Noname manuscript No.
(will be inserted by the editor)

Tracking Multiple Moving Targets with Swarms of Mobile Robots

Received: date / Accepted: date

Abstract This paper presents a distributed approach to enabling mobile robot swarms to track multiple targets moving unpredictably. The proposed approach consists of two constituent algorithms: local interaction and target tracking. When the robots are faster than the targets, Lyapunov theory can be applied to show that the robots converge asymptotically to each vertex of the desired equilateral triangular configurations while tracking the targets. Toward practical implementation of the algorithms, it is important to realize the observation capability of individual robots in an inexpensive and efficient way. A new proximity sensor that we call dual rotating infrared (DRIr) sensor is developed to meet these requirements. Both our simulation and experimental results employing the proposed algorithms and DRIr sensors confirm that the proposed distributed multi-target tracking method for a swarm of robots is effective and easy to implement.

Keywords robot swarms · local interactions · triangle lattice · target tracking · DRIr sensor

1 Introduction

The object tracking problem by a single robot with the required high-level capabilities has been studied over the past several decades toward real applications such as surveillance or reconnaissance. In recent years, increasing attention has been paid to swarms of simple robots with limited sensing and computational capabilities. It may offer many advantages over a single robot in terms of efficiency, fault-tolerance, adaptability, and so on [1][2]. Exploiting such features that swarms of robots can exhibit, new applications have emerged, such as order localization or plume tracing [3]-[6], and have expanded to support multiple target tracking [7]-[11].

Zarzhitsky *et al.* reported a chemical plume tracing method imitated from fluid physics [3] under the artificial physics framework [13]. Jatmiko *et al.* presented an algorithm for odor source localization in a changing environment based on the particle swarm optimization [4]. There also has been a wide range of discussion on the problem of sensor planning. Spletzer and Taylor [7] controlled the configuration of a team of mobile robots to optimally estimate the positions of a group of visible targets. Jung and Sukhatme [8] proposed a combination strategy of a local tracking controller and a high-level behavior-based framework, when a topological map of the environment is given, that distributed robots into regions according to target density. These works can be classified as cooperative target tracking. Other related works were mainly devoted to developing decentralized tracking strategies of robot swarms or mobile sensor networks for multiple stationary targets [9]-[11]. Krishnanand *et al.* addressed a problem for multiple odor source localization using mobile robot swarms [9]. In [10], a decentralized strategy for tracking multiple targets was introduced with a mobile sensor network based on the principle of triangulation.

This paper addresses the problem of coordinated multi-target tracking for a swarm of autonomous mobile robots, when the target is visible to only a limited number of robots. The main purpose is to develop a new distributed approach that enables large-scale robot swarms with limited sensing capabilities to track and/or capture multiple moving targets while achieving dynamic formations. This approach can be further applied to robotic sensor networks for tracking and/or capturing multiple toxic and hazardous substances or automated wide area surveillance. We em-

ploy the relative degree of attraction from individual targets based on the universal law of gravitation [26] and local interactions [23]. Specifically, the proposed coordinated tracking is achieved without using any leader, identifiers, common coordinate frame, memory of previous states, or explicit communication. The convergence properties of the proposed algorithms are proven using the theory of Lyapunov functions. Toward practical implementation of the algorithms, it is important to realize the observation capability of individual robots in an inexpensive and efficient way. For the purpose, the low-cost proximity sensor that we call "dual rotating infrared (DRIr) sensor" capable of 360 degree observation is developed and mounted on the front and rear edge of each robot. This allows mobile robots in various shapes 1) to obtain relative position measurements of neighboring robots, and 2) to follow a specific moving target, both in all directions. We explain how to realize each robot's observation function through the use of DRIr sensors. Extensive simulations and real robot experiments are performed to show the validity of the proposed method.

The rest of this paper is organized as follows. Section 2 gives a brief description on the state-of-the-art of local interactions. Section 3 presents the computational model and the physical robot configuration used in this paper. Section 4 illustrates how to realize each robot's observation function through the use of DRIr sensors. Section 5 describes the proposed tracking algorithm, its convergence properties, and simulation and experimental results. Section 6 draws our conclusions.

2 Background

In order to allow a swarm of robots to achieve the goal of a collaborative task, the motions of individual robots should be coordinated preferably in a decentralized way. Most of decentralized coordination is based on local rules of behavior observed from physical phenomena. Many approaches used such physical phenomena as van der Waals forces [12], gravitational forces [13], electric charges [14], spring forces [15]-[17], potential fields [18][19], line forces [20], equilibrium of molecules [21], and other virtual force [22]. Those works mostly use some sort of force balance between inter-individual interactions exerting an attractive or repulsive force on each other. This is mainly because the force-based interaction rules are considered simple but effective, and provide an intuitive understanding on local behavior.

Local formations achieved by the aforementioned local interactions may result in a mesh type. These formations offer multiple redundant connections ensuring maximum reliability and flexibility from the standpoint of topology. Depending on the degree of interaction among the robots, the network can be classified into fully or partially-connected topologies [24]. The fully-connected topologies have each robot interact with all of other robots simultaneously within a certain range. Thus, it poses too tight constraints on robot motion, and develops more computational complexity. Notably, it might lead to deadlocks where some of the robots have become trapped interstitially. These problems arise in most of the previous works [13][15][21][22].

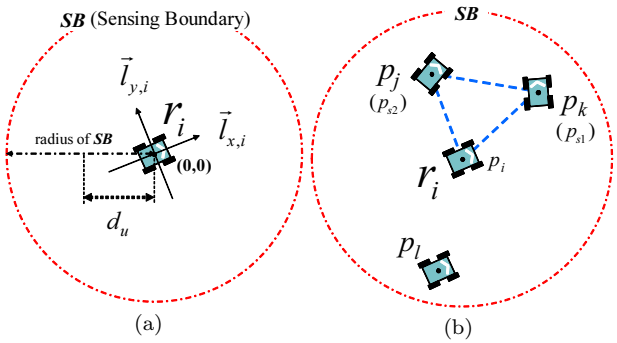


Fig. 1 Notations and definitions ((a) r_i's local coordinates and sensing boundary SB, (b) observation set $\mathbf{O}_i$, neighbor set $\mathbf{N}_i$, and triangular configuration $\mathbb{T}_i$)

On the contrary, using the partially-connected topology, robots interact selectively with other robots, but can be connected to all other robots by aggregating the local formations. In [16][17], partial graph pairs of robots were proposed that exert virtual forces to each other when their connection is part of the graph. For example, robots may choose to exert forces in a certain direction [16], where this selective interaction helps prevent them from being too tightly constrained. Due to similar reason, robots are enabled to achieve faster formation without deadlocks [17]. Our local interaction approach [23] based on such partially-connected topology is to construct uniformly spaced equilateral triangles that can reduce the number of interacting robots in a given location, yet establish an energy-efficient routing [25]. More importantly, compared with the computation of virtual spring forces to calculate an equilibrium of force balance, each robot utilizes only relative distance information of other robots.

