

A Morphogenetic Approach to Flexible and Robust Shape Formation for Swarm Robotic Systems

Yan Meng^{a*}, Hongliang Guo^a, and Yaochu Jin^b

^a *Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA, yan.meng@stevens.edu, hguo@stevens.edu*

^b *Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK yaochu.jin@surrey.ac.uk*

** Corresponding author: email: yan.meng@stevens.edu, phone: 1-201-216-5496, fax: 1-201-216-8246*

Abstract

Embryonic development of multi-cellular organisms is governed by gene regulatory networks (GRNs), which are a collection of genes that interact with one another and with other chemicals in the cell. Inspired by the morphogenesis of biological organisms, in this paper, we propose a morphogenetic approach using a gene regulatory network (GRN) for swarm robotic systems to form complex shapes in a distributed manner. The target pattern, represented by non-uniform rational B-spline (NURBS), is embedded into the gene regulatory model, analogous to the morphogen gradients in multi-cellular development. Since the total number of robots is unknown to each robot, a dynamic neighborhood adaptation mechanism is proposed to evenly deploy the robots on the boundary of the target pattern. A theoretical proof of the system convergence is provided. Various simulation studies demonstrate that the proposed algorithm offers an effective and robust distributed control mechanism for swarm robotic systems to construct complex shapes. Furthermore, proof-of-concept experiments were successfully undertaken using e-puck mobile robots, which demonstrate that the proposed model works well with physical constraints of real robots.

Keywords: Gene regulatory network, morphogenesis, swarm robots, shape formation, self-organization, distributed systems.

1. Introduction

Swarm robots involve a large number of simple robots with limited computing, communication and sensing capabilities. Compared to more expensive and sophisticated individual robots, swarm robots are able to accomplish inherently distributed tasks in the presence of uncertainties. To this end, we need to develop self-organizing algorithms for swarm robotic systems, whose global behaviors emerged from local interactions of the robots are flexible and robust to the changing environments.

Major applications of self-organizing swarm robotic systems include surveillance and monitoring of large areas, urban search and rescue, deployment of sensor networks [1, 2], large area exploration [3], hazardous material collection, mine detection, map building [4], among others. One common requirement in these tasks is that robots need to form complex shapes to accomplish the given task. Therefore, we concentrate in this work on complex shape formation for swarm robotic systems, which, nevertheless serves as an example of real-world problems where self-organization and self-repair are indispensable.

In the last two decades, multi-robot pattern formation has been extensively studied. Initially, most research work focuses on leader/neighbor-following formation [5-10], where the major control goal is to keep the consensus of the

robots' movements, and most of the methods are supported by a rigorous theoretical proof of the system convergence. Later on, some potential-field based methods [11-21] have been developed for formatting various patterns, where robots follow the gradients of the potential fields, which is the sum of the virtual attractive and repulsive forces. However, due to the inherent features of potential fields, sometimes robots may get trapped at some local optimal points.

Recently, a few bio-inspired methods have been proposed for robust pattern formation, which either use hormone-based models [22, 23] or cellular mechanisms [24-29]. Most of these algorithms have demonstrated promising results for simple clustering or migration tasks. However, more complex behaviors such as deploying a group of robots onto the boundary of particular complex patterns remain to be achieved. Furthermore, most bio-inspired approaches only provide heuristic local rules and cannot provide theoretical proof of the system convergence. To tackle these issues, in this paper, we develop a new bio-inspired approach for distributed multi-robot pattern formation with a theoretical proof of system convergence.

This work is mainly inspired by the biological morphogenesis in multi-cellular organisms. Increasing evidence has shown that biological morphogenesis is a self-organizing and self-assembling process through cellular and molecular interactions under genetic and environmental

control [30]. The process of multi-cellular morphogenesis is under the control of gene regulatory networks (GRNs), and many computational models for GRNs have been proposed [31-33]. These models not only help us in understanding biological processes, but also provide us new powerful tools for designing engineering systems, such as self-organizing robotic systems.

In biological systems, zygotes containing the same genetic material will grow into a multi-cellular organism with the similar phenotype despite that they are in slightly different environments and that genetic changes occur from time to time during the development. Biological morphogenesis can be seen as a self-organization process where cells move to their destination governed by gene regulatory dynamics and cell-cell interactions.

The basic idea of the self-organizing algorithm proposed in this work is inspired from biological morphogenesis. In the approach, each robot is seen as a cell containing a virtual DNA (a GRN-based controller), in which the target global pattern formation is encoded. By following the control dynamics described by this GRN-based controller, the robots are able to form the global target pattern in a distributed manner.

A preliminary GRN-based model for multi-robot formation was reported in our previous work [34], where we showed that the model is scalable to the number of robots, and is robust to parameter changes and individual robot failures [35]. However, the previous model has several major limitations. First, robots can only build simple shapes such as circles or squares as an analytic function has to be used for describing the target shape. Second, the total number of robots in the system is assumed to be known to the robots for an even deployment on the target pattern. Third, the model was verified in simulations only.

This paper substantially extends our preliminary model in the following ways: (1) A new shape representation method is proposed using the non-uniform rational B-spline (NURBS) model [36], which is a powerful tool for representing arbitrarily complex 2D or 3D shapes that may or may not be described by an analytic function. (2) The total number of robots in the system needs not to be known beforehand. To this end, a neighborhood adaptation mechanism is proposed to automatically adjust the distance between the neighboring robots so that robots will be evenly deployed on the target pattern. (3) A theoretical proof of the system convergence to the target pattern is provided to guarantee that the multi-robot system can eventually converge to any target pattern that can be defined by a NURBS representation. (4) Two proof-of-concept experiments using multiple e-puck mobile robots are performed in an indoor environment to evaluate the feasibility and effectiveness of the proposed model with physical constraints of robots.

The paper is organized as follows. Section 2 describes the problem statement. The morphogenetic approach using a GRN-based controller for self-organizing swarm robotic

systems is presented in Section 3, together with a theoretical proof of system convergence. The neighborhood adaptation mechanism is also described in Section 3. To evaluate the effectiveness and robustness of the proposed method, various simulation scenarios have been studied and discussed in Section 4. Experimental results using e-puck mobile robots are given in Section 5. Conclusions and future work are presented in Section 6.

2. Problem Statement

The problem we are considering in this paper is to deploy a swarm of autonomous mobile robots onto the boundary of arbitrarily complex 2D/3D target pattern as evenly as possible. A general assumption is that each robot has limited computing power and limited communication and sensing capabilities.

A few more specific assumptions have also been made for this model. First, each robot can localize itself within a global coordinate system using its onboard sensors, such as encoders and gyroscope. Second, each robot can measure the relative distance to its neighbors if they are within its sensor range using their onboard distance sensors, such as sonar or infrared sensors. Each robot can communicate with its local neighbors within its sensor communication range. Third, since this is a distributed homogeneous multi-robot system, each robot has to make its own decisions using the same proposed GRN-based controller based on its own local environment and interactions with other robots. Last, the global target pattern represented by the NURBS model is embedded into the GRN-based controller. But each robot doesn't know its own destination position on the target pattern. In other words, each robot has to make its own decisions (based on its local view of the environment) to find an appropriate destination position on the target pattern.

Please be noted that providing the information of the target pattern to each robot doesn't mean that the proposed method is a centralized approach since no robot has a global view of the whole system and there is no centralized control in the system, as discussed in [37].

3. The Approach

3.1. Computational Models of Gene Regulation

Multi-cellular morphogenesis is under the control of gene regulatory networks. When a gene is expressed, information stored in the genome is transcribed into mRNA and then translated into proteins. Some of these proteins are transcription factors that can regulate the expression of their own or other genes, thus resulting in a complex network of interacting genes termed as a gene regulatory network (GRN). To understand the emergent morphology resulting from the interactions of genes in a regulatory network, reconstruction of gene regulatory pathways using a computational model has become popular in systems biology [38]. A large number of computational models for GRNs have been suggested [31-33, 39, 40,]. Among others,

ordinary or partial differential equations have widely been used to model regulatory networks.

The model used for self-organizing swarm robotic systems in this work is basically inspired from a GRN model for describing the gene expression data of developmental processes [41], which can be considered as a generalized reaction-diffusion model with a sigmoid function:

$$\frac{dg_{ij}}{dt} = -\gamma_j g_{ij} + \phi \left[\sum_{l=1}^{n_g} W^{jl} g_{il} + \theta_j \right] + D_j \nabla^2 g_{ij}, \quad (1)$$

where g_{ij} denotes the concentration of j -th gene product (protein) in the i -th cell. The first term on the right-hand side of Eqn. (1) represents the degradation of the protein at a rate of γ_j , the second term specifies the production of protein g_{ij} , and the last term describes protein diffusion at a rate of D_j .

ϕ is an activation function for the protein production, which is usually defined as a sigmoid function $\phi(z) = 1/(1 + \exp(-\mu z))$. The interaction between the genes is described with an interaction matrix W^{jl} , the element of which can be either active (a positive value) or repressive (a negative value). θ_j is a threshold for activation of gene expression. n_g is the number of proteins.

3.2. Shape Representation Using NURBS

We use the non-uniform rational B-spline (NURBS) model to represent complex shapes to be constructed by the robots. NURBS is a mathematical model commonly used in computer graphics and structural design for generating and representing curves and surfaces. NURBS can offer one common mathematical form for both analytic and freeform shapes.

