IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 1, 2019, accepted March 24, 2019, date of publication March 28, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2907960

Leader-Follower Consensus Multi-Robot Formation Control Using Neurodynamic-Optimization-Based Nonlinear Model Predictive Control

HANZHEN XIAO[1] AND C. L. P. CHEN[1,2], (Fellow, IEEE)
[1]Faculty of Science and Technology, University of Macau, Macau 999078, China
[2]Dalian Maritime University, Dalian 116026, China

Corresponding author: C. L. P. Chen (philip.chen@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572540, Grant 61751202, Grant U1813203, Grant U1801262, and Grant 61751205, in part by the Key Program for International S&T Cooperation Projects of China under Grant 2016YFE0121200, in part by the Macau Science and Technology Development Fund (FDCT) under Grant 079/2017/A2, Grant 024/2015/AMJ, and Grant 019/2015/A1, and in part by the Multiyear Research Grants of the University of Macau.

ABSTRACT This paper investigates a nonlinear-model-predictive-control (NMPC)-strategy-based distributed leader–follower consensus multi-robot formation system. The control objective of this system is to design a group of nonholonomic robots to converge into the desired geometric pattern and to track a designed path. A directed graph that specifies communication topology for the formation is given. A leader–follower consensus formation problem based on the mobile robot kinematic model is obtained, which is further reformulated into a constrained nonlinear minimization problem through the NMPC strategy. A general projection neural network (GPNN) is implemented to efficiently derive the optimal control inputs for the robots. The simulation results verify the effectiveness of the proposed formation algorithm.

INDEX TERMS Nonholonomic Multi-robot formation, leader-follower consensus system, nonlinear model predictive control (NMPC), graph theory, general projection neural network (GPNN).

I. INTRODUCTION

In recent years, robot formation, which is one of the most important research areas in multi-robot coordination, has become more and more attractive. Many researchers are interested in its application prospects such as surveillance, transportation, mine sweeping, rescue operations, and geographical exploration. Compared to single robot, a team of robots can offer many superiorities on working. The consensus formation, whose objective is to control a group of robots to reach and maintain a designed geometric pattern during moving, is a typical formation scheme. Meanwhile, owing to Brockett's theorem [1], it is hard to directly implement the differentiable, or even continuous, pure state feedback algorithm on the nonholonomic-robot-based distributed consensus formation problem.

Generally, there are two control paradigms for robot formation: centralized and distributed. In centralized formation, the formation system normally relies on one single chief leader or external resource. The host exchanges information among the robot members and the control inputs are calculated in the host depending on the received information of the whole formation system. While in many cases, robots in the formation only have limited communication ability, i.e., it is hard for the robots to receive all global information, so the centralized formation is hard to be achieved. Different from centralized control, in distributed formation, the robots have more independence where the action of each robot moves according on the behaviors observed from itself and its neighbors. Recently, due to the development of the distributed consensus control, many works have used the graph theory cooperated distributed consensus method to control the multi-agent dynamical system [2], [3] and the formation system [4], [5]. In the distributed consensus multi-robot formation system, a communication topology of the robots can be described by directed graph. The distributed control input for each robot can be obtained only based on the information from its neighbors and itself. Compared with centralized

The associate editor coordinating the review of this manuscript and approving it for publication was Jinpeng Yu.

method, the distributed consensus control approach is superior in computing cost and its flexibility.

There are many methods that have been developed for the distributed robot formation, such as Lyapunov-based control [6], graph theory [7], feedback linearization [8], nonlinear control [9], persistent generation [10] and sliding mode [11]. In the work [4], a nonholonomic formation system is transformed into a consensus state problem and a distributed controller is applied. However, in above works, the state and input constraints are not adequately considered. The additional handling for the system's constraints (such as [12], [13], and [14]) may sometimes be inconvenient. On the other hand, the nonlinear model predictive control (NMPC) strategy can incorporate the state and input limitations into the cost function and obtain a minimization closed-loop optimal problem based on the consensus formation model over a predictive control horizon in each sampling time. In the previous works [15], [16], and [17], NMPC method is used to control the leader-follower wheeled robot formation system with separation-bearing orientation scheme (SBOS) framework and the system's boundaries can be considered. However, in [15]–[18] and [19], the formation systems are constructed by calculating robots' relative relationships but not in a consensus form. In [20], a homogeneous multi-agent consensus system is controlled through the distributed MPC method with input and state constraints. In [21], MPC is used to control the second-order multi-agent flocking system, the input constraints can be handled. However, most of the MPC-based consensus formations are implemented on the coordinate level and there is less work on the wheeled mobile robots. In this work, we apply the NMPC method on the consensus wheeled mobile robot formation system. For dealing with the nonholonomic property brought by the implemented robot, the consensus system is divided into to two subsystems, where the MPC strategy, respectively, can be implemented and the distributed optimal control input for each robot can be obtained accordingly

To deal with the NMPC's constrained optimal problem efficiently, the neurodynamic optimization approach is implemented. In existing works such as [22]–[24] and [25], duality and projection based neurodynamic models have been built for dealing with the convex and pseudoconvex optimization. Compared with other optimization method, the neurodynamic optimization algorithm has the superior performances with robustness global convergence [26], low computational complexity and can process the information in a distributed and parallel way. Inspired by the work in [25], here, for the constrained Quadratic Programming (QP) problem, a general projection neural network (GPNN) is implemented to obtain the optimal solution and the online computational efficiency can be improved.

In this paper, we propose a distributed control method for leader-follower consensus multi-robot formation system with the MPC method. A virtual leader is employed to decide the moving trajectory and is regard as the geometric center of the formation. For the robots which only have limited communication ability, a directed graph is used to describe the communication topology and a consensus error system model is formed. Compared with existing works on controlling consensus formation, the contributions of this work can be list as follows:

1) To overcome the nonholonomic property, the mobile robot kinematic system is transformed and divided into two consensus error subsystems so that the control objective can be achieved through stabilizing these two subsystems in sequence.
2) A constrained NMPC method is proposed for controlling the consensus formation by transforming the consensus subsystems in to QP optimization problems. The system's constraints can be handled by incorporating them into the coefficients of the QP problems.
3) To obtain the optimal inputs for the robots, a neural-dynamic optimization is proposed to solve the constrained QP problem in real time with its high efficiency and low computational complexity.

This work is organized as follows. Section II gives some preliminary knowledge of this work. Section III describes the leader-follower consensus formation system. Section IV introduces the proposed MPC method; the optimization method is shown in Section V. Finally, Section VI gives the results of simulation to verify the effectiveness of the developed algorithm and Section VII concludes this work.

II. PRELIMINARIES

A. GRAPH THEORY

A directed graph $G = (V, E, A)$ is applied to represent the communication relation of the robots. In the graph G, $V = 1, 2, \cdots, M$ represents the nonempty set of M following robots which can be labeled as $R_1, R_2, \cdots, R_M$; the directed edges are represented as $E = \{(i,j), i,j \in V,\ i \neq j\}$; the matrix $A = (a_{ij}) \in R^{M\times M}$ is used to represent the relevant weighted adjacency. We can describe the communication relation of robots as follows: if and only if the information can be transferred from robot j to robot i, $(j, i) \in E$ exists, a_{ij} of A is nonnegative. Here we set: if $(j, i) \notin E$ or $i = j$, $a_{ij} = 0$ and if $(j, i) \in E$, $a_{ij} = 1$.

Define the diagonal matrix $D \in R^{M\times M}$ as:

$$D = diag(d_1, d_2, \cdots, d_M) \tag{1}$$

where d_i is called the in-degree and is defined as:

$$d_i = \sum_{j=1}^{M} a_{ij} \tag{2}$$

$L = D - A$ is the Laplacian matrix, $L \in R^{M\times M}$.

