Multi-robot formation control using distributed null space behavioral approach

Ahmad, Shakeel; Feng, Zhi; Hu, Guoqiang

2014

Ahmad, S., Feng, Z., & Hu, G. (2014). Multi-robot formation control using distributed null space behavioral approach. 2014 IEEE International Conference on Robotics and Automation (ICRA), 3607-3612.

https://hdl.handle.net/10356/79314

https://doi.org/10.1109/ICRA.2014.6907380

Multi-Robot Formation Control Using Distributed Null Space Behavioral Approach

Shakeel Ahmad, Zhi Feng, and Guoqiang Hu

***Abstract*—This paper presents a distributed formation control method for a group of robots. The global objective of achieving a desired formation is obtained by dividing it into a set of local objectives which are achieved in a distributed manner. A basic repetitive pattern in the desired formation is identified and a corresponding unique differentiable task function is defined based on the position coordinates of the robots forming the pattern. Neighbor selection rules are designed for the robots in such a way that each robot is part of one or more such patterns. A singularity-robust task-priority inverse kinematics method is used to design velocity controllers to achieve these patterns. Since a robot can receive multiple control actions being part of multiple task functions or patterns, a distributed null space behavioral (NSB) approach is designed to combine such multiple control actions in a prioritized way. A comprehensive stability analysis of the proposed approach based on Lyapunov methods is presented. Simulation results are provided to verify the effectiveness of the proposed approach.**

***Index Terms*—Multi-robot Systems; Formation Control; NSB Approach;**

I. Introduction

The rise of interest in multi-robot systems can be attributed to the versatility of scenarios in which they can outperform single robots. Many complex tasks or missions in industry, military and everyday applications can be carried out by teams of robots efficiently. Working with a team of simpler robots could be more cost-effective than working with a single costly robot. Multiple robots add redundancy to the overall system, where the group can manage to achieve the objective even if some robots become faulty.

Multi-robot formation control has been attempted extensively. In [1], artificial potential functions were used to study the control of a swarm or group of robots. A leader-follower method has been studied for formation control [2]–[4]. Decentralized control was used in [5] to connect independent robotic vehicles so that they act together to achieve some common formation. The work in [6] deals with the decomposition of complex formations into simpler formation tasks and the formation solution design in a decentralized manner. Consensus strategies have also been employed to study the same problem [7], [8]. Adaptive control techniques for formation control can be found in [9]. Recently, distributed formation control strategies have been proposed. In [10], a distributed formation control algorithm is developed based on distributed position estimation using relative position measurements. Circular formation for a group of unicycles is achieved by using distributed control laws based on graph theory in [11].

A major motivation for control of autonomous multi-robot systems comes from the biological world. Different behaviors of animals acting in groups such as ants, birds and fishes have been studied extensively and used in modeling of multi-robot systems. In [12], formation behaviors were fused with other navigational behaviors to achieve different tasks for a multi-robot system.

In behavior based schemes, complex tasks are divided into a group of smaller and simpler tasks. Each task generates a control action for the robots. Problem arises when control actions issued by different tasks conflict with each other. Several control strategies such as competitive, cooperative and priority based approaches have been proposed to address this problem. In competitive methods, the contradiction are solved by choosing a winning task or behavior based on a deciding function [13]. In cooperative methods [14], the control actions from the contradicting tasks are combined together using some weights. In [15], the authors provided an experimental validation of the null space behavioral (NSB) approach in formation control. NSB approach is a behavior based control approach which has its roots in singularity-robust task priority inverse kinematics [16]. NSB approach is a task priority method [16]–[20], which makes sure the completion of high priority tasks by projecting the low priority tasks to the null space of higher priority tasks and low priority tasks are fulfilled only if they do not contradict with high priority tasks.

Unlike previous works which use centralized NSB, we propose a distributed NSB based approach in which the task of achieving a global formation is treated as a combination of multiple local tasks which are executed locally by small groups of robots. These local tasks are represented by task functions which depend on the positions of robots constituting the group. Using the task-priority inverse kinematics approach, controllers are designed for achieving these local tasks. Since in a formation structure, a robot may be a part of more than one groups or tasks, it will get multiple control actions. NSB based method is designed to compile the multiple control actions for an individual robot in a prioritized way. The stability analysis of the proposed approach for triangular formation and simulation results are presented to show the effectiveness of the proposed approach.

S. Ahmad, Z. Feng, and G. Hu are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798. This work was supported in part by Singapore MOE AcRF Tier 1 Grant RG60/12(2012-T1-001-158) and the PhD grant from the Institute for Media Innovation, Nanyang Technological University. E-mail: {shakeel001, zfeng001}@e.ntu.edu.sg, gqhu@ntu.edu.sg.

