Formation Control and Obstacle Avoidance for Multiple Mobile Robots

YANG Tian-Tian[1] LIU Zhi-Yuan[1] CHEN Hong[2] PEI Run[1]

Abstract This paper considers the problem of formation control and obstacle avoidance for a group of nonholonomic mobile robots. On the basis of suboptimal model predictive control, two control algorithms are proposed. Both algorithms are formulated such that they solve as solving the optimal control problems in which the cost functions are coupled with the dynamics of each interacting robot. A potential function is used to define the terminal state penalty term, and a corresponding terminal state region is added to the optimization constraints. Moreover, the main issues inclusive of stability and safety are also discussed. Simulation results show the feasibility of the proposed control strategies.

Key words Nonholonomic mobile robot, formation control, model predictive control, obstacle avoidance

The problem of controlling formations of multiple mobile robots has drawn a considerable amount of research efforts (see [1] for an account of recent results). Many control approaches have been put forward to solving the problem. Leader-follower strategy is the most studied formation control strategy, which uses a hierarchical arrangement of individual controllers. So the problem of formation control is reduced to individual tracking problems[2−4]. In [5−6], a behavior-based approach was presented. The basic behaviors were assigned to the independent systems to form a guidance algorithm. Then the controllers for achieving different objectives were combined. The concept of virtual structure was first introduced in [7]. In the virtual structure architecture, control methods were developed to force a group of robots to behave in a rigid formation[8]. Similar to the leader-follower controller, a cyclic architecture was formed by connecting individual robot controllers[9]. The difference is that the cyclic controller connections are not hierarchical.

In [10], model predictive control (MPC) was applied to cooperative control of multiple robots. MPC is a feedback control scheme, in which an optimization problem is solved at each sampling time. Centralized MPC has been widely developed for constrained systems, with many results related to stability and robustness[11]. However, solving a single optimization problem for the entire team typically requires significant computation, which scales with the size of the system (e.g., the number of robots in the team). In [12], it was proved that under mild conditions feasibility rather than optimality is sufficient for stability, then the computational requirement is reduced to finding a control profile that satisfies a set of constraints.

Recently, focus is on distributed MPC because of issues related to computation and communication in a large-scale application[13]. When decisions are made in a distributed fashion, although the requirement for stability and performance criteria can be met for individual subsystem, the coupled system may show serious instabilities. So the key point is that the actions of each subsystem must be consistent with those of other subsystems, so that decisions taken independently can guarantee the coupled system stability.

Obstacle avoidance is another well-studied problem in mobile robotics. One of the obstacle avoidance algorithm is potential field method, where an artificial potential function depending on the obstacles is constructed[14−15]. In [15], an artificial potential function called navigation function was proposed when the obstacles were represented by star shaped sets. This potential function has no local minima except the global minimum at the goal. Furthermore, it is continuously differentiable and attains its maximum value at all the obstacle boundaries.

In this paper, we propose two suboptimal model predictive control methods for formation control and obstacle avoidance of multiple mobile robots on the basis of the problem formulation and results in [13]. First, we present an integrated optimal control problem by introducing the potential function as the terminal state penalty term. A centralized suboptimal model predictive control method that achieves the control objective is stated, with stability and safety guaranteed. Then by decomposing the integrated optimal control problem into a family of simple optimal control problems, a distributed suboptimal model predictive control method is developed. Under this scheme, on-line optimization of the whole multi-robot system is decomposed into the optimization of several cooperative robots. These robots can cooperate and communicate with each other to achieve the control objective. Thus, the computational complexity can be significantly reduced.

The rest of the paper is organized as follows. In Section 1, the model to describe the dynamics of the mobile robot is established and the goal of the controller design is stated. In Section 2, two suboptimal control algorithms are presented. Moreover, stability and safety of the closed-loop system are discussed. Some simulation results are presented to validate the theoretical analysis in Section 3.

1 Dynamic model and problem formulation

In a group of N mobile robots, the dynamics of the i-th robot is given as follows[3].

$$\dot{\varsigma}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \\ \dot{v}_i \\ \dot{\omega}_i \end{bmatrix} = \begin{bmatrix} v_i\cos\theta_i \\ v_i\sin\theta_i \\ \omega_i \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{1}{m_i} & 0 \\ 0 & \dfrac{1}{I_i} \end{bmatrix} \begin{bmatrix} F_i \\ N_i \end{bmatrix} \tag{1}$$

$$i = 1, \cdots, N$$

where (x_i, y_i) is the inertial position, θ_i is the orientation, v_i is the linear speed, ω_i is the angular speed, F_i is the applied force, N_i is the applied torque, m_i is the mass, and I_i is the moment of inertia.

