Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Optimal Flocking Control for a Mobile Sensor Network Based a Moving Target Tracking

Hung Manh La
School of Electrical and Computer Engineering
Oklahoma State University
Stillwater, OK, 74078, USA
hung.la@okstate.edu

Trung Hien Nguyen* (nguyenhientrung@tnut.edu.vn),
Cong Huu Nguyen** Hien Nhu Nguyen*
Electrical Engineering Department*
Electronics Engineering Department**
Thai Nguyen University of Technology, Thai Nguyen , Vietnam

***Abstract*—Target tracking is an important task in sensor networks, especially in mobile sensor networks. Flocking control is used to control a mobile sensor network to track a target. However, there are some existing problems in this control method, such as the problem of how to design an optimal flocking control algorithm with optimal flocking parameters to allow the network to catch up the target as fast as possible in order to minimize the tracking time and power consumption. This paper presents an optimization problem in flocking control for a mobile sensor network to track a moving target. A non convex optimization method based on genetic algorithms is developed. The overall purpose of this approach is to find out the optimal solutions of flocking parameters that deliver desired swarm behaviors to minimize the cost function. This cost function represents the time it takes all mobile sensors (robots) in the network to catch up the moving target. The experimental tests are obtained to demonstrate our approach.**

Keywords: Flocking control, Target tracking, Mobile sensor networks, Evolutionary algorithms.

I. INTRODUCTION

A. Motivation

Sensor networks [1], especially mobile sensor networks [2] have been extensively studied 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. Therefore mobile sensor networks can be applied in many applications such as target tracking [3] in underwater submarine detection and protection of endangered species [4]. Here we consider a mobile sensor in a mobile sensor network as a mobile robot which is equipped motors or actuators made by artificial muscle such as Ionic Polymer Metal Composite (IPMC) [5] for mobility, and camera, sonar or laser for target sensing and navigation.

A main issue for multiple mobile sensors to track a moving target is that these sensors have to move together without collision among them during tracking which requires the use of cooperative control methods [6], [7], [8], [9]. One of these methods is flocking control [10]. We know that flocking is a phenomenon in which a number of mobile agents move together and interact with each other while ensuring no collision, velocity matching, and flock centering [11]. In nature, schools of fish, birds, ants, and bees, etc. demonstrate the phenomena of flocking. The problem of flocking has been studied for many years. It has greatly attracted many researchers in physics [12], [13], mathematics [14], biology [15], and especially in control science in recent years [10], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25].

The optimization problem of dynamic target tracking in a mobile sensor network is that how fast each member in the network can catch up the target while maintaining the network formation. An important issue of the ability to perform fast computational of flocking dynamics is to optimize the interaction of the flock members so that they can obtain the desired task. In order to achieve such purpose, this paper develops a genetic algorithm to tackle the objective function which is non convex and non-differentiable where the classical gradient-based optimization techniques are not robust and usually trapped by the local minima.

B. Related Work

In this section, we review existing works in flocking control which are related to our approach.

Flocking control has been studied by many researchers. Wang and Gu [19] presented a survey of recent research achievements of robot flocking. Their paper gave an overview of the related basic knowledge of graph theory, potential function, network communication and system stability analysis. In [10], a theoretical framework for design and analysis of distributed flocking algorithms was proposed. These algorithms solved the flocking control in the absence and presence of obstacles. The static and dynamic virtual leaders were used as a navigational feedback for all mobile agents. An extension of the flocking control algorithms in [10], flocking of agents with a virtual leader in the case of a minority of informed agents and in the case of varying velocity of virtual leader, was presented in [16] and [17]. La and Sheng [26], [27], [28], [29] proposed the distributed flocking control with Single-CoM (Center of Mass) and Multi-CoM to improve the tracking performance in the obstacle space, and to deal with the complex or changing environment they proposed the distributed adaptive flocking control algorithm that can allow the network maintains the connectivity, tracking performance and similar formation in both free and obstacle spaces. In addition, to solve the problem of slitting/merging mobile agents in the case of multiple dynamic targets tracking they proposed the Seed Growing

Graph Partition (SGGP) algorithm. The SGGP algorithm allows the network to automatically decompose into multiple sub-groups to track multiple targets in term of minimizing the power consumption and time consuming. Shi and Wang [18] 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 was proposed. Anderson *et al*. [20] 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 meeting the shape constraints . Tanner *et al*. [21], [22] studied the stability properties of a system of multiple mobile agents with double integrator dynamics in the case of fixed and dynamic topology. In addition, the experimental implementation of flocking algorithms proposed in [21] and [22] on wheeled mobile robots was presented in [23]. Gervasi and Prencipe [24] 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 [30] addressed 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 the sensors to estimate the target's position. In [25], 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.

