International Journal of Performability Engineering, Vol. 8, No. 2, March, 2012, pp. 141-150.

Printed in India

Mobile Sensor Deployment and Coverage Using Multi-Agent-based Collective Formation Schemes

ZHAO CHENG [1], LI BAI [1 *], DHRITIMAN CHOUDHURY[1], SAROJ BISWAS [1] and JIE WU[2]

[1] *Department of Electrical and Computer Engineering, Temple University, PA 19027*

[2]*Department of Computer Science, Temple University, PA 19027, USA*

(Received on October 30, 2010; Revised on April 18, 2011)

Abstract: In this paper, we present a novel mobile wireless sensor deployment and network coverage technique using multi-agent-based collective formation control. Most of the existing approaches of sensor deployment are focused on centralized methods, which restrain the sensor nodes to maintain fixed distance among all neighboring nodes. Therefore, these approaches have some drawbacks, vulnerabilities and inflexibility, especially when some of the sensor nodes are not functioning due to unexpected node failure, *e.g.*, power loss. As a result, sensor coverage could be compromised or diminished. To address these problems, we propose to incorporate an *attractive/repulsive* (AR) collective formation model to control the dynamics of wireless sensor nodes which are considered as autonomous agents. We show that sensor deployment with AR model provides robustness and flexibility. When some sensor nodes are lost unexpectedly, movement of neighboring nodes are relatively localized so that the least amount of energy will be used to regain control of the network coverage. Consequently, the proposed method can significantly improve the time-efficiency, network stability and sensing coverage when sensor nodes are deployed to explore harsh terrains and unpredictable environments.

Keywords: *Mobile sensor network, collective formation, sensing coverage, multi-agent system*

1. Introduction

Sensor management in mobile sensor networks has attracted a lot of attention and research interest in recent years. The self organizing characteristics of collective motion in nature, such as a flock of birds, a school of fish, or a swarm of locusts, are resourceful stimuli to the study of sensor management. Collective motion has been studied in [6], [12], [16], [18], [19] in which agents (or self-propelled particles) move agreeing upon certain quantities of interest, such as position, temperature, and voltage etc. These quantities of interest have implications on general design of mobile sensor networks, sensor network data fusion, attitude alignment of satellite clusters, congestion control of communication networks and multi-agent formation control [1], [2], [9], [11]. A flocking model of multi-agent formation was proposed by Reynolds [15] where three heuristic rules were assumed: (i) separation: steer to avoid crowding and collision, (ii) alignment: steer towards an average heading, and (iii) cohesion: steer to move towards the average position. These rules have been proven effective and are often used in the design of bio-group dynamic models (see [5], [13], [14]).

In 2004, Olfati-Saber [13] developed a theoretical framework for flocking dynamics. In this work, a mass model is proposed based on the Newton's law of motion which

[*]Corresponding author's email: lbai@temple.edu

shows that the moving agents eventually gather and form a lattice. The case of flocking in free-space with multiple obstacles is also considered. Simulation results showed that the formed lattice had good performance in avoiding complex obstacles. Stability of formation was also proved using Lyapunov's approach based on collective potential of the agents. Vicsek [17] discussed the universal patterns of formation of living organisms as well as non-living objects, such as interacting robots. He focused on one of the most common and spectacular manifestation of coordinated actions, which describe the essential aspects of collective motion of a wide selection of systems ranging from colonies of tissue cells to flocks of birds to collectively moving robots. It has the potential of improving the interpretation of collective behavior in both living and inert systems to understanding the interrelation of the systems by learning similar phenomena in the two domains of nature.

Recently, Juanico [8] proposed a modified kinematic model with AR pair-wise interaction, and showed an interesting simulation that shows that agents form several stellate patterns due to changes of distribution of preferred pair-wise length. He also gave the definition and analysis of order parameter as a measure of pattern meta-stability. Although the variation among moving agents is a fact rooted in natural and artificial swarm systems, the role of diversity in the self-organized pattern formation has not been previously explored. In 2009, Chen and Cheng [3] introduced the AR functional link between each pair of agents. By changing the slope of an AR function, they found a dramatic transition between two different formation patterns. In the liquid-like pattern, the outer agents are sparsely distributed while the inner ones are packed densely. In contrast, agents maintain a constant distance from each other in a crystal-like pattern.