Because of violation of the Brockett′s condition, system (1) cannot be asymptotically stabilized using continuous static state feedback[16]. By using the results in [17], the dynamics of the mobile robot can be converted to the canonical form. With the state transformation

$$x_{1i} = x_i, x_{2i} = y_i, x_{3i} = \tan\theta_i, x_{4i} = v_i\cos\theta_i, x_{5i} = \frac{\omega_i}{\cos^2\theta_i}$$

and the input transformation

$$u_{1i} = \frac{F_i}{m_i}\cos\theta_i - v_i\omega_i\sin\theta_i, u_{2i} = \frac{N_i}{I_i\cos^2\theta_i} + \frac{2\sin\theta_i}{\cos^3\theta_i}\omega_i^2$$

Received January 11, 2007; in revised form November 10, 2007
Supported by National Natural Science Foundation of China (60234030)
1. Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, P. R. China 2. Department of Control Science and Engineering, Jilin University, Changchun 130025, P. R. China
DOI: 10.3724/SP.J.1004.2008.00588

(1) is transformed into

$$\dot{\boldsymbol{\xi}}_i = \begin{bmatrix} \dot{x}_{1i} \\ \dot{x}_{2i} \\ \dot{x}_{3i} \\ \dot{x}_{4i} \\ \dot{x}_{5i} \end{bmatrix} = \begin{bmatrix} x_{4i} \\ x_{3i}x_{4i} \\ x_{5i} \\ u_{1i} \\ u_{2i} \end{bmatrix} = \boldsymbol{f}_i(\boldsymbol{\xi}_i, \boldsymbol{u}_i), \quad i = 1, \cdots, N \tag{2}$$

Obviously, states $\boldsymbol{q}_i = [x_{1i}, x_{2i}]^{\rm T}$ are the position of the i-th robot in the X-Y plane. States $\boldsymbol{p}_i = [x_{3i}, x_{4i}, x_{5i}]^{\rm T}$ are generalized angle, linear speed, and angular speed, respectively. $\boldsymbol{u}_i = [u_{1i}, u_{2i}]^{\rm T}$ are generalized inputs.

By concatenating the states and inputs into the vectors as $\boldsymbol{\xi}(t) = [\boldsymbol{\xi}_1^{\rm T}(t), \cdots, \boldsymbol{\xi}_N^{\rm T}(t)]^{\rm T}$, $\boldsymbol{u}(t) = [\boldsymbol{u}_1^{\rm T}(t), \cdots, \boldsymbol{u}_N^{\rm T}(t)]^{\rm T}$, the multirobot system can be described as follows

$$\dot{\boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{\xi}_1, \boldsymbol{u}_1) \\ \vdots \\ \boldsymbol{f}_N(\boldsymbol{\xi}_N, \boldsymbol{u}_N) \end{bmatrix} = \boldsymbol{F}(\boldsymbol{\xi}, \boldsymbol{u}) \tag{3}$$

In this paper we consider the problem of moving a group of robots towards a final destination with obstacle avoidance. There are two objectives. The first objective is to move the robots to their final destinations while maintaining formation as specified. The second objective is to avoid collision with the obstacles.

To incorporate these competing objectives, the object functions will be defined respectively. Let E_g be the center error, $\boldsymbol{q}_d$ be the desired destination of the core robots that are used to relate to the desired center of geometry location. Without loss of generality, the core robots are robot 1, robot 2, and robot 3. Then we have

$$E_g = \boldsymbol{g}^{\rm T} Q_g \boldsymbol{g} \tag{4}$$

where Q_g is a symmetric positive definite matrix, and $\boldsymbol{g} = 1/3(\boldsymbol{q}_1 + \boldsymbol{q}_2 + \boldsymbol{q}_3) - \boldsymbol{q}_d$. Similarly, define E_f as the formation error. Introduce a map $R : i \in \{1, \cdots, N\} \rightarrow R(i) \subset \{1, \cdots, N\}$, where $R(i)$ represents the set of robots that communicate with robot i.

$$E_f = 1/2 \sum_{i=1}^{N} \sum_{j \in R(i)} \boldsymbol{f}_{ij}^{\rm T} Q_{fi} \boldsymbol{f}_{ij} \tag{5}$$

where Q_{fi} is a symmetric positive definite matrix. $\boldsymbol{f}_{ij} = \boldsymbol{q}_i - \boldsymbol{q}_j - \boldsymbol{d}_{ij}$, and $\boldsymbol{d}_{ij}$ is the desired relative vector between any two neighbors i and j.

Then the total objective for formation control is

$$E_t = E_g + E_f + E_p \tag{6}$$

where $E_p = \sum_{i=1}^{N} (\boldsymbol{p}_i - \boldsymbol{p}_i^c)^{\rm T} Q_{pi} (\boldsymbol{p}_i - \boldsymbol{p}_i^c)$, $\boldsymbol{p}_i^c$ is the desired state, and Q_{pi} is a symmetric positive definite matrix.

Obviously $E_t = 0$ is equivalent to $E_f = 0$, $E_g = 0$, and $E_p = 0$. Note that $E_f = 0$, if and only if $\boldsymbol{f}_{ij} = 0$ for all i. This means that $\boldsymbol{q}_i - \boldsymbol{q}_j = \boldsymbol{d}_{ij}$, which will only be true if the robots are in the desired formation. When $E_g = 0$, we have $1/3(\boldsymbol{q}_1 + \boldsymbol{q}_2 + \boldsymbol{q}_3) = \boldsymbol{q}_d$, which means that the center of the robots arrives at the desired position. Therefore, when $E_t = 0$, the group of robots achieves the desired destination, i.e., $\boldsymbol{q}_i = \boldsymbol{q}_i^c$. Here, $\boldsymbol{q}_i^c$ is the desired position for the i-th robot.

