

Article

Decentralization of Virtual Linkage in Formation Control of Multi-Agents via Consensus Strategies

Yi Liu ^{1,2} , Junyao Gao ^{1,*}, Xuanyang Shi ^{1,3} and Chunyu Jiang ¹

¹ Intelligent Robotics Institute, School of Mechatronical Engineering, Beijing Institute of Technology, Beijing 100081, China; YiLiu@bit.edu.cn (Y.L.); 3120170111@bit.edu.cn (X.S.); jiangchunyu@bit.edu.cn (C.J.)

² Key Laboratory of Biomimetic Robots and Systems, Ministry of Education, Beijing 100081, China

³ Beijing Advanced Innovation Center for Intelligent Robots and Systems, Beijing 100081, China

* Correspondence: gaojunyao@bit.edu.cn

Received: 28 September 2018; Accepted: 20 October 2018; Published: 23 October 2018



Featured Application: The method proposed in this paper can be used for formation control autonomous robots, such as nonholonomic mobile robots, unmanned aerial vehicles and has potential applications in search and rescue missions, area coverage and reconnaissance, etc.

Abstract: Featured Application: This paper addresses the formation control of a team of agents based on the decentralized control and the recently introduced reconfigurable virtual linkage approach. Following a decentralized control architecture, a decentralized virtual linkage approach is introduced. As compared to the original virtual linkage approach, the proposed approach uses decentralized architecture rather than hierarchical architecture, which does not require role assignments in each virtual link. In addition, each agent can completely decide its movement with only exchanging states with part of the team members, which makes this approach more suitable for situations when a large number of agents and/or limited communication are involved. Furthermore, the reconfiguration ability is enhanced in this approach by introducing the scale factor of each virtual link. Finally, the effectiveness of the proposed method is demonstrated through simulation results.

Keywords: formation control; virtual linkage; virtual structure; formation reconfiguration; mobile robots; robotics

1. Introduction

Formation control is one of the most leading research areas in robotics. It has been extensively studied by researchers around the world on different platforms: mobile robots, aerial robots, spacecraft, and autonomous surface and underwater vehicles [1–8].

In the literature, formation control approaches can be classified into three basic strategies: leader-following, behavior-based, and virtual structure. In the leader–follower approach [9–11], some agents are considered as leaders, while others act as followers which track the leaders with predefined offset. However, the leader’s motion is independent of the followers. When a follower fails, the leader will keep on moving as predefined and results in the break of the formation shape. In the behavior approach [12–14], several reactive behaviors are prescribed (e.g., move-to-goal, avoid-robot, avoid-static-obstacles, and maintain-formation). The action of each agent is derived by a weighted sum of all the behaviors. The main problem with this approach is that it is difficult to formalize the group mathematically and the team of agents is not guaranteed to converge to the desired formation configuration. The virtual linkage approach considers the entire formation as a single rigid body and is able to maintain the formation shape in high precision during manoeuvres [15–17]. Perhaps the

main criticism of the virtual structure approach is that it has poor reconfiguration ability and needs to refresh the relative positions of all the team members when a different formation pattern changes.

Although these three approaches have been used in many applications, they focus more on maintaining a specified formation pattern throughout a task and few studies address the effects of formation reconfiguration. However, situations also exist where different formation patterns are needed, for example, a group of agents might need to reconfigure into different patterns to go through a gallery. A reconfigurable formation control method named virtual linkage is proposed by the authors in Reference [18]. Instead of treating the whole formation as a single rigid body, as in the virtual structure, the virtual linkage approach considers the formation as a mechanical linkage which is a collection of rigid bodies connected by joints. A virtual linkage is defined as an assembly of virtual structures (named “virtual link”) connected by virtual joints. By coordinating the value of each virtual joint, the virtual linkage approach is able to reconfigure a group of agents into different formation shapes.

Currently, the virtual linkage approach uses hierarchical architecture. In detail, the states of the virtual linkage are implemented in a virtual linkage server and broadcast to all the virtual link servers. Each virtual link’s state is in turn calculated in the corresponding virtual link server and transmits to all its virtual link members. The principal limitation of this hierarchical architecture is that it does not scale well with the number of agents in the team. In addition, due to the communication range limitations, the virtual linkage server might lose communication with some virtual link servers when the agent group covers a large area. A possible way to solve these drawbacks is to use decentralized architecture in which each agent runs a consensus strategy and totally decides its moving action with communication with parts of the members.

The concept of consensus is an important idea in control and information theory, and it has been applied to the formation control of multiple agents [2,16,19–21]. The basic idea of a consensus algorithm is that each agent updates its state’s information only based on its neighbors’ state’s information and finally enable the convergence of all the agent’s state’s information.

The main contribution of this paper is the decentralization of the virtual linkage approach via consensus strategies. Motivated by the pros and cons of the virtual linkage approach and consensus algorithm, a decentralized virtual linkage approach is presented in this paper. Instead of using hierarchical architecture, the proposed method instantiates a local copy of the virtual linkage’s state implements the same consensus algorithm on each agent to facilitate the reconfigurable formation control of a team of agents. The decentralized virtual linkage approach has several advantages as compared to the original virtual linkage approach. First, the decentralized architecture overcomes the limitations of the hierarchical architecture. In details, this approach scales well with the number of agents in the group and only requires each agent to communicate with its local neighbors. Second, with the introduction of expansion/contraction rates for each virtual link, this approach has a stronger reconfiguration ability than the traditional virtual linkage approach.

The paper is organized as follows. In Section 2, the preliminary knowledge is presented. Section 3 illustrates the control strategy for the formation movement. Simulation results are in Section 4. Finally, in Section 5, some concluding remarks of this paper are given.

2. Problem Statement

2.1. Virtual Linkage

The virtual linkage is a reconfigurable formation control method proposed by the authors in Reference [18]. The main idea of virtual linkage is to consider the entire formation as a mechanical linkage which is a collection of rigid bodies connected by joints. It can be defined as a collection of virtual structures connected by virtual joints. Instead of specifying each agent’s desired position relative to a single reference frame, as in the virtual structure, the virtual linkage approach tells each agent the virtual link it belongs to and the relative position in the corresponding virtual link. In this

way, the designed virtual linkage can be reconfigured into different formation patterns by coordinating the value of each virtual joint.

Definition 1 (Virtual Joints [18]). A virtual joint is a connection between two virtual structures and imposes constraints on their relative movement.

Definition 2 (Virtual Linkage (VL) [18]). A virtual linkage is an assembly of virtual structures (named “virtual link”) connected by virtual joints.

Figure 1 shows the comparison of an intuitive example of a virtual linkage. Three agents are designed into a virtual linkage composed of two virtual links. They are able to be configured into line formation and arrow formation by only changing the virtual joint angle.

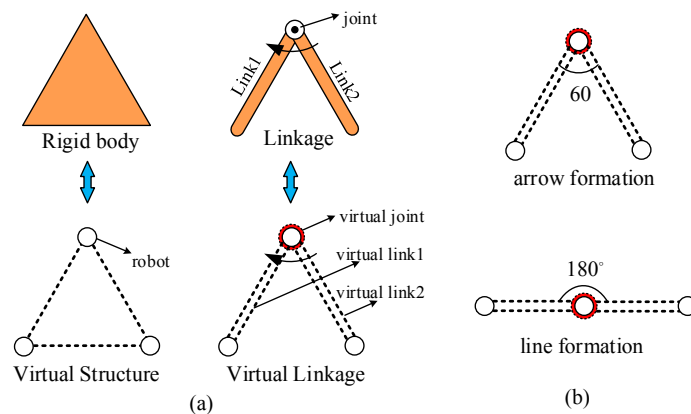


Figure 1. (a) Principle of virtual structure and virtual linkage. (b) Different formation using a specified virtual linkage.