In the directed graph, the link path between i to $j(i \neq j)$ can be represented as a sequence of directed edges $(i, i_1), (i_1, i_2), \cdots, (i_e, i_j)$ where $i_k \in V$, $k = 1, 2, \cdots, e$. The directed path between two robots is not unique. In a directed graph, if and only if there is at least one node in V has a directed path to all the other nodes that a directed spanning tree is existed.

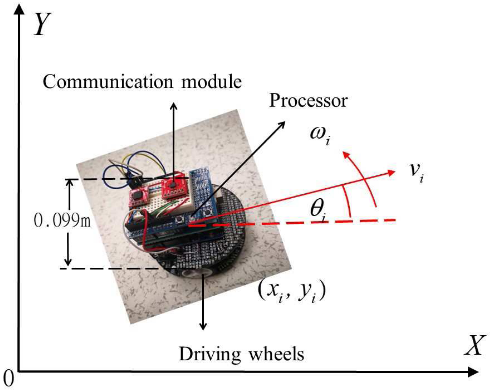


FIGURE 1. Nonholonomic wheeled mobile robot.

In this paper, an virtual robot R_L is considered as the leader robot and the relation between it and other following robots $R_i(i = 1, \cdots, M)$ can be described through a new directed graph $\bar{G}$. The leader R_L cannot be affected by the other follower R_i and can only send the information to a few followers. Define the leader-follower connection weight matrix $B = diag(b_1, b_2, \cdots, b_M)$. $b_i \geq 0$ for $i = 0, 1, \cdots, M$, if information sent by R_L can be received by the follower R_i, $b_i = 1$, otherwise $b_i = 0$. Assume that at least one follower can receive leader's information.

Lemma 1: [27] If a directed spanning tree exists in the directed graph $\bar{G}$, the matrix $F = L + B$ is invertible.

B. NONHOLONOMIC WHEELED MOBILE ROBOT

Fig. 1 shows a typical nonholonomic wheeled mobile robot, its position coordinate can be represented as (x_i, y_i) and its orientation is $\theta_i(t)$. This robot equips two driving wheels with 1.6cm radius for moving, a communication module for exchanging information with other robots and a processor to process the data. The maximum of its linear and angular velocities are $v_{max} = 10m/s$ and $\omega_{max} = 5rad/s$. Its diameter is 9.9cm. The state vector of this robot i can be defined as $X_i = [x_i, y_i, \theta_i]^T$, the kinematics model can be represented as:

$$\begin{aligned} \dot{x}_i(t) &= v_i(t)\cos\theta_i(t) \\ \dot{y}_i(t) &= v_i(t)\sin\theta_i(t) \\ \dot{\theta}_i(t) &= \omega_i(t) \end{aligned} \tag{3}$$

This type of robots can not slip in a lateral direction. Meanwhile, define a virtual leader robot R_L and its moving state is defined as $X_L = [x_L, y_L, \theta_L]^T$.

III. PROBLEM STATEMENT

A. FORMATION OBJECTIVE

As shown in Fig. 2, the desired formation pattern can be defined as $\mathbf{P} = [(p_{1x}, p_{1y}), (p_{2x}, p_{2y}), \ldots, (p_{Mx}, p_{My})]$, where $(p_{ix}, p_{iy})(i = 1, 2, \cdots, M)$ is the desired geometric pattern's orthogonal coordinate of robot R_i. Suppose that the total

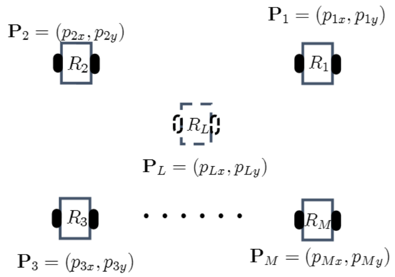


FIGURE 2. The desired formation pattern of the virtual leader robot and follower robots.

M desired geometric patterns satisfy

$$\sum_{i=1}^{M} p_{ix} = p_{Lx}, \sum_{i=1}^{M} p_{iy} = p_{Ly} \tag{4}$$

where (p_{Lx}, p_{Ly}) is the center of the formation pattern and is normally set as original point, i.e., $p_{Lx} = 0$, $p_{Ly} = 0$. The control objective of the formation system can be represented as:

$$\begin{aligned} &\lim_{t\to\infty} (x_i(t) - x_j(t)) = p_{ix} - p_{jx} \\ &\lim_{t\to\infty} (y_i(t) - y_j(t)) = p_{iy} - p_{jy} \\ &\lim_{t\to\infty} (\theta_i(t) - \theta_L(t)) = 0 \end{aligned} \tag{5}$$

$$\begin{aligned} &\lim_{t\to\infty} (x_M(t) - x_L(t)) = 0 \\ &\lim_{t\to\infty} (y_M(t) - y_L(t)) = 0 \end{aligned} \tag{6}$$

B. CONSENSUS ERROR SUBSYSTEM TRANSFORMATION

For each robot R_i $(i = 1, 2, \cdots, M)$ in the formation, we define following transformation:

$$\begin{aligned} z_{1i} &= \theta_i \\ z_{2i} &= (x_i - p_{ix})cos\theta_i + (y_i - p_{iy})sin\theta_i + \alpha sign(u_{1i})z_{3i} \\ z_{3i} &= (x_i - p_{ix})sin\theta_i - (y_i - p_{iy})cos\theta_i \\ u_{1i} &= \omega_i \\ u_{2i} &= v_i - (1 + \alpha^2)u_{1i}z_{3i} \end{aligned} \tag{7}$$

u_{1i} and u_{2i} are the inputs of the transformed system, $sign(\cdot)$ is the signum function, $\alpha > 0$. Let $z_i = [z_{1i}, z_{2i}, z_{3i}]^T$ represents the state vector of the ith system, the dynamic system of (7) can be represented as:

$$\dot{z}_i = \begin{bmatrix} \dot{z}_{1i} \\ \dot{z}_{2i} \\ \dot{z}_{3i} \end{bmatrix} = \begin{bmatrix} u_{1i} \\ u_{2i} + \alpha|u_{1i}|z_{2i} \\ u_{1i}z_{2i} - \alpha|u_{1i}|z_{3i} \end{bmatrix} \tag{8}$$

Then, the control objective (6) becomes

$$\begin{aligned} &\lim_{t\to\infty} (z_{1i}(t) - z_{1L}(t)) = 0 \\ &\lim_{t\to\infty} (z_{2i}(t) - z_{2L}(t)) = 0 \\ &\lim_{t\to\infty} (z_{3i}(t) - z_{3L}(t)) = 0 \\ &\lim_{t\to\infty} (u_{1i}(t) - u_{1L}(t)) = 0 \end{aligned} \tag{9}$$

Lemma 2: [4] If equations (9) hold for $i = 1, 2, \cdots, M$, then equations (5)-(6) can be satisfied, i.e., all the M following robots can reach the desired formation pattern $\mathbf{P}$.

Further, (8) can be transformed into two subsystems. Let $\xi_i = [\xi_{1i}, \xi_{2i}]^T = [z_{2i}, z_{3i}]^T$, we have

$$\dot{z}_{1i} = u_{1i} \tag{10}$$

$$\dot{\xi}_i = \begin{bmatrix} u_{2i} + \alpha|u_{1i}|\xi_{1i} \\ u_{1i}\xi_{1i} - \alpha|u_{1i}|\xi_{2i} \end{bmatrix} \tag{11}$$

Assumption 1: The state of the first subsystem z_{1i} is bounded and u_{1i} is persistent exciting $(1 = 1, 2, \cdots, M)$.