II. Problem Statement and Control Design

Consider a team of N mobile robots which are initially at random positions in a workplace. The robots are required to achieve a desired formation which can be a combination of identical patterns or a combination of different patterns, in a distributed manner using the position information of the other robots.

A. Robot Model

For the team of N mobile robots, the position of robot i is denoted as

$$p_i(t) = [x_i(t),\ y_i(t)]^T,\ i = 1, 2, \cdots, N. \tag{1}$$

The velocity vector for each robot is denoted by

$$v_i(t) = \dot{p}_i(t),\ i = 1, 2, \cdots, N. \tag{2}$$

For subsequent design and analysis, let

$$p = [p_1, p_2, \cdots, p_N]^T,\ v = [v_1, v_2, \cdots, v_N]^T.$$

B. NSB Design

Under the framework of the NSB approach, the desired task is divided into a set of M simpler tasks. Let the j^{th} task be represented by the task function α_j with

$$\alpha_j = f_j(p),\ j = 1, 2, \cdots, M, \tag{3}$$

where $f_j(\cdot)$: $\mathbb{R}^{2N} \to R^m$ is a continuously differentiable vector valued function. Thus, the time derivative of (3) can be written as

$$\dot{\alpha}_j = \sum_{i=1}^{N} \frac{\partial f_j(p)}{\partial p_i} v_i = J_j(p)v, \tag{4}$$

where $J_j(p) \in \mathbb{R}^{m\times 2N}$ is the Jacobian matrix for $f_j(p)$ and $v \in \mathbb{R}^{2N}$ is the velocity. Using a singularity-robust task-priority inverse kinematics technique, the control action for the robots to achieve desired value of task function i.e., $\alpha_{j,d}$ is designed as

$$v = J_j^{\dagger}(p)\dot{\alpha}_{j,d}, \tag{5}$$

where $J_j^{\dagger}$ is a Moore-Penrose pseudoinverse of the task function Jacobian matrix J_j and for underdetermined Jacobian matrix, it is given by

$$J_j^{\dagger} = J_j^T (J_j J_j^T)^{-1}. \tag{6}$$

In order to avoid the problem of numerical drift, the controller is redesigned as

$$v = J_j^{\dagger}(p)(\dot{\alpha}_{j,d} - K_j \tilde{\alpha}_j), \tag{7}$$

where $\tilde{\alpha}_j = \alpha_j - \alpha_{j,d}$ and K_j is a diagonal controller gain matrix with positive diagonal elements.

Using the NSB approach, the task with highest priority is always executed while lower priority tasks are executed by projecting them onto the subspaces where they do not oppose the execution of higher priority tasks. For M tasks, with 'T_1' being the highest priority index and 'T_M' being the lowest, using NSB the overall control action can be written as

$$v = v_{T_1} + (I - J_1^{\dagger} J_1)v_{T_2} + \cdots + (I - J_{1\cdots M-1}^{\dagger} J_{1\cdots M-1})v_{T_M}, \tag{8}$$

where $\left(I - J_1^{\dagger} J_1\right)$ is the null-space projector of task T_1 and $(I - J_{1\cdots M-1}^{\dagger} J_{1\cdots M-1})$ represents the null-space projector of all the tasks from T_1 to T_{M-1} obtained by stacking their corresponding Jacobian matrices [15], [18].

Another way of combining multiple tasks using NSB is given by

$$v = v_{T_1} + (I - J_1^{\dagger} J_1)v_{T_2} + \cdots + \prod_{i=1}^{M-1} (I - J_i^{\dagger} J_i)v_{T_M}, \tag{9}$$

which projects lower priority tasks onto the null space of its successive higher priority task only instead of projecting onto the null space of all higher priority tasks as in (8).

C. Control Development

The global objective of achieving a desired formation is divided into a set of local objectives which are represented by their corresponding task functions. These local objectives will be achieved by small groups of robots.

One leader is chosen which selects its neighbors where number of neighbors depends on desired formation defined by a task function. The rest of the robots mimic the leader and each of them selects same number of neighbours as leader. A set of neighbor selection rules for the robots will be defined based on a minimum distance criteria.

Once the groups are formed, each group of robots works to achieve the similar task function as achieved by group containing leader. A problem arises when a robot is a part of more than one group and gets different control actions from different groups. In this scenario, the NSB method is utilized to avoid the contradiction in deciding the control velocity for the robot by assigning priorities to the control actions coming from different groups. The proposed strategy is explained using a case study for triangular formation in Section III.

The previous implementations of NSB in the literature usually deal with one group of robots performing multiple tasks. Whereas, this implementation is focused on multiple groups of robots, performing different or same tasks with one or more robots in common among the groups, acting as the source of contention. This contention is avoided by assigning priorities among the groups.