3 Computational Model and Physical System Description

3.1 Computational Model and Definitions

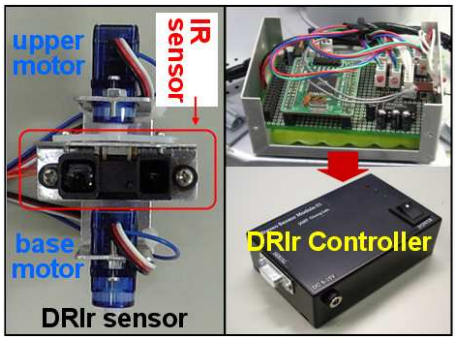
We consider a swarm of autonomous mobile robots denoted as $r_1, \cdots, r_n$. It is assumed that an initial distribution of all robots is arbitrary and their positions

are distinct. Each robot autonomously moves on a two-dimensional (2-D) plane. Robots have no leader and no identifiers. They do not share any common coordinate system, and do not retain any memory of past actions that gives inherently self-stabilizing property [27][28]. Due to a limited observation range, each robot can detect the positions of other robots only within its sight range or line-of-sight. In addition, robots do not communicate explicitly with other robots. Let the position $\mathbf{p}_i(t)$ of a robot r_i at time t be denoted as a state vector $\mathbf{p}_i(t) = [p_{i,x}\ p_{i,y}]^T$. We can define r_i's kinematics by $\dot{p}_{i,x} = u_i cos\theta_i, \quad \dot{p}_{i,y} = u_i sin\theta_i$, where u_i and θ_i are the translational and angular velocity of r_i, respectively. Based on the above model, all the robots execute an identical algorithm, and act independently and asynchronously of each other.

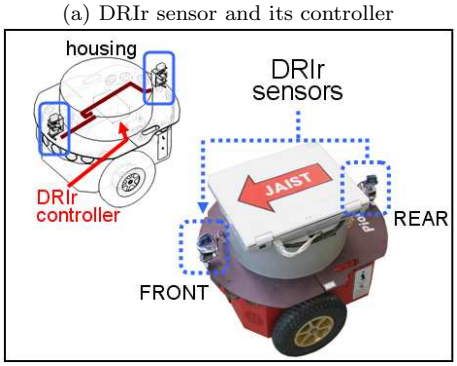
From now, we introduce the notation and definitions frequently used in this paper. Let us consider a robot r_i with local coordinates $\overrightarrow{l}_{x,i}$ and $\overrightarrow{l}_{y,i}$ seen in Fig. 1-(a). Here, $\overrightarrow{l}_{x,i}$ defines the vertical axis of r_i's coordinate system as its heading direction, and $\overrightarrow{l}_{y,i}$ denotes the horizontal axis by rotating the vertical axis 90 degrees counterclockwise. The center position of r_i is denoted as $\mathbf{p}_i(t))$ (p_i, for simplicity, hereafter). Accordingly, p_i is $(0, 0)$ with respect to r_i's local coordinates. The distance between the robot r_i's position p_i and the robot r_j's position p_j is denoted as $dist(p_i, p_j)$. We define a desired distance d_u between r_i and r_j. Next, r_i observes other robots located within its sensing boundary SB. As illustrated in Fig. 1-(b), it estimates the center positions of the observed robots, yielding a set of the positions $\mathbf{O}_i$ (=$\{p_j, p_k, p_k\}$) with respect to its local coordinates. Now, r_i can select two robots r_{s1} and r_{s2} within its SB that we call the neighbors of r_i and denote the set of their positions, $\{p_{s1}, p_{s2}\}$, as $\mathbf{N}_i$. Given p_i and $\mathbf{N}_i$, the *Triangular Configuration*, denoted by $\mathbb{T}_i$, is defined as a set of three distinct positions $\{p_i, p_{s1}, p_{s2}\}$, where the internal angle $\angle p_{s1} p_i p_{s2}$ of r_i is denoted by α_i. We define the *Equilateral Configuration*, denoted by $\mathbb{E}_i$, as a configuration that all the distance permutations of $\mathbb{T}_i$ are equal to d_u. We need a measure indicating to what degree $\mathbb{T}_i$ is configured into $\mathbb{E}_i$. Given $\mathbb{T}_i$, we can express the distance permutations with respect to r_i as the following matrix D_i

$$D_i = \begin{cases} \Big(dist(p_m, p_n) - d_u\Big)^2 & \text{if } m \neq n \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $\{\{p_m, p_n\} |\ p_m, p_n \in \mathbb{T}_i = \{p_i, p_{s1}, p_{s2}\}\}$. We will denote $(dist(p_m, p_n) - d_u)^2$ for simplicity as $(d_k - d_u)^2$. Using $\mathbb{T}_i$ and $\mathbb{E}_i$, we can formally define the *Local Interaction* as follows: Given $\mathbb{T}_i$, the local interaction allows p_i of r_i to maintain d_u with $\mathbf{N}_i$ at each time toward forming $\mathbb{E}_i$.



(a) DRIr sensor and its controller



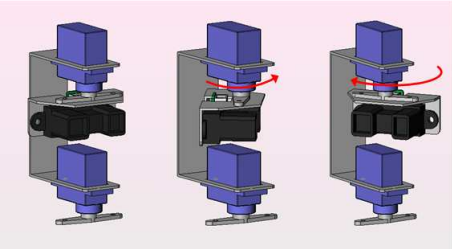
(b) sensor installation

Fig. 2 Mobile robot equipped with DRIr sensors

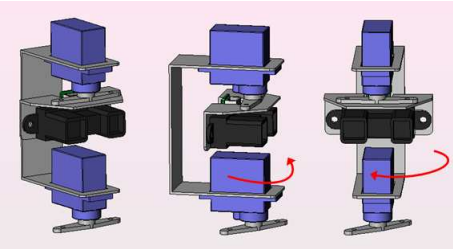
Several additional assumptions are made in constructing our robot model: 1) r_i is faster than the moving targets, 2) r_i is capable of estimating the distance and bearing of the target individually, if the target is within its SB, and 3) r_i is assigned to a single target at each time. Now we formally describe the *Target Tracking* problem based on the local interaction as follows: *Given robots $r_1, \cdots, r_n$ located at arbitrarily distinct positions and moving targets, how to enable robots to track the targets with their positions formed into $\mathbb{E}_i$.*

3.2 DRIr Sensor

Fig. 2 illustrates a pair of dual rotating infrared (DRIr) sensors and their controller, mounted on top of a Pioneer 3-DX mobile robot. A DRIr sensor has two MiniStudio MiniS RB90 servo motors and one Sharp GP2Y0A02YK infrared sensor. In detail, the Atmel ATmega128 microcontroller controls each servo motor rotating the infrared sensor and feeds the measured data to the main controller of the robot. The DRIr sensor controller forwards two-channel control signals to the front and rear DRIr sensors. One signal controls the rotation angle of each servo motor by pulse width modulation. The other signal is used for on-off control of the infrared sensor. Moreover, the analog output voltage representing the distance to the detecting surface is fed to the controller



(a) rotation of the upper motor



(b) rotation of the base motor

Fig. 3 Combined motion of sensor rotation

and converted to 10-bit digital values. From the relationship between the analog voltage level and the measured distance, the range from 12 *cm* to 180 *cm*, where the voltage level decreases with increasing distance in a unimodal fashion, can be used to estimate the distance. Each robot can be provided with a sensing range up to 400 *cm* including the size of the robot body.

The servo motors are independently controlled by the controller. One servo motor rotates up to 180 degrees, thus two identical motors can sweep a full 360 degrees. As illustrated in Fig. 3, the base motor enables the infrared sensor to be directed toward a specific direction, while the upper motor can rotate 180 degrees with respect to the direction of the base motor. By the combination of the base and upper motors, a wide variety of emitting directions of infrared rays can be effectively controlled. Specifically, the front DRIr sensor scans from -120 degrees to 120 degrees in azimuth with respect to the heading of the robot, where the base motor rotates 180 degrees and the upper motor adds another 60 degrees. The remaining 120 degree range cannot be observed since the line-of-sight path is blocked by the controller housing, but is covered by the rear DRIr sensors (that scan the same amount of range in the opposite direction). Therefore, a pair of DRIr sensors can cover a full 360 degrees.

