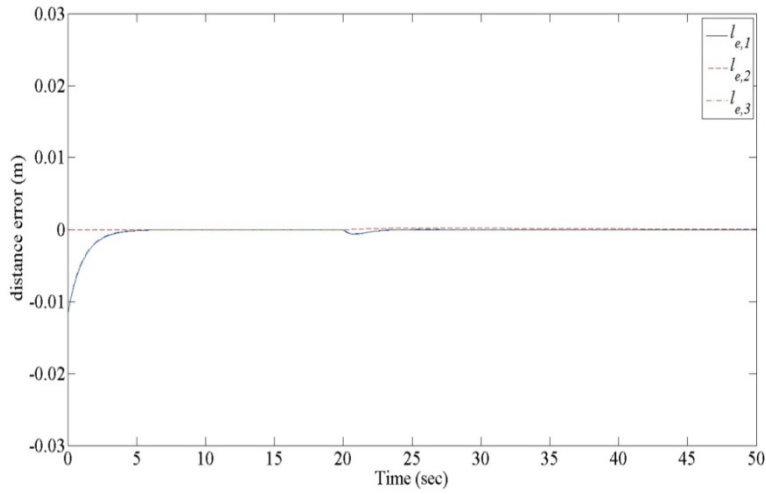


Fig. 9 Relative distance errors between the robots during tracking a line

that three robots maintain a triangular formation while the center of the formation tracks the given reference path using only the local information, i.e., the state of its upstream neighbor. Figure 9 shows the relative distance errors between the robots, where the errors converge to zero, with a small deviation when the direction of the reference path is changed.

Finally, three robots are tested with a circle reference path given by Ref. path 2 in Table 1. The robots are placed with the initial configuration of  $p_1 = [0.8, 0.065, 1.5]^T$ ,  $p_2 = [0.75, -0.05, 1.5]^T$ , and  $p_3 = [0.85, -0.05, 1.5]^T$ . The other parameters are selected to be  $l_{r,i} = 0.1$ ,  $p_{v,i} = [(\sqrt{3}/6)l_{r,i}, (\sqrt{3}/6)l_{r,i}, 0]^T$ ,  $Q = \text{diag}[0.1, 0.1, 0.1, 0.2, 0.01]$ , and  $R = \text{diag}[0.1, 0.1]$ . Figure 10

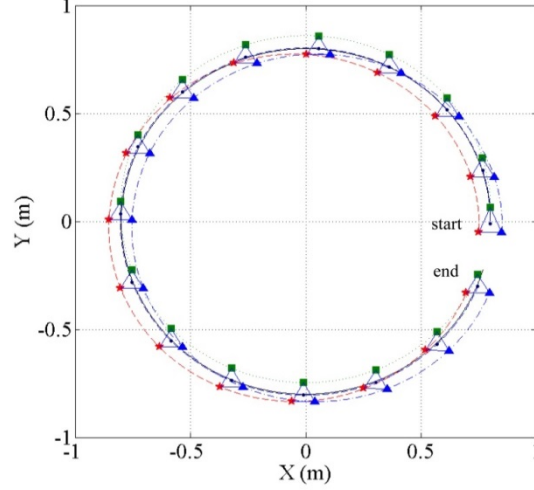


Fig. 10 Triangular formation tracking a circle. The robot locations sampled at every 8 s are indicated by squares, stars, and triangles for  $i = 1, 2, 3$ , and the black dots denote the center of the formation. The headings are tangential to the robot's path