Another main challenge in formation control for multiple mobile robots is obstacle avoidance. In this paper, it is assumed that there are M walls or isolated obstacles, such as a pillar on the ground and each obstacle $\mathrm{O}_k (k = 1, \cdots, M)$ can be expressed by a real-valued obstacle function $O_k (k = 1, \cdots, M)$ as follows[15]

$$O_k = \{\boldsymbol{r} \in \mathbf{R}^2 | O_k(\boldsymbol{r}) \le 0\} \tag{7}$$

where $\boldsymbol{r} \in \mathbf{R}^2$ is the position vector in $\mathbf{R}^2$. The obstacle function O_k is zero on the boundary of the obstacle, positive outside of the obstacle, and negative inside of the obstacle. When the mobile robot moves to the desired point avoiding collision with the obstacles, the state variable $\boldsymbol{r}$ must satisfy $O_k(\boldsymbol{r}) > 0$. In this paper, we introduce the potential function $NF(\cdot)$, which is a continuous approximation of the length of the shortest obstacle-free path to the goal[15].

Then, the problem considered in the next section is to drive the formation control objective E_t asymptotically to zero while avoiding collision with the obstacles.

2 Control algorithms

In this section, two suboptimal MPC algorithms are proposed for solving the problem described in Section 1. First, a centralized optimal problem is constructed.

Problem 1. At each update time t, given the current states $\boldsymbol{\xi}(t)$, solve the following optimal problem

$$J^*(\boldsymbol{\xi}(t), \boldsymbol{u}^*(\cdot), T) = \min_{u(\cdot)} J(\boldsymbol{\xi}(t), \boldsymbol{u}(\cdot), T) \tag{8}$$

with

$$J(\boldsymbol{\xi}(t), \boldsymbol{u}(\cdot), T) = \sum_{i=1}^{N} J_i(\boldsymbol{\xi}_i(t), \boldsymbol{\xi}_j(t), \boldsymbol{u}_i(\cdot), T) \tag{9}$$

$$J_i(\boldsymbol{\xi}_i(t), \boldsymbol{\xi}_j(t), \boldsymbol{u}_i(\cdot), T) = \int_t^{t+T} L_i(\boldsymbol{\xi}_i(\tau, \boldsymbol{\xi}_i(t)), \boldsymbol{\xi}_j(\tau, \boldsymbol{\xi}_j(t)), \boldsymbol{u}_i(\tau)) {\rm d}\tau + g_i(\boldsymbol{\xi}_i(t+T)) \tag{10}$$

$$L_i(\boldsymbol{\xi}_i, \boldsymbol{\xi}_j, \boldsymbol{u}_i) = \frac{1}{2} \sum_{j \in R(i)} \boldsymbol{f}_{ij}^{\rm T} Q_{fi} \boldsymbol{f}_{ij} + \boldsymbol{p}_i^{\rm T} Q_{pi} \boldsymbol{p}_i + \boldsymbol{u}_i^{\rm T} R_i \boldsymbol{u}_i + T_c \tag{11}$$

$$T_c = \begin{cases} \frac{1}{3} E_g & i = 1,\ 2,\ 3 \\ 0 & \text{otherwise} \end{cases}$$

$$g_i(\boldsymbol{\xi}_i) = NF(\boldsymbol{q}_i - \boldsymbol{q}_i^c) + \boldsymbol{p}_i^{\rm T} H_i \boldsymbol{p}_i$$

subject to

$$\begin{cases} \dot{\boldsymbol{\xi}}_i(\tau) = \boldsymbol{f}(\boldsymbol{\xi}_i(\tau), \boldsymbol{u}_i(\tau)) \\ \boldsymbol{u}_i(\tau) \in U \\ O_k(\boldsymbol{q}_i(\tau)) > 0 \end{cases} \quad \tau \in [t, t+T] \tag{12}$$

$$\boldsymbol{\xi}_i(t+T) \in \Omega_i \tag{13}$$

$$i = 1, \cdots, N,\ k = 1, \cdots, M$$

where R_i and H_i are symmetric positive definite matrices, and T is the open-loop horizon time.

It is obvious that $J(\boldsymbol{\xi}(t), \boldsymbol{u}(\cdot), T) \ge 0$, and $J(\boldsymbol{\xi}(t), \boldsymbol{u}(\cdot), T) = 0$ if and only if $\boldsymbol{\xi}(t) = \boldsymbol{\xi}^c$, with $\boldsymbol{\xi}^c = [\boldsymbol{\xi}_1^{c\rm T}, \cdots, \boldsymbol{\xi}_N^{c\rm T}]^{\rm T}$, $\boldsymbol{\xi}_i^c = [\boldsymbol{q}_i^{c\rm T}, \boldsymbol{p}_i^{c\rm T}]^{\rm T}$.

(13) is called terminal constraint, as it is a constraint enforced only at the terminal or end time. Here, a potential function $NF(\cdot)$ is developed as the terminal penalty term, and a corresponding terminal state controller is designed to guarantee the stability.