3.3 Physical Robot Integration

Our customized mobile robot largely consists of three parts: a pair of DRIr sensors, MobileRobots Pioneer 3-DX platform, and the main controller. The DRIr sensors are mounted on the front and rear edge of the robot, which allows each robot to detect other robots in the front and rear direction simultaneously. The circular controller housing designed for the controller board represents the surface geometry whose center point is easy to detect irrespective of the robot's heading. Specifically, the center of the housing is coincident with the center of the mobile robot. A laptop PC is used as the main controller on top of the robot. It consists of the observation function and the tracking function. Fig. 4 shows the control architecture of the overall system. The inputs to the main controller include the measurement data obtained by the DRIr sensors and the predefined d_u between neighboring robots. At each time, our proposed tracking function enables r_i to compute its movement position at the next time step based on the observation function (see Section 4). A series of these iterative activations are controlled by the main controller.

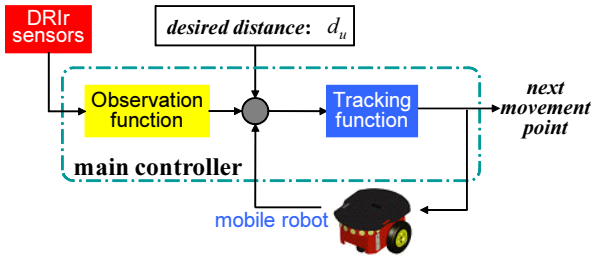


Fig. 4 System integration overview

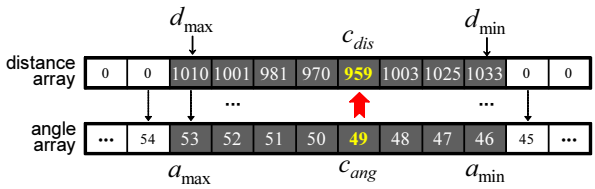
4 Observation Function

4.1 Observation Function Algorithm

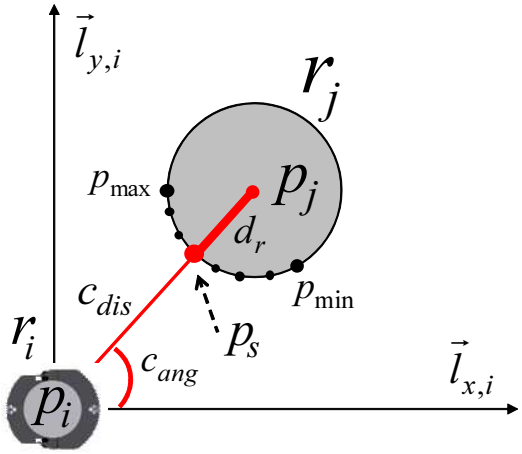
The observation function gives reliable estimates of the surface of neighboring robots, which can be obtained through the steps detailed below.

The measurement step constructs two one-dimensional arrays in the memory of each robot as illustrated in Fig. 5-(a). Here, the dimension of each array can be automatically adjusted according to the angular interval of the servo motor. When r_i scans the environment using its DRIr sensors at regular intervals, the distance to the surface of neighboring robots is recorded in the corresponding cell of the first array. At the same time, the servo motor angle is recorded in the second array so that the distance array corresponds to the motor angle array. Next, r_i checks their distance array cells that contain a non-zero value (from the lower bound d_{min} to the upper bound d_{max}) and reads the corresponding angle array cells.

The update step calibrates the measurement data with respect to a reference. For the purpose, a 100 $\times$

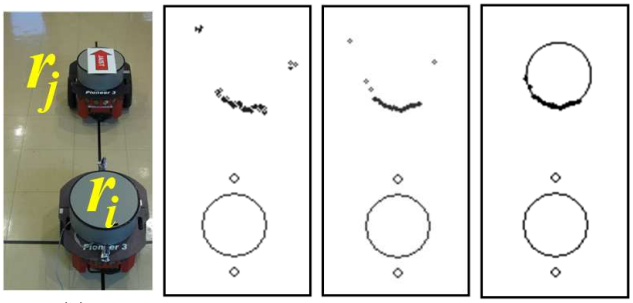


(a) distance and angle arrays of r_i



(b) r_i observes the neighbor robot r_j

Fig. 5 Observation scanning of the surface of a robot r_j



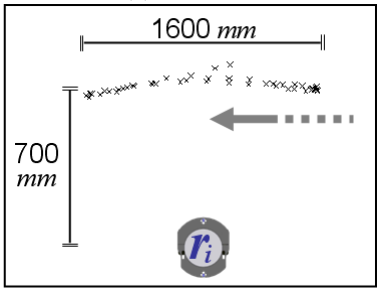
(a) (b) (c) (d)

Fig. 6 Observation process of the neighbor r_j by the robot r_i ((a) test scene, (b) measurement step, (c) update step, (d) recognition step)

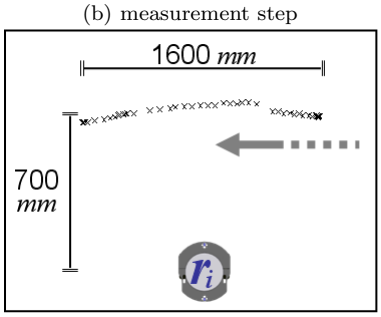
100 2-D grid with 4 cm × 4 cm unit cells is built. While recording data in the distance and angle arrays, the estimated distance is simultaneously stored in the corresponding cell of the grid as an integer intensity value. Once a full 360 degree scanning is completed, the Sobel edge detection algorithm [29] improves the original surface detection data.

The recognition step identifies the positions of robots. r_i collects the cells with the non-zero value from d_{min} to d_{max} in the updated distance array. Then, three feature points, p_{min}, p_{max}, and p_s, are specified using d_{min}, d_{max}, c_{dis}, and their corresponding cells in the angle array, respectively. By computing the average of a sequence of numeric values in the motor angle array, r_i selects the cell containing the value equal or closest to the average, and sets the center angle c_{ang} to this value. The distance cell corresponding c_{ang} is defined as the center distance c_{dis}. As shown in Fig. 5-(b), p_s is computed based on c_{dis} with the minimum distance value among the cells and c_{ang}. Next, r_i computes $dist(p_{min}, p_{max})$ and checks whether $dist(p_{min}, p_{max})$ is shorter than the controller housing diameter. If this distance exceeds the diameter, the collected cells can be considered as an arena border. Otherwise, these cells are considered as a robot. Through the above process, if robots are recognized, their center point p_j can be obtained by adding c_{dis} to the radius d_r of the controller housing (see Fig. 5-(b)). Consequently, the outputs of the observation function are O_i of neighboring robots.

