

# Multi-Agent Motion Control in Cluttered and Noisy Environments

Hung Manh La (+), Weihua Sheng (++)

+: Center for Advanced Infrastructure and Transportation, Rutgers University, Piscataway, NJ 08854, USA

++: School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA

Emails: hung.la11@rutgers.edu, weihua.sheng.@okstate.edu

**Abstract**—Birds, bees, and fish often flock together in groups to find the source of food (target) based on local information. Inspired by this natural phenomenon, flocking control algorithms are designed to coordinate the activities of multiple agents in cluttered and noisy environments, respectively. First, to allow agents to track and observe the target better in cluttered environments, two new approaches are proposed to control the center of mass (CoM) of positions and velocities of all mobile agents in the network (*Single-CoM*), and the center of mass of positions and velocities of each agent and its neighbors (*Multi-CoM*), respectively. With these approaches, the flock can better track the target. Second, to deal with noisy measurements we proposed two flocking control algorithms, *Multi-CoM-Shrink* and *Multi-CoM-Cohesion*. Based on these algorithms, all agents can form a network and maintain connectivity, even with noisy measurements. We also investigate the stability of our algorithms. The numerical experimental tests are performed to demonstrate the effectiveness of the proposed approach.

**Keywords:** Flocking control, Dynamic target tracking, Multi-agent systems, Mobile agent networks

## 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], [4], and it is designed based on three basic flocking rules proposed by Reynolds in [5]: 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 [6], [7], mathematics [8], biology [9] and especially in control science in recent years [4], [10], [11], [12], [13], [14], [1], [2], [3], [15], [16], [17].

Flocking control has wide applications in mobile robots and mobile sensor networks. Early works on flocking control includes [1], [2], [3], [4]. Tanner *et al.* [1], [2], [3] studied the stability properties of a system of multiple mobile agents with double integrator dynamics in the case

of fixed and dynamic topologies. However, in their work the target tracking problem and sensing errors are not considered. In the context of target tracking, Olfati-Saber [4] proposed a theoretical framework for design and analysis of distributed flocking algorithms. These algorithms solve the flocking problem in free space and in the presence of obstacles. Based on his flocking control algorithm, all agents can flock together and track the target quite well in free space. However, the target tracking performance is poor in the obstacle space. Specifically, the target is not at the center of the flock. Moreover, every agent is assumed to know the position and velocity of the target precisely. To relax this assumption, a distributed Kalman filter was developed in [18] for each agent to estimate the target's position. In addition, the flocking algorithm in [4] assumes that all agents have the information of the target in order to maintain cohesion and avoid fragmentation. To solve this problem, Su *et al.* [10], [11], [19] extended Olfati-Saber's flocking control algorithm [4] to deal with the situations of a minority of informed agents and varying velocity of the target. However, their work does not consider the tracking problem in cluttered and noisy environments.

In this paper we propose new flocking control algorithms for more realistic environments. The main differences between our algorithms and those of the above related work are:

1. In cluttered environments, the agents usually get stuck behind the obstacles and sometimes can not follow the target [4]. To handle this problem we present new approaches for multi-agent systems to track a moving target while avoiding obstacles. The main motivation of these approaches is to make the CoM (Center of Mass) of the network track the moving target better in cluttered environments where the traditional flocking control algorithms [4], [18], [10], [11], [19] have poor tracking performance. In our methods all mobile agents can surround the target closely in the obstacle space. This will allow the network to observe and recognize the target better. Specifically, in our *Single-CoM* algorithm, the center of mass of positions and velocities of all mobile agents in the network is controlled to track a moving target. This algorithm works well in small networks, but it has limited scalability in large networks. In contrast with the *Single-CoM* algorithm, we proposed another flocking control algorithm called *Multi-CoM* where the center of

---

This work is supported by the MOET (Ministry of Education and Training) under the 322 project, Vietnamese Government, the DoD (Department of Defense) ARO DURIP grant 55628-CS-RIP, USA, and the NSF (National Science Foundation) MRI project CISE/CNS MRI 0923238, USA.

mass of positions and velocities of each agent and its neighbors is controlled to track a moving target. This algorithm allows agents to work better in large networks in a distributed fashion.

2. In real flocking control environments, noise handling is always an important issue since the noise usually causes broken network or connectivity loss. This problem exists in most of the previous work on flocking control [3], [4], [19], [18]. To make the flocking control more applicable in real environments we consider the effect of position and velocity measurement errors of the agent itself, the agent's neighbors and the target. None of the flocking control algorithms in the above related work considers this noise issue. We propose two flocking control algorithms, *Multi-CoM-Shrink* and *Multi-CoM-Cohesion*, which are based on the extensions of the *Multi-CoM* flocking control algorithm. Our algorithms allow the flock to preserve connectivity, avoid collision, and follow the target in such noisy environments. We demonstrate that by applying our algorithms the agents can flock together in the presence of noise with better connectivity and tracking performance.

The rest of this paper is organized as follows. In the next section we present the flocking control algorithms, *Single-CoM* and *Multi-CoM*, respectively, for tracking and observing a moving target in cluttered environments. Section III presents flocking control algorithms, *Multi-CoM-Shrink* and *Multi-CoM-Cohesion*, respectively, for tracking a moving target in noisy environments. Section IV shows the main results on stability analysis of flocking control algorithms in both cluttered and noisy environments. Section V demonstrates the experimental results. Finally, Section VI concludes this paper.

## II. FLOCKING CONTROL IN CLUTTERED ENVIRONMENTS

In this section we present the flocking control algorithms in cluttered environments.

### A. Flocking Control Background

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

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

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  $\vartheta = \{1, 2, \dots, n\}$  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:

$$N_i^\alpha = \{j \in \vartheta : \|q_j - q_i\| \leq r, \vartheta = \{1, 2, \dots, n\}, j \neq i\}, \quad (2)$$

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  $\|\cdot\|$  is the Euclidean distance. The superscript  $\alpha$  indicates the actual neighbors ( $\alpha$  neighborhood agents) of agent  $i$  that is used to distinguish from virtual neighbors ( $\beta$  neighborhood agents) in the case of obstacle avoidance discussed later.

The geometry of flocks is modeled by an  $\alpha$ -lattice [4] that meets the following condition:

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

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 (3) is rewritten in term of  $\sigma$  - norm [4] as follows:

$$\|q_j - q_i\|_\sigma = d^\alpha, j \in N_i^\alpha, \quad (4)$$

here the constraint  $d^\alpha = \|d\|_\sigma$  with  $d = r/k_c$ , where  $k_c$  is the scaling factor. The  $\sigma$  - norm,  $\|\cdot\|_\sigma$ , of a vector is a map  $R^m \implies R_+$  defined as

$$\|z\|_\sigma = \frac{1}{\epsilon} [\sqrt{1 + \epsilon \|z\|^2} - 1], \quad (5)$$

here  $\epsilon > 0$ . Unlike the Euclidean norm  $\|z\|$ , which is not differentiable at  $z = 0$ , the  $\sigma$  - norm  $\|z\|_\sigma$ , is differentiable every where. This property allows to construct a smooth collective potential function for agents.

The flocking control law in [4] controls all agents to form an  $\alpha$ -lattice configuration. This algorithm consists of three components as follows:

$$u_i = f_i^\alpha + f_i^\beta + f_i^\gamma. \quad (6)$$

The first component of (6)  $f_i^\alpha$ , which consists of a gradient-based component and a consensus component (more details about these components see [20], [21], [22]), is used to regulate the potentials (repulsive or attractive forces) and the velocity among agents.

$$f_i^\alpha = c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i), \quad (7)$$

where  $c_1^\alpha$  and  $c_2^\alpha$  are positive constants, and each term in (7) is computed as follows [4]:

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

$$\phi_\alpha(z) = \rho_h(z/r_\alpha) \phi(z - d^\alpha) \quad (8)$$

with the uneven sigmoidal function  $\phi(z)$  defined as  $\phi(z) = 0.5[(a+b)\sigma_1(z+c) + (a-b)]$ , here  $\sigma_1(z) = z/\sqrt{1+z^2}$  ( $z$  is an arbitrary variable), 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

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ 0.5[1 + \cos(\pi(\frac{z-h}{1-h}))], & z \in [h, 1) \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

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

$$n_{ij} = (q_j - q_i) / \sqrt{1 + \epsilon \|q_j - q_i\|^2}. \quad (10)$$

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

$$a_{ij}(q) = \begin{cases} \rho_h(\|q_j - q_i\|_\sigma / r_\alpha), & \text{if } j \neq i \\ 0, & \text{if } j = i. \end{cases} \quad (11)$$

The second component of Equation (6)  $f_i^\beta$  is used to control the mobile agents to avoid obstacles,

$$f_i^\beta = c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q) (\hat{p}_{i,k} - p_i) \quad (12)$$

where  $c_1^\beta$  and  $c_2^\beta$  are positive constants, and the set of  $\beta$  neighbors (virtual neighbors) of agent  $i$  at time  $t$  with  $k$  obstacles is

$$N_i^\beta(t) = \left\{ k \in \vartheta_\beta : \|\hat{q}_{i,k} - q_i\| \leq r', \vartheta_\beta = \{1, 2, \dots, k\} \right\} \quad (13)$$

here  $r'$  is selected to be less than  $r$ , in our simulations  $r' = 0.6r$ .  $\vartheta_\beta$  is a set of obstacles.  $\hat{q}_{i,k}, \hat{p}_{i,k}$  are the position and velocity of agent  $i$  projecting on the obstacle  $k$ , respectively. The virtual neighbors are used to generate the repulsive force to push the agents away from the obstacles.

Similar to vector  $n_{ij}$  defined in Equation (10), vector  $\hat{n}_{i,k}$  is defined as

$$\hat{n}_{i,k} = (\hat{q}_{i,k} - q_i) / \sqrt{1 + \epsilon \|\hat{q}_{i,k} - q_i\|^2}. \quad (14)$$

The elements  $b_{i,k}(q)$  of the adjacency matrix  $[b_{i,k}(q)]$  are defined as

$$b_{i,k}(q) = \rho_h(\|\hat{q}_{i,k} - q_i\|_\sigma / d_\beta) \quad (15)$$

where  $d_\beta = \|r'\|_\sigma$ .

