Decentralized Flocking Control with a Minority of Informed Agents

Hung Manh La, Ronny Salim Lim, Weihua Sheng
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK 74078, USA
Email: (hung.la, ronny.lim, weihua.sheng)@okstate.edu

Heping Chen
Ingram School of Engineering
Texas State University
San Marcos, TX 78666, USA
Email: hc15@txstate.edu

***Abstract*—In this paper we study the flocking control in the case of a small subset of informed agents. In nature, only few agents in a group have the information of the target, such as knowledge about the location of a food source, or the migration route. However, they can still flock together in a group based on local information. Inspired by this natural phenomenon, a flocking control algorithm is designed to coordinate the motion of multiple agents. Based on our algorithm, all agents can form a network, maintain connectivity and track the target even only very few of them know the information of the target. The experiments and simulations are performed to demonstrate the effectiveness of the proposed algorithm.**

Keywords: Flocking control, multi-agent systems.

I. Introduction

Flocking is a phenomenon in which a number of agents move together and interact with each other. In nature, schools of fish, birds, ants, and bees, etc. demonstrate the phenomena of flocking. Flocking control for multiple mobile agents has been studied in recent years [1], [2], [3], and it is designed based on three basic flocking rules proposed by Reynolds in [4]: flock centering (agents try to stay close to nearby flock-mates), collision avoidance (agents try to avoid collision with nearby flock-mates), and velocity matching (agents try to match their velocity with nearby flock-mates). The problems of flocking have also attracted many researchers in physics [5] and biology [6].

Early work on flocking control includes [1], [2], [3]. Tanner *et al.* [1], [2] studied flocking control of a system of multiple mobile agents with double integrator dynamics in the case of fixed and dynamic topologies. In [3], the theoretical framework for design and analysis of distributed flocking algorithm was proposed. This established a foundation for flocking control design for a group of agents. As an extension of the flocking algorithm in [3], flocking control of agents with a virtual leader in the case of a minority of informed agents and varying velocity of virtual leader was presented in [7]. However, in their work the network can not maintain its connectivity because some agents may fall out of the network.

In this paper we study how to utilize a minority of informed agents to lead the whole network to track the target while maintaining the connectivity. The main differences with the above related work are:

1. We adopt a target navigation term in order to reduce the large tracking force at the initial tracking time so that the connectivity is maintained.

2. We use a damping force term to reduce the tracking overshoot.

Overall, we propose a new flocking control algorithm that allows the flock to preserve connectivity, avoid collision, and track the target without overshooting. We demonstrate that by applying our algorithm the agents can flock together and maintain connectivity better compared to existing flocking control algorithms.

The rest of this paper is organized as follows. In the next section we present the background of flocking control and the problem formulation. Section III describes the flocking control algorithm with a minority of informed agents. Section IV shows the experimental and simulation results. Finally, Section V concludes this paper.

II. Flocking Backgrounds and Problem Formulation

In this section we present flocking control background and the problem formulation.

A. Flocking Control Backgrounds

We consider n agents moving in an m (e.g., $m = 2,3$) dimensional Euclidean space. The dynamic equations of each agent are described as:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i, \quad i = 1,2,...,n. \end{cases} \tag{1}$$

here $q_i, p_i \in R^m$ are the position and velocity of node i, respectively, and u_i is the control input of agent i.

To describe the topology of flocks we consider a dynamic graph G consisting of a vertex set $\vartheta = \{1,2...,n\}$ and an edge set $E \subseteq \{(i,j) : i,j \in \vartheta, j \neq i\}$. In this topology each vertex denotes one member of the flock, and each edge denotes the communication link between two members.