A NURBS curve is defined by its order, a set of weighted control points, and a knot vector. The control points define the shape of the curve, and the knot vector is a set of parameters that determines where and how the control points affect the NURBS curve. A NURBS model can represent both curve and surface in a two- or three-dimensional Cartesian space. Let $B_{i,k}(u)$ be the B-spline basis functions of the NURBS model, where i corresponds to i -th control point, and k denotes the degree of the basis function. In the NURBS model, a curve can be defined as a combination of a set of piecewise rational basis functions with $n+1$ control points \mathbf{p}_i and the associated weights w_i as follows [36]:

$$\mathbf{c}(u) = \frac{\sum_{i=1}^n \mathbf{p}_i w_i B_{i,k}(u)}{\sum_{j=1}^n w_j B_{j,k}(u)} \quad (2)$$

where n is the number of control points, u is a parameter in the NURBS representation. For basis functions of degree $k-1$, a NURBS curve has $n+k+1$ knots t_i in a non-decreasing sequence: $t_0 \leq t_1 \leq \dots \leq t_{n+k}$. The basis functions are defined recursively as:

$$B_{i,k}(u) = \begin{cases} 1 & \dots \dots \dots t_i \leq u \leq t_{i+1} \\ 0 & \dots \dots \dots \text{otherwise} \end{cases}$$

with

$$B_{i,k}(u) = \frac{u - t_i}{t_{i+k+1} - t_i} B_{i,k-1}(u) + \frac{t_{i+k} - u}{t_{i+k} - t_{i+1}} B_{i+1,k-1}(u)$$

The range of the parameter is $t_{k-1} \leq u \leq t_{k+1}$. The readers are referred to [42] and [36] for a more detailed description of NURBS.

3.3. The GRN-based Model

In the GRN-based model, it is assumed that each robot corresponds to a single cell in a cell-robot metaphor. Within each cell, there are two different types of protein products, namely types G and P. Protein type G consists of two proteins, which correspond to the x and y positions of a robot in a 2D environment, respectively. If a 3D shape is to be formed, then three proteins of type G are needed to describe the position of the robot. Similarly, protein type P consists of two proteins for a 2D environment and three proteins for a 3D environment, which represent an internal state vector of the robot. Meanwhile, proteins of type G can diffuse into the neighboring cells, thus influencing the protein production in these cells. This kind of local diffusion through cell-cell signaling can, in the robot metaphor, prevent the robot from colliding with its neighbors. Finally, the production of proteins is regulated by a maternal morphogen gradient M, which corresponds to the embedded information of the target pattern for the robots to form.

From the biological point of view, the regulatory relationship between the artificial morphogen M, protein type P and protein type G can be depicted in Fig. 1. In Fig. 1, we can see that morphogen M regulates the production of both protein type P and protein type G (indicated by the arrow). Protein type P can also regulate the production of protein type G, and protein type P can auto-regulate itself.

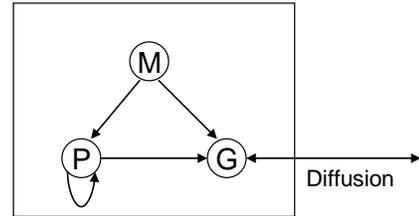


Fig. 1. The GRN structure in our model (from biological system point of view).

Based on the structure illustrated in Fig. 1, the dynamics of the GRN controller for robots to build a 2D pattern can be described by the following differential equations:

$$\begin{aligned} \frac{dg_{i,x}}{dt} &= -aw_{i,x} + mp_{i,x} \\ \frac{dg_{i,y}}{dt} &= -aw_{i,y} + mp_{i,y} \end{aligned} \quad (3)$$

$$\begin{aligned}\frac{dp_{i,x}}{dt} &= -cp_{i,x} + kf(w_{i,x}) + bD_{i,x} \\ \frac{dp_{i,y}}{dt} &= -cp_{i,y} + kf(w_{i,y}) + bD_{i,y}\end{aligned}\quad (4)$$

where i is the index of robot, $i=1,2,\dots,n$, and n is the total number of robots in the system. $g_{i,x}$ and $g_{i,y}$ are x -, y -position of robot i , respectively (corresponding to protein concentrations of type G in cell). $p_{i,x}$ and $p_{i,y}$ represent the internal states of robot i (corresponding to protein concentrations of type P in cell i). For any 3D pattern formation, we only need to add one more vector on the z -coordinate which can be defined similarly as x - and y -coordinate. This rule will be applied to all the following variables as well.

In Eqn. (4), the first term denotes the decay of the internal state vector $\mathbf{p} = \{p_{i,x}, p_{i,y}\}$, the third term denotes the diffusion process. D_i denotes the sum of normalized distances to robot i from its neighboring robots (corresponding to the overall diffused protein concentration perceived by cell i emitted from its neighboring cells), which is defined as:

$$D_{i,x} = \sum_{j=1}^{N_i} D_{i,x}^j, \quad D_{i,y} = \sum_{j=1}^{N_i} D_{i,y}^j \quad (5)$$

where N_i denotes the number of neighbors of robot i . $D_{i,x}^j$ and $D_{i,y}^j$ represent the normalized distances in x - and y -coordinate from robot j to robot i , respectively (corresponding to the protein diffusions from cell j to cell i) which are defined as followings:

$$D_{i,x}^j = \frac{(g_{i,x} - g_{j,x})}{\sqrt{(g_{i,x} - g_{i,x})^2 + (g_{i,y} - g_{j,y})^2 + (g_{i,z} - g_{j,z})^2}} \quad (6)$$

$$D_{i,y}^j = \frac{(g_{i,y} - g_{j,y})}{\sqrt{(g_{i,x} - g_{i,x})^2 + (g_{i,y} - g_{j,y})^2 + (g_{i,z} - g_{j,z})^2}} \quad (7)$$

a, b, c, m , and k are constant coefficients.

The target pattern defined by NURBS into the GRN-based model for pattern formation is embedded into the second term in Eqn. (4), where $f(w_{i,x})$ and $f(w_{i,y})$ is defined as the following sigmoid function:

$$f(w_{i,x}) = \frac{1 - e^{-w_{i,x}}}{1 + e^{-w_{i,x}}} \quad f(w_{i,y}) = \frac{1 - e^{-w_{i,y}}}{1 + e^{-w_{i,y}}} \quad (8)$$

And $\mathbf{w} = \{w_{i,x}, w_{i,y}\}$ is defined as

$$w_{i,x} = g_{i,x} - c_{i,x}(u), \quad w_{i,y} = g_{i,y} - c_{i,y}(u) \quad (9)$$

where $c_{i,x}(u)$ and $c_{i,y}(u)$ are x - and y -coordinates of a 2D target curve represented by the NURBS model. Here, u is the parameter of NURBS model ranging from 0 to 1. According to our model, given a fixed $u \in [0,1]$ and Eqns. (3) and (4), robots should be able to converge to a point on the target pattern represented by the NURBS model. Ideally, robot i

should go to an unoccupied point on the target pattern closest to its current position. To this end, the u for the robot i (u_i) should satisfy the following condition:

$$u_i = \arg \min(w_{i,x}^2 + w_{i,y}^2). \quad (10)$$

Since we do not have an analytic function to calculate $c_{i,x}(u)$ and $c_{i,y}(u)$, we discretize u in the range of $[0,1]$ based on the estimated number of robots in the system, which will be discuss in detail in Section 3.7. Assume that there are n robots in the system based on the estimation, u can be discretized into $u = [0, 1/(n-1), 2/(n-1), \dots, 1]$.

Since *none of the robots have any predefined position on the target pattern*, each robot has to randomly pick one of the values (i.e., one destination point) from u . Thus, it is possible that more than one robot picks the same destination point. However, due to the diffusion term in Eqn. (4), robots will adjust their dynamics automatically to keep a certain distance from each other. Eventually, each robot will pick a unique u value that leads to the minimization of Eqn. (10) based on the local interaction between the robots. Similarly, the diffusion term in Eqn. (4) can also be used for robot-obstacle avoidances. When a robot is near an obstacle, it would sense its distance to the obstacle, and thus a diffusion term is generated and added into Eqn. (4). This diffusion term would influence the motion dynamics of the robot so as to keep the robot away from the obstacle.

3.4. Discussions

From Eqn. (3), we can see that the robot aims to reduce the position errors from its current position to the dynamically-selected destination position on the target pattern represented by NURBS model. Meanwhile, it is also regulated by the internal state vector \mathbf{p} , which consists of auto-regulation from itself and diffusion regulation from/to other robots or obstacles.

From the engineering control point of view, the proposed GRN model can be illustrated in Fig. 2, where it is a position control system with several intricately interplayed regulation terms: feedback regulation, feed forward regulation, auto regulation and diffusion regulation. It is this intricate interplay of different regulations in the GRN model that makes the proposed model to be robust to both internal and external system perturbations.

Different to the potential-field based gradient-following algorithms, in which the gradient information is passed to the robot only once, in our GRN-based model, the morphogen gradient will regulate two types of proteins (i.e., G and P) simultaneously, as shown in Fig. 1. This coherent regulation is helpful for preventing the robots from being trapped in local optima. As we know that local optimal are the points that make either Eqn. (3) or (4) equal to zero. In our model, even when either Eqn. (3) or (4) equals zero, the other equation will still enable the robot to change its dynamics, which get the robot move out of the local optimum. Note, however, the problem of getting trapped in a local optimum has not been fully resolved in the proposed model.

Nevertheless, the probability of a robot being getting trapped in a local optimum can be reduced significantly in this model.