In summary, in the flocking control most of existing works focused on the coordination, formation and splitting/merging problems in both fixed and switching topologies. In addition, the problem of controlling the size of the network in adaptive and decentralize fashion to deal with complex environment and the problem of designing the flocking control to deal with multi-target have been studied. However, the problem of how to design an optimal flocking control algorithm with optimal flocking parameters to allow the network to catch up the target as fast as possible to minimize the tracking time and power consumption is still an open research.

The rest of this paper is organized as follows. In the next section we present the distributed flocking control algorithm with obstacle avoidance for a moving target tracking and problem formulation. Section III presents the optimization problem of flocking control via genetic algorithms. Section IV presents the experimental tests. Finally, the conclusion of this paper is given in the Section V.

II. Flocking control and problem formulation

A. Distributed flocking control algorithm

We consider n robots moving in an m dimensional Euclidean space. We address the motion control problem for a robot group with the objective of dynamic target tracking. In this problem we suppose that each robot has a limited communication range to allow it to communicate with others and to sense the target.

The dynamic equation of each robot is described as follows:

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

here $q_i, p_i \in R^m$ (e.g.,$m = 2,3$) are the position and velocity of robot i, respectively.

Given an communication range r, each mobile robot only interacts with a set of its neighborhood at time t as follows:

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

where $\|.\|$ is the Euclidean norm.

The dynamic topology of all mobile robots is considered as a dynamic graph $G(\vartheta, E)$ consisting of a vertex set $\vartheta = \{1,2...,n\}$ and an edge set $E \subseteq \{(i,j) : i,j \in \vartheta, j \neq i\}$. In this topology each vertex denotes one member of flocks, and each edge denotes the communication link between two members.

The geometry of flocks is modeled by α-lattices [10] that has the following condition:

$$\|q_i - q_j\| = d \tag{3}$$

here d is a positive constant indicating the distance between robot i and its neighbor j.

Firstly, based on Olfati-Saber's flocking algorithm with obstacle avoidance [10] we design a flocking control algorithm with a dynamic γ-agent. In this scenario, the dynamic γ-agent is considered as a moving target.

$$\begin{aligned} u_i \quad = \quad & c_1^\alpha \sum_{j \in N_i^\alpha} \phi_\alpha(\|q_j - q_i\|_\sigma) n_{ij} + c_2^\alpha \sum_{j \in N_i^\alpha} a_{ij}(q)(p_j - p_i) \\ & + c_1^\beta \sum_{k \in N_i^\beta} \phi_\beta(\|\hat{q}_{i,k} - q_i\|_\sigma) \hat{n}_{i,k} + c_2^\beta \sum_{k \in N_i^\beta} b_{i,k}(q)(\hat{p}_{i,k} - p_i) \\ & - c_1^t (q_i - q_t) - c_2^t (p_i - p_t) \end{aligned} \tag{4}$$

In this control protocol, the pair $(q_t,\ p_t)$ is the position and velocity of the moving target respectively. $(c_1^\alpha,\ c_2^\alpha)$, $(c_1^\beta,\ c_2^\beta)$, and $(c_1^t,\ c_2^t)$ are positive constants. The $\sigma - norm$, $\|.\|_\sigma$ of a vector is a map $R^m \Longrightarrow R_+$ defined as $\|z\|_\sigma = \frac{1}{\varepsilon}[\sqrt{1+\varepsilon\|z\|^2} - 1]$. $\phi_\alpha(z)$ and $\phi_\beta(z)$ are the action functions to control the attractive or repulsive forces between robot i and its neighbor j, and the repulsive force between robot i and its obstacle k, respectively. n_{ij} and $\hat{n}_{i,k}$ are the vectors along the line to connect the pair $(q_i,\ q_j)$, and the pair $(\hat{q}_{i,k},\ q_i)$, respectively. $a_{ij}(q)$ and $b_{i,k}(q)$ are adjacency matrices. More details of the these terms, see [10].