We know that during the movement of agents, the relative distance between them may change, hence the neighbors of each agent also change. Therefore, we can define a neighborhood set of agent i as follows:

$$N_i = \left\{ j \in \vartheta : \|q_j - q_i\| \leq r,\ \vartheta = \{1,2,...,n\}, j \neq i \right\},$$

here r is an active range (radius of neighborhood circle in the case of two dimensions, $m = 2$, or radius of neighborhood sphere in the case of three dimensions, $m = 3$), and $\|.\|$ is the Euclidean distance.

The geometry of flocks is modeled by an α-lattice [3] that meets the following condition:

$$\|q_j - q_i\| = d, j \in N_i, \quad (2)$$

here d is a positive constant indicating the distance between agent i and its neighbor j. However, at singular configuration $(q_i = q_j)$ the collective potential used to construct the geometry of flocks is not differentiable. Therefore, the set of algebraic constrains in (2) is rewritten in term of σ - norm [3] as follows:

$$\|q_j - q_i\|_\sigma = d^\alpha, j \in N_i, \quad (3)$$

here the constraint $d^\alpha = \|d\|_\sigma$ with $d = r/k_c$, where k_c is the scaling factor. The σ norm, $\|.\|_\sigma$, of a vector is a map $R^m \Longrightarrow R_+$ defined as $\|z\|_\sigma = \frac{1}{\varepsilon}[\sqrt{1+\varepsilon\|z\|^2} - 1]$ with $\varepsilon > 0$. Unlike the Euclidean norm $\|z\|$, which is not differentiable at $z = 0$, the σ norm $\|z\|_\sigma$, is differentiable everywhere. This property allows to construct a smooth collective potential function for agents.

The flocking control law in [3] controls all agents to form an α-lattice configuration. This algorithm consists of two components as follows:

$$u_i = f_i^\alpha + f_i^t. \quad (4)$$

The first component of (4) f_i^α, which consists of a gradient-based component and a consensus component, is used to regulate the potentials (repulsive or attractive forces) and the velocity among agents,

$$f_i^\alpha = \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + \sum_{j \in N_i} a_{ij}(q)(p_j - p_i), \quad (5)$$

where each term in (5) is computed as follows [3]:

1. The action function $\phi_\alpha(z)$ that vanishes for all $z \geq r^\alpha$ with $r^\alpha = \|r\|_\sigma$ is defined as follows:

$$\phi_\alpha(z) = \rho_h(z/r_\alpha)\phi(z - d^\alpha)$$

with the uneven sigmoidal function $\phi(z)$ defined as $\phi(z) = 0.5[(a+b)\sigma_1(z+c) + (a-b)]$, here $\sigma_1(z) = z/\sqrt{1+z^2}$, and parameters $0 < a \leq b$, $c = |a-b|/\sqrt{4ab}$ to guarantee $\phi(0) = 0$. The bump function $\rho_h(z)$ with $h \in (0,1)$ is

$$\rho_h(z) = \begin{cases} 1, & z \in [0,h) \\ 0.5[1+cos(\pi(\frac{z-h}{1-h}))], & z \in [h,1) \\ 0, & otherwise. \end{cases}$$

2. The vector along the line connecting q_i to q_j is

$$n_{ij} = (q_j - q_i)/\sqrt{1+\varepsilon\|q_j - q_i\|^2}.$$

3. The elements $a_{ij}(q)$ of the adjacency matrix $[a_{ij}(q)]$ are defined as

$$a_{ij}(q) = \begin{cases} \rho_h(\|q_j - q_i\|_\sigma / r_\alpha), & if \quad j \neq i \\ 0, & if \quad j = i. \end{cases}$$

The pairwise attractive/repulsive potential $\Psi_\alpha(z)$ is defined as:

$$\Psi_\alpha(z)) = \int_{d_\alpha}^{z} \phi_\alpha(s) ds.$$

This function is illustrated in Figure 1.