2.2. Architecture of the Previous Virtual Linkage

Figure 2 shows the hierarchical architecture of the original virtual linkage approach proposed in Reference [18]. The states of the virtual linkage are implemented in a virtual linkage server and broadcast to all the virtual link servers. Each virtual link’s state is in turn calculated in the corresponding virtual link server and transmits to all the virtual link members. As can be seen in Figure 2, there is a demand for the virtual link server F_{vli} to exchange message with all the agents which belong to the corresponding virtual link. Meanwhile, the virtual linkage server F also has to communicate with all the virtual link servers. The disadvantages of this hierarchical architecture lie in two aspects. First, it does not scale well with the number of agents in the team with limited communication bandwidth. Second, the virtual linkage server might lose communication with some virtual link servers when the agent group covers a large area.

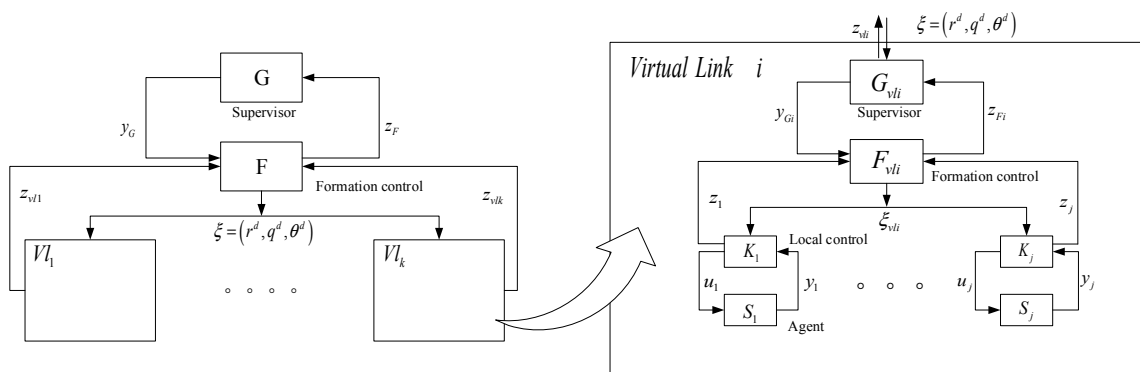


Figure 2. Hierarchical architecture of the previous virtual linkage.

2.3. Preliminaries of Digraphs and Consensus

To implement the decentralized formation control, the communication topology among a group of agents is represented as a digraph $G = (R, \varepsilon)$. In detail, $R = \{R_i | i = \{1, 2, \dots, n\}\}$ is a set of agents and are called as *nodes*. $\varepsilon \in R \times R = \{(R_i, R_j)\}$ is a set of unordered pair of nodes and called as *edge*. If (R_i, R_j) is an edge of the digraph, then information flow from R_i to R_j is allowed and R_i, R_j are neighbors. Especially, self-loop edges in the form (R_i, R_i) are not allowed. Another way to represent G is called the adjacency matrix $A = [a_{ij}] \in R^{n \times n}$. The elements a_{ij} equals 1 if there exists an edge $(R_i, R_j) \in \varepsilon$, otherwise $a_{ij} = 0$.

Theorem 1 (Consensus algorithm [22]). Let $A = [a_{ij}] \in R^{n+1 \times n+1}$ be the adjacency matrix, where $a_{ij} = 1, \forall i, j \in \{1, \dots, n\}$ once the agent j 's formation state estimate is available to agent i and 0 otherwise. $a_{i(n+1)} = 1$ if agent i has knowledge of reference value ξ_{contr} and 0 otherwise, and $a_{(n+1)k} = 0, \forall k \in \{1 \dots n + 1\}$. Then the consensus algorithm

$$\dot{\xi}_i = \frac{1}{\eta_i} \sum_{j=1}^n a_{ij} [\dot{\xi}_j - \kappa(\xi_i - \xi_j)] + \frac{1}{\eta_i} a_{i(n+1)} [\dot{\xi}_{contr}^r - \kappa(\xi_i - \xi_{contr})] \quad (1)$$

guarantees that $\xi_i \rightarrow \xi_d, \forall i$, asymptotically if and only if the graph of A has a directed spanning tree.

2.4. Agent Model

In this paper, a group of n fully actuated agents is considered. The agents are assumed to know their position in a global coordinate frame and can move in any direction with any specified velocity. The model of the agent is considered as follows:

$$\begin{aligned} \dot{x}_k &= u_{xk} \\ \dot{y}_k &= u_{yk} \end{aligned} \quad (2)$$

where $p_k = [x_k, y_k]$ and $u_k = (u_{xk}, u_{yk})$ are the position coordinate and control input of the k th agent respectively.

3. Decentralization of Virtual Linkage Approach

This section illustrates the decentralization of virtual linkage approach. First, the decentralized architecture is introduced to illustrate the advantages as compared to the hierarchical architecture. Then the consensus formation control is presented to enable each agent to decide its movement independently with only exchanging state information with its local neighbors.

3.1. Decentralized Coordination Architecture

In this paper, instead of using hierarchical architecture in which the desired destination of each agent is informed by the corresponding virtual link server, a decentralized architecture is adopted in the proposed approach. As compared to the hierarchical architecture, there does not exist virtual linkage and virtual link servers, each agent only needs to exchange information with its local neighbors.

Figure 3 shows the architecture diagram of the proposed decentralized virtual linkage approach. Each agent instantiates a local copy of consensus module, denoted as F_i . The consensus module F_i is responsible for calculating the instantiation of virtual linkage states $\xi_i = (r_i, q_i, \Theta_i, \lambda_i)$ for the i th agent, with the inputs of instantiations of virtual linkage states $\xi_j = (r_j, q_j, \Theta_j, \lambda_j)$ produced by its local neighbors. The main aim of consensus module is to drive each instantiation of virtual linkage state to converge into the desired states $\xi^d = (r^d, q^d, \Theta^d, \lambda^d)$.

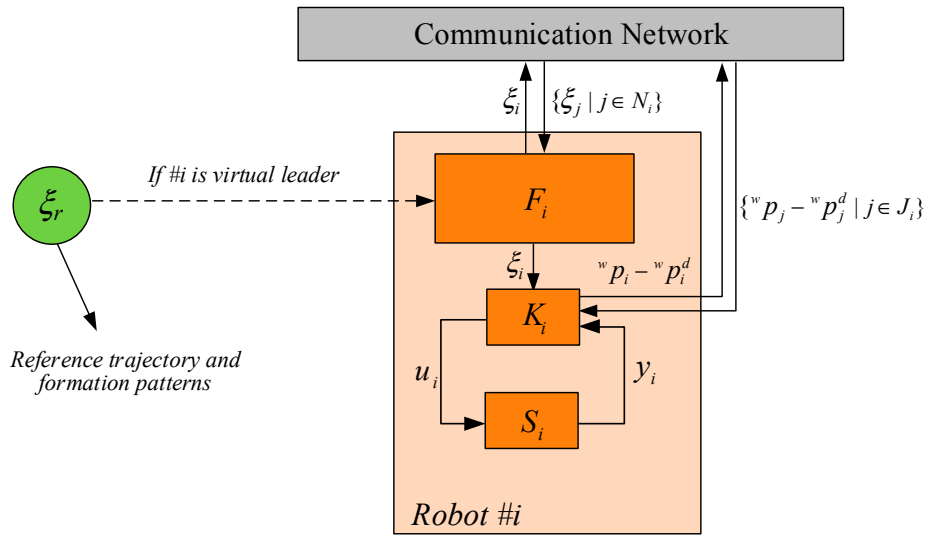


Figure 3. Decentralized virtual linkage architecture.