(a) test scene



(b) measurement step

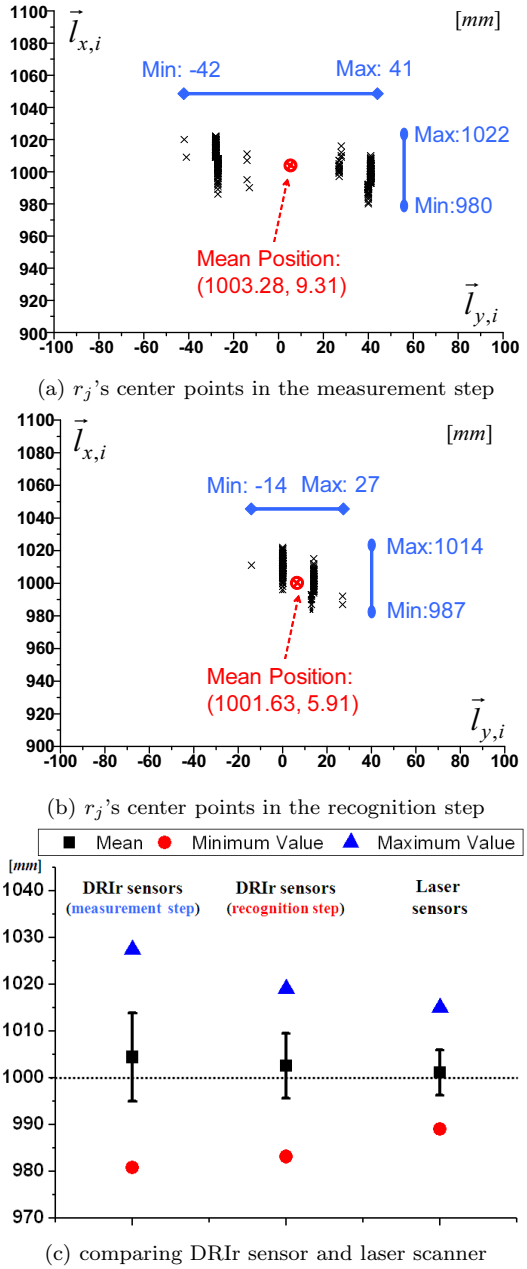


(c) recognition step

Fig. 7 Observation of the moving robot r_j by the robot r_i

4.2 Preliminary Tests

As shown in Figs. 6 through 8, we performed three kinds of tests evaluating the effectiveness of the observation function. In the first test seen in Fig. 6-(a), the robot r_i observes its neighbor r_j located 100 cm away. Figs. 6-(b), (c), and (d) show the data processing results obtained through the measurement, update, and recognition steps, respectively. Compared with Fig. 6-



(a) r_j's center points in the measurement step

(b) r_j's center points in the recognition step

(c) comparing DRIr sensor and laser scanner

Fig. 8 Estimation of r_j's center points in Fig. 6-(a)

(b), Fig. 6-(c) shows the enhanced surface detection by eliminating blurred and distorted edges. Fig. 7 presents the observation result for a moving robot r_j. Similarly, compared with Fig. 7-(b), Fig. 7-(c) shows that r_i could compute the center points of r_j moving with a reasonable velocity. Fig. 8 shows the results of 300 trials in the condition of Fig. 6-(a). Figs. 8-(a) and (b) show the estimation results of r_j's center points through each step of the process. Fig. 8-(c) shows the results of statistical analysis of r_j's distances estimated by the DRIr sensor and Hokuyo's URG laser scanner. Here, the square, triangle, and circle indicate the mean value, maximum value and minimum value, respectively. The error bars represent the 95% confidence intervals. The laser scanner outperforms the DRIr sensor in terms of accuracy, but the DRIr sensor also shows reasonably good accuracy. Note that the observation function algorithm requires robots to be initially positioned a minimum distance of 100 *mm* apart from DRIr sensor, with a clear line of sight. From the results that we have seen so far, the DRIr sensor scanning observation capability can be considered quite satisfactory for practical use.

ALGORITHM-1 LOCAL INTERACTION (code executed by r_i)

FUNCTION $\varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$

1 $p_{ct,x} := (p_{s1,x} + p_{s2,x} + p_{i,x})/3$
2 $p_{ct,y} := (p_{s1,y} + p_{s2,y} + p_{i,y})/3$
3 $p_{ct} := (p_{ct,x}, p_{ct,y})$ // centroid
4 $\phi :=$ angle between $\overline{p_{s1}p_{s2}}$ and $\overrightarrow{l}_{y,i}$
5 $p_{ti,x} := p_{ct,x} + d_u \cos(\phi + \pi/2)/\sqrt{3}$
6 $p_{ti,y} := p_{ct,y} + d_u \sin(\phi + \pi/2)/\sqrt{3}$
7 $p_{ti} := (p_{ti,x}, p_{ti,y})$ // output

5 Tracking Function

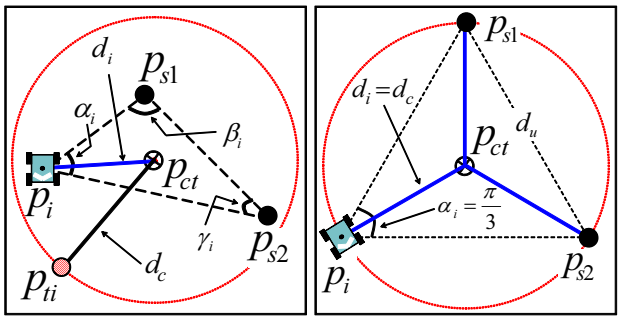
5.1 Local Interaction Algorithm

Here we explain the local interaction algorithm, termed ALGORITHM-1, that enables three neighboring robots to generate $\mathbb{E}_i$ of side length d_u from an arbitrary $\mathbb{T}_i$.

ALGORITHM-1 includes the function $\varphi_{interaction}$ whose arguments are p_i and $\mathbf{N}_i$ at each time. Consider r_i and its two neighbors r_{s1} and r_{s2} located within its SB. As shown in Fig. 9-(a), three robots are configured into $\mathbb{T}_i$ whose vertices are p_i, p_{s1}, and p_{s2}, respectively. First, r_i calculates the centroid of the triangle $\triangle p_i p_{s1} p_{s2}$, denoted by p_{ct}, with respect to its local coordinates, and measures the angle ϕ between the line connecting the two neighbors $\overline{p_{s1}p_{s2}}$ and $\overrightarrow{l}_{y,i}$. Using p_{ct} and ϕ, r_i calculates the next movement point p_{ti} by its current observation of neighboring robots. Intuitively, under ALGORITHM-1, r_i may maintain d_u with its two neighbors at each time. In other words, each robot attempts to form an isosceles triangle with $\mathbf{N}_i$ at each time, and by repeatedly doing this, three robots configure themselves into $\mathbb{E}_i$.

As illustrated in Fig. 9-(b), we consider the circumscribed circle of an equilateral triangle whose centroid is p_{ct} of $\triangle p_i p_{s1} p_{s2}$ and radius d_c is $d_u/\sqrt{3}$. The local interaction determines the position of each robot by controlling the distance d_i from p_{ct} and α_i (see Fig. 9-(a)). First, d_i is controlled by the following equation:

$$\dot{d}_i(t) = -a(d_i(t) - d_c), \tag{2}$$



(a) two control parameters: range d_i and bearing α_i

(b) desired equilateral triangular configuration

Fig. 9 Illustration of ALGORITHM-1

where a is a positive constant. Indeed, the solution of (2) is $d_i(t) = |d_i(0)|e^{-at} + d_c$ that converges exponentially to d_c as t approaches infinity. Secondly, α_i is controlled by the following equation:

$$\dot{\alpha}_i(t) = k(\pi/3 - \alpha_i(t)), \tag{3}$$

where k is a positive number. Since the total internal angle of a triangle is 180 degrees, $(\pi/3 - \alpha_i(t))$ can be derived from $\frac{1}{3}(\beta_i(t) + \gamma_i(t) - 2\alpha_i(t))$ where β_i and γ_i indicate $\angle p_i p_{s1} p_{s2}$ and $\angle p_i p_{s2} p_{s1}$, respectively (see Fig. 9-(a)). Likewise, the solution of (3) is $\alpha_i(t) = |\alpha_i(0)|e^{-kt} + \frac{\pi}{3}$ that converges exponentially to 60 degrees as t approaches infinity.

Note that (2) and (3) imply that the trajectory of r_i converges to d_c and 60 degrees, an equilibrium state $\mathbf{x}_d$ shown as $[d_i(t)\ \alpha_i(t)]^T$ in Fig. 9-(b). This also implies that three neighboring robots eventually form $\mathbb{E}_i$. In order to prove the convergence of the local interactions, we demonstrate the application of Lyapunov stability theory [30]. Now, the desired $\mathbf{x}_d$ can be regarded as one that minimizes the energy level of a Lyapunov function.