A sensor network is a networked system that is composed of a large number of sensor nodes [1]. Mobile sensor networks are sensor networks in which sensor nodes have the capability of motion under their own control. In 2004, Ogren [11] presented a stable control strategy for sensor nodes to move and reconfigure cooperatively in response to a sensed and distributed environment. Gradient climbing strategies are applied to artificial potentials to drive the sensor nodes. Nguyen [10] introduced the concept of an artificial potential field to guide the movement of a robot. In his research, the robot was built with artificial potential fields which was able to navigate itself to a particular location through a path with obstacles. Later, the artificial-potential filed approach was applied in the management of a mobile sensor network to improve sensor performance. Heo [7] proposed and analyzed distributed energy-efficient deployment algorithms for mobile sensor networks. In 2008, Ma [9] introduced non-Newton's model and discussed sensor coverage problems in sensor management of mobile sensor networks. This research is based on simulations and analysis in three cases: spatial coverage, spatial migration and spatial retreats.

In this paper, we propose the use of AR dynamics for mobile wireless sensor deployment to maximize sensor coverage. We mainly used simulation to show performance and robustness of distributed sensor deployment schemes. We demonstrated that the crystal-like deployment scheme is suitable for exploring open areas without many obstacles, however liquid-like deployment scheme performs better in a narrow and short pathway or environment with many obstacles. We show that even if some sensor nodes are lost unexpectedly, movements of neighboring sensor nodes are relatively localized so that the least amount of energy will be used to regain control of the network coverage. Our proposed method can significantly improve the time-efficiency, network stability and sensing coverage when deploying sensor nodes to explore harsh terrains and unpredictable environments.

The rest of the paper is organized as follows: Section 2 details some background research, Section 3 provides some simulation results and discussions, and Section 4 presents the conclusion and some ideas for future work.

2. Models of Collective Motion

There are three popular types of models describing relationship between sensors nodes in collective groups: i) the Unit-vector Model, ii) the Kinematic Model and iii) the Mass Model. The Unit-vector Model is a steering control model, in which the direction of each agent is determined by a control algorithm. However, in the third model, the Mass Model, agents obey the Newton's law of motion: both direction and speed of each agent is updated at each step. Similarly the second model, named as Kinematic Model, is also a system whose agents have both their direction and scalar velocities updated according to the control algorithm. But, they are still different, since the control input is applied directly on the velocity of each agent in Kinematic Model instead of acceleration (derivative of velocity) in the Mass Model. Thus, the Kinematic Model can also be viewed as an approximation of Mass Model. Note that the Kinematic Model is a first-order system while the Unit-vector Model and the Mass Model are second-order systems. Generally, the second-order systems have better stability performance than first-order systems. The Mass Model is also the closest one to movement mechanism of a real object over all three models. Because of these reasons, the Mass Model is adopted and discussed in the following sections.

Consider a group of *n* moving agents (or particles) in a square cell with periodic boundary condition with equations of motion. It is common to study the communication problems among agents by means of a graph. A graph G is defined as a pair (V,E) consisting of a set of vertices $\mathsf{V}=\{1,2\ldots N\}$ and a set of edges $\mathsf{E}\subseteq\{(i,j):i,j\in\mathsf{V},i\neq j\}$. The moving agents are the vertices of the graph G and the connection between a pair of agents is an edge defined in G. Let $x_i,v_i,u_i\in\mathsf{R}^m$ denote the position, velocity and control input of node *i* for all $i\in\mathsf{V}$, respectively in the Euclidean space, R^m. Let *r* denote the interaction range between two agents, then the set of spatial neighbors of agent i is defined by

$$N_i=\left\{j\in V:\left\|x_j-x_i\right\|<r\right\} \tag{1}$$

where $\|\cdot\|$ is the Euclidean norm in R^m. Define $|N_i|$ as the total number of elements in the set N_i. Fig. 1 shows the pairwise interaction of the agent 1. The agents 2 and 3 are neighboring agents of agent 1 since their distance to agent 1 is less than r.