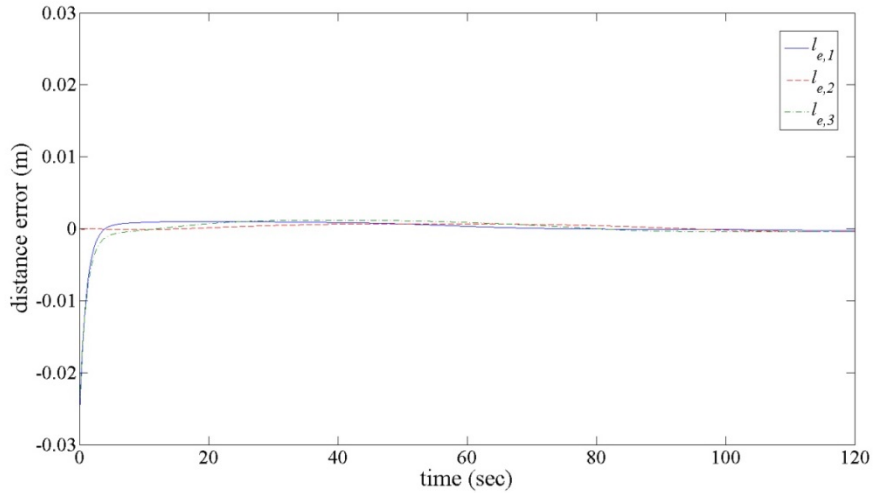


Fig. 11 Relative distance errors between the robots during tracking a circle

shows the robots maintaining the desired geometric formations while following the reference path. Fig. 11 shows the relative distance errors,  $l_{e,i}$  between the robots. However, there are still formation errors among the robots as shown in Fig. 11. The errors are caused by suboptimal solutions obtained through the linearization of the multiple nonlinear systems. Although the errors can be reduced by increasing the prediction horizon or increasing the weight matrix  $Q$ , these can result in some problems: the increased prediction horizon results in increased computation time and the increased  $Q$  also causes large control inputs. Thus the prediction horizon and weight matrix should

be set to moderate values.

## 5. Conclusions

In this paper, a distributed predictive control scheme was developed for the GPF problem with nonholonomic mobile robots. On a directed cyclic graph, the overall formation is decomposed into pairs of robots, and then the state between the robots is defined. Referring to the control strategies of the others, each robot calculates the optimal control strategy using the proposed algorithm based on the Nash equilibrium to achieve its goal. In the simulations, using the proposed algorithm, the robots moved to track a given reference path, while maintaining the desired formation. Moreover, even with a reference path having sharp edges, it was verified that the formation of robots could still track the reference path by predicting the future states.

## Acknowledgments

This research was supported by the MOTIE (The Ministry of Trade, Industry and Energy), Korea, under the Human Resources Development Program for Convergence Robot Specialists support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2013-H1502-13-1001). This research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (grant number NRF-2013R1A1A1A05011746). Mr. H. Kim was supported by Korea Ministry of Land, Infrastructure and Transport (MOLIT) as U-City Master and Doctor Course Grant Program.

## References

- Balch, T. and Arkin, R.C. (1998), "Behavior-based formation control for multi-robot teams", *IEEE Trans. Robot. Autom.*, **14**(6), 926-939.
- van den Broek, T.H.A., van de Wouw, N. and Nijmeijer, H. (2009), "Formation control of unicycle mobile robots: a virtual structure approach", *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, December 16-18.
- Chen, C. (1999), *Linear System Theory and Design*, (3rd Edition), Oxford University Press.
- Chen, J., Sun, D., Yang, J. and Chen, H. (2010), "Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme", *Int. J. Robot. Res.*, **29**(6), 727-747.
- Consolini, L., Morbidi, F., Prattichizzo, D. and Tosques, M. (2008), "Leader-follower formation control of nonholonomic mobile robots with input constraints", *Automatica*, **44**(5), 1343-1349.
- Desai, J.P., Ostrowski, J.P. and Kumar, V. (2001), "Modeling and control of formations of nonholonomic mobile robots", *IEEE Trans. Robot. Autom.*, **17**(6), 905-908.
- Dierks, T., Brenner, B. and Jagannathan, S. (2013), "Neural network-based optimal control of mobile robot formations with reduced information exchange", *IEEE Trans. Control Syst. Technol.*, **21**(4), 1407-1415.
- Do, K.D. and Pan, J. (2007), "Nonlinear formation control of unicycle-type mobile robots", *Robot. Auton. Syst.*, **55**(3), 191-204.
- Dong, W. and Farrell, J.A. (2008), "Cooperative control of multiple nonholonomic mobile agents", *IEEE Trans. Autom. Control*, **53**(6), 1434-1448.
- Dunbar, W.B. and Murray, R.M. (2006), "Distributed receding horizon control for multi-vehicle formation