The repulsive action function of  $\beta$  neighbors is defined as

$$\phi_\beta(z) = \rho_h(z/d_\beta)(\sigma_1(z - d_\beta) - 1). \quad (16)$$

The third component of (6)  $f_i^\gamma$  is a distributed navigational feedback.

$$f_i^\gamma = -c_1^\gamma(q_i - q_\gamma) - c_2^\gamma(p_i - p_\gamma) \quad (17)$$

where  $c_1^\gamma$  and  $c_2^\gamma$  are positive constants, and the  $\gamma$ -agent  $(q_\gamma, p_\gamma)$  is the virtual leader (more information of virtual leader, see [23]) defined as follows

$$\begin{cases} \dot{q}_\gamma = p_\gamma \\ \dot{p}_\gamma = f_\gamma(q_\gamma, p_\gamma) \end{cases} \quad (18)$$

## B. Single-CoM and Multi-CoM flocking Control Algorithms

In environments populated by obstacles, the agents hardly follow the target because of repulsive forces generated from the obstacles, and this causes poor tracking performance. To address this issue, two possible approaches, named *Single-CoM* and *Multi-CoM*, respectively, are investigated. In the *Single-CoM* algorithm, the CoM of positions and velocities of all agents is controlled to track the moving target. In this case, each agent need know the position and velocity of all other agents. To address the scalability problem the *Multi-CoM* (CoM of positions and velocities of each agent and its neighbors) algorithm is proposed, where each agent only need know the positions and velocity of its neighbors.

1) *Single-CoM tracking*: Firstly, based on Olfati-Saber's flocking algorithm (6) we slightly modify it with a dynamic  $\gamma$ -agent. In this scenario, the dynamic  $\gamma$ -agent is considered as the moving target.

$$u_i = c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q) (p_j - p_i) + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q) (\hat{p}_{i,k} - p_i) - c_1^t (q_i - q_t) - c_2^t (p_i - p_t) \quad (19)$$

here the pair  $(q_t, p_t)$  is the position and velocity of the moving target, respectively, and  $c_1^t, c_2^t$  are positive constants, and  $c_2^t = 2\sqrt{c_1^t}$ .

By observing the control protocol (19), we see that the CoM is difficult to reach the target in the presence of obstacles since the agents are sometimes stuck behind the obstacles. This causes the poor tracking performance. Therefore this protocol should be extended with more constraint on the CoM as follows:

$$u_i = f_i^\alpha + f_i^\beta + f_i^t \quad (20)$$

where  $f_i^t$  is a tracking feedback term applied to agent  $i$  by a moving target with position and velocity  $(q_t, p_t)$ , respectively.

$$f_i^t = -c_1^t (q_i - q_t) - c_2^t (p_i - p_t) - c_1^{sc} (\bar{q} - q_t) - c_2^{sc} (\bar{p} - p_t) \quad (21)$$

here  $c_1^{sc}, c_2^{sc}$  are positive constants. The pair  $(\bar{q}, \bar{p})$  is the center of mass (CoM) of positions and velocities of all agents defined in (22).

$$\begin{cases} \bar{q} = \frac{1}{n} \sum_{i=1}^n q_i \\ \bar{p} = \frac{1}{n} \sum_{i=1}^n p_i, \end{cases} \quad (22)$$

here  $\bar{q}$ , and  $\bar{p}$  are also called the global average of position and velocity, respectively.

Consequently, the extended control protocol (20) is as follows:

$$\begin{aligned}
 u_i = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} \\
 & + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \\
 & + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} \\
 & + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i) \\
 & - c_1^t(q_i - q_t) - c_2^t(p_i - p_t) \\
 & - c_1^{sc}(\bar{q} - q_t) - c_2^{sc}(\bar{p} - p_t) \quad (23)
 \end{aligned}$$

In control algorithm (23), each mobile agent at each time  $t$  need know the position and velocity of all other agents to compute the CoM  $(\bar{q}, \bar{p})$ . This means that the scalability is limited because at each time  $t$  all other agents have to send their positions to agent  $i$ .

2) *Multi-CoM tracking*: To make the algorithm scalable we implement a distributed tracking algorithm called *Multi-CoM tracking* in which the CoM of each agent and its neighbors is controlled to track the target. Hence, we design the tracking term  $f_i^t$  as

$$\begin{aligned}
 f_i^t = & -c_1^t(q_i - q_t) - c_2^t(p_i - p_t) \\
 & -c_1^l(\bar{q}_i - q_t) - c_2^l(\bar{p}_i - p_t), \quad (24)
 \end{aligned}$$

here  $(c_1^l, c_2^l)$  are the positive constants.  $\bar{q}_i$  and  $\bar{p}_i$  are the local averages of position and velocity of agent  $i$  and its neighbors, respectively. They are defined as

$$\begin{cases} \bar{q}_i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} q_j \\ \bar{p}_i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} p_j. \end{cases} \quad (25)$$

here  $|N_i^\alpha \cup \{i\}|$  is the number of agents in agent  $i$ 's local neighborhood including agent  $i$  itself. Since the local averages are the local CoMs we name this algorithm as *Multi-CoM*.

Consequently, the *Multi-CoM* flocking control algorithm is proposed as

$$\begin{aligned}
 u_i = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} \\
 & + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \\
 & + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} \\
 & + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i) \\
 & - c_1^t(q_i - q_t) - c_2^t(p_i - p_t) \\
 & - c_1^l(\bar{q}_i - q_t) - c_2^l(\bar{p}_i - p_t) \quad (26)
 \end{aligned}$$

In control algorithm (26), each mobile agent only need have local knowledge about the position and velocity of itself and its neighbors. Therefore this algorithm can scale up to lager mobile agent networks.

### III. FLOCKING CONTROL OF MULTIPLE AGENTS IN NOISY ENVIRONMENTS

The above flocking control algorithms are designed under the following assumptions: each agent can sense the position and velocity of itself, the neighbors and the target precisely. However, in reality these assumptions are not valid because sensing always has noise. Motivated by this observation we study how to design distributed flocking control algorithms which can still perform well when the measurements are corrupted by noise.

In this section we are going to design two algorithms in noisy environments. The first one is the *Multi-CoM-Shrink* flocking control algorithm. The main idea of this algorithm is to shrink the size of the network in order to keep the connectivity. The second one is the *Multi-CoM-Cohesion* flocking control algorithm, and its main idea is based on the position and velocity cohesion feedback to create the strong cohesion between the agent and the network. Both algorithms are based on the *Multi-CoM* flocking control algorithm presented in Section II.

#### A. Multi-CoM-Shrink Algorithm

Assume that the estimates of the position and velocity of agent  $i$  are:  $\hat{q}_i = q_i + \epsilon_q^i$  and  $\hat{p}_i = p_i + \epsilon_p^i$ , where  $\epsilon_q^i$  and  $\epsilon_p^i$  are the position and velocity measurement errors, respectively. Then we have:

$$\begin{aligned}
 \hat{q}_i - \hat{q}_j &= q_i - q_j + \epsilon_q^{ij}; \hat{p}_i - \hat{p}_j = p_i - p_j + \epsilon_p^{ij}, \\
 \text{here } \epsilon_q^{ij} &= \epsilon_q^i - \epsilon_q^j \text{ and } \epsilon_p^{ij} = \epsilon_p^i - \epsilon_p^j.
 \end{aligned}$$

Similarly, the estimates of the position and velocity of the target are:

$$\hat{q}_t = q_t + \epsilon_q^t \text{ and } \hat{p}_t = p_t + \epsilon_p^t,$$

where  $\epsilon_q^t$  and  $\epsilon_p^t$  are the position and velocity measurement errors, respectively. Then we have:

$$\begin{aligned}
 \hat{q}_i - \hat{q}_t &= q_i - q_t + \epsilon_q^{it}; \hat{p}_i - \hat{p}_t = p_i - p_t + \epsilon_p^{it}, \\
 \text{here } \epsilon_q^{it} &= \epsilon_q^i - \epsilon_q^t \text{ and } \epsilon_p^{it} = \epsilon_p^i - \epsilon_p^t.
 \end{aligned}$$

If all noises are bounded, one possible method to maintain connectivity in noisy environments is to shrink the size of the network. We assume that the noise  $\epsilon_q^i$  satisfies  $\|\epsilon_q^i\| \leq r_w$  as shown in Figure 1.

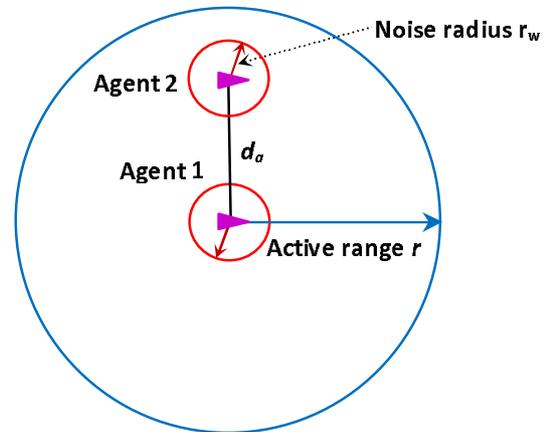


Figure 1. Agent 2 is considered as a neighbor of agent 1 because the estimated distance  $\hat{d}_a$  is less than the active range  $r$ .

Let us denote  $d_a = \|q_i - q_j\|$  to be the actual distance between agent  $i$  and agent  $j$ . Then to maintain the connectivity and no collision among agents we need

$$0 < d_a \leq r. \quad (27)$$

Denote  $\hat{d}_a$  to be the estimate of the actual distance  $d_a$ , then we have

$$\hat{d}_a = \|\hat{q}_i - \hat{q}_j\| \leq \|q_i - q_j\| + \|\epsilon_q^{ij}\|. \quad (28)$$

Since  $\|\epsilon_q^i\| \leq r_w$  we have  $\|\epsilon_q^{ij}\| \leq 2r_w$ , and we obtain

$$\|q_i - q_j\| - 2r_w \leq \hat{d}_a \leq \|q_i - q_j\| + 2r_w. \quad (29)$$

With  $\|q_i - q_j\| = d_a$  we have

$$d_a - 2r_w \leq \hat{d}_a \leq d_a + 2r_w, \quad (30)$$

or,

$$\hat{d}_a - 2r_w \leq d_a \leq \hat{d}_a + 2r_w. \quad (31)$$

Since the control algorithm (19) guarantees that  $\hat{d}_a$  converges to the desired distance  $d$ . Then from (31) we obtain

$$d - 2r_w \leq d_a \leq d + 2r_w. \quad (32)$$

From (27) and (32) we should have

$$\begin{cases} d - 2r_w > 0 \\ d + 2r_w \leq r. \end{cases} \quad (33)$$

Hence from (33) we obtain  $d$  to be

$$2r_w < d \leq r - 2r_w. \quad (34)$$

Equation (34) shows that we need to design the distance  $d$  within the range  $(2r_w, r - 2r_w]$  to maintain connectivity and no collision among agents. However if we select  $d$  to be smaller than  $r - 2r_w$  then each agent will have more neighbors than necessary. Hence, we choose  $d = r - 2r_w$ .