The dynamic target is defined as follows:

$$\begin{cases} \dot{q}_t = p_t \\ \dot{p}_t = f_t(q_t, p_t) \end{cases} \tag{5}$$

.

In the control protocol (4), the first two terms are used to control the formation (collision avoidance and velocity matching among robots). The third and fourth terms are used to allow robots avoiding obstacles. The last terms is used for target tracking. If the last term is absent the control will lead to the fragmentation of the robot network [10].

B. Problem Formulation

The problem here is to find out the optimal solution of coefficients of the interaction forces $(c_1^\alpha,\ c_2^\alpha)$, $(c_1^\beta,\ c_2^\beta)$, $(c_1^t,\ c_2^t)$ which deliver desired swarm robots-like behavior to minimize the normalized cost function (7) while maintaining the α-lattice formation. Namely, the pair $(c_1^\alpha,\ c_2^\alpha)$ is used to adjust the attractive/repulsive forces among agent i and its actual neighbors (α-agent), and the pair $(c_1^\beta,\ c_2^\beta)$ is used to adjust the repulsive forces among agent i and its virtual neighbors (β-agent) that is generated from agent i when it see the obstacles, and the pair $(c_1^t,\ c_2^t)$ is used to adjust the attractive forces between agent i and its target. The bigger $(c_1^t,\ c_2^t)$ the faster convergence to the target. However if $(c_1^t,\ c_2^t)$ is too big the center of mass (CoM) as defined in Equation (6)

$$\begin{cases} \overline{q} = \frac{1}{n}\sum_{i=1}^{n} q_i \\ \overline{p} = \frac{1}{n}\sum_{i=1}^{n} p_i \end{cases} \tag{6}$$

oscillates around the target, and the formation of network is not guaranteed. In addition, in order to guarantee that no agent hit obstacle 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 $(c_1^t,\ c_2^t)$. Finally we have the relationship among these pairs as: $(c_{1,2}^\alpha < c_{1,2}^t < c_{1,2}^\beta)$.

The α-lattice formation can be checked by T_{cf} which is the convergent formation time and is computed based on the condition $Var(\|q_i - q_j\|) = \frac{1}{|E|}\sum(\|q_i - q_j\| - \frac{1}{n}\sum_{(i,j)\in E}\|q_j - q_i\|)^2 \le \Theta$ with $i, j = 1, 2, ...,$ n, here Θ is the given threshold, and $i \neq j$, and they are on the same edge $\in E$.

$$F = \frac{\sum_{i=1}^{n}\int_0^T \|q_i(t) - q_t(t)\| dt}{T\sum_{i=1}^{n}\|q_i(t=0) - q_t(t=0)\|} \tag{7}$$

here T is the total simulation time.

The cost function (7) represents the following terms:

-The time it takes all robots in the network to catch up the moving target.

-The distance of all robots in the network away from the target.

Clearly see that the function (7) is non-convex and non-differentiable. Hence to minimize this function the classical gradient-based deterministic optimization methods are unstable because these methods are usually trapped by the local minima. To solve this problem, Genetic Algorithms (GAs), a certain class of non-derivative search techniques, especially advanced GAs [31] such as NSGA II (Non-Dominated Sorting GA) [32], SPEA II (Strength Pareto EA) [33], RDGA (Rank-Density-based Genetic Algorithm) [34] and a temporally-adaptive iterative scheme of GA [35], etc. can be applied.

The term $\sum_{i=1}^{n}\|q_i(t=0) - q_t(t=0)\|$ in denominator of (7) is the total distance between all robots and the target at initial time. We see that this is the maximum distance because at initial time all robots are furthest from the target. In addition, the term $\sum_{i=1}^{n}\|q_i(t) - q_t(t)\|$ is the total distance between all robots and the target at time t. This distance will be decreased during tracking process when all robots apply the flocking algorithm (4), and this leads to $\sum_{i=1}^{n}\int_0^T\|q_i(t) - q_t(t)\|dt$ also decreasing. The question here is how fast the cost function (7) decreases. This depends on how the coefficients $(c_1^\alpha,\ c_2^\alpha)$, $(c_1^\beta,\ c_2^\beta)$, $(c_1^t,\ c_2^t)$ are sought. The answer of this question will be found after the function (7) is minimized by advanced GAs.