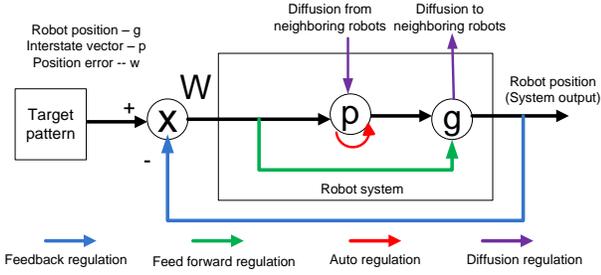


Fig. 2 The GRN-based model from engineering perspective

So far, we have introduced the morphogenetic approach for swarm robots pattern formation. Two important questions remain to be answered. First, how to guarantee that all robots will converge to the target pattern given the dynamics described in Eqns. (3) and (4)? Second, how do we optimize the local behaviors of each robot to achieve optimal global behavior? We will address these two issues in the next sections.

3.5. Theoretical Analysis of System Convergence

In this subsection, we will analyze the dynamics of the GRN-based model and show theoretically that robots driven by the dynamics described in Eqns. (3) and (4) will move to the target pattern defined by the NURBS model and stabilize there.

As clarity, we will first neglect the diffusion term in Eqns. (4) in the following proof. We will come back to this point later. By neglecting the diffusion term, Eqns. (3) and (4) can be rewritten as follows:

$$\frac{dg_x}{dt} = -aw_x + mp_x \quad (11)$$

$$\frac{dg_y}{dt} = -aw_y + mp_y$$

$$\frac{dp_x}{dt} = -cp_x + kf(w_x) \quad (12)$$

$$\frac{dp_y}{dt} = -cp_y + kf(w_y)$$

Here, we also omit the subscript i since it is a homogeneous system and all robots share the same dynamics.

We will introduce the following lemma before giving the main result on the system convergence.

Lemma 1: For the sigmoid activation function defined in Eqn. (8), $|f(x)| \leq |x|$ holds for all x .

The proof is straightforward and thus omitted here.

Theorem 1: The system states g_x and g_y in Eqn. (11) will converge to the target pattern defined by the NURBS model and the states p_x and p_y in Eqn. (12) will converge to zero, provided that $m \cdot k \leq a \cdot c$ and $k, c, a, m > 0$.

Proof: According to Invariant Set Theory [43], we can conclude that the system defined by Eqns. (11) and (12) is asymptotically stable if we can find a scalar function $V(g_x, g_y, p_x, p_y)$ that satisfies the following conditions:

(1) $V(g_x, g_y, p_x, p_y)$ is positive definite;

(2) $\dot{V}(g_x, g_y, p_x, p_y)$ is negative semi-definite;

(3) The set of points that satisfy the condition $\frac{dV}{dt} = 0$

are all on the target pattern defined by the NURBS model.

The four parameters, a , m , c , and k in the GRN-based model should all be positive for the system to be stable. The following scalar function can be defined:

$$V(g_x, g_y, p_x, p_y, s) = \frac{1}{2}w_x^2 + \frac{1}{2}w_y^2 + \frac{1}{2}s \cdot p_x^2 + \frac{1}{2}s \cdot p_y^2 \quad (13)$$

where $s = \frac{a \cdot c}{k^2} > 0$. From Eqn. (9), we know

that $w_x = g_x - c_x(u)$ and $w_y = g_y - c_y(u)$ and. Although u may change occasionally to satisfy Eqn. (10) during the convergence, it is nevertheless a time-invariant constant. Therefore, we can assume that $d(c_x(u))/dt = 0$ and $d(c_y(u))/dt = 0$. Then we have

$$\frac{dw_x}{dt} = \frac{dg_x}{dt} \quad \text{and} \quad \frac{dw_y}{dt} = \frac{dg_y}{dt} \quad (14)$$

Now we will follow the steps of Invariant Set Theory to prove that the system will converge asymptotically.

(1) $V(s) \geq 0$, which is obvious since $s > 0$

$$\begin{aligned} (2) \quad \frac{dV}{dt} &= w_x \frac{dw_x}{dt} + w_y \frac{dw_y}{dt} + s \cdot p_x \cdot \frac{dp_x}{dt} + s \cdot p_y \cdot \frac{dp_y}{dt} \\ &= w_x \frac{dg_x}{dt} + w_y \frac{dg_y}{dt} + s \cdot p_x \cdot \frac{dp_x}{dt} + s \cdot p_y \cdot \frac{dp_y}{dt} \\ &= w_x \cdot (-a \cdot w_x + m \cdot p_x) + w_y \cdot (-a \cdot w_y + m \cdot p_y) \\ &\quad + s \cdot p_x \cdot (-c \cdot p_x + k \cdot f(w_x)) + s \cdot p_y \cdot (-c \cdot p_y + k \cdot f(w_y)) \\ &= -aw_x^2 - aw_y^2 - c \cdot s \cdot p_x^2 - c \cdot s \cdot p_y^2 + m \cdot p_x \cdot w_x \\ &\quad + m \cdot p_y \cdot w_y + k \cdot s \cdot p_x \cdot f(w_x) + k \cdot s \cdot p_y \cdot f(w_y) \\ &\leq -a \cdot w_x^2 - a \cdot w_y^2 - c \cdot s \cdot p_x^2 - c \cdot s \cdot p_y^2 + m \cdot |p_x| \cdot |w_x| \\ &\quad + m \cdot |p_y| \cdot |w_y| + k \cdot s \cdot |p_x| \cdot |f(w_x)| + k \cdot s \cdot |p_y| \cdot |f(w_y)| \\ &\leq -\left(\sqrt{a}w_x - \sqrt{c \cdot s} \cdot p_x\right)^2 - \left(\sqrt{a}w_y - \sqrt{c \cdot s} \cdot p_y\right)^2 \\ &\quad - 2\sqrt{a \cdot c \cdot s} \cdot |w_x| \cdot |p_x| - 2\sqrt{a \cdot c \cdot s} \cdot |w_y| \cdot |p_y| \\ &\quad + (m + k \cdot s) \cdot |p_x| \cdot |w_x| + (m + k \cdot s) \cdot |p_y| \cdot |w_y| \end{aligned}$$

$$\begin{aligned}
&= -\left(\sqrt{a} \cdot w_x - \sqrt{c \cdot s} \cdot p_x\right)^2 - \left(\sqrt{a} \cdot w_y - \sqrt{c \cdot s} \cdot p_y\right)^2 \\
&+ \left(k \cdot s - 2\sqrt{a \cdot c} \cdot s + m\right) \cdot \left(\left|p_x\right| / w_x + \left|p_y\right| / w_y\right) \\
&= -\left(\sqrt{a} \cdot w_x - \sqrt{c \cdot s} \cdot p_x\right)^2 - \left(\sqrt{a} \cdot w_y - \sqrt{c \cdot s} \cdot p_y\right)^2 \\
&+ k \cdot \left[\left(\sqrt{s} - \frac{\sqrt{a \cdot c}}{k} \right)^2 - \frac{a \cdot c}{k^2} + \frac{m}{k} \right] \cdot \left(\left|p_x\right| / w_x + \left|p_y\right| / w_y\right)
\end{aligned}$$

Since $s = \frac{a \cdot c}{k^2}$, we can get

$$\begin{aligned}
\frac{dV}{dt} &\leq -\left(\sqrt{a} \cdot w_x - \sqrt{c \cdot s} \cdot p_x\right)^2 - \left(\sqrt{a} \cdot w_y - \sqrt{c \cdot s} \cdot p_y\right)^2 \\
&+ k \cdot \left(-\frac{a \cdot c}{k^2} + \frac{m}{k} \right) \cdot \left(\left|p_x\right| / w_x + \left|p_y\right| / w_y\right)
\end{aligned}$$

If $-\frac{a \cdot c}{k^2} + \frac{m}{k} \leq 0$, i.e., $m \cdot k \leq a \cdot c$, then $\frac{dV}{dt} \leq 0$.

Now we need to verify that the states in Eqn. (11) will converge to the target pattern, provided that $\frac{dV}{dt} = 0$ holds.

(3) When $\frac{dV}{dt} = 0$, all the ‘less than’ conditions should satisfy the ‘equal to’ condition, which means that $|f(w)| \leq |w|$ becomes $|f(w)| = |w|$. From **Lemma 1**, $|f(w)| = |w|$ if and only if $w = 0$. Therefore, if $\frac{dV}{dt} = 0$, we have $w_x = 0$ and $w_y = 0$ which means that $g_x - g_x(u) = 0$ and $g_y - g_y(u) = 0$, where u is a constant ranging from 0 to 1. In other words, $g_x = g_x(u)$ and $g_y = g_y(u)$ hold when the system is stable. Since (g_x, g_y) denotes the position of the robots, and $(g_x(u), g_y(u))$ defines a certain point on the target pattern, we can conclude that when $\frac{dV}{dt} = 0$, all robots driven by the regulatory dynamics described by Eqns. (11) and (12) will converge to the target pattern. This completes the proof.

According to the above theorem, we show that all robots will converge to the target pattern, provided that $m \cdot k \leq a \cdot c$ and $k, c, a, m > 0$.

The above proof was given by omitting the diffusion term in Eqn. (4). We now explain that this omission does not affect the convergence proof if the neighborhood of the robots is defined such that $d_n \leq T$, where d_n is the neighborhood size, T is a threshold, which is defined as $T = L/n$ for closed-form shapes and $T = (L-1)/n$ for open-form shapes, where L is the total length of the periphery of the target shape, and n is the number of robots. We can distinguish three cases under this condition. In the first case, the distance between all

neighboring robots on the target shape is larger than d_n , then the diffusion term defined in Eqn. (5) equals zero automatically. In the case that the initial distance between some of the neighboring robots on the target shape is smaller than the neighborhood size, these two robots will push away from each other driven by the diffusion term until the distance is equal to or greater than the neighborhood size. Then the diffusion term will disappear too. Ideally, if $d_n = T$, then the robots will distribute evenly on the target shape while the convergence condition is not violated, but this statement holds only for those shapes that consist of linear curves. For shapes that consist of non-linear curves, d_n is actually greater than T , which belongs to the third case. Finally, if $d_n > T$, which is the third case, then some of the robots may be pushed away from the target shape until the diffusion term equals zero. Note however, that a neighborhood adaptation has been proposed, which is able to adapt the neighborhood size such that d_n will be smaller than or equal to T .