The pairwise force function between two sensor nodes is represented as ψ which varies according to the interval x_{ij} between agents i and j. It reaches the global minimum 0 at $x=d$ (see Fig. 2) which is the equilibrium point, and *r* is the scope of an agent. The ratio of the scope and the equilibrium point of an agent is denoted by $k=r/d$.

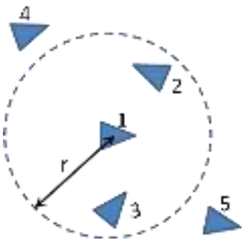


Figure 1: Pairwise Interaction of an Agent

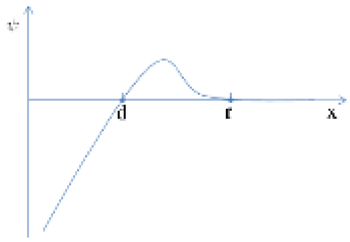


Figure 2: Pairwise Force $\psi(x)$

The equation of motion is defined as,

$$\begin{cases} \dot{x}_i = v_i \\ \dot{v}_i = u_i \end{cases} \tag{2}$$

where $i \in V$ and the control input of agent i consisting of three terms

$$\begin{aligned} u_i \quad = \quad & \underbrace{w_1 \sum_{j \in \mathrm{V}} \psi(\|x_j - x_i\|) n_{ij}}_{gradient term} + \underbrace{w_2 \sum_{j \in \mathrm{N}_i} (v_j - v_i)}_{consensus term} \\ + \quad & \underbrace{w_3 \left(-k_x (x_i - x_r) - k_v (v_i - v_r)\right)}_{navigation term} \end{aligned} \tag{3}$$

where n_{ij} is a unit vector pointing from x_i to x_j, $n_{ij} = \dfrac{x_j - x_i}{\|x_j - x_i\|}$, $i \in \mathrm{V}$, w_1, w_2 and w_3 are weights for gradient term, consensus term and navigation term, respectively. The gradient term is the Attractive/Repulsive force determined by the distance between two particles, which guides two particles to move to a desired distance at *x*=*d* and avoid collision (due to repulsive force). The consensus term ensures the velocity consensus with other particles, while the navigation term is actually proportional feedback control based on desired distance and/or velocity. Note that the navigation term is not necessary if there is no position or velocity goal assigned to the sensors/particles. In this paper, we consider the control input depending on the gradient and the consensus terms only, i.e., $w_1 = 1$, $w_2 = 1$ and $w_3 = 0$. Note that the velocity consensus term is only required for simulations presented in Fig. 4-6. For other simulations discussed in this paper, this term is not required. Since the consensus term takes effect only when particles are moving with unsynchronized speed, it will be zero when particles are stable.

To define the ψ function, we consider two piecewise functions i) an exponential function (attractive force) and ii) a second-order polynomial function (repulsive force) as

$$\psi(x) = \begin{cases} Ax^2 + Bx + a, & x \in [0, d), \\ \dfrac{b}{c}(x-d)\exp\left(-\dfrac{(x-d)^2}{c}\right), & x \in [d, \infty). \end{cases} \tag{4}$$

The second-order polynomial $Ax^2 + Bx + a$ must satisfy the following conditions to ensure continuity and differentiability of $\psi(x)$ at $x = d$:

$$\psi(x)|_{x\to d^+} = \psi(x)_{x=d^-} \qquad \psi'(x)|_{x\to d^+} = \psi'(x)_{x=d^-}$$