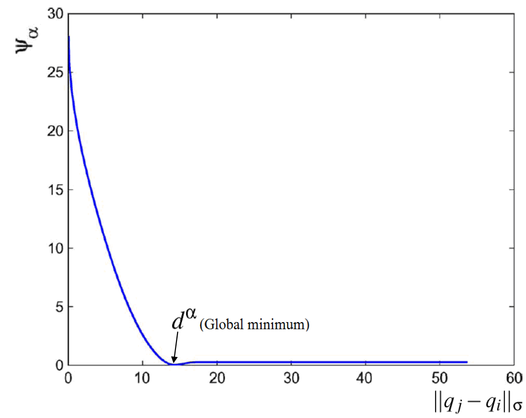

Fig. 1. Smooth pairwise potential function $\Psi_\alpha(\|q_j - q_i\|_\sigma)$.

The second component of (4) f_i^t is designed for distributed target tracking,

$$f_i^t = -c_1^t(q_i - q_t) - c_2^t(p_i - p_t) \quad (6)$$

where c_1^t and c_2^t are positive constants, and (q_t, p_t) are the position and velocity of the stationary or moving target defined as follows

$$\begin{cases} \dot{q}_t = p_t \\ \dot{p}_t = f_t(q_t, p_t) \end{cases}$$

Finally, the Olfati-Saber flocking control law [3] in free space is:

$$\begin{aligned} u_i \quad = \quad & \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + \sum_{j \in N_i} a_{ij}(q)(p_j - p_i) \\ & -c_1^t(q_i - q_t) - c_2^t(p_i - p_t). \end{aligned} \quad (7)$$

B. Problem Formulation

The control law (7) is designed under the assumption that all agents need information of the position and velocity of the target in order to flock together. However, in reality this assumption is not valid. It can be seen in many cases that only very few agents have information of the target due to their limited sensing range. For example, in fish schools and bird flocks, only some agents have knowledge about the location of a food source, or of a migration route [8], [9]. Motivated by these observations we will study how to design a distributed flocking control algorithm which can still maintain good tracking performance and connectivity when only few agents have information of the target.

III. Decentralized Flocking Control with a Minority of Informed Agents

In this section, we design a distributed flocking control algorithm for multi-agent systems in the case that only a few agents are informed with the position and velocity of the target. We call these agents as informed agents. Let us define N_I as a subset of informed agents and N_{UI} as a subset of uninformed agents with $N_I << N_{UI}$. Hence we have $N_I \cup N_{UI} = N$, here N is the set of all agents (uninformed and informed agents).

Paper [7] proposed the following flocking control algorithm based on the algorithm (7):

$$\begin{aligned} u_i &= \sum_{j\in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma)n_{ij} + \sum_{j\in N_i} a_{ij}(q)(p_j - p_i) \\ &\quad -c_1^t(q_i - q_t)I_i - c_2^t(p_i - p_t)I_i. \end{aligned} \tag{8}$$

here if $I_i = 1$ the agent i has information (position and velocity) of the target. Otherwise $I_i = 0$ agent i does not have information of the target.

We implemented the algorithm (8) in which we let some agents closest to the target have the information (position and velocity) of the target. The result is shown in Figure 2.

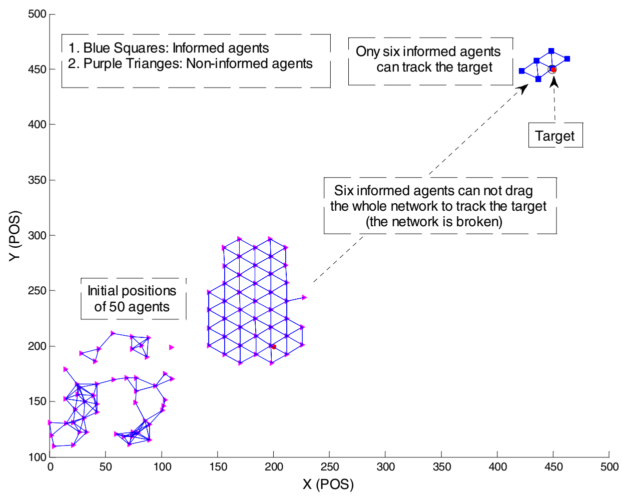

Fig. 2. Snapshots of the agents when applying the flocking control algorithm (8). We select 6 out of 50 agents which are closest to the target to have the information (position and velocity) of the target.

