

## Dynamic target tracking and observing in a mobile sensor network

Hung Manh La<sup>a</sup>, Weihua Sheng<sup>b,\*</sup>

<sup>a</sup> Center for Advanced Infrastructure and Transportation, Rutgers, The State University of New Jersey, Piscataway, NJ 08854, USA

<sup>b</sup> School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK, 74078, USA

### ARTICLE INFO

#### Article history:

Received 3 May 2009

Received in revised form

29 June 2011

Accepted 26 March 2012

Available online 21 April 2012

#### Keywords:

Flocking control  
Adaptive flocking control  
Multiple target tracking  
Multi-agent systems  
Mobile sensor networks  
Graph partitioning

### ABSTRACT

This paper presents novel approaches to (1) the problem of flocking control of a mobile sensor network to track and observe a moving target and (2) the problem of sensor splitting/merging to track and observe multiple targets in a dynamic fashion. First, to deal with complex environments when the mobile sensor network has to pass through a narrow space among obstacles, we propose an adaptive flocking control algorithm in which each sensor can cooperatively learn the network's parameters to decide the network size in a decentralized fashion so that the connectivity, tracking performance and formation can be improved. Second, for multiple dynamic target tracking, a seed growing graph partition (SGGP) algorithm is proposed to solve the splitting/merging problem. To validate the adaptive flocking control we tested it and compared it with the regular flocking control algorithm. For multiple dynamic target tracking, to demonstrate the benefit of the SGGP algorithm in terms of total energy and time consumption when sensors split, we compared it with the random selection (RS) algorithm. Several experimental tests validate our theoretical results.

© 2012 Elsevier B.V. All rights reserved.

## 1. Introduction

### 1.1. Motivation

Sensor networks [1–4], especially mobile sensor networks [5], have been attracting considerable research interest in recent years. Mobile sensor networks have several advantages over stationary sensor networks, such as the adaptation to environmental changes and reconfigurability for better sensing performance. Mobile sensor networks can be used in many applications such as target tracking [6] in underwater submarine detection and protection of endangered species [7]. A typical mobile sensor is a mobile robot equipped with actuators made by either DC motors or artificial muscles [8] as well as various sensors.

A main issue for multiple mobile sensors to track a moving target is that these sensors have to move together without collision among them, which requires the use of cooperative control methods [9–12]. One of these methods is flocking control [13]. We know that flocking is a phenomenon that a number of mobile agents move together and interact with each other while ensuring no collision, velocity matching, and flock centering [14]. In nature, schools of fish, birds, ants, and bees, etc., demonstrate the phenomenon of flocking. The problem of flocking has been studied for

many years. It has attracted many researchers in physics [15,16], mathematics [17], biology [18], and especially in control science in recent years [19–21,13,22–28].

In addition to these basic constraints, there are two important issues that should be addressed. The first issue is how to control the mobile sensor network to deal with complex environments populated by obstacles. For example how the mobile sensor network can pass through a narrow space among obstacles while maintaining connectivity, tracking performance and similar formation. This may require the design of an adaptive flocking control algorithm instead of a regular flocking control algorithm. The second issue in a mobile sensor network is how to track multiple targets simultaneously in a dynamic fashion. This requires that some sensors should split from the existing formation(s) to track new targets while ensuring the least disturbance to other sensors. This raises the question of which sensors should split from the existing formation(s) and how they should split so that the total energy consumption and time consumption are minimized. In addition, when some targets disappear the sensors which are tracking these targets should rejoin the existing groups.

In this paper, we first present a novel approach to the problem of flocking control of a mobile sensor network to track and observe a moving target. To deal with complex environments when the mobile sensor network has to pass through a narrow space among obstacles, we propose an adaptive flocking control algorithm in which each sensor can cooperatively learn the network's parameters to decide the size of network in a decentralized fashion so that the connectivity, tracking performance and formation can

\* Corresponding author.

E-mail addresses: [hung.la11@rutgers.edu](mailto:hung.la11@rutgers.edu) (H.M. La), [weihua.sheng@okstate.edu](mailto:weihua.sheng@okstate.edu) (W. Sheng).

be improved. Second, in the scenario of multiple dynamic targets, to solve the problem of sensor splitting/merging in a mobile sensor network, a *seed growing graph partition* (SGGP) algorithm is proposed.

## 1.2. Literature review

In this section, we review existing research related to our work, which includes flocking control, adaptive flocking control, and multiple target tracking in mobile sensor networks.

Flocking control has been studied by many researchers in recent years. Wang and Gu [28] presented a survey of recent research achievements in robot flocking. Their paper gave an overview of the basic knowledge of graph theory, potential function, network communication and system stability analysis. In [13], a theoretical framework for design and analysis of distributed flocking algorithms was proposed. These algorithms realized flocking control in the absence and presence of obstacles. Static and dynamic virtual leaders were used as navigational feedback for all mobile agents. An extension of the flocking control algorithms in [13], flocking of agents with a virtual leader in the case of a minority of informed agents and in the case of varying velocity of the virtual leader, was presented in [22,23]. Shi and Wang [24] investigated the dynamic properties of mobile agents for the case where the state of the virtual leader is time varying and the topology of the neighboring relations between agents is dynamic. Anderson et al. [19] demonstrated a new technique for generating the constrained group animations of flocks in which users can impose constraints on agents' positions at any time in the animation, or control the entire group to meet the shape constraints. Tanner et al. [25,26] studied the stability properties of a system of multiple mobile agents with double integrator dynamics in the case of fixed and dynamic topologies. Gervasi and Prencipe [21] studied the distributed coordination and control of a set of asynchronous, anonymous, memoryless mobile vehicles in the case of no communication among the vehicles. In particular, their paper analyzed the problem of flocking in a certain pattern and following a designated leader vehicle, while maintaining the pattern. Olfati-Saber [29] developed a distributed flocking algorithm for mobile sensor networks to track a moving target. In his paper, an extension of a distributed Kalman filtering algorithm was used for the sensors to estimate the target's position. In [20], a scalable multi-vehicle platform was developed to demonstrate a cooperative control algorithm in mobile sensor networks. Their flocking algorithm was implemented with five TXT-1 monster truck robots.

Adaptive flocking control, an extension of flocking control, has also gained research attention in recent years. Yang et al. [30] proposed an adaptive flocking control algorithm to avoid collision among robots themselves and between robots and obstacles. However, their algorithm did not consider the problem of formation, connectivity and tracking performance in complex environments. In addition, their algorithm only considered a static target (a rendezvous point) for all agents to get there. Lee and Chong [31] introduced a motion planning framework for a large number of autonomous robots that enables the robots to configure themselves adaptively into an area of arbitrary geometry. Their proposed method allows the robots to converge to the uniform distribution by forming an equilateral triangle with their two neighbors. Based on their geometric approach, the swarm could be uniformly self-deployed and adapt to unknown environments in a decentralized manner. However, the problem of target tracking was not addressed in their work. An extension of their work was developed in [32] by the same authors to allow the swarm of robots to go to predetermined rendezvous points. Their approach was based on a decentralized approach for adaptive

flocking of swarms of mobile robots to navigate autonomously in complex environments populated by obstacles. The problem of splitting/merging mobile robots in the network according to the environment conditions is addressed in their paper. Namely, when the swarm of robots detects obstacles, each robot splits from the network and determines its direction toward the static goal based on the width of space among obstacles. However, in reality it is difficult for each robot to sense the whole environment to compute the width of space among obstacles. Also, in their work the problem of controlling the size of the network was not considered, and the connectivity and formation were not guaranteed in a complex environment.

Multiple target tracking in mobile sensor networks has been investigated by several researchers in the last decade. Jung et al. [33] introduced a region-based approach to multiple target tracking using a network of communicating robots and stationary sensors. A coarse deployment controller distributes robots across regions using a topological map, and a target-following controller maximizes the number of tracked targets within a region. Tang and Ozguner [34] investigated the motion planning for a limited number of mobile sensor agents in an environment with multiple dynamic targets. The motion planning problem is formulated as an optimization problem that minimizes the average time between two consecutive target observations. Kolling and Carpin [35] presented a distributed control algorithm for multiple targets surveillance by multiple robots. Their algorithm utilizes information from sensors and communication to predict the minimum time before a robot loses a target. As a fundamental technique in multiple target tracking, sensor network partitioning has been studied by many researchers. The methods for network partitioning can be divided into centralized and decentralized. For centralized graph partition, there are several algorithms, such as the decomposition scheme that partitions a given graph into compactly connected two-terminal subgraphs [36], the graph clustering method based on the minimum cuts within a graph [37], the new graphical adaptation of the  $k$ -medoids algorithm [38], and the Girvan–Newman method [39]. For distributed graph partition, Derbel and Mosbah [40] proposed a linear time distributed algorithm to decompose a graph into a disjointed set of clusters. In [41,42], Goebels et al. presented a neighborhood-based strategy, a border switch strategy, and an exchange target strategy for the partitioning of large sets of agents into multiple groups.

In general, for flocking control most works focus on the configuration and topology of flocks. For single target tracking, the literatures address the problem of sensing model by using the distributed Kalman filter and the problem of target tracking with a minority of agents having knowledge of the target. Their algorithms work well in free space, but in the obstacle space they have some limitations, such as poor tracking performance, low speed and loss of connectivity. For adaptive flocking control most existing works focus on the coordination, formation and splitting/merging problems in both fixed and switching topologies. The problem of how to control the size of the network in a decentralized and adaptive fashion in complex or changing environments while maintaining connectivity, tracking performance and similar formation is still open. For multiple target tracking, existing literatures solve the tracking problem in both stationary and mobile sensor networks without paying attention to the network formation.

This paper will address the above problems inherited in the existing works. The rest of this paper is organized as follows. Section 2 describes the background for flocking control. The adaptive flocking control algorithm for single target tracking and observing is presented in Section 3. Section 4 presents the algorithm for multiple dynamic target tracking and observing. The experimental tests are conducted in Section 5. Finally, conclusions are given in Section 6.

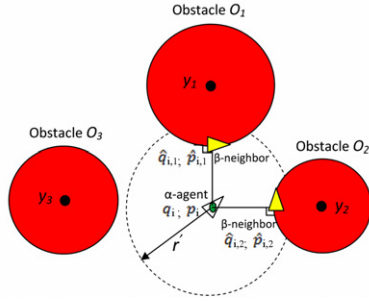


Fig. 1. The projection method to find the positions and velocities of  $\beta$ -neighbors of each  $\alpha$ -sensor.

## 2. Flocking control background

In this section, we present the flocking control background which establishes three basic flocking rules: no collision among agents, velocity matching among agents, and flocking centering. To describe a dynamic topology of flocks or swarms we consider a dynamic graph  $G(\mathcal{V}, E)$  consisting of a vertex set  $\mathcal{V} = \{1, 2, \dots, n\}$  and an edge set  $E \subseteq \{(i, j) : i, j \in \mathcal{V}, i \neq j\}$ . In this topology each vertex denotes one member of the flock, and each edge denotes the communication link between two members.