The parameters A and B can be easily determined by three other parameters a, b, and c, through ensuring continuity and differentiability at $x = d$, i.e.,

$$A = \frac{a}{d^2} + \frac{b}{cd} \qquad B = -\frac{2a}{d} - \frac{b}{c}$$

Parameters *a* and *b* are related to the size of simulation and determine the maximum value of repulsion and attraction force, respectively. To make a square area with edge value of 200, we let a=5 and b=0.2. We have two free parameters *c* and *d*. The parameter *d* is an equilibrium point of two agents and determines whether two agents attract or repel. The parameter *c* determines *r* (cutoff distance), i.e., the distance within which neighboring agents interact. To ensure stability, we need to find the appropriate range for parameter *c*. A good approximation (see [4]) of collective behavior to ensure stability requirements is obtained when we have $a = 5$, $b = 0.2$ and $5 \le c \le 17$. Also, Cheng [4] found that two distinct patterns are determined by the value of k

1) $1 < k < 2$: agents form collective behavior as a crystal-like pattern, and
2) $k \ge 2$: agents form collective behavior as a liquid-like pattern.

3. Simulations of Sensor Management

Collective behavior has many applications. Here, we will demonstrate its use in mobile wireless sensor network deployment by tuning the parameter *k* for i) liquid-like deployment scheme ($k > 2$) and ii) crystal-like deployment scheme ($1 < k < 2$). Through simulation, we will make a detailed analysis for spatial coverage, obstacle avoidance and fault-tolerant of two deployment schemes.

We mainly focus on the comparison between a liquid-like deployment scheme and a crystal-like deployment scheme. Three case studies are presented based on simulations in a 2D square cell with elastic boundaries, 1) area covered analysis, 2) fault tolerance analysis, and 3) obstacle avoidance analysis.

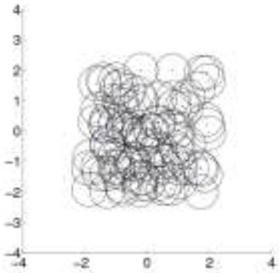
In the following simulations, N particles are generated with random velocity and position. The agents are synchronized and move autonomously according to equation (3) at each simulation step *t*.

3.1 Area Covered Analysis

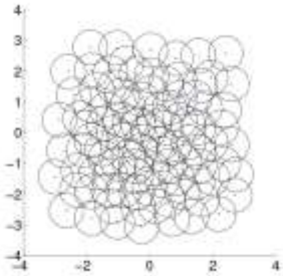
First, we look at the sensor coverage of our proposed method. Let N agents deploy in a small boundary area. Initially, we expect that agents will expand throughout the whole area (due to repulsive force) and finally fill the entire square area (see Fig. 3). In this figure, each dot represents a particle and the edge connecting two particles is the connection paired between two neighboring particles.

Sub-figures are presented to show the progression of sensor node expansion at a) $t = 0$, b) $t = 50$, c) $t = 250$ and d) $t = 1{,}250$. In this simulation, we can see that agents are initialized in a small square area but expand and cover the whole area quickly (roughly at $t = 250$). This expansion is mainly dominated by the repulsive force defined in equation (3) because agents are initialized in a small contained area.

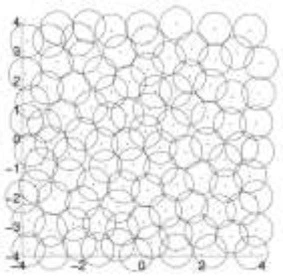
For a large sensor coverage area, we can see how agents can quickly reorganize and change sensor coverage destinations as shown in Fig. 4. They represent a crystal-like deployment scheme and a liquid-like deployment scheme, respectively. The objective of this simulation is to show robustness of the proposed sensor deployment scheme. The simulation is set up in such a way that a group of nodes are directed to a specific direction and is requested to change the sensor coverage area. The trajectory of the group shows a stable and rigid sensor coverage area. However, we can see a significant difference at the vertex of direction changes for both deployment schemes. The crystal-like scheme requires a larger area to redirect the deployment direction. In contrast, the liquid like scheme is not affected by the change of direction. It suggests that the liquid like deployment scheme might be more robust for sensor coverage problems that require many direction changes.