The solutions of Problem 1 yield the optimal control law applied to the system

$$\boldsymbol{u}^*(t) = [\boldsymbol{u}_1^*(t)^{\rm T}, \cdots, \boldsymbol{u}_N^*(t)^{\rm T}]^{\rm T} \tag{14}$$

Theorem 1. (**Stability**) Suppose Problem 1 is feasible at time $t = 0$. With the MPC control algorithm (14), system (3) is asymptotically stable at the point $\boldsymbol{\xi}^c$ if there is a terminal state controller $\boldsymbol{u}_L(t) = [\boldsymbol{u}_{1L}^{\mathrm{T}}(t), \cdots, \boldsymbol{u}_{NL}^{\mathrm{T}}(t)]^{\mathrm{T}}$, such that the following conditions are satisfied

$$\dot{g}_i(\boldsymbol{\xi}_i(t)) + L_i(\boldsymbol{\xi}_i(t), \boldsymbol{\xi}_j(t), \boldsymbol{u}_{iL}(t)) \le 0 \quad i = 1, \cdots, N \tag{15}$$

for any state $\boldsymbol{\xi}_i(t)$ belonging to the terminal region Ω_i.

Proof. The proof is similar to stability theorem in [18] and hence omitted. □

Theorem 2. (**Safety**) If the control algorithm (14) is used and the robots start at rest in an unoccupied position, then, the robots will not run into obstacles.

Proof. At each update time t, the control profile can be split into two parts: $\boldsymbol{u}_i(\tau),\ t \le \tau \le t + T$, and $\boldsymbol{u}_i(\tau),\ \tau > t + T$. Because $\boldsymbol{u}_i(\tau),\ t \le \tau \le t + T$ are free decision variables, constraints on $\boldsymbol{\xi}_i(\tau, \boldsymbol{\xi}_i(t)),\ t \le \tau \le t + T$ can be imposed directly. From the definition of obstacle function, the robot i can avoid the obstacles if $O_k(\boldsymbol{q}_i(\tau)) > 0$ with $k = 1, \cdots, M$. For future motion in the terminal region Ω_i, the proof of safety is obtained from [15]. □

In Problem 1, the control profile is found by minimizing the objective globally. At each sampling time t, only the first control is applied to the system. In the next time, the controller needs to solve the constrained nonlinear optimization problem again. Because the online computational burden related to MPC of large nonlinear system hampers its real-time application, another method is proposed to reduce computation time. It was presented in [12] that the proved stability does not necessarily depend on the optimality of the cost function. So a suboptimal MPC control algorithm is built up by using a suboptimal solution or feasible solution to the system. Next, two suboptimal MPC algorithms are presented by using the result in [12].

Algorithm 1. (Centralized suboptimal MPC)

Step 1. At time $t = 0$, get the initial state $\boldsymbol{\xi}_i(0)$ with $i = 1, \cdots, N$. An open loop optimal control $\boldsymbol{u}^*(0) = [\boldsymbol{u}_1^*(0)^{\mathrm{T}}, \boldsymbol{u}_2^*(0)^{\mathrm{T}}, \cdots, \boldsymbol{u}_N^*(0)^{\mathrm{T}}]^{\mathrm{T}}$ can be found by solving the optimization problem (8) subject to (12), (13), and (15), with terminal state controller $\boldsymbol{u}_{iL}(t) = K_i\boldsymbol{\xi}_i(t)$. Subsequently, apply $\boldsymbol{u}^*(0)$ to the system.

Step 2. At time t, choose control sequences that are constructed on the basis of $\boldsymbol{u}_i^*(\tau, \boldsymbol{\xi}_i(t-\delta))$ and the terminal state controller

$$\hat{\boldsymbol{u}}_i(\tau, \boldsymbol{\xi}_i(t)) = \begin{cases} \boldsymbol{u}_i^*(\tau, \boldsymbol{\xi}_i(t-\delta)) & t \le \tau \le t + T - \delta \\ \boldsymbol{u}_{iL}(\tau, \boldsymbol{\xi}_i(t-\delta)) & t + T - \delta \le \tau \le t + T \end{cases} \tag{16}$$

Then, $\hat{\boldsymbol{u}}(t) = [\hat{\boldsymbol{u}}_1(t)^{\mathrm{T}}, \cdots, \hat{\boldsymbol{u}}_N(t)^{\mathrm{T}}]^{\mathrm{T}}$ which is applied to the system.

Step 3. Set $t = t + \delta$, and return to Step 2 at the next sampling time.

Step 4. At time $t + T_0$, $T_0 \le T$, find a new control sequence as is done at time $t = 0$.

Given the current state $\boldsymbol{\xi}_i(t)$ and a suboptimal control $\hat{\boldsymbol{u}}_i(\tau, \boldsymbol{\xi}_i(t))$ with $\tau \in [t, t+T]$, the suboptimal state $\hat{\boldsymbol{\xi}}_i(\tau)$ is computed using the dynamic model (2). Because of the use of the constructed control (16), the feasible solution to the optimization problem always exists as long as the initial feasible solution exists. Now, it will be shown that our suboptimal MPC control algorithm is also asymptotically stable by examining the three conditions of Theorem 1 in [12].

Theorem 3. By using Algorithm 1, system (3) is asymptotically stable at point $\boldsymbol{\xi}^c$. Moreover, the robots will not run into obstacles.