In this figure we clearly see that the network is broken, and only the agents which have information of the target can track the target. Additionally, we find that the target tracking performance has big overshoot. In order to solve these two problems we introduce two terms. The first term is a navigation term, and the second one is a damping force term. The main purpose of the navigation term is to maintain the connectivity among agents, and the purpose of the damping force term is to reduce the tracking overshoot.

A. Navigation Term

The navigation term allows the agents to stay together. The main idea behind this term is that if we let the informed agents keep strong cohesion to uninformed agents at the initial time of the target tracking process, the connectivity can be maintained. In order to do this, we have to reduce the initial momentum of the attractive force to the target for the informed agents. This means that we should have small attractive force at the initial time when the distance between the informed agent and the target is large. Based on this analysis we design the navigation term as shown in Algorithm 1. In this algorithm the constant K_1 chosen between 0.9 and 1 is to ensure that a small attractive force is applied at the initial time of the target tracking process. The weights, $\frac{K_2}{\|q_i^{inf}(t)-q_t(t)\|}$ and $\frac{K_3}{\|q_i^{inf}(t)-q_t(t)\|}$ are designed so that the attractive force is small enough at the initial time, and then it becomes bigger when the distance $\|q_i^{inf}(t) - q_t(t)\|$ decreases.

Algorithm 1: Design of the Navigation Term

for *each informed agent* j*,* $j \in N_I$ **do**
  if $\|q_i^{inf}(t) - q_t(t)\| > K_1\|q_i^{inf}(0) - q_t(0)\|$ **then**

$$\begin{aligned} f_j^t &= -\frac{K_2}{\|q_i^{inf}(t) - q_t(t)\|}(q_j^{inf} - q_t) \\ &\quad -\frac{K_3}{\|q_i^{inf}(t) - q_t(t)\|}(p_j^{inf} - p_t) \end{aligned}$$

    here, $0.9 < K_1 < 1$, $K_2 > 0$ and $K_3 > 0$,
  else

$$f_i^t = -c_1^t(q_j^{inf} - q_t) - c_2^t(p_j^{inf} - p_t)$$

  end
end

Algorithm 2: Design of the Damping Force Term

for *each informed agent* j*,* $j \in N_I$ **do**
  if $\|q_i^{inf}(t) - q_t(t)\| < K_4 r$ **then**

$$f_j^{dam} = -K_{dam} p_j^{inf}$$

    here, $0 < K_4 < 1$ and $K_{dam} > 0$,
  else

$$f_j^{dam} = 0$$

  end
end

B. Damping Force Term

Since only the informed agents N_I have the information of the target, the damping force can be only applied to these agents. The idea behind this damping force is to reduce the tracking overshoot when the informed agents are close to the target. That is, the damping force for the informed agents is only effective if the distance between the informed agent and the target is less than a certain threshold. This threshold is designed based on the active range r. This means that when the target is inside the active range of the informed agent j the damping force f_j^{dam} is applied, otherwise $f_j^{dam} = 0$. In order to do that the constant K_4 is used $(0 < K_4 < 1)$. When the damping force f_j^{dam} is applied, the informed agent j will reduce its speed gradually to approach the target. Hence, the tracking overshoot is reduced. Overall, the damping force is designed in Algorithm 2.

Finally the whole decentralized flocking control algorithm is proposed in Algorithm 3. In this algorithm we have 2 options of the initial network configuration, and both options are to allow the network of agents to be connected at the beginning.