Note that (2) and (3) imply that three robots eventually form an equilateral triangle of side length d_u. In order to show the convergence, consider the following scalar function

$$f_{l,i} = \frac{1}{2}(d_i - d_c)^2 + \frac{1}{2}(\frac{\pi}{3} - \alpha_i)^2 \tag{4}$$

that is always positive definite except $d_i \neq d_c$ and $\alpha_i \neq \frac{\pi}{3}$. The derivative of the scalar function is given by $\dot{f}_{l,i} = (d_i - d_c)\dot{d}_i - (\frac{\pi}{3} - \alpha_i)\dot{\alpha}_i$. Using (2) and (3), the derivative is rewritten by

$$\dot{f}_{l,i} = -a(d_i - d_c)^2 - k(\frac{\pi}{3} - \alpha_i)^2. \tag{5}$$

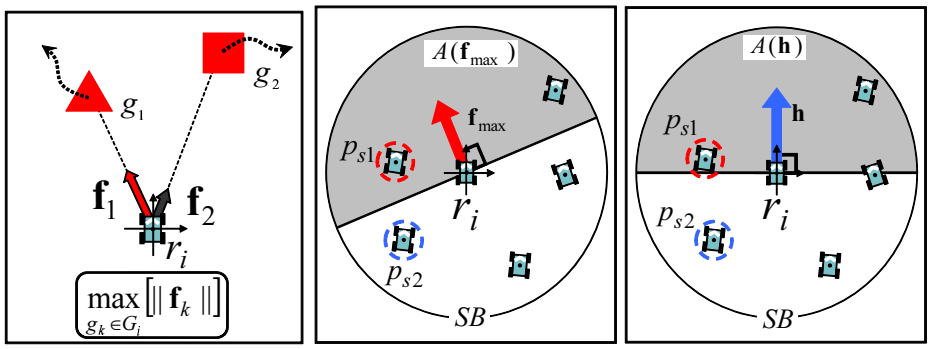
Since a and k are positive, (5) is negative definite. The scalar function $f_{l,i}$ is radially unbounded since it tends to infinity as $\| \mathbf{x}_d \| \rightarrow \infty$. Therefore, $\mathbf{x}_d$ is asymptotically stable, implying that r_i reaches a vertex of $\mathbb{E}_i$. Regarding the scalability of the algorithm, see [23].

5.2 Target Tracking Algorithm

The target tracking algorithm, ALGORITHM-2, gives a solution to how to find r_i's moving direction toward a desired target, and simultaneously how to configure the positions of the neighboring robots into $\mathbb{E}_i$ according to the desired target direction.

Under ALGORITHM-2, it is assumed that r_i is faster than the moving targets. When detecting multiple targets, defined $\{g_k | 1 \leq k \leq m\}$, in Fig. 10-(a), r_i selects its direction toward a target g_k. Imitated from the law of gravitation, r_i determines its direction by using the relative degree of attraction from the targets, termed the favorite vector $\mathbf{f}_k$, whose magnitude is given by $\|\mathbf{f}_k\| = \|1/d_k^2\|$ where d_k denotes the distance between g_k and r_i. Thus, the set of detected targets G_i is represented by the set of favorite vectors $\{\mathbf{f}_k | 1 \leq k \leq m\}$. Then r_i selects the maximum magnitude of $\mathbf{f}_k$, denoted by $\|\mathbf{f}_k\|_{\max}$. From $\mathbf{f}_k$, r_i is assigned to a single target within its SB at each time. As shown in Fig. 10-(b), r_i defines a maximum favorite target area $A(\mathbf{f}_{\max})$ within its SB intersected with the upper half plane along the direction of $\|\mathbf{f}_k\|_{\max}$. r_i checks whether there exist any neighbors in $A(\mathbf{f}_{\max})$. If neighbors are found, r_i selects the first neighbor r_{s1} located the shortest distance away from p_i to define p_{s1}. Otherwise, r_i spots a virtual point p_v located at some distance d_v away from p_i along $\|\mathbf{f}_k\|_{\max}$ to define p_{s1}. When no target is observed, r_i finds the first neighbor within its SB. As illustrated in Fig. 10-(c), r_i defines its heading $\mathbf{h}$ with respect to its local coordinates. Let $A(\mathbf{h})$ denote the area of heading direction within SB intersected with the upper half-plane along $\mathbf{h}$. r_i checks whether there exist any neighbors in $A(\mathbf{h})$. If neighbors exist within $A(\mathbf{h})$, r_i selects r_{s1} with the shortest distance away from p_i. Otherwise, r_i finds r_{s1} within SB by the same method above. The second neighbor r_{s2} is selected such that the total distance from p_{s1} to p_i passing through p_{s2} is minimized. As a result, using p_i and $\mathbf{N}_i$, p_{ti} can be obtained by $\varphi_{interaction}$ in ALGORITHM-1.

Note that the robots are always required to maintain their formation of equilateral triangles, denoted by $\sum_{i=1}^{n} \mathbb{E}_i$. When the targets are detected, r_i is required to determine its direction toward $\|\mathbf{f}_k\|_{\max}$. $\|\mathbf{f}_k\|_{\max}$ forces r_i to move to a certain direction while being configured into $\mathbb{E}_i$. Therefore, $\|\mathbf{f}_k\|_{\max}$ can be used for controlling individual robot motions given by $\|\mathbf{f}_k\|_{\max} = \ddot{f}_{l,i}$. If r_i follows $\|\mathbf{f}_k\|_{\max}$ through local interactions, we can prove its convergence into $\mathbb{E}_i$ while tracking g_k as detailed below.



(a) computation of favorite vectors

(b) computation of neighbor selection within $A(\mathbf{f}_{\max})$

(c) computation of neighbor selection within $A(\mathbf{h})$

Fig. 10 Illustration of ALGORITHM-2

ALGORITHM-2 TARGET TRACKING (code executed by r_i)

FUNCTION $\varphi_{tracking}(O_i, G_i, p_i)$
1 **If** $\{\exists g_k \in G_i\}$ **Then**
2 $\mathbf{f}_{\max} := \max_{g_k \in G_i} [\|\mathbf{f}_k\|]$
3 $A(\mathbf{f}_{\max})$:= favorite target area
4 A_p := {set of positions of robots located in $A(\mathbf{f}_{\max})$}
5 **If** $\{\exists p \in A_p\}$ **Then**
6 $p_{s1} := \min_{p \in A_p - \{p_i\}} [dist(p_i, p)]$
7 **Else**
8 $p_{s1} := p_v$
9 **End If**
10 **Else**
11 $\mathbf{h}$:= heading direction
12 $A(\mathbf{h})$:= heading direction area
13 A_p := {set of robot positions located in $A(\mathbf{h})$}
14 **If** $\{\exists p \in A_p\}$ **Then**
15 $p_{s1} := \min_{p \in A_p - \{p_i\}} [dist(p_i, p)]$
16 **Else**
17 $p_{s1} := \min_{p \in O_i - \{p_i\}} [dist(p_i, p)]$
18 **End If**
19 **End If**
20 $p_{s2} := \min_{p \in O_i - \{p_i, p_{s1}\}} [dist(p_{s1}, p) + dist(p, p_i)]$
21 $\varphi_{interaction}(\{p_{s1}, p_{s2}\}, p_i)$

Here, Lyapunov's theory is applied to show the convergence of r_i using the positive definite scalar function $f_{t,i}$ given by

$$f_{t,i} = \frac{1}{2}(\dot{f}_{l,i})^2 + f_{l,i} + \sum_{\mathbb{T}_i}(d_k - d_u)^2, \tag{6}$$

where $f_{l,i}$ indicates the scalar function of local interactions in (4), and $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ is defined as the constant value D_i associated with $\mathbb{T}_i$ at each time (see (1)). A symmetric matrix D_i can be said to be positive definite, if $\mathbf{x}^T D_i \mathbf{x} > 0$ for every nonzero $\mathbf{x}$ [31]. Moreover $f_{l,i}$ is always positive definite except $d_i \neq d_c$ and $\alpha_i \neq \frac{\pi}{3}$. (If $\mathbb{T}_i$ is equal to $\mathbb{E}_i$, it is easily seen that $\sum_{\mathbb{T}_i}(d_k - d_u)^2$ reaches 0, resulted from $d_c = d_u/\sqrt{3}$.) Next, The derivative of $f_{t,i}$ is given by

$$\dot{f}_{t,i} = \dot{f}_{l,i}(\ddot{f}_{l,i}) + \dot{f}_{l,i} \;=\; \dot{f}_{l,i}(\ddot{f}_{l,i} + 1). \tag{7}$$

It is evident that $\dot{f}_{t,i}$ is negative definite. Therefore, based on Lyapunov's theory, the motion of r_i can eventually converge into $\mathbb{E}_i$ while following $\|\mathbf{f}_k\|_{\max}$.