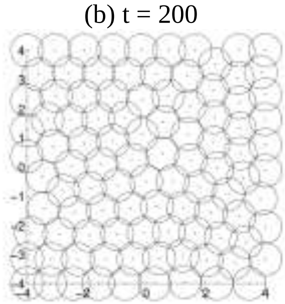
(a) t = 0



(b) t = 200

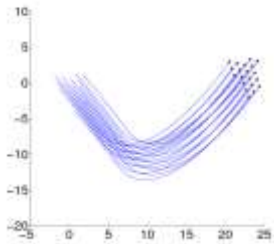


(c) t = 500

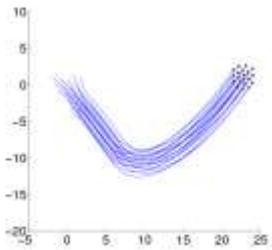


(d) t = 4000

Figure 3: Mobile Sensors Span and fill a Square Area.



(a) A crystal-like deployment scheme



(b) A liquid-like deployment scheme

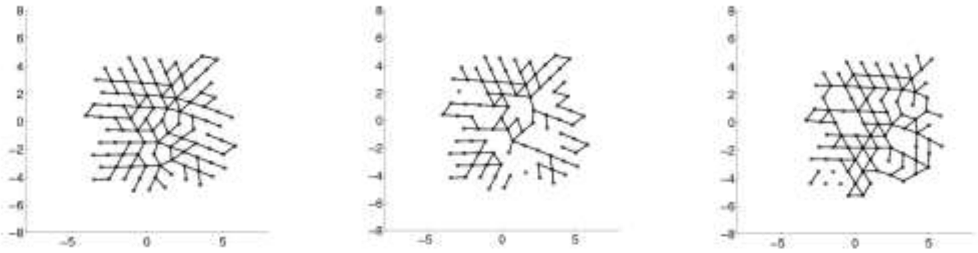
Figure 4: Mobile Sensors Movement in Different Deployment Schemes

Also, agents are compact in a liquid-like deployment scheme whereas agents have flexibility in crystal-like deployment scheme. For this reason, agent paths in the crystal-like deployment scheme are wider due to repulsive behavior among agents. Velocity is another factor that plays a vital role in direction changes. The combination of velocity and repulsive force affects the agents' movements in crystal-like deployment. Consequently, we see that agents can move more robustly in the liquid like deployment scheme.

3.2 Fault Tolerance Analysis

For this study, we evaluate how network topology recovers from random sensor node failures. It gives us the ability to understand how fault tolerant two deployment schemes are. In this simulation, some sensor nodes were removed randomly at t = 2,000, and we observe that the remaining sensors regroup quickly and regain control of the whole sensor coverage area. Total simulation time is 5,000 steps.

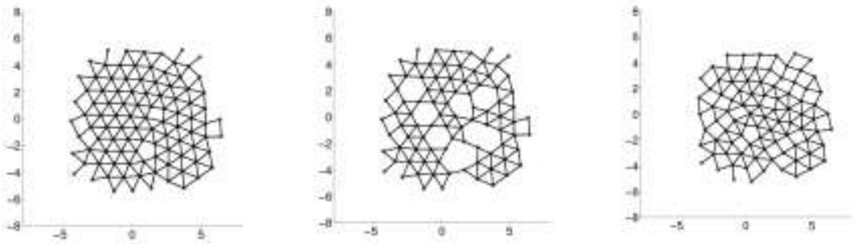
Figure 5 illustrates how the network topology recovers in the crystal-like deployment scheme. At t = 2,000, some sensor nodes are randomly removed. Sensor nodes regrouped and reconstructed the formation, however some nodes are still not connected at t = 5,000.



(a) t = 1980 (b) t = 2000 (c) t = 5000

Figure 5: Fault Tolerance of Mobile Sensor Network in Crystal-like Deployment Scheme

Figure 6 shows that the network topology recovers in the liquid-like deployment scheme. At t = 2,000, some sensor nodes are randomly removed. We observe that the remaining sensor nodes regrouped and reconstructed the formation. At t = 5,000, the remaining sensors are fully connected with neighboring nodes.