Remark 1: From the system (11) we can see that, if the input u_{1i} in the subsystem (10) vanishes, the subsystem (11) will lost its controllability. In this work, because of the Assumption 1, input u_{1i} dose not converge to 0, so that the proposed nonholonomic system can be controlled.

For controlling the ith robot system $(1 = 1, 2, \cdots, M)$, the angular velocity input $\omega_i = u_{1i}$, while the linear velocity input v_i needs a transformation from u_{2i}:

$$v_i = u_{2i} + (1 + \alpha^2)u_{1i}\xi_{2i}. \tag{12}$$

For each robot i, it can only receive the state information from its neighbors, the communication topology is described by a directed graph $\bar{G}$ in Subsection II-A. Through applying directed graph $\bar{G}$, the consensus errors of two subsystems can be defined as follows:

$$e_{1i} = \sum_{j=1}^{M} a_{ij}(z_{1i} - z_{1j}) + b_i(z_{1i} - z_{1L}) \tag{13}$$

$$e_{2i} = \sum_{j=1}^{M} a_{ij}(\xi_i - \xi_j) + b_i(\xi_i - \xi_L) \tag{14}$$

where e_{1i} and e_{2i} are the consensus errors of two subsystems, a_{ij} is the relevant adjacency weight. Define error vectors $e_1 = [e_{11}, e_{12}, \cdots, e_{1M}]^T \in R^M$ and $e_2 = [e_{21}^T, e_{22}^T, \cdots, e_{2M}^T]^T \in R^{2M}$, the generalized consensus error system including M robots can be formulated as:

$$e_1 = F\tilde{z}_1 \tag{15}$$

$$e_2 = F \otimes 1_2\tilde{\xi} \tag{16}$$

where $F = L+B$, $\tilde{z}_1 = [z_{11}-z_{1L}, z_{12}-z_{1L}, \cdots, z_{1M}-z_{1L}]^T$, $\tilde{\xi} = [(\xi_1-\xi_L)^T, (\xi_2-\xi_L)^T, \cdots, (\xi_M-\xi_L)^T]^T$, $\otimes$ represents the Kronecker product, $1_2 = [1, 1]^T$.

Further, define $u_1 = [u_{11}, u_{12}, \cdots, u_{1M}]^T \in R^M$ and $u_2 = [u_{21}, u_{22}, \cdots, u_{2M}]^T \in R^M$, subsystems (15) and (16) can be represented as following nonlinear affine systems:

$$\dot{e}_1 = h_1(z_1) + s_1(z_1)u_1 \tag{17}$$

$$\dot{e}_2 = h_2(\xi, u_1) + s_2(\xi)u_2 \tag{18}$$

where

$$h_1(z_1) = -Fu_{1L}, s_1(z_1) = F \in R^{M\times M}$$

$$h_2(\xi, u_1) = F \otimes 1_2 \begin{bmatrix} -u_{2L}+\alpha|u_{11}|\xi_{11}-\alpha|u_{1L}|\xi_{1L} \\ u_{11}\xi_{11}-\alpha|u_{11}|\xi_{21}+\alpha|u_{1L}|\xi_{2L} \\ \vdots \\ -u_{2L}+\alpha|u_{1M}|\xi_{1M}-\alpha|u_{1L}|\xi_{1L} \\ u_{1M}\xi_{1M}-\alpha|u_{1M}|\xi_{2M}+\alpha|u_{1L}|\xi_{2L} \end{bmatrix}$$

$$s_2(\xi) = F \otimes 1_2.$$

After above transformation, the control objective (9) can be turned into stabilizing the two transformed consensus subsystems (17) and (18). For each following robot i in the formation, the individual consensus error subsystems can be represented as:

$$\dot{e}_{1i} = h_{1i}(z_1) + s_{1i}(z_{1i})u_{1i} \tag{19}$$

$$\dot{e}_{2i} = h_{2i}(\xi_i, u_{1i}) + s_{2i}(\xi_i)u_{2i} \tag{20}$$

h_{1i}, h_{2i}, s_{1i} and s_{2i} represent the ith row of h_1, h_2, s_1, and s_2 respectively. Through stabilizing these two consensus error subsystems, the robots in the formation can reach the desired geometric pattern and the formation objective can be achieved. That is to achieve:

$$\text{For } i = 1 \cdots M \text{as} t \to \infty, \quad e_{1i} \to 0,\ e_{2i} \to 0 \tag{21}$$

C. DISCRETIZATION OF CONSENSUS FORMATION SYSTEM

As the control method need to be implemented on the robot platforms, the discrete-time form is required. Above two subsystems can be discretized as follows:

$$e(k+1) = e + T\dot{e} \tag{22}$$

where T is the sampling period. Let

$$\mathbf{h}_{1i}(e_{1i}(k)) = e_{1i}(k) + Th_{1i}(z_{1i}),$$
$$\mathbf{s}_{1i}(e_{1i}(k)) = Ts_{1i}(z_{1i}),$$
$$\mathbf{h}_{2i}(e_{2i}(k)) = e_{2i}(k) + Th_{2i}(\xi_i, u_{1i}),$$
$$\mathbf{s}_{2i}(e_{2i}(k)) = Ts_{2i}(\xi),$$

the previous two consensus error subsystems (17) and (18) can be discretized as:

$$e_{1i}(k+1) = \mathbf{h}_{1i}(e_1(k)) + \mathbf{s}_{1i}(e_{1i}(k))u_{1i}(k) \tag{23}$$

$$e_{2i}(k+1) = \mathbf{h}_{2i}(e_{2i}(k), u_{1i}(k)) + \mathbf{s}(e_{2i}(k))u_{2i}(k) \tag{24}$$

IV. NONLINEAR MODEL PREDICTIVE CONTROL STRATEGY

A. CONSENSUS FORMATION SYSTEM WITH INPUT AND STATE CONSTRAINTS

For achieving the control objective, a discrete-time closed-loop optimal control problem can be formed through the nonlinear model predictive control (NMPC) strategy. Then, one of the most important issues should be considered is the system's constraints. As the nonlinear affine systems (23) and (24) have the similar form, we can use e, u, $\mathbf{h}$ and $\mathbf{s}$ to representing e_{ni}, $\mathbf{h}_{ni}$, $\mathbf{s}_{ni}$ and u_{ni}, for $n = 1, 2$, $i = 1, \cdots, M$. The consensus error subsystems with constraints in discrete-time can be represented as:

$$e(k+1) = \mathbf{h}(e(k)) + \mathbf{s}(e(k))u(k) \tag{25}$$

subject to

$$\Delta u_{min} \leqslant \Delta u(k) \leqslant \Delta u_{max} \tag{26}$$

$$u_{min} \leqslant u(k) \leqslant u_{max} \tag{27}$$

$$e_{min} \leqslant e(k) \leqslant e_{max} \tag{28}$$

where $m = 1$ or 2 depends on the ith system; $\mathbf{u} \in R$ and $\Delta\mathbf{u} \in R$ represent the input vector and input increment vector, respectively; $e \in R^m$ represents the state vector; $\mathbf{h}(\cdot)$ and $\mathbf{s}(\cdot)$ are nonlinear continuous functions; $\mathbf{h}(0) = 0$; N and N_u are the prediction horizon and control horizon, respectively, and both satisfy $0 \leq N_u \leq N$; note that the inequalities of constraints (27) means that: for the vector a, its ith element is bounded by relative ith elements in a_{max} and a_{min}.