Next, the collective scalar function $\mathtt{F}_t$ of a swarm of robots is a nonzero function with the property that any solution of the set of algebraic constraints on range and bearing (see Fig. 9-(b)) is closely related to a set of equilibria for $\{r_i | 1 \leq i \leq n\}$ and vice versa. Without loss of generality, $\mathtt{F}_t$ is a diminishing energy function with a scalar potential. Therefore, $\mathtt{F}_t$ for a swarm of n robots is defined as $\mathtt{F}_t = \sum_{i=1}^{n} f_{t,i}$. It is straightforward to verify that $\mathtt{F}_t$ is positive definite and $\dot{\mathtt{F}}_t$ is negative definite. Consequently, a swarm of n robots converges into $\mathbb{E}_i$ with their N_i while tracking g_k by $\|\mathbf{f}_k\|_{\max}$ within their SB.

5.3 Simulation and Experimental Results

The effectiveness of our tracking function is verified through extensive simulations and experiments. Simulations are conducted to investigate the ability of the algorithm to scale to large number of robots and multiple targets. The targets are assumed to be either a light or odor source whose position can be measured with high precision within SB. Moreover, robots do not know the time-varying velocities and moving directions of targets in advance. In simulation tests, we set the distance d_v between p_v and p_i to 1.2 times longer than d_u and the range of SB to 3.5 times longer than d_u. In the real robot experiments, each robot is equipped with a pair of DRIr sensors. They move with a linear velocity of 60 *mm/s* and an angular velocity of 100 *deg/s*. d_u is set to 80 *cm* and DRIr sensors measure relative distance up to 200 *cm* by emitting an infrared ray every one degree

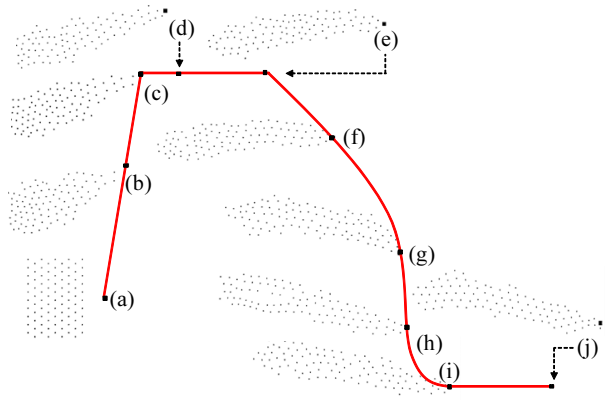


Fig. 11 Simulation of tracking a moving target

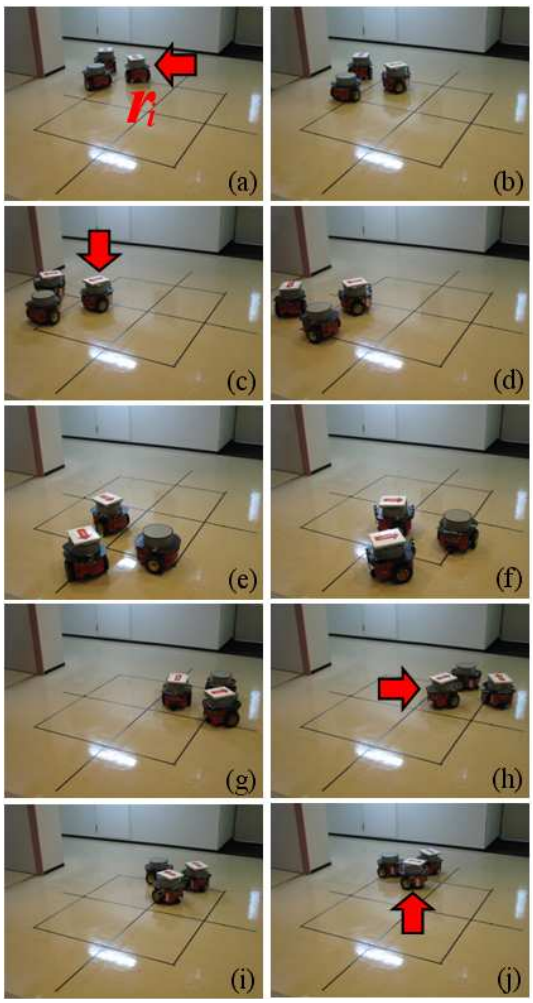
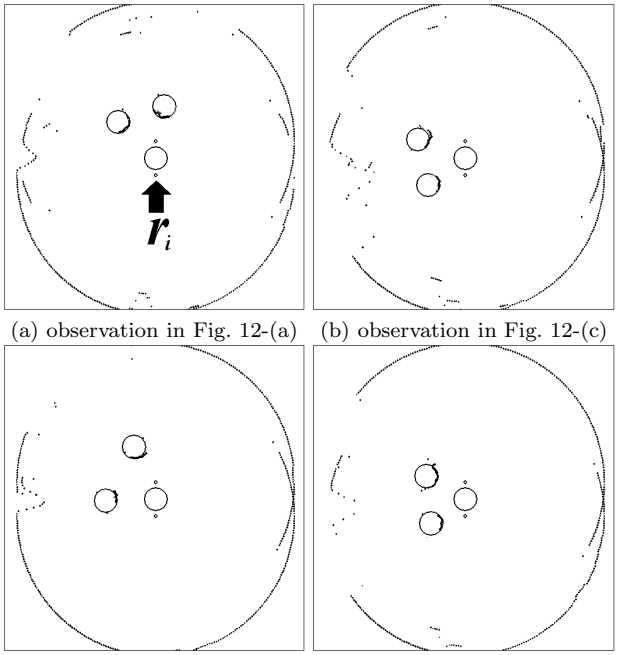


Fig. 12 Tracking a target moving along a square trajectory

while rotating with 308 *deg/s*. One robot, teleoperated by a human operator, is assumed to be a moving target.

Fig. 11 shows that a swarm of 100 robots tracks a moving target indicated by the bold red line. As the target moves from its initial location, robots start to track the target, where traffic congestion happens



(a) observation in Fig. 12-(a) (b) observation in Fig. 12-(c)

(c) observation in Fig. 12-(h) (d) observation in Fig. 12-(j)

Fig. 13 Observation result of r_i in Fig. 12

seen in Figs. 11-(b) through (d). This is because the inter-robot distance does not converge uniformly in a short period, as robots with limited visibility, unless the target is detected, interact with their neighbors to form $\mathbb{E}_i$. Similar phenomena are observed in Fig. 17. This congestion is relieved in Fig. 11-(e) as the target moves ahead to a certain extent. Even though an uneven spatio-temporal congestion may exist, each robot can track the target successfully while forming $\mathbb{E}_i$. As shown in Fig. 11-(j), once the target stops, robots starts to self-configure themselves into $\sum_{i=1}^{n} \mathbb{E}_i$.