- stabilization", *Automatica*, **42**(4), 549-558.
- Egerstedt, M. and Hu, X. (2001), "Formation constrained multi-agent control", *IEEE Trans. Robot. Autom.*, **17**(6), 947-951.
- Engwerda, J.C. (2005), *LQ Dynamic Optimization and Differential Games*, Wiley.
- Fontes, F.A.C.C. (2001), "A general framework to design stabilizing nonlinear model predictive controllers", *Syst. Control Lett.*, **42**(2), 127-143.
- Gross, J.L. and Yellen, J. (2004), *Handbook of Graph Theory*, CRC Press.
- Gu, D. and Hu, H. (2005), "A stabilizing receding horizon regulator for nonholonomic mobile robots", *IEEE Trans. Robot.*, **21**(5), 1022-1028.
- Gu, D. (2008), "A differential game approach to formation control", *IEEE Trans. Control Syst. Technol.*, **16**(1), 85-93.
- Jank, G. and Abou-Kandil, H. (2003), "Existence and uniqueness of open-loop Nash equilibria in linear-quadratic discrete time games", *IEEE Trans. Autom. Control*, **48**(2), 267-271.
- Lewis, M.A. and Tan, K. (1997), "High precision formation control of mobile robots using virtual structures", *Auton. Robot.*, **4**(4), 387-403.
- Meenakshi, A.R. and Rajian, C. (1999), "On a product of positive semidefinite matrices", *Linear Alg. Appl.*, **295**(1-3), 3-6.
- Moshtagh, N., Michael, N., Jadbabaie, A. and Daniilidis, K. (2009), "Vision-based, distributed control laws for motion coordination of nonholonomic robots", *IEEE Trans. Robot.*, **25**(4), 851-860.
- Ou, M., Du, H. and Li, S. (2012), "Finite-time formation control of multiple nonholonomic mobile robots", *Int. J. Robust Nonlinear Control*, doi: 10.1002/rnc.2880.
- Sun, D., Wang, C., Shang, W. and Feng, G. (2009), "A synchronization approach to trajectory tracking of multiple mobile robots while maintaining time-varying formations", *IEEE Trans. Robot.*, **25**(5), 1074-1086.
- Tabuada, P., Pappas, G. J. and Lima, P. (2005), "Motion feasibility of multi-agent formations", *IEEE Trans. Robot.*, **21**(3), 387-392.
- Tanner, H.G., Pappas, G.J. and Kumar, V. (2004), "Leader-to-formation stability", *IEEE Trans. Robot. Autom.*, **20**(3), 443-455.

## Appendix

### A. Jacobian matrices of the formation state

In (5), the system state transition matrix  $A_i$  can be obtained by the Jacobian as follows

$$A_i = \begin{bmatrix} 1 & \Delta T \omega_{c,i} & \Delta T a_{13} & \Delta T a_{14} & \Delta T a_{15} \\ -\Delta T \omega_{c,i} & 1 & \Delta T a_{23} & \Delta T a_{24} & \Delta T a_{25} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \Delta T a_{43} & 1 & \Delta T a_{45} \\ 0 & 0 & \Delta T a_{53} & \Delta T a_{54} & 1 + \Delta T a_{55} \end{bmatrix} \quad (12)$$