The states of the virtual linkage are defined as a coordination variable $\zeta = (r, q, \Theta, \lambda)$, where r and q are the desired position and attitude of the virtual linkage’s reference frame, respectively, $\Theta = (\theta_1, \theta_2, \dots, \theta_{k-1}, \theta_k)$ is the desired virtual joint angles, where k is the number of virtual links in the virtual linkage. In addition, $\lambda = (\lambda_{1x}, \lambda_{1y}, \lambda_{1z}, \dots, \lambda_{kx}, \lambda_{ky}, \lambda_{kz})$ is a vector which represents the expansion rates of all virtual links. The benefit of introducing λ lies that a group of agents is able to reconfigure into more formation shapes since the length and width of each virtual link can be specified now.

3.2. Consensus Control

3.2.1. Implication of Consensus of Virtual Linkage’s State ζ_i

As mentioned above, each agent has an instantiation of virtual linkage states $\zeta_i = (r_i, q_i, \theta_i, \lambda_i)$ and the consensus module implement consensus strategies to ensure each instantiation converge into the desired value. In this part, the implication of ζ_i is illustrated.

In the virtual linkage approach, each agent is specified with a vector $\chi = (l_i, v^j p_i^{lni})$ before task. The l_i is the ID number of virtual link which the i th agent belongs to, and $v^j p_i^{lni}$ is the i th agent’s relative position in the j th unit virtual link. Meanwhile, each agent is randomly initialized with a $\zeta_i = (r_i, q_i, \Theta_i, \lambda_i)$. With knowing the χ and ζ_i , each agent now is able to calculate its global position in the world.

Note that each ζ_i corresponds to a state of the virtual linkage (See Figure 4). The consensus module will ensure all the ζ_i converge into a common value

$$\zeta_1 = \zeta_2 = \dots = \zeta_n = {}^r \zeta_{contr}. \tag{3}$$

The team of agents forms virtual linkages and moves in desired formation shapes along a specified path. Note that the formation pattern can be easily reconfigured by reconfiguring the coordinate variable ${}^r \zeta_{contr} = ({}^r r_{contr}, {}^r q_{contr}, {}^r \Theta_{contr}, {}^r \lambda_{contr})$.

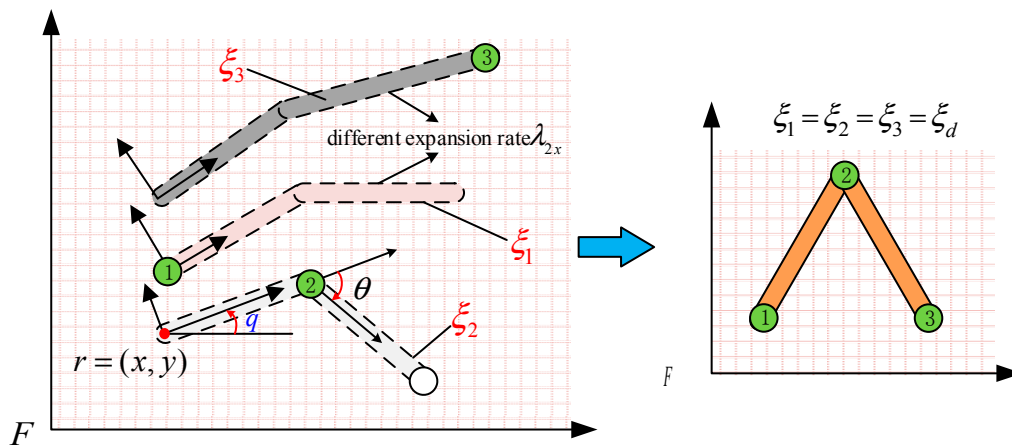


Figure 4. Implication of virtual linkage states ζ_i .

3.2.2. Consensus Module

With the previous section, each agent now is initialized with a $\zeta_i = (r_i, q_i, \theta_i, \lambda_i)$. This part aims to design a consensus law and drive ζ_i to the desired value ${}^r\zeta_{contr}$. Here, the consensus tracking algorithm in Reference [22]

$$\dot{\zeta}_i = \frac{1}{\eta_i} \sum_{j=1}^n a_{ij} [\zeta_j - \kappa(\zeta_i - \zeta_j)] + \frac{1}{\eta_i} a_{i(n+1)} [\zeta_{contr}^r - \kappa(\zeta_i - \zeta_{contr})] \quad (4)$$

is directly used, where a_{ij} are the elements of the adjacency matrix $A^c = [a_{ij}^c] \in R^{n+1 \times n+1}$ and $\eta_i = \sum_{j=1}^{n+1} a_{ij}^c$. The consensus law consists of two parts. The first term uses the information of its neighbors to make all the ζ_i converge into a common value which leads to the desired formation shape. The existence of the second term is to make formation move along the desired path ζ_{contr} . For a connected graph, consensus to the reference value is guaranteed [22].

3.3. Local Control of Each Agent

After each agent has calculated the $\dot{\zeta}_i$, then each agent is able to update the value of ζ_i using Equation (5). Note that $\lambda = [\lambda_{1x}, \lambda_{1y}, \dots, \lambda_{kx}, \lambda_{ky}]$ represents the expansion/contraction rates of each virtual link along their coordinate frame's axis (See Equation (6)). Figure 5 shows the geometry definition of the virtual linkage. The position of a specified agent can be calculated using manipulation kinematics in Equation (7). Here, ${}^j p_i$ is the relative position of Agent i in the corresponding virtual link j which it belongs to

$$\zeta_i(r_i, q_i, \Theta_i, \lambda_i) = \zeta_i + \dot{\zeta}_i \cdot dt \quad (5)$$

$${}^j p_i = \begin{bmatrix} \lambda_{iy} \cdot {}^j p_{ix}^{Ini} \\ \lambda_{iy} \cdot {}^j p_{iy}^{Ini} \\ 1 \end{bmatrix} \quad (6)$$

$${}^w p_i^d = {}^w T_R(r_i, q_i) \cdot {}^R T_{vj}^d(\Theta_i, \lambda_i) \cdot {}^j p_i \quad (7)$$

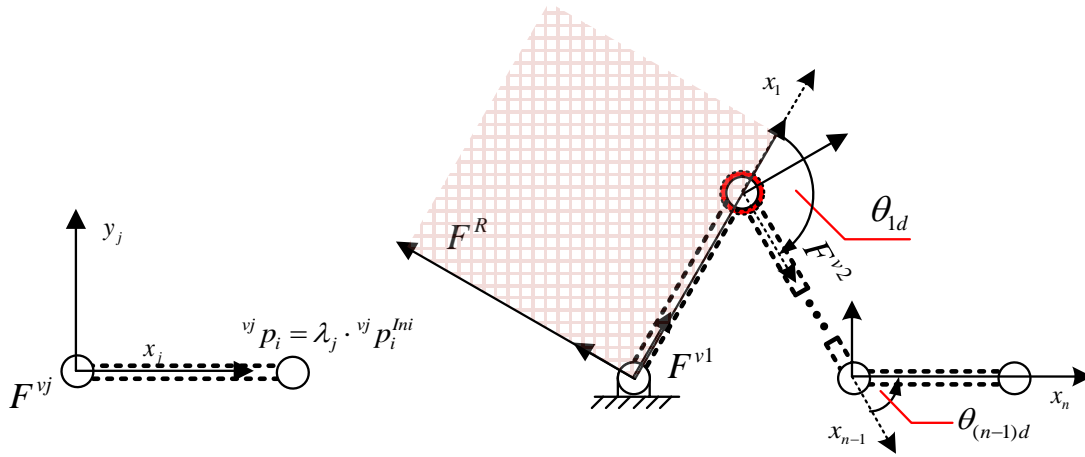


Figure 5. Geometry definition of the virtual linkage.

Finally, the desired absolute position is passed onto the local controller to position the vehicle. Each agent is supposed to know its own position ${}^w p_i$, and the consensus control algorithm in Reference [22]

$$u_i = {}^w \dot{p}_i^d - k_p \cdot ({}^w p_i - {}^w p_i^d) - \sum_{j=1}^n a_{ij}^v [({}^w p_i - {}^w p_i^d) - ({}^w p_j - {}^w p_j^d)] \quad (8)$$

is used, where a_{ij}^v are the elements of the adjacency matrix $A^v = [a_{ij}^v] \in R^{n \times n}$. For a connected graph, consensus to the reference value is guaranteed.