III. Optimal Design of Flocking Control via Genetics Algorithm

In this section, a genetic algorithm will be developed to deal with the non-convexity of the function (7) in the flocking control design. The similar idea of this approach can be found in [35]. The main idea here is that the flocking parameters form a genetic string and then it is tested via the objective function. A survival one is applied to a population in such string. The entire process is stated as follows.

Step 1. A population (M) of different designs are generated randomly with Gaussian distribution in the certain searching space. The genetic strings here are represented for the design parameters (6 flocking parameters as shown above).

Step 2. The performance of each design is tested via the objective function. The best genetic string is the one with the smallest objective function. Especially, if some conditions (see pseudo code) are not satisfied this genetic string will be eliminated and replaced by the new one.

Step 3. The designs are ranked from the top to the bottom according their performance which base on the fitness values.

Step 4. The best designs are mated pairwise producing two offsprings. Namely, each best pair exchanges information by taking random convex combinations of design components of the parents's genetic strings.

Step 5. The worst performing genetic strings are eliminated.

Step 6. New replacement strings are added into the remaining population of best performing genetic strings.

The pseudo code of optimal design of flocking control via genetics algorithm is given as below.

1. Randomly generate M starting genetic strings.
$\Omega^I = (\omega_1^I, \omega_2^I, \omega_3^I, \omega_4^I, \omega_5^I, \omega_6^I,) =$
$= (c_1^\alpha,\ c_2^\alpha,\ c_1^\beta,\ c_2^\beta,\ c_1^t,\ c_2^t)^I,\ I = 1, 2..., M.$

2. Compute fitness of each genetic string I, $I = 1, 2..., M$.

Test each genetic string I, and if some conditions are satisfied then compute its fitness value via fitness function $F(\Omega^I)$.

for $k = 1$ to K_d **do**
  for $i = 1$ to n **do**
    Update position of robot i.
    $q_i(k+1) = q_i(k) + \frac{p_i(k)}{\Delta_t} + \frac{u_i(k+1)}{\Delta_t^2}$.
    Update velocity of robot i.
    $p_i(k+1) = \frac{q_i(k+1) - q_i(k)}{\Delta_t}$.
    Compute iteration difference.
    $diff(i) = \|q_i(k+1) - q_i(k)\|$.
    Compute error.
    $e(k) = \frac{\sum_{i=1}^{n} diff(i)}{\sum_{i=1}^{n}\|q_i(k+1)\|}$.
    Check the convergent distance T_{cd} and the convergent formation T_{cf}.
    $T_{cd} = \|\frac{1}{n}\sum_{i=1}^{n} q_i(k) - q_t(k)\|$.
    $T_{cf} = \frac{1}{|E|}\sum(\|q_i - q_j\| - \frac{1}{n}\sum_{(i,j)\in E}\|q_j - q_i\|)^2$.

if $e(k) \le \Theta_1$, $T_{cd} \le \Theta_2$, $T_{cf} \le \Theta_3$, and $k < K_d$ **then**

Compute the fitness function.

$F(\Omega^I) = \frac{\sum_{i=1}^{n} \int_0^T \|q_i(t) - q_t(t)\| dt}{T \sum_{i=1}^{n} \|q_i(t=0) - q_t(t=0)\|}$.

else if One of $(e(k), T_{cd}, T_{cf})$ is not satisfying, and $k < K_d$ **then**

Jump to next iteration $k = k+1$

else

Not accepting this genetic string and replacing by the new one

end if

end for

end for

3. Rank the genetic strings, $\Omega^I (I = 1, 2, ..., M)$

4. Mate the nearest pairs.

$\omega^I = c_r^1 \Omega^I + (1 - c_r^1)\Omega^{I+1}$.

$\omega^I = c_r^2 \Omega^I + (1 - c_r^2)\Omega^{I+1}$, $(c_r^1, c_r^2) \in [0,\ 1]$ are the random crossover rates.

5. Kill the botom genetic strings $(B < M)$ and keep the top K parents.

6. $New\ parrents = K \cup B_{new}$ then repeat with each $\Omega^I = \omega^I (I = 1, 2, ..., M)$.

In this pseudo code, M is the number of genetic strings, K_d is the desired iterations, and n is the number of robots. $q_i(0)$ is randomly distributed in a certain space, and $p_i(0)$ is set to zero. Θ_1, Θ_2 and Θ_2 are the given thresholds. B_{new} is the new genetic strings to be introduced into the top K parents.