Consider a scenario where robot 'i' is a part of three different groups a, b and c with a population of N_a, N_b and N_c robots, respectively. The group a has the highest priority. Each group tries to achieve objectives represented by their task functions. Using the task-priority inverse kinematics technique, the corresponding velocities $v_a \in \mathbb{R}^{2N_a}$, $v_b \in \mathbb{R}^{2N_b}$, and $v_c \in \mathbb{R}^{2N_c}$, are designed as

$$v_i = J_i^{\dagger}(p_i)\dot{\alpha}_{i,d}, \tag{10}$$

where i=a, b, c and p_a, p_b, and p_c represent the positions of robots in groups a, b, and c, respectively.

The above velocity actions cannot be combined using the usual NSB implementation as each velocity vector pertains to a different set of robots with one or more common robots among them. The control action for robot 'i' is obtained by combining the related sections from v_a and null projectors of v_a on the other tasks as

$$v_i = [v_a]_i + [(I - J_a^\dagger J_a)v_b]_i + [(I - J_{ab}^\dagger J_{ab})v_C]_i, \quad (11)$$

where index 'i' represents the section of controller corresponding to robot 'i'.

III. Triangular Formation

In this section, the proposed method is demonstrated through a group of robots which form a triangular formation in a distributed manner. In the triangular formation, the robot placed at the top vertex of triangle is considered as the leader robot. To form its local triangle, it selects two neighbors from the population based on a minimum distance criteria. For the rest of the robots, including the neighbors of the leader robot, each selects two neighbors from the remaining population of robots and moves towards making a triangle with their neighbors. Mathematically, each triangle group will be assigned the same task function due to the same local geometric shape. If the desired formation comprises different shapes for different groups of robots, then different task functions can be assigned to different groups.

To facilitate the neighbor selection for the robots, a set of neighbor selection rules are designed. In absence of these rules, conflict can occur among robots over neighbour selection and they may not be able to converge to the desired formation. These rules can be changed if need arises for a different formation structure. For triangular formation, the neighbor selection rules are defined as follows.

Algorithm 1: (Neighbor Selection Algorithm)

1). The leader robot selects its two neighbors (left and right) from $N-1$ robots based on a minimum distance rule.

2). A list containing the selected neighbors is updated

$$C = \{r_i, r_j\}.$$

3). A list of robots available to be selected as neighbors is maintained

$$A = \{r_1, r_2, \cdots, r_{N-1}\} - C.$$

4). The first element from C is picked. Two minimum distance neighbors are selected from A. Repeat Steps 2 and 3.

5). The second element from C is picked and its left neighbor is selected the same as the right neighbor of the first element of C. The right neighbor is selected based on the minimum distance rule. Steps 2 and 3 are repeated.

6). The first two elements of C are deleted and Steps 4 and 5 are repeated until list A is depleted.

Remark 1: The above rules use centralized neighbour selection based on distance information only. To implement these rules in a distributed manner, distance information alone is not enough and additional information such as communication among the robots shall be used. The centralized neighbor selection is conducted only when the robot formation control is initiated, but not throughout the whole formation control process.

Since the basic building block of the desired formation is a triangle and it is repeated throughout the formation, one task function describing a triangle can be assigned to all groups of robots. One group includes three robots with one as the parent node and the other two as left and right children nodes.

The task function defining an equilateral triangle is given as

$$\alpha_i = \left[y_i - y_l,\, y_i - y_r,\, x_r - x_l,\, x_i - \frac{x_l}{2} - \frac{x_r}{2}\right]^T, \quad (12)$$

where 'i' represents the leader of group and 'l' and 'r' represent the left and right neighbors of 'i', respectively.

There are different ways to define the above task function provided that it is a differentiable function and its Jacobian is non-singular.

The desired value of the task function is given as

$$\alpha_{i,d} = \left[\begin{array}{cccc} altitude, & altitude, & base, & 0 \end{array}\right]^T. \quad (13)$$

The values of *altitude* and *base* can be selected to obtain a basic triangle of desired size. The corresponding Jacobian matrix for the above task function is given by

$$J_i = \begin{bmatrix} 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 1 & 0 & -\frac{1}{2} & 0 & -\frac{1}{2} & 0 \end{bmatrix}$$

Since a robot may get control laws from multiple groups, a priority assignment for combining different control laws can be written in a descending priority order as

1). Control law from task function in which robot 'i' is right neighbor.

2). Control law from task function in which robot 'i' is left neighbor.

3). Control law from task function in which robot 'i' is parent member.

