Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Robustness analysis and failure recovery of a bio-inspired self-organizing multi-robot system

Yaochu Jin, Honglian Guo, Yan Meng

2009

Preprint:

This is an accepted article published in Third IEEE International Conference on Self-Adaptive and Self-organizing Systems. The final authenticated version is available online at: https://doi.org/[DOI not available]

Robustness Analysis and Failure Recovery of A Bio-Inspired Self-Organizing Multi-Robot System

Yaochu Jin
Honda Research Institute Europe
Carl-Legien-Str. 30
63073 Offenbach, Germany
Email: yaochu.jin@honda-ri.de

Hongliang Guo
Dept. of Electrical and Computer Eng.
Stevens Institute of Technology
Hoboken, NJ 07030
Email: hguo@stevens.edu

Yan Meng
Dept. of Electrical and Computer Eng.
Stevens Institute of Technology
Hoboken, NJ 07030
Email: yan.meng@stevens.edu

Abstract

Biological systems can generate robust and complex behaviors through limited local interactions in the presence of large amount of uncertainties. Inspired by biological organisms, we have proposed a gene regulatory network (GRN) based algorithm for self-organizing multiple robots into different shapes. The self-organization process is optimized using a genetic algorithm. This paper focuses on the empirical analysis of robustness of the self-organizing multi-robot system to the changes in tasks, noise in the robot system and changes in the environment. We investigate the performance variation when the system is optimized for one shape and then employed for a new shape. The influence of noise in sensors for distance detection and self-localization on the final positioning error is also examined. In case of a complete self-localization failure, we introduce a recovery algorithm based on trilateration combined with a Kalman filter. Finally, we study the system's performance when the number of robots changes and when there are moving obstacles in the field. Various simulation results demonstrate that the proposed algorithm is efficient in shape formation and that the self-organizing system is robust to sensory noise, partial system failures and environmental changes.

1. Introduction

In a multi-robot system (MRS), robots are supposed to work together efficiently and reliably to accomplish tasks that are intrinsically distributed in space, time, or functionality. However, these properties are often achieved at the cost of increasing system complexity, especially in control and communication [17]. It is usually not hard to implement a centralized controller for multi-robot systems that can perform complex tasks in a well-defined environment [16]. Unfortunately, multi-robot systems with a centralized controller will often perform rather poorly under unknown dynamic environments. Therefore, distributed methods are more attractive for multi-robot systems due to their robustness, flexibility, and adaptivity.

Designing a distributed, self-adaptive multi-robot system is however nontrivial. As indicated by Nolfi and Floreano [23], it is tedious to predict an emerging behavior resulting from a given set of local interaction rules in a distributed control system. It is even more intractable to identify the rules behind an observed global behavior.

Biological organisms have evolved to perform and survive in a world characterized by rapid changes, abundant uncertainties, and limited information [24]. For this reason, researchers have recently turned to biological systems for distributed control principles. For example, swarm intelligence, now an efficient, distributed computational methodology for solving complex problems, was inspired from behaviors of social insects. Among other applications, swarm intelligence has shown to be a successful approach to solving MRS problems [6], [12], [21], [22], [25], [31].

Distributed self-organization control has also widely observed in the embryonic development of all biological organisms. With the rapid advances in systems biology [1], it has been found that the embryonic developmental process is largely under the control of the gene regulatory networks (GRNs), which are a number of genes that interact with each other and with the environment. It appears, however, that not much work has been conducted on multi-robot systems using GRNs or cellular mechanisms until recently, though research on pattern formation in biological development has relatively a long history in mathematical biology [9], [14], [19]. A cell-based approach to self-assembly of arbitrary 2D shapes is proposed using multi-agents [2], where each agent can be represented as a cell, which can produce gradient information that can be perceived by their neighbors. Reference points must be selected among the agents so that the whole system can grow along them to formulate a shape. Shen et al. [26] proposed a digital hormone model (DHM) as a bio-inspired distributed control method for robot swarms and self-organization. Essentially, they applied Turing's reaction-diffusion model [30] to describe the interactions between the hormones. The DHM integrated dynamic network, topolog-

ical stochastic action selection, and distributed control by hormone reaction-diffusion. Taylor [29] suggested a gene regulatory network inspired real-time controller for a group of underwater robots. A genetic algorithm (GA) was applied to evolve the controller for a simple clustering task. Mamei et al. [20] proposed a decentralized morphogen gradient based approach to swarm robot pattern generation, where robots can communicate with their neighbors to receive and propagate morphogen gradients and estimate the local density of robots. A decentralized approach to formation control for swarms of mobile agents was proposed in [3], where the predefined shape information is superimposed on the agents coordinate systems using a container, and then a gas expansion model is employed to disperse the robots within the container. Most recently, a gene regulation based algorithm for multi-robot construction has been suggested in [10], where the shape information is embedded into the regulatory dynamics. It has been shown with a theoretic proof that robots will converge to the target shape asymptotically.

The purpose of this paper is to investigate in more depth the property of the self-organization algorithm suggested in [10]. We focus on analyzing the robustness of the system to changes in target shape, number of robots in the system, obstacles in the environment, and sensory noises or even failure in self-localization. These analysis are important for understanding the dynamics of the system and for practical implementation on real robots.

The paper is organized as follows. Section 2 introduces briefly existing approaches to computational modeling of GRNs. The GRN based method for self-organizing multi-robot systems is presented in Section 3. To evaluate the robustness of the proposed method, we conduct a variety of case studies on the GRN based self-organizing system in Section 4, where we also introduce a trilateration based algorithm combined with a Kalman filter to recover the position of robots that completely fail in self-localization. Conclusions of the paper are summarized in Section 5.

