2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

Role of Stochasticity in Self-Organization of Robotic Swarms

Manish Kumar, Dejan Milutinović and Devendra P. Garg

Abstract— **This paper investigates the effectiveness of designed random behavior in self-organization of swarm of robotic agents. Inspired by the self-organization observed in biological cells and the role played by random forces in providing robustness in cell self-organization, we investigate the possibility of designing a decentralized controller for a swarm of agents in which the stochastic process is included. This paper considers flocking as a self-organizing behavior example to validate our findings. The controller is designed in the framework of Lyapunov function, and it is based on the artificial potential due to interactions among agents. Our analysis shows that the flocking behavior of the swarm is improved and is more robust when the stochastic process is included in the agent controller.**

I. INTRODUCTION

A swarm of robotic agents performing coherent activities while controlled in a decentralized manner is an example of a self-organized system [12]. There are several examples in nature where individual units carry out operations based on local interaction and local information without a complete knowledge of other units' operation. Yet the overall emerging behavior of the system appears to be highly organized, coherent, and efficient in achievement of its objectives. The best examples of such systems are populations of biological cells that possess the ability to self-organize into specific formations, form different types of organs and, ultimately, develop into a living organism. Most importantly, their ability to self-organize is extraordinarily robust.

The cell behavior is guided by biochemical signals and the structure of the environment. Intracellular biochemical signaling networks are involved in the detection of the environment and they drive the cell behavior, their function and motility. The role of signaling networks for cells is similar to the one that the robot hardware plays for the algorithms that guide the robot communication and sensory systems, as well as the robot behavior [10]. Because of this, the study of self-organized cellular systems is more relevant for the design of robotic swarms as compared to the study of biological multi-agent systems, such as flocks of birds or schools of fish, in which the presence of natural intelligence among biological entities cannot be ruled out. In this respect, the design of agent swarms inspired by a self-organized cellular system is a very promising direction of research.

This research was performed while the first author, Dr. Manish Kumar, held a National Research Council Associateship award through the Army Research Office. He is currently with the Dept. of Mech., Indu., and Nuclear Engg. at the Univ. of Cincinnati, OH. manish.kumar@uc.edu

Dejan Milutinović is with the Laboratory for Computational Immunology at Duke University, NC dejan.m@duke.edu

Dr. Devendra P. Garg is with the Dept. of Mech. Engg. and Matls. Sci. at Duke Univ., NC dpgarg@duke.edu

This paper investigates the role random forces can play in attaining robust behaviors in swarm systems. The paper draws inspiration from the role of randomness in several search algorithms and in several biological systems. Randomness has been widely used as a component in several stochastic search algorithms. For example, evolutionary search algorithms [5] have heavy stochastic components (e.g. choice of chromosomes for reproduction, crossover, and mutation are driven by probabilities based on their fitness function). Reinforcement learning techniques [18] have two components: exploitation and exploration. Exploration is a random walk in the search space that makes the algorithm investigate new regions. Apart from randomness being traditionally used in search and optimization algorithms, random components have shown to play critical role in modeling many biological systems. Bateson [3] calls the mind a stochastic system and cognitive learning process a stochastic process. Contemporary cognitive scientists consider mental processes as stochastic processes such as evolutionary algorithms where hypotheses or ideas are proposed, tested, and either accepted or rejected by a population. Random or trial-and-error learning techniques provide ways to create new varieties of solutions for problems. Random behavior is ubiquitous in biological systems. Chaotic behavior of a hooked fish, random behavior among preys for predator avoidance, and zig-zagging of a chased rabbit through a meadow are all examples of existence and heavy use of random behaviors among animals. Lorenz [9], in his intuitive chapter entitled "Oscillation and Fluctuation as Cognitive Functions", has described the importance of a random behavior in organisms' motion for search, as well as for escaping dangers. We use this kind of random behavior as a component in the control laws for swarm robotic systems.