IV. Experimental and Simulation Results

In this section we test our proposed flocking control algorithm 3 and compare it with the existing flocking control algorithm (8) in the case of a minority of informed agents. First we test our algorithm with 7 real robots. Then to show the effectiveness and the scalability of our algorithm we test it with 50 robots in simulation. In addition, we show a metric to evaluate the network connectivity of our algorithm and the existing algorithm.

A. Experimental setup

In this experiment we use 7 Rovio robots [10] that have omni-directional motion capability. Basically, these robots can freely move in 6 directions. The dynamic model of the Rovio robot can be approximated by Equation (1). However, the accuracy of the localization of the Rovio robot is low, and the robot does not have any sensing device to sense the pose (position and velocity) of its neighbors or the obstacles. Hence we use a VICON motion capture system [11] in our lab (Figure 3) that includes 12 cameras to track objects. This tracking system can give the location and velocity of each moving object with over 95 percent of accuracy. We use the following parameters:

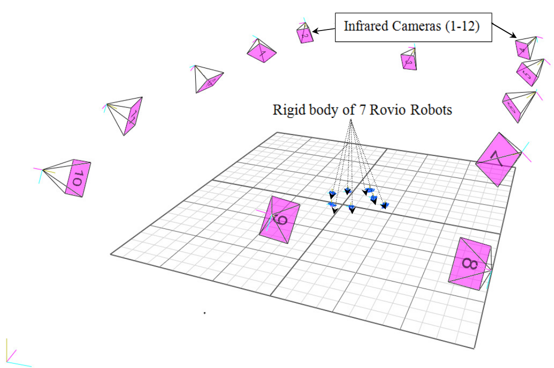

Fig. 3. Motion Capture System from VICON [11] for experimental setup.

- Parameters of flocking: $a = b = 5$; $d = 600mm$; the scaling factor $k_c = 1.2$; the active range $r = k_c.d$; $\epsilon = 0.1$ for the σ norm; h = 0.2 for the bump function ($\phi_\alpha(z)$); h = 0.9 for the other bump function ($\phi_\beta(z)$).

- Parameters of the target: The target location is at $[0,\ 500mm]$. The velocity vector $p_t = [5,\ 5]$.

B. Simulation setup

In the simulation 50 agents are randomly distributed in the square area of 120 × 120 size, and we use the following parameters:

Algorithm 3: Decentralized Flocking Control Algorithm with a Minority of Informed Agents

---

Input : - Position and velocity of each agent ($q_i,\ p_i$).
- Position and velocity of the target ($q_t,\ p_t$) for the informed agents (N_I).
Output: Control law for each agent u_i

Initialization phase:

- **Option 1.** Randomly deploy the agents so that they can form a connected network;
- **Option 2.** All agents are programmed based on flocking algorithm (7) to go to the rendezvous point so that they can form a connected network;

Implementation phase:

for *each agent i* **do**
  Compute:

$$f_i^\alpha = \sum_{j \in N_i} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + \sum_{j \in N_i} a_{ij}(q)(p_j - p_i),$$

end
for *each informed agent j, $j \in N_I$* **do**
  if $\|q_i^{inf}(t) - q_t(t)\| > K_1 \|q_i^{inf}(0) - q_t(0)\|$ **then**

$$f_j^t = -\frac{K_2}{\|q_i^{inf}(t) - q_t(t)\|}(q_j^{inf} - q_t) - \frac{K_3}{\|q_i^{inf}(t) - q_t(t)\|}(p_j^{inf} - p_t)$$