B. NONLINEAR MODEL PREDICTIVE CONTROL

In the MPC, at each sampling time, the states of system can be predicted within the predictive horizon based on the control model. So a cost function can be formulated by using the predictive state and input sequences. The iterative online optimization process is the distinction between the MPC method and other traditional control methods. Define $a(k+j|k)$ as the predicted value of a at the future time instance $k+j$ based on the information at the current time instance k. For $n = 1, 2$, the predictive states of consensus subsystems (23) and (24) of the ith robot at the future time $k+j$ can be represented as $e_{ni}(k+j|k), j = 1, 2, \ldots, N$, which can be obtained as following predictive process:

$$
\begin{aligned}
&e_{ni}(k+1|k) = \mathbf{h}_{ni}(e_{ni}(k|k-1)) + \mathbf{s}_{ni}(e_{ni}(k|k-1)) \\
&\times(u_{ni}(k-1) + \Delta u_{ni}(k|k)) \\
&e_{ni}(k+2|k) = \mathbf{h}_{ni}(e_{ni}(k+1|k-1)) \\
&+\mathbf{s}_{ni}(e_n(k+1|k-1)) \\
&\times(u_{ni}(k-1) + \Delta u_{ni}(k|k) + \Delta u_{ni}(k+1|k)) \\
&\vdots \\
&e_{ni}(k+N|k) = \mathbf{h}_{ni}(e_{ni}(k+N|k-1)) \\
&+\mathbf{s}_{ni}(e_{ni}(k+N-1|k-1)) \\
&\times(u_{ni}(k-1) + \Delta u_{ni}(k|k) + \ldots + \Delta u_{ni}(k+N_u-1|k))
\end{aligned} \tag{29}
$$

where $u_{ni}(k-1)$ is the previous control input; $\Delta u_{ni}(k+j|k)$ is system's future input increment over the control horizon; $u_{ni}(k+j|k) = u_{ni}(k-1) + \Delta u_{ni}(k|k) + \ldots + \Delta u_{ni}(k+j|k)$ is system's future input. Then cost function of the nth consensus subsystem of the ith robot can be built up as:

$$
\begin{aligned}
J_{ni}(k) = &\sum_{j=1}^{N} e_{ni}^T(k+j|k) Q_{ni} e_{ni}(k+j|k) \\
&+ \sum_{j=0}^{N_u-1} \Delta u_{ni}^T(k+j|k) R_{ni} \Delta u_{ni}(k+j|k)
\end{aligned} \tag{30}
$$

where Q_{ni} and R_{ni} represent appropriate weighting matrices.

Define:

$$
\begin{aligned}
\bar{e}_{ni}(k) &= [e_{ni}(k+1|k), \ldots, e_{ni}(k+N|k)]^T \in R^{mN} \\
\bar{u}_{ni}(k) &= [u_ni(k|k), \ldots, u_{ni}(k+N_u-1|k)]^T \in R^{N_u} \\
\Delta\bar{u}_{ni}(k) &= [\Delta u_{ni}(k|k), \ldots, \Delta u_{ni}(k+N_u-1|k)]^T \in R^{N_u}
\end{aligned}
$$

so the predicted consensus system errors can be represented as:

$$
\bar{e}_{ni}(k) = S_{ni} \Delta\bar{u}_{ni}(k) + \tilde{\mathbf{h}}_{ni} + \tilde{\mathbf{s}}_{ni} \tag{31}
$$

where

$$
S_{ni} = \begin{bmatrix} \mathbf{s}_{ni}(e_{ni}(k|k-1)) & \cdots & 0 \\ \mathbf{s}_{ni}(e_{ni}(k+1|k-1)) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \mathbf{s}_{ni}(e_{ni}(k+N-1|k-1)) & \cdots & \mathbf{s}_n(e_{ni}(k+N-1|k-1)) \end{bmatrix}
$$

$$
\tilde{\mathbf{h}}_{ni} = \begin{bmatrix} \mathbf{h}_{ni}(e_{ni}(k|k-1)) \\ \mathbf{h}_{ni}(e_{ni}(k+1|k-1)) \\ \vdots \\ \mathbf{h}_{ni}(e_{ni}(k+N-1|k-1)) \end{bmatrix},
$$

$$
\tilde{\mathbf{s}}_{ni} = \begin{bmatrix} \mathbf{s}_{ni}(e_n(k|k-1))u_{ni}(k-1) \\ \mathbf{s}_{ni}(e_{ni}(k+1|k-1))u_{ni}(k-1) \\ \vdots \\ \mathbf{s}_{ni}(e_{ni}(k+N-1|k-1))u_{ni}(k-1) \end{bmatrix}.
$$

$S_{ni} \in R^{mN \times N_u}$, $\tilde{\mathbf{h}}_{ni}$ and $\tilde{\mathbf{s}}_{ni} \in R^{mN}$, $m = 1, 2$ depends on the nth subsystem. Through substituting (31) into (30), we can get the optimal problem:

$$
min\ J_{ni}(k) = ||S_{ni}\Delta\bar{u}_{ni}(k) + \tilde{\mathbf{h}}_{ni} + \tilde{\mathbf{s}}_{ni}||^2_{Q_{ni}} + ||\Delta\bar{u}_{ni}||^2_{R_n} \tag{32}
$$

subject to

$$
\Delta\bar{u}_{ni_{min}} \leqslant \Delta\bar{u}_{ni}(k) \leqslant \Delta\bar{u}_{ni_{max}} \tag{33}
$$

$$
\bar{u}_{ni_{min}} \leqslant \bar{u}_{ni}(k-1) \leqslant \bar{u}_{ni_{max}} \tag{34}
$$

$$
\bar{u}_{ni_{min}} \leqslant \bar{u}_{ni}(k-1) + \tilde{I}\Delta\bar{u}_{ni}(k) \leqslant \bar{u}_{ni_{max}} \tag{35}
$$

$$
\bar{e}_{ni_{min}} \leqslant \tilde{\mathbf{h}}_{ni} + \tilde{\mathbf{s}}_{ni} + S_{ni}\Delta\bar{u}_{ni}(k) \leqslant \bar{e}_{ni_{max}} \tag{36}
$$

where $\tilde{I} = \begin{bmatrix} I & 0 & \cdots & 0 \\ I & I & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ I & I & \cdots & I \end{bmatrix} \in R^{N_u \times N_u}$.

Remark 2: Note that when formulating of the optimization problem, robot R_i needs to get the predictive values of S_{ni}, $\mathbf{h}_{ni}$ and $\mathbf{s}_{ni}$. However, in the distributed formation, one robot cannot obtain the predicted state values of its neighbors. So in the practical application, the predictive states of R_i's neighbors can be estimated numerically using neighbors' previous states. Even in controlling a nominal undisturbed system, the predicted value and the actual closed-loop values of NMPC is not necessary to be the same. Hence the S_{ni}, $\mathbf{h}_{ni}$ and $\mathbf{s}_{ni}$ can be obtained based on the estimated predictive states of their neighbors. The errors between the estimated values and the predicted one can be reduced through tuning Q_{ni}, R_{ni}, N and N_u [28] as well as setting compatibility input constraints and sufficient small sampling period [18], meanwhile, the closed-loop stability can be achieved.