In this paper, we consider flocking which is the most investigated example of swarm self-organization. *Flocking* is a self-organized behavior in which agents, initially distributed over the operating space, group together and organize into a specific formation. An example of this kind of flocking is the formation of germinal centers inside the lymph node [16]. Flocking has been extensively studied in multi-agent literature. Reynolds [15] has been able to reproduce, in his computer models, behaviors representing flocking in birds and schooling in fish using simple rules based on local interactions among agents. Drawing inspiration from Reynolds' approach, many researchers have focused on designing a decentralized controller for achieving flocking behavior [2], [4], [13], [19]. A control system based on the methods presented in these references should be able to yield a single flock of agents based on local information. However, if

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

the local information can only be collected over a finite range, it can lead to the formation of more than one flock. This paper studies scenarios in which local interaction leads to the fragmentation of groups and investigates how the introduction of random processes in an agent controller helps to eliminate or alleviate the problem of fragmentation.

In this paper, we consider flocking in the Lyapunov function framework. The Lyapunov function for the controller design of a multi-agent system is based on the artificial potential function of agent interactions [8], [14]. Being inspired by the role played by the random fluctuation in driving self-organizing processes in cells, we propose to include a random process in the decentralized controller for an agent swarm. Consequently, we analyze the swarm self-organization using the stochastic Lyapunov function [7]. We show that our proposed decentralized controller provides the robust flocking of the swarm of agents.

This paper is organized as follows. We introduce a model of swarming agents in section II. In section III, we describe a deterministic decentralized design of controllers providing flocking in a population of agents. Dealing with the problem of local minima, we suggest including a stochastic process into the controller in section IV. In section V, we derive emergent behavior related to the center of the mass of the flock. Section VI presents numerical simulations illustrating the performance of the controller, and finally section VII presents the conclusions derived from the present work.

II. Modeling of swarm system and Lyapunov function definition

The group of mobile agents consists of N fully actuated agents, each of whose dynamics is given by the double integrator:

$$\begin{aligned} \dot{q}_i &= p_i \\ \dot{p}_i &= u_i(t) \qquad i = 1, 2, \ldots, N \end{aligned} \tag{1}$$

where q_i, p_i and u_i are m-dimensional position, velocity and control vectors of agent i, respectively. This double integrator representing particle dynamics [14], [8], [20] is a popular and realistic model to represent the motion of agents in a multi-agent system. It facilitates the implementation of decentralized control algorithms, and provides a mechanism to include limitations due to sensing and communication as compared to other models, such as continuum model [6], [11], which represents the collective motion of agents in the form of particle density functions. The model given in equation (1) can be written as:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \tag{2}$$

where vector $\mathbf{x}(t) = [\mathbf{q}(t)\ \mathbf{p}(t)]^T$ and

$$\mathbf{q}(t) = \begin{bmatrix} q_1 \\ \vdots \\ q_N \end{bmatrix},\ \mathbf{p}(t) = \begin{bmatrix} p_1 \\ \vdots \\ p_N \end{bmatrix},\ \mathbf{u}(t) = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} \tag{3}$$

and matrices $\mathbf{A}$ and $\mathbf{B}$ are

$$\mathbf{A}(\mathbf{x}, t) = [\mathbf{I}_{N\times N}\ \mathbf{0}_{N\times N}]^T,\ \mathbf{B} = [\mathbf{0}_{N\times N}\ \mathbf{I}_{N\times N}]^T. \tag{4}$$

In order to carry out a stability analysis of the collective motion of agents, a Lyapunov function can be chosen as the total energy

$$\phi(\mathbf{q}, \mathbf{p}) = V(\mathbf{q}) + \frac{1}{2}\mathbf{p}^T\mathbf{p} \tag{5}$$