According to the analysis, the neighborhood size is critical to the system convergence. Therefore, we propose a neighborhood adaptation mechanism (will be discussed next), which will ensure the robots to reach an optimal neighborhood.

3.6. Parameter Selection through Multi-Objective Optimization

The proof in the previous section indicates under which condition the robots converge to the target pattern. In this section, we will further optimize the system performance by tuning the parameters in the regulatory model. In this work, we consider two performance indices, namely, the total traveling distance of all robots and the time for the whole system to converge to the target pattern. This can be seen as a multi-objective optimization (MOO) problem, where the objective function is no longer a scalar value, but a vector. As a consequence, a number of Pareto-optimal solutions should be achieved instead of one single optimal solution.

In this paper, NSGA-II [44], which is a popular and efficient evolutionary algorithm for solving multi-objective optimization problems, has been adopted for optimizing the parameters of the GRN-based self-organization model. Simulated binary crossover (SBX) [45-46] and polynomial mutation [44] have been employed to generate offspring. After the offspring population is generated, the elitist non-dominated sorting algorithm is used for selecting parents for the next generation. As a result, a set of Pareto-optimal solutions that tradeoff between the convergence time and the travel distance of the robots have been achieved. It should be pointed out that by using NSGA-II, which is a stochastic search algorithm, we cannot guarantee that we can always find the global Pareto-optimal solutions. Fortunately, our problem is a small optimization problem with 5 parameters that are well defined to satisfy the convergence condition. Meanwhile, it is also sufficient for us even if we achieve a set of local Pareto-optimal solutions.

3.7. Neighborhood Adaptation Mechanism

We have assumed that the total number of robots in the system is unknown to the robots. In order to deploy the robots on the boundary of the target pattern as evenly as possible, an algorithm for adapting the local neighborhood size is needed.

Since the target pattern is given, the length of the perimeter of the target pattern can be calculated, which is notated as L . The adaptation process starts with an initial neighborhood size d_0 . Theoretically, the needed number of robots is $n' = L/d_0$ to cover the boundary evenly. With this estimated number of robots, n' points can be generated on the target pattern according to the NURBS. The robots will then move to the generated points autonomously driven by the GRN dynamics. If d_0 is too small, i.e., the number of generated desired points is larger than the number of robots, some of the points will not be covered by any robot. As a result, some of the robots will detect that they have only one neighboring robot within its neighborhood. In this case, neighborhood size d_0 should be increased.

On the other hand, if d_0 is too large, i.e., the number of generated desired points is smaller than the number of robots, there must be some robots competing for the same point on the target pattern, resulting in a large positioning error, which can be detected by comparing the real and the desired positions of the robot. Here, the desired position can be calculated from the parameterized target pattern representation and the real position can be estimated from the self-localization mechanism of the robot.

A robot that either has insufficient number of neighbors or has a large positioning error will send a signal to its neighbors (and its neighbors will send this information to their neighbors, etc.) so that all robots know that an adaptation of the neighborhood size is necessary. Note that the neighborhood size should be limited by two physical parameters of the robot, i.e., the bumper range and the sensor range.

The major steps of the neighborhood size adaptation are listed as follows.

- 1) Initialization: d_{min} and d_{max} are initialized as the bumper range and sensor detection range of a robot, respectively, and the initial neighborhood size is estimated by

$$d_0 = \frac{d_{min} + d_{max}}{2} \quad (15)$$

- 2) If d_0 is too small, we keep d_{max} the same and update d_{min} and d_0 as follows:

$$d_{min} = d_0 \quad (16)$$

$$d_0 = \frac{1}{2}(d_0 + d_{max}) \quad (17)$$

- 3) If d_0 is too large, we keep d_{min} the same and update d_{max} and d_0 as follows:

$$d_{max} = d_0 \quad (18)$$

$$d_0 = \frac{1}{2}(d_0 + d_{min}) \quad (19)$$

- 4) Repeat step (2) or (3) until the optimal neighborhood size is found.

To save the computational cost, when

$$|d_0 - d^*| \leq \varepsilon, \quad (20)$$

we assume that a robot has found the optimal neighborhood and does not need to update d_0 anymore, where ε is a predefined threshold.

From the adaptation process, we can see that the system requires communications among the robots. Furthermore, in the adaptation mechanism, a robot only needs to communicate with its neighbors when it realizes that d_0 needs to be updated (either the robot has insufficient number of neighbors or it has a large positioning error). Therefore, the number of iterations that robots need to update d_0 can be used as the metric to measure the communication load.

The proposed neighborhood adaptation mechanism is a typical searching problem. The largest number of communication iteration needed between robots to find the optimal neighborhood can be estimated by:

$$n = \left\lfloor \log_2 \left(\frac{d_{max} - d_{min}}{\varepsilon} \right) \right\rfloor \quad (21)$$

where $\lfloor x \rfloor$ denotes the floor of x (i.e., the largest integer less than or equal to x).

The probability that a robot needs k ($0 \leq k \leq n$) communication iteration with others to find the optimal neighborhood range is:

Case 1: when $k = 0$, which means that d^* is within the range of $[d_0 - d^*, d_0 + d^*]$, so the probability is

$$P_k = \frac{2\varepsilon}{d_{max} - d_{min}} \quad (22)$$

Case 2: when $0 < k < n$, every time d_0 is updated and a new search is started, the search space will shrink by half. Thus, the probability at which d^* is found on this iteration will be twice that of the previous iteration, but this is a conditional probability given that d^* has not been found in the previous $(k-1)$ iterations, so the probability will be:

$$P_k = \left(1 - \sum_{i=0}^{k-1} P_i\right) \cdot \frac{2^{k+1} \cdot \varepsilon}{d_{max} - d_{min}} \quad (23)$$

Case 3: when $k = n$, d^* has been found, so the probability is equal to the probability at which d^* has not been found in the previous $n-1$ iteration:

$$P_k = 1 - \sum_{i=0}^{k-1} P_i \quad (24)$$

Therefore, the expected number of updates can be expressed as:

$$E = \sum_{k=1}^n kP_k \quad (25)$$

Due to the complexity of Eqns. (23) and (24), it is hard to provide a simple analytical function of the expected number of updates. We will give an example to show the number of needed update iterations for the robots to reach d^* . Here, we shall clarify that each time a robot updates d_0 , all other robots will perform the same update so that all robots keep the same d_0 . In this way, all the robots in the system have the same communication load. The system's communication load can be gauged as n times of individual robot's communication load, where n is the number of robots in the system.

Suppose that $d_{\min} = 2$, $d_{\max} = 200$, and $\varepsilon = 0.5$, 30000 independent experimental runs have been conducted, where d^* is set to be a random variable with a uniform distribution between d_{\min} and d_{\max} . In this case study, the maximum number n required for updates can be estimated by Eqn. (21) as:

$$n = \left\lceil \log_2 \left(\frac{d_{\max} - d_{\min}}{\varepsilon} \right) \right\rceil = 8.$$

Fig. 3 shows the distribution of the number of updates needed. From Fig. 3, we can see that after 5 or 6 iterations of adjustments of d_0 , the robots will reach the expected optimal neighborhood size. In other words, the communication load for this procedure is acceptable. Furthermore, from Eqns. (22) to (25), we can see that each robot's communication load is independent of the number of robots in the system. The overall communication load of the system will be increased proportionally to the number of robots in the system. In other words, the communication load is scalable to large-scale systems.

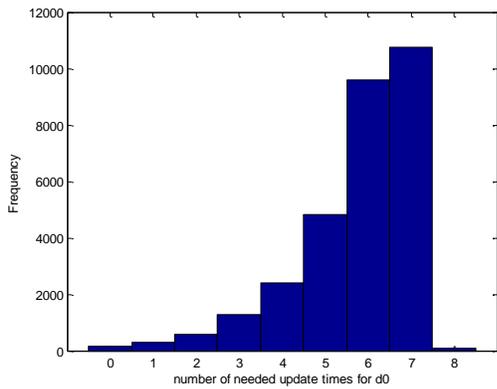


Fig.3. The histogram of the number of iterations needed to achieve the optimal neighborhood size with the mean value of 5.7576 and the standard deviation of 1.3903.

Please be noted that as a robot moves around, the number of neighbors of the robot is changing. If there are n robots in the system, for any robot, it can have up to $(n-1)$ neighbors. The probability of any two robots being neighbors can be

defined as $p = \frac{\pi d_0^2}{S}$, where d_0 is the robot's neighbor range

and S is the area size of the environment. Therefore, the expected number of neighbors of a robot can be estimated by

$$E = (n-1) \cdot \frac{\pi d_0^2}{S}.$$

3.8. Computational Complexity

Since each robot needs to calculate two differential equations (i.e., Eqns. (3) and (4)) independently, the computational complexity of the proposed model does not change as the number of robots increases. Therefore, the general computational complexity of the GRN-based controller is $O(1)$.

However, there are two other factors that will affect the computational complexity of the proposed algorithm. As the number of neighbors of a robot increases, the computation complexity will increase accordingly because it needs to sum up the diffused protein values from those neighbors. The complexity can be represented as $O(q)$, where q is the number of neighbors of a robot. As we mentioned in the last sub-section, q is changing as the robot moves around.