Now, from (5) we obtain  $d_{new}^\alpha$  as

$$d_{new}^\alpha = \|d\|_\sigma = \frac{1}{\epsilon} [\sqrt{1 + \epsilon(r - 2r_w)^2} - 1]. \quad (35)$$

From (8) we obtain a new action function  $\phi_\alpha^{new}(\|\hat{q}_j - \hat{q}_i\|_\sigma)$  as follows:

$$\phi_\alpha^{new}(\|\hat{q}_j - \hat{q}_i\|_\sigma) = \rho_h(\|\hat{q}_j - \hat{q}_i\|_\sigma / r_\alpha) \phi(\|\hat{q}_j - \hat{q}_i\|_\sigma - d_{new}^\alpha). \quad (36)$$

From (25) we have the local average of position and velocity for each agent  $i$ ,  $\hat{q}_i$  and  $\hat{p}_i$  with noise computed as

$$\begin{cases} \hat{q}_i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} \hat{q}_j \\ \hat{p}_i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} \hat{p}_j, \end{cases} \quad (37)$$

From (10) and (11) we obtain  $\hat{n}_{ij}$  and  $\hat{a}_{ij}(q)$  as

$$\hat{n}_{ij} = (\hat{q}_j - \hat{q}_i) / \sqrt{1 + \epsilon \|\hat{q}_j - \hat{q}_i\|^2} \quad (38)$$

$$\hat{a}_{ij}(q) = \begin{cases} \rho_h(\|\hat{q}_j - \hat{q}_i\|_\sigma / r_\alpha), & \text{if } j \neq i \\ 0, & \text{if } j = i, \end{cases} \quad (39)$$

Now, we propose a *Multi-CoM-Shrink* algorithm with  $d_{new}^\alpha$  as

$$\begin{aligned} u_i = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha^{new}(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \\ & + c_2^\alpha \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q) (\hat{p}_j - \hat{p}_i) \\ & - c_1^t (\hat{q}_i - \hat{q}_t) - c_2^t (\hat{p}_i - \hat{p}_t) \\ & - c_1^l (\hat{q}_i - \hat{q}_t) - c_2^l (\hat{p}_i - \hat{p}_t). \end{aligned} \quad (40)$$

## B. Multi-CoM-Cohesion Algorithm

In this subsection we describe the *Multi-CoM-Cohesion* algorithm, which introduces local position and velocity cohesion feedbacks to each agent.

We have the following definitions:

$d_{il} = q_i - \bar{q}_i$  is the relative distance between node  $i$  and its local average of position;

$v_{il} = p_i - \bar{p}_i$  is the relative velocity between node  $i$  and its local average of velocity;

However, because agent  $i$  senses its own position and velocity with noise, hence the estimates  $\hat{d}_{il}$  and  $\hat{v}_{il}$  are also corrupted by noise ( $\epsilon_d^i, \epsilon_v^i$ ) as:

$$\begin{cases} \hat{d}_{il} = \hat{q}_i - \hat{\bar{q}}_i = q_i + \epsilon_q^i - (\bar{q}_i + \bar{\epsilon}_q^i) = d_{il} + \epsilon_d^i \\ \hat{v}_{il} = \hat{p}_i - \hat{\bar{p}}_i = p_i + \epsilon_p^i - (\bar{p}_i + \bar{\epsilon}_p^i) = v_{il} + \epsilon_v^i, \end{cases} \quad (41)$$

here  $\epsilon_d^i = \epsilon_q^i - \bar{\epsilon}_q^i$  with  $\bar{\epsilon}_q^i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{i=1}^{|N_i^\alpha \cup \{i\}|} \epsilon_q^i$ ,

and  $\epsilon_v^i = \epsilon_p^i - \bar{\epsilon}_p^i$  with  $\bar{\epsilon}_p^i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{i=1}^{|N_i^\alpha \cup \{i\}|} \epsilon_p^i$ .

Based on the above definitions, we design a distributed flocking control law, *Multi-CoM-Cohesion*, in noisy environments as:

$$\begin{aligned} u_i = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \\ & + c_2^\alpha \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q) (\hat{p}_j - \hat{p}_i) \\ & - c_{pos} \hat{d}_{il} - c_{ve} \hat{v}_{il} \\ & - c_1^t (\hat{q}_i - \hat{q}_t) - c_2^t (\hat{p}_i - \hat{p}_t) \\ & - c_1^l (\hat{q}_i - \hat{q}_t) - c_2^l (\hat{p}_i - \hat{p}_t), \end{aligned} \quad (42)$$

here  $\hat{d}_{il}$ ,  $\hat{v}_{il}$  are the estimates of  $d_{il}$  and  $v_{il}$ , respectively, and  $c_{pos}$  and  $c_{ve}$  are positive constants. The terms  $-c_{pos} \hat{d}_{il}$  and  $-c_{ve} \hat{v}_{il}$  are called local position and velocity cohesion feedbacks, respectively. The role of these negative feedbacks is to maintain position and velocity cohesions. This means that each agent tries to stay close to the local average of position and minimize the velocity mismatch between its velocity and the local average of velocity in noisy environments.

In this algorithm, to make it simpler in the stability analysis provided later we dropped the obstacle avoidance term. However, in real applications, to allow each agent to avoid both static and dynamic obstacles we only need to add the second component (12) to the control algorithm (42). In general, this component does not affect the properties of the global stability of the whole system.

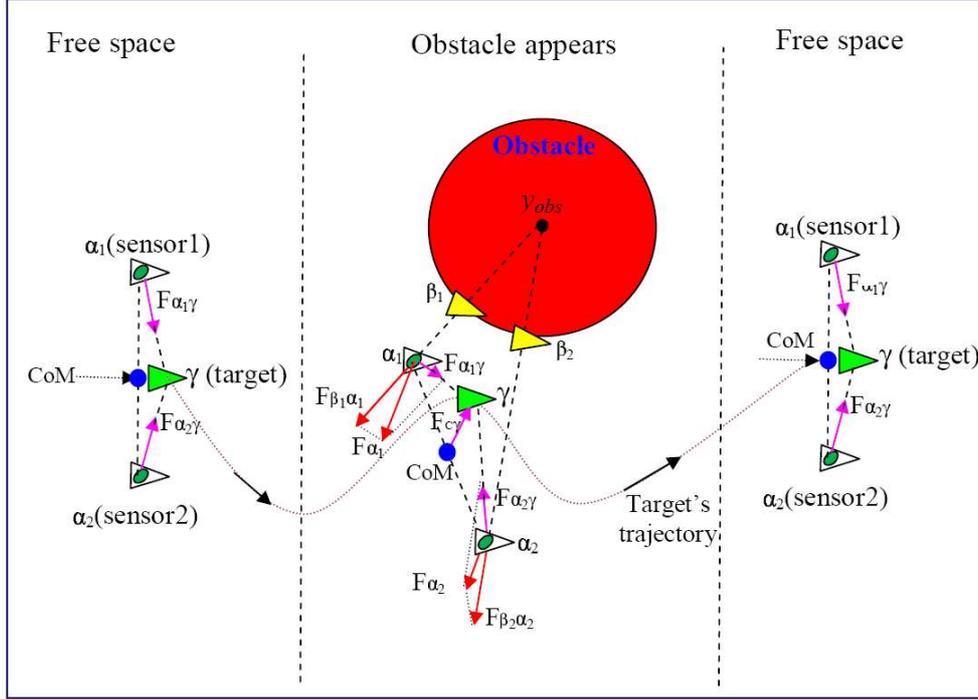


Figure 2. Demonstration of two agents (2 mobile sensors) tracking the moving target with Single-CoM in both free and obstacle spaces.

#### IV. STABILITY ANALYSIS

##### A. Stability Analysis of Flocking in Cluttered Environments

In this sub-section we will analyze the stability of our algorithms, the flocking control with *Single-CoM* and *Multi-CoM*, respectively, in cluttered environments, and we will explain why the tracking performance in the presence of the CoM constraint is better than without the CoM constraint in the obstacle space.

**Theorem 1.** In cluttered environments, consider a system of  $n$  mobile agents where each agent has dynamics (1) and is controlled by (23). Then the following statements hold:

1. The CoM of positions and velocities of all agents in the network will exponentially converge to the target in the free space.
2. The error between the CoM's position and the target's position is reduced in the obstacle space.

*Proof:*

*Proof of part 1:* In free space,  $\sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) = 0$ , hence we can rewrite the control algorithm (23) by ignoring constants  $c_\eta^\nu$  (for  $\forall \eta = 1, 2$  and  $\nu = \alpha, \beta$ ) as follows:

$$\begin{aligned} u_i = & - \sum_{j \in N_i^\alpha} \nabla_{q_i} \psi_\alpha(\|q_j - q_i\|_\sigma) \\ & + \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \\ & - c_1^t(q_i - q_t) - c_2^t(p_i - p_t) \\ & - c_1^{sc}(\bar{q} - q_t) - c_2^{sc}(\bar{p} - p_t). \end{aligned} \quad (43)$$

where  $\psi_\alpha(z) = \int_{d_\alpha}^z \phi_\alpha(s) ds$  is the pairwise attractive/repulsive potential function. From (43), we can com-

pute the average of the control law  $u$  as follows:

$$\begin{aligned} \bar{u} = \frac{1}{n} \sum_{i=1}^n u_i = & \frac{1}{n} \sum_{i=1}^n \left( - \sum_{j \in N_i^\alpha} \nabla_{q_i} \psi_\alpha(\|q_j - q_i\|_\sigma) \right. \\ & + \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \\ & - (c_1^t + c_1^{sc})(\bar{q} - q_t) \\ & \left. - (c_2^t + c_2^{sc})(\bar{p} - p_t) \right). \end{aligned} \quad (44)$$

Obviously, we see that the pair  $(\psi_\alpha, a(q))$  is symmetric. Hence we can rewrite (44) as:

$$\bar{u} = - (c_1^t + c_1^{sc})(\bar{q} - q_t) - (c_2^t + c_2^{sc})(\bar{p} - p_t) \quad (45)$$

Equation (45) implies that

$$\begin{cases} \dot{\bar{q}} = \bar{p} \\ \dot{\bar{p}} = - (c_1^t + c_1^{sc})(\bar{q} - q_t) - (c_2^t + c_2^{sc})(\bar{p} - p_t). \end{cases} \quad (46)$$

The solution of (46) indicates that the position and velocity of the CoM will exponentially converge to those of the target.