### 2.1. Actual and virtual neighbors

Let  $q_i, p_i \in R^m$  ( $m = 2$ ) be the position and velocity of node  $i$ , respectively. We know that during the motion of sensors, the relative distance between them may change, hence the neighbors of each sensor also change. Therefore, we define the neighborhood of sensor  $i$  at time  $t$  as follows:

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

here  $r$  is an interaction range (radius of neighborhood circle), and  $\|\cdot\|$  is the Euclidean distance. The superscript  $\alpha$  indicates the actual neighbors ( $\alpha$  neighborhood sensors) of sensor  $i$  that is used to distinguish with virtual neighbors ( $\beta$  neighborhood sensors) in the case of obstacle avoidance. The virtual neighbors are used to generate the repulsive force to push the sensors away from the obstacles.

The set of  $\beta$  neighbors (virtual neighbors) [13] of sensor  $i$  at time  $t$  with  $k$  obstacles is

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

here  $r'$  is obstacle detection range.  $\mathcal{V}_\beta$  is a set of obstacles.  $\hat{q}_{i,k}$  is the position of sensor  $i$  projecting on the obstacle  $k$ .

Now we explain how to find out  $\beta$  neighbors ( $\hat{q}_{i,k}, \hat{p}_{i,k}$ ) generated by each  $\alpha$  sensor. Firstly, we assume that the obstacles are convex regions in  $R^m$  with smooth boundaries, for example, circles or spheres with radius  $R_k$  and center  $y_k$ . We project each sensor to the obstacles and find out if the projections of sensors satisfy the condition  $\|\hat{q}_{i,k} - q_i\| \leq r'$ , and the obtained  $\hat{q}_{i,k}$  are  $\beta$  neighbors of sensor  $i$ . Eq. (3) illustrates the projection method to find the positions and velocities of the  $\beta$  neighbors generated by sensor  $i$ .

$$\hat{q}_{i,k} = \mu q_i + (1 - \mu) y_k, \quad \hat{p}_{i,k} = \mu P p_i \quad (3)$$

where  $\mu = R_k / \|q_i - y_k\|$ .  $P = I - a_k a_k^T$  is the projection matrix with  $a_k = (q_i - y_k) / \|q_i - y_k\|$  and a unit matrix or identity matrix  $I$ .

For example, as shown in Fig. 1, we have three obstacles  $O_1, O_2$  and  $O_3$ . After projecting  $\alpha$ -sensor  $i$  on all obstacles, we see that only two projections ( $\beta$ -neighbors) on the obstacles  $O_1$  and  $O_2$  satisfy

the condition (2). The obstacle  $O_3$  is out of active range  $r'$ , hence there is no projection of  $\alpha$ -sensor  $i$  on it. Consequently, we obtain two  $\beta$ -neighbors ( $\hat{q}_{i,1}, \hat{p}_{i,1}$ ) and ( $\hat{q}_{i,2}, \hat{p}_{i,2}$ ) for  $\alpha$ -sensor  $i$ .

### 2.2. Flocking control algorithm

We consider  $n$  sensors moving in a 2 dimensional Euclidean space. We assume that each sensor has a limited communication range to allow it to communicate with others and a large enough sensing range to make it sense the target. We also assume that each sensor is equipped with sonars or laser sensors that allow it to estimate the position and velocity of the target. The dynamic equation of each sensor is described as follows:

$$\begin{cases} \dot{q}_i = p_i \\ \dot{p}_i = u_i, \quad i = 1, 2, \dots, n. \end{cases} \quad (4)$$

The geometry of a flock is modeled by an  $\alpha$ -lattice [13] that has the following condition:

$$\|q_i - q_j\| = d, \quad j \in N_i^\alpha(t) \quad (5)$$

here  $d$  is a positive constant indicating the distance between sensor  $i$  and its neighbor  $j$ . The configuration which approximately satisfies the condition (5) is called a quasi  $\alpha$ -lattice, i.e.  $(\|q_i - q_j\| - d)^2 < \delta^2$ , with  $\delta \ll d$ . However, at singular configuration ( $q_i = q_j$ ) the collective potential is not differentiable. Therefore, the set of algebraic constraints in (5) is rewritten in terms of  $\sigma$ -norm [13] as follows:

$$\|q_j - q_i\|_\sigma = d^\alpha, \quad j \in N_i^\alpha(t) \quad (6)$$

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

In [13], Olfati-Saber proposed a control law for flocking of multiple mobile agents. This algorithm consists of three components as follows:

$$u_i = f_i^\alpha + f_i^\beta + f_i^\gamma. \quad (7)$$

The first component of Eq. (7)  $f_i^\alpha$  is used to regulate the gradient of potentials (impulsive or attractive forces) and the velocity among sensors. It consists of a gradient-based component and a consensus component (more details about these components see [43–45]).

$$f_i^\alpha = c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q) (p_j - p_i) \quad (8)$$

where each term in (8) is computed as follows [13]:

(1) The action function  $\phi_\alpha(z)$  which vanishes for all  $z \geq r_\alpha$  with  $r_\alpha = \|r\|_\sigma$  is used to construct a smooth pairwise attractive/repulsive potential function,  $\psi_\alpha(z) = \int_{d_\alpha}^z \phi_\alpha(s) ds$ , which is illustrated in Fig. 2. This action function  $\phi_\alpha(z)$  is defined as follows:

$$\phi_\alpha(z) = \rho_h(z/r_\alpha) \phi(z - d_\alpha) \quad (9)$$

where  $\phi(z)$  is the uneven sigmoidal function

$$\phi(z) = 0.5[(a + b)\sigma_1(z + c) + (a - b)] \quad (10)$$

here  $\sigma_1(z) = z / \sqrt{1 + z^2}$ , and parameters  $0 < a \leq b, c = |a - b| / \sqrt{4ab}$  to guarantee  $\phi(0) = 0$ , and constraints  $d_\alpha = \|d\|_\sigma$  with  $d = r/k$  for  $k$  being the scaling factor (in the simulations in this paper  $k = 1.2$ ).

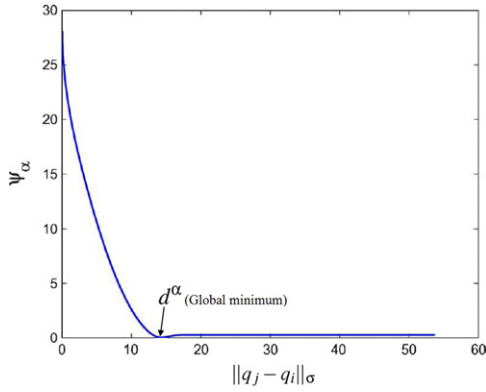


Fig. 2. Smooth pairwise potential function  $\Psi_\alpha(\|q_j - q_i\|_\sigma)$ .

The bump function  $\rho_h(z)$  with  $h \in (0, 1)$

$$\rho_h(z) = \begin{cases} 1, & z \in [0, h) \\ 0.5 \left[ 1 + \cos \left( \pi \left( \frac{z-h}{1-h} \right) \right) \right], & z \in [h, 1] \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

(2) The vector  $n_{ij}$  along the line connecting  $q_i$  and  $q_j$  is defined as

$$n_{ij} = (q_j - q_i) / \sqrt{1 + \epsilon \|q_j - q_i\|^2}. \quad (12)$$

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

$$a_{ij}(q) = \begin{cases} \rho_h(\|q_j - q_i\|_\sigma / r_\alpha), & \text{if } j \neq i \\ 0, & \text{if } j = i. \end{cases} \quad (13)$$

The second component of Eq. (7)  $f_i^\beta$  is used to control the mobile sensors to avoid obstacles,

$$f_i^\beta = c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q) (\hat{p}_{i,k} - p_i) \quad (14)$$

where each term in (14) is computed as follows [13]:

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

$$\phi_\beta(z) = \rho_h(z/d_\beta) (\sigma_1(z - d_\beta) - 1). \quad (15)$$

(2) Vector  $\hat{n}_{i,k}$  along the line connecting  $q_i$  and  $\hat{q}_{i,k}$  is defined as

$$\hat{n}_{i,k} = (\hat{q}_{i,k} - q_i) / \sqrt{1 + \epsilon \|\hat{q}_{i,k} - q_i\|^2}. \quad (16)$$

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

$$b_{i,k}(q) = \rho_h(\|\hat{q}_{i,k} - q_i\|_\sigma / d_\beta) \quad (17)$$

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

The third component of (7)  $f_i^\gamma$  is for distributed navigational feedback.

$$f_i^\gamma = -c_1^\gamma \sigma_1(q_i - q_\gamma) - c_2^\gamma (p_i - p_\gamma) \quad (18)$$

where  $\sigma_1(q_i - q_\gamma) = (q_i - q_\gamma) / \sqrt{1 + \|q_i - q_\gamma\|^2}$ , and the  $\gamma$ -sensor  $(q_\gamma, p_\gamma)$  is the virtual leader [46] defined as follows

$$\begin{cases} \dot{q}_\gamma = p_\gamma \\ \dot{p}_\gamma = f_\gamma(q_\gamma, p_\gamma). \end{cases} \quad (19)$$

The constants of three components used in (7) are chosen as  $c_1^\alpha < c_1^\beta < c_1^\gamma$ , and  $c_2^\nu = 2\sqrt{c_1^\nu}$ . Here  $c_\eta^\nu$  are positive constants for  $\forall \eta = 1, 2$  and  $\nu = \alpha, \beta, \gamma$ .

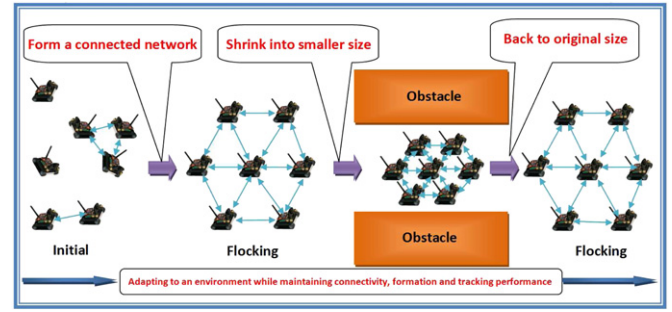


Fig. 3. Illustration of the adaptive flocking control.

### 3. Adaptive flocking control for moving target tracking and observing

In this section, an adaptive flocking control algorithm is designed for a mobile sensor network to deal with complex environments while maintaining connectivity, tracking performance and similar formation.