2. Computational Modeling of Gene Regulation

As we mentioned in Section 1, the process of individual development is under the control of gene regulatory networks. When a gene is expressed, information stored in an organism's genome is transcribed into mRNA and translated into proteins [1]. Some of these proteins are transcription factors that can regulate the expression of their own genes or others. Understanding cellular organization and general biological principles such as robustness and evolvability have become one of the most challenging yet fascinating research topics in systems biology [13], [15], [18]. To this end, a large number of computational models for GRNs have been suggested [5], [7], [11], [28]. Among others, ordinary or partial differential equations have been used to model the reaction kinetics of regulatory systems. For example, the expression of a gene with negative auto-regulation in a uni-cellular organism can be described by the following differential equations:

$$\frac{dg_i}{dt} = -\gamma_g g_i + \alpha_g f(p_i) \tag{1}$$

$$\frac{dp_i}{dt} = -\gamma_p p_i + \alpha_P g_i \tag{2}$$

where g_i is the expression level (measured by the concentration of its mRNA product) of gene i and p_i is the concentration of protein i. γ_g and γ_P are the decay rate of mRNA and protein concentration, respectively. α_g and α_p are the synthesis rate of mRNA and protein concentration, respectively. $f(x)$ is a sigmoid function, which can be defined as:

$$f(x) = \frac{\beta}{\theta^n + x^n} \tag{3}$$

where $\beta$is the maximum expression level, θ is the activation coefficient, and n is known as the Hill coefficient.

In a multi-cell organism, the expression of the genes in a cells can also be influenced by the gene product (transcription factors) of other cells. Thus, it is necessary to model the intercellular communication. In addition to the internal dynamics of the cell, we should also include external factors such as protein gradients and physical interactions between cells into the GRNs model. Salazar-Ciudad et al. [27] proposed a GRN model with reaction-diffusion mechanism as follows:

$$\frac{dx_{ij}}{dt} = f_j(\mathbf{x}_i, \mathbf{u}) - \gamma_i x_{ij} + D_j \nabla^2 x_{ij},\ 1 \le i \le n, 1 \le j \le m \tag{4}$$

where x_{ij} is the concentration of gene product j in cell i. The first term specifies the production of x_{ij}, the second term is its degradation, and the last term specifies the diffusion component at diffusion rate D_j. f_j is a nonlinear update function of gene product j, which is usually defined as a sigmoid function as $f(x) = \frac{1}{1+e^x}$. $\mathbf{u}$ is the vector of external input signals. γ_i is the degradation rate of product i. n is the number of gene products, and m is the number of cells.

3. GRN Based Self-Organization of Multi-Robots

The objective of the multi-robot construction is to deploy multiple robots uniformly on a predefined two-dimensional shape, for example: a circle or a square, through a distributed

1. Most recent findings in molecular biology suggest that gene expression is also subject to epigenetic regulations such as DNA methylation, histone modification and RNA silencing.

control approach. Each robot only knows its local information without any global observer.

In our GRN inspired control model, we assume that each robot corresponds to a single cell. Within each cell's genome, there are two genes, one for x-position and one for y-position in a 2D environment. Each gene can produce a protein. Each protein can provide the following three functions. First, it can regulate the expression of the gene that produces the protein (i.e. auto regulation), thus adjusting the robots' behavior. Second, it is can interact with a certain 'morphogen gradient' in the environment, which contains information on the target shape. Third, it can diffuse into environment, which can thus be perceived by its neighboring robots to prevent collision of the robots.

Inspired by equations (1)-(4), the system dynamics of the GRN for multi-robot construction are defined as follows:

$$\begin{array}{l} \frac{dg_{i,x}}{dt} = -a\mathbf{z}_{i,x} + mp_{i,x} \\ \frac{dg_{i,y}}{dt} = -a\mathbf{z}_{i,y} + mp_{i,y} \end{array} \tag{5}$$

$$\begin{array}{l} \frac{dp_{i,x}}{dt} = -cp_{i,x} + kf(\mathbf{z}_{i,x}) + bD_{i,x} \\ \frac{dp_{i,y}}{dt} = -cp_{i,y} + kf(\mathbf{z}_{i,y}) + bD_{i,y} \end{array} \tag{6}$$

where $g_{i,x}$and $g_{i,y}$ are the expression levels of the i^{th} robot's gene for x-position and y-position, respectively. $p_{i,x}$ and $p_{i,y}$ are the concentration of the i^{th} robot's proteins for x-position gene and y-position gene, respectively.

In order to embed the 2D target shape, which is global information, into the regulatory dynamics, we define $f(\mathbf{z}_i)$ to be the following sigmoid functions:

$$\begin{array}{l} f(\mathbf{z}_{i,x}) = \frac{1-e^{-\mathbf{z}_{i,x}}}{1+e^{-\mathbf{z}_{i,x}}} \\ f(\mathbf{z}_{i,y}) = \frac{1-e^{-\mathbf{z}_{i,y}}}{1+e^{-\mathbf{z}_{i,y}}}, \end{array} \tag{7}$$

where $\mathbf{z}_{i,x}$ and $\mathbf{z}_{i,y}$ are the gradients along x-axis and y-axis, respectively, of a predefined function h at the robot's current expression level, which are defined as:

$$\mathbf{z}_{i,x} = \frac{\partial h}{\partial g_{i,x}},\ \mathbf{z}_{i,y} = \frac{\partial h}{\partial g_{i,y}}, \tag{8}$$