4. Simulation and Results

In this section, the proposed decentralized virtual linkage approach is applied to a multi-agents' formation control scenario using MATLAB. In the scenarios, nine agents are required to move around a circle while maintaining line formation shapes with a uniform distance of 0.1 m or performing formation reconfigurations by coordinating the desired virtual joint angles Θ_d and the virtual linkage extract/expansion rates λ_d .

4.1. Simulation Setup

In the scenarios, nine agents are designed as a virtual linkage which consists of two virtual links (See Figure 6) and move around a circle with a radius of 1 m in 10 s. The states of $\chi = (l_i, {}^{vj} p_i^{Lni})$ for each agent is predefined with Equations (9) and (10).

$$\text{Virtuallinkage} = \begin{cases} \text{virtuallink1} = \{\text{agent1}, \text{agent2}, \text{agent3}, \text{agent4}, \text{agent5}\} \\ \text{virtuallink2} = \{\text{agent5}, \text{agent6}, \text{agent7}, \text{agent8}, \text{agent9}\} \end{cases} \quad (9)$$

Recall that ${}^{vj} p_i$ is the representation of the i th point in the j th virtual link coordinate frame. The nine agents are initialized with:

$$\begin{aligned} {}^{v1} p_1^{Lni} &= [0, 0], {}^{v1} p_2^{Lni} = [0.2, 0], {}^{v1} p_3^{Lni} = [0.4, 0], {}^{v1} p_4^{Lni} = [0.6, 0], {}^{v1} p_5^{Lni} = [1, 0] \\ {}^{v2} p_5^{Lni} &= [0, 0], {}^{v2} p_6^{Lni} = [0.2, 0], {}^{v2} p_7^{Lni} = [0.4, 0], {}^{v2} p_8^{Lni} = [0.6, 0], {}^{v2} p_9^{Lni} = [1, 0] \end{aligned} \quad (10)$$

Meanwhile, each agent has an instantiation of virtual linkage states $\xi_i = (r_i, q_i, \Theta_i, \lambda_i)$ and is initialized as:

$$\xi_i = \text{randn}(6, 1) \quad (11)$$

Moreover, there does not exist leader selection for each virtual link. Figure 7 shows the communication topologies used for these two simulations. Notice that apart from agent 2, each agent only needs to exchange ξ_i and its own position ${}^w p_i$ with its two neighbors.

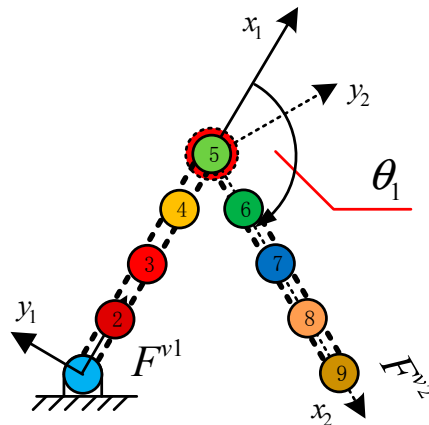


Figure 6. Predefined virtual linkage used in these two simulations.

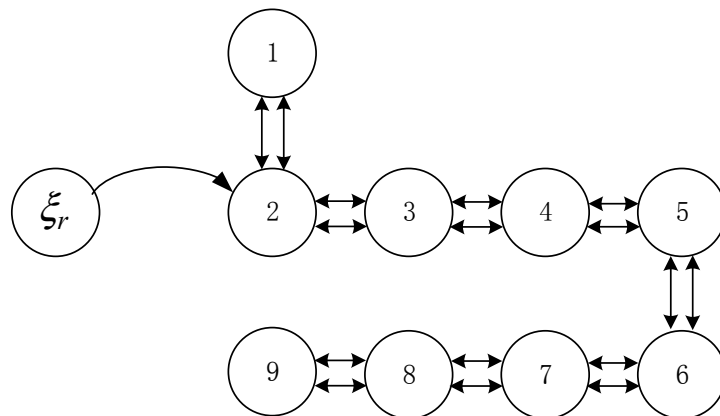


Figure 7. Information-exchange topologies.

Initially, the nine agents are required to align in a line formation with a uniform distance of 0.1 m. Then different formation shapes are reconfigured by coordinating the virtual joint angles Θ_d and the virtual linkage extract expansion rates λ_d . It is worth mentioning that instead of refreshing the relative positions of all the agents, the virtual linkage approach can be reconfigured into different shapes by only changing Θ_d and λ_d .

In the following simulations, the required trajectory and formation shapes are specified by:

$$\xi_{contr} = (r_d, q_d, \Theta_d, \lambda_d) \tag{12}$$

4.2. Formation Moving Using Decentralized Virtual Linkage

In this section, the nine agents are required to align in a line formation with a uniform distance of 0.1 m and move around a circle with a radius of 1 m in 100 s. To perform such tasks, the trajectory of the virtual linkage is specified:

$$r_d = \left(-\frac{9\pi}{500} \cos\left(\frac{\pi}{50}t\right), \frac{9\pi}{500} \sin\left(\frac{\pi}{50}t\right) \right) \tag{13}$$

Meanwhile, the attitude of the virtual linkage can be expressed as the angle from the x direction of the virtual link1 to the world coordination frame x direction and is also specified as a function of time

$$q_d = \frac{\pi}{50}t + \frac{\pi}{2} \tag{14}$$

Using the virtual linkage approach, the nine agents are able to maintain a line formation with a uniform distance of 0.1 m by specifying:

$$\Theta_d = \theta_1 = 0^\circ \tag{15}$$

$$\lambda_d = [\lambda_{1x}, \lambda_{1y}, \lambda_{2x}, \lambda_{2y}] = [0.4, 1, 0.4, 1] \tag{16}$$

Figure 8 shows the snapshots during the simulation for 100 s.

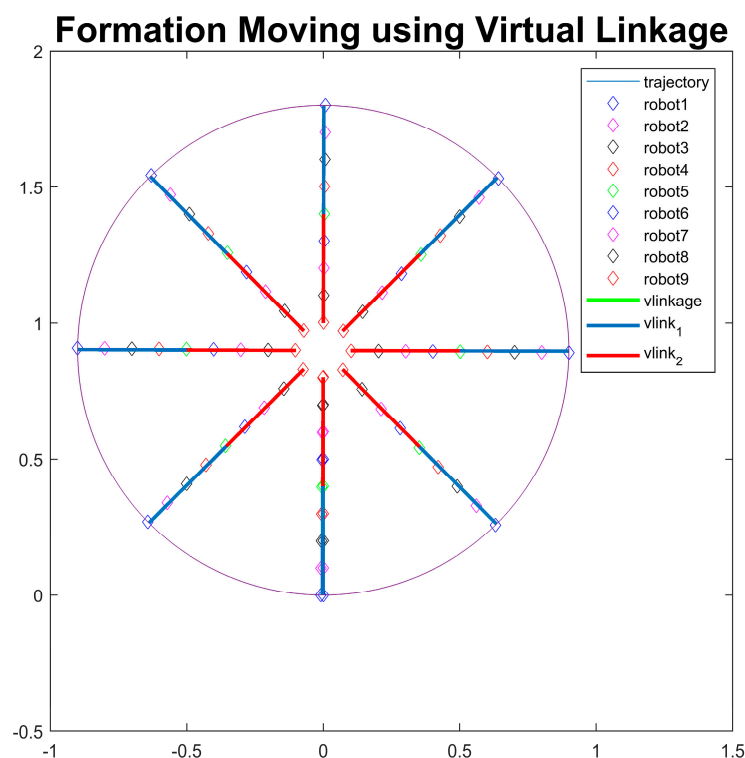


Figure 8. Snapshot of the formation moving using virtual linkage.