#### 3.1. Problem formulation

We consider the  $\gamma$  sensor as a moving target. Hence, based on Olfati-Saber's flocking algorithm [13] we first design an algorithm with a dynamic  $\gamma$ -sensor.

$$\begin{aligned} u_i = & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q) (p_j - p_i) \\ & + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q) (\hat{p}_{i,k} - p_i) \\ & - c_1^{mt} (q_i - q_{mt}) - c_2^{mt} (p_i - p_{mt}) \end{aligned} \quad (20)$$

here  $q_{mt}$  and  $p_{mt}$  are the position and velocity of the moving target, respectively, and  $c_1^{mt}, c_2^{mt}$  are positive constants.

In reality, a mobile sensor network has to deal with changing or complex environments. For example, the sensors may have to track a target that moves through a narrow space among obstacles, so the sensors should be able to track the target while maintaining connectivity, tracking performance and similar formation. In this situation the existing flocking control algorithm (20) has some limitations which include:

1. Connectivity is lost because of the fragmentation phenomenon.
2. Low speed or getting stuck behind obstacles causes poor tracking performance.
3. Formation of the network is totally changed.

We develop a novel approach to this problem. In this approach, each sensor cooperatively learns the network's parameters to decide its size in a decentralized fashion so that the connectivity, tracking performance and formation can be improved when avoiding obstacles. The reason for maintaining connectivity and similar formation is that when the network shrinks to deal with complex environments, the neighbors of each sensor can be maintained. This allows the network to keep the same topology to reduce the complexity of control during the tracking process. One example of such flocking control is illustrated in Fig. 3.

#### 3.2. Adaptive flocking control

To control the size of the network, we need to control the set of algebraic constraints in Eq. (6), which means that if we want the

size of the network to be smaller then  $d^\alpha$  should be smaller. This raises the question of how small the size of the network should be reduced to and how to control the size in a decentralized and dynamic fashion.

To control the constraint  $d^\alpha$  one possible method is based on the knowledge of obstacles obtained by any sensor in the network, which will broadcast a new  $d^\alpha$  to all other sensors. However, it is difficult for a single sensor to learn the size of the obstacles due to its limited sensing range. To overcome this problem we propose a method based on the repulsive force,  $\sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma)$ , which is generated by the  $\beta$ -sensors projected on the obstacles. If any sensor of the network gets this repulsive force it will shrink its own  $d_i^\alpha$ . If this repulsive force is big (the sensor is close to obstacles)  $d_i^\alpha$  will be further reduced. Then, in order to maintain the neighbors the active range of each sensor is changed. To create the agreement on the relative distance and active range among sensors in a decentralized way, a consensus or a local average update law is proposed. Furthermore, to maintain connectivity each sensor is designed with an adaptive weight of attractive force from the target and an adaptive weight of interaction force from its neighbors so that the network reduces or recovers the size gradually. In other words, if an sensor has a weak connection to the network it should have a big weight of attraction force to the target and a small weight of interaction force from its neighbors.

Firstly, we control the set of algebraic constraints as

$$\|q_j - q_i\|_\sigma = d_i^\alpha, \quad j \in N_i^\alpha, \quad (21)$$

and let each sensor have its own  $d_i^\alpha$ , which is designed as

$$d_i^\alpha = \begin{cases} d^\alpha, & \text{if } \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) = 0 \\ \frac{d^\alpha}{c_a \left( \sum_{k \in N_i^\beta} |\phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma)| + 1 \right)}, & \text{else} \end{cases} \quad (22)$$

here  $c_a$  is the positive constant. From Eq. (22) we see that if the repulsive force generated from the obstacles  $\sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma)$  equals to zero or  $N_i^\beta \in \emptyset$  (empty set) then the sensor will keep its original  $d^\alpha$ . When the sensor senses the obstacles it reduces its own  $d_i^\alpha$ , and the value of  $d_i^\alpha$  depends on the repulsive force that the sensor gets from obstacles.

In order to control the size of the network each sensor needs its own  $r_i^\alpha$  that relates to  $d_i^\alpha$  as follows:  $r_i^\alpha = \|kd\|_\sigma$  with  $\|d\|_\sigma = d^\alpha$  or  $d = \sqrt{\frac{(\epsilon d_i^\alpha + 1)^2 - 1}{\epsilon}}$ . Explicitly,  $r_i^\alpha$  is computed as

$$r_i^\alpha = \begin{cases} r^\alpha, & \text{if } \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) = 0 \\ \frac{1}{\epsilon} \left[ \sqrt{k^2 \frac{(\epsilon d_i^\alpha + 1)^2 - 1}{\epsilon}} + 1 - 1 \right], & \text{else.} \end{cases} \quad (23)$$

Similarly,  $r_i$  can be computed as

$$r_i = \begin{cases} r, & \text{if } \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) = 0 \\ \sqrt{\frac{1}{\epsilon} [(\epsilon r_i^\alpha + 1)^2 - 1]}, & \text{else.} \end{cases} \quad (24)$$

It should be pointed out that the active range  $r_i$  is different from the physical communication (sensing) range. The active range is

the range that each sensor uses to decide its neighbors, but the physical communication range is the range determined by the RF module. This implies that even though a sensor can communicate with all other sensors in the network, it will only talk (interact) with sensors in its active range. That is why we want to control the active range of each sensor in order to reduce communication and maintain a similar formation when the network shrinks.

To achieve agreement on  $d_i^\alpha$ ,  $r_i^\alpha$  and  $r_i$  among sensors in the connected network we use the following updating equations based on the local average for  $d_i^\alpha$ ,  $r_i^\alpha$  and  $r_i$ :

$$\begin{cases} d_i^\alpha = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} d_j^\alpha \\ r_i^\alpha = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} r_j^\alpha \\ r_i = \frac{1}{|N_i^\alpha \cup \{i\}|} \sum_{j=1}^{|N_i^\alpha \cup \{i\}|} r_j \end{cases} \quad (25)$$

here  $|N_i \cup \{i\}|$  is the number of agents in agent  $i$ 's local neighborhood plus agent  $i$  itself.

The center of mass (CoM) of positions and velocities of all sensors respectively is defined as follows:

$$\begin{cases} \bar{q} = \frac{1}{n} \sum_{i=1}^n q_i \\ \bar{p} = \frac{1}{n} \sum_{i=1}^n p_i. \end{cases} \quad (26)$$

In addition, to better maintain the network connectivity each sensor should have an adaptive weight for attractive force from the target and interaction force from its neighbors as discussed before. Firstly, in the control protocol (20), the first two terms are used to control the formation (velocity matching, collision avoidance among sensors). The third and fourth terms are used to allow sensors to avoid obstacles, and the last term is used for target tracking. If the last term is absent the control will lead to network fragmentation [13]. The coefficients of the interaction forces  $(c_1^\beta, c_2^\beta)$ ,  $(c_1^\alpha, c_2^\alpha)$  and attractive force  $(c_1^{mt}, c_2^{mt})$  are used to adjust the weight of interaction forces and attractive force. The bigger  $(c_1^{mt}, c_2^{mt})$  the faster the convergence to the target. However if  $(c_1^{mt}, c_2^{mt})$  is too big the center of mass (CoM) as defined in Eq. (26) oscillates around the target, and the formation of the network is not guaranteed. In addition, in order to guarantee that no sensor hits obstacles the pair  $(c_1^\beta, c_2^\beta)$  is selected to be bigger than the other two pairs,  $(c_1^\alpha, c_2^\alpha)$  and  $(c_1^{mt}, c_2^{mt})$ . Finally we have the relationship among these pairs as:  $(c_{1,2}^\alpha < c_{1,2}^{mt} < c_{1,2}^\beta)$ .

From the above analysis we see that these adaptive weights allow the network to reduce and recover the size gradually. This also allows the network to maintain connectivity during obstacle avoidance. In the  $\alpha$ -lattice configuration if the sensor has less than 3 neighbors it is considered to have a weak connection to the network. This means that this sensor is on the border of the network, or far from the target hence it should have bigger weight of attractive force from its target and smaller weight of interaction forces from its neighbors to get closer to the target. From this analysis  $c_{1,2}^\alpha$  and  $c_{1,2}^{mt}$  of each sensor are designed as follows:

$$c_1^\alpha(i) = \begin{cases} c_1^\alpha, & \text{if } |N_i^\alpha| \geq 3 \\ c_1^{\alpha'}, & \text{if } |N_i^\alpha| < 3 \end{cases} \quad (27)$$

here  $c_1^{\alpha'} < c_1^\alpha$ ,  $c_2^\alpha(i) = 2\sqrt{c_1^\alpha(i)}$ , and  $i = 1, 2, \dots, n$ .

$$c_1^{mt}(i) = \begin{cases} c_1^{mt}, & \text{if } |N_i^\alpha| \geq 3 \\ c_1^{mt'}, & \text{if } |N_i^\alpha| < 3 \end{cases} \quad (28)$$

here  $c_1^{mt'} > c_1^{mt}$ ,  $c_2^{mt}(i) = 2\sqrt{c_1^{mt}(i)}$ , and  $i = 1, 2, \dots, n$ .

Hence, the neighborhood of sensor  $i$  at time  $t$  ( $N_i^\alpha(t)$ ), the new adjacency matrix [ $a'_{ij}(q)$ ] and the new action function  $\phi'_\alpha(z)$  are defined as:

$$N_i^\alpha(t) = \{j \in \vartheta : \|q_j - q_i\| \leq r_i, \vartheta = \{1, 2, \dots, n\}, j \neq i\} \quad (29)$$

$$[a'_{ij}(q)] = \begin{cases} \rho_h(\|q_j - q_i\|/r_i^\alpha), & \text{if } j \neq i \\ 0, & \text{if } j = i \end{cases} \quad (30)$$

$$\phi'_\alpha(\|q_j - q_i\|_\sigma) = \rho_h(\|q_j - q_i\|_\sigma/r_\alpha)\phi(\|q_j - q_i\|_\sigma - d_i^\alpha). \quad (31)$$

Finally, the adaptive flocking control law for dynamic target tracking is

$$\begin{aligned} u_i = & c_1^\alpha(i) \sum_{j \in N_i^\alpha} \phi'_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha(i) \sum_{j \in N_i^\alpha} a'_{ij}(q)(p_j - p_i) \\ & + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i) \\ & - c_1^{mt}(i)(q_i - q_{mt}) - c_2^{mt}(i)(p_i - p_{mt}). \end{aligned} \quad (32)$$

#### 4. Multiple dynamic target tracking and observing

In many surveillance applications mobile sensor networks have to deal with the dynamic situation of targets appearing and disappearing in the field. In this section we first address the problem of sensor network partitioning and then discuss multiple dynamic target tracking through sensor splitting and merging.