where the function h defines the target shape on which the robots are supposed to be deployed uniformly. The predefined function h can be seen as the morphogen gradients for cell sorting in the early development of biological organisms. To facilitate the generation of the desired dynamics, we defined h to be the square of the target shape function. For example, if we want to deploy the robots onto a unit circle, the shape function can be defined as:

$$s(g_{i,x}, g_{i,y}) = g_{i,x}^2 + g_{i,y}^2 - 1 = 0 \tag{9}$$

Then function h can be defined as:

$$h = (g_{i,x}^2 + g_{i,y}^2 - 1)^2. \tag{10}$$

We use D_i to define the protein diffusion that aims at keeping the robot away from its neighbors. The size of neighborhood varies according to the target shapes and the number of robots in the environment. Take the circular shape as an example, the neighborhood size can be defined as $\frac{2\pi L}{N}$, where L is the length of the circumference, and N is the total number of robots to be deployed on the circle. However, we will show in the simulations that the performance of the system is not sensitive to this parameter setup.

When a robot detects a neighboring robot, it will sense the protein concentration emitted from that neighbor so that it can keep itself away from that neighbor to avoid collision. Summing up the concentration of the diffused proteins from all neighbors, we have

$$D_{i,x} = \sum_{j=1}^{N_i} D_{i,x}^j,\ D_{i,y} = \sum_{j=1}^{N_i} D_{i,y}^j \tag{11}$$

where N_i denotes the number of its neighbors, and $D_{i,x}^j$and $D_{i,y}^j$are the diffusion along x-axis and y-axis, respectively, on robot i emitted from the neighbor robot j, which is defined as:

$$D_{i,x}^j = \frac{(g_{i,x} - g_{j,x})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}} \tag{12}$$

$$D_{i,y}^j = \frac{(g_{i,y} - g_{j,y})}{\sqrt{(g_{i,x} - g_{j,x})^2 + (g_{i,y} - g_{j,y})^2}}, \tag{13}$$

where the directions of $D_{i,x}^j$ and $D_{i,y}^j$ are defined to be the one from robot j to robot i along x-axis and y-axis, respectively.

Initially, the robots are located randomly in a 2D space. By following the dynamic defined in Equations (5) and (6), the robots can be deployed uniformly on the predefined shape autonomously. In other words, the system can converge to an equilibrium state defined by the target shape. Essentially, the shape information is the global information, which can be nicely embedded into the dynamics of each individual robot through the function $f(\mathbf{z}_i)$. It must be stressed, however, that each robot does not know its target position beforehand.

During the movement to the target shape, each robot is regulated by two dynamics. One is the morphogen gradient embedded in the regulation dynamics that drive the robot to the predefined shape, and the other is diffusion dynamics that is used to avoid collision between the robots. A good balance of the two dynamics can be achieved by optimizing the parameters in the model using a genetic algorithm, as we will discuss in Section 4.

A major concern of the gene regulatory network based self-organization algorithm is that under which condition the robots will converge to the target shape. Fortunately, we are able to provide a theoretic proof of the system's convergence to the target shape according to the Lyapunov theory as stated in the following theorem.

Theorem 1: For the dynamic system described by Equations (5)-(6), the systems states in Equation 5 will converge asymptotically to the predefined shape $h(g_x, g_y) = 0$, if $m \cdot k \leq a \cdot c$ and $k, c, a, m > 0$.

The reader is referred to [10] for a detailed proof of the convergence.

4. Robustness Analysis and Failure Recovery

While it is nice to have the theoretic convergence proof for the proposed self-organization system, it is still unclear how robust the convergence is against changes or uncertainties in the system and environment. For this purpose, several case studies are carried out on the system to verify its efficiency for shape formation and analyze its robustness in the presence of various of uncertainties which a real multi-robot system is very likely subject to. The efficiency verification and robustness analysis consists of the following steps. First, the parameters of the self-organization algorithm are optimized using a genetic algorithm for a given target shape and a given number of robots. Second, we examine the performance of the system when the number of robots changes and when the target shape changes. Third, we study the influence of the diffusion dynamics on the convergence of the system. Finally, system's robustness to sensory noise, self-localization failure and changes in the environment are studied.

4.1. Parameter Tuning Using A Genetic Algorithm

In Equations (5) and (6), five parameters, i.e., a, m, c, k, and b, need to be set up for the system. Generally, these parameters can be determined heuristically provided that the convergence condition in Theorem 1 can be satisfied. In implementing the system, we found that the setup of the parameters influences greatly on the time for the robots to converge to its final position and the trajectory length of the robots. Therefore, a genetic algorithm with simulated binary crossover (SBX) [4] is adopted at first to optimize these parameters to minimize the convergence time, provided that the conditions for the system's convergence as stated in Theorem 1 are not violated.

In the simulation study for minimizing the convergence time, we set up the number of robots to be 10, and set the population size of the GA to 20. The crossover probability is set to 0.9 and the distribution index for the SBX is 20. Mutation probability is by default set to be inversely proportional to the number of the decision variables, which is 5 in our case, therefore, is set to be 0.2, and the distribution index for mutation is set to be 20. The parameter setup, except for the population size, has been recommended in [4]. The simulation is run for 10 generations, which is relatively small, however, is found to be sufficient for optimizing the gene regulatory dynamics. All parameters in the dynamic system are initialized randomly between 1 and 200.

We define the target shape as a unit circle centered at (0, 0) and of a radius of 1. The robot system consists of 10 robots randomly distributed in the environment. During the deployment, the robots should approach to the unit circle while avoiding collision.