*Proof of part 2:* To see why the tracking performance in the presence of obstacles of the flocking control with *Single-CoM* is better than that of the flocking control without CoM (*No-CoM*) (19), we analyze the forces acting on the  $\alpha$ -agents (actual agents) when they avoid the obstacle as shown in Figure 2. In this figure, without losing generality we simply consider two agents tracking the target ( $\gamma$ -agent) which moves along an arbitrary trajectory.

Firstly, when two agents track the target in the free space (without obstacle), in the equilibrium state the CoM is close to the target (in this case, agent 2 is the neighbor

of agent 1). The total interaction forces between two agents are equal to zero, and also because of velocity matching, the sum of different velocities between these agents is equal to zero. Hence we obtain  $\bar{u}$  as in (45). This means that the CoM  $(\bar{q}, \bar{p})$  converges to the target  $(q_t, p_t)$ .

When these agents move in the obstacle space they project themselves to the surface of obstacle and get their virtual neighbors,  $\beta$ -agents. In this scenario, two  $\beta$ -agents,  $\beta_1$  and  $\beta_2$ , of two agent 1 and agent 2 are created, respectively (see Figure 2). These  $\beta$ -agents generate the repulsive forces,  $F_{\beta_1\alpha_1}$  and  $F_{\beta_2\alpha_2}$  to push these agents away from the obstacles. However, the presence of  $\gamma$ -agent (target) is necessary to steer both agents around the obstacle. We have the synthesized forces (vector form)  $\vec{F}_{\alpha_1} = \vec{F}_{\beta_1\alpha_1} + \vec{F}_{\alpha_1\gamma}$  and  $\vec{F}_{\alpha_2} = \vec{F}_{\beta_2\alpha_2} + \vec{F}_{\alpha_2\gamma}$  of agent 1 and agent 2, respectively. To prevent each agent from hitting the obstacles, the weights of the repulsive forces,  $c_1^\beta$  and  $c_2^\beta$ , are set to be bigger than those of the attractive force between the target and each agent,  $c_1^t$  and  $c_2^t$ . This leads to  $F_{\beta_1\alpha_1} > F_{\alpha_1\gamma}$  and  $F_{\beta_2\alpha_2} > F_{\alpha_2\gamma}$ . Therefore, this causes the agents being pushed away in a certain distance from the target, or the CoM is no longer close to the target. When the CoM considered as a virtual agent is directly controlled to track the target, the offset distance between the CoM and the target,  $\|\bar{q} - q_t\|$ , creates the negative feedback to the whole system then it makes the CoM converge to the target faster.

In addition, the weights of the attractive force between the target and the CoM  $c_1^{sc}$  and  $c_2^{sc}$  are freely set so that the CoM can converge to the target as soon as possible. Namely, the bigger weight the faster convergence, but if it is too big the overshoot will appear. Keep in mind that the choice of  $c_1^{sc}$ ,  $c_2^{sc}$  does not cause the collision with the obstacle. This is different from the choice of  $c_1^t$  and  $c_2^t$  which are selected less than  $c_1^\beta$  and  $c_2^\beta$ , respectively. As shown in Figure 2, when the CoM is controlled to track the target directly the force  $F_{C\gamma}$  is created to support the agents to move back to surround the target faster, or the CoM will converge to the target faster. For more information, see the simulation results. ■

For the *Multi-CoM* flocking control algorithm, we have the following statement for the stability properties.

In cluttered environments, consider a system of  $n$  mobile agents, that have dynamics (1) and are controlled by the *Multi-CoM* flocking algorithm (26). Then based on our observations which are shown in the simulation results we see that:

1. The CoM of positions and velocities of all agents in the network will exponentially converge to the target in the free space.

2. The error between the CoM's position and the target's position is reduced in the obstacle space.

The results of the *Multi-CoM* flocking algorithm are similar to the *Single-CoM* flocking algorithm. However, the benefit of the *Multi-CoM* flocking algorithm is that each agent is controlled locally instead of globally as in

the *Single-CoM* flocking algorithm.

### B. Stability Analysis of Flocking in Noisy Environments

Before analyzing the stability of the flocking control algorithm, *Multi-CoM-Cohesion*, we build the error dynamic model of the flocking system in noisy environments.

1) *Error Dynamic Model*: To study the stability properties, we have the error dynamics of the system given as follows:

$$\begin{cases} \dot{d}_{ig} = v_{ig} \\ \dot{v}_{ig} = u_i - \frac{1}{n} \sum_{j=1}^n u_j = u_i - \bar{u}, \quad i = 1, 2, \dots, n. \end{cases} \quad (47)$$

here  $\bar{u} = \frac{1}{n} \sum_{j=1}^n u_j$ .

We have following definitions:

$d_{ig} = q_i - \bar{q}$  is the relative distance between node  $i$  and its global average of position;

$v_{ig} = p_i - \bar{p}$  is the relative velocity between node  $i$  and its global average of velocity;

Then we have the following relations:

$$\begin{aligned} d_{il} &= q_i - \bar{q}_i = d_{ig} + \bar{q} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} q_j \\ &= d_{ig} + \bar{q} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} (d_{jg} + \bar{q}) \\ &= d_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg}. \end{aligned} \quad (48)$$

Then similar to  $d_{il}$ ,  $v_{il}$  is obtained as follows:

$$v_{il} = v_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg}. \quad (49)$$

The estimates of the local average of position and velocity, respectively in (37) is rewritten as

$$\hat{q}_i = q_i - d_{ig} + \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg} + \bar{\epsilon}_q^i. \quad (50)$$

$$\hat{p}_i = p_i - v_{ig} + \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg} + \bar{\epsilon}_p^i. \quad (51)$$

Now, we can rewrite the control law (42) with considering (41), (50) and (51):

$$\begin{aligned} u_i &= c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \\ &\quad + c_2^\alpha \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q)(\hat{p}_j - \hat{p}_i) \\ &\quad + (c_1^l - c_{pos})(d_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg}) \\ &\quad + (c_2^l - c_{ve})(v_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg}) \\ &\quad - (c_1^t + c_1^l)(q_i - q_t) - (c_2^t + c_2^l)(p_i - p_t) \end{aligned}$$

$$\begin{aligned}
 & -c_{pos}\epsilon_d^i - c_{ve}\epsilon_v^i - c_1^l\bar{\epsilon}_q^i - c_2^l\bar{\epsilon}_p^i \\
 & - (c_1^t + c_1^l)\epsilon_q^{it} - (c_2^t + c_2^l)\epsilon_p^{it}
 \end{aligned} \quad (52)$$

The average of control law for composite system is

$$\begin{aligned}
 \bar{u} = & \frac{c_1^\alpha}{n} \sum_{i=1}^n \left[ \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \right] \\
 & + \frac{c_2^\alpha}{n} \sum_{i=1}^n \left[ \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q)(\hat{p}_j - \hat{p}_i) \right] \\
 & + \left( \frac{c_1^l - c_{pos}}{n} \right) \sum_{i=1}^n \left( d_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg} \right) \\
 & + \left( \frac{c_2^l - c_{ve}}{n} \right) \sum_{i=1}^n \left( v_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg} \right) \\
 & - \left( \frac{c_1^t + c_1^l}{n} \right) \sum_{i=1}^n (q_i - q_t) - \left( \frac{c_2^t + c_2^l}{n} \right) \sum_{i=1}^n (p_i - p_t) \\
 & - \frac{1}{n} \sum_{i=1}^n [c_{pos}\epsilon_d^i + c_{ve}\epsilon_v^i + c_1^l\bar{\epsilon}_q^i + c_2^l\bar{\epsilon}_p^i \\
 & + (c_1^t + c_1^l)\epsilon_q^{it} + (c_2^t + c_2^l)\epsilon_p^{it}]
 \end{aligned} \quad (53)$$

Substitute  $u_i$  in (52) and  $\bar{u}$  in (53) into (47) we obtain:

$$\begin{aligned}
 \dot{v}_{ig} = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \\
 & - \frac{c_1^\alpha}{n} \sum_{i=1}^n \left[ \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \right] \\
 & + c_2^\alpha \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q)(\hat{p}_j - \hat{p}_i) \\
 & - \frac{c_2^\alpha}{n} \sum_{i=1}^n \left[ \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q)(\hat{p}_j - \hat{p}_i) \right] \\
 & - \left( \frac{c_1^l - c_{pos}}{|N_i^\alpha \cup \{i\}|} \right) \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg} \\
 & - \left( \frac{c_2^l - c_{ve}}{|N_i^\alpha \cup \{i\}|} \right) \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg} \\
 & - \left( \frac{c_1^l - c_{pos}}{n} \right) \sum_{i=1}^n \left( d_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg} \right) \\
 & - \left( \frac{c_2^l - c_{ve}}{n} \right) \sum_{i=1}^n \left( v_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg} \right) \\
 & - (c_{pos} - c_1^l)d_{ig} - (c_{ve} - c_2^l)v_{ig} \\
 & - (c_1^t + c_1^l)d_{ig} - (c_2^t + c_2^l)v_{ig} \\
 & - c_{pos}\epsilon_d^i - c_{ve}\epsilon_v^i - c_1^l\bar{\epsilon}_q^i - c_2^l\bar{\epsilon}_p^i \\
 & - (c_1^t + c_1^l)\epsilon_q^{it} - (c_2^t + c_2^l)\epsilon_p^{it} \\
 & + \frac{1}{n} \sum_{i=1}^n [c_{pos}\epsilon_d^i + c_{ve}\epsilon_v^i + c_1^l\bar{\epsilon}_q^i + c_2^l\bar{\epsilon}_p^i \\
 & + (c_1^t + c_1^l)\epsilon_q^{it} + (c_2^t + c_2^l)\epsilon_p^{it}] \\
 = & - (c_1^t + c_{pos})d_{ig} - (c_2^t + c_{ve})v_{ig} \\
 & + \Phi_i + \Omega_i(V) + \zeta_i,
 \end{aligned} \quad (54)$$