Figure 9a,b show the reference state and desired trajectory of the virtual linkage defined in Equations (13)–(16). The individual elements of each ξ_i are plotted in Figure 9c–h respectively. As can be seen from the figures, the nine random initialized $\xi_i (i = 1, 2, \dots, 9)$ finally converge into reference state defined in Equations (13)–(16). Recall the implication of virtual linkage states ξ_i illustrated in Section 3.2.1 in which the convergence of ξ_i indicates that the nine agents finally forms a specified virtual linkage and moves in desired formation shapes along the specified path. Therefore, the simulation results indicate the effectiveness of moving in formation using the virtual linkage approach.

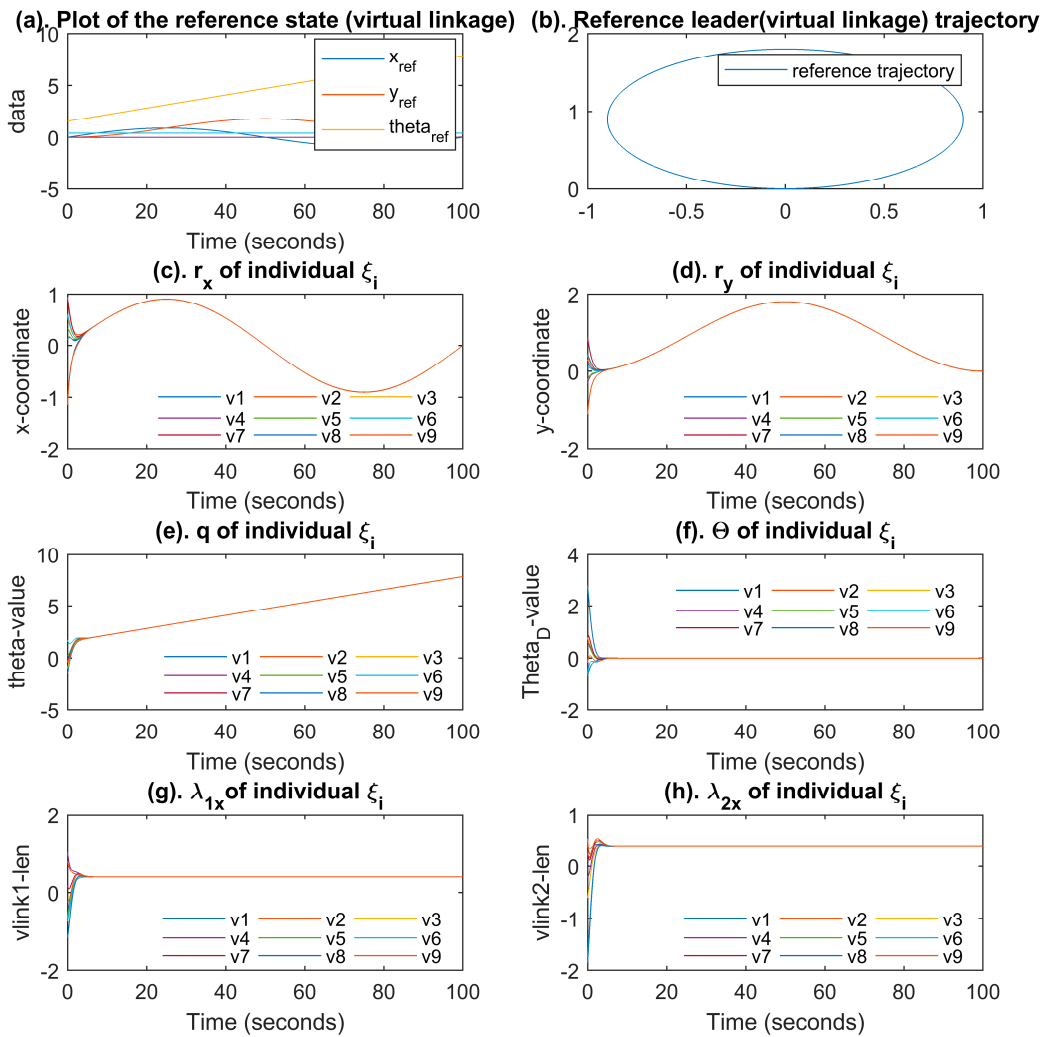


Figure 9. Simulation results of nine agents move around a circle in line formation.

4.3. Formation Reconfiguration Using Decentralized Virtual Linkage Approach

In this part, formation reconfiguration simulation is performed to show the virtual linkage approach’s reconfiguration ability by coordinating the desired virtual joint angle Θ_d and expansion/contraction vector λ_d .

In the original virtual linkage approach [18], the designed virtual linkage is able to present different formation shapes by coordinating the desired virtual joint angle Θ_d . Moreover, an expansion/contraction vector λ_d is introduced in this decentralized virtual linkage approach to enable the designed virtual linkage to be reconfigured into more formation patterns, as compared to the hierarchical virtual linkage approach. In this simulation, the group of agents is designed as the virtual linkage defined in Equations (9) and (10) and move along the trajectory in Equations (13) and (14). The Θ_d and λ_d are specified as Equations (17) and (18) to illustrate the formation reconfiguration ability.

$$\Theta_d = \theta_1 = \begin{cases} \frac{\pi}{100}t & t \leq 50 \\ -\frac{\pi}{100}t + \frac{\pi}{2} & 50 < t < 100 \end{cases} \quad (17)$$

$$\lambda_d = [\lambda_{1x}, \lambda_{1y}, \lambda_{2x}, \lambda_{2y}] = [0.4 - 0.002t, 1, 0.4 - 0.002t, 1] \quad (18)$$

Figure 10 shows the snapshots during the simulation for 100 s. As can be seen, the team of agents move around a circle with varying formation patterns. It is worth noting that the two virtual links

have different lengths at each moment during the simulation, which indicates that the introduction of λ_d has provided a stronger reconfiguration ability to the virtual linkage approach.

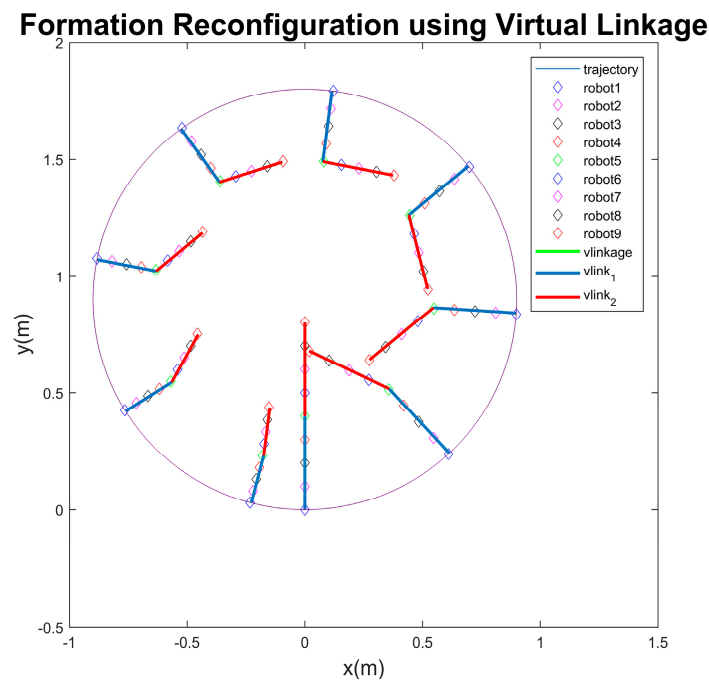


Figure 10. Snapshot of the formation reconfiguration using virtual linkage.