Proof. There are two cases: first, $\boldsymbol{u}(t)$ is an optimal solution, and $\boldsymbol{u}(t+\delta)$ is a suboptimal solution; and second, $\boldsymbol{u}(t)$ and $\boldsymbol{u}(t+\delta)$ are both suboptimal solutions.

If $\boldsymbol{u}(t)$ is an optimal solution and $\boldsymbol{u}(t+\delta)$ is a suboptimal solution, the stability is directly from Theorem 1.

Then, if $\boldsymbol{u}(t)$ and $\boldsymbol{u}(t+\delta)$ are both suboptimal solutions, the stability will be shown by examining the three conditions of Theorem 1 in [12].

Let $\hat{\boldsymbol{\xi}}(t) = [\hat{\boldsymbol{\xi}}_1(t)^{\mathrm{T}}, \cdots, \hat{\boldsymbol{\xi}}_N(t)^{\mathrm{T}}]^{\mathrm{T}}$ denote the suboptimal state trajectory and $\hat{J}(\hat{\boldsymbol{\xi}}(t))$ denote the suboptimal model predictive value function. Then, we have

$$\hat{J}(\hat{\boldsymbol{\xi}}(t)) = \sum_{i=1}^{N} \int_t^{t+T} L_i(\hat{\boldsymbol{\xi}}_i(\tau, \hat{\boldsymbol{\xi}}_i(t)), \hat{\boldsymbol{\xi}}_j(\tau, \hat{\boldsymbol{\xi}}_j(t)), \hat{\boldsymbol{u}}_i(\tau))\mathrm{d}\tau + \sum_{i=1}^{N} g_i(\hat{\boldsymbol{\xi}}_i(t+T))$$

From [13], we have

$$\sum_{i=1}^{N} L_i(\hat{\boldsymbol{\xi}}_i(t), \hat{\boldsymbol{\xi}}_j(t), \hat{\boldsymbol{u}}_i(t)) = ||\hat{\boldsymbol{\xi}}(t) - \boldsymbol{\xi}^c||_Q^2 + ||\hat{\boldsymbol{u}}||_R^2 \tag{17}$$

where $\boldsymbol{\xi}^c = [\boldsymbol{\xi}_1^{c\,\mathrm{T}}, \cdots, \boldsymbol{\xi}_N^{c\,\mathrm{T}}]^{\mathrm{T}}$, and Q is a positive definite and symmetric matrix.

Obviously, we have

$$\hat{J}(t, \hat{\boldsymbol{\xi}}(t)) \ge ||\hat{\boldsymbol{\xi}}(t) - \boldsymbol{\xi}^c||_Q^2 \tag{18}$$

This proves the condition 1 of Theorem 1 in [12].

Because of the stability condition (15), we can have the following result at two sampling times t and $t + \delta$

$$\hat{J}(t+\delta, \hat{\boldsymbol{\xi}}(t+\delta)) - \hat{J}(t, \hat{\boldsymbol{\xi}}(t)) \le -\int_t^{t+\delta} L(\hat{\boldsymbol{\xi}}(\tau), \hat{\boldsymbol{u}}(\tau))\mathrm{d}\tau$$

where $L(\hat{\boldsymbol{\xi}}(t), \hat{\boldsymbol{u}}(t)) = \sum_{i=1}^{N} L_i(\hat{\boldsymbol{\xi}}_i(t), \hat{\boldsymbol{\xi}}_j(t), \hat{\boldsymbol{u}}_i(t))$. This proves the condition 2 of Theorem 1 in [12].

Finally, by selecting a terminal state controller $\boldsymbol{u}_{iL}(t) = K_i\boldsymbol{\xi}_i(t)$, the condition 3 of Theorem 1 in [12] is satisfied.

The proof of safety is similar to the proof of Theorem 2. □

Note that Algorithm 1 is a centralized fashion, which will bring large online computational burden. A distributed solution to the problem is desirable for potential scalability and improved tractability of the approach. So we will define N separate optimal problems, which are solved and implemented in a distributed fashion. The main idea is to break the centralized MPC controller into a number of distinct MPC controllers of smaller sizes. For each robot, the current state and model of its neighbors are used to predict their possible trajectories.

Problem 2. For every robot $i = 1, \cdots, N$ and at each update time t, given the current state $\boldsymbol{\xi}_i(t)$, $\hat{\boldsymbol{\xi}}_j(t) = \boldsymbol{\xi}_j(t)$, solve the following optimal problem

$$J_i^*(\boldsymbol{\xi}_i(t), \hat{\boldsymbol{\xi}}_j(t), \boldsymbol{u}_i^*(\cdot), T) = \min_{u_i(\cdot)} J_i(\boldsymbol{\xi}_i(t), \hat{\boldsymbol{\xi}}_j(t), \boldsymbol{u}_i(\cdot), T) \tag{19}$$

subject to

$$\begin{cases} \dot{\boldsymbol{\xi}}_i(\tau) = \boldsymbol{f}(\boldsymbol{\xi}_i(\tau), \boldsymbol{u}_i(\tau)) & \\ \dot{\hat{\boldsymbol{\xi}}}_j(\tau) = \boldsymbol{f}(\hat{\boldsymbol{\xi}}_j(\tau), \hat{\boldsymbol{u}}_j(\tau)) & \tau \in [t, t+T] \\ \boldsymbol{u}_i(\tau) \in U & \\ ||\boldsymbol{u}_i(\tau) - \hat{\boldsymbol{u}}_i(\tau)|| \le \delta^2\gamma & \\ O_k(\boldsymbol{q}_i(\tau)) > 0, \quad k = 1, \cdots, M & \end{cases} \tag{20}$$

$$\boldsymbol{\xi}_i(t+T) \in \Omega_i \tag{21}$$

where δ is the model predictive update time, and γ is a positive constant.