Then, a QP problem can be formed from the optimization problem (32)

$$
\begin{aligned}
&min\ \frac{1}{2}\Delta\bar{u}_{ni}^T H_{ni} \Delta\bar{u}_{ni} + r_{ni}^T \Delta\bar{u}_{ni} \\
&s.t\ \ l_{ni} \leqslant A_{ni}\Delta\bar{u}_{ni} \leqslant \hbar_{ni}
\end{aligned} \tag{37}
$$

where

$$H_{ni} = 2(S_{ni}^T Q_{ni} S_{ni} + R_{ni}), \quad r_{ni} = 2S_{ni}^T Q_{ni}(\tilde{\mathbf{s}}_{ni} + \tilde{\mathbf{h}}_{ni})$$

$$A_{ni} = \begin{bmatrix} -\tilde{I} \\ \tilde{I} \\ -S_{ni} \\ S_{ni} \\ I \end{bmatrix}, \quad f_{ni} = \begin{bmatrix} -\bar{u}_{ni_{min}} + \bar{u}_{ni}(k-1) \\ \bar{u}_{ni_{max}} + \bar{u}_{ni}(k-1) \\ -\bar{e}_{ni_{min}} + \tilde{\mathbf{h}}_{ni} + \tilde{\mathbf{s}}_{ni} \\ \bar{e}_{ni_{max}} - \tilde{\mathbf{h}}_{ni} - \tilde{\mathbf{s}}_{ni} \end{bmatrix},$$

$$l_{ni} = \begin{bmatrix} -\infty \\ \bar{u}_{ni_{min}} \end{bmatrix}, \quad \hbar_{ni} = \begin{bmatrix} f_{ni} \\ \bar{u}_{ni_{max}} \end{bmatrix}$$

where $H_{ni} \in R^{N_u \times N_u}$, $r_{ni} \in R^{N_u}$, $A_{ni} \in R^{(2mN+3N_u)\times N_u}$, $f_{ni} \in R^{2mN+2N_u}$. l_{ni} and $\hbar_{ni}$ are the upper/lower bounds of $A_{ni}\Delta\bar{u}_{ni}$ and $l_{ni}/\hbar_{ni} \in R^{2mN+3N_u}$.

The stability analysis of the implemental MPC method have been shown in the previous work [17]. The closed-loop stability of the system can be achieved when there exists optimal input sequences $\bar{u}_{ni}^*(k+1)$ for the QP optimization problem (37) at each time instance k. So the next step is to find an efficient and effective method for obtaining the optimal inputs for each robot.

V. GENERAL PROJECTION NETWORK OPTIMIZATION

After formulating the QP problem (37), for the ith robot ($i = 1, 2, \ldots, M$), we need to solve (37) to obtain the optimal input increment sequences $\Delta\bar{u}_{ni}$ ($n = 1, 2$) for its two relative consensus error subsystems. For the sake of simplicity, $\Delta\bar{u}$ is used to represent the input increment we need to obtain. Firstly, we have following theorem:

Theorem 1: To find an optimal solution for the QP problem (37) is equivalent to find a vector $\Delta\bar{u} \in R^{N_u}$ satisfying following piecewise equation:

$$\varphi\vartheta^+(\Delta\bar{u} - \zeta) + \varrho = K_\Lambda(\varrho - \vartheta^+(\Delta\bar{u} - \zeta) + \varphi\vartheta^+(\Delta\bar{u} - \zeta)) \tag{38}$$

where ϑ^+ is the pseudo-inverse of ϑ; $\vartheta = H_{ni}^{-1}A_{ni}^T$; $\varphi = A_{ni}H_{ni}^{-1}A_{ni}^T$; $\zeta = -H_{ni}^{-1}r_{ni}$; $\varrho = -A_{ni}H_{ni}^{-1}r_{ni}$.
The proof of the Theorem 1 has been illustrated in the work [25].

$K_\Lambda(\cdot)$ is the projection operator as follows:

$$K_\Lambda(a_i) = \begin{cases} a^- & \text{if } a_i < x^-, \\ a_i & \text{if } a^- \leqslant a_i \leqslant a^+, \ \forall i \in R^{2mN+3N_u} \\ a^+ & \text{if } a_i > x^+, \end{cases} \tag{39}$$

and the minimum/maximum boundaries are $a^- = l$ and $a^+ = h$.

Through defining two continuous differentiable vector-valued functions: $R(\Delta\bar{u}) = \varphi\vartheta^+(\Delta\bar{u} - \zeta) + \varrho$, $D(\Delta\bar{u}) = \vartheta^+(\Delta\bar{u} - \zeta)$, neural network's dynamic equation with $\Delta\bar{u} \in R^{N_u}$ as the state vector can be represented as

$$\gamma\frac{d\Delta\bar{u}}{dt} = \beta[K_\Lambda(R(\Delta\bar{u}) - D(\Delta\bar{u})) - R(\Delta\bar{u})] \tag{40}$$

Fig. 3 shows the structure of GPNN, where $\beta = \vartheta$, β_i represents the ith row of the scaling matrix β; γ is a positive constant.

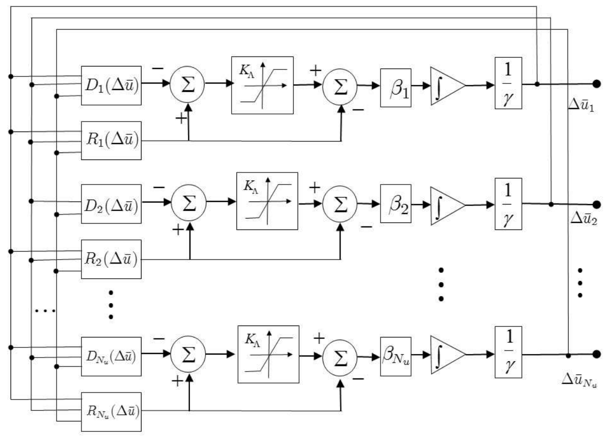


FIGURE 3. Block diagram of the GPNN.

The work in [25] verifies the Lyapunov stability of the optimal problem (37) with this method, the optimal solution $\Delta\bar{u}^*$ is of globally exponentially convergence. Finally, the first element of the outputs in (40) will be used to obtain the optimal inputs for the subsystem.

Remark 3: Generally, there are $2N_u$ additions/subtractions, N_u integrators, N_u processors of projection operator $K_\Lambda(\cdot)$ and N_u processors of vector-valued function $R(\Delta\bar{u})$ and $D(\Delta\bar{u})$ in GPNN. In this paper, for each subsystem in one robot, the dimension of GPNN's state is N_u, so there are totally $2MN_u$ dimension for all M robots. In each iteration, there are totally $2MN_u * (6MN + 6MN_u)$ multiplication, $2MN_u$ integrators, $2 * (6MN + 6MN_u)$ additions/subtractions, and $6MN + 6MN_u$ processes of $K_\Lambda(\cdot)$, hence the computing complexity is $O((MN_u)^2)$ [17]. On the other hand, the traditional sequential quadratic programming (SQP) using gradient descent methods to solve the QP problem. The SQP method requires computation of the Hessian matrix [29] repeatedly with $O((3NM)^4 + (14MNu + 8MN)^3 + (6MN_u + 4MN) * (3MN_u)^2 + 3MN)$ operations; therefore the complexity is $O((MN_u)^4)$, which cannot satisfy our requirement for controlling the formation system in real time. In general, the GPNN have a low computational burden and is an efficient way for solving the QP problem.

Remark 4: Compared with other existing optimal method, in each sampling time, the GPNN can obtain the optimal solutions just by solving the differential equation (40), in which the system constraints are incorporated, and avoid iterative computation. As to other frequently used methods like particle swarm optimization (PSO) [16] and the cooperative coevolutionary algorithm (CCEA) [18], the optimal problems are solved through iteratively updating their global best positions and the constraints need to be considered additionally. However, the complexities of these two methods can up to $O(MN_uN_iN_p)$, where N_i is the maximum iteration number and N_p is the particle number when the global best position cannot directly be found. So, the GPNN is more suitable for the real-time optimization.