The Lyapunov function is composed of the total artificial potential energy $V(\mathbf{q})$ and the kinetic energy, the second term of the sum given in equation (5). We define the potential energy as a non-negative function

$$V(\mathbf{q}) = \frac{1}{2}\sum_{i=1}^{N} V_i = \frac{1}{2}\sum_{i=1}^{N}\sum_{j\in N_i} V_{ij}(\|q_i - q_j\|) \tag{6}$$

$$V_{ij}(\|q_i - q_j\|) \geq 0 \tag{7}$$

where V_i is the total artificial potential associated with an agent i. This energy depends on the set N_i comprised of the agents inside the communication range of the agent i. The artificial potential function V_{ij} depends upon the Eucledian distance $\|q_i - q_j\|$ between the agent i and j, and it is given by:

$$V_{ij} = \begin{cases} a(ln(\|q_i - q_j\|) + \frac{d_0}{\|q_i - q_j\|}), & 0 \leq \|q_i - q_j\| \leq d_1 \\ a(ln(d_1) + \frac{d_0}{d_1}), & \|q_i - q_j\| > d_1 \end{cases} \tag{8}$$

where, a is a scalar parameter. The parameters d_0 and d_1 respectively represent the inter-agent distance below which the interaction force is repulsive (negative) and above which the interaction force is zero. Figure 1 shows the interaction potential plotted against the inter-agent distance. It can be easily seen that the potential becomes minimal when the inter-agent distance is d_0. The interaction among agents happens with the help of sensing or communication devices. The parameter d_1, then, can be regarded as a sensing or a communication range.

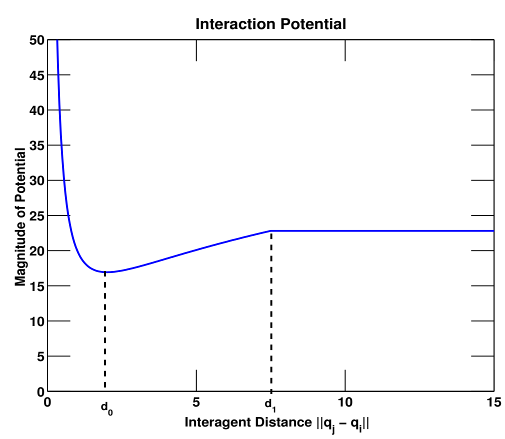

Fig. 1. Interaction potential versus inter-agent distance

III. Controller Design for Flocking

Flocking is a form of self-organized behavior of agent swarms in which agents meet or come together. The collective dynamics of the system can be analyzed using a

Lyapunov function. We can differentiate $\phi(\mathbf{q},\mathbf{p})$ with respect to time, and using expression (5) and model (1), one gets:

$$\begin{aligned}\dot{\phi}(\mathbf{q},\mathbf{p}) &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\dot{\mathbf{p}} \\ &= \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\mathbf{u}(t)\end{aligned}$$

Using arguments based on the Lyapunov function stability analysis, the state configuration $\mathbf{p}$, $\mathbf{q}$ will be stable if

$$\dot{\phi}(\mathbf{q},\mathbf{p}) = \mathbf{p}^T\nabla V(\mathbf{q}) + \mathbf{p}^T\mathbf{u}(t) \leq 0 \tag{9}$$

The control law ensuring this type of stability is described by the following lemma of which a detailed proof is provided in references [14] and [8].

Lemma 3.1: Consider a system of N mobile agents. Each of the agents follows dynamics given by model (1) and with the feedback control law given by

$$u_i = -\nabla_{q_i} V_i + f_i^v \tag{10}$$

where ∇_{q_i} represents the gradient with respect to position q_i of agent i, $c > 0$ is a scalar gain and f_i^v is given by:

$$f_i^v = c\sum_{j\in N_i}(p_j - p_i) \tag{11}$$

For any initial condition belonging to the level set of $\phi(\mathbf{q},\mathbf{p})$ given by $\Omega_C = \{(\mathbf{q},\mathbf{p}) : \phi(\mathbf{q},\mathbf{p}) \leq C)\}$ with $C > 0$, and when the underlying graph of the system is connected and cohesive, the system asymptotically converges to an invariant set $\Omega_I \subset \Omega_C$ such that the points in Ω_I have a velocity that is bounded and the velocities of all agents match.