##### 4.1. Sensor network partitioning

To deal with an emerging target, the sensor network should automatically decompose into equal sub-groups. It is desirable that the formation of each sub-group is maintained during splitting, and the total energy and time consumption is minimized. Then each sub-group will be assigned to track one target. For example, consider  $M$  targets at time  $t$  and  $M$  sensor groups ( $G_1, G_2, \dots, G_M$ ) which are tracking these targets (each group has about  $n/M$  sensors). If the  $(M + 1)$ th target appears then  $\frac{n}{M+1}$  sensors should split off from these existing groups to form a new group to track the new target. On the other hand to deal with a disappearing target, the sensors which are tracking this target should split and merge with the existing groups.

As discussed in Section 2, the mobile sensor network can be considered as a dynamic graph (dynamic topology). Hence we can apply graph partitioning algorithms to decompose the graph into sub-graphs (sub-groups). However, many existing methods for graph partitioning are centralized methods, which means that each sensor need global knowledge of the whole network's state, which is not practical. There are also some distributed graph partitioning or distributed graph clustering methods, but they are usually based on the density of node's distribution. Hence the size of sub-groups is not predetermined, and the number of sensors in each sub-group is different.

This paper proposes a seed growing graph partition (SGGP) algorithm to decide which sensor in the network should track new targets. This means that the mobile sensor which is closest

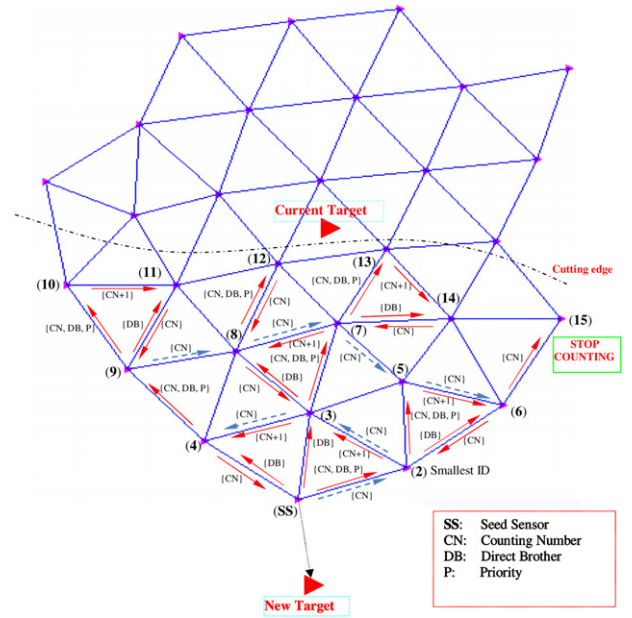


Fig. 4. Example of seed growing graph partition.

to the new target will initiate the growth of the sensors into a new group by broadcasting the message to its sons in a recursive fashion until the number of sensors in the subgroup is equal to a predetermined threshold ( $\vartheta_s$ ). By growing the number of sensors in each generation from the seed sensor, the formation of each subgroup is maintained during splitting. This leads to minimized total energy and time consumption.

Assume all mobile sensors already formed a network with an  $\alpha$ -lattice configuration (see Fig. 4). In this configuration if the sensor has 6 neighbors (6 is the maximum number of neighbors in this configuration) this sensor will be inside the network. If the sensor has less than or equal to 5 neighbors it will be on the border of the network. This sensor is called a border sensor. Based on this fact, the SGGP algorithm has the following steps:

- Step 1. Each sensor checks to find how many neighbors it has and decides if it is a border sensor.
- Step 2. Each border sensor computes the distance to the new target and forwards this distance information to other border sensors, and receives the distances from other border sensors.
- Step 3. Each border sensor compares its distance with the received distances from other border sensors and finds the sensor with smallest distance to be set as the Seed Sensor (SS).
- Step 4. The SS counts its sons and broadcasts the predetermined size of the new group to its sons. If the size of the new group is less than the predetermined size the sons will continue passing the message to their sons. This process is repeated until the size of the new group is equal to the predetermined size.

**Remark.** In the SGGP algorithm, the number of sons of sensor  $i$  is defined as:

$$|S_i| = |N_i| - |F_i| - |DB_i| \quad (33)$$

here  $|S_i|$ ,  $|N_i|$ ,  $|F_i|$  and  $|DB_i|$  are the number of sons, neighbors, fathers and direct brothers of sensor  $i$ , respectively. For example in Fig. 4, SS is the father of sensors 2, 3 and 4. Sensor 3 is the direct brother of sensor 2, hence the sons of sensor 2 are only sensors 5 and 6. Sensor 2 can know sensor 3 being its direct brother because its father (SS) sends a message  $\{DB\}$  to tell which sensor is its direct

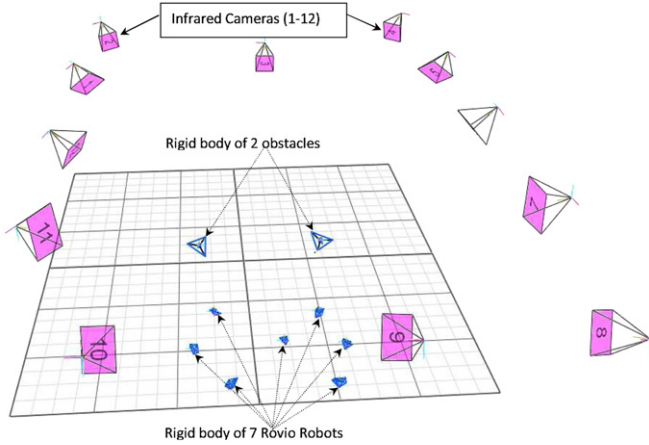


Fig. 5. Experimental setup for adaptive flocking control.

brother. In addition, two or more sensors can have the same son, but if a sensor has the priority  $\{P\}$  to count this same son first the other sensors will not count this son again. As an example of this situation, sensors 2 and 3 have the same son, sensor 5, but because of its smaller ID sensor 2 receives a message consisting of  $\{P\}$  from its father (SS) hence it has priority to count sensor 5 as its son first then it sends the counting number (CN) to its direct brother sensor 3. Fig. 4 also shows the message exchange when applying the SGGP algorithm. The dashed green arrows represent the counting number (CN), and the solid red arrows represent the message exchange.

#### 4.2. Multiple dynamic target tracking

In the multiple targets scenario, to track targets  $(q_{mt_l}, p_{mt_l})$  with  $l = 1, 2, \dots, M$ , we have the following flocking control law,

$$u_i = c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i) - c_1^{mt} (q_i - q_{mt_l}) - c_2^{mt} (p_i - p_{mt_l}). \quad (34)$$

As discussed in Section 2, the dynamic target  $(q_{mt_l}, p_{mt_l})$  in (34) is exactly the navigation term that makes the mobile sensors move together. Without this term the sensor network leads to fragmentation. This means that if sensor  $i$  is assigned to track another target it only need switch to another navigation term. On the other hand, in the merging case, three sensor subgroups are tracking three targets. If one of these targets disappears then this subgroup will decompose into two equal parts and each will merge into one of remaining subgroups to track the existing targets by switching to another navigation term.

### 5. Test and evaluation

In this section we test and evaluate the two algorithms: the adaptive flocking control algorithm for single target tracking and the SGGP algorithm for multi-target tracking.

#### 5.1. Adaptive flocking control

We first conduct simulation of the adaptive flocking control. The parameters of flocking in simulation are: number of sensors = 50 randomly distributed in the box  $100 \times 100$ . Parameters

$a = b = 5$ ;  $d = 7$ ; the scaling factor  $k = 1.2$  the active range  $r = k * d = 8.4$ ;  $\epsilon = 0.1$  for the  $\sigma$ -norm;  $h = 0.2$  for the bump function  $(\phi_\alpha(z))$ ;  $h = 0.9$  for the bump function  $(\phi_\beta(z))$ . The target moves in the line trajectory:  $q_{mt} = [100 + 130t, 1t]^T$  with  $0 \leq t \leq 3.5$ , and  $p_{mt} = (q_{mt}(t) - q_{mt}(t-1))/\Delta_t$  with step size  $\Delta_t = 0.002$ .

We also conduct experiments on adaptive flocking control. In the experiment we use 7 Rovio robots [47] that have omnidirectional motion capability. The dynamic model of the Rovio robot can be approximated by Eq. (4). However, the localization accuracy 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 12-camera VICON motion capture system [48] in our lab (Fig. 5) to track the robots.

The parameters of flocking in the experiment are:  $a = b = 5$ ;  $d = 1100$  mm; the scaling factor  $k_c = 1.2$ ; the active range  $r = k_c * d$ ;  $\epsilon = 0.1$  for the  $\sigma$ -norm;  $h = 0.2$  for the bump functions  $(\phi_\alpha(z), \phi'_\alpha(z))$ ;  $h = 0.9$  for the bump function  $(\phi_\beta(z))$ . The virtual target moves in the line trajectory:  $q_t = [230+t, -3000+130t]^T$ .

To analyze the connectivity of the network we define a connectivity matrix  $c_{ij}(t)$  as follows:

$$c_{ij}(t) = \begin{cases} 1, & \text{if } j \in N_i^\alpha(t), i \neq j \\ 0, & \text{if } j \notin N_i^\alpha(t), i \neq j \end{cases} \quad (35)$$

and  $c_{ii} = 0$ .

Because the rank of Laplacian of a connected graph [13],  $c_{ij}(t)$ , of order  $n$  is at most  $(n-1)$  or  $\text{rank}(c_{ij}(t)) \leq (n-1)$ , the relative connectivity of a network at time  $t$  is defined as