Figs. 12 through 14 show how robots form an equilateral triangle while tracking a target. Fig. 12 shows that two robots track a target that moves along a square trajectory. Fig. 13 displays the results of observation of other robots performed by the robot r_i, tagged with the red arrow in Fig. 12. Starting from the initial random distribution in Fig. 12-(a), two robots successfully tracked the target with which they form an equilateral triangle. We test the next case, where the target moves straight away as shown in Fig. 14. The left hand side figures present the snapshots of the target tracking. Correspondingly, the outputs of r_i's observation function is shown on the right hand side figures. As r_i observed the target in Figs. 14-(g) and (h), under ALGORITHM-2, r_i interacted with its two neighbors and formed themselves into $\mathbb{E}_i$. If there exist a larger number of robots around r_i, it may select a specific subset of robots as its neighbors considering the direction of the nearest target. Moreover, the area border (wall) was also detected

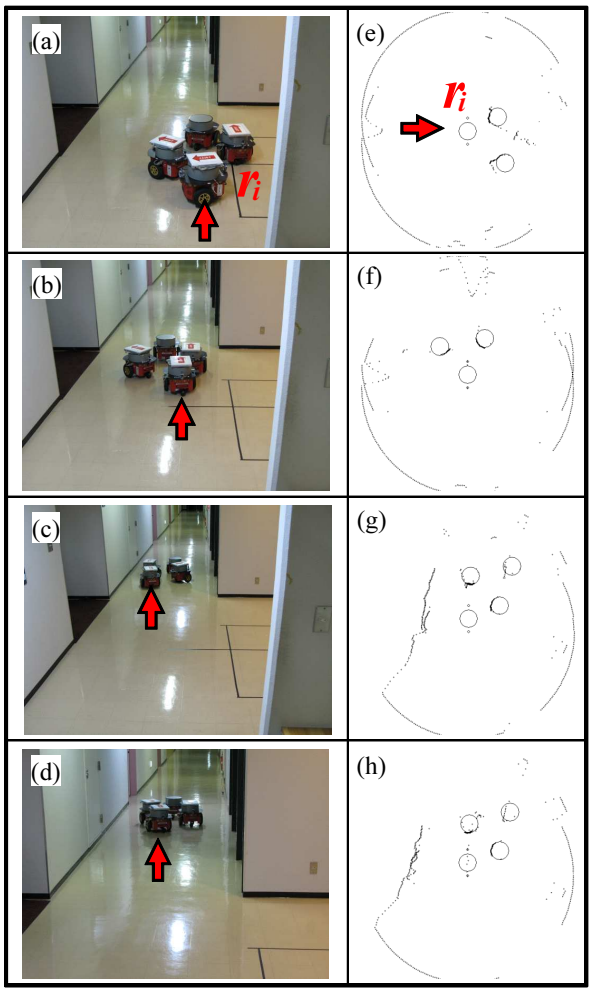


Fig. 14 Experiment of tracking by three real robots

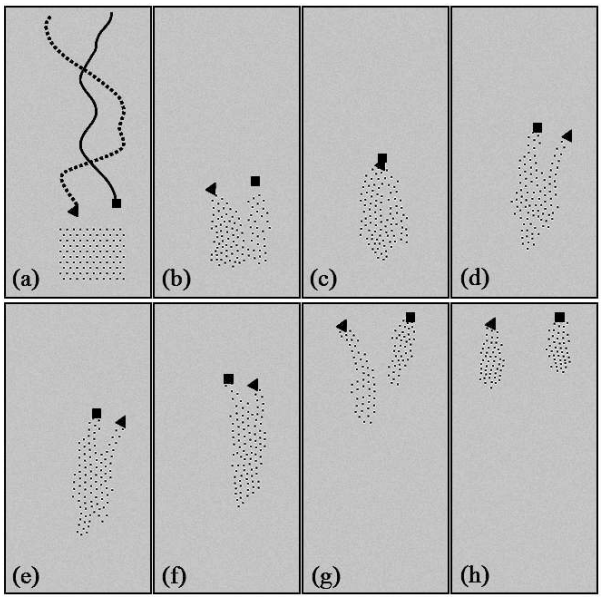


Fig. 15 Simulation results of tracking two slow moving targets

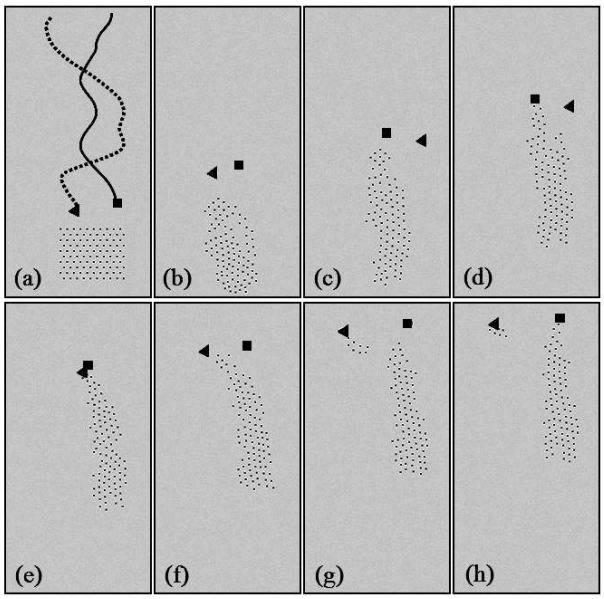


Fig. 16 Simulation results of tracking two fast moving targets

by the DRIr sensors. From these results, we have verified that robots equipped with DRIr sensors tracked a moving target satisfactorily in an equilateral triangle formation under our laboratory conditions.

Next, we performed simulations to investigate how a swarm of 100 robots tracks multiple moving targets. Figs. 15 and 16 show the difference in tracking capability for slow and fast moving targets, respectively. In both cases, robots can locate and keep track of a target nearest to them. As they get closer to other targets, they switch to the nearest target while forming $\mathbb{E}_i$. Fig. 17 shows the results of tracking three targets. In Figs. 17-(a) through 17-(c), robots track the targets moving at the same velocity in the same direction. As the targets change their velocity and direction, robots spilt themselves into three groups in Figs. 17-(d) through (h). It can be observed that robots always keep track of the nearest target regardless of the number of targets. In most cases, a considerable majority of robots may not directly detect any targets, but they can track a target through local interactions with their neighbors. A small number of robots that stay close to targets can cause the remaining robots to move toward their individual target.

5.4 Discussion

Our tracking approach offers several notable advantages. First, an equilateral triangle lattice network can be built with a partially connected mesh topology. Among all the possible types of regular polygons, the equilateral triangle lattice can minimize the computational

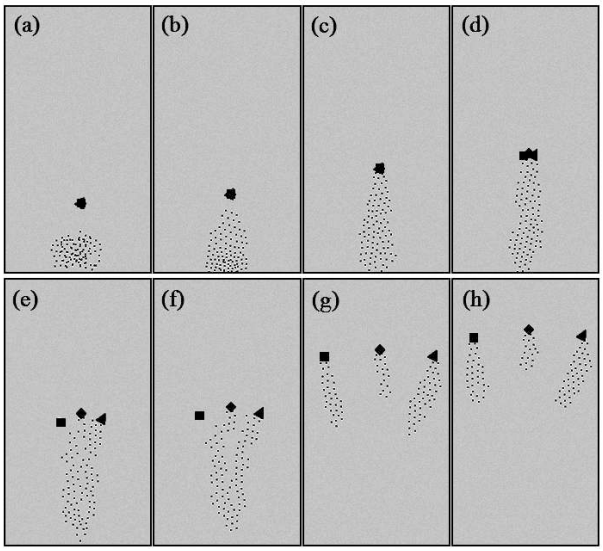


Fig. 17 Simulation results of tracking three moving targets

burden, because only a limited number of robots interact with each other. It is also highly scalable. Secondly, each robot utilizes only position information of other robots. On the contrary, many related works require the computation of relative velocities or accelerations to calculate attractive or repulsive forces. Thirdly, robots compute the target position without requiring memories of past actions or states. This oblivious algorithm can effectively cope with transient errors. It is known that nonoblivious algorithms might not work in situations where robots are activated asynchronously, or the number of robots increases or decreases. Most importantly, we attempt to solve the tracking problem by eliminating such major assumptions as robot identifiers, common coordinates, and explicit communication, often made in other works.