To illustrate this lemma, it is worth mentioning that the control law (10) results from the fact that it can be equivalently described in the vector form as

$$\mathbf{u}(t) = -\nabla V(\mathbf{q}) - \hat{\mathbf{L}}(\mathbf{q})\mathbf{p} \tag{12}$$

where $\hat{L}(\mathbf{q}) \in R^{mN\times mN}$ is m-dimensional graph Laplacian (see reference [14]), which is a positive semi-definite matrix. Obviously, this control satisfies condition (9). Also, from Lasalle's Invariance Principle, all solutions of the system starting in Ω_C will converge to the largest invariant set $\Omega_I = \left\{(\mathbf{q},\mathbf{p}) \in \Omega_C : \dot{\phi}(\mathbf{q},\mathbf{p}) = 0)\right\}$, and this occurs when the velocities of all agents match.

IV. Controller design based on the stochastic Lyapunov function

Using the deterministic Lyapunov function controller design, agents always move in such a way that the covered distances and their directions do not increase the Lyapunov function. It is ultimately expected that the agents reach a stable formation (flock) in which the Lyapunov function attains its extremum value. If the agents that we are dealing with have a limited communication range, it is possible that robots reach stable formations with more than one cluster. In this configuration, agents from one cluster are out of the communication range of any agent from the other clusters. The graph Laplacian $\hat{L}(\mathbf{q})$ is no longer positive semi-definite in this case, and a local minimum of the Lyapunov function is reached.

Here, we introduce a controller which includes the random process term, providing means to an escape from local minima. Using the notation of section II, the model of a robot population can be written in the matrix form as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{\Sigma}\xi(t) \tag{13}$$

where

$$\mathbf{\Sigma} = \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & diag(\sigma_1,\sigma_2,\ldots\sigma_N)\end{array}\right]_{2N\times 2N} \tag{14}$$

and the controller effect is composed of the part which is assumed to be computed based on the local agent information $\mathbf{Bu}(t)$ and the stochastic part $\mathbf{\Sigma}\xi(t)$.

Using the Lyapunov function as before and using the Îto formula, we can find the time derivative of the Lyapunov function

$$\dot{\phi} = \sum_i p_i\frac{\partial\phi}{\partial q_i} + u_i(t)\frac{\partial\phi}{\partial p_i} + \frac{1}{2}\sum_i\sigma_i^2\frac{\partial^2\phi}{\partial p_i^2} + \sum_i\sigma_i\frac{\partial\phi}{\partial p_i}\xi_i(t) \tag{15}$$

that results in

$$\begin{aligned}\dot{\phi} = \quad & \textstyle\sum_i p_i\frac{\partial V_i(q_1,q_2,\ldots q_N)}{\partial q_i} + \sum_i u_i(t)p_i + \\ & \textstyle\frac{1}{2}\sum_i\sigma_i^2 + \sum_i\sigma_i p_i\xi_i(t)\end{aligned} \tag{16}$$

The latter expression means that in this case the Lyapunov function ϕ of the robotic population is a stochastic process.

Similar to the deterministic case, if we would like to design a controller which aligns the robot velocities, i.e., provides flocking, we can define u_i as

$$u_i = -\frac{\partial V_i(q_1,q_2,\ldots q_N)}{\partial q_i} - [\hat{L}(\mathbf{q})\mathbf{p}]_i \tag{17}$$

Under this condition

$$\dot{\phi} = -\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p} + \frac{1}{2}\sum_i\sigma_i^2 + \sum_i\sigma_i p_i\xi_i(t) \tag{18}$$

which means that the total Lyapunov function value is a stochastic process. The stochasticity provides escape from the local minima. We assume, without losing generality, that $\sigma_i = \sigma$. The intensity of this stochastic process is governed by the parameter σ which needs to be determined by taking appropriate considerations of factors explained below.