$$C(t) = \frac{1}{n-1} \text{rank}(c_{ij}(t)). \quad (36)$$

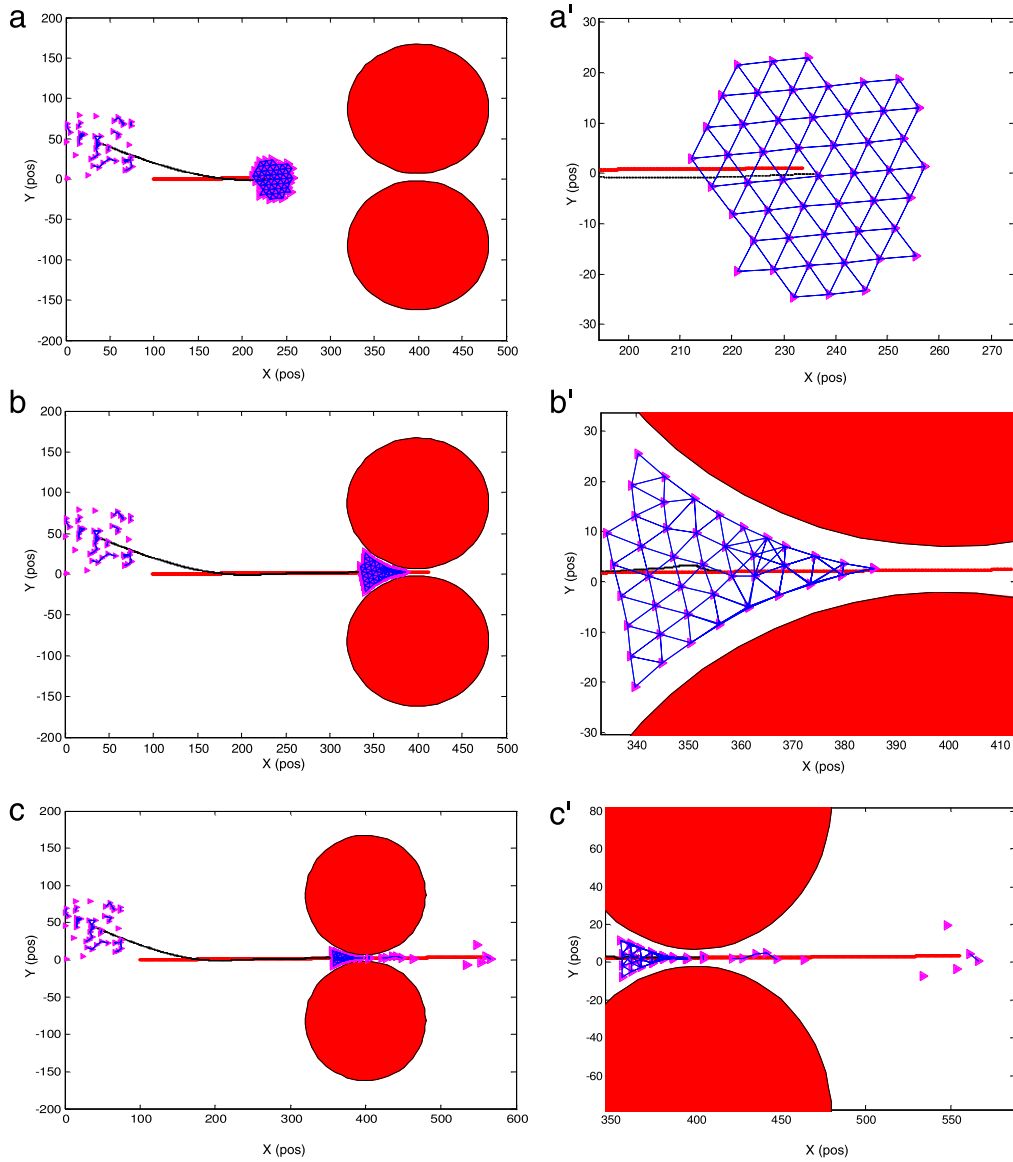
If  $0 \leq C(t) < 1$  the network is broken, and  $C(t) = 1$  the network is connected. Based on this metric we can evaluate the adaptive flocking control algorithm (32).

Fig. 6 represents the results of moving target tracking in the line trajectory using the normal flocking control algorithm (7). Fig. 7 represents the results of moving target tracking in the line trajectory using the adaptive flocking control algorithm (32). It can be seen that when the network enters the small gap between two obstacles its size shrinks in order to pass this space, and the network size grows back after it passes. Therefore the connectivity and similar formation are maintained.

Fig. 8 shows the results of velocity matching among sensors (a, a'), connectivity (b, b') and error positions between the CoM and the target (c, c') of both flocking control algorithms (32) and (7). To compare these algorithms we use the same initial state (position and velocity) of mobile sensors. By comparing these figures we see that by applying the adaptive flocking control algorithm (32) the connectivity, tracking performance and similar formation are maintained when the network passes through the narrow space while the normal flocking control algorithm (7) could not handle these problems.

Fig. 9 shows the average of the velocity matching among sensors, connectivity and tracking error, respectively for 10 trials. We see that the results in Fig. 9 are similar to the ones in Fig. 8 (a, b).

Fig. 10 shows the snapshots of the experiment result for 7 Rovio robots using our adaptive flocking algorithm (32). The results look similar to the simulation results in Fig. 7. The left part of Fig. 11 shows the trajectories of 7 robots in simulation, and the right part of Fig. 11 compares the trajectories of 7 robots in both simulation and experiment.



**Fig. 6.** Snapshots of the mobile sensor network (a) when the mobile sensors form a network; (b) when the mobile sensors avoid obstacles; (c) when the mobile sensors get stuck in the narrow space between two obstacles. (a', b', c') are a closer look of (a, b, c), respectively. These results are obtained by using algorithm (7).

5.2. Multiple target tracking

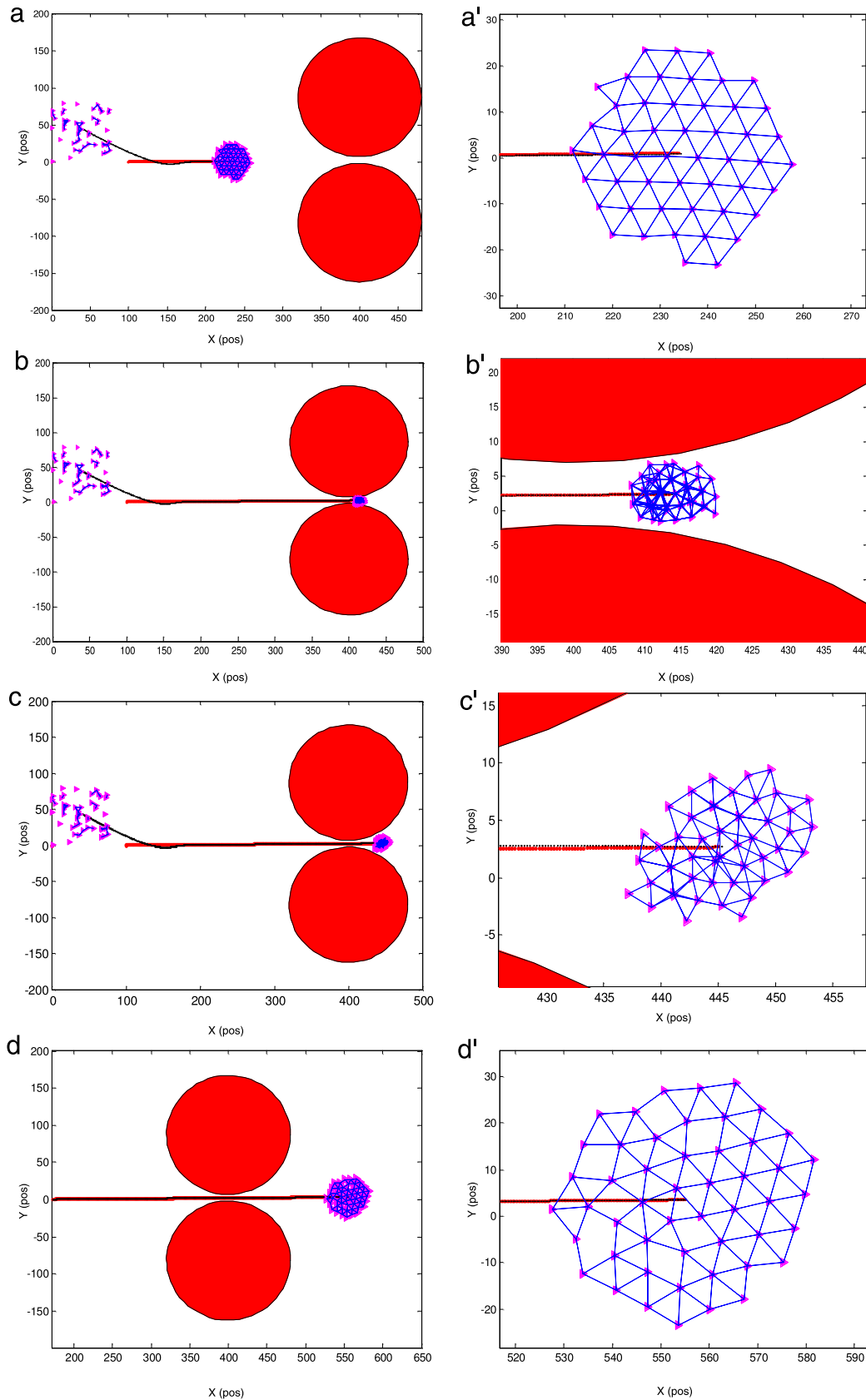
We tested the SGGP algorithm and the flocking control algorithm (34) in two different cases of sensor splitting and merging. Parameters used in this simulation are specified as follows:

Case 1. Two targets appear one by one and no target disappears.

The parameters of flocking are: Number of sensors = 120 (randomly distributed in the square area with the size of  $90 \times 90$ ). The communication range  $r = 1.2 * d$  with  $d = 7.5$ ;  $\epsilon = 0.1$  for the  $\sigma$ -norm, and other parameters are chosen the same with those in Section 5.1. The targets move in the sine wave trajectory: For the target 1,  $q_{mt_1} = [50 + 35t, 295 - 35 \sin(t)]^T$  with  $0 \leq t \leq 8.5$ , and for the target 2,  $q_{mt_2} = [85 + 35t, 55 - 35 \sin(t)]^T$  with  $1.26 \leq t \leq 8.5$ , and  $\Delta_t = 0.002$  is the step size.

Fig. 12(a) displays the result of tracking where the targets appear one by one and move in a sine wave trajectory. Firstly, the whole group of 120 mobile sensors form an  $\alpha$ -lattice configuration and track target 1. Then, at step 840 target 2 appears and the

network decides which sensors will split and track this target. By applying the SGGP algorithm, the sensor network automatically decomposes into 2 equal sub-groups (60 sensors in each sub-group). The second sub-group which is closest to target 2 tracks target 2, and the first sub-group keeps tracking target 1. The SGGP algorithm allows two sub-groups to maintain their formation when they split. Fig. 12(b) shows the error between the average of positions in the whole network and target 1 (from step 1 to 839), and the error between the average position in sub-group 1 and target 1 (from step 840 to the end). Fig. 12(c) represents the error between the average of positions in sub-group 2 and target 2. We see that at step 840, the average of positions of sensors slightly changes because at this time the average sensors's positions in sub-group 1 will replace that of the whole network. In this figure we see that all tracking errors are small in free space. However in the presence of obstacles, the errors are significant because the repulsive forces generated from obstacles push the sensors away from them.



**Fig. 7.** Snapshots of the mobile sensor network (a) when the mobile sensors form a network; (b) when the mobile sensors avoid obstacles; (c) when the mobile sensors successfully pass through the narrow space between two obstacles; (d) when the mobile sensors recover the original size. (a', b', c', d') are a closer look of (a, b, c, d), respectively. These results are obtained by using algorithm (32).