  $(0.9 < K_1 < 1,\ K_2 > 0$ and $K_3 > 0)$
  else

$$f_i^t = -c_1^t(q_j^{inf} - q_t) - c_2^t(p_j^{inf} - p_t)$$

  end
  if $\|q_i^{inf}(t) - q_t(t)\| < K_4 r$ **then**

$$f_j^{dam} = -K_{dam} p_j^{inf}$$

  $(0 < K_4 < 1$ and $K_{dam} > 0)$
  else

$$f_j^{dam} = 0$$

  end
end
for *each uninformed agent k, $k \in N_{UI}$* **do**

$$f_k^{dam} = 0$$

$$f_k^t = 0$$

end

Update the control law for each agent i:

$$u_i = f_i^\alpha + f_i^{dam} + f_i^t$$

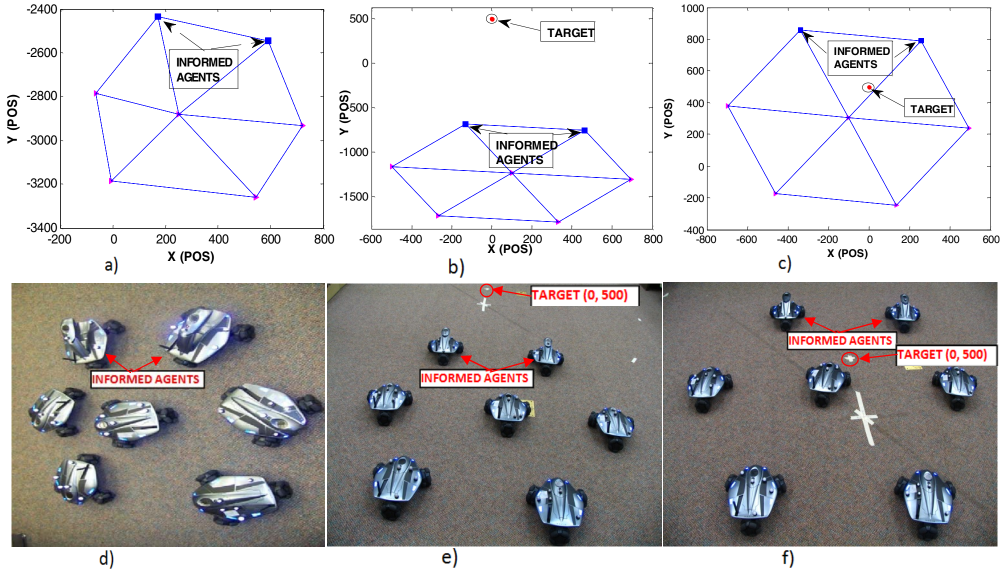

Fig. 4. Snapshots of 7 Rovio robots flocking together when applying our proposed flocking control Algorithm 3.

- Parameters of flocking: $a = b = 5$; $d = 16$ units; the scaling factor $k_c = 1.2$; the active range $r = k_c * d$; $\varepsilon = 0.1$ for the σ norm; h = 0.2 for the bump function ($\phi_\alpha(z)$); h = 0.9 for the other bump function ($\phi_\beta(z)$).

- Parameters of the target: The target location is at $[450,\ 450]$. The velocity vector $p_t = [5,\ 5]$.

C. Network's connectivity evaluation

To evaluate the network connectivity maintenance, first we know that the link (connectivity) between node i and node j is maintained if the distance $0 < \|q_i - q_j\| \leq r$. Otherwise this link is considered broken. For graph connectivity, a dynamic graph $G(\vartheta, E)$ is connected at time t if there exists a path between any two vertices.

Based on the above analysis, to analyze the connectivity of the network we define a connectivity matrix $[c_{ij}(t)]$ as follows:

$$[c_{ij}(t)] = \begin{cases} 1, & if \quad j \in N_i(t), i \neq j \\ 0, & if \quad j \notin N_i(t), i \neq j \end{cases}$$

and $c_{ii} = 0$. Since the rank of Laplacian [3] of a connected graph $[c_{ij}(t)]$ of order n is at most $(n-1)$ or $rank([c_{ij}(t)]) \leq (n-1)$, the relative connectivity of a network at time t is defined as: $C(t) = \frac{1}{n-1} rank([c_{ij}(t)])$. If $0 \leq C(t) < 1$ the network is broken, and if $C(t) = 1$ the network is connected. Based on this metric we can evaluate the network connectivity in our proposed flocking control Algorithm 3 and the existing flocking control algorithm (8).