The general combined velocity controller for a single robot using the above priority rules is defined as follows.

$$\begin{aligned} v_i \;=\; & -[J_a^\dagger(K_a\tilde{\alpha}_a + \dot{\alpha}_{a,d})]_{(5:6,1)} \\ & -[(I - J_a^\dagger J_a)J_b^\dagger(K_b\tilde{\alpha}_b + \dot{\alpha}_{b,d})]_{(3:4,1)} \\ & -[(I - J_{ab}^\dagger J_{ab})J_c^\dagger(K_c\tilde{\alpha}_c + \dot{\alpha}_{c,d})]_{(1:2,1)}, \quad (14) \end{aligned}$$

where 'a', 'b' and 'c' represent the task functions in which 'i' appears as the right neighbor, left neighbor and parent, respectively, and K_a, K_b, and K_c are the gains for three control actions. If a robot is not a part of all these three task functions, then only the relevant portion of controller will be implemented.

IV. Stability Analysis

For triangular formation, a robot can be a part of at most three groups. Stability analysis for each case is presented to show that even if a robot is shared by multiple groups, the proposed method drives the robots to the desired formation.

A. Each Robot is subject to only one task function

This is the most basic case with only one task function. It indicates that the robot is involved in only one basic triangle.

The task function is given as

$$\alpha_a = \begin{bmatrix} y_1 - y_2, & y_1 - y_3, & x_3 - x_2, & x_1 - \frac{(x_2+x_3)}{2} \end{bmatrix}^T \tag{15}$$

The desired value of task function $\alpha_{a,d}$ is given as in (13).

Based on (5-7), the velocity controller for this case is presented as

$$v_a = -J_a^{\dagger}[(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})]. \tag{16}$$

Under the designed controller (16), a desired formation for this case is achieved if the designed controller gain matrix K_a is positive.

Stability analysis: Consider a Lyapunov function based on the task function error

$$V = \frac{1}{2}\left\|\tilde{\alpha}_a\right\|^2. \tag{17}$$

Based on (5-7) and the velocity controller, we have

$$\dot{\alpha}_a = J_a v_a. \tag{18}$$

From (18) and (16), we have

$$\dot{\alpha}_a = -K_a\tilde{\alpha}_a + \dot{\alpha}_{a,d}. \tag{19}$$

Thus, using (7) and (19), the derivative of (17) with respect to time is

$$\dot{V} = \tilde{\alpha}_a^T(\dot{\alpha}_a - \dot{\alpha}_{a,d}) = -\lambda_{\min}(K_a)\left\|\tilde{\alpha}_a\right\|^2. \tag{20}$$

Since the controller gain matrix K_a is positive, it is not difficult to obtain that $V \in L_\infty$. Therefore, according to Barbalat's Lemma, one can conclude that $\alpha_a \to \alpha_{a,d}$ as $t \to \infty$.

B. Each robot is subject to not more than two task functions

This case considers a basic formation pattern where each robot gets not more than two control actions from two different groups. The following analysis shows that using the proposed controller, the robot can ensure that task functions of all of its groups can be satisfied. Fig.1 represents the scenario.

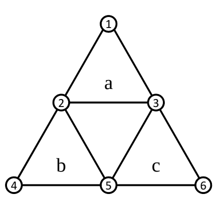


Fig. 1: Each robot is subjected to not more than two task functions

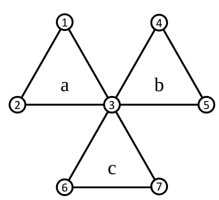


Fig. 2: Each robot is subjected to not more than three task functions

Here, robots 2, 3, and 5 get two control actions from their corresponding groups. Combining the two control actions using NSB, the completion of tasks of all three groups involved can be guaranteed.

The task functions governing the three triangles are

$$\begin{aligned}
\alpha_a &= \begin{bmatrix} y_1 - y_2 & y_1 - y_3 & x_3 - x_2 & x_1 - \frac{(x_2+x_3)}{2} \end{bmatrix}^T, \\
\alpha_b &= \begin{bmatrix} y_2 - y_4 & y_2 - y_5 & x_5 - x_4 & x_2 - \frac{(x_4+x_5)}{2} \end{bmatrix}^T, \\
\alpha_c &= \begin{bmatrix} y_3 - y_5 & y_3 - y_6 & x_6 - x_5 & x_3 - \frac{(x_5+x_6)}{2} \end{bmatrix}^T.
\end{aligned} \tag{21}$$

Based on the proposed velocity controller designed in (14), the corresponding controllers for different task functions are expressed as follows.