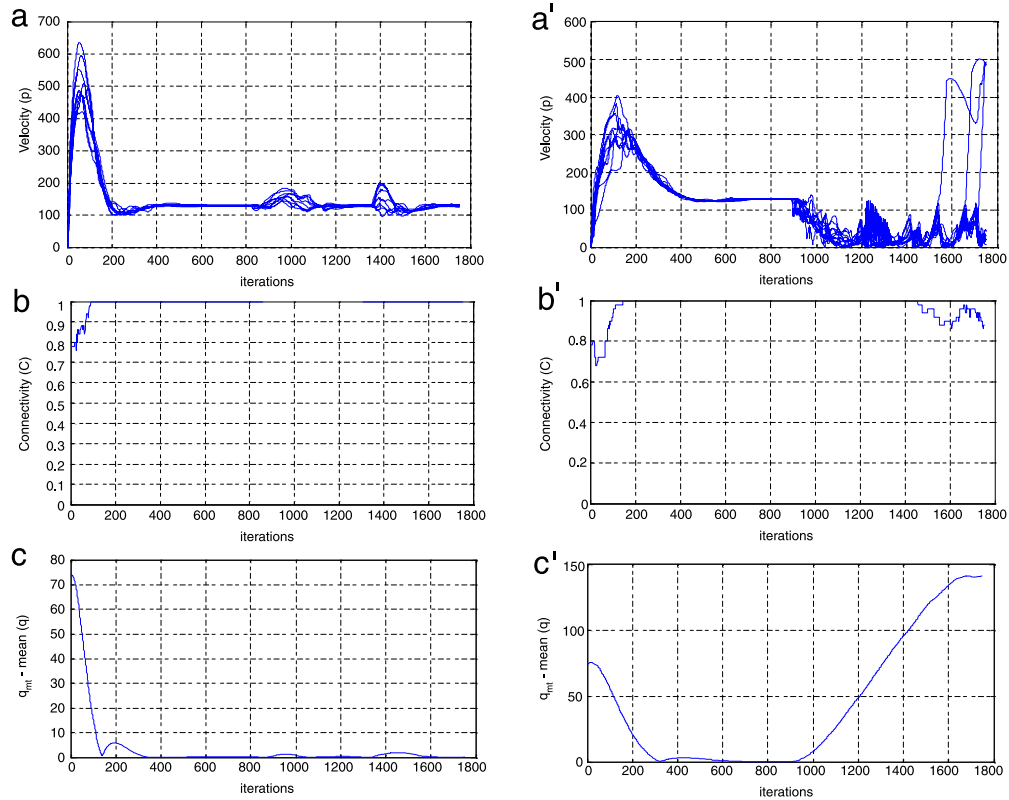


Fig. 8. Velocity matching among sensors, connectivity, and error of positions between the CoM and the moving target in (a, b, c) using algorithm (32), and (a', b', c') using algorithm (7), respectively.

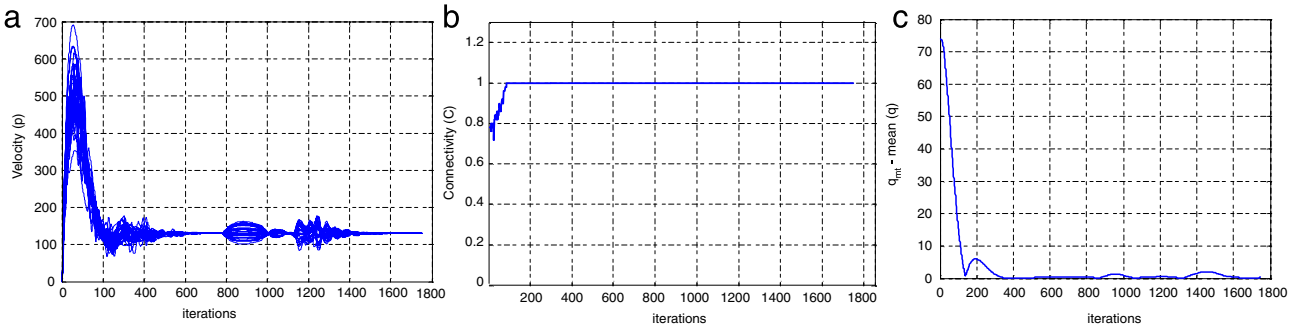


Fig. 9. (a) Average of the velocity matching among sensors for 10 trials; (b) Average of the connectivity for 10 trials; (c) Average of the tracking error for 10 trials, using algorithm (32).

Case 2. Two targets appear one by one and one target disappears.

The parameters of flocking are the same as those in Case 1. The parameters of target movement are the same as in Case 1, but target 1 is set to run in the interval time  $0 \leq t \leq 12.5$ , and target 2 appears at time  $t = 1.26$  (at step 840) and disappears at time  $t = 8.4$  (at step 4200).

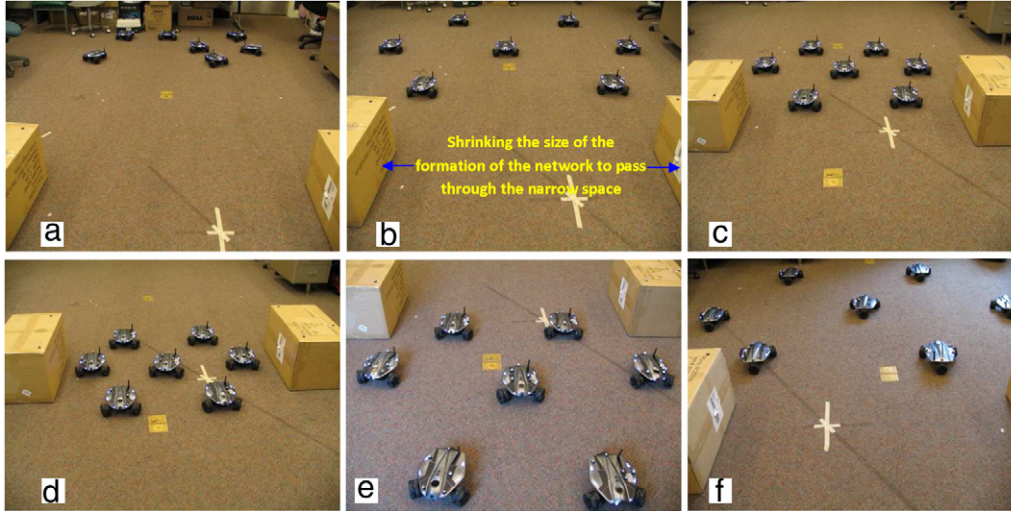
Fig. 13 shows the results of tracking when the targets appear one by one and then one disappears. When target 2 appears at step 840 the results are similar with Fig. 12. When target 2 disappears at step 4200 sub-group 2 which is tracking this target will rejoin to sub-group 1 and continue to track target 1. The tracking result of the whole group after merging is good with small tracking error between the average of sensors's positions and target 1 in the free space, which can be seen in Fig. 13(b) (from step 4200 to the end).

### 5.2.1. Comparison between the SGGP algorithm and the random selection (RS) algorithm

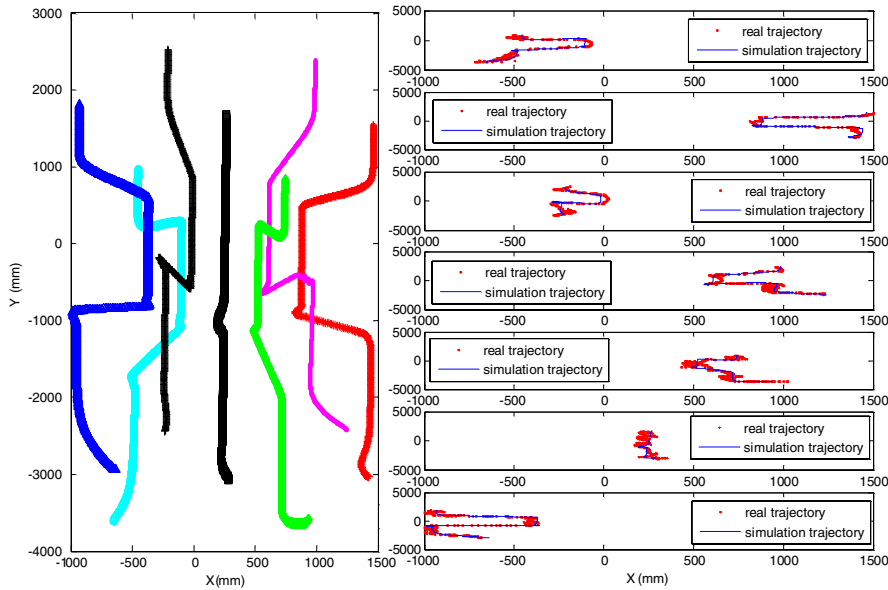
Here we compare the SGGP algorithm with the random selection (RS) algorithm, in terms of tracking time, formation time, and total distance of all sensors for each sub-group to track its target. In the RS algorithm when the new target appears half of the sensors in the network are randomly selected to track the new target. With this algorithm two sub-groups do not maintain their formation, and all sensors in each sub-group need certain time to reform a network. This is the main drawback of this algorithm.

We first define the following metrics to compare the two algorithms.

- (1)  $D_{\text{tt}}$  is the total travel distance for all sensors in each group to track its target. It is measured from when the network is decomposed into sub-groups to when the average of positions of sensors in each sub-group reaches the target (this is



**Fig. 10.** Snapshots of adaptive flocking control with 7 Rovio robots using our adaptive flocking control algorithm (32). (a) seven robots are randomly distributed. (b) Seven robots form a lattice formation. (c) Seven robots begin to shrink the size of the network. (d) Seven robots pass through the narrow space between 2 obstacles. (e) Seven robots begin to recover the size of the network. (f) Seven robots completely recover the size of the network.



**Fig. 11.** Trajectories of 7 robots using the adaptive flocking control algorithm (32).

evaluated based on the same condition as used to compute  $t_T$  below).

- (2)  $t_T$  is the tracking time which is computed based on the following condition:  $\|\frac{1}{n_{G_l}} \sum_{i=1}^{n_{G_l}} q_i - q_l\| \leq \Theta_T$ ,  $l = 1, 2$ ; here  $n_{G_l}$  is number of sensors in each sub-group, and  $\Theta_T$  is a given threshold.
- (3)  $t_F$  is the formation time representing the time that it costs all mobile sensors to form a network. This formation time is computed based on the following condition:
- (4)  $\text{Var}(\|q_i - q_j\|) = \frac{1}{|E_l|} \sum (\|q_i - q_j\| - \frac{1}{n_{G_l}} \sum_{(i,j) \in E_l} \|q_i - q_j\|)^2 \leq \Theta_3$  with  $i, j = 1, 2, \dots, n_{G_l}$ ;  $l = 1, 2$ ; here  $\Theta_F$  is a given threshold, and  $i \neq j$ .

Based on the above metrics, the SGGP algorithm and the RS algorithm are compared in Table 1.