Since the final goal of the multi-robot system is to distribute the robots evenly on the circle, the position error to the target shape should be as small as possible after the system converges. Therefore, we define a threshold for the averaged position error between the robots' final position to the target shape as a constraint of this optimization problem when using the GA. In the following experiments, we set this the threshold to be $\frac{1}{r \cdot N}$, where r is the circumference the circle, and N is the total number of robots to be deployed on the circle. This distance constraint will be applied to all the following simulations for circle construction.

Fig. 1 shows the profile of average convergence time of the multi-robot system over generations. We can see that the system's convergence time decreases dramatically over the first generations, which shows that the GA-based optimization is effective.

In the following, we perform a number of tests to verify the efficiency and robustness of the GRN-based multi-robot self-organization algorithm. In the experiments, we use the optimized solution at the 10^{th} generation, unless otherwise stated. The parameters are as follows: $k = a$= 89.2685, c = 7.0350, b = 23.5424, and m = 3.4584.

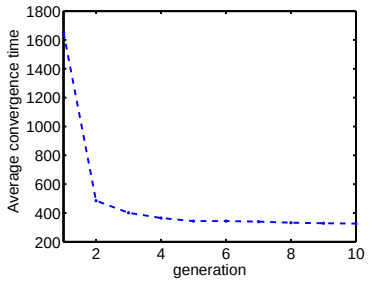
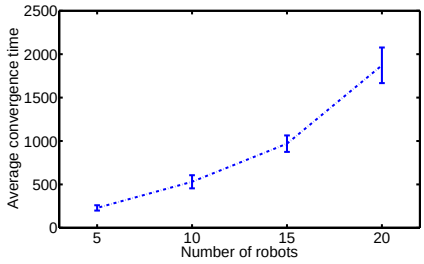


Figure 1. Average convergence time of the robot system over generations using the GA.

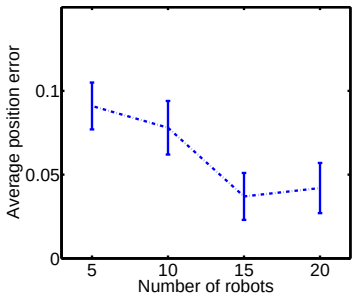
4.2. Deployment of Multiple Robots to a Unit Circle

The purpose of this case study is to compare the performance of the system using the parameters achieved by the GA in the previous section when the number of robots changes. The target shape is the same as in the previous section. Four groups of robots have been used to evaluate the proposed algorithm, which contains 5, 10, 15 and 20 robots, respectively. We performed 35 independent runs for each group, and the results of the resulting convergence

time (averaged over the robots) and average position error (averaged over the robots) are plotted in Fig. 2, where both the mean and standard deviation have been plotted. Due to space limit, we plot here only the trajectories with 20 robots in Fig. 3. Two observations can be made from Fig. 2. First, the average position error does not increase as the number robots increases. Second, the convergence process becomes much slower when the number of the robots increases. However, it must be pointed out that the increase in convergence time is mainly attributed to the time needed for the final tuning for an even distribution of the robots on the target shape, refer to the results in Section 4.4.



(a)



(b)

Figure 2. Mean and standard deviation of the (a) convergence time and (b) position error for 5, 10, 15 and 20 robots distributed to a circle.

4.3. Deployment of Multiple Robots to a Triangle

To show the flexibility of the GRN-based self-organization algorithm, we change the target shape from a circle to an equilateral triangle with the same parameter setup as before. The edge of the triangle is $\sqrt{3}$. The formation can be done in two steps. We first deploy the robots to a unit circle centered at (0,0) via the mechanism mentioned above. Then, we define the new h function as follows:

$$h(g_{i,x}, g_{i,y}) = \begin{cases} (g_{i,x} - \frac{\sqrt{3}}{3} g_{i,y} + \frac{\sqrt{3}}{3})^2 \\ (g_{i,x} + \frac{\sqrt{3}}{3} g_{i,y} - \frac{\sqrt{3}}{3})^2 \\ (g_{i,y} + \frac{1}{2})^2 \end{cases}$$

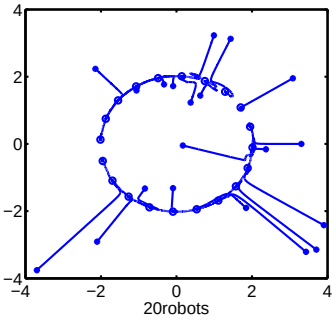


Figure 3. The trajectories of 20 robots using GRN-based method to construct a circle. Initial positions are denoted as a dot and final positions are denoted as a small circle.

$$\text{if} \quad \begin{array}{l} g_{i,y} > \sqrt{3} g_{i,x} + 1 \\ g_{i,y} > -\sqrt{3} g_{i,x} + 1 \\ g_{i,y} < \frac{1}{2} \end{array}$$

Note that after the first stage of deployment, robots have been deployed on the unite circle as shown in Fig. 4. By following the GRN dynamics and using the new h function, we can deploy multiple robots to an equilateral triangle.

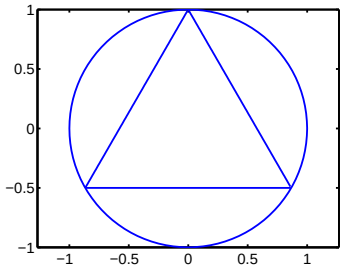
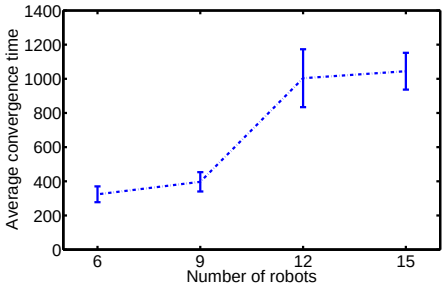


Figure 4. The relationship between the equilateral triangle and the unit circle.