Finally, the general control processes of the leader-follower distributed consensus multi-robot formation can be listed as follows:

1) Let $k = 1$, choose the parameters like the control horizon N_u, prediction horizon N, weight matrices R_{ni} and Q_{ni}, the constants γ, α, and period T, desired formation pattern $\mathbf{P} = [(p_{1x}, p_{1y}), (p_{2x}, p_{2y}), \ldots, (p_{Mx}, p_{My})]$, let $k = 1$.
2) For each follower robot $R_i (i = 1, 2, \ldots, M)$, two subsystems (10) and (11) is obtained based on its kinematic model by transformed and dividing processes.
3) Two consensus error systems (19) and (20) are obtained based on the directed graph $\bar{G}$ and the subsystems in 2).
4) For each robot ith $i = 1, 2, \ldots, M$ and $n = 1, 2$, formulate the QP problem (37) and get $H_{ni}, A_{ni}, f_{ni}, r_{ni}$ as well as the upper/lower bounds $l_{ni}, \hbar_{ni}$.
5) Solve the differential equation (40) of GPNN and obtain optimal control increment sequences $\Delta\bar{u}^*_{1i}(k)$ of all the 1th subsystems (15). Only the first terms of $\Delta\bar{u}^*_{1i}(k)$ are used to form the angular velocities $\omega_i(k+1) = u^*_{1i}(k+1) = u^*_{1i}(k) + \Delta u^*_{1i}(k+1)$ for every follower robots.
6) Similar to step 5), solve and obtain the $\Delta\bar{u}^*_{2i}(k)$ of all the 2nd subsystems (16), then linear velocities $v_i(k+1)$ can be obtain by (12).
7) After inputting the velocities to each robots, calculate the posture $X_i = [x_i, y_i, \theta_i]^T$ of all the followers and the $X_L = [x_L, y_L, \theta_L]^T$ of the leader for the calculation of next period.
8) Go back to 2) if the formation moving keeps on.

VI. SIMULATION RESULTS

In this section, simulation results are performed to show the effectiveness of proposed method. The control strategy is applied on a group of mobile robots. In the simulation, the parameters of robot are referred to the practical nonholonomic wheeled mobile robot platform in Section II-B.

Considering the boundaries of the velocities, $v_{max} = 10m/s$, $\omega_{max} = 5rad/s$, $\Delta v_{max} = 2m/s$ and $\Delta\omega_{max} = 1rad/s$ for the robots. The input and input increment limitations of the ith subsystem are chosen as $\bar{u}_{1i_{max}} = [\omega_{max} \cdots \omega_{max}]^T \in R^{N_u}$, $\bar{u}_{1i_{min}} = -\bar{u}_{1i_{max}}$, $\Delta\bar{u}_{1i_{max}} = [\Delta\omega_{max} \cdots \Delta\omega_{max}]^T \in R^{N_u}$ and $\Delta\bar{u}_{1i_{min}} = -\Delta\bar{u}_{1_{max}}$. Refer to (12), v_i is consisted of u_{1i}, u_{2i} and ξ_{2i}, so we set $u_{2i_{max}} = v_{max} - (1+\alpha^2)\omega_{min}\xi_{2i}$, $u_{2i_{min}} = v_{min} - (1+\alpha^2)\omega_{max}\xi_{2i}$. Hence $\bar{u}_{2i_{max}} = [u_{2i_{max}} \cdots u_{2i_{max}}]^T \in R^{N_u}$ and $\bar{u}_{2_{min}} = [u_{2i_{min}} \cdots u_{2i_{min}}]^T \in R^{N_u}$. $\Delta\bar{u}_{2_{max}} = [\Delta v_{max} \cdots \Delta v_{max}]^T \in R^{N_u}$ and $\Delta\bar{u}_{2_{min}} = -\Delta\bar{u}_{2_{max}}$. For the nth consensus subsystem, its bounds of the consensus errors are set as $\bar{e}_{n_{max}} = [10\,10 \cdots 10]^T \in R^{mN}$, $\bar{e}_{n_{min}} = [-10\,-10 \cdots -10]^T \in R^{mN}$, $m = 1$ or 2 depends on the subsystem. The parameters of MPC are set as $N_u = 2$, $N = 3$, $Q_{ni} = 10^4 I$, $R_{ni} = 10I$. $\alpha = 1$.

In the simulation, the robot R_L is set as the leader, 5 follower robots $R_1 - R_5$ are applied. The initial states of robots are set as $X_L(0) = [x_L(0), y_L(0), \theta_L(0)]^T = (3.5\ m, 2.2\ m, 0.0\ rad)^T$, $X_1(0) = [x_1(0), y_1(0), \theta_1(0)]^T = (3.5\ m, 2.5\ m, 0.0\ rad)^T$, $X_2(0) = [x_2(0), y_2(0), \theta_2(0)] = [3.8\ m, 2.2\ m, 0\ rad]^T$, $X_3(0) = [x_3(0), y_3(0), \theta_3(0)]^T = (3.3\ m, 1.7\ m, 0.0\ rad)^T$, $X_4(0) = [x_4(0), y_4(0), \theta_4(0)]^T = (2.4\ m, 1.6\ m, 0.0\ rad)^T$, $X_5(0) = [x_5(0), y_5(0), \theta_5(0)]^T = (2.5\ m, 1.8\ m, 0.0\ rad)^T$.

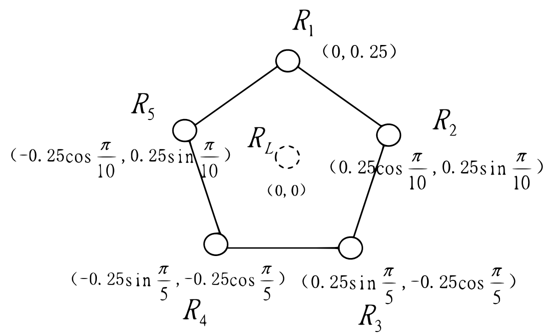


FIGURE 4. The desired geometric pattern of formation.

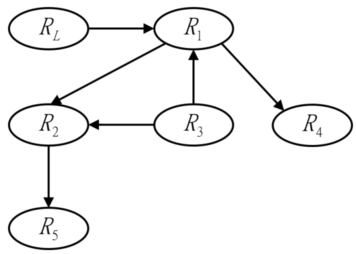


FIGURE 5. Communication topology of the robots.

As shown in Fig 4, the desired geometric pattern $\mathbf{P}$ of formation is defined as

$$\begin{aligned}
(p_{1x}, p_{1y}) &= (0, 0.25)\\
(p_{2x}, p_{2y}) &= (0.25\cos(\pi/10), 0.25\sin(\pi/10))\\
(p_{3x}, p_{3y}) &= (0.25\sin(\pi/5), -0.25\cos(\pi/5))\\
(p_{4x}, p_{4y}) &= (-0.25\sin(\pi/5), -0.25\cos(\pi/5))\\
(p_{5x}, p_{5y}) &= (-0.25\cos(\pi/10), 0.25\sin(\pi/10))
\end{aligned}$$

The moving duration is set as $30.0s$ and the sampling time is set as $T = 0.1s$.