There is a stochastic steady-state (see reference [7], page 50, Theorem 6) for the value of ϕ in which the following condition is satisfied:

$$E\left\{\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p}\right\} = \frac{1}{2}\sum_i\sigma_i^2 = \frac{N\sigma^2}{2} \tag{19}$$

There are two limits for σ that should be avoided. One is $\sigma < \sigma_L$ that results in the deterministic controller. In this case the swarm does not flock robustly. The second limit is $\sigma > \sigma_H$, when σ is large and leads to a large expected value of $\mathbf{p}^T\hat{L}(\mathbf{q})\mathbf{p}$. This means that the robot velocities are poorly aligned. For reasons of the robust flocking, σ should be in the range of σ_L and σ_H. Out of this range, the flocking does not happen robustly due to the small ($\sigma < \sigma_L$) or large

($\sigma > \sigma_H$) intensity of the random process, respectively. The value of σ has to be tuned taking into account the constraints of the actuator that drives the robots and it is a part of the controller design. Introduced controller does not guarantee the attainment of global minimum of the Lyapunov function, but it can guarantee that the formation of the system is, most of the time, around this minimum.

V. Emergent Behavior of a Stochastic Control for Flocking

The cluster is a group of robots in which each robot is in the communication range of at least one robot. Motion of each agent i obeys the following stochastic differential equation

$$\begin{aligned} \dot{q}_i &= p_i && (20) \\ \dot{p}_i &= u_i(t) + \sigma_i \xi_i(t) \ . && (21) \end{aligned}$$

This expression includes two parts: the first part based on the local information and the second part based on the stochastic process of intensity σ_i. We assume again that ξ is the Gaussian white noise of the unit intensity. For reasons of simplicity, but without losing generality, we will consider that $\sigma_i = \sigma$.

Since each robot has a unique label i, the cluster C_j can be defined as the set of labels i of the robots being in the cluster j. Assuming that the cluster exists within the time interval $\tau = [t_1 \ t_2]$, the motion of the C_j cluster's center of the mass is described by

$$\begin{aligned} \dot{q}_j^{CM} &= p_j^{CM} && (22) \\ \dot{p}_j^{CM} &= \frac{1}{|C_j|} \sum_{i \in C_j} \sigma_i \xi_i(t) = \frac{\sigma}{\sqrt{|C_j|}} \xi(t) && (23) \end{aligned}$$

where the number of the robots in the cluster is denoted by $|C_j|$. The terms dependent on u_i are cancelled due to the symmetry in the interaction between any two robots in the cluster. The sum of white Gaussian noises is also the white Gaussian noise and the last term in (23) includes this properly scaled unit intensity Gaussian noise.

From (23), we can conclude that the center of the cluster mass moves randomly through the operating space. The smaller the cluster, the faster it "explores" the operating space before it meets other robots to form a larger cluster. We quote the word "explore", because it is a behavior that emerges from the inclusion of the random process in the robot controller.

Obviously, while exploring, the cluster may decompose into two smaller clusters. The probability of cluster decomposition can be made very small by appropriately choosing the intensity σ to be small in comparison to the intensity of u_i. Regardless of this, the cluster decomposition does not influence our further analysis, because we analyze the steady-state property of the robot formation independently of how it is reached. In the limit when $|C_j|$ is large, the center of the cluster mass will move with $q_j^{CM} \approx 0$.

VI. Simulation Results

Before we present the simulation results, let us explain *social* and *hierarchical social* entropy metrics we use to quantify the degree of the self-organization of agent populations. Social entropy, inspired by Shannon's information entropy [17], has been used in multi-agent systems [1] as a metric for diversity in behavior or properties of agents, including diversity in their spatial locations. This metric captures an important feature of diversity, which is the number of differentiable groups, i.e., clusters, in a system and the size of these clusters.

We define the *cluster* as a group of agents in which the distance between the agent and its closest neighbor from the same cluster is smaller than some threshold value h, which is the communication range distance. If an agent does not have the neighbor within a distance smaller that h, then this agent is alone in the cluster.

If a group of n agents, represented by the set R, can be divided into m clusters $C_1, C_2 \ldots C_m$, such that there is no agent which is simultaneously in two clusters, then the social entropy H of the system R is given by

$$H(R) = -\sum_{i=1}^{m} \frac{|C_i|}{|R|} \log \frac{|C_i|}{|R|} \qquad (24)$$

with $|.|$ representing the number of agents in the cluster and $\sum_i |C_i| = |R|$, where $|R| = n$ is the number of robots in the group.