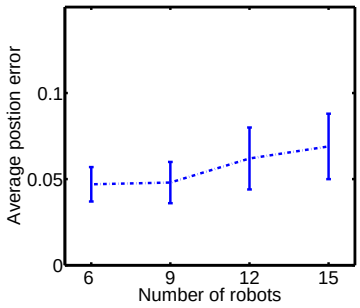
In order to ensure ultimate uniform distribution of the robots, we define robots' neighbor range to be $d = \frac{L_{edge}}{(N/3)+1}$, where N refers to the number of robots in the system and L_{edge} refers to the length of the triangle's edge.

Again, we conducted 35 independent simulation runs for 6, 9, 12, and 15 robots. The results describing the mean and standard deviation of the average convergence time and average position error are presented in Fig. 5. Similar to the circle case, the convergence time increase considerably as the number of robots increase, while the position error is much less sensitive to the number of robots and are acceptable. For example, the mean average position error in case of 15 robots is 0.069, which is reasonably small on a triangle whose edge length equals 1.732. Notice also that the convergence time may vary dramatically for different initial positions.

Fig. 6 shows the trajectories of the robots moving from



(a)



(b)

Figure 5. Mean and standard deviation of the (a) convergence time and (b) position error for 6, 9, 12 and 15 robots distributed to a triangle.

the initial positions to the target positions on the predefined triangle, where we use 9 robots randomly initialized in a $4x4$ 2-D environment. From Fig. 6, we can see that the initially randomly distributed robots can be deployed to the equilateral triangle uniformly under the control of the GRN-based dynamics. One constraint of the triangle deployment is that the total number of robots in the system should be dividable by 3.

4.4. Influence of the Diffusion on Dynamics

In the theoretical analysis of system convergence, we have omitted the diffusion term for the sake of simplicity by assuming that the diffusion terms do not influence the convergence substantially. To empirically verify this assumption, we conducted 35 independent runs without the diffusion term when 5, 10, 15, and 20 robots are distributed to a unit circle. The results are presented in Fig. 7. Compared those results with the diffusion term presented in Fig. 2, no significant difference can be observed in the position error. By contrast, the convergence time becomes much shorter. As mentioned in Section 4.2, this is due to the fact that it takes much more time for the robots to adjust their final position under the influence of the diffusion dynamics than the time for them to converge to the desired shape.

Fig. 8 shows the robots' trajectories converging to the

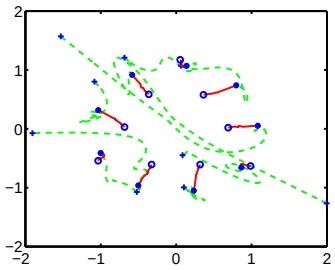
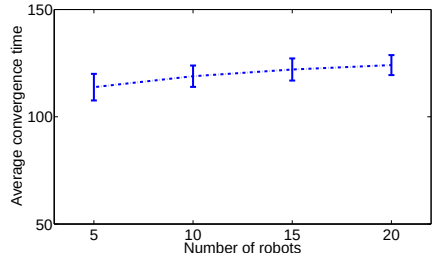
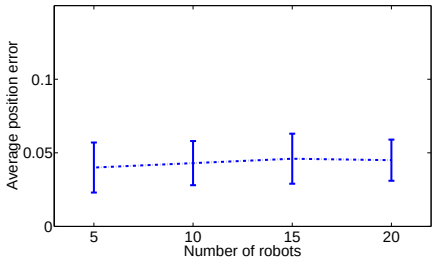


Figure 6. The robots' trajectories moving from random initial positions to the predefined triangle. The initial positions are represented by "+", the intermediate positions on the unit circle are represented by a dot , and the final positions on the triangle are represented by a small circle. The trajectories of the first phase deployment (deployment to a circle) are plotted in dashed lines and the second phase trajectories (deployment to the triangle) are plotted in solid lines.



(a)

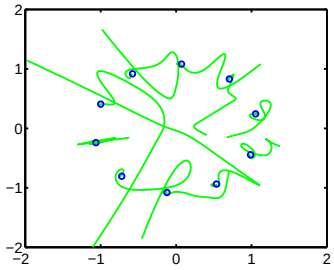


(b)

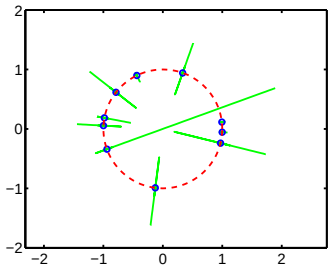
Figure 7. Mean and standard deviation of the (a) convergence time and (b) position error for 5, 10, 15 and 20 robots distributed to a circle without diffusion.

unit circle from their initial positions. From Fig. 8, we can see that both groups of robots can converge to the target shape eventually, although the robots under the dynamics with protein diffusion can deploy on the target shape uniformly, whereas the robots controlled by the regulatory dynamics without the diffusion deploy on the target shape in a random manner. This simple study verifies empirically that the protein diffusion among the robots do not affect substantially the system's ability to converge to the target

shape. Nevertheless a rigorous theoretical proof is still under investigation.



(a)



(b)

Figure 8. The trajectories of the robots using the GRN-based method with and without protein diffusion, where the small circles denote the robots, and the solid lines are the moving trajectories. (a) With diffusion; (b) Without diffusion.