Figure 11a,b report the reference state and desired trajectory of the virtual linkage defined in Equations (13), (14), (17) and (18). The individual elements of each ζ_i are plotted in Figure 11c–h, respectively. As can be seen from the figures, the nine random initialized ζ_i finally converge into the same value, which indicates that the nine agents finally form a specified virtual linkage and move in desired formation shapes along the specified path. Notice that $\lambda_{1x}, \lambda_{2x}$ and Θ of each ζ_i track well with the varying function. Recall that a virtual linkage can reconfigure into different formation patterns by changing the joint angles and extracting/expanding each link with the scale factor λ_d . Thus, the simulation results indicate the effectiveness of formation reconfiguration using the proposed decentralized virtual linkage approach. What is important for us to recognize here, is the coordinate variable $\zeta_r = (r_d, q_d, \Theta_d, \lambda_d)$ in the proposed approach can be arbitrarily set, which provides great potential to perform complicated tasks. For example, when a group of agents is required to go through a gallery, the desired trajectory and varying formation shapes ($\zeta_r = (r_d, q_d, \Theta_d, \lambda_d)$) can be solved by plan method.

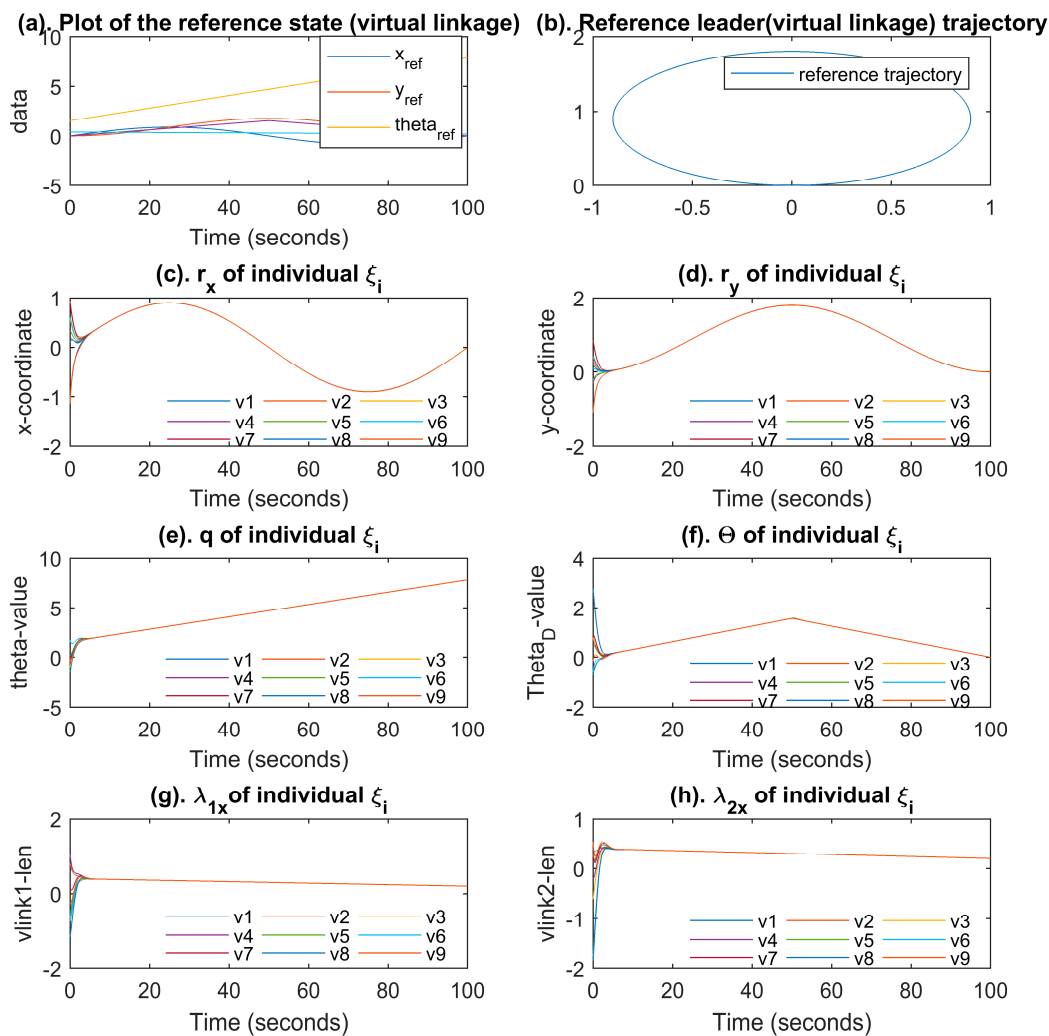


Figure 11. Simulation results formation reconfiguration using virtual linkage.

5. Comparison with Hierarchical Virtual Linkage Approach

In this section, the advantages of the proposed decentralized virtual linkage approach are illustrated through the comparison with the hierarchical virtual linkage approach and the performance is discussed.

5.1. Role Assignments

The detailed architecture of the hierarchical virtual linkage approach when performing the same simulation in Section 4.2 are presented in Figure 12. In the hierarchical virtual linkage approach, there exist virtual link/linkage servers which handle part of the formation control and communicate with part of agents. In reality, the virtual link/linkage server can be implemented in a specified agent, which implies there exist some agents (virtual link/linkage server) which need to perform the formation control computations. However, other agents only need to communicate with their corresponding virtual link server and move according to the states of the corresponding virtual link. In conclusion, there exist different kinds of agents which play different roles and need extra role assignments in the previous approach.

In the decentralized virtual linkage approach, there does not need to be extra role assignment in Figure 3 since all the agents run the same formation control algorithm in Equations (4) and (8).

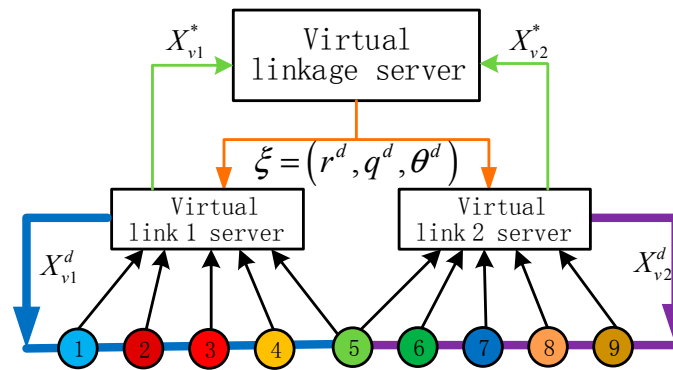


Figure 12. Detailed architecture of the hierarchical virtual linkage approach for simulation.

5.2. Scalability

As described in Section 3.1, each agent can completely decide its movement and only exchange states with its local neighbors. To verify the scalability of the proposed approach, five agents are required to align in a line formation with a uniform distance of 0.1 m and perform the same simulation task in Section 4.2 using the proposed decentralized virtual linkage approach.

To perform such a task, a virtual linkage which also consists of two virtual links is designed as follows:

$$Virtuallinkage_B = \begin{cases} virtuallink1 = \{agent1, agent2, agent3\} \\ virtuallink2 = \{agent3, agent4, agent5\} \end{cases} \tag{19}$$

$$\begin{aligned} v1 p_1^{Ini} &= [0, 0], v1 p_2^{Ini} = [0.1, 0], v1 p_3^{Ini} = [0.2, 0], \\ v2 p_3^{Ini} &= [0, 0], v2 p_4^{Ini} = [0.1, 0], v2 p_5^{Ini} = [0.2, 0] \end{aligned} \tag{20}$$

The extract/expansion rate is set as $\lambda_d = [\lambda_{1x}, \lambda_{1y}, \lambda_{2x}, \lambda_{2y}] = [1, 1, 1, 1]$. Figure 13 shows the designed virtual linkage and communication topologies used for this simulation. Figure 14 shows the snapshot during the simulation.