For real world applications, there still remain several issues. In order to distinguish between other robots and various objects, it could be advantageous to fuse DRIr and RFID sensor data for obstacle-cluttered environments. It is also required to develop adequate sensors and recognition algorithms suited for a specific type of target. As an alternative way of interaction, some system for direct communication may be used. In this case, robots need to have such information as individual identification numbers or global coordinates. Likewise, it may face many difficulties including limited bandwidth, range, and interference. The required robot capabilities and resources, and possible interaction models are left for future work.

6 Conclusion

This paper presented a real-time tracking approach, enabling a swarm of mobile robots to follow multiple moving targets while forming regular triangle meshes through local interactions. The proposed algorithms are distributed and deadlock free, without requiring any leader, identifiers, common coordinate system, memory of previous states, or explicit means of communication. We also addressed practical design and hardware implementation of the proximity DRIr sensors that provide robots with full 360 degree azimuth scanning capability. By employing a pair of DRIr sensors, each robot could obtain relative positioning information as well as the surface geometry of neighboring robots. The major contribution of this work can be summarized as follows: 1) The convergence properties of the proposed tracking algorithm were proven mathematically, and verified through extensive simulations and real robot experiments. 2) A new proximity sensor was developed. Its features include low cost, high reliability, and easy integratability into commercial mobile robots. 3) The proposed algorithm and devices can be effectively applied to mobile robotic sensor networks for surveillance missions or capturing toxic and hazardous substances.

References

1. H. Choset, "Coverage for robotics-a survey of recent results," Annals of Mathematics and Artificial Intelligence, vol.31, no.1-4, pp.113-126 (2001)
2. E. Sahin, "Swarm robotics: from sources of inspiration to domains of application," Proc. 8th International Conference on the Simulation of Adaptive Behavior, *LNCS*, vol.3342, pp.10-20 (2005)
3. D. Zarzhitsky, D. Spears, and W. Spears, "Distributed robotics approach to chemical plume tracing," Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems, pp.4034-4039 (2005)
4. W. Jatmiko, K. Sekiyama, and T. Fukuda, "A particle swarm-based mobile sensor network for odor source localization in a dynamic environment," M. Gini and R. Voyles (eds.), *Distributed autonomous robotic systems 7*, Springer Japan, pp.71-80 (2007)
5. A. T. Hayes, A. Martinoli, and R. M. Goodman, "Swarm robotic odor localization: off-line optimazation and validation with real robots," Robotica, vol.21, no.4, pp.427-441 (2003)
6. A. Dhariwal, G. S. Sukhatme, and A. A. G. Requicha, "Bacterium-inspired robots for environmental monitoring," Proc. IEEE Int. Conf. Robotics and Automation, pp.1436-1443 (2004)
7. J. R. Spletzer and C. J. Taylor, "Dynamic sensor planning and control for optimally tracking targets," International Journal of Robotics Research, vol.22, no.1, pp.7-20 (2003)
8. B. Jung and G. S. Sukhatme, "Tracking targets using multiple robots: the effect of environment occlusion," Autonomous Robots, vol.13, no.3, pp.191-205 (2002)
9. K. N. Krishnanand, P. Amruth, and M. H. Guruprasad, "Glowworm-inspired robot swarm for simultaneous taxis towards multiple radiation sources," Proc. IEEE Int. Conf. Robotics and Automation, pp.958- 963 (2006)

10. S. Kamath, E. Meisner, and V. Isler, "Triangulation based multi target tracking with mobile sensor networks," Proc. IEEE Int. Conf. Robotics and Automation, pp.3283-3288 (2007)
11. X. Cui, C. T. Hardin, R. K. Ragade, and A. S. Elmaghraby, "A swarm approach for emission sources localization," Proc. 16th IEEE Int. Conf. Tools with Artificial Intelligence, pp.424-430 (2004)
12. Y. F. Zheng and W. Chen, "Mobile robot team forming for crystallization of protein," Autonomous Robots, vol.23, no.1, pp.69-78 (2007)
13. W. Spears, D. Spears, J. Hamann, and R. Heil, "Distributed, physics-based control of swarms of vehicles," Autonomous Robots, vol.17, no.2-3, pp.137-162 (2004)
14. A. Howard, M. J. Mataric, and G. S. Sukhatme, "Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem," Proc. Int. Sym. Distributed Autonomous Robotic Systems, pp.299-308 (2002)
15. K. Fujibayashi, S. Murata, K. Sugawara, and M. Yamamura, "Self-organizing formation algorithm for active elements," Proc. IEEE Sym. Reliable Distributed Systems, pp.416-421 (2002)
16. J. McLurkin and J. Smith, "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots," Proc. Int. Sym. Distributed Autonomous Robotic Systems, pp.831-890 (2004)
17. B. Shucker, T. D. Murphey, and J. K. Bennett, "Convergence-preserving switching for topology-dependent decentralized systems," IEEE Transactions on Robotics, vol.24, no.6, pp.1405-1415 (2008)
18. T. Balch and M. Hybinette, "Social potentials for scalable multi-robot formations," Proc. IEEE Int. Conf. Robotics and Automation, pp.73-80 (2000)
19. J. Reif and H. Wang, "Social potential fields: a distributed behavioral control for autonomous robots," Robotics and Autonomous Systems, vol.27, no.3, pp.171-194 (1999)
20. E. Martison, D. Payton, "Lattice formation in mobile autonomous sensor arrays," Proc. 8th International Conference on the Simulation of Adaptive Behavior, LNCS, vol.3342, pp.98-111 (2005)
21. N. Heo and P. K. Varshney, "A distributed self spreading algorithm for mobile wireless sensor networks," Proc. IEEE Wireless Communication and Networking Conference, pp.1597-1602 (2003)
22. G.-L. Wang, G. Cao, and T. L. Porta, "Movement-assisted sensor deployment," Proc. IEEE Infocom Conference, pp.2469-2479 (2004)
23. G. Lee and N. Y. Chong, "A geometric approach to deploying robot swarms," Annals of Mathematics and Artificial Intelligence, vol.52, no.2-4, pp.257-280 (2009)
24. S. Ghosh, K. Basu, and S. K. Das, "An architecture for next-generation radio access networks," IEEE Network, vol.19, no.5, pp.35-42 (2005)
25. P. C. Gurumohan and J. Hui, "Topology design for free space optical networks," Proc. 12th International Conference on Computer Communications and Networks, pp.576-579 (2003)
26. D. Halliday, R. Resnick, and J. Walker. *Fundamentals of physics*, Wiley, 5th Ed. (1997)
27. I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: formation of geometric patterns," SIAM Journal of Computing, vol.28, no.4, pp.1347-1363 (1999)
28. S. Dolev, *Self-Stabilization*, MIT Press (2000)
29. R. C. Gonzalez and R. E. Woods, *Digital image processing*, 2nd ed. Prentice Hall (2002)
30. J. E. Slotine and W. Li, *Applied nonlinear control*. Prentice-Hall (1991)
31. C.-T. Chen, *Linear system theory and design*. 3rd ed. Oxford University Press (1999)