4.5. Robustness to Sensory Noise

When we intend to apply the gene regulatory network based self-organization algorithm to real robots, we must take a few additional constraints into account. These constraints include noise in distance measurement and self-localization, and eventually a complete failure in self-localization. We first consider the influence of sensory noise in this section by deliberately adding noise into the sensory measurements and localization. We perform 35 independent runs using 10 robots with random position initialization, and calculate the mean and standard deviation of the final position errors to the unit circle with 5% and 10% noise in robot localization, as listed in Tables 1 and 2, respectively. From the tables, we can conclude that the position errors of the system are insensitive to the noise in measurements.

4.6. Recovery from Self-Localization Failures

We notice from the previous section that the system is robust to noise in self-location. In the extreme case, robots may get lost in the environment due to a complete self-localization failure. In other words, some of the robots may fail to localize themselves in the global coordinates.

Table 1. Mean and standard deviation of the position error when the distance measurements are subject to sensory noise.

	Without noise	5% noise	10% noise
mean	0.0421	0.0437	0.0470
std	0.024	0.0096	0.010

Table 2. Mean and standard deviation of position error when the self-localization is subject to noise.

	Without noise	5% noise	10% noise
mean	0.0421	0.0428	0.0445
std	0.024	0.0083	0.01

In this case study, we compare the performance of the system with or without a recovery algorithm. The recovery algorithm is to estimate the positions of the robots experiencing self-localization failure using the trilateration method combined the Kalman filter.

In developing the recovery algorithm, we assume that robots are equipped with distance sensors that can detect the relative distances to its neighbors. Therefore, a robot that fails to localize itself can get its neighbors' positions through local communications. We further assume that localization and distance measurements are contaminated by zero-mean white Gaussian noise. All measurements are supposed to be mutually independent. With these assumptions, a trilateration method can be used to recover the current position of the robot that fails to self-localize its position. If the current position of the lost robot is denoted by (x, y), which is to be estimated, the positions of three neighbors are denoted by (x_i, y_i), where i = 1, 2, 3, and the distances from the lost robot to the three neighbors are measured as d_i, where i = 1, 2, 3, then the position of the lost robot can be estimated by solving the following equations:

$$(x - x_i)^2 + (y - y_i)^2 = d_i^2, \text{ for } i = 1, 2, 3. \quad (14)$$

To improve the accuracy of the position estimation, the Kalman filter can be applied, which consists of the following main steps:

1) State equation:

$$\mathbf{x}(k+1) = \mathbf{F}(k)\mathbf{x}(k) + \mathbf{G}(k)\mathbf{v}(k) \quad (15)$$

2) Predict:

$$\hat{\mathbf{x}}(k+1|k) = \mathbf{F}(k)\hat{\mathbf{x}}(k|k) \quad (16)$$

$$\mathbf{P}(k+1|)) = \mathbf{F}(k+1)\mathbf{P}(k|k)\mathbf{F}^T(k+1) + \mathbf{Q}(k) \quad (17)$$

3) Update:

$$\tilde{\mathbf{y}}(k+1) = \mathbf{z}(k+1) - \mathbf{H}(k+1)\hat{\mathbf{x}}(k+1|k)$$
$$\hat{\mathbf{x}}(k+1|k+1) = \hat{\mathbf{x}}(k+1|k) + \mathbf{K}(k+1)\tilde{\mathbf{y}}(k) \quad (18)$$

where

$$\mathbf{z}(k+1) = \mathbf{H}(k+1)\mathbf{x}(k+1) + \mathbf{w}(k+1) \quad (19)$$

$$\mathbf{P}(k+1)|k+1) = \mathbf{P}(k+1|k) - \mathbf{K}(k+1)\mathbf{H}(k+1)\mathbf{P}(k+1|k) \quad (20)$$

$$\mathbf{S}(k+1) = \mathbf{H}(k+1)\mathbf{P}(k+1|k)\mathbf{H}^T(k+1) + \mathbf{R}(k+1) \quad (21)$$

$$\mathbf{K}(k+1) = \mathbf{P}(k+1|k)\mathbf{H}^T(k+1)\mathbf{S}(k+1)^{-1} \quad (22)$$

$\mathbf{x}(k) = [x, y]^T$ is the system state vector, where x and y are the x-position and y-position of lost robot in a 2-D environment. $\mathbf{F}(k)$ is the state transition matrix, which is defined as $\mathbf{F}(k) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. $\mathbf{G}(k)$ is defined as $\mathbf{G}(k) = \begin{bmatrix} \tau & 0 \\ 0 & \tau \end{bmatrix}$, where τ is the length of the time step interval and is defined as τ= 0.01 in the simulation. $\mathbf{H}(k+1)$ is the observation matrix, and is defined as $\mathbf{H}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are zero-mean white Gaussian process noise and measurement noise with covariance $\mathbf{Q}(k)$ and $\mathbf{R}(k)$, respectively. $\mathbf{Q}(k)$ is initialized as $\mathbf{Q}(1) = cov(\mathbf{Gv}) = E((\mathbf{Gv})(\mathbf{Gv})^T) = \mathbf{GG}^T$. For simplicity, we initialize $\mathbf{R}(k)$ as $\mathbf{R}(1) = cov(\mathbf{ww}^T) = E(\mathbf{ww}^T) = \mathbf{I}$. $\mathbf{z}(k)$ are the measurements to the Kalman filter, which are the estimated location of the lost robot using the above described trilateration method.

In the case study, ten robots are initially randomly distributed in the environment. We assume one of the robots loses its position. When this self-localization failure occurs, the recovery algorithm based on trilateration method combined with the Kalman filter is triggered. White Gaussian noise is added to the distance measurements and the positions of neighbors. The time interval of the Kalman filter is defined to be 0.01. Fig. 9 shows the differences of the estimated values and the true values of x-position and y-position of the lost robot during the recovery process, from which we can see that the Kalman filter based recovery algorithm works properly.