The directed communication topology of these robots is described as Fig. 5. So the adjacency matrix A, degree matrix D and connection weight matrix B are represented as:

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0\\ 1 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 1 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 1 & 0 & 0 & 0 & 0\\ 0 & 2 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$B = diag(1, 0, 0, 0, 0)$$

The velocities of R_L are set as:

$$v_L(k) = 0.4\ m/s, \omega_L(k) = 0.2\ rad/s,\ k = 0, \ldots, 300. \quad (41)$$

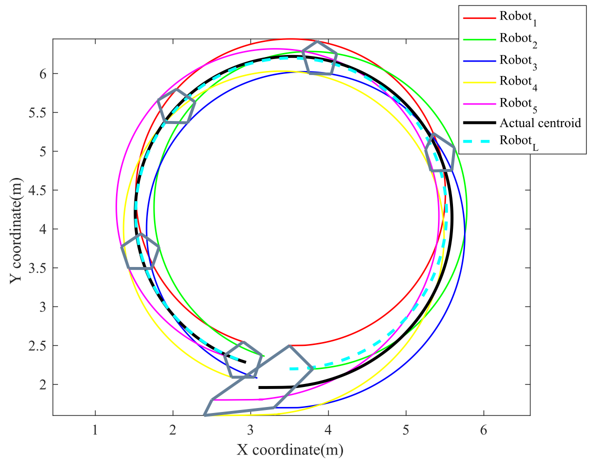


FIGURE 6. The trajectories of the robots in the formation.

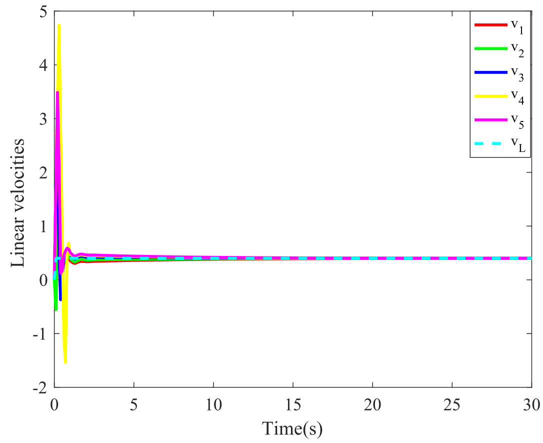


FIGURE 7. The linear velocities of robots.

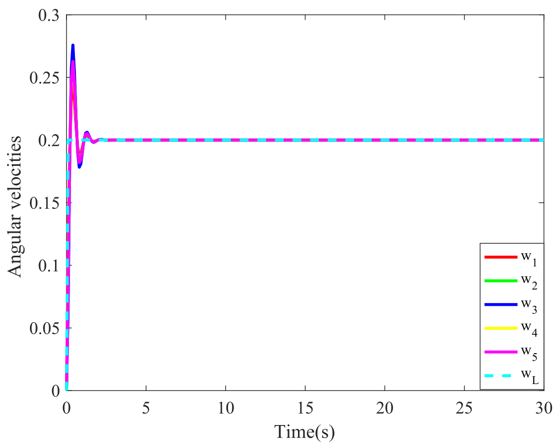


FIGURE 8. The angular velocities of robots.

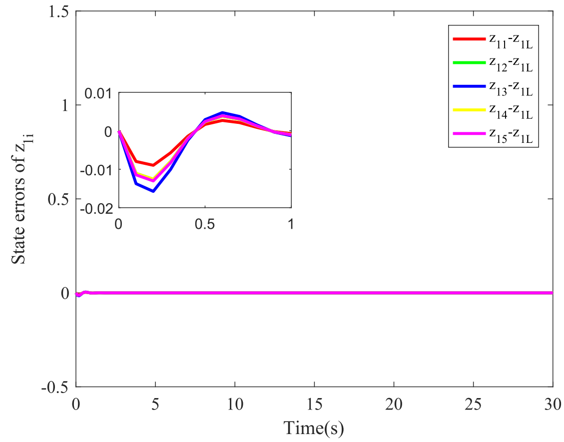


FIGURE 9. The consensus errors of z_{2i} $(i = 1, 2, 3, 4, 5)$.

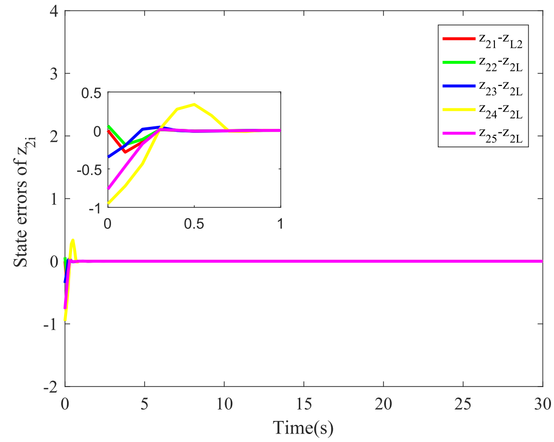


FIGURE 10. The consensus errors of z_{2i} $(i = 1, 2, 3, 4, 5)$.

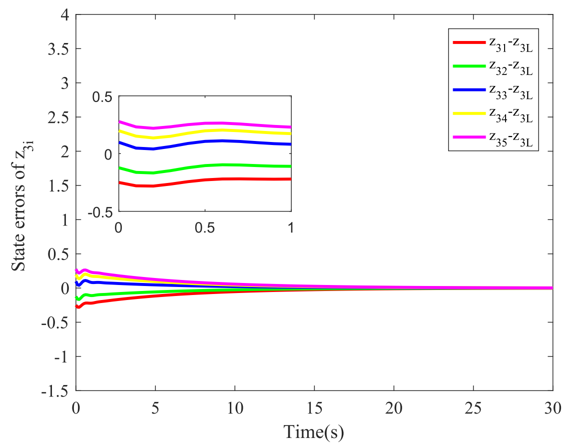


FIGURE 11. The consensus errors of z_{3i} $(i = 1, 2, 3, 4, 5)$.

Fig. 6 shows the trajectories of the whole formation, the alight blue dotted line is the trajectory of the virtual leader robot R_L while others are the trajectories of $R_1 - R_5$. Initially, the five followers formed an irregular pentagon and the formation's centroid (black line) was not on the desired path (R_L's trajectory). Fig. 7 shows the linear velocities while Fig. 8 shows the angular velocities of follower and leader robots. Fig. 9-11 show the changing of the state errors between z_i and z_L and finally the errors converge to the origin. Above results show that the developed method can drive all the followers R_i forming and maintaining a desired pentagon and the centroid of them can track the desired path, the consensus errors of each robots can also be stabilized. Thus the effectiveness of this method can be demonstrated.

VII. CONCLUSION

This paper has developed a NMPC-based distributed leader-follower consensus control strategy for nonholonomic multi-robot formation. For describing the communication topology of these robots, a directed graph is applied. After the transforming, the leader-follower consensus formation system for each nonholonomic robot is obtained and is further divided into two subsystems. A NMPC method is applied to transformed two consensus error systems into constrained QP problems iteratively and the input and state constraints are incorporated into this optimization problem. For solving the QP problem, a GPNN is utilized. The GPNN can obtain the distributed optimal input for each robot with low computational complexity. In the end, simulation results of the proposed method on the multi-robot formation show the effectiveness of the proposed approach. In our future work, we will research the application of the proposed method on some practical cases such as the unmanned wheeled robot convoying. In the convoying process, the leader can be replaced by the protected person and the followers are the automotive escorts. On the other hand, the collision and the communication boundary problems can also be taken into account.

REFERENCES

[1] R. W. Brockett, "Asymptotic stability and feedback stabilization," *Differ. Geometric Control Theory*, vol. 27, no. 1, pp. 181–191, Dec. 1983.

[2] G. Wen, C. L. P. Chen, Y.-J. Liu, and Z. Liu, "Neural network-based adaptive leader-following consensus control for a class of nonlinear multiagent state-delay systems," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2151–2160, Aug. 2017.