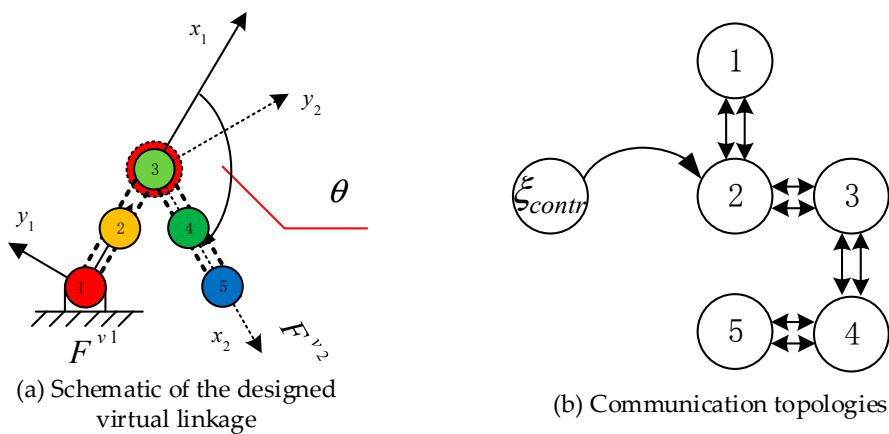


Figure 13. Schematic of designed virtual linkage and communication topologies.

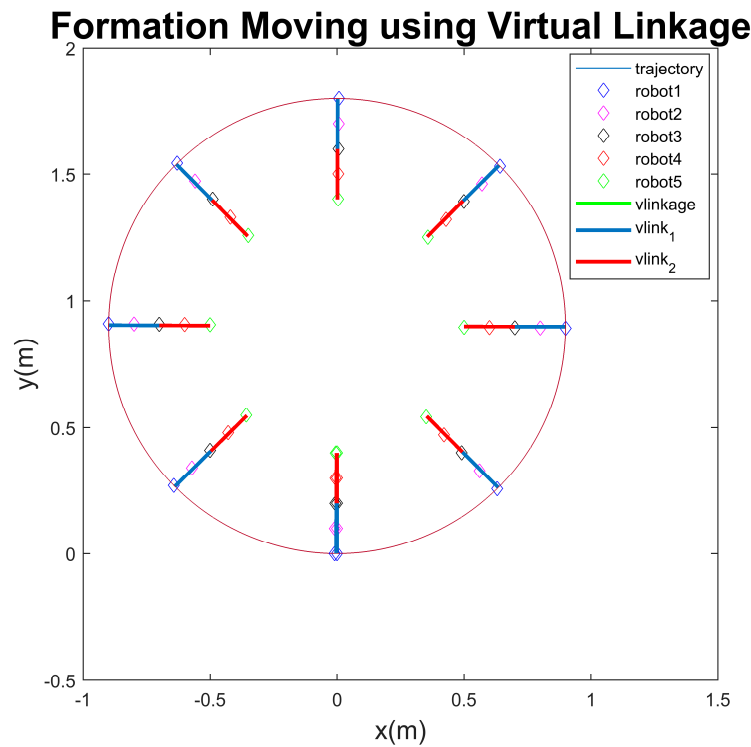


Figure 14. Snapshot of the formation moving of five agents.

Recall that in Figure 3, each agent exchanges its own understanding of virtual linkage states $\xi_i = (r_i, q_i, \Theta_i, \lambda_i)$ and the individual tracking error ${}^w p_i - {}^w p_i^d$ with its neighbors. Supposing $6K$ bits are required to encode each instantiation of virtual linkage ξ_i , $2K$ bits for individual tracking error ${}^w p_i - {}^w p_i^d$, then each agent only needs exchange these $8K$ bits message with its neighbors. If an agent has M neighbors and communicates with them for every period of time T and then the required bandwidth in units of bits per second is:

$$BW = 8K \cdot M/T \tag{21}$$

The maximum communication bandwidth BW_1 in this five agents involved simulation occurs in agent 2, since it has three neighbors (See Figure 13b):

$$BW_1 = 8K \cdot 3/T = 24K/T \tag{22}$$

Meanwhile, nine agents are involved in the simulation in Section 4.2, the maximum communication bandwidth also occurs in agent 2:

$$BW_2 = 8K \cdot 3/T = 24K/T = BW_1 \tag{23}$$

It is worth noting that, the maximum communication bandwidth in the team is determined by the maximum number of agents' neighbors and does not increase with the number of agents in the group.

In contrast to the decentralized virtual linkage approach, the hierarchical virtual linkage approach does not scale well with the number of agents. As illustrated in Reference [18], the maximum communication bandwidth increases with the number of agents in the team. The results indicate that the proposed decentralized virtual linkage approach is more suitable for situations when a large number of agents and/or limited communication are involved.

5.3. Reconfiguration Ability

As compared to the virtual structure approach, an important property of the hierarchical virtual linkage approach is that it can reconfigure the group of agents into different formation patterns by coordinating the value of virtual joint angles. To evaluate the reconfiguration ability of the virtual linkage approach, the definition of reconfiguration space is proposed:

Definition 3 (Reconfiguration Space). *In virtual linkage approach, the reconfiguration space is defined as the set of formation configuration that can be realized by the designed virtual linkage model. In other words, it corresponds to the formation patterns to which the designed virtual linkage can be reconfigured.*

To compare the reconfiguration ability between the original and proposed approach, the reconfiguration spaces of these two approaches are evaluated. In the proposed approach, instead of using $\zeta^d = (r^d, q^d, \Theta^d)$ to control the designed behavior of the team like the hierarchical virtual linkage approach (See Figure 12), $\zeta^d = (r^d, q^d, \Theta^d, \lambda^d)$ is used as the coordinate variable. Meanwhile, each agent i record its relative percentage position ${}^{vj}p_i^{Ini}$ in the corresponding virtual link j rather than the relative position ${}^{vj}p_i$. The benefit of this is that the virtual linkage is now able to extract/expansion the virtual link using Equation (6), which facilitates the reconfiguration ability of the proposed virtual linkage.

In detail, the reconfiguration spaces of the hierarchical and decentralized approaches are denoted as RS_{hier} and RS_{dec} respectively. Recall that ${}^d r, {}^d q$ are the predefined trajectory and attitude of the virtual linkage, the formation pattern is only determined by the vectors ${}^d \Theta = (\theta_1, \dots, \theta_n)$ and $\lambda_d = [\lambda_{1x}, \lambda_{1y}, \dots, \lambda_{nx}, \lambda_{ny}]$. Then the RS_{hier} and RS_{dec} can be expressed as vector spaces:

$$RS_{hier} = ({}^d \Theta, {}^d \lambda_{const}) = (\theta_1, \dots, \theta_n, 1, 1, \dots, 1, 1) \tag{24}$$

$$RS_{dec} = ({}^d \Theta, {}^d \lambda) = (\theta_1, \dots, \theta_n, \lambda_{1x}, \lambda_{1y}, \dots, \lambda_{nx}, \lambda_{ny}) \tag{25}$$

It is worth noting that RS_{hier} can be seen a special case of RS_{dec} when

$$\lambda_d = [\lambda_{1x}, \lambda_{1y}, \dots, \lambda_{nx}, \lambda_{ny}] = [1, 1, \dots, 1, 1] \tag{26}$$

and the decentralized approach has stronger reconfiguration ability than the hierarchical approach since

$$RS_{hier} \subset RS_{dec} \tag{27}$$

Figure 15 shows an intuitive way of illustrating the stronger reconfiguration ability by introducing the scale factor λ_d . In the original approach, the designed virtual linkage corresponds to a mechanical linkage which has fixed length and width and is only able to reconfigure into different shapes by coordinating the joint angles. However, proposed approach enables each virtual link to extract/expansion its length and width by introducing the scale factor λ_d , which allows the mechanical linkage to reconfigure into more shapes by extracting/expanding each link and changing the joint angles simultaneously.