The values of  $D_{tt}$ ,  $t_T$ , and  $t_F$  are obtained based on the average value of 50 running times. From the above table it can be seen that

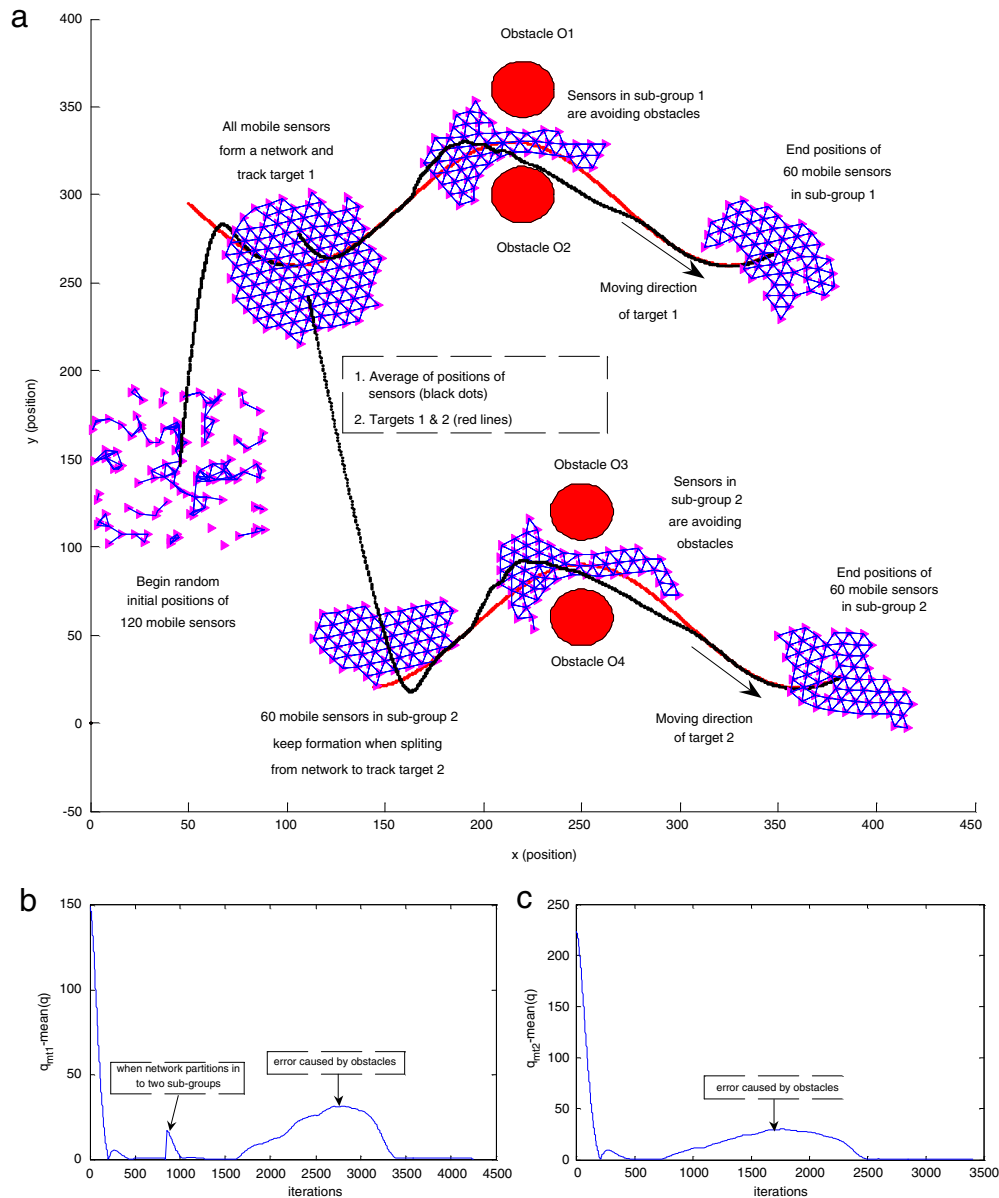
**Table 1**

Comparison between two algorithms (SGGP and RS).

Algorithms	$D_{tt}$ (units)	$t_T$ (s)	$t_F$ (s)
RS ( $G_1$ )	1184.7	1.000801	8.345623
RS ( $G_2$ )	14194	11.770489	11.125117
SGGP( $G_1$ )	1185.6	1.203569	0.0
SGGP( $G_2$ )	13126	9.007456	0.0

the maximum of the tracking time and formation time in the SGGP algorithm  $t_{SGGP}^{\max} = \max(t_T, t_F)_{G_1} + \max(t_T, t_F)_{G_2} = 10.211(s)$  while in the RS algorithm  $t_{RS}^{\max} = 20.1161(s)$ , or  $t_{SGGP}^{\max}$  is 49.28% less than  $t_{RS}^{\max}$ . The total distance in the SGGP algorithm  $D_{SGGP}^t = D_{tt}^{G_1} + D_{tt}^{G_2} = 14311.6(\text{units})$  while in the RS algorithm  $D_{RS}^t = 15378.7(\text{units})$ , or  $D_{SGGP}^t$  is 7% shorter than  $D_{RS}^t$ .

In all the above simulation and experiment results, all sensors keep their formation (excepting in the case of the RS algorithm)



**Fig. 12.** (a) Snapshots of the mobile sensor network when the mobile sensors are at the initial positions, forming a network at time  $t = 1.26$ , and decomposing into two sub-groups, respectively to track the targets moving in the sine wave trajectories. (b) Error between the average of sensor positions in the whole network and the moving target 1 (step 1 to 839), and between average of sensor positions in sub-group 1 and the moving target 1 (step 839 to the end). (c) Error between the average of sensor positions in sub-group 2 and the moving target 2. This result is obtained using the flocking control algorithm (34) and the SGGP algorithm.

and no collision occurs among them while tracking the moving target, and all sensors avoid obstacles successfully in a narrow space.

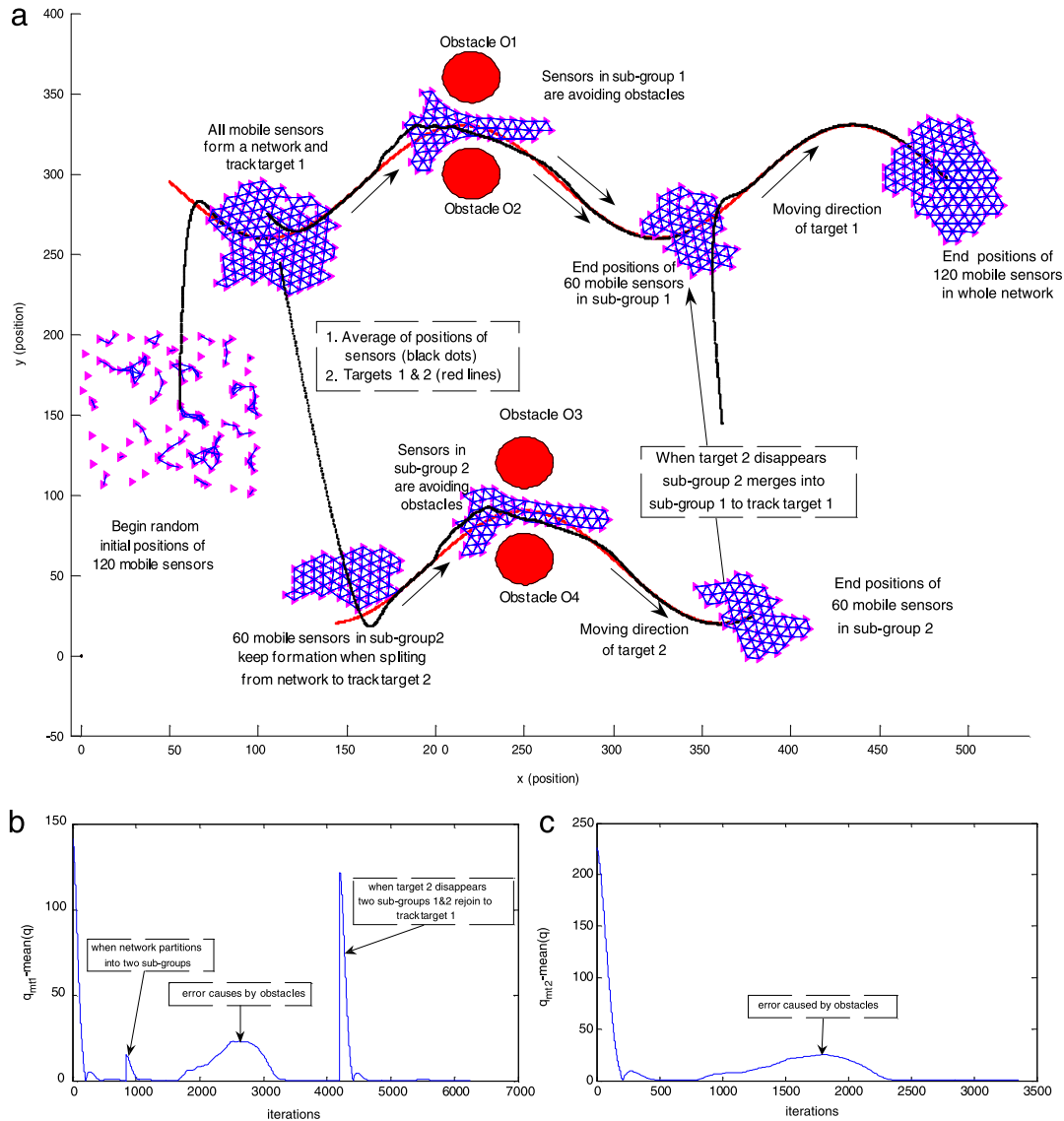
## 6. Conclusions

This paper studied the approach to flocking control of a mobile sensor network to track and observe both a single target and multiple targets. For single target tracking, to deal with complex environments the adaptive flocking control algorithm is proposed in which each sensor can cooperatively learn the network's parameters in a decentralized fashion to change the size of the network in order to maintain connectivity, tracking performance and similar formation when passing through a narrow space among obstacles. To see the benefit of the adaptive flocking algorithm we compared it with the regular flocking control algorithm, and we found that the connectivity, tracking

performance and similar formation in the adaptive flocking control algorithm are better than those in the regular flocking control algorithm. For multiple dynamic target tracking, the SGGP algorithm is proposed to solve the problem of splitting/merging the sensor agents from the network. Also, to demonstrate the benefit of this algorithm we compare it with the RS algorithm, and the results show that the maximum of the convergent distance and formation time in the SGGP algorithm is faster than that in the RS algorithm. In addition, the convergent distance in the SGGP algorithm is shorter than that in the RS algorithm. Several experimental tests were conducted with two different cases of splitting and merging sensor agents to demonstrate our theoretical results.