The social entropy H obviously depends on the threshold h. In order to develop a measure that can accurately represent the diversity in a population of agents regardless of the scale or the value h, *hierarchical social entropy* has been used in literature [1], which is given by:

$$E(R) = \int_{h=0}^{\infty} H(R,h)dh \qquad (25)$$

where $H(R,h)$ is the social entropy given by equation (24).

To illustrate flocking in the absence and the presence of the random force, we simulate a group of 20 agents during a 500 sec. time span. In our simulations, the agents have a limited communication range, which is modeled with the artificial potential function parameter $d_1 = 20m$ (see equation (8)). The other parameter of this function is $d_0 = 6m$, and the intensity of the random process $\sigma = 0.25$. In the simulation scenario, the agents are allowed to move only in the 100×100 rectangular area (see Fig. 2). To define this area inside the simulation, we introduce an additional artificial potential function resulting in short-range repulsive forces making agents move away from the area border.

Figure 2 shows the agent configurations at different points of time of the simulation when random force in the controller is not applied. The agents start from an initial randomly chosen position. By the terminal time of the simulation $T = 500$ sec, the agents form three separate clusters, which are the consequence of the limited communication range. In the next figure, Fig. 3, we illustrate the simulation in which the agents start from the same configuration as in the previous example, but now the random force inside the controller is applied. We note that by the terminal simulation time the agents form a single cluster.

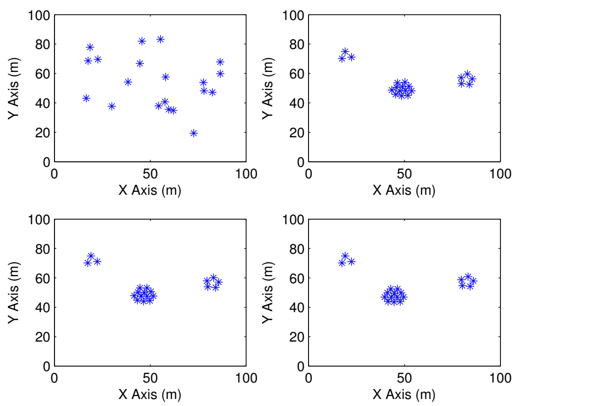

Fig. 2. Configurations of agents at times T=0 (top left), T=167 sec (top right), T=333 sec (bottom left) and T=500 sec (bottom right) when no random motion is applied

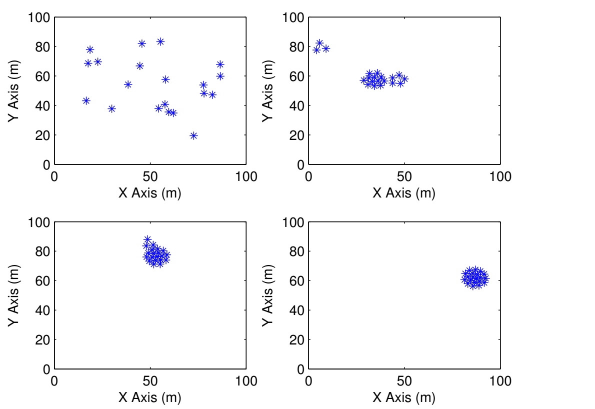

Fig. 3. Configurations of agents at times T=0 (top left), T=167 sec (top right), T=333 sec (bottom left) and T=500 sec (bottom right) when random motion is applied

The hierarchical social entropy, given by equation (25), for these two examples is plotted against the simulation time in Fig. 4. From this figure, it is evident that the application of random force results in lower entropy values, meaning that agents aggregate closer to one another forming bigger clusters.