The difference between Problem 1 and Problem 2 is that in the latter problem $\hat{\boldsymbol{\xi}}_j$ is a suboptimal solution and correspondingly a control comparison constraint is added.

Algorithm 2. (Distributed suboptimal MPC)

Step 1. At time $t = 0$, get the initial states $\boldsymbol{\xi}_i(0)$ and $\boldsymbol{\xi}_j(0)$, and set $\hat{\boldsymbol{u}}_i(\tau, \boldsymbol{\xi}_i(0)) = 0$ for all $\tau \in [0, T]$. For each robot i, solve the optimization problem (19) subject to (20), (21), and (15), with terminal state controller $\boldsymbol{u}_{iL}(t) = K_i\boldsymbol{\xi}_i(t)$. Then, $\boldsymbol{u}^*(0) = [\boldsymbol{u}_1^*(0)^{\rm T}, \cdots, \boldsymbol{u}_N^*(0)^{\rm T}]^{\rm T}$ which is applied to the system.

Step 2. At time t, for each robot i, choose a control sequence

$$\hat{\boldsymbol{u}}_i(\tau, \boldsymbol{\xi}_i(t)) = \begin{cases} \boldsymbol{u}_i^*(\tau, \boldsymbol{\xi}_i(t-\delta)) & t \leq \tau \leq t+T-\delta \\ \boldsymbol{u}_{iL}(\tau, \boldsymbol{\xi}_i(t-\delta)) & t+T-\delta \leq \tau \leq t+T \end{cases} \tag{22}$$

Then, $\hat{\boldsymbol{u}}(t) = [\hat{\boldsymbol{u}}_1(t)^{\rm T}, \cdots, \hat{\boldsymbol{u}}_N(t)^{\rm T}]^{\rm T}$ which is applied to the system.

Step 3. Set $t = t + \delta$, and return to Step 2 at the next sampling time.

Step 4. At time $t = t + T_0$ where $T_0 \leq T$, for each robot i, get the current state $\boldsymbol{\xi}_i(t)$ and obtain $\boldsymbol{\xi}_j(t)$, $\hat{\boldsymbol{u}}_j(\tau, \boldsymbol{\xi}_j(t))$ for all $j \in N_i$. Solve the optimization problem (19) subject to (20), (21), and (15), with terminal state controller $\boldsymbol{u}_{iL}(t) = K_i\boldsymbol{\xi}_i(t)$. Then, $\boldsymbol{u}^*(t) = [\boldsymbol{u}_1^*(t)^{\rm T}, \cdots, \boldsymbol{u}_N^*(t)^{\rm T}]^{\rm T}$ which is applied to the system.

Step 5. Set $t = t + \delta$, and return to Step 2 at the next sampling time.

Theorem 4. By application of Algorithm 2 with sufficiently fast update (small δ), the robots will achieve the desired formation asymptotically without colliding with obstacles.

Proof. The proof is quickly obtained from Theorem 3 and the result in [13]. □

In [13], an upper bound on the update time was presented, and the limit for δ was conservative. In practice, values for δ much larger than those specified by the theory are successful. The next section reveals that for a fixed small value δ, convergence can still be obtained with good accuracy.

3 Simulation results

This section presents the results of numerical simulation to show the application of the control algorithms described in Section 2. The objective is to drive five robots to track the reference trajectory in the desired formation without colliding with the obstacle. Initially, the robots have detected the obstacle. After initialization they avoid the obstacle and then track the reference trajectory with formation maintained. Both of the two control algorithms are implemented, with comparison to each other.

It is assumed that there is an obstacle O_1

$$O_1 = \left[\boldsymbol{r} - \left[\begin{array}{c} 6 \\ -1 \end{array}\right]\right]^{\rm T} \left[\boldsymbol{r} - \left[\begin{array}{c} 6 \\ -1 \end{array}\right]\right] - 1$$

As discussed in [15], the navigation function depending on the obstacle O_1 can be designed as follows.

$$NF(\boldsymbol{r}) = \frac{||\boldsymbol{r}||^2}{\left(\parallel \boldsymbol{r} \parallel^2 + \left\|\left[\boldsymbol{r} - \left[\begin{array}{c} 6 \\ -1 \end{array}\right]\right]\right\|^2 - 1\right)^{0.5}}$$