where

$$\begin{aligned}
 \Phi_i = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \\
 & - \frac{c_1^\alpha}{n} \sum_{i=1}^n \left[ \sum_{j \in N_i^\alpha} \phi_\alpha(\|\hat{q}_j - \hat{q}_i\|_\sigma) \hat{n}_{ij} \right] \\
 & + c_2^\alpha \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q)(\hat{p}_j - p_i) \\
 & - \frac{c_2^\alpha}{n} \sum_{i=1}^n \left[ \sum_{j \in N_i^\alpha} \hat{a}_{ij}(q)(\hat{p}_j - p_i) \right]; \\
 \Omega_i(V) = & - \left( \frac{c_1^l - c_{pos}}{|N_i^\alpha \cup \{i\}|} \right) \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg} \\
 & - \left( \frac{c_2^l - c_{ve}}{|N_i^\alpha \cup \{i\}|} \right) \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg} \\
 & - \left( \frac{c_1^l - c_{pos}}{n} \right) \sum_{i=1}^n \left( d_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_{jg} \right) \\
 & - \left( \frac{c_2^l - c_{ve}}{n} \right) \sum_{i=1}^n \left( v_{ig} - \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} v_{jg} \right); \\
 \zeta_i = & \frac{1}{n} \sum_{i=1}^n [c_{pos}\epsilon_d^i + c_{ve}\epsilon_v^i + c_1^l\bar{\epsilon}_q^i + c_2^l\bar{\epsilon}_p^i \\
 & + (c_1^t + c_1^l)\epsilon_q^{it} + (c_2^t + c_2^l)\epsilon_p^{it}] \\
 & - [c_{pos}\epsilon_d^i + c_{ve}\epsilon_v^i + c_1^l\bar{\epsilon}_q^i + c_2^l\bar{\epsilon}_p^i \\
 & + (c_1^t + c_1^l)\epsilon_q^{it} + (c_2^t + c_2^l)\epsilon_p^{it}]
 \end{aligned}$$

here, we define  $V_i = [d_{ig} \ v_{ig}]^T$  and  $V = [V_1, V_2, \dots, V_n]^T$ .

Rewrite (54) in state space representation

$$\begin{bmatrix} \dot{d}_{ig} \\ \dot{v}_{ig} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -k_1 I & -k_2 I \end{bmatrix} \begin{bmatrix} d_{ig} \\ v_{ig} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} (\Phi_i + \Omega_i(V) + \zeta_i), \quad (55)$$

here  $k_1 = (c_1^t + c_{pos})$ ,  $k_2 = (c_2^t + c_{ve})$ , and  $I$  is an  $m \times m$  identity matrix.

Then we can rewrite (55) as

$$\dot{V}_i = \begin{bmatrix} 0 & I \\ -k_1 I & -k_2 I \end{bmatrix} V_i + \begin{bmatrix} 0 \\ I \end{bmatrix} (\Phi_i + \Omega_i(V) + \zeta_i) \quad (56)$$

Let the matrix  $A_i = \begin{bmatrix} 0 & I \\ -k_1 I & -k_2 I \end{bmatrix}$ , then we have the characteristic equation as:

$$\det(\lambda I - A_i) = (\lambda^2 + k_2 \lambda + k_1)^m = 0. \quad (57)$$

Since  $k_1 > 0$ ,  $k_2 > 0$ , and if  $k_2 < 2\sqrt{k_1}$  then all roots of the characteristic equation (57) have negative real parts ( $\text{Re}(\lambda_i) < 0$ ).

2) *Stability Analysis of the Multi-CoM-Cohesion algorithm*: In this subsection we will analyze the stability of the flocking control algorithm, *Multi-CoM-Cohesion*, in noisy environments based on the Lyapunov approach.

We assume that the errors of sensing position and velocity have linear relationship with the magnitude of the state of the error system. That is because two agents are far away from each other, the sensing errors will usually increase. Hence, we have

$$\begin{cases} \|\epsilon_d^i(t)\| \leq c_{ed1}^i \|V_i(t)\| + c_{ed2}^i \\ \|\epsilon_v^i(t)\| \leq c_{ev1}^i \|V_i(t)\| + c_{ev2}^i, \quad i = 1, 2, \dots, n. \end{cases} \quad (58)$$

We also assume that the noise  $\epsilon_q^{it}$  and  $\epsilon_p^{it}$  on the target tracking terms (negative feedbacks) are bounded as

$$\begin{cases} \|\epsilon_q^{it}(t)\| \leq c_{eq}^i \\ \|\epsilon_p^{it}(t)\| \leq c_{ep}^i, \quad i = 1, 2, \dots, n, \end{cases} \quad (59)$$

and the noise  $\bar{\epsilon}_q^i$  and  $\bar{\epsilon}_p^i$  on the estimates of local average of position and velocity are bounded as

$$\begin{cases} \|\bar{\epsilon}_q^i(t)\| \leq \bar{c}_{eq}^i \\ \|\bar{\epsilon}_p^i(t)\| \leq \bar{c}_{ep}^i, \quad \bar{i} = 1, 2, \dots, n. \end{cases} \quad (60)$$

here  $\bar{c}_{eq}^i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{i=1}^{|N_i^\alpha \cup \{i\}|} c_{eq}^i$ , and  $\bar{c}_{ep}^i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{i=1}^{|N_i^\alpha \cup \{i\}|} c_{ep}^i$ .

**Theorem 2.** Consider a system of  $n$  mobile agents with dynamics (1) and controlled by (42), and all noise are bounded by (58), (59) and (60). Let

$$c_{pv}^1 = \frac{(c_{pos} + 1)^2 + c_{ve}^2}{2c_{pos}c_{ve}} + \sqrt{\left(\frac{c_{pos} + c_{ve}^2 - 1}{2c_{pos}c_{ve}}\right)^2 + \frac{1}{c_{pos}^2}},$$

and if

$$c_{pos}c_{ed1}^i + c_{ve}c_{ev1}^i \leq \frac{1}{c_{pv}^1},$$

and the parameters are such that

$$\begin{aligned} & \sum_{j=1}^m \left[ \frac{2c_{pv}^1 \sqrt{(c_1^j - c_{pos})^2 + (c_2^j - c_{ve})^2}}{(1 - \epsilon_i)[1 - c_{pv}^1(c_{pos}c_{ed1}^i + c_{ve}c_{ev1}^i)]} \right. \\ & \left. - \frac{2c_{pv}^1 \frac{1}{n}(c_{pos}c_{ed1}^i + c_{ve}c_{ev1}^i)}{(1 - \epsilon_i)[1 - c_{pv}^1(c_{pos}c_{ed1}^i + c_{ve}c_{ev1}^i)]} \right] < 1, \end{aligned}$$

here  $0 < \epsilon_i < 1$  for  $\forall i$ , then the trajectories of (56) are bounded.

*Proof:* To study the stability of the error dynamics (56), one possible choice is to choose the Lyapunov function for each agent as

$$L_i(V_i) = V_i^T P V_i, \quad (61)$$

here  $P = P^T$  is a  $2m \times 2m$  positive-definite matrix ( $P > 0$ ). Then, the Lyapunov function for the composite system is

$$L(V) = \sum_{i=1}^n V_i^T P V_i.$$

From (61) we have

$$\dot{L}_i(V_i) = V_i^T P \dot{V}_i + \dot{V}_i^T P V_i. \quad (62)$$

Then, substitute  $\dot{V}_i$  in (56) into (62) we obtain

$$\begin{aligned} \dot{L}_i(V_i) &= V_i^T (P A_i + A_i^T P) V_i \\ &\quad + 2V_i^T P B (\Phi_i + \Omega_i(V) + \zeta_i) \\ &= -V_i^T C V_i + 2V_i^T P B (\Phi_i + \Omega_i(V) + \zeta_i), \end{aligned}$$

here  $B = \begin{bmatrix} 0 \\ I \end{bmatrix}$ , and  $C = -(P A_i + A_i^T P)$ .

The remaining part of this proof is to show  $\dot{L}_i(V_i) < 0$ . The detailed proof of  $\dot{L}_i(V_i) < 0$  is similar to that in the reference [24]. ■

## V. SIMULATION RESULTS

### A. Flocking Control in Cluttered Environments

In this subsection we test our proposed algorithms in cluttered environments, *Single-CoM* (23) and *Multi-CoM* (26), respectively in simulation with different trajectories of the moving target and compare them with *No-CoM* (19).

First of all we test our algorithms for the case that the target moves in a sine wave trajectory. Parameters used in this simulation are specified as follows:

- Parameters of flocking: number of agents = 120; the initial positions of agents are randomly distributed in the square area of 90 x 90 size; the initial velocities of agents are set to zero. Parameters  $a = b = 5$ ; the interaction range  $r = 1.2d = 9$ ;  $\epsilon = 0.1$  for the  $\sigma$ -norm;  $h = 0.2$  for the bump function ( $\phi_\alpha(z)$ );  $h = 0.9$  for the bump function ( $\phi_\beta(z)$ ).

- Parameters of target movement: The target moves in a sine wave trajectory:  $q_t = [50 + 35t, 295 - 35\sin(t)]^T$  with  $0 \leq t \leq 8.5$ .

Second we test our algorithms for the case in which the target moves in a circle trajectory. Parameters used in this simulation are specified as follows:

- Parameters of flocking: parameters used in this case are the same with those in the sine trajectory case.

- Parameters of target movement: The target moves in a circle trajectory:  $q_t = [310 - 160\cos(t), 255 + 160\sin(t)]^T$  with  $0 \leq t \leq 5$ .

To compare three algorithms, *No-CoM* (19), *Single-CoM* (23) and *Multi-CoM* (26) we use the same initial state (position and velocity) of mobile agents.

Figure 3 represents the snapshots of mobile agents tracking the target moving in the sine wave and circle trajectories, respectively, and Figure 4 shows the error between the CoM's positions and the target's positions (called tracking performance) using three flocking control algorithms, *No-CoM*, *Single-CoM* and *Multi-CoM*, respectively.

In Figure 3, the snapshots of mobile agents when they are avoiding the obstacles are captured at the same time. Comparing the results we can see that in Figure 3 (b, b', c, c') more agents (agents) can pass through the narrow space between two obstacles than those in Figure 3 (a, a').