In order to verify that the introduction of random motion does lead to better flocking behavior of agents, we carried out two sets of 100 simulation runs. Each run was carried out under the same conditions as in the two presented examples. In one simulation set, the random force term of the controller is applied and in the other, it is not. For each simulation run we computed the social entropy at the terminal time $T = 500$ sec. The distribution of the computed social entropy values, for each set of simulations, is presented in the form of the cumulative distribution function shown in Fig. 5. It can be seen that the increase in the number of

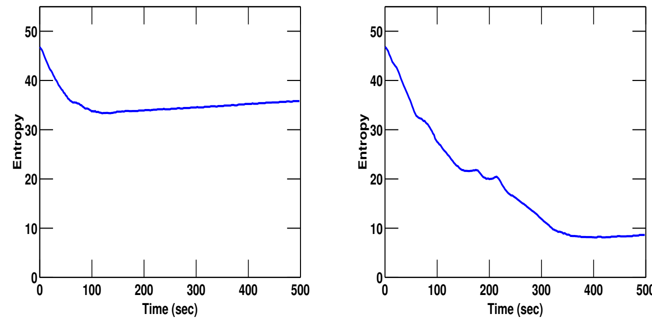

Fig. 4. Hierarchical social entropy of the population of agents for the case when no random motion is applied (left), and the case when random motion is applied (right)

simulation occurrences for the case when random motion is not applied is slower than that of the case when random motion is applied. This is because of the fact that most of the simulations with random motion have an entropy which is smaller as compared to simulations with no random motion. A smaller entropy indicates a smaller (spatial) diversity, i.e., better flocking behavior.

The ultimate evidence that the application of the random force term leads to a better flocking behavior is the histogram of number of simulations with specific number of clusters in the final configuration for the two cases, as shown in Fig. 6. As can be seen from the figure, a lesser number of clusters are formed for that case when random motion is applied as compared to the case when random motion is not applied. The average number of clusters for the case when no random motion is applied is 3.02, while the average number of clusters when random motion is applied is 1.51. This shows a marked improvement in flocking behavior and the formation of one giant cluster as compared to the case when no random motion is applied. It may be noted that the separate clusters formed in the second case move randomly in a confined space. This random motion within a confined space results into the probability of clusters finding each other approaching unity when the time approaches infinity.

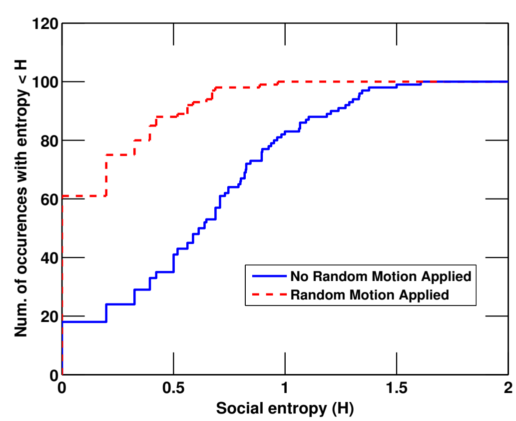

Fig. 5. Simple social entropy at final configuration of agents for 100 simulation runs

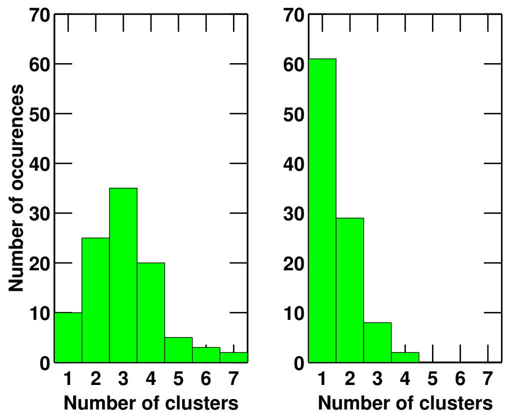

Fig. 6. Number of clusters of agents at final configuration for 100 simulation runs when no random motion is applied (left) and when random motion is applied (right)

VII. CONCLUSION

Inspired by the role of random forces in robustness of self-organization of cells and other biological systems, we investigated the possibility of designing a decentralized controller for a swarm of agents in which a stochastic process is included. We considered the flocking behavior of a swarm and described the previously considered deterministic Lyapunov function controller design based on the artificial potential of interactions among agents. The deterministic design may not lead to desired swarm behaviors because of the existence of agent configurations in which the total potential of swarm interactions has local minima.