## Acknowledgments

We would like to thank the Vietnamese Government, especially the MOET (Ministry of Education and Training) under the 322 grant



**Fig. 13.** (a) Snapshots of the mobile sensor network when the mobile sensors are at the initial positions, when the mobile sensors form a network at time  $t = 1.26$ , when the mobile sensors decompose into two sub-groups, and when two sub-groups merge. (b) Error between the average of sensor positions in the whole network and the moving target 1 (step 1 to 839, and step 4200 to the end), and between the average of sensor positions in sub-group 1 and the moving target 1 (step 840 to 4200). (c) The error between the average of sensor positions in sub-group 2 and the moving target 2. This result is obtained using the flocking control algorithm (34) and the SGGP algorithm.

and the USA DoD DURIP under the 55628-CS-RIP grant, which supported us to implement this project.

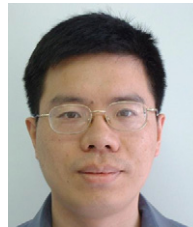
## References

- [1] F. Akyildiz, W. Su, Y. Sankarasubramanian, E. Cayirci, A survey of sensor networks, *IEEE Communications Magazine* (2002) 102–114.
- [2] D. Culler, D. Estrin, M. Srivastava, Overview of sensor networks, *IEEE Computer* 37 (8) (2004) 41–49.
- [3] V. Kumar, D. Rus, S. Singh, Robot and sensor networks for first responders, *IEEE Pervasive Computing* 3 (4) (2004) 24–33.
- [4] M. Tubaishat, S. Madria, Sensor networks: an overview, *IEEE Potentials* (2003) 20–23.
- [5] S. Kamath, E. Meisner, V. Isler, Triangulation based multi target tracking with mobile sensor networks, in: *IEEE International Conference on Robotics and Automation*, 2007, pp. 3283–3288.
- [6] A. Ganguli, S. Susca, S. Martinez, F. Bullo, J. Cortes, On collective motion in sensor networks: sample problems and distributed algorithms, in: *Proc. 44th IEEE Conf. Decision and Control, and the European Control Conference 2005*, December 12–15, 2005, pp. 4239–4244.
- [7] E. Biagioni, K. Bridges, The application of remote sensor technology to assist the recovery of rare and endangered species, *International Journal of High Performance Computing Applications* 16 (3) (2002) 112–121.
- [8] H.M. La, W. Sheng, Robust adaptive control with leakage modification for a nonlinear model of ionic polymer metal composites (IPMC), in: *Proceedings of the 2008 IEEE International Conference on Robotics and Biomimetics, ROBIO 2008*, Bangkok, Thailand, 2008.
- [9] A. Jadbabaie, J. Lin, A.S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules, *IEEE Transactions on Automatic Control* 48 (6) (2003) 988–1001.
- [10] Z. Lin, M.E. Broucke, B.A. Francis, Local control strategies for groups of mobile autonomous agents, *IEEE Transactions on Automatic Control* 49 (4) (2004) 622–629.
- [11] R.M. Murray, Recent research in cooperative control of multivehicle systems, *Journal of Dynamic Systems, Measurement, and Control* (2007) 571–583.
- [12] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, S.S. Sastry, Distributed control applications within sensor networks, *Proceedings of the IEEE* 91 (8) (2003) 1235–1246.
- [13] R. Olfati-Saber, Flocking for multi-agent dynamic systems: algorithms and theory, *IEEE Transactions on Automatic Control* 51 (3) (2006) 401–420.
- [14] C. Reynolds, Flocks, birds, and schools: a distributed behavioral model, *Computer Graphics, ACM SIGGRAPH'87 Conference Proceedings*, Anaheim, California, vol. 21 (4) pp. 25–34, 1987.
- [15] T. Vicsek, A. Czirok, E.B. Jacob, I. Cohen, O. Schochet, Novel type of phase transitions in a system of self-driven particles, *Physical Review Letters* 75 (1995) 1226–1229.
- [16] H. Levine, W.J. Rappel, I. Cohen, Self-organization in systems of self-propelled particles, *Physical Review E* 63 (2000) 017101–017104.

- [17] A. Mogilner, L. Edelstein-Keshet, L. Bent, A. Spiros, Mutual interactions, potentials, and individual distance in a social aggregation, *Journal of Mathematical Biology* 47 (2003) 353–389.
- [18] I.D. Couzin, J. Krause, R. James, G.D. Ruxton, N.R. Franks, Collective memory and spatial sorting in animal groups, *Journal of Theoretical Biology* 218 (2002) 1–11.
- [19] M. Anderson, E. McDaniel, S. Chenney, Constrained animation of flocks, in: *Eurographics/SIGGRAPH Symposium on Computer Animation*, 2003, pp. 1–12.
- [20] D. Cruz, J. McClintock, B. Perteet, O.A. Orqueda, Y. Cao, R. Fierro, Decentralized cooperative control—a multivehicle platform for research in networked embedded systems, *IEEE Control Systems Magazine* (2007) 58–78.
- [21] V. Gervasi, G. Prencipe, Coordination without communication: the case of the flocking problem, *Discrete Applied Mathematics* (2004) 324–344.
- [22] H. Su, X. Wang, Z. Lin, Flocking of multi-agents with a virtual leader part I: with a minority of informed agents, in: *Proc. 46th IEEE Conf. Decision and Control*, 2007, pp. 2937–2942.
- [23] H. Su, X. Wang, Z. Lin, Flocking of multi-agents with a virtual leader part II: with a virtual leader of varying velocity, in: *Proc. 46th IEEE Conf. Decision and Control*, 2007, pp. 1429–1434.
- [24] H. Shi, L. Wang, T. Chu, F. Fu, M. Xu, Flocking of multi-agents with a virtual leader, in: *Proc of the 2007 IEEE Symposium on Artificial Life*, 2007, pp. 287–294.
- [25] H.G. Tanner, A. Jadbabai, G.J. Pappas, Stable flocking of mobile agents, part I: fixed topology, in: *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 2010–2015.
- [26] H.G. Tanner, A. Jadbabai, G.J. Pappas, Stable flocking of mobile agents, part II: dynamic topology, in: *Proceedings of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 4917–4922.
- [27] A. Regmi, R. Sandoval, R. Byrne, H. Taner, C.T. Abdallah, Experimental implementation of flocking algorithms in wheeled mobile robots, in: *American Control Conference*, 2005, pp. 4917–4922.
- [28] Z. Wang, D. Gu, A survey on application of consensus protocol to flocking control, in: *Proc of the 12th Chinese Automation and Computing Society Conference in the UK, England*, 2006, pp. 1–8.
- [29] R. Olfati-Saber, Distributed tracking for mobile sensor networks with information driven mobility, in: *Proceedings of the 2007 American Control Conference*, 2007, pp. 4606–4612.
- [30] Y. Yang, N. Xiong, N.Y. Chong, X. Defago, A decentralized and adaptive flocking algorithm for autonomous mobile robots, in: *The 3rd International Conference on Grid and Pervasive Computing—Workshops*, IEEE Computer Society, 2008, pp. 262–268.
- [31] G. Lee, N.Y. Chong, Adaptive self-configurable robot swarms based on local interactions, in: *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 4182–4187.
- [32] G. Lee, N.Y. Chong, Adaptive flocking of robot swarms: algorithms and properties, *IEICE Transactions on Communications* (9) (2008) 2848–2855.
- [33] B. Jung, G.S. Sukhatme, Tracking targets using multiple robots: the effect of environment occlusion, *Autonomous Robots Journal* 13 (3) (2002) 191–205.
- [34] Z. Tang, U. Ozguner, Motion planning for multitarget surveillance with mobile sensor agents, *IEEE Transactions on Robotics* 21 (5) (2005) 898–908.
- [35] A. Kolling, S. Carpin, Cooperative observation of multiple moving targets: an algorithm and its formalization, *The International Journal of Robotics Research* 26 (9) (2007) 935–953.
- [36] H. Ariyoshi, I. Shirakawa, H. Ozaki, Decomposition of graph into compactly connected two terminal subgraphs, *IEEE Transactions on Circuit Theory* 18 (4) (1971) 430–435.
- [37] G.W. Flake, R.E. Tarjan, K. Tsoutsoulis, Graph and minimum cut trees, *Internet Mathematics* 1 (4) (2004) 385–408.
- [38] L. Kaufman, P. Rousseeuw, *Finding Groups in Data. An Introduction to Cluster Analysis*, in: *Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics*, Wiley, New York, 1990.
- [39] M. Girvaan, M. Newman, Community structure in social and biological networks, *Proceedings of the National Academy of Sciences of the United States of America* 99 (12) (2002) 7821–7826.
- [40] B. Derbel, M. Mosbah, A fully distributed linear time algorithm for cluster network decomposition, in: *Proceeding of Parallel and Distributed Computing and Systems*, 2004.
- [41] A. Goebels, H.K. Buning, S. Priesterjahn, A. Weimer, Multi target partitioning of sets based on local information, in: *Proceedings of the fourth IEEE Workshop on Soft Computing as Transdisciplinary Science and Technology*, WSTST.
- [42] A. Goebels, H.K. Buning, S. Priesterjahn, A. Weimer, Towards online partitioning of agent sets based on local information, in: *Proceeding of Parallel and Distributed Computing and Networks*, 2005.
- [43] P. Ogren, E. Fiorelli, N.E. Leonard, Cooperative control of mobile sensor networks: adaptive gradient climbing in a distributed environment, *IEEE Transactions on Automatic Control* 49 (8) (2006) 1292–1302.
- [44] R. Olfati-Saber, R.M. Murray, Consensus problems in networks of agents with switching topology and time-delays, *IEEE Transactions on Automatic Control* 49 (9) (2004) 1520–1533.
- [45] R. Olfati-Saber, J. Alex Fax, R.M. Murray, Consensus and cooperative in networked multi-agent systems, *Proceedings of the IEEE* 95 (1) (2007) 215–233.
- [46] N.E. Leonard, E. Fiorelli, Virtual leaders, artificial potentials, and coordinated control of groups, in: *Proc. 40th IEEE Conf. Decision and Control*, 2001, pp. 2968–2973.
- [47] Rovio robot, 2011. <http://www.wowwee.com/en/support/rovio>.
- [48] VICON motion capture system, 2011. <http://www.vicon.com/>.



**Hung Manh La** is a post-doc at the Center for Advanced Infrastructure and Transportation, Rutgers University, USA, and a lecturer at the Department of Electronics Engineering, Thai Nguyen University of Technology, Vietnam. He received his Ph.D. 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 assistant professor at the School of Electrical and Computer Engineering, Oklahoma State University, USA. He received his Ph.D. in Electrical and Computer Engineering from Michigan State University in May 2002, 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 was promoted to associate professor there before he joined Oklahoma State University. He is a senior member of IEEE and has participated in organizing many IEEE international conferences and workshops in the area of intelligent robots and systems. He is the author of one US patent and more than 110 papers in major journals and international conferences. His current research interests include wearable computing, mobile robotics, human robot interaction, 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.