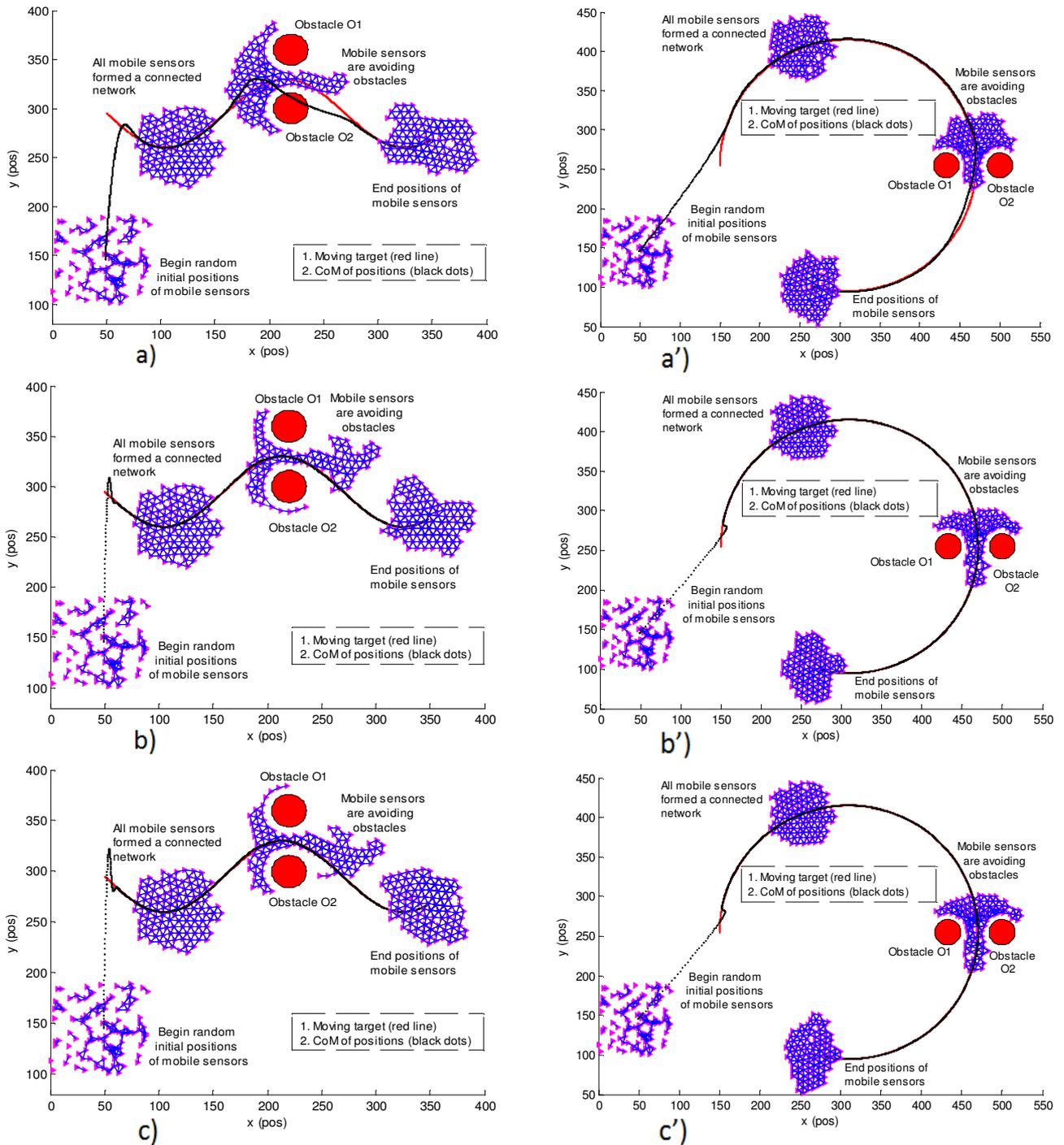


Figure 3. Snapshots of the positions of the mobile agents at the beginning, forming a connected network, avoiding obstacles and at the ending when they are tracking the target moving in the sine wave trajectory (a, b, c) and the circle trajectory (a', b', c') using flocking control algorithms with *No-CoM* (19), *Single-CoM* (23) and *Multi-CoM* (26), respectively. Here, (a, a') are for *No-CoM*, (b, b') are for *Single-CoM*, and (c, c') are for *Multi-CoM*.

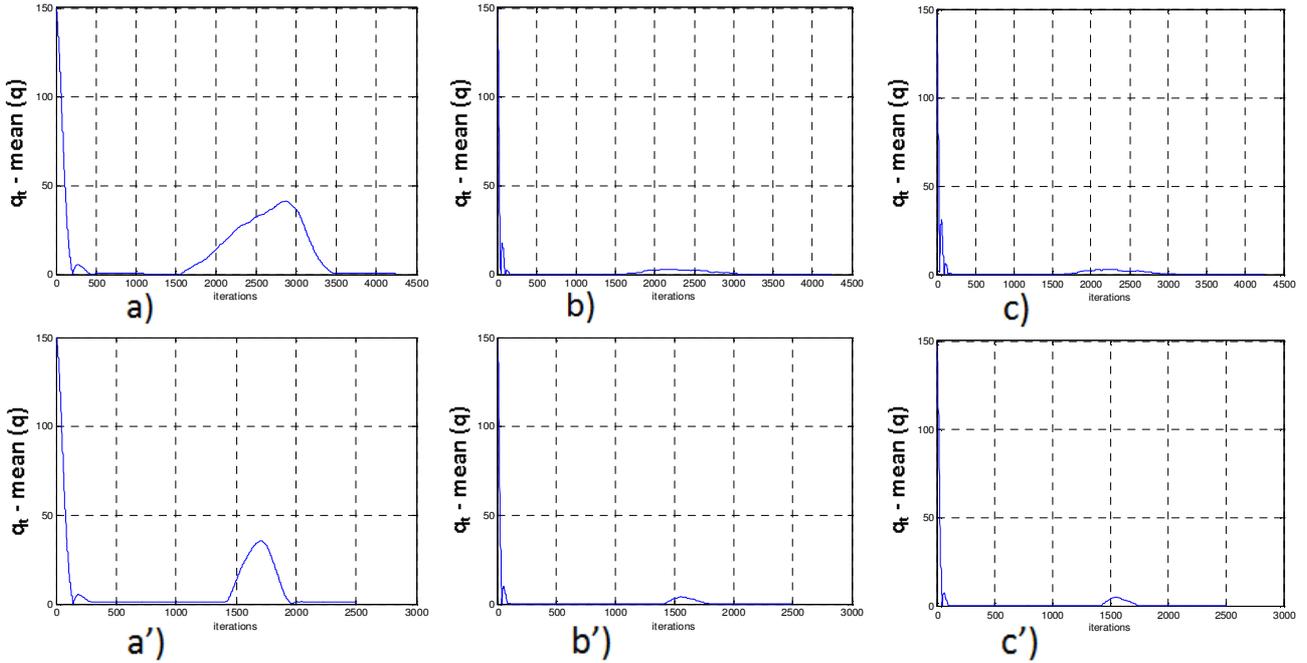


Figure 4. Error between the CoM's positions and the target's positions for the cases: sine wave trajectory (a, b, c) and circle trajectory (a', b', c') using flocking control algorithms with *No-CoM* (19), *Single-CoM* (23) and *Multi-CoM* (26), respectively. Here, (a, a') are for *No-CoM*, (b, b') are for *Single-CoM*, and (c, c') are for *Multi-CoM*.

This means that by controlling the CoM in the algorithms, *Single-CoM* and *Multi-CoM*, respectively (Figure 4 b, b', c, c') the agents can more closely follow the target than those in the *No-CoM* algorithm (Figures 4 a, a'). We can see that the results of tracking performance in Figure 4 (b, b', c, c') for both trajectories of the target using the *Single-CoM* and *Multi-CoM* algorithms, respectively, are better than those in Figure 4 (a, a') using the *No-CoM* algorithm.

In addition, by observing the movement of the agents in the obstacle space we clearly see that by controlling the CoM all agents can easily pass through the obstacles. Hence, this makes all agents to better follow and surround the target. Moreover, in free space all agents can track the target faster while still maintaining the formation, and especially in the initial tracking time (about 70 - 100 iterations) all agents quickly catch up the target (Figure 4 (b, b', c, c')). However without controlling the CoM the agents usually are stuck behind the obstacles that causes the poor tracking performance as shown in Figure 4 (a, a'), and in free space all agents track the target slower such as in the initial tracking time it takes about 300 to 400 iterations for all agents to catch up the target.

We tested our algorithms in several contexts of obstacle distributions and different shapes of the obstacles such as the wall obstacles, and we see that our proposed algorithms, *Single-CoM* and *Multi-CoM*, work better than the *No-CoM* in both free and obstacle spaces. Since the space of the paper is limited we do not show more results here.

### B. Flocking Control in Noisy Environments

In this subsection we first discuss a metric to evaluate the network connectivity. Then we test our proposed flocking control algorithms, *Multi-CoM-Shrink* (40), *Multi-CoM-Cohesion* (42), and compare them with the existing flocking control algorithm, *No-CoM* (19), in noisy environments.

To evaluate the network connectivity maintenance, first we know that the link (connectivity) between node  $i$  and node  $j$  is maintained if the distance between them  $0 < \|q_i - q_j\| \leq r$ . Otherwise this link is considered broken. Then for graph connectivity, a dynamic graph  $G(\vartheta, E)$  is connected at time  $t$  if there exists a path between any two vertices. An example of graph connectivity is shown in Figure 5.

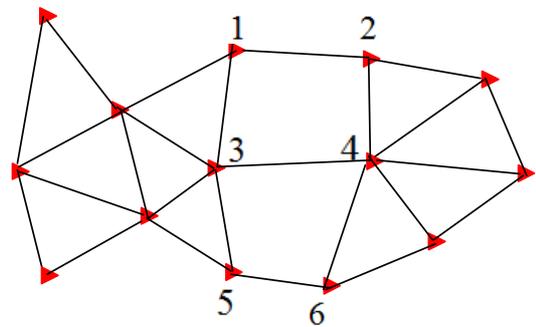


Figure 5. If one or two of the links (1,2), (3,4), (5,6) is broken the graph connectivity is still remained, but if all of that links is broken the graph connectivity is lost.