The convergence process assisted with the position recovery algorithm is shown in Fig. 10. The average convergence time and position error, as well as the position error of the lost robot with and without Kalman filter based trilateration are listed in Table 3. From the Table, we notice that the Kalman filter has dramatically improved the position estimation of the lost robot.

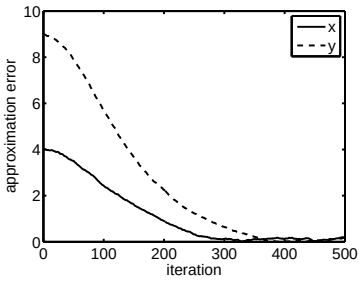


Figure 9. The difference between the estimated position and the true position of the robot in a typical run.

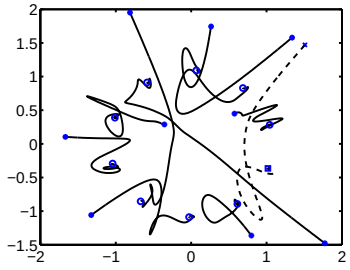


Figure 10. The trajectory of the deployment with one lost robot. The dashed line represents the trajectory of the lost robot whose position is estimated with the recovery algorithm.

4.7. Robustness to Environmental Perturbations

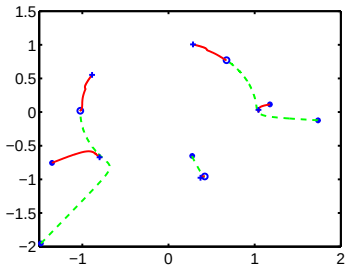
In this case study, we evaluate the robustness of the system to the changes in the number of robots after shape formation has already finished and perturbations in the environment, such as a moving obstacle. More specifically, two scenarios are considered. In the first scenario, we test if the robots can reorganize themselves when additional robots join the team after convergence. In the second test, we check the robots' behavior given a moving obstacle in the environment.

Fig. 11 plots the trajectories of robots before newcomers join, and the self-reorganization process of the robots when the newcomers join the team. This result demonstrates the autonomous re-organization capability of the proposed approach, because no additional controlled process is needed for re-organization.

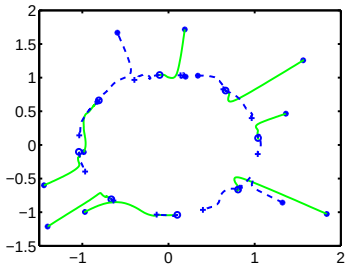
Next, we examine the behavior of the robots when there is a moving target in the environment. Fig. 12 shows two snapshots captured before an obstacle moves towards the robot team and shortly before the obstacle moves away from them. From the snapshots, we can see that the robots deployed on the shape are able to move temporarily away to avoid a collision with the obstacle. Although only one obstacle is implemented in our simulation, similar behaviors are observed for multiple moving obstacles. This simulation study demonstrates that the system can autonomously adapt

Table 3. Trilateration recovery with and without Kalman filter.

	Convergence time (mean ± std)	Position error (mean ± std)	The lost robot's position error (mean ± std)
With location recovery	609.67 ± 103.53	0.105 ± 0.024	0.188 ± 0.032
Without location recovery	4073.6 ± 3106.30	0.693 ± 0.178	5.1377 ± 1.803

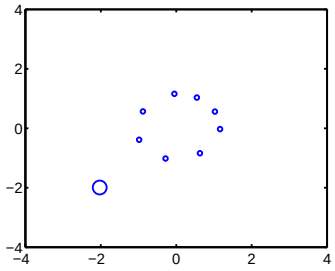


(a)

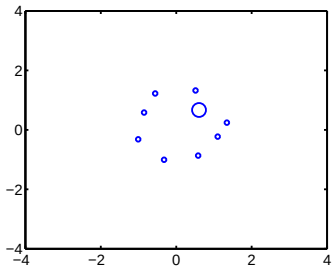


(b)

Figure 11. Trajectories of multi-robots during self-organization in the presence of newcomers. The initial positions of the robots are denoted by '*', the intermediate positions where the first batch of robots are located are denoted by 'o', and the final positions of all the robots are denoted by '+'. The dash lines represent the initial deployment trajectories of the first batch of robots and the solid lines represent the trajectories of all the robots after incorporating newcomers. (a) 3 robots with 2 newcomers; (b) 8 robots with 4 newcomers.



(a)



(b)

Figure 12. A set of snapshots of the robot's behavior of reacting to a moving obstacle. (a) The obstacle is moving towards the robots (b) robots are adapting to avoid the obstacle.

itself to the environmental changes.

5. Conclusions

In this paper, we have presented a number of case studies investigating the performance of a decentralized GRN based approach to self-organizing multi-robot systems suggested in [10]. Three main aspects of the algorithm are studied. First, we examine the sensitivity of the system's performance on the changes in the number of robots and the target shape. This is implemented by checking the convergence time and position error of the system using a set of parameters optimized for a particular situation. Second, we study the robustness of the system to sensory noise in distance measurement and self-localization. As an extreme case, we also look into the case when a complete self-localization failure occurs. In this case, a location recovery algorithm is introduced with the help of trilateration and a Kalman filter. Third, we check the re-organization capability of the system in two situations. In the first situation, new robots need to join a set of robots that have converged to the target shape. In the second situation, an obstacle is moving around in the region where the target shape is situated.