For the first task function α_a, the velocity controller is

$$\begin{aligned}
v_{a,1} &= -[J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})]_{(1:2,1)}, \\
v_{a,2} &= -[J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})]_{(3:4,1)} \\
&\quad -[(I - J_a^{\dagger}J_a)J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})]_{(1:2,1)}, \\
v_{a,3} &= -[J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})]_{(5:6,1)} \\
&\quad -[(I - J_a^{\dagger}J_a)J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})]_{(1:2,1)},
\end{aligned} \tag{22}$$

For the second task function α_b, the velocity controller is

$$\begin{aligned}
v_{b,2} &= -[J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})]_{(3:4,1)} \\
&\quad -[(I - J_b^{\dagger}J_b)J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})]_{(1:2,1)}, \\
v_{b,4} &= -[J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})]_{(3:4,1)}, \\
v_{b,5} &= -[J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})]_{(5:6,1)} \\
&\quad -[(I - J_b^{\dagger}J_b)J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})]_{(3:4,1)},
\end{aligned} \tag{23}$$

For the third task function α_c, the velocity controller is

$$\begin{aligned}
v_{c,3} &= -[J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})]_{(5:6,1)} \\
&\quad -[(I - J_a^{\dagger}J_a)J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})]_{(1:2,1)}, \\
v_{c,5} &= -[J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})]_{(5:6,1)}, \\
&\quad -[(I - J_b^{\dagger}J_b)J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})]_{(3:4,1)}, \\
v_{c,6} &= -[J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})]_{(5:6,1)},
\end{aligned} \tag{24}$$

Based on the proposed controller (22-24), a desired formation for this case will be achieved, provided that the following conditions are satisfied:

$$\begin{aligned}
\lambda_{\min}(K_a) &> \frac{1}{2\gamma}(\|R_2K_a\| + \|R_4K_a\|), \\
\lambda_{\min}(R_1K_b) &> \frac{1}{2}(\|R_2K_a\| + 2), \\
\lambda_{\min}(R_3K_c) &> \frac{1}{4}(2\|R_4K_a\| + \|R_5K_b\|^2).
\end{aligned} \tag{25}$$

Stability analysis: Consider a Lyapunov function consisting of the errors for the three task functions

$$V = \frac{\gamma}{2}\left\|\tilde{\alpha}_a\right\|^2 + \frac{1}{2}\left\|\tilde{\alpha}_b\right\|^2 + \frac{1}{2}\left\|\tilde{\alpha}_c\right\|^2, \tag{26}$$

where γ is a positive constant.

The time derivative of (26) is

$$\begin{aligned}
\dot{V} &= \gamma\tilde{\alpha}_a^T\dot{\tilde{\alpha}}_a + \tilde{\alpha}_b^T\dot{\tilde{\alpha}}_b + \tilde{\alpha}_c^T\dot{\tilde{\alpha}}_c \\
&= \gamma\tilde{\alpha}_a^T(\dot{\alpha}_a - \dot{\alpha}_{a,d}) + \tilde{\alpha}_b^T(\dot{\alpha}_b - \dot{\alpha}_{b,d}) \\
&\quad +\tilde{\alpha}_c^T(\dot{\alpha}_c - \dot{\alpha}_{c,d}).
\end{aligned} \tag{27}$$

Based on the proposed controllers for different task functions in (22-24), we can rewrite them in matrix form as

$$\begin{aligned}
v_a &= -J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d}) \\
&\quad -E_1[(I - J_a^{\dagger}J_a)J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \\
&\quad -E_5[(I - J_a^{\dagger}J_a)J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})],
\end{aligned} \tag{28}$$

$$\begin{aligned}
v_b &= -E_2[J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \\
&\quad -E_4[J_a^{\dagger}(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})] \\
&\quad -E_3[(I - J_a^{\dagger}J_a)J_b^{\dagger}(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \\
&\quad -E_6[(I - J_b^{\dagger}J_b)J_c^{\dagger}(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})],
\end{aligned} \tag{29}$$

$$
\begin{aligned}
v_c &= -E_3[(I - J_a^\dagger J_a)J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})] \\
&\quad -E_7[(I - J_b^\dagger J_b)J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})] \\
&\quad -E_8[J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})] \\
&\quad -E_9[J_a^\dagger(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})] \\
&\quad -E_{10}[J_b^\dagger(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})], \qquad (30)
\end{aligned}
$$

where E_1 to E_{10} are matrices with 0 and 1 elements, which are used to extract components from the controller related to a particular robot.

Due to the selection of same task functions for each triangle, it is not difficult to obtain

$$
\begin{aligned}
(I - J_a^\dagger J_a)J_b^\dagger &= (I - J_a^\dagger J_a)J_b^\dagger \qquad (31)\\
&= (I - J_a^\dagger J_a)J_c^\dagger = 0, \\
\dot{\alpha}_{a,d} &= \dot{\alpha}_{b,d} = \dot{\alpha}_{c,d} = 0. \qquad (32)
\end{aligned}
$$