IV. Experimental tests

In this section we consider the trajectory of the moving target being the sine wave trajectory. The reason for selecting the sine wave trajectory is that the velocity of target in this trajectory keeps changing in whole simulation. Parameters used in this simulation are specified as follows:

- Parameters of GAs: the searching space of flocking parameters is setup as $1 < (c_1^\alpha,\ c_2^\alpha) < 100$, $100 < (c_1^\beta,\ c_2^\beta) < 1500$, $1 < (c_1^t,\ c_2^t) < 100$. The number of genetic strings is $M = 15$, and the top ten offsprings are selected in each generation. The total generation is set to $G = 15$. The desired iteration $K_d = 400$. The total simulation time is computed as $T = \Delta_t * K_d$.

- Parameters of flocking: number of sensors = 115 (randomly distributed in the box [0 100]x[0 100]). Positions of obstacles y_k= [260 135; 260 195]^T; Radii of obstacles R_k= [20; 20; 4], and other parameters $a = b = 5$; the interaction range $r = 7.8$; $\varepsilon = 0.1$ for the σ-norm; h = 0.2 for the bump function $(\phi_\alpha(z))$; h = 0.9 for the bump function $(\phi_\beta(z))$.

- Parameters of target tracking: The target moves in the sine wave trajectory: $q_t = [100 + 35t,\ 135 - 35sin(t)]^T$ with $0 \le t \le 8$, and $p_t = (q_t(t) - q_t(t-1))/\Delta_t$,, and $p_t = (q_t(t) - q_t(t-1))/\Delta_t$.

Figure 2 (a) shows the result of fitness function for the case of 45 agents. $M = 15$ is the total number of genetic strings in the population and $K = 10$ is the number of parents kept in each generation, or the last four parents $B = 5$ are killed after each generation. Figure 2 (b) depicts the tracking performances of the various size of agents. These tracking performances represent the errors between the mean of positions of all mobile sensors and the position of target. As shown in this figure, all errors go to near zero after 180 iterations.

Figures 2 (c) and 3 represent the results of the moving target tracking in the sine wave trajectory using flocking algorithm (4) with the optimal parameters,$(c_1^\alpha,\ c_2^\alpha)$, $(c_1^\beta,\ c_2^\beta)$, $(c_1^t,\ c_2^t)$, as shown in the Table 1. In these figures we see that all mobile sensors (115) track the moving target in the free space (without obstacles) very well with small error, or all mobile sensors can surround the target. However, in the obstacle space, the error between the mean of positions of all sensors and the position of target as shown in Figures 2 (c) is remarkable. This phenomenon can be explained in the Remark 1.

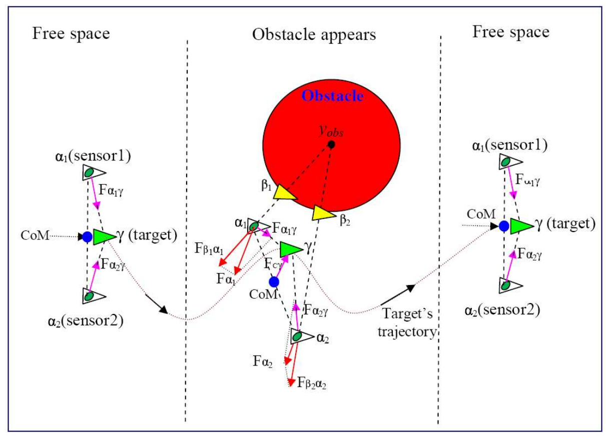

Fig. 1. Demonstration of two sensors tracking the moving target in both free and obstacle spaces.

Remark 1. To explain why the tracking performance in presence of obstacles is not good as in the free space, we analyze the forces acting on the α-agents (sensors) when they pass through the obstacles as shown in the Figure 1. In this figure, without losing general we simply consider two sensors tracking the target (γ-agent) which moves in arbitrary trajectory.

Firstly, when two sensors track the target in the free space (without obstacle), in the equilibrium state the CoM (Center of Mass) defined as (6) is close to the target (in this case sensor2 is the neighbor of sensor1). The total interaction forces between two sensors are equal to zero, and also because of velocity matching the the sum of different velocities between these sensors is equal to zero. Hence from (4) we have:

$$\overline{u} = \frac{1}{n}\sum_{i=1}^{n} u_i \quad = \quad -(c_1^t)(\overline{q} - q_t) - (c_2^t)(\overline{p} - p_t) \qquad (8)$$

The Equation (8) means that the CoM $(\overline{q},\ \overline{p})$ converges to the target $(q_t,\ p_t)$ (more details please see [26]).