Another term that will affect the complexity of the framework is the number of iterations needed for the neighborhood adaptation mechanism. As we have explained in Section 3, we used an efficient searching algorithm to reduce the number of communication iterations. The largest iterations needed for the adaptation mechanism is $\lceil \log_2(d_{\max} - d_{\min}) \rceil$, where $\lceil x \rceil$ represents the ceiling of x (i.e., the largest integer less than or equal to x). d_{\min} and d_{\max} are the bumper range and sensor detection range of the robot, respectively. The complexity of this term can be represented as $O(1 + \lceil \log_2(d_{\max} - d_{\min}) \rceil / 2)$. Thus, the overall complexity of the presented algorithm for each robot is $O(1) + O(q) + O(1 + \lceil \log_2(d_{\max} - d_{\min}) \rceil / 2)$.

4. Simulation Results and Analysis

4.1. Parameters Setup using NSGA-II

To evaluate the effectiveness and robustness of the proposed method, we conducted a sequence of case studies using MATLAB. Five parameters, a , m , c , k , and b in Eqns. (3) and (4), are optimized using NSGA-II. The goal of the optimization is to minimize the travel distance of robots and the convergence time while ensuring the system convergence, as discussed in the previous section.

In our simulation, the number of robots is 20 and the target pattern is the capital letter "R". The population size of NSGA-II is 100. The crossover probability is 0.9 and the distribution index for SBX is 20. Mutation probability is defined to be inversely proportional to the number of the decision variables, which is 5 in our case, therefore, the mutation probability is 0.2 and the distribution index for mutation is 20. The above parameter setup is chosen as recommended in [44]. The evolution runs for 50 generations.

Parameters k , c , a , and m are randomly initialized between 1 and 100 and b is randomly initialized between 200 and 1000. Here, we predefine a large value for the diffusion constant b to allow the robots to move far away from each other so that they can select different target positions when robots are close to each other.

Since the final goal of the swarm robotic system is to distribute the robots onto the target pattern, the position error to the target pattern should be as small as possible after convergence. Therefore, we define a threshold for the average position error between the robots' final position and the target pattern as a constraint of this optimization problem when using NSGA-II. In the following experiments, we set this threshold to be T , and T is defined as 0.05 for the following case studies.

The Pareto-optimal solutions achieved by NSGA-II are plotted in Fig. 4, where there are largely three groups of Pareto-optimal solutions. The optimized parameters of a typical solution in each of the three groups (i.e., those solutions indicated by the arrows) are listed in Table I. The unit for convergence time is second and the unit for travel distance is meter. This unit definition is applicable to all the following simulation results.

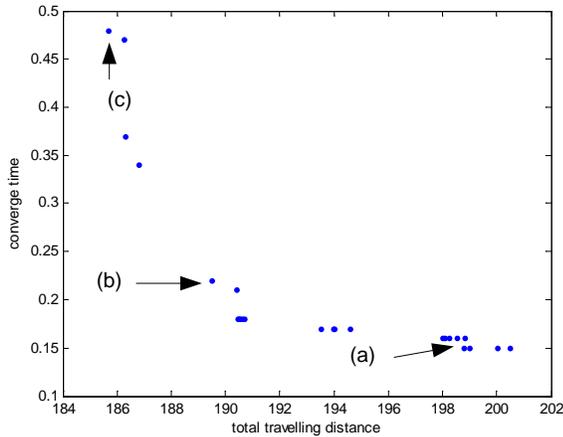


Fig. 4. The Pareto-optimal solutions achieved by NSGA-II. The unit for travel distance is meter and the unit for convergence time is second. (Note: each arrow indicates a single solution it is pointing to.)

From Fig. 4, we can see that if we want to minimize the total travel distance, we have to sacrifice the system convergence time, and vice versa. This is always true from the control point of view, where the system's response time and convergence time always conflict with each other. The Pareto-optimal solutions depicted in Fig. 4 are helpful for users to design the system parameters based on their specific requirements. For example, if the user concerns more about the convergence time, the Pareto-optimal solutions close to solution (a) is preferred. If the user wants to have a shorter travel distance to save energy, it is better to pick the Pareto-optimal solutions close to solution (c). Without loss of generality, we pick solution (b) for the following case studies, which has a good balance between the total travel distance and convergence time. The parameters of solution (b)

are used in the following simulations unless otherwise specified.

TABLE I
THE PARETO-OPTIMAL SOLUTIONS

	k	c	B	a	m	td (m)	ct (s)
(a)	45.88	69.13	387.5	69.28	63.77	198.76	0.15
(b)	79.46	70.06	420.8	44.39	6.85	189.49	0.22
(c)	91.81	57.65	599.9	16.04	1.00	185.67	0.48

Note: td refers to travel distance and ct refers to the convergence time.

4.2. Formation of 2D/3D Shapes

First, we conduct a set of experiments in a 2D environment, as shown in Fig. 5. The simulation environment is a 44m x 44m square. The target pattern consists of five capital letters 'NURBS'. This shape information defined by NURBS is embedded in the GRN-based controller for each robot. A set of snapshots of the self-organization procedure using 56 robots for forming the five letters are shown Fig. 5. Additionally, we conduct a set of experiments in a 3D environment of 10m x 10m x 10m, as shown in Fig. 6 and Fig. 7, to form 3D curves with 12 robots and 3D surfaces with 24 robots, respectively. For the 3D simulations, we add $g_{i,z}$ and $p_{i,z}$ to the GRN-based model, which are governed by the same update law as $g_{i,x}$ and $p_{i,x}$.

In the case that the target pattern is a 3D curve, the robots form a linear curve in a 3D space starting from their random initial positions. Then, an ellipse is formed by the robots starting from the linear curve. Both 3D curves are defined using NURBS. In the case where the target pattern is a 3D surface, the robots first construct three parallel circles, and then two parallel squares, as shown in Fig. 7.

35 independent runs have been performed for each case. The mean and standard deviation of the convergence time, the average travel distance, and the average position error are listed in Table II. The average position error is defined as the average shortest distance between the final positions of the robots to the desired shape. The average travel distance is defined as the average travel distance of robots from the initial position to the final position.

It can be observed from Table II that the convergence time and the average travel distance for the "NURBS" target pattern are much larger than that for the other two target patterns. The main reason for this is that the number of robots for forming "NURBS" is much larger than the other two cases, where a larger number of robots may compete for the same location on the target pattern, and therefore leading to a longer travel distance and a larger convergence time. However, this observation does not affect the scalability of the proposed GRN-based model since the computational cost for each individual robot does not increase exponentially with the number of robots. Even in the cases with robot failures, although an increase in the number of robots will increase the communication load proportionally in each robot,

and therefore a longer system convergence time, it won't affect the system scalability.

It can be seen from Fig. 5, Fig. 6, Fig. 7 and Table II that a swarm of robots can form a variety of 2D/3D complex shapes without a centralized. These results demonstrate the effectiveness of the proposed GRN-based model for multi-robot shape formation.

4.3. Robustness to Sensor Noise and Localization Error

In this case study, we will evaluate the robustness performance of the proposed method to the sensory noise and localization errors. First, we perform 35 independent runs with 10 robots randomly initialized in a 10x10 environment. The target pattern is a unit circle being placed in the center of the environment. The final position errors with the robot sensory noise and localization errors are listed in Tables III and IV, respectively. Here, if a sensory data is x , 5% noise means that we will randomly pick a number from $[(1-5\%)x, (1+5\%)x]$ as the current sensory data for the system.

TABLE III
MEAN AND STANDARD DEVIATION OF THE CONVERGENCE TIME AND POSITION ERRORS WHEN THE DISTANCE MEASUREMENT ARE SUBJECT TO SENSORY NOISE

Without noise	5% noise	10% noise
Mean: 0.0421	Mean: 0.0459	Mean: 0.0464
STD: 0.024	STD: 0.0104	STD: 0.0109

TABLE IV
MEAN AND STANDARD DEVIATION OF THE CONVERGENCE TIME AND POSITION ERRORS WHEN THE ROBOT LOCALIZATION IS SUBJECT TO SENSORY NOISE

Without noise	5% noise	10% noise
Mean: 0.0421	Mean: 0.0455	Mean: 0.0482
STD: 0.024	STD: 0.0082	STD: 0.0097

It is well known that the localization errors will be accumulated over time using the odometry method. This accumulated localization error can be reduced by using the Kalman filter method we proposed in our previous work [35].

From Table III and Table VI, we can see that the position errors of the system using the proposed method only increase

slightly in the presence of various noise terms. Thus, we can draw the conclusion that the system is robust to the sensory noise and localization errors.

4.4. Robustness to Environmental Perturbations

To evaluate the system's robustness to external environmental perturbations, we implement the following experiment. A mobile obstacle approaches a swarm of robots that have formed the target pattern, for example, a letter "R", and then moves away from the robots. Fig. 8 shows a set of snapshots of this scenario. It can be seen that the robots can autonomously avoid the mobile obstacle, and re-organize themselves after the obstacle moves away. No explicit obstacle avoidance strategy is needed here since robots can detect the obstacle using their onboard sensors and avoid the obstacle through the diffusion term in Eqn. (4). Note, however, that the behavior of the obstacles cannot be influenced by the robots.

4.5. Robustness to Robot Failures

In this case study, we will show that robots can autonomously re-organize themselves to deploy on the boundary of the target pattern in case several robots fail. In the simulation, when some robots fail, they will be eliminated and only the functioning robots will be shown. Here, we only consider the case in which robots have formed the target shape and some robots fail. In this sense, the neighbors of each robot are fixed rather than dynamic. If the robots fail during the pattern formation process, we do not need the neighbor robots to report a missing neighbor. All the remaining robots will just continue to form the shape until they have formed the shape. After all of the remaining robots have formed the shape, some of the robots either report a missing neighbor (d_0 needs to be increased) or report a large position error (d_0 needs to be increased), then the neighborhood adaptation mechanism is started.