Through all the case studies regarding the effectiveness and robustness of the self-organization algorithm mentioned above, we can conclude that gene regulatory network based method for self-organization of multiple robots for formation is effective, insensitive to parameter setup and robust to changes in the system or in the environment. Nevertheless, in case of a complete self-localization failure, accurate recovery of the position information of the lost robot is important so that all robots can be distributed to the target shape. Of

course, if we only care about that the target shape is properly covered when there are sufficient robots, the system is also insensitive to localization failure of a few individual robots.

References

[1] Alon, U.: *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC (2006)

[2] Kondacs, A.: Biologically inspired self-assembly of two-dimensional shapes using global to local compilation. In: *Int. Joint Conf. on Artificial Intelligence*, pp. 633-638 (2003)

[3] Cheng, J., Cheng, W., Nagpal, R.: Robust and self-repairing formation control for swarms of mobile agents. In: *Twentieth National Conference on Artificial Intelligence*. pp. 59-64 (2005)

[4] Deb, K., Agrawal, R.B.: Simulated binary crossover for continuous search space. *Complex Systems*, 9:115-148 (1995)

[5] de Jong, H.: Modeling and simulation of genetic regulatory systems: a literature review. *Journal of Computational Biology*, 9:67–103(2002)

[6] Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a Colony of Cooperating Robots. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 26:29-41 (1996)

[7] Endy, D., Brent, R.: Modeling cellular behavior. *Nature*, 409:391–395 (2001)

[8] Geard, N.: Modeling gene regulatory networks: systems biology to complex systems. ACCS Technical Report (2004)

[9] Gierer, A.: Generation of biological patterns and form: some physical, mathematical, and logical aspects. *Prog. Biophys. Mol. Biol.*, 37:1–47 (1981)

[10] Guo, H., Meng, Y., Jin, Y.: A Cellular Mechanism for Multi-Robot Formations via Evolving Gene Regulatory Networks. *BioSystems* (2009) (Under review)

[11] Hasty, J. and McMillen, D., Isaacs, F., Collins, J.J.: Computational studies of gene regulatory networks: in numero molecular biology. *Nat. Rev. Genet.*, 2:268–279 (2001)

[12] Jatmiko, W., Sekiyama, K., Fukuda, T.: A PSO-based Mobile Robot for Odor Source Localization in Dynamic Advection-Diffusion with Obstacles Environment: Theory, Simulation and Measurement. *IEEE Computational Intelligence Magazine*, 2:37–51 (2007)

[13] Jin, Y., Meng, Y., Sendhoff, B.: Influence of Regulation Logic on the Easiness of Evolving Sustained Oscillation for Gene Regulatory Networks. In: *IEEE Symposium on Artificial Life*, pp.61-68 (2009)

[14] Kauffman, S.A.: *The Origins of Order: Self-organization and Selection in Evolution*, Oxford University Press, New York (1993)

[15] Kitano, K.: Biological robustness. Nature, 5:826–837 (2004)

[16] Khoshnevis, B., Bekey, G.A.: Centralized sensing and control of multiple mobile robots. *Computers in Industrial Engineering*, 35:503–506 (1998)

[17] Klavins, E.: *Communication Complexity of Multi-Robot Systems*. Springer Tracts in Advanced Robotics. 7:275–292 (2003)

[18] Kwon, Y.-K., Cho, K.-H.: Analysis of feedback loops and robustness in network evolution based on Boolean models. *BMC Bioinformatics*, 8:430 (2007)

[19] Maini, P., Painter, J.K., Nguyen, P.C.: Spatial pattern formation in chemical and biological systems. *J. Chem. Soc., Garaday Trans.*, 93:3601–3610 (1997)

[20] Mamei, M., Vasirani, M., Zambonelli, F.: Experiments in morphogenesis in swarms of simple robots. *Applied Artificial Intelligence*, 18, 9-10:903-919

[21] Meng, Y., Gan, J.: LIVS: Local interaction via virtual stigmergy coordination in distributed search and collective cleanup. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1371-1376 (2007)

[22] Meng, Y., Gan, J.: A Distributed Swarm Intelligence based Algorithm for a Cooperative Multi-Robot Construction Task. In: *IEEE Swarm Intelligence Symposium*, pp. 21–23 (2008)

[23] Nolfi, S., Floreano, D.: *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, Cambridge, MA (2000)

[24] Pfeifer, R., Lungarella, M., Iida, F.: Self-organization, embodiment, and biologically inspired robotics. *Science*, 318:1088-1093 (2007)

[25] Pugh, J., Martinoli, A.: Inspiring and Modeling Multi-Robot Search with Particle Swarm Optimization, In: *Proceedings of the 2007 IEEE Swarm Intelligence Symposium*, pp. 332-339 (2007)

[26] Shen, W., Will,P., Galstyan, A: Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms, *Autonomous Robots*, 17:93-105 (2004)

[27] Salazar-Ciudad, I., Garcia-Fernandez, H., Sole, R.V.: Gene networks capable of pattern formation: from induction to reaction-diffusion. *Journal of Theoretical Biology*, 205:587–603 (2000)

[28] Smolen, P., Baxter, D.A., Byrne, J.H.: Modeling transcriptional control in gene networks: Methods, recent results, and future directions. *Bull. Math. Biol.*, 62:247-292 (2000)

[29] Taylor, T.: A Genetic Regulatory Network-Inspired Real-Time Controller for a Group of Underwater Robots. In: *Proceedings of Eighth Conference on Intelligent Autonomous Systems*, pp. 403-412 (2004)

[30] Turing, A.M.: The chemical basis of morphogenesis. *Philos Trans. R. Soc. London B*, 237:37-72 (1952)

[31] Werfel, J.: Building blocks for multi-robot construction. In: *Seventh International Symposium on Distributed Autonomous Robotic Systems*, pp. 285-294 (2004)