When these sensors move in the obstacle space (the obstacle in the sensing ranges of both sensors) the projection of each sensor on the surface of obstacle is called β-agent (virtual agent). In this scenario, two β-agents, β_1 and β_2, of two sensor1 and sensor2 are created, respectively (see Figure 1).

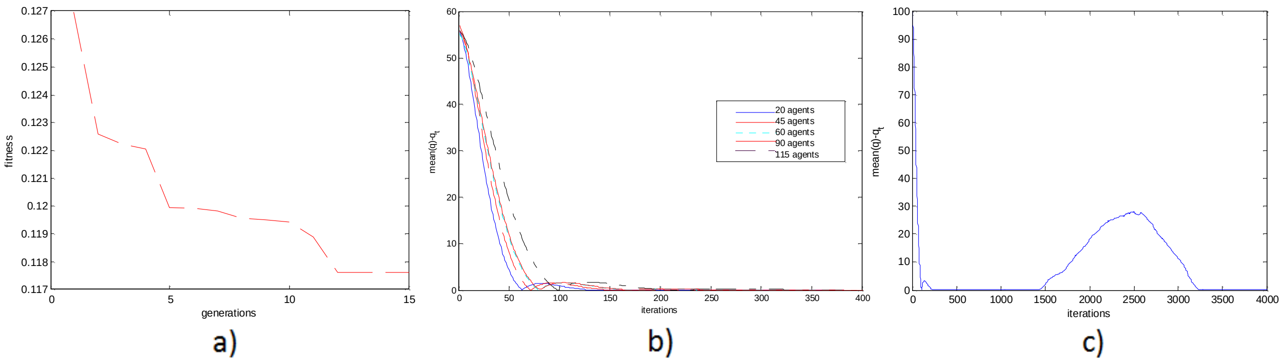

Fig. 2. (a) The values of fitness function during 15 generations for 45 agents case, (b) Errors between the mean of positions of all mobile sensors and the position of target for only 400 iterations, (c) Error between the mean of positions of all mobile sensors and the position of target for whole tracking process in the case of 115 agents.

TABLE I
THE OPTIMAL COEFFICIENTS FOR VARIOUS SIZES OF FLOCKING CONTROL.

Number of agents	c_1^α	c_2^α	c_1^t	c_2^t	c_1^β	c_2^β
20	31.99523	23.32457	45.05901	32.92191	919.91231	52.57912
45	33.00721	24.65109	46.08719	34.66901	940.15267	56.33411
60	35.32712	25.66322	46.45919	36.67810	949.91510	64.45910
90	38.79923	26.73234	47.13214	39.74959	1090.92536	70.53211
115	41.48864	26.89120	50.29770	43.87073	1200.88581	71.44092

These β-agents generate the repulsive forces, $F_{\beta,1\alpha_1}$ and $F_{\beta_2\alpha_2}$ to push these sensors away from the obstacle. However, the presence of γ-agent (target) is necessary to steer both sensors around the obstacle. The synthesized forces $\vec{F}_{\alpha_1} = \vec{F}_{\beta_1\alpha_1} + \vec{F}_{\alpha_1\gamma}$ and $\vec{F}_{\alpha_2} = \vec{F}_{\beta_{12}\alpha_2} + \vec{F}_{\alpha_2\gamma}$ of sensor1 and sensor2, respectively. To avoid each sensor hitting the obstacle the weights of repulsive force of the obstacle c_1^β, c_2^β are set bigger than that of attractive force between the target and each sensor c_1^t, c_2^t, this leads to the force $F_{\beta_1\alpha_1} > F_{\alpha_1\gamma}$ and $F_{\beta_2\alpha_2} > F_{\alpha_2\gamma}$. This causes these sensors being pushed away in a certain distance from the target, or the CoM is no longer close to the target. Otherwise, in general situation (more than two sensors) a sensor i might stay near a distance r_0 $(0 < r_0 < r)$ from the obstacle for a long time due to "*peer panic*" by other sensors j that are on the way of sensor i and keep sensor i back [36]. In all above simulation results, all sensors keep their formation and no collision occurs among them while tracking the moving target, and all sensors avoid obstacles successfully in a narrow space. For more details please see the video file posted in the below link:

http://www.youtube.com/watch?v=5dfdo4L07jY

V. CONCLUSION

This paper studied the optimization problem in the distributed flocking control for a mobile sensor network to track a moving target. The non-convex objective function was constructed, then the optimization method based on GAs was developed. The optimal flocking coefficients for various size of members in flocking control algorithm were found out. Analysis of the flocking behavior based moving target tracking in both free space and obstacle space is given. The numerical tests were shown to validate our approach.