D. Experimental Results

Initially, the seven Rovio robots are randomly deployed so that they can form a connected network (see Option 1 in Algorithm 3). Then, two robots which are closest to the target point are selected to be the informed agents (two robots have cameras facing up as shown in snapshot (d) in Figure 4). We obtained the results of our flocking control Algorithm 3 in

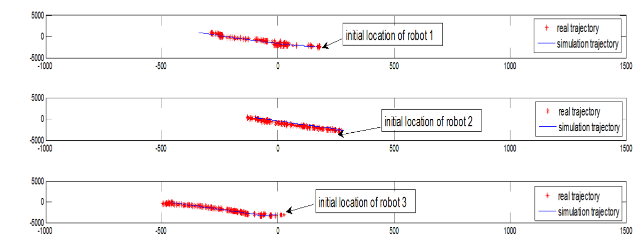

Fig. 5. Trajectories of simulation and real robots when applying our proposed Algorithm 3.

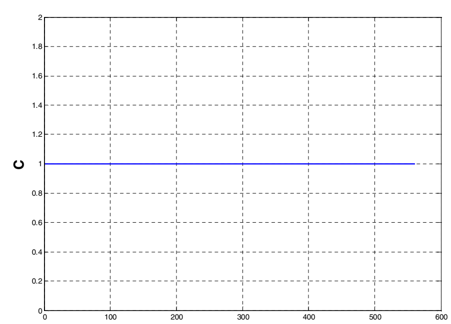

Fig. 6. Connectivity evaluation in experiment of 7 Rovio robots when applying our proposed Algorithm 3.

Figures 4, 5 and 6. Specially, Figure 4 (a, b, c) show the snapshots of simulation results for seven robots, and Figure 4 (d, e, f) show the snapshots of experiment results for seven robots. In Figure 5 we compare the trajectories of three out of seven robots in both simulation and experiment, and we see that the experimental trajectories have small difference with the ones in simulation since the motion of the robots is limited to only six directions. In addition, Figure 6 shows the connectivity result, and we clearly see that the seven robots can flock together even only two of them know the information of the target.

E. Simulation Results

In simulation, we test our proposed Algorithm 3 with fifty robots which are randomly deployed so that they do not form a connected network initially. Then, these robots are programed based on the flocking algorithm (7) to go to the rendezvous point (see Option 2 in Algorithm 3). This step is to make sure that the fifty robots form a connected network at the rendezvous point. After that we let two robots (blue squares) which are closest to the target know the position and velocity of the target. By observing Figure 7 we can see that the two informed robots can drag all 48 other robots (purple triangles) to flock together. The connectivity for the proposed Algorithm 3 and the algorithm (8) is shown in Figure 9, and from this figure we can see that the connectivity is maintained for Algorithm 3 while it is broken when applying algorithm (8). The tracking overshoot is evaluated in Figure 8, and we see that without the damping force term the tracking overshoot is big, and the network oscillates around the target.

For more details about the simulation and experiment results please see some video files available at the following website: *http : //ascc.okstate.edu/projectshung.html*

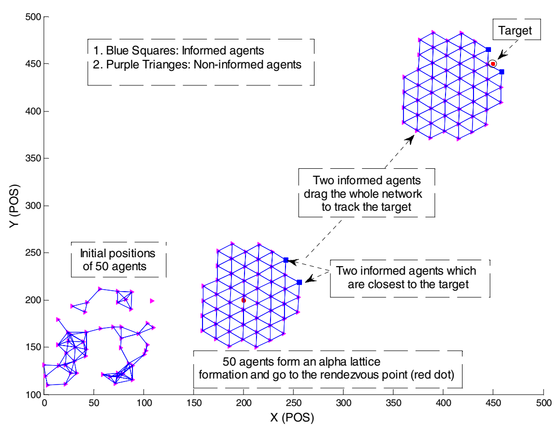

Fig. 7. Snapshots of 50 robots flocking together (simulation) with two of them knowing the information of the target.