[3] C.-E. Ren, L. Chen, and C. L. P. Chen, "Adaptive fuzzy leader-following consensus control for stochastic multiagent systems with heterogeneous nonlinear dynamics," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 1, pp. 181–190, Feb. 2017.

[4] Z. Peng, G. Wen, A. Rahmani, and Y. Yu, "Distributed consensus-based formation control for multiple nonholonomic mobile robots with a specified reference trajectory," *Int. J. Syst. Sci.*, vol. 46, no. 8, pp. 1447–1457, Jun. 2015.

[5] X. Yu and L. Liu, "Leader-follower formation of vehicles with velocity constraints and local coordinate frames," *Sci. China Inf. Sci.*, vol. 60, no. 7, Jul. 2017, Art. no. 070206.

[6] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 947–951, Dec. 2001.

[7] J. P. Desai, J. P. Ostrowski, and V. Kumar, "Modeling and control of formations of nonholonomic mobile robots," *IEEE Trans. Robot. Autom.*, vol. 17, no. 6, pp. 905–908, Dec. 2001.

[8] H. G. Tanner, G. J. Pappas, and V. Kumar, "Leader-to-formation stability," *IEEE Trans. Robot. Autom.*, vol. 20, no. 3, pp. 443–455, Jun. 2004.

[9] K. Do and J. Pan, "Nonlinear formation control of unicycle-type mobile robots," *Robot. Auto. Syst.*, vol. 55, no. 3, pp. 191–204, Mar. 2007.

[10] C. L. P. Chen, D. Yu, and L. Liu, "Automatic leader-follower persistent formation control for autonomous surface vehicles," *IEEE Access*, vol. 7, pp. 12146–12155, 2018.

[11] C.-E. Ren and C. L. P. Chen, "Sliding mode leader-following consensus controllers for second-order non-linear multi-agent systems," *IET Control Theory Appl.*, vol. 9, no. 10, pp. 1544–1552, Jun. 2015.

[12] L. Liu, Y.-J. Liu, and S. Tong, "Fuzzy based multi-error constraint control for switched nonlinear systems and its applications," *IEEE Trans. Fuzzy Syst.*, to be published.

[13] T. Gao, Y.-J. Liu, L. Liu, and D. Li, "Adaptive neural network-based control for a class of nonlinear pure-feedback systems with time-varying full state constraints," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 5, pp. 923–933, Sep. 2018.

[14] D. Li, C. L. P. Chen, Y.-J. Liu, and S. Tong, "Neural network controller design for a class of nonlinear delayed systems with time-varying full-state constraints," *IEEE Trans. neural Netw. Learn. Syst.*, to be published.

[15] H. Xiao, Z. Li, and C. L. P. Chen, "Formation control of leader–follower mobile robots' systems using model predictive control based on neural-dynamic optimization," *IEEE Trans. Ind. Electron.*, vol. 63, no. 9, pp. 5752–5762, Sep. 2016.

[16] H. Xiao and C. L. P. Chen, "Leader-follower multi-robot formation system using model predictive control method based on particle swarm optimization," in *Proc. 32nd Youth Academic Ann. Conf. Chin. Assoc. Autom. (YAC)*, May 2017, pp. 480–484.

[17] H. Xiao and C. L. P. Chen, "Incremental updating multirobot formation using nonlinear model predictive control method with general projection neural network," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4502–4512, Jun. 2019.

[18] S.-M. Lee, H. Kim, H. Myung, and X. Yao, "Cooperative coevolutionary algorithm-based model predictive control guaranteeing stability of multirobot formation," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 37–51, Jan. 2015.

[19] H. Fukushima, K. Kon, and F. Matsuno, "Model predictive formation control using branch-and-bound compatible with collision avoidance problems," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1308–1317, Oct. 2013.

[20] B. Ding, L. Ge, H. Pan, and P. Wang, "Distributed MPC for tracking and formation of homogeneous multi-agent system with time-varying communication topology," *Asian J. Control*, vol. 18, no. 3, pp. 1030–1041, May 2016.

[21] H.-T. Zhang, Z. Cheng, G. Chen, and C. Li, "Model predictive flocking control for second-order multi-agent systems with input constraints," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 62, no. 6, pp. 1599–1606, Jun. 2015.

[22] Q. Liu and J. Wang, "A one-layer projection neural network for nonsmooth optimization subject to linear equalities and bound constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 5, pp. 812–824, May 2013.

[23] Y. Zhang, S. S. Ge, and T. H. Lee, "A unified quadratic-programming-based dynamical system approach to joint torque optimization of physically constrained redundant manipulators," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 34, no. 5, pp. 2126–2132, Oct. 2004.

[24] Z. Li, S. S. Ge, and S. Liu, "Contact-force distribution optimization and control for quadruped robots using both gradient and adaptive neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 8, pp. 1460–1473, Aug. 2014.

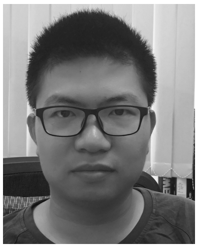
[25] Y. Xia, G. Feng, and J. Wang, "A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations," *Neural Netw.*, vol. 17, no. 7, pp. 1003–1015, Sep. 2004.

[26] Z. Li, Y. Xia, C.-Y. Su, J. Deng, J. Fu, and W. He, "Missile guidance law based on robust model predictive control using neural-network optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 8, pp. 1803–1809, Aug. 2015.

[27] S. Khoo, L. Xie, and Z. Man, "Robust finite-time consensus tracking algorithm for multirobot systems," *IEEE/ASME Trans. Mechatronics*, vol. 14, no. 2, pp. 219–228, Apr. 2009.

[28] Z. Yan and J. Wang, "Model predictive control for tracking of underactuated vessels based on recurrent neural networks," *IEEE J. Ocean. Eng.*, vol. 37, no. 4, pp. 717–726, Oct. 2012.

[29] F.-T. Cheng, R.-J. Sheu, and T.-H. Chen, "The improved compact QP method for resolving manipulator redundancy," *IEEE Trans. Syst., Man, Cybern.*, vol. 25, no. 11, pp. 1521–1530, Nov. 1995.



HANZHEN XIAO received the B.Eng. degree in automation and the M.S. degree in control theory and technology from the College of Automation Science and Engineering, South China University of Technology, Guangzhou, China, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in computer science with the Faculty of Science and Technology, University of Macau.

His research interests include model-predictive control, mobile robots, and neural network control and optimization.

C. L. P. CHEN (S'88–M'88–SM'94–F'07) received the M.S. degree in electrical engineering from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1988.

He is currently a Chair Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. The University of Macau's Engineering and Computer Science programs receiving the Hong Kong Institute of Engineers'(HKIE) Accreditation and Washington/Seoul Accord is his utmost contribution in engineering/computer science education for Macau as the former Dean of the Faculty. His current research interests include systems, cybernetics, and computational intelligence.

Dr. Chen is a Fellow of AAAS, IAPR, CAA, and HKIE. He was the IEEE SMC Society President, from 2012 to 2013, and is currently a Vice President of the Chinese Association of Automation (CAA). He was the Chair of the TC 9.1 Economic and Business Systems of International Federation of Automatic Control (2015–2017) and also a Program Evaluator of the Accreditation Board of Engineering and Technology Education (ABET) of the U.S. for computer engineering, electrical engineering, and software engineering programs. He received the 2016 Outstanding Electrical and Computer Engineers Award from his Alma Mater, Purdue University, after he graduated from the University of Michigan. He is the Editor-in-Chief of the IEEE Transactions on Systems, Man, and Cybernetics: Systems and an Associate Editor for several IEEE Transactions.