ACKNOWLEDGEMENT

We would like to thank the Vietnamese Government, and specially thank the MOET (Ministry of Education and Training), which supported us to implement this project.

REFERENCES

[1] F. Akyldiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. A survey of sensor networks. *IEEE Communications Magazine*, pages 102–114, 2002.

[2] S. Kamath, E. Meisner, and V. Isler. Triangulation based multi target tracking with mobile sensor networks. *IEEE International Conference on Robotics and Automation*, pages 3283–3288, 2007.

[3] A. Ganguli, S. Susca, S. Martinez, F. Bullo, and 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*, pages 4239–4244, December 12-15 2005.

[4] E. Biagioni and K. Bridges. The application of remote sensor technology to assist the recovery of rare and endangered species. *Special issue on Distributed Sensor Networks for the International Journal of High Performance Computing Applications*, 16(3):112–121, 2002.

[5] H. M. La and 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.

[6] R. M. Murray. Recent research in cooperative control of multivehicle systems. *Journal of Dynamic Systems, Measurement, and Control*, pages 571–583, 2007.

[7] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry. Distributed control applications within sensor networks. *Proceedings of the IEEE*, 91(8):1235–1246, 2003.

[8] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[9] Z. Lin, M. E. Broucke, and B. A. Francis. Local control strategies for groups of mobile autonomous agents. *IEEE Transactions on Automatic Control*, 49(4):622–629, 2004.

[10] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

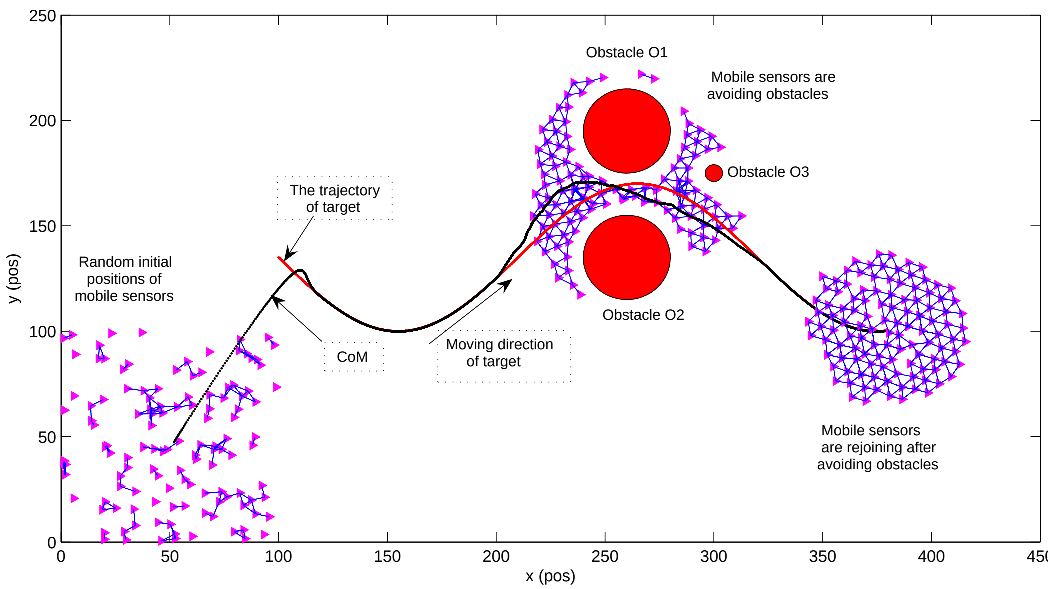

Fig. 3. Snapshots of the beginning positions, avoiding obstacles positions and ending positions of 115 mobile sensors which are tracking the target moving in the sine wave trajectory.