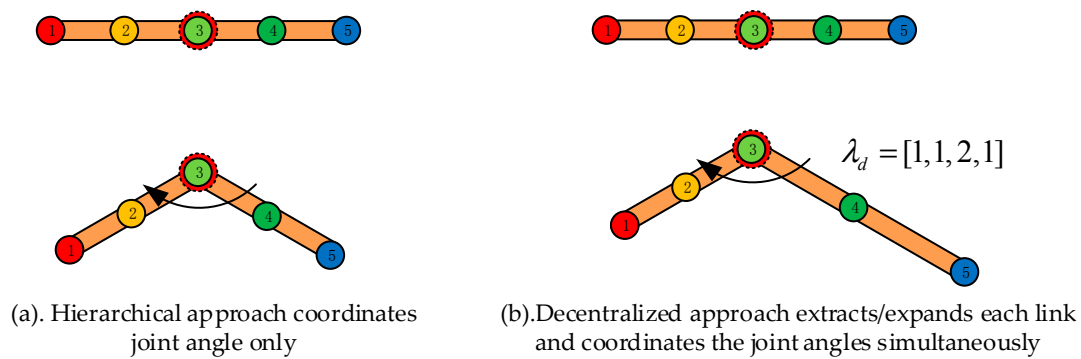


Figure 15. Reconfiguration ability of hierarchical and decentralized virtual linkage approach.

6. Conclusions

In this paper, the decentralized virtual linkage approach is presented. Instead of using a hierarchical architecture, as with the original virtual linkage approach, the proposed approach uses decentralized architecture in which each agent can independently determine its movement with only the exchange of state information with its local neighbors. The simulation results show the effectiveness of the decentralized virtual linkage approach. There are several advantages as compared with the original virtual linkage approach. First, the proposed approach uses decentralized architecture rather than hierarchical architecture, which does not require role assignments in each virtual link. In detail, there does not exist an agent which needs to exchange with all the agent members in the same virtual link. Meanwhile, each agent can completely decide its movement with only the exchange of states with its local neighbors, which makes this approach more suitable for situations when a large number of agents and/or limited communication are involved. Last but not least, the reconfiguration ability is enhanced in this approach by introducing the scale factor λ_d .

In future work, more attention will be paid to the formation feedback of the virtual linkage approach. Furthermore, the dynamical formation pattern generation and route planning for unknown environments is also an attractive direction and is worth researching.

Author Contributions: Y.L. conceived of the presented idea. Y.L. and J.G. developed the theoretical formalism. Y.L. and J.G. designed the simulation experiments and analyzed the data. Y.L., J.G., X.S. and C.J. discussed the results and wrote the paper.

Acknowledgments: This work is based on work supported in part by the National Key Technology R&D Program of China under Grant 2013BAK03B03 and National Defense Basic Research Project under Grant B2220132014.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van den Broek, T.H.A. Formation Control of Unicycle Mobile Robots: A Virtual Structure Approach. In Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference (International Conference), Shanghai, China, 15–18 December 2009; pp. 8328–8333.
2. Wang, W.; Huang, J.; Wen, C.; Fan, H. Distributed adaptive control for consensus tracking with application to formation control of nonholonomic mobile robots. *Automatica* **2014**, *50*, 1254–1263. [[CrossRef](#)]
3. Rezaee, H.; Abdollahi, F. A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots. *IEEE Trans. Ind. Electron.* **2014**, *61*, 347–354. [[CrossRef](#)]
4. Dong, X.; Yu, B.; Shi, Z.; Zhong, Y. Time-varying formation control for unmanned aerial vehicles: Theories and applications. *IEEE Trans. Control Syst. Technol.* **2015**, *23*, 340–348. [[CrossRef](#)]
5. Abbasi, Y.; Moosavian, S.A.A.; Novinzadeh, A.B. Formation control of aerial robots using virtual structure and new fuzzy-based self-tuning synchronization. *Trans. Inst. Meas. Control* **2017**, *39*, 1906–1919. [[CrossRef](#)]
6. Ren, W.; Beard, R. Decentralized Scheme for Spacecraft Formation Flying via the Virtual Structure Approach. *J. Guid. Control Dyn.* **2004**, *27*, 73–82. [[CrossRef](#)]

7. Qian, D.; Xi, Y. Leader–Follower Formation Maneuvers for Multi-Robot Systems via Derivative and Integral Terminal Sliding Mode. *Appl. Sci.* **2018**, *8*, 1045. [[CrossRef](#)]
8. Cui, R.; Ge, S.S.; How, B.V.E.; Choo, Y.S. Leader–follower formation control of underactuated autonomous underwater vehicles. *Ocean Eng.* **2010**, *37*, 1491–1502. [[CrossRef](#)]
9. Consolini, L.; Morbidi, F.; Prattichizzo, D.; Tosques, M. Leader-follower formation control of nonholonomic mobile robots with input constraints. *Automatica* **2008**, *44*, 1343–1349. [[CrossRef](#)]
10. Chen, J.; Sun, D.; Yang, J.; Chen, H. Leader-follower formation control of multiple non-holonomic mobile robots incorporating a receding-horizon scheme. *Int. J. Rob. Res.* **2010**, *29*, 727–747. [[CrossRef](#)]
11. Das, A.K.; Fierro, R.; Kumar, V.; Ostrowski, J.P.; Spletzer, J.; Taylor, C.J. A Vision-Based Formation Control Framework. *IEEE Trans. Robot. Autom.* **2002**, *18*, 813–825.
12. Balch, T.; Arkin, R.C. Behavior Based Formation Control for Multirobot Teams. *IEEE Trans. Robot. Autom.* **1998**, *14*, 926–939. [[CrossRef](#)]
13. Antonelli, G.; Arrichiello, F.; Chiaverini, S. Flocking for multi-robot systems via the Null-space-based behavioral control. *Swarm Intell.* **2010**, *4*, 37–56. [[CrossRef](#)]
14. Xu, D.; Zhang, X.; Zhu, Z.; Chen, C.; Yang, P. Behavior-based formation control of swarm robots. *Math. Probl. Eng.* **2014**, *2014*. [[CrossRef](#)]
15. Lewis, M.A.; Tan, K.H. High Precision Formation Control of Mobile Robots Using Virtual Structures. *Auton. Robots* **1997**, *4*, 387–403. [[CrossRef](#)]
16. Dong, L.; Chen, Y.; Qu, X. Formation Control Strategy for Nonholonomic Intelligent Vehicles Based on Virtual Structure and Consensus Approach. *Procedia Eng.* **2016**, *137*, 415–424. [[CrossRef](#)]
17. Chen, L.; Baoli, M. A nonlinear formation control of wheeled mobile robots with virtual structure approach. In Proceedings of the 2015 34th Chinese Control Conference (CCC), Hangzhou, China, 28–30 July 2015; pp. 1080–1085.
18. Liu, Y.; Gao, J.; Liu, C.; Zhao, F.; Zhao, J. Reconfigurable Formation Control of Multi-Agents Using Virtual Linkage Approach. *Appl. Sci.* **2018**, *8*, 1109. [[CrossRef](#)]
19. Ren, W.; Randal, B. *Distributed Consensus in Multi-Vehicle Cooperative Control*; Springer: London, UK, 2008.
20. Ren, W. Decentralization of Virtual Structures in Formation Control of Multiple Vehicle Systems via Consensus Strategies. *Eur. J. Control* **2008**, *14*, 93–103. [[CrossRef](#)]
21. Ren, W. *Consensus Seeking, Formation Keeping, and Trajectory Tracking in Multiple Vehicle Cooperative Control*; Brigham Young University: Provo, UT, USA, 2004.
22. Ren, W.; Sorensen, N. Distributed coordination architecture for multi-robot formation control. *Rob. Auton. Syst.* **2008**, *56*, 324–333. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).