When a robot fails, its neighbor(s) will not be able to communicate with the failed robot and consequently find out that the neighbor robot fails. These neighboring robots will pass this message to their neighbors and initiate the neighborhood size adaptation mechanism. The self-reorganization process after four robots fail is illustrated in Fig. 9.

TABLE II
THE STATISTICS DATA FOR CONVERGENCE TIME, AVERAGE TRAVEL DISTANCE, AND AVERAGE POSITION ERROR

	"NURBS" (mean \pm std)	3D curves (mean \pm std)	3D surfaces (mean \pm std)
Convergence time	1823.50 \pm 305.64	252.77 \pm 74.05	197.48 \pm 35.42
Average travel distance	9.8096 \pm 1.2232	0.7726 \pm 0.0955	1.3478 \pm 0.1202
Average position error	0.0160 \pm 0.0033	0.0214 \pm 0.0059	0.0209 \pm 0.0019

Note: In the 3D curves column, the data is only for the period from the initial state to the formation of the 3D line in Fig. 4. In 3D surfaces, the data is only for the period from the initial state to the formation of the three parallel circles in Fig. 5. (Units: distance in meters and time in seconds)

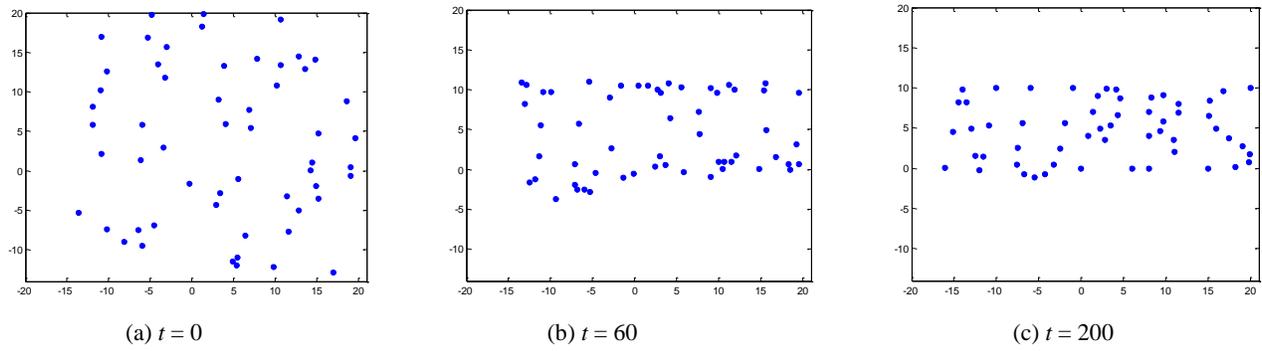


Fig. 5. A set of snapshots of 56 robots to self-organize different 2D letters “N” “U” “R” “B” “S”, where t represents the time step running in the Matlab, not in the unit of second. The video of this experiment can be downloaded from <http://www.ece.stevens-tech.edu/~ymeng/Projects.htm>.

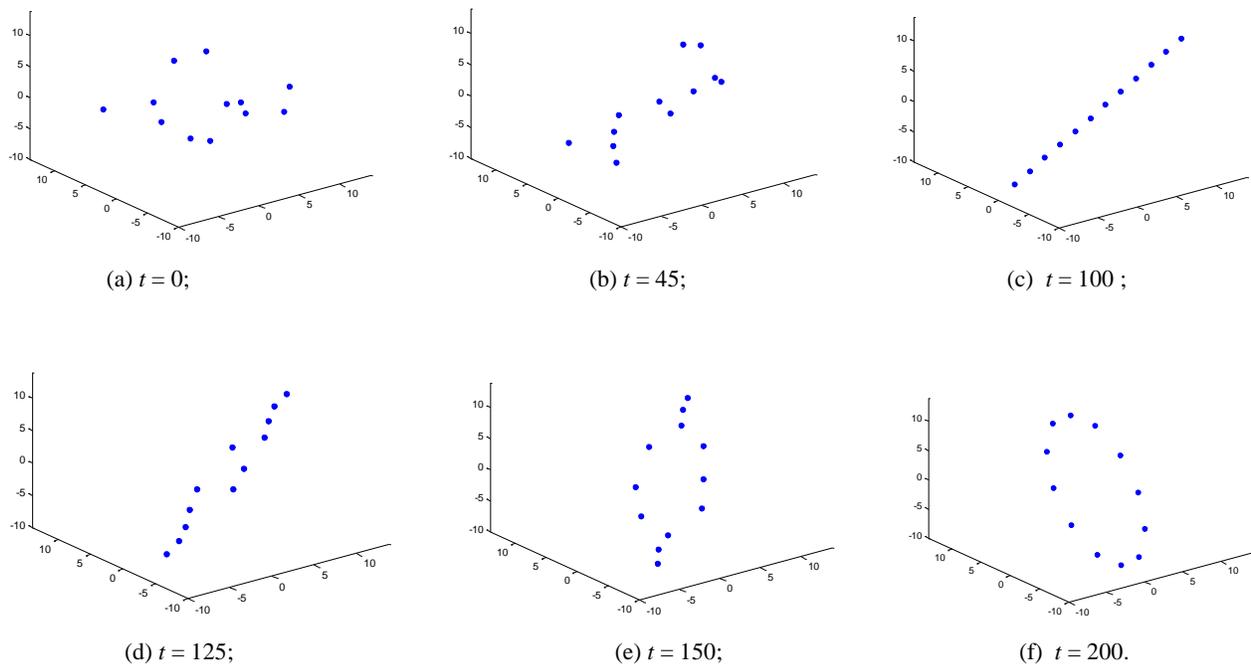
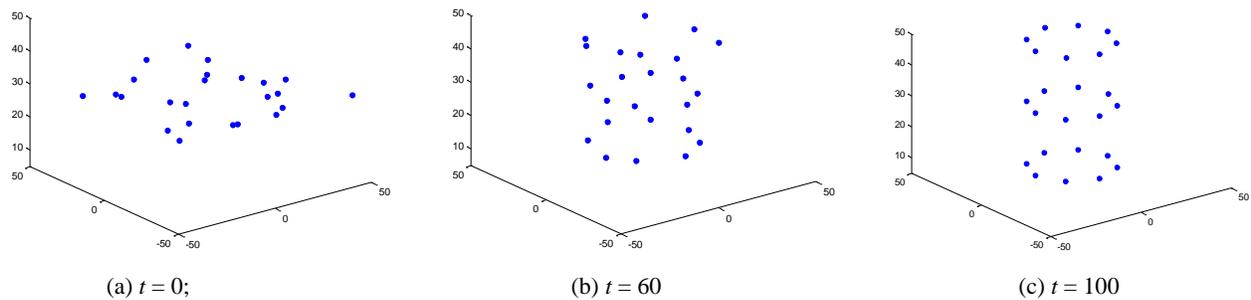


Fig.6. A set of snapshots of 12 robots self-constructing different 3D curves. The video of this experiment can be downloaded from <http://www.ece.stevens-tech.edu/~ymeng/projects.htm>.



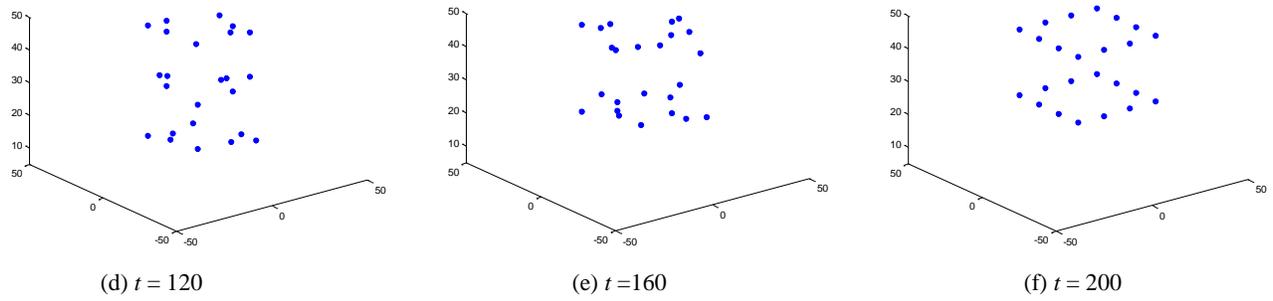


Fig.7. A set of snapshots of 24 robots self-constructing different 3D surfaces.

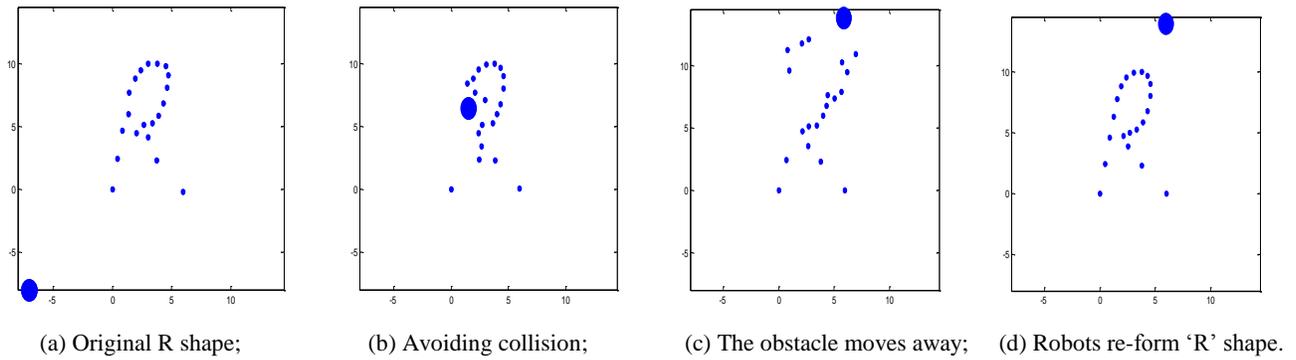


Fig. 8: Adaptation test with a mobile obstacle using 20 robots. Robots avoid a mobile obstacle and reorganize themselves to the original shape. Please be noted that only the procedure of mobile obstacle moving forward is shown in Fig.7 due to paper limitation. The video of this experiment can be downloaded from <http://www.ece.stevens-tech.edu/~ymeng/Projects.htm>.