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

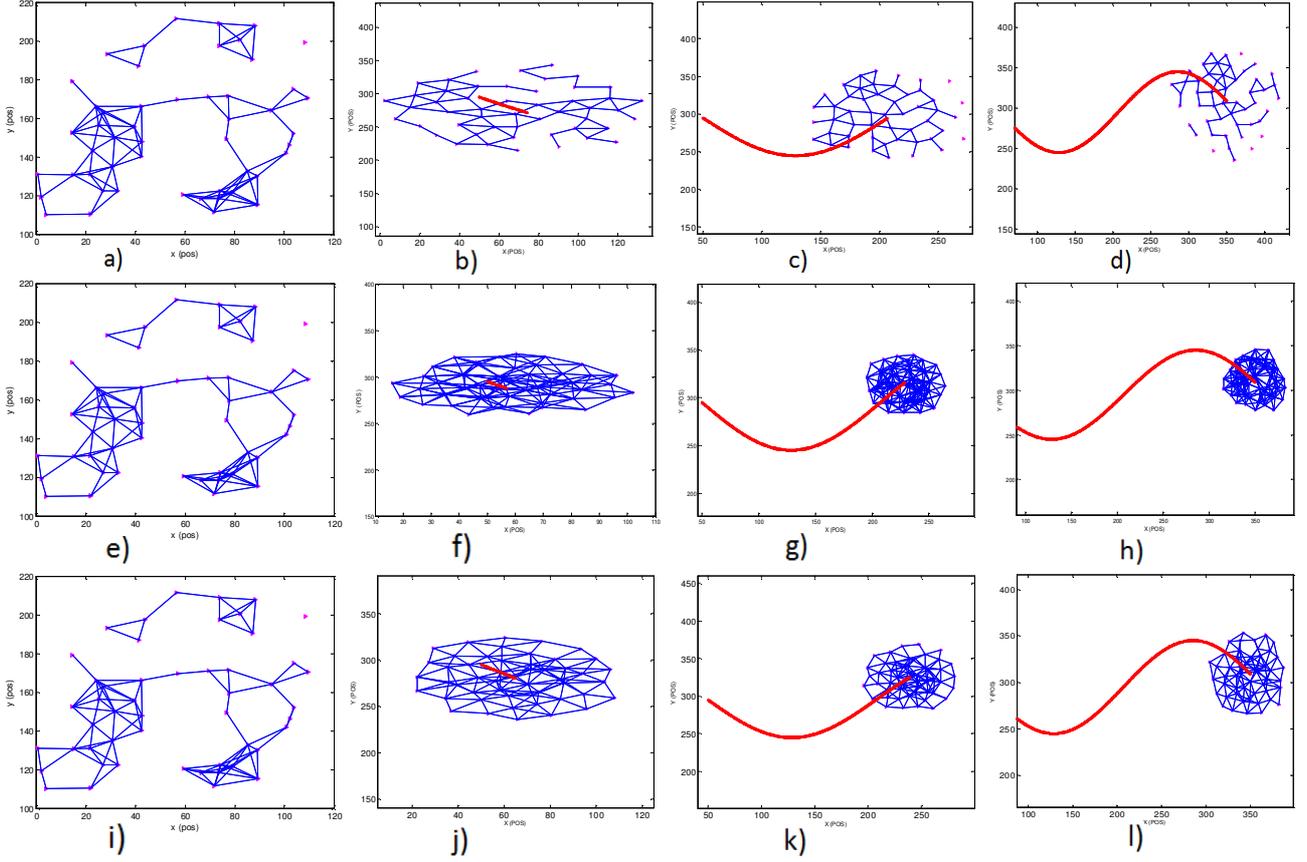


Figure 6. Snapshots of agents when they are randomly distributed (a, e, i), and when they form a network and track a target (red/dark line) moving in a sine wave trajectory (b, c, d; f, g, h; j, k, l), where (a, b, c, d) are for the algorithm (19), (e, f, g, h) are for the *Multi-CoM-Shrink* algorithm, and (i, j, k, l) are for the *Multi-CoM-Cohesion* algorithm.

follows:

$$[c_{ij}(t)] = \begin{cases} 1, & \text{if } j \in N_i(t), i \neq j \\ 0, & \text{if } j \notin N_i(t), i \neq j \end{cases} \quad (63)$$

and  $c_{ii} = 0$ .

Since the rank of the Laplacian of a connected graph  $[c_{ij}(t)]$  of order  $n$  is at most  $(n - 1)$  [4] or  $\text{rank}([c_{ij}(t)]) \leq (n - 1)$ , the relative connectivity of a network at time  $t$  is defined as

$$C(t) = \frac{1}{n - 1} \text{rank}([c_{ij}(t)]). \quad (64)$$

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 algorithms.

The parameters used in this simulation are specified as follows:

- Parameters of flocking: number of agents = 50 (randomly distributed in the square area of 120 x 120 size);  $a = b = 5$ ; the active range  $r = 19$ ;  $\epsilon = 0.1$  for the  $\sigma$ -norm;  $h = 0.2$  for the bump functions ( $\phi_\alpha^{new}(z)$ ,  $\phi_\alpha(z)$ );  $h = 0.9$  for the bump function ( $\phi_\beta(z)$ ). The desired distance for the algorithms (19) and *Multi-CoM-Cohesion*,  $d = 16$ . For the *Multi-CoM-Shrink* algorithm,  $r_w = 3.4$ , hence  $d = r - 2r_w = 19 - 2 \times 3.4 = 12.2$ .

- Parameters of target movement:

Case 1: The target moves in a sine wave trajectory:  $q_t = [50 + 50t, 295 - 50\sin(t)]^T$  with  $0 \leq t \leq 6$

Case 2: The target moves in a circle trajectory:  $q_t = [310 - 160\cos(t), 255 + 160\sin(t)]^T$  with  $0 \leq t \leq 4$ .

- The noise used in the simulation is Gaussian with zero mean and a variance of 1.

Figures 6 and 7 show the results of the moving target (red/dark line) tracking in the sine wave and circle trajectories in noisy environments for three algorithms, (19), *Multi-CoM-Shrink* and *Multi-CoM-Cohesion*. Especially, Figures 6(a, b, c, d) and 7(a, b, c, d) are for the flocking control algorithm (19). Figures 6(e, f, g, h) and 7(e, f, g, h) are for the proposed flocking control algorithm *Multi-CoM-Shrink*. Figures 6(i, j, k, l) and 7(i, j, k, l) are for the proposed flocking control algorithm *Multi-CoM-Cohesion*.

To compare our proposed flocking control algorithms, *Multi-CoM-Shrink* and *Multi-CoM-Cohesion* with the existing flocking algorithm (19), we use the same initial state (position and velocity) of the mobile agents. Figure 8 shows the results of the tracking performance and the connectivity, respectively: (a, c) are for the flocking control algorithm (19), (b, d) are for the *Multi-CoM-Shrink* flocking control algorithm, and (e, f) are for the *Multi-CoM-Cohesion* flocking control algorithm. Comparing the results in these figures we clearly see that:

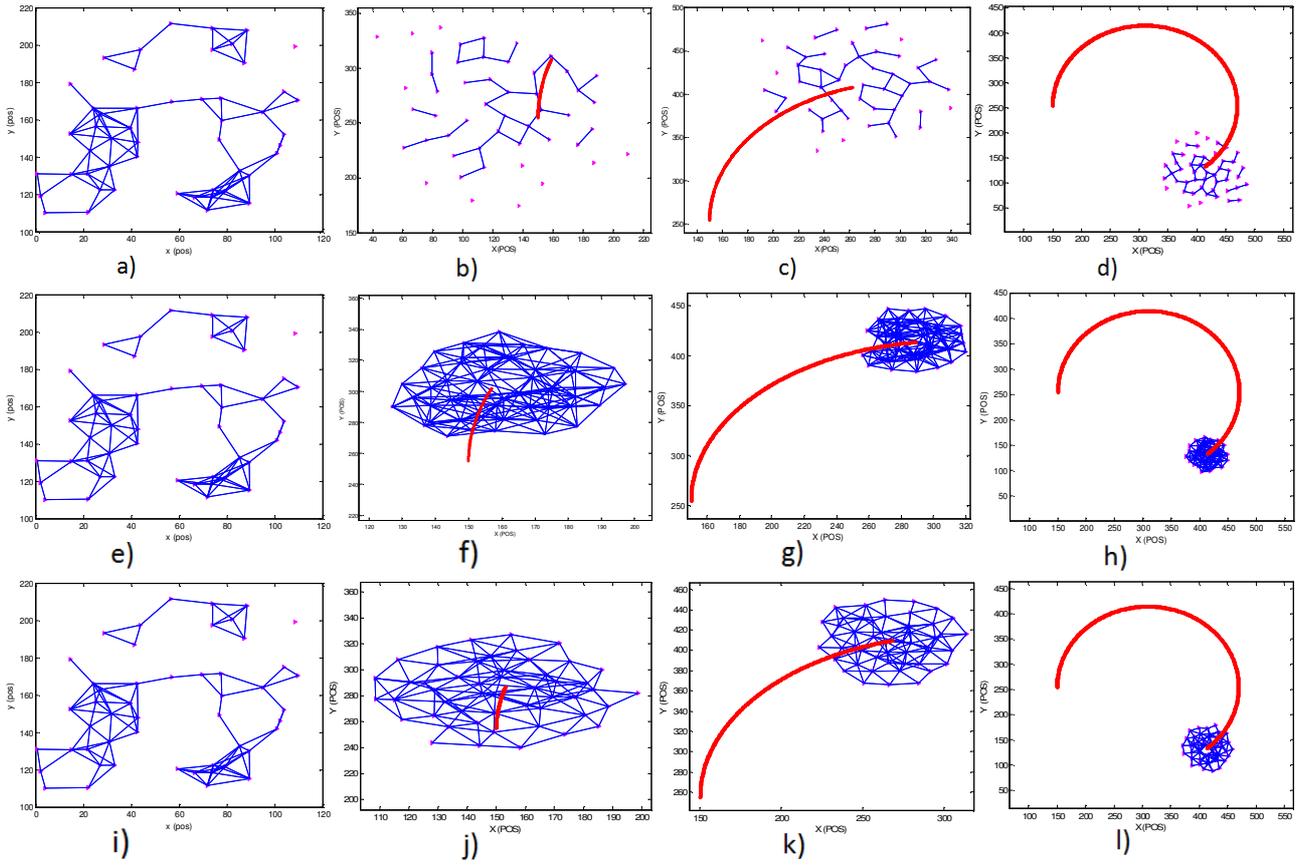


Figure 7. Snapshots of agents when they are randomly distributed (a, e, i), and when they form a network and track a target (red/dark line) moving in a circle trajectory (b, c, d; f, g, h; j, k, l), where (a, b, c, d) are for the algorithm (19), (e, f, g, h) are for the *Multi-CoM-Shrink* algorithm, and (i, j, k, l) are for the *Multi-CoM-Cohesion* algorithm.