The design we considered is an extension of the deterministic one. We used the same Lyapunov function. However, due to the introduced stochastic process, the Lyapunov function is also a stochastic process. Consequently, this controller provides a means of escaping from the local minima of the total potential of the swarm interactions.

The intensity of the included stochastic process is of great significance. If its intensity is small, then the swarm behavior is the same as if controlled by the deterministic controller. If the intensity is high, then the flocking behavior cannot be established because the swarm can even escape from the region around the global minimum. Only if the intensity of the stochastic process lies within some intermediate range, does the swarm escape the local minima and randomly explores configurations, which can ultimately lead the swarm towards configurations close to the global minimum of the total potential of interactions. The drawback of this design is that even in the global minimum configuration, the stochastic process forces the swarm to search for a better configuration. Tuning the intensity of the stochastic process of the controller is identical to consideration that appears in the design of the realistic model of cell behavior, or to the problem of providing physiological conditions for self-organization in biological cells.

The performance of the proposed controller was illustrated by the derivation of the emergent swarm behavior and an extensive simulation study.The performance of the controller with the random process term was compared to the performance of the deterministic controller. The simulation results suggest that the inclusion of the random process in the controller can certainly improve the performance of the robotic swarm in achieving the flocking behavior.

REFERENCES

[1] T. Balch. Hierarchic social entropy: An information theoretic measure of robot group diversity. *Autonomous Robots*, 8(3):209–237, 2000.
[2] T. Balch and R. Arkin. Behavior-based formation control for multirobot systems. *IEEE Transactions on Robotics and Automation*, 14(6):926–939, 1998.
[3] G. Bateson. *Mind and Nature*. New York: E. P. Dutton, 1979.
[4] J. P. Desai, J. P. Ostrowski, and V. Kumar. Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation*, 17(6):905–908, 2001.
[5] D. Goldberg. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley Longman, Inc., 1989.
[6] E. W. Justh and P. S. Krishnaprasad. Steering laws and continuum models for planar formations. In *IEEE International Conference on Decision and Control*, pages 3609–3614, 2003.
[7] H. J. Kushner. *Stochastic Stability and Control*. Academic Press, New York, 1967.
[8] N. Leonard and E. Fiorelli. Virtual leaders, artificial potentials and coordinated control of groups. In *IEEE International Conference on Decision and Control*, pages 2968–2973, 2001.
[9] K. Lorenz. *Behind the Mirror: A Search for a Natural History of Human Knowledge*. New York: Harcourt Brace Jovanovich, 1973.
[10] D. Milutinović and P. Lima. *Cells and Robots : Modeling and Control of Large-Size Agent Populations*. Springer, 2007.
[11] D. Milutinović, P. Lima, and M. Athans. Biologically inspired stochastic hybrid control of multi-robot systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, 2003.
[12] G. Nicolis and I. Prigogine. *Self-organization in non-equilibrium systems*. Wiley, 1977.
[13] P. Ogren, M. Egerstedt, and X. Hu. A control Lyapunov function approach to multi-agent coordination. *IEEE Transactions on Robotics and Automation*, 18(5):847–851, 2002.
[14] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, Mar. 2006.
[15] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics (SIGGRAPH '87 Conference Proceedings)*, 21(4):25–34, 1987.
[16] T. A. Schwickert, R. L. Lindquist, G. Shakhar, G. Livshits, D. Skokos, M. H. Kosco-Vilbois, M. L. Dustin, and M. C. Nussenzweig. In vivo imaging of germinal centres reaveals a dynamic open structure. *Nature*, 446:83–87, 2007.
[17] C. E. Shannon and W. Weaver. *A Mathematical Theory of Communication*. University of Illinois Press, Urbana, IL, 1949.
[18] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction, MIT Press*. MIT Press, 1998.
[19] H. Tanner, A. Jadbabaie, and G. J. Pappas. Stable flocking of mobile agents, part i: Fixed topology. In *IEEE International Conference on Decision and Control*, pages 2010–2015, 2003.
[20] H. G. Tanner, A. Jadbabaie, and G. J. Pappas. Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control*, 52(5):863–868, May 2007.