[11] C. Reynolds. Flocks, birds, and schools: A distributed behavioral model. *Computer Graphics,ACM SIGGRAPH '87 Conference Proceedings, Anaheim, California*, 21(4):25–34, 1987.

[12] T. Vicsek, A. Czirok, E.B. Jacob, I. Cohen, and O. Schochet. Novel type of phase transitions in a system of self-driven particles. *Phys. Rev. Lett*, 75:1226–1229, 1995.

[13] H. Levine, W. J. Rappel, and I. Cohen. Self-organization in systems of self-propelled particles. *Phys. Review. E*, 63:017101–017104, 2000.

[14] A. Mogilner, L. Edelstein-Keshet, L. Bent, and A. Spiros. Mutual interactions, potentials, and individual distance in a social aggregation. *J. Math. Biol*, 47:353–389, 2003.

[15] I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective memory and spatial sorting in animal groups. *J. Theor.Biol*, 218:1–11, 2002.

[16] H. Su, X. Wang, and 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*, pages 2937–2942, 2007.

[17] H. Su, X. Wang, and 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*, pages 1429–1434, 2007.

[18] H. Shi, L. Wang, T. Chu, F. Fu, and M. Xu. Flocking of multi–agents with a virtual leader. *in Proc of the 2007 IEEE Symposium on Artificial Life*, pages 287–29, 2007.

[19] Z. Wang and 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*, pages 1–8, 2006.

[20] M. Anderson, E. McDaniel, and S. Chenney. Constrained animation of flocks. *Eurographics/SIGGRAPH Symposium on Computer Animation*, pages 1–12, 2003.

[21] H. G. Tanner, A. Jadbabai, and G. J. Pappas. Stable flocking of mobile agents, part i: fixed topology. *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2010–2015, 2003.

[22] H. G. Tanner, A. Jadbabai, and G. J. Pappas. Stable flocking of mobile agents, part ii: dynamic topology. *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 2016–2021, 2003.

[23] A. Regmi, R. Sandoval, R. Byrne, H. Taner, and C. T. Abdallah. Experimental implementation of flocking algorithms in wheeled mobile robots. *American Control Conference*, pages 4917–4922, 2005.

[24] V. Gervasi and G. Prencipe. Coordination without communication: the case of the flocking problem. *Discrete Applied Mathematics*, pages 324–344, 2004.

[25] D. Cruz, J. McClintock, B. Perteet, O. A. orqueda, Y. Cao, and R. Fierro. Decentralized cooperative control–a multivehicle platform for research in networked embedded systems. *IEEE Control Systems magazine*, pages 58–78, 2007.

[26] H. M. La and W. Sheng. Flocking control of a mobile sensor network to track and observe a moving target. *in Proceedings of the 2009 IEEE International Conference on Robotics and Automation (*ICRA 2009*), Kobe, Japan*, 2009.

[27] H. M. La and W. Sheng. Adaptive flocking control for dynamic target tracking in a mobile sensor network. *Submitted to the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (*IROS 2009*), St. Louis, MO, USA*, 2009.

[28] H. M. La and W. Sheng. Moving targets tracking and observing in a distributed mobile sensor network. *in Proceedings of the 2009 American Control Conference (*ACC 2009*), St. Louis, MO, USA*, 2009.

[29] H. M. La and W. Sheng. Dynamic targets tracking and observing in a mobile sensor network. *Submitted to the Elsevier journal on Robotics and Autonomous Systems*, 2009.

[30] R. Olfati-Saber. Distributed tracking for mobile sensor networks with information driven mobility. *Proceedings of the 2007 American Control Conference,*, pages 4606–4612, 2007.

[31] L. Davis. Handbook of genetic algorithms. *Van Nostrand Reinhold, New York*, 1991.

[32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[33] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2:Improving the strength pareto evolutionary algorithm. *TIK-Report 103-Swiss federal institue of technology (*ETH*) Zurich*, pages 1–21, 2001.

[34] H. Lu and G. G. Yen. Rank-density-based multiobjective genetic algorithm and benchmark test function study. *IEEE transactions on evolutionary computation*, 7(4):325–343, 2003.

[35] T. I. Zohdi. Computational design of swarms. *International journal for numerical methods in engineering*, 57:2205–2219, 2003.

[36] R. Olfati-Saber. Flocking with obstacle avoidance. *Technique Report 2003-006, California Institute of Technology,* CA,, 2003.