(a) t = 1,980 (b) t = 2,000 (c) t = 5,000

Figure 6: Fault Tolerance of Mobile Sensor Network in Liquid-like Deployment Scheme

From these two simulations, we see the liquid-like deployment scheme has better recoverability because sensor nodes are reconnected quickly. In comparison, the crystal-like deployment scheme takes much longer time to reconstruct the formation and to regain network coverage. The main reason is that the liquid like deployment scheme has the longer distance r between two adjacent nodes in the outskirts of the formation where they interact. Also, the liquid-like schemes allow any formation shape. Even when some nodes are removed abruptly, the blank spaces are quickly filled up by the rest of the nodes. Repulsive force plays another vital role for stabilization of the formation. Nodes in the crystal-like deployment scheme are too close to each other, and are not aware of nodes being missing. In order to fill up the blank space, partial group nodes cannot complete the recovery process unless all nodes know that they need to fill the empty coverage gap. Hence, it will take longer time to complete the whole system to regain control of network coverage.

3.3 Obstacle Avoidance Analysis

This subsection presents the collective behavior of sensor nodes when an area of coverage has obstacles. We can see that nodes behave differently among two sensor deployment schemes.

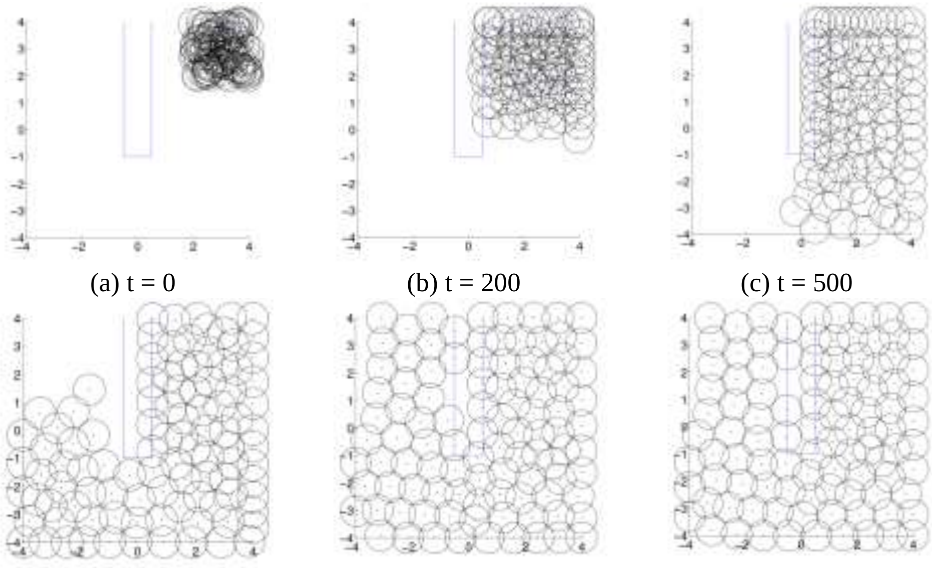
Figure 7 represents how efficient and effective sensor nodes deploy in the whole coverage area with an obstacle. For our simulation, a 2D elastic plane of 8x8 units is chosen. The wall of the square is considered to be an elastic boundary. In other words, sensor nodes will experience a backward force when nodes encounter the wall. The square plane is partially split by a wall (or obstacle) of length 5x1 units which provides a narrow pathway. The objective of the sensor group is to cover up the whole space by avoiding the obstacle. Each dot in the simulation represents a sensor node, and the circle surrounding each dot represents the interaction range of the node agent. Initially, agents are concentrated at the top right corner of the plane. Due to the velocity of agents and repulsive behavior among the agents, they will start covering the square plane avoiding the obstacle. The results are depicted in Fig. 7, where a) $t = 0$, agents are initialized and randomly distributed in the top right corner; b) t=200, agents start to expand due to the velocity and repulsive forces among neighboring node agents; c) $t = 500$, agents start to cover up more than 50% of the area; d) t = 2,000, agents cover more than 75% of the area; e) t = 5,000, agents cover most of the area; and f) $t = 8{,}000$, agents cover almost the whole area and nodes stop moving. Every node connects with at least another agent.