The same initial conditions are used for the two algorithms: $\boldsymbol{\varsigma}_1 = [4,\ 1,\ 0.78,\ 0,\ 0]^{\rm T}$, $\boldsymbol{\varsigma}_2 = [2, 2,\ 0.78,\ 0,\ 0]^{\rm T}$, $\boldsymbol{\varsigma}_3 = [2,\ 0,\ 0.78,\ 0,\ 0]^{\rm T}$, $\boldsymbol{\varsigma}_4 = [0,\ 3,\ 0.78,\ 0,\ 0]^{\rm T}$, and $\boldsymbol{\varsigma}_5 = [0,\ -1,\ 0.78,\ 0,\ 0]^{\rm T}$. The relative vectors are defined as $\boldsymbol{d}_{12} = [2, -1]^{\rm T}$, $\boldsymbol{d}_{13} = [2, 1]^{\rm T}$, $\boldsymbol{d}_{24} = [2, -1]^{\rm T}$, and $\boldsymbol{d}_{35} = [2, 1]^{\rm T}$. The desired reference trajectory for the core robots is $\boldsymbol{q}_d = [0.5t + 10, 0]^{\rm T}$ with $t \geq 0$. For clarity, we mark the initial formation with "∘" marks, and the final formation with "△" marks. The following parameters are used: $\delta = 0.5\,$s, $T = 3\,$s, $T_0 = 1.5\,$s, $Q_g = I, Q_{fi} = 2I, Q_{pi} = I$, $R_i = I, H_i = I$, and $K_i = \left[\begin{array}{ccccc} 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0.8 \end{array}\right]$ with $i = 1, \cdots, 5$.

For the distributed suboptimal control algorithm, the control comparison constraint is enforced by setting $\gamma = 5$. The position trajectories of the five robots are shown in Fig. 1. It is obvious that after initialization the robots successfully avoid the obstacle in the environment and then, achieve tracking the reference trajectory. Moreover, the formation is almost maintained throughout the motion, which can be confirmed from Fig. 2.

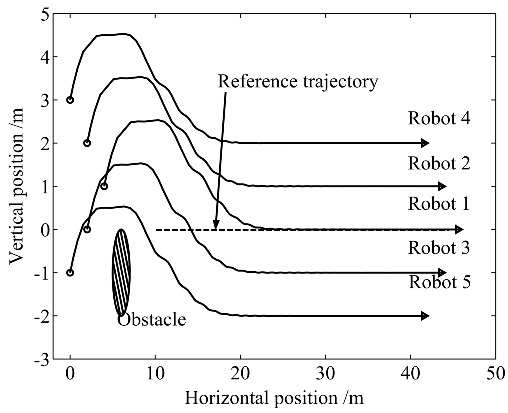

Fig. 1 Position trajectories of five robots using distributed suboptimal algorithm

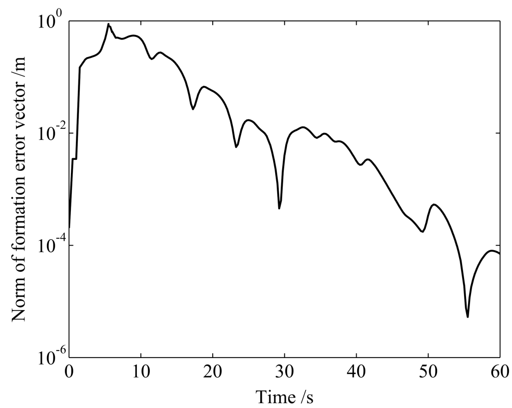

Fig. 2 2-norm of formation error vector using distributed suboptimal algorithm

For the update time δ, the centralized suboptimal control algorithm is implemented, and the simulation result is shown in Fig. 3. From the figure, the position trajectories are close to those of the distributed implementation, and the formation is observed to meet the objective.

Figs. 2 and 4 show the norms of the formation error vectors using the two control algorithms. We can observe both the algorithms and keep the 2-norm of the formation error vectors almost zero. Moreover, under the distributed control algorithm, the formation meets the objective to almost the same precision as in the centralized fashion.

The simulations for the two algorithms are run in Matlab on a single Windows XP computer with a Celeron 2.4 GHz processor. The mean computation time for the distributed algorithm is 14.7 min and is 96.8 min for the centralized algorithm. It is clear that the distributed method offers a considerable computational advantage, averaging almost seven times faster than the centralized method. The larger problems, e.g., with six or more robots, that are not attempted use the centralized method as the computation time becomes prohibitively long.

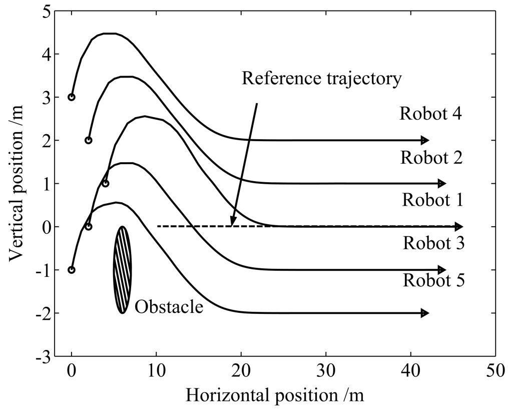

Fig. 3 Position trajectories of the five robots using centralized suboptimal algorithm