Combining (4), (28-31) and the fact that $J_i^\dagger J_i = I$, $i = a, b, c$ gives

$$
\begin{aligned}
\dot{\alpha}_a &= J_a v_a = -K_a\tilde{\alpha}_a, \qquad (33)\\
\dot{\alpha}_b &= -R_1K_b\tilde{\alpha}_b - R_2K_a\tilde{\alpha}_a, \qquad (34)\\
\dot{\alpha}_c &= -R_3K_c\tilde{\alpha}_c - R_4K_a\tilde{\alpha}_a - R_5K_b\tilde{\alpha}_b, \qquad (35)
\end{aligned}
$$

where R_1 to R_5 are constant matrices obtained by multiplying different Jacobian and E matrices. Using (33-35) in (27) and applying Young's inequality, it can be established that

$$
\begin{aligned}
\|R_2K_a\|\,\|\tilde{\alpha}_b\|\,\|\tilde{\alpha}_a\| &\leq \frac{\|R_2K_a\|}{2}(\|\tilde{\alpha}_a\|^2 + \|\tilde{\alpha}_b\|^2), \\
\|R_4K_a\|\,\|\tilde{\alpha}_c\|\,\|\tilde{\alpha}_a\| &\leq \frac{\|R_4K_a\|}{2}(\|\tilde{\alpha}_c\|^2 + \|\tilde{\alpha}_a\|^2), \\
\|R_5K_b\|\,\|\tilde{\alpha}_c\|\,\|\tilde{\alpha}_b\| &\leq \|\tilde{\alpha}_b\|^2 + \frac{\|R_5K_b\|^2}{4}\|\tilde{\alpha}_c\|^2 . \qquad (36)
\end{aligned}
$$

Based on (36), the expression in (27) becomes

$$
\begin{aligned}
\dot{V} &\leq -[\gamma\lambda_{\min}(K_a) - \frac{1}{2}(\|R_2K_a\| + \|R_4K_a\|)]\,\|\tilde{\alpha}_a\|^2 \qquad (37)\\
&\quad -[\lambda_{\min}(R_1K_b) - \frac{1}{2}(\|R_2K_a\| + 2)]\,\|\tilde{\alpha}_b\|^2 \\
&\quad -[\lambda_{\min}(R_3K_c) - \frac{1}{4}(2\|R_4K_a\| + \|R_5K_b\|^2)]\,\|\tilde{\alpha}_c\|^2 .
\end{aligned}
$$

According to the conditions (25), it is not difficult to get that $\dot{V}$ is negative definite, which implies that $V \in L_\infty$. Therefore, Barbalat's Lemma is utilized to conclude that $\alpha_a \to \alpha_{a,d}$, $\alpha_b \to \alpha_{b,d}$, and $\alpha_c \to \alpha_{c,d}$ as $t \to \infty$.

C. Each robot is subject to not more than three task functions

Fig. 2 represents those parts of overall formation where a single robot is shared by three different groups. The task functions for this case are still defined as that in (21). Based on the priority assignment described in the previous section, the combined controllers in matrix form for this case are given as follows.

$$
\begin{aligned}
v_a &= -J_a^\dagger(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d}) \qquad (38)\\
&\quad -E_6[(I - J_a^\dagger J_a)J_b^\dagger(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \\
&\quad -E_5[(I - J_{ab}^\dagger J_{ab})J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})], \\
v_b &= -E_{11}[J_b^\dagger(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \qquad (39)\\
&\quad -E_{10}[J_a^\dagger(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})] \\
&\quad -E_7[(I - J_a^\dagger J_a)J_b^\dagger(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \\
&\quad -E_1[(I - J_{ab}^\dagger J_{ab})J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})], \\
v_c &= -E_2[J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})] \qquad (40)\\
&\quad -E_9[J_a^\dagger(K_a\tilde{\alpha}_a - \dot{\alpha}_{a,d})] \\
&\quad -E_4[(I - J_a^\dagger J_a)J_b^\dagger(K_b\tilde{\alpha}_b - \dot{\alpha}_{b,d})] \\
&\quad -E_3[(I - J_{ab}^\dagger J_{ab})J_c^\dagger(K_c\tilde{\alpha}_c - \dot{\alpha}_{c,d})],
\end{aligned}
$$

Based on the proposed controller (38-40), the sufficient conditions for achieving a desired formation for this case are

$$
\begin{aligned}
\lambda_{\min}(K_a) &> \frac{1}{2\beta}(\|R_5K_a\| + \|R_7K_a\|), \\
\lambda_{\min}(R_4K_b) &> \frac{1}{2}\|R_5K_a\|, \\
\lambda_{\min}(R_6K_c) &> \frac{1}{2}\|R_7K_a\|. \qquad (41)
\end{aligned}
$$

Stability analysis: Consider a Lyapunov function consisting of the errors in the three task functions

$$
V = \frac{\beta}{2}\|\tilde{\alpha}_a\|^2 + \frac{1}{2}\|\tilde{\alpha}_b\|^2 + \frac{1}{2}\|\tilde{\alpha}_c\|^2, \qquad (42)
$$

where β is a positive constant.