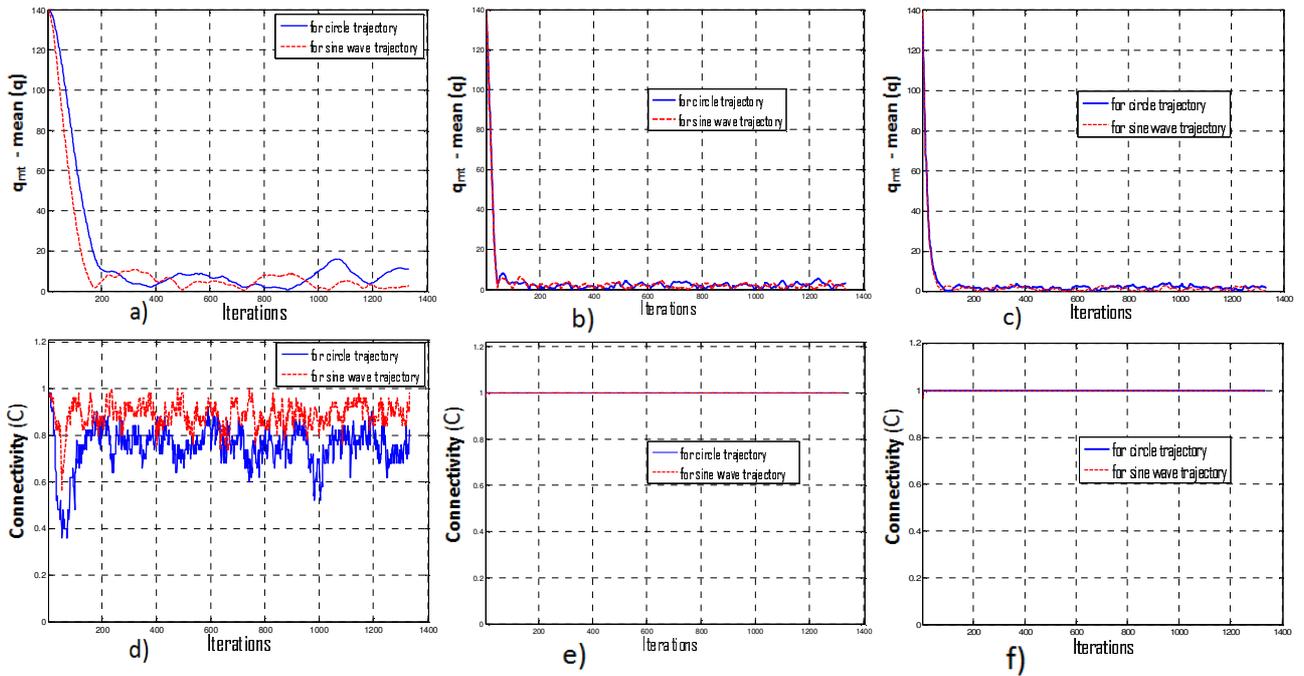


Figure 8. The tracking performance results (error between the CoM and target positions): (a) is for the algorithm(19), (b) is for the *Multi-CoM-Shrink* algorithm, and (c) is for the *Multi-CoM-Cohesion* algorithm. The connectivity is evaluated by the  $C(t)$  value: (d) is for the algorithm (19), (e) is for the *Multi-CoM-Shrink* algorithm, and (f) is for the *Multi-CoM-Cohesion* algorithm.

- For the flocking control algorithm (19): The tracking performance has big errors, and it makes the target out of the center of the network. In addition, the connectivity is lost, or the network is broken ( $C(t) < 1$ ).
- For the *Multi-CoM-Cohesion* algorithm: The tracking performance has small errors. In addition, the agents can quickly form a network (only five iterations) and then maintain connectivity ( $C(t) = 1$ ).
- For the *Multi-CoM-Shrink* algorithm: The tracking performance also has small errors, and the connectivity is maintained after six iterations. However, the size of the network is smaller than that of the *Multi-CoM-Cohesion* flocking control algorithm, and each agent has more neighbors because each agent tries to reduce the distance to its neighbor in order to keep connection to them.

## VI. CONCLUSION

In this paper, we considered the problem of controlling a group of mobile agents to track a target in cluttered and noisy environments, respectively. First, flocking control algorithms with *Single-CoM* and *Multi-CoM* are designed to enable mobile agents to track and observe the moving target more effectively in cluttered environments while maintaining their formation and collision avoidance among agents. By controlling the CoM explicitly, the mobile agents can track and observe the moving target better. In addition, flocking control with *No-CoM*, flocking control with *Single-CoM*, and flocking control with *Multi-CoM* are compared. The numerical simulations are done with different target trajectories to demonstrate our theoretical results. Second, in noisy environments, two flocking control algorithms, *Multi-CoM-Shrink* and *Multi-CoM-Cohesion*, are proposed. In the *Multi-CoM-Shrink* algorithm our approach is to shrink the size of the network by reducing the distance among agents. In the *Multi-CoM-Cohesion* algorithm our approach integrates local position and velocity cohesion feedbacks in order to deal with the noise. As a result the network connectivity preservation is improved, and collision avoidance among agents is guaranteed in both cluttered and noisy environments. In addition, the stability of the proposed algorithms is investigated.

## 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*, pp. 2010–2015, 2003.
- [2] —, "Stable flocking of mobile agents, part II: dynamic topology," *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 2016–2021, 2003.
- [3] —, "Flocking in fixed and switching networks," *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, May 2007.
- [4] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.
- [5] C. Reynolds, "Flocks, birds, and schools: A distributed behavioral model," *Computer Graphics, ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California*, vol. 21, no. 4, pp. 25–34, 1987.
- [6] T. Vicsek, A. Czirok, E. Jacob, I. Cohen, and O. Schochet, "Novel type of phase transitions in a system of self-driven particles," *Phys. Rev. Lett.*, vol. 75, pp. 1226–1229, 1995.
- [7] H. Levine, W. J. Rappel, and I. Cohen, "Self-organization in systems of self-propelled particles," *Phys. Review. E*, vol. 63, pp. 017 101–017 104, 2000.
- [8] A. Mogilner, L. Edelstein-Keshet, L. Bent, and A. Spiros, "Mutual interactions, potentials, and individual distance in a social aggregation," *J. Math. Biol.*, vol. 47, pp. 353–389, 2003.
- [9] 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.*, vol. 218, pp. 1–11, 2002.
- [10] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader, part I: with a minority of informed agents," *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 2937–2942, 2007.
- [11] —, "Flocking of multi-agents with a virtual leader, part II: with a virtual leader of varying velocity," *Proceedings of the 46th IEEE Conference on Decision and Control*, pp. 1429–1434, 2007.
- [12] H. Shi, L. Wang, T. Chu, F. Fu, and M. Xu, "Flocking of multi-agents with a virtual leader," *Proceedings of the 2007 IEEE Symposium on Artificial Life*, pp. 287–29, 2007.
- [13] Z. Wang and D. Gu, "A survey on application of consensus protocol to flocking control," *Proceedings of the 12th Chinese Automation and Computing Society Conference in the UK, England*, pp. 1–8, 2006.
- [14] M. Anderson, E. McDaniel, and S. Chenney, "Constrained animation of flocks," *Eurographics/SIGGRAPH Symposium on Computer Animation*, pp. 1–12, 2003.
- [15] A. Regmi, R. Sandoval, R. Byrne, H. Taner, and C. T. Abdallah, "Experimental implementation of flocking algorithms in wheeled mobile robots," *American Control Conference*, pp. 4917–4922, 2005.
- [16] V. Gervasi and G. Prencipe, "Coordination without communication: the case of the flocking problem," *Discrete Applied Mathematics*, pp. 324–344, 2004.
- [17] D. Cruz, J. McClintock, B. Perteet, O. A. Orqueda, Y. Cao, and R. Fierro, "Decentralized cooperative control—a multivehicle platform for research in networked embedded systems," *IEEE Control Systems Magazine*, pp. 58–78, 2007.
- [18] R. Olfati-Saber, "Distributed tracking for mobile sensor networks with information driven mobility," *Proceedings of the 2007 American Control Conference*, pp. 4606–4612, 2007.
- [19] H. Su, X. Wang, and Z. Lin, "Flocking of multi-agents with a virtual leader," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 293–307, 2009.
- [20] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 8, pp. 1292–1302, 2006.
- [21] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.
- [22] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperative in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.
- [23] N. E. Leonard and E. Fiorelli, "Virtual leaders, artificial potentials, and coordinated control of groups," *Proceedings of the 40th IEEE Conference on Decision and Control*, pp. 2968–2973, 2001.

- [24] Y. Liu and K. M. Passino, "Stable social foraging swarms in a noisy environment," *IEEE Transactions on Automatic Control*, vol. 49, no. 1, pp. 30–44, 2004.



Hung Manh La is a research associate at the Center for Advanced Infrastructure and Transportation (CAIT), Rutgers University, USA, and a lecturer at the Department of Electronics Engineering, Thai Nguyen University of Technology, Vietnam. He had been a post-doctoral associate at CAIT, Rutgers University from 2011 to 2012. He received his Ph.D. degree in Electrical and Computer Engineering from Oklahoma State University, USA in 2011, his M.S and B.S. degrees in Electrical Engineering from Thai Nguyen University of Technology, Vietnam in 2003 and 2001, respectively. He has been actively involved in research projects with the Federal Highway Administration, Department of Transportation, Department of Defense and National Science Foundation. His research has resulted in two best paper awards in the 2009 and 2010 Conferences on Theoretical and Applied Computer Science, respectively, and one best presentation of session in the 2009 American Control Conference. His research includes mobile robots, mobile sensor networks, cooperative control, learning and sensing, intelligent transportation systems.



Weihua Sheng is an associate professor at the School of Electrical and Computer Engineering, Oklahoma State University, USA. He received his Ph.D. degree in Electrical and Computer Engineering from Michigan State University in May 2002, his M.S and B.S. degrees in Electrical Engineering from Zhejiang University, China in 1997 and 1994, respectively. During 1997-1998, He was a research engineer at the R&D center in Huawei Technologies Co., China. During 2002-2006, he taught in the Electrical and Computer Engineering Department at Kettering University (formerly General Motor Institute). He is a senior member of IEEE and has participated in organizing several IEEE international conferences and workshops in the area of intelligent robots and systems. He is the author of one US patent and more than 120 papers in major journals and international conferences. His current research interests include wearable computing, mobile robotics, human robot interaction, distributed sensing and control, and intelligent transportation systems. His research is supported by NSF, DoD, DEPSCoR, DoT, etc. Dr. Sheng is currently an Associate Editor for *IEEE Transactions on Automation Science and Engineering*.