We have also tested other scenarios of obstacles and coverage area by changing the x-axis length from -2 to 2 units y-axis blocking length from 0-7 units with increments of 0.5 unit. The complete coverage time varies slightly due to different setting of the obstacle; however the simulations runs demonstrate that mobile sensors will eventually cover the whole area.

In this paper, we mainly focused on two deployment schemes. Formation patterns of nodes depend on the parameter c. For this example, we can tune the agent group as the crystal-like deployment scheme by setting $c = 0.06$, and as liquid-like deployment scheme by setting $c = 0.20$. Parameter d also plays the key roles in deployment scheme. We can use parameter d to determine the interval among the node agents. Parameter d is set as 1 and 2 for crystal-like and liquid-like deployment schemes, respectively. Figure 8 shows how a sensor group can cover the area with an obstacle quickly by turning parameter c from crystal-like deployment ($c = 0.06$) schemes to liquid-like deployment schemes ($c = 0.2$). Parameter d can also vary with respect to parameter c. When parameter $c = 0.06$, the simulation time to cover up the area with an obstacle is long. From the simulation, we can conclude that we need a longer time to cover an area with an obstacle if we use a crystal like deployment scheme. As we change the parameter c close to 0.20, the time to cover the area becomes faster and with smaller fluctuations. This signifies that, we can deploy mobile sensor nodes in liquid-like deployment schemes to gain quick coverage, even with an obstacle.

4. Conclusion and Future Work

In this paper, we developed a mobile wireless sensor deployment technique using a multi-agent-based AR collective formation control approach. The proposed approach and analysis embodies several tunable parameters for different deployment schemes in order to incorporate the formation of the model to control the dynamics of the sensor nodes, which considered as multiple autonomous agents. In our future work, we can consider problems such as coverage areas with more obstacles. Also, we can develop more performance-based metrics to analyze two different deployment schemes.



(a) t = 0 (b) t = 200 (c) t = 500

(d) t = 2000 (e) t = 5000 (f) t = 8000

Figure 7: Mobile Sensor Nodes Fill in Partially Blocked Area.

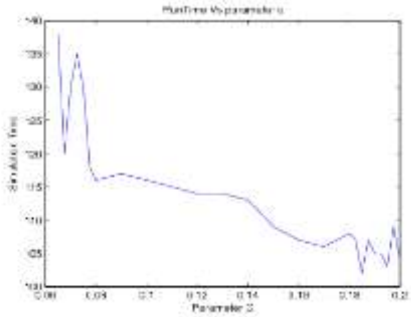


Figure 8: Complete Overage Time for Mobile Wireless Sensor Nodes

References

[1] Akyildiz. I. F. and H. Rudin. *Computer networks, the journal, and computer networks, the technology.* Computer Networks-the International Journal of Computer and Telecommunications Networking 2002; 40: 475–475.

[2] Bertsekas. D. P. and J. N. Tsitsiklis. *Comments on* "*coordination of groups of mobile autonomous agents using nearest neighbor rules*". IEEE Transactions on Automatic Control 2007; 52: 968–969.

[3] Chen. M. Z. Q., Z. Cheng, H.-T. Zhang, T. Zhou, and I. Postlethwaite. *Collective aggregation pattern dynamics control via attractive/repulsive function*. in Complex Sciences, ser. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, J. Zhou, Ed. Springer-Verlag, 2009; 5: 2064–2077.

[4] Cheng. Z., *Collective behavior and sensor network a multi-agent dynamic system approach*. Master's thesis, Temple University, August 2010.

[5] Gazi. V. and K. M. Passino. *Stability analysis of swarms*. IEEE Transactions on Automatic Control 2003; 48: 692–697.