The time derivative of (42) is written as

$$
\begin{aligned}
\dot{V} &= \beta\tilde{\alpha}_a^T\dot{\tilde{\alpha}}_a + \tilde{\alpha}_b^T\dot{\tilde{\alpha}}_b + \tilde{\alpha}_c^T\dot{\tilde{\alpha}}_c \\
&= \beta\tilde{\alpha}_a^T(\dot{\alpha}_a - \dot{\alpha}_{a,d}) + \tilde{\alpha}_b^T(\dot{\alpha}_b - \dot{\alpha}_{b,d}) \\
&\quad + \tilde{\alpha}_c^T(\dot{\alpha}_c - \dot{\alpha}_{c,d}). \qquad (43)
\end{aligned}
$$

Similar to equation (31), the null projector terms are zero. Thus, it follows from (4) using (38-40) that

$$
\begin{aligned}
\dot{\alpha}_a &= J_a v_a = -K_a\tilde{\alpha}_a, \qquad (44)\\
\dot{\alpha}_b &= -R_4K_b\tilde{\alpha}_b - R_5K_a\tilde{\alpha}_a, \qquad (45)\\
\dot{\alpha}_c &= -R_6K_c\tilde{\alpha}_c - R_7K_a\tilde{\alpha}_a, \qquad (46)
\end{aligned}
$$

where R_6 and R_7 are constant matrices obtained by multiplying different Jacobian and E matrices.

Submitting (44-46) into (43) gives

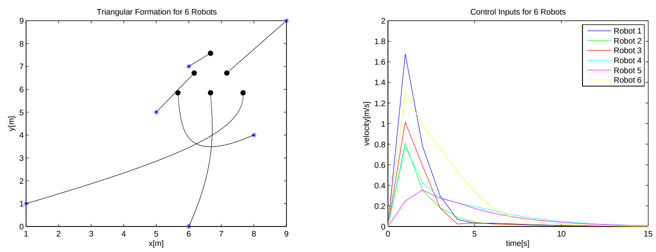
$$
\begin{aligned}
\dot{V} &\leq -[\beta\lambda_{\min}(K_a) - \frac{1}{2}(\|R_5K_a\| + \|R_7K_a\|)]\,\|\tilde{\alpha}_a\|^2 \\
&\quad -[\lambda_{\min}(R_4K_b) - \frac{1}{2}\|R_5K_a\|]\,\|\tilde{\alpha}_b\|^2 \\
&\quad -[\lambda_{\min}(R_6K_c) - \frac{1}{2}\|R_7K_a\|]\,\|\tilde{\alpha}_c\|^2, \qquad (47)
\end{aligned}
$$

where Young's Inequality is utilized to upper bound the product terms. Therefore, it follows from the conditions (41) that $V \in L_\infty$ which implies that $\alpha_a \to \alpha_{a,d}$, $\alpha_b \to \alpha_{b,d}$, and $\alpha_c \to \alpha_{c,d}$ as $t \to \infty$ by using Barbalat's Lemma.

Remark 2: The above analysis addresses all the three cases on how a robot can appear in the triangular formation. Considering the fact that the proposed method is distributed, we can infer that the effect of a robot control action is limited to itself and its neighbors. Combining this fact with our analysis that stability is achievable in all the three cases, we can claim that the proposed method exhibits stability for a larger number of robots.

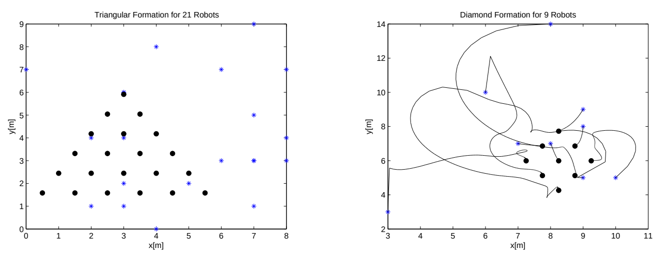
V. Simulation Results

In the simulation, a random population of robots is generated with random initial starting positions. According to Algorithm 1, the selection of neighbors is done and the proposed distributed controller in (11) is executed which results in a global triangular formation for the team of robots. The desired value of all local task functions was set to $\alpha_d = [\ 0.866 \quad 0.866 \quad 1 \quad 0\ ]^T$ to achieve formation with a unit equilateral triangle as its basic building block.



(a) Formation control (b) Control inputs

Fig. 3: Formation control for 6 robots. Asteriks represent initial positions while dots represent final positions.



(a) Triangular formation (b) Diamond formation

Fig. 4: Triangular and diamond formation for a team of 21 and 9 robots, respectively. Asteriks represent initial positions while dots represent final positions.