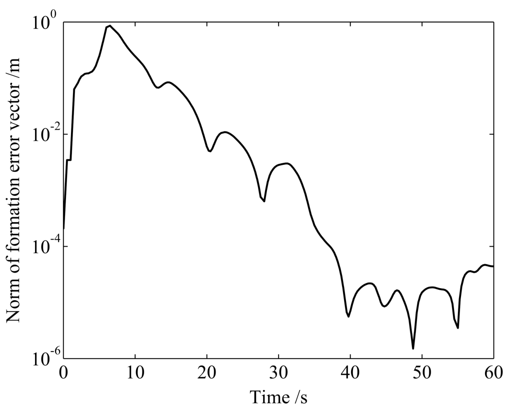

Fig. 4 2-norm of formation error vector using centralized suboptimal algorithm

4 Conclusion

This paper presents the methods for formation control and obstacle avoidance of multiple mobile robots. Two suboptimal MPC algorithms are proposed. Both algorithms can guide a group of robots in a maintained formation to the specified destination while avoiding collision with obstacles. Because the distributed algorithm offers a considerable computational advantage, it is more suitable for larger problems, e.g., with five or more robots. The feasibility of the algorithms is also verified by simulations.

References

1 Chen Y Q, Wang Z M. Formation control: a review and a new consideration. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems. Edmonton, Canada: IEEE, 2005. 3181−3186

2 Desai J P, Ostrowski J P, Kumar V. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 2001, **17**(6): 905−908

3 Chen X P, Serrani A, Ozbay H. Control of leader-follower formations of terrestrial UAVs. In: Proceedings of the 42nd IEEE Conference on Decision and Control. Maui, USA: IEEE, 2003. 498−503

4 Fujimori A, Fujimoto T, Bohacs G. Distributed leader-follower navigation of mobile robots. In: Proceedings of International Conference on Control and Automation. Budapest, Hungary: IEEE, 2005. 960−965

5 Balch T, Arkin R C. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, 1998, **14**(6): 926−939

6 Cao Z Q, Xie L J, Zhang B, Wang S, Tan M. Formation constrained multi-robot system in unknown environments. In: Proceedings of IEEE International Conference on Robotics and Automation. Taipei, China: IEEE, 2003. 735−740

7 Tan K H, Lewis M A. Virtual structures for high-precision cooperative mobile robotic control. In: Proceedings of IEEE/RSJ International Conference Intelligent Robots and Systems. Osaka, Japan: IEEE, 1996. 132−139

8 Ren W, Beard R W. A decentralized scheme for spacecraft formation flying via the virtual structure approach. In: Proceedings of the 2003 American Control Conference. Denver, USA: IEEE, 2003. 1746−1751

9 Young B J, Beard R W, Kelsey J M. A control scheme for improving multi-vehicle formation maneuvers. In: Proceedings of the 2001 American Control Conference. Arlington, USA: IEEE, 2001. 704−709

10 Keviczky T, Borrellit F, Balas G J. Hierarchical design of decentralized receding horizon controllers for decoupled systems. In: Proceedings of the 43rd IEEE Conference on Decision and Control. Nassau, Bahamas: IEEE, 2004. 1592−1597

11 Mayne D Q, Rawlings J B, Rao C V, Scokaert P O M. Constrained model predictive control: stability and optimality. *Automatica*, 2000, **36**(6): 789−814

12 Scokaert P O M, Mayne D Q, Rawlings J B. Suboptimal model predictive control: feasibility implies stability. *IEEE Transactions on Automatic Control*, 1999, **44**(3): 648−654

13 Dunbar W B, Murray R M. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 2006, **42**(4): 549−558

14 Rimon E, Kaditrchek D E. Exact robot navigation using artificial potential functions. *IEEE Transactions on Robotics and Automation*, 1992, **8**(5): 501−518

15 Takateru U, Kenji O, Yukio T. Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions. In: Proceedings of IEEE International Conference on Intelligent Robots and Systems. Sendai, Japan: IEEE, 2004. 2428−2433

16 Bloch A M, Reyhanoglu M R, McClamroch N H. Control and stabilization of nonholonomic dynamic systems. *IEEE Transactions on Automatic Control*, 1992, **37**(11): 1746−1757

17 Murray R M, Sastry S S. Nonholonomic motion planning: steering using sinusoids. *IEEE Transactions on Automatic Control*, 1993, **38**(5): 700−716

18 Gu D B, Hu H S. A stabilizing receding horizon regulator for nonholonomic mobile robots. *IEEE Transactions on Robotics*, 2005, **21**(5): 1022−1028

YANG Tian-Tian Ph. D. candidate in Department of Control Science and Engineering at Harbin Institute of Technology. Her research interest covers formation control of mobile robots and distributed robust control.
E-mail: creance@126.com

LIU Zhi-Yuan Professor at Harbin Institute of Technology. His research interest covers automotive elective control, robotics, robust control, and model predictive control. Corresponding author of this paper. E-mail: liuzy_hit@hit.edu.cn

CHEN Hong Professor at Jilin University. Her research interest covers model predictive control, optimal and robust control, and applications in process engineering and mechatronic systems. E-mail: chenh@jlu.edu.cn

PEI Run Professor at Harbin Institute of Technology. His research interest covers high-precision servo-system, robotics, computer control, and random vibration control.
E-mail: creekduo@163.com