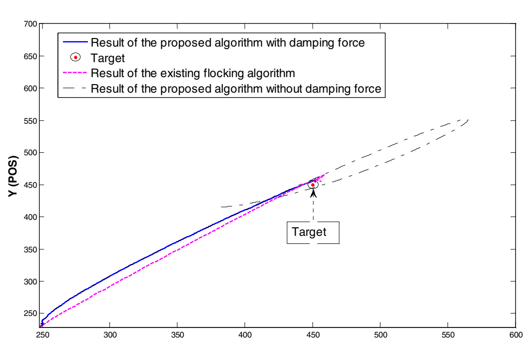

Fig. 8. Average of positions of 2 informed robots (tracking performance) in simulation.

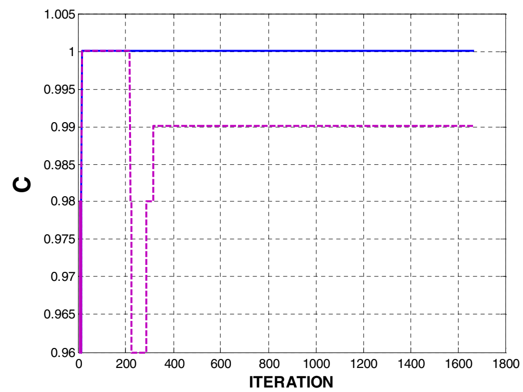

Fig. 9. Connectivity evaluation in simulation of 50 robots. Solid line is for our proposed Algorithm 3, and dash line is for the existing algorithm (8).

V. Conclusion

In this paper, we considered the behavior of a group of agents when only a subset of them have the information of the target. We proposed a decentralized flocking control algorithm to deal with the network partition and reduce the tracking overshoot. Our algorithm is based on adding the target navigation term and the damping force term. As a result, the network connectivity preservation is improved, and the overshoot is eliminated. Both simulation and experimental results were collected to demonstrate the effectiveness of our proposed flocking control algorithm.

Acknowledgment

This project is supported by the DoD ARO DURIP grant 55628-CS-RIP, USA, the MOET (Ministry of Education and Training) program, Vietnamese Government, and the Center for Telecommunications and Network Security, Oklahoma State University.

References

[1] H. G. Tanner, A. Jadbabai, and G. J. Pappas. Stable flocking of mobile agents, part I: fixed topology. *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2010–2015, 2003.
[2] H. G. Tanner, A. Jadbabai, and G. J. Pappas. Stable flocking of mobile agents, part II: dynamic topology. *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2016–2021, 2003.
[3] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.
[4] C. Reynolds. Flocks, birds, and schools: A distributed behavioral model. *Computer Graphics,ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California*, 21(4):25–34, 1987.
[5] H. Levine, W. J. Rappel, and I. Cohen. Self-organization in systems of self-propelled particles. *Phys. Review. E*, 63:017101–017104, 2000.
[6] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *J. Theor.Biol*, 218:1–11, 2002.
[7] H. Su, X. Wang, and Z. Lin. Flocking of multi–agents with a virtual leader. *IEEE Transactions on Automatic Control*, 54(2):293–307, 2009.
[8] I. D. Couzin, J. Krause, N. R. Franks, and S. A. Levin. Effective leadership and decision making in animal groups on the move. *Letters to Nature*, 433(3):513–516, 2005.
[9] S. G. Reebs. Can a minority of informed leaders determine the foraging movements of a fish shoal? *Animal Behavior*, 59:403–409, 2000.
[10] Rovio robot: http://www.wowwee.com/en/support/rovio.
[11] VICON motion capture system: http://www.vicon.com/.