The elements are

$$\begin{aligned} a_{13} &= -v_r \sin \theta_e + \omega_{c,i} \xi_{1,i}^{\theta_e} - \frac{v_i}{l_i} \left( \xi_{1,i}^{\theta_e} \cos(\psi_i - \theta_i) - 2\xi_{1,i} \sin(\psi_i - \theta_i) \right) \\ &\quad + \frac{v_{-i}}{l_i} \left( \xi_{1,i}^{\theta_e} \cos(\psi_i - \theta_{-i}) - 2\xi_{1,i} \sin(\psi_i - \theta_{-i}) \right), \\ a_{14} &= -\frac{\xi_{1,i}}{l_i^2} (v_i \cos(\psi_i - \theta_i) - v_{-i} \cos(\psi_i - \theta_{-i})), \\ a_{15} &= \omega_{c,i} \xi_{1,i}^{\psi_e} - \frac{v_i}{l_i} \left( \xi_{1,i}^{\psi_e} \cos(\psi_i - \theta_i) + \xi_{1,i} \sin(\psi_i - \theta_i) \right) + \frac{v_{-i}}{l_i} \left( \xi_{1,i}^{\psi_e} \cos(\psi_i - \theta_{-i}) + \xi_{1,i} \sin(\psi_i - \theta_{-i}) \right), \\ a_{23} &= v_r \cos \theta_e + \omega_{c,i} \xi_{2,i}^{\theta_e} - \frac{v_i}{l_i} \left( \xi_{2,i}^{\theta_e} \cos(\psi_i - \theta_i) - 2\xi_{2,i} \sin(\psi_i - \theta_i) \right) \\ &\quad + \frac{v_{-i}}{l_i} \left( \xi_{2,i}^{\theta_e} \cos(\psi_i - \theta_{-i}) - 2\xi_{2,i} \sin(\psi_i - \theta_{-i}) \right), \\ a_{24} &= -\frac{\xi_{2,i}}{l_i^2} (v_i \cos(\psi_i - \theta_i) - v_{-i} \cos(\psi_i - \theta_{-i})), \\ a_{25} &= \omega_{c,i} \xi_{2,i}^{\psi_e} - \frac{v_i}{l_i} \left( \xi_{2,i}^{\psi_e} \cos(\psi_i - \theta_i) + \xi_{2,i} \sin(\psi_i - \theta_i) \right) + \frac{v_{-i}}{l_i} \left( \xi_{2,i}^{\psi_e} \cos(\psi_i - \theta_{-i}) + \xi_{2,i} \sin(\psi_i - \theta_{-i}) \right), \\ a_{43} &= -2(v_i \cos(\psi_i - \theta_i) - v_{-i} \cos(\psi_i - \theta_{-i})), \\ a_{45} &= v_i \cos(\psi_i - \theta_i) - v_{-i} \cos(\psi_i - \theta_{-i}), \\ a_{53} &= \frac{2}{l_i} (v_i \sin(\psi_i - \theta_i) - v_{-i} \sin(\psi_i - \theta_{-i})), \\ a_{54} &= -\frac{1}{l_i^2} (v_i \cos(\psi_i - \theta_i) - v_{-i} \cos(\psi_i - \theta_{-i})), \\ a_{55} &= -\frac{1}{l_i} (v_i \sin(\psi_i - \theta_i) - v_{-i} \sin(\psi_i - \theta_{-i})), \end{aligned}$$

where

$$\begin{aligned}
 \xi_{1,i}^{\theta_e} &\equiv \frac{\partial \xi_{1,i}}{\partial \theta_e} = -p_x \cos(\theta_{c,i} - \psi_i) - p_y \sin(\theta_{c,i} - \psi_i), \\
 \xi_{1,i}^{\psi_e} &\equiv \frac{\partial \xi_{1,i}}{\partial \psi_e} = p_x \cos(\theta_{c,i} - \psi_i) + p_y \sin(\theta_{c,i} - \psi_i), \\
 \xi_{2,i}^{\theta_e} &\equiv \frac{\partial \xi_{2,i}}{\partial \theta_e} = p_x \sin(\theta_{c,i} - \psi_i) - p_y \cos(\theta_{c,i} - \psi_i), \\
 \xi_{2,i}^{\psi_e} &\equiv \frac{\partial \xi_{2,i}}{\partial \psi_e} = -p_x \sin(\theta_{c,i} - \psi_i) + p_y \cos(\theta_{c,i} - \psi_i).
 \end{aligned}$$

The input matrices  $B_i$  and  $B_{-i}$  are also given by

$$B_i = \Delta T \begin{bmatrix} -\frac{1}{2} - \cos(\psi_i - \theta_i) \frac{\xi_{1,i}}{l_i} & \frac{1}{2}(y_e + \xi_{1,i}) \\ -\cos(\psi_i - \theta_i) \frac{\xi_{2,i}}{l_i} & -\frac{1}{2}(x_e - \xi_{2,i}) \\ 0 & -1/2 \\ -\sin(\psi_i - \theta_i) & 0 \\ -\cos(\psi_i - \theta_i)/l_i & 0 \end{bmatrix}, \quad (13)$$

$$B_{-i} = \Delta T \begin{bmatrix} -\frac{1}{2} + \cos(\psi_i - \theta_{-i}) \frac{\xi_{1,i}}{l_i} & \frac{1}{2}(y_e + \xi_{1,i}) \\ \cos(\psi_i - \theta_{-i}) \frac{\xi_{2,i}}{l_i} & -\frac{1}{2}(x_e - \xi_{2,i}) \\ 0 & -1/2 \\ \sin(\psi_i - \theta_{-i}) & 0 \\ \cos(\psi_i - \theta_{-i})/l_i & 0 \end{bmatrix}. \quad (14)$$