Fig. 3 shows a team of 6 robots and corresponding control inputs required to converge towards a triangular formation. Fig. 4(a) shows the scalability of the proposed method as formations with a large number of robots are achieved. The application of proposed method to other repetitive formation structures is demonstrated by applying it to a team of 9 robots to achieve a diamond formation as shown in Fig. 4(b). One advantage of the proposed method is integration of different behaviors as shown in Fig. 5, where formation behavior is combined with a target seeking behavior with formation control being the higher priority task.

VI. Conclusions

In this work, a new implementation of NSB for formation control of multi-robot systems has been presented. Based on the desired formation, neighbors are assigned and small groups of robots are made which perform local tasks. To resolve the contradictions for robots getting control actions from more than one group, NSB is implemented individually for that particular robot. Stability analysis and simulations for triangular formation are provided to verify and evaluate the proposed method.

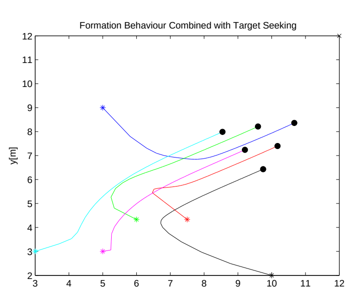


Fig. 5: Formation behavior combined with target seeking behavior. Asteriks represent initial positions while dots represent final positions. 'x' represents the target position which is at the right top corner of the figure.

References

[1] V. Gazi, "Swarm aggregations using artificial potentials and sliding-mode control," *IEEE Transactions on Robotics*, vol. 21, pp. 1208-1214, 2005.
[2] J. Shao, G. Xie, and L. Wang, "Leader-following formation control of multiple mobile vehicles," *IET Control Theory & Applications*, vol. 1, pp. 545-552, 2007.
[3] T. Gustavi and X. Hu, "Observer-Based Leader-Following Formation Control Using Onboard Sensor Information," *IEEE Transactions on Robotics*, vol. 24, pp. 1457-1462, 2008.
[4] X. Wang, W. Ni, and X. Wang, "Leader-Following Formation of Switching Multirobot Systems via Internal Model," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, pp. 817-826, 2012.
[5] J. T. Feddema, C. Lewis, and D. A. Schoenwald, "Decentralized control of cooperative robotic vehicles: theory and application," *IEEE Transactions on Robotics and Automation*, vol. 18, pp. 852-864, 2002.
[6] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," *IEEE Transactions on Robotics and Automation*, vol. 19, pp. 933-941, 2003.
[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and Cooperation in Networked Multi-Agent Systems," *Proceedings of the IEEE*, vol. 95, pp. 215-233, 2007.
[8] W. Ren, "Consensus strategies for cooperative control of vehicle formations," *IET Control Theory & Applications* vol. 1, pp. 505-512, 2007.
[9] B. S. Park, J. B. Park, and Y. H. Choi, "Adaptive Formation Control of Electrically Driven Nonholonomic Mobile Robots With Limited Information," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 41, pp. 1061-1075, 2011.
[10] K. Oh and H. Ahn, "Formation Control of Mobile Agents Based on Distributed Position Estimation," *IEEE Transactions on Automatic Control*, vol. 58, pp. 737-742, 2013.
[11] M. I. El-Hawwary and M. Maggiore, "Distributed Circular Formation Stabilization for Dynamic Unicycles," *IEEE Transactions on Automatic Control*, vol. 58, pp. 149-162, 2013.
[12] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE Transactions on Robotics and Automation*, vol. 14, pp. 926-939, 1998.
[13] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal of Robotics and Automation*, vol. 2, pp. 14-23, 1986.
[14] R. C. Arkin, "Motor schema based navigation for a mobile robot: An approach to programming by behavior," *Proc. IEEE International Conference on Robotics and Automation,* 1987, pp. 264-271.
[15] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control," *IEEE Transactions on Control Systems Technology*, vol. 17, pp. 1173-1182, 2009.
[16] G. Antonelli and S. Chiaverini, "Kinematic Control of Platoons of Autonomous Vehicles," *IEEE Transactions on Robotics*, vol. 22, pp. 1285-1292, 2006.
[17] S. Chiaverini, "Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators," *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 398-410, 1997.
[18] N. Gans, G. Hu, K. Nagarajan, and W. Dixon, "Keeping Multiple Moving Targets in the Field of View of a Mobile Camera," *IEEE Transactions on Robotics*, vol. 27, pp. 822-828, 2011.
[19] A. Marino, L. E. Parker, G. Antonelli, and F. Caccavale, "A decentralized architecture for multi-robot systems based on the Null-Space-Behavioral control with application to multi-robot border patrolling,"*Journal of Intelligent & Robotic Systems,* vol. 71, pp. 423-444, 2013.
[20] G. Antonelli, F. Arrichiello, and S. Chiaverini, "Stability analysis for the Null-Space-based Behavioral control for multi-robot systems," *IEEE Conference on Decision and Control,* 2008, pp. 2463-2468.