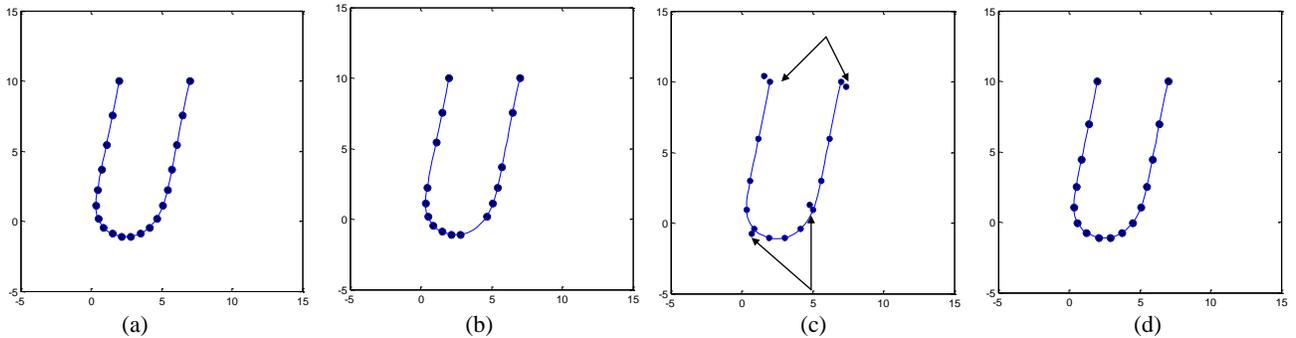


Fig.9. Re-organization process in the presence of robot failures. (a) 20 robots were distributed evenly on “U” curve. (b) Four robots failed, resulting in some gaps uncovered by robots. (c) The adaptation mechanism started to work, which first tried to increase the neighbor size. However, the neighborhood size is too large and therefore, three clusters of robots were formed (pointed by arrows). (d) The second iteration of adaptation was performed and no robot reported failed neighbors. After the re-organization, the remaining 16 robots were distributed again relatively evenly on the target curve.

5. Experimental Results using Physical Robots

To evaluate the proposed GRN-based model in real robotic systems, experiments have been performed for a swarm robotic system consisting of eight e-puck education robots (<http://www.e-puck.org/>). As shown in Fig. 10, each e-puck robot is approximately six centimeters in diameter with a circumferential ring of eight infrared proximity sensors, a pair of step motors in a differential-drive configuration, three microphones, and a ZigBee wireless communication card. Infrared proximity sensors are used for

distance detection. Microphones are used to trigger the start of the experiments, and wireless card is used for debugging and uploading the software on the e-puck robots. Robots do not explicitly communicate with each other. In the experiments, each robot is provided with a starting position in a global coordinate system and the description of the target pattern represented by NURBS model. Each robot has to decide autonomously to which point on the target pattern it

should approach using the GRN model based on its local interactions with the environment and other robots.

To implement the GRN-based model on the physical robots, we have to consider a few real-world constraints. First, it is assumed that robots are holonomic in the simulations. However, e-puck robots are differential-drive robots and non-holonomic. Second, self-localization of the robots in an indoor environment may become an issue.



Fig. 10: The e-puck educational robot.

The GRN-based model does not consider nonholonomic constraints imposed by the differential-drive robot. Therefore, there must be a translation between the desired motion of the GRN dynamics and the robot’s actual motion. For this proof-of-concept implementation, self-localization is performed by an open-loop estimation using an odometry method with the onboard encoders. As we know, the localization measurement errors using the odometry method may get accumulated over time. To mitigate errors in measurement of the e-puck’s geometry, a scaled version of the UMBMark [47] calibration procedure was performed on each e-puck robot.

Fig. 11 shows snapshots of the experimental results using 8 e-puck robots to form a letter “R” from random initial positions in an indoor environment. The parameters for the GRN-based model for this experiment are set as follows: $a = k = 0.1$, $c = 1.0$, $b = 20$ and $m = 1.0$. We did not measure the final position errors of robots to the patten “R” manually, but from Fig. 11, we can see that the robots move to the pattern as we expected. The major constraints on this experiment are a lack of a robust localization system and the use of short-range, non-uniform, and noisy proximity sensors. Given these constraints, the experiment was able to demonstrate a successful example of the shape formation using a swarm robotic system. While the localization system used was effective enough for a simple experiment with a limited duration, the lack of a robust localization scheme severely hampered the ability of the robots to perform more complex tasks.

The previous experiment demonstrates that robots can be deployed to the target pattern with physical constraints. However, in that case, we are focusing on the deployment process with the assumption that the initial neighborhood size d_0 is near the actual optimal neighborhood size d^* , thus the neighborhood adaptation mechanism does not function during the deployment process.

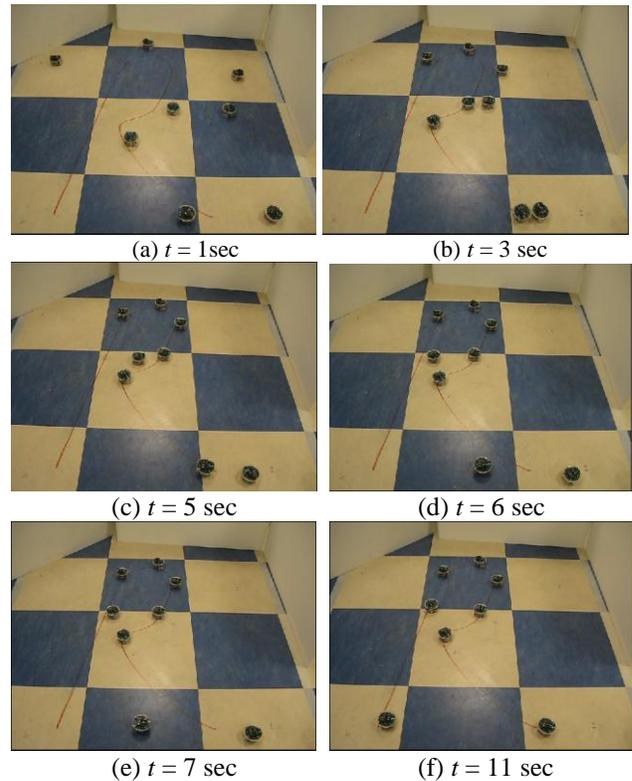


Fig. 11. Snapshots of the experiments showing 8 e-puck robots forming a letter “R” from random initial positions. The video of this experiment can be downloaded from <http://www.ece.stevens-tech.edu/~ymeng/Projects.htm>.

To verify the effectiveness of the neighborhood adaptation mechanism, another experiment is conducted. In this experiment, 6 e-puck robots have been used to form a circle. However, the initial neighborhood size d_0 is much smaller than the optimal neighborhood size d^* . Thus, robots need to adapt the neighborhood size several times to ensure an even distribution. Since this experiment focuses on the neighborhood adaptation mechanism, we select a relatively simple shape (a circle). Snapshots of this proof-of-concept experiment are shown in Fig. 12. From Fig. 12, it can be seen that the proposed extended GRN-based model can work efficiently with the neighborhood adaptation mechanism in a distributed manner.



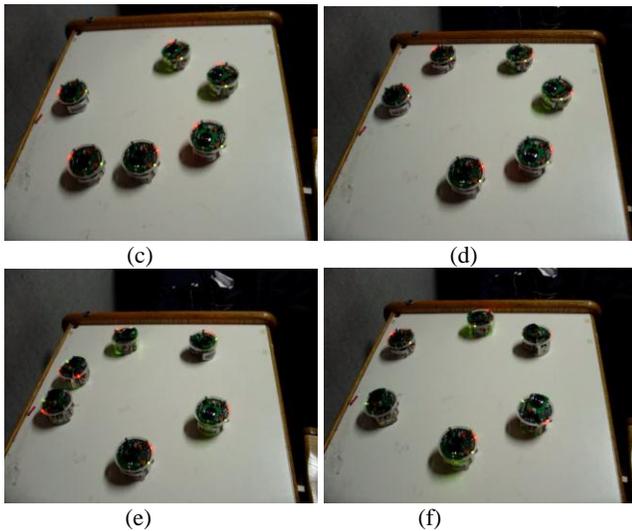


Fig. 12. Snapshots of the experiment. (a) The initial distribution of robots, (b) (c) The pattern (a circle) formation process. (d) After the formation, there is a vacant place, indicating that the neighborhood is shorter than expected. (e) The pattern formation after neighborhood adjustment, this time two robots are stuck together, which means that the neighborhood is larger than expected. (f) The pattern formation after another phase of neighborhood adjustment. Robots are distributed uniformly on the circle. The video of this experiment can be downloaded from <http://www.ece.stevens-tech.edu/~ymeng/Projects.htm>.

6. Conclusion and Future Work

In this paper, we have presented a novel GRN-based framework for a distributed swarm robotic system to construct complex shapes in a 2D or 3D environment. Simulation results show the effectiveness and robustness of the proposed model. Two proof-of-concept experiments using e-puck mobile robots demonstrated the feasibility and effectiveness of the proposed model with physical robotics constraints.