[6] Gregoire. G. and H. Chate, *Onset of collective and cohesive motion*. Phys Rev Letter, 2004; 92: 025702.
[7] Heo N. and P. K. Varshney. *Energy-efficient deployment of intelligent mobile sensor networks*. IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans 2005; 35: 78–92
[8] Juanico. D. E. O. *Self-organized pattern formation in a diverse attractive-repulsive swarm*. EPL 2009; 86 (4).
[9] Ma. K., Y. Y. Zhang, and W. Trappe. *Managing the mobility of a mobile sensor network using network dynamics*. IEEE Trans. on Parallel & Distributed Systems 2008; 19: 106–120.
[10] Nguyen. B., Y.-L. Chuang, D. Tung, C. Hsieh, Z. Jin, L. Shi, D. Marthaler, A. Bertozzi, and R. Murray. *Virtual attractive-repulsive potentials for cooperative control of second order dynamic vehicles on the caltech mvwt*. Proc. 2005 Am. Control Conf 2005; pp. 1084–1089.
[11] Ogren. P., E. Fiorelli, and N. E. Leonard. *Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment*. IEEE Transactions on Automatic Control 2004;. 49: 1292–1302.
[12] Olfati-Saber. R. *Flocking for multi-agent dynamic systems: Algorithms and theory*. IEEE Transactions on Automatic Control 2006; 51: 401–420.
[13] Olfati-Saber. R. and R. M. Murray. *Consensus problems in networks of agents with switching topology and time-delays*. IEEE Trans. on Automatic Control 2004; 49: 1520–1533.
[14] Ren. W. and R. W. Beard. *Consensus seeking in multiagent systems under dynamically changing interaction topologies*. IEEE Trans. on Automatic Control 2005; 50: 655–661.
[15] Reynolds. C. W. *Flocks, herds, and schools: A distributed behavioral model*. Computer Graphics 1987; 21: 25–34.
[16] Vicsek. T., A. Czirok, E. Ben-Jacob, I. I. Cohen, and O. Shochet. *Novel type of phase transition in a system of self-driven particles*. Phys Rev Letter 1995; 75: 1226–1229.
[17] Vicsek. T. *Universal patterns of collective motion from minimal models of flocking*. Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, SASO '08, 2008; pp. 3–11.
[18] H.-T. Zhang, M. Z. Q. Chen, and T. Zhou. *Improve consensus via decentralized predictive mechanisms*. EPL 2009; 86 (4).
[19] Zhang. H.-T., M. Z. Chen, G.-B. Stan, T. Zhou, and J. M. Maciejowski. *Collective behavior coordination with predictive mechanisms*. IEEE Circuit & Systems Magazine 2008; 3:65–87.

Zhao Cheng received his Master's degree from Temple University. He is currently a PhD candidate in Dartmouth College. His research interest includes, cooperative control for multi-agent systems, mobile sensor networks, robotics, medical/acoustic imaging.

Li Bai is an Associate Professor at Temple University. His research interests are performability in distributed systems, such as reliability, security and monitoring, and multi-agent systems.

Dhritiman Choudhury received his Master's degree from Temple University in Electrical and Computer Engineering. Currently he is working in Siemens mobility under a project with MTA. His area of interest includes distributed sensor networks, robotics, telemedicine systems, multi-agent systems.

Saroj Biswas is a Professor of Electrical and Computer Engineering at Temple University, Philadelphia. He received Ph.D. in Electrical Engineering from the University of Ottawa, Canada, in 1985. His current research interests focus on multiagent systems, intelligent control, optimal and nonlinear control, and power systems. Dr. Biswas is a member of IEEE and Sigma Xi.

Jie Wu is the chair and a professor in the Department of Computer and Information Sciences, Temple University. Prior to joining Temple University, he was a program director at National Science Foundation. His research interests include wireless networks and mobile computing, routing protocols, fault-tolerant computing, and interconnection networks. Dr. Wu is a Fellow of the IEEE.