In the current model, one major limitation is that the system is working under a global coordinate system where the robots need to localize within this global coordinate system. As we mentioned in the robot experiment, the lack of a robust localization scheme may degrade the system performance. To address this issue, currently we are working on the system where only relative position is needed without a global coordinate system. Although the current model is adaptive to certain system or environmental changes, e.g., avoiding a mobile obstacle, the target pattern (relevant to the mission the swarm robotic system should accomplish) has to be defined in advance by the user. In real-world applications, the multi-robot system needs to adapt its mission autonomously if the environment changes drastically. In the future, we will work on designing a gene regulatory model that is able to generate a target pattern online and then drive the robots to the target pattern.

ACKNOWLEDGMENT

This project is partially supported by the Honda Research

Institute Europe GmbH, Germany. The authors would like to thank Louis Simons for writing the software program and helping debug the e-puck robots implementations for the experiments.

REFERENCES

- [1] A.Tiderko, T.Bachran, F.Hoeller, D.Schulz, "RoSe — A framework for multicast communication via unreliable networks in multi-robot systems", *Robotics and Autonomous Systems*, Volume 56, Issue 12, 31 December 2008, pp: 1017-1026
- [2] C. Clark "Probabilistic Road Map sampling strategies for multi-robot motion planning", *Robotics and Autonomous Systems*, Volume 53 Issue 3-4, December, 2005. pp: 244—264
- [3] W. Sheng, Q.Yang, J. Tan and N.Xi, "Distributed multi-robot coordination in area exploration", *Robotics and Autonomous Systems*, Volume 54 Issue 12, December, 2006. pp:945-955
- [4] R. Rocha, J. Dias, A. Carvalho, "Cooperative multi-robot systems:: A study of vision-based 3-D mapping using information theory", *Robotics and Autonomous Systems*, Volume 53 Issue 3-4, December, 2005. pp: 282—311
- [5] Y. Chen; Y.Tian. A backstepping design for directed formation control of three-coleader agents in the plane *International Journal of Robust and Nonlinear Control* May 2009. Vol.19,Iss.7; p.729-745.
- [6] J.Choi; Y.Kim Fuel efficient three dimensional controller for leader-follower UAV formation flight;. *International Conference on Control, Automation and Systems, (ICCAS) 2007*. Oct. 2007. pp. 806-811.
- [7] A. Jadbabaie, J. Lin, and A. S. Morse, Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp: 988-1001, 2003.
- [8] K. Kanjanawanishkul and A. Zell; A model-predictive approach to formation control of omnidirectional mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008*. Sept. 2008. pp: 2771-2776.
- [9] I. Kelly; A. Martinoli. A scalable, on-board localization and communication system for indoor multi-robot experiments *Sensor Review*; Bradford, 2004. Vol.24, Iss.2; pp: 167-182
- [10] J. Shao, G. Xie, J. Yu, and L. Wang, A tracking controller for motion coordination of multiple mobile robots *IEEE/RSJ International Conference on Intelligent Robots and Systems, (IROS 2005)* Aug. 2005. p.783-788.
- [11] N. Boonpinon; Sudsang, A Heterogeneity driven circular formation. *IEEE International Conference on Robotics and Biomimetics, ROBIO 2006*. pp: 971-976.
- [12] S.W. Ekanayake and P. N. Pathirana, Geometric formations in swarm aggregation: An artificial formation force based approach, *Proceedings of the 3rd International Conference on Information and Automation for Sustainability*, 2007. pp: 82-87.
- [13] M. A. Hsieh and V. Kumar, "Pattern Generation with Multiple Robots," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, Orlando, Florida-May 2006.
- [14] M. Hsieh, V. Kumar and L. Chaimowicz, Decentralized controllers for shape generation with robotic swarms, *Robotica*, Volume 26, Issue 5, pp: 691-701, 2008.
- [15] S. Kalantar and R. Zimmer. Distributed Shape Control of Homogeneous Swarms of Autonomous Underwater Vehicles, *Autonomous Robots 2006*
- [16] S. Kalantar and R. Zimmer. Control of Open Contour Formations of Autonomous Underwater Vehicles, *International Journal of Advanced Robotic Systems*, Dec. '05
- [17] C. Mai and F. Lian, Analysis of Formation Control and Communication Pattern in Multi-Robot Systems;. *SICE-ICASE International Joint Conference (89-950038-4-7)* Oct. 2006. p.640-645
- [18] A. Masoud, "Decentralized self-organizing potential field-based control for individually motivated mobile agents in a cluttered environment: a vector-harmonic potential field approach", *IEEE Trans. on Systems, Man, and Cybernetics – Part A: Systems and Humans*, Vol. 37, No. 3, May 2007. pp. 372-390.

- [19] M. Willam, R. Heil and D. Zarzhitsky, "Artificial physics for Mobile Robot Formations." *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp. 2287-2292, 2005.
- [20] L. Whitcomb and D. E. Koditschek, "Automatic assembly planning and control via potential functions", *IEEE/RSJ International Workshops on Intelligent Robots and Systems (IROS 91)*, Nov. 3-5, 1991, Osaka, Japan.
- [21] K.N. Krishnanand, D.Ghose, "Formations of minimalist mobile robots using local-templates and spatially distributed interactions" *Robotics and Autonomous Systems*, Volume 53 Issue 3-4, December, 2005. pp: 194-213
- [22] W. Shen, P. Will and A. Galstyan, "Hormone-Inspired Self-Organization and Distributed Control of Robotic Swarms", *Autonomous Robots*, 17, pp.93-105, 2004.
- [23] J. Stradner, H. Hamann, T. Schmickl, and K. Crailsheim, Analysis and implementation of an artificial homeostatic hormone system: a first case study in robotic hardware. *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009*.
- [24] R. Bloom, C. Chang, and A. Kondacs, "Compilation and biological-inspired self-assembly of two-dimensional shapes", *International Joint Conference on Artificial Intelligence, 2003*.
- [25] J. Cheng, W. Cheng, and R. Nagpal, "Robust and self-repairing formation control for swarms of mobile agents", *AAAI 2005*.
- [26] M. Mamei, M. Vasirani, F. Zambonelli, "Experiments in morphogenesis in swarms of simple mobile robots", *Applied Artificial Intelligence*, 18, 9-10: 903-919, 2004.
- [27] T. Taylor, "A Genetic Regulatory Network-Inspired Real-Time controller for a Group of Underwater Robots", *Proceedings of Eighth Conference on Intelligent Autonomous Systems (IAS-8)*, 2004.
- [28] T. Taylor, P. Ottery and J. Hallam. "Pattern formation for multi-robot applications: Robust, self-repairing systems inspired by genetic regulatory networks and cellular self-organisation", *Informatics Research Report, 2007*.
- [29] K. Ioannidis, G.Ch. Sirakoulis, I. Andreadis, "Cellular ants: A method to create collision free trajectories for a cooperative robot team" *Robotics and Autonomous Systems*, In Press, Accepted Manuscript, Available online 30 October 2010
- [30] K. Simons, E. Karsenti, S. D., C. Wijer, and S. Swaminathan, Eds., *Self-Organization and Morphogenesis in Biological Systems*, Schloss Ringberg, Tegernsee, Germany, Dec. 2006.
- [31] H. De Jong, "Modeling and simulation of genetic regulatory systems: a literature review." *Journal of Computational Biology*, vol. 9, no. 1, pp. 67-103, 2002.
- [32] D. Endy and R. Brent. Modeling cellular behavior. *Nature* 409, 391-395. 2001.
- [33] J. Hasty, D. McMillen, F. Isaacs., and J.J. Collins, "Computational studies of gene regulatory networks: In numero molecular biology", *Nat. Rev. Genet.* 2, 268-279. 2001.
- [34] H. Guo, Y. Meng and Y. Jin, Cellular Mechanism for Multi-Robot Construction via Evolutionary Multi-Objective Optimization of a Gene Regulatory Network, *BioSystems*. Vol. 98, Issue 3, Dec. 2009. pp. 193-203.
- [35] Y. Jin, H. Guo, and Y. Meng, Robustness Analysis and Failure Recovery for a Bio-inspired Self-Organizing Multi-Robot System. *Third IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2009)*. San Francisco, California, Sept. 14-18, 2009.
- [36] L. Piegler and W. Tiller. *The NURBS*. Springer, Second edition, 1997
- [37] G. Di Marzo Serugendo, M.P. Gleizes, and A. Karageorgos. Self-organization and emergence in MAS: An overview. *Informatica*, vol. 30, pp.45-54, 2006
- [38] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC, July 2007.
- [39] H. H. McAdams, and A. Arkin. "Simulation of prokaryotic genetic circuits". *Ann. Rev. Biophys. Biomol. Struct.* 27, 199-224. 1998.
- [40] P. Smolen, D.A. Baxter and J.H. Byrne, "Modeling transcriptional control in gene networks: Methods, recent results, and future directions", *Bull. Math. Biol.* 62, 247-292. 2000.
- [41] E. Mjolsness, D. H. Sharp, and J. Reinitz, *A connectionist model of development*, *Journal of Theor. Biol.*, 152, pp/429-456, 1991.
- [42] W. Bohm, G. Farin, and J. Kahmann. A survey of curve and surface methods in CAGD. *Computer Aided Geometric Design*, 1(1):1-60, 1984.
- [43] H. K. Khalil, *Nonlinear Systems*, 3rd Edition, ISBN 0-13-067389-7, Prentice Hall, 2002.
- [44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 2, 2002. pp.182-197.
- [45] K. Deb and R. B. Agrawal. Simulated Binary Crossover for Continuous Search Space. *Complex Systems*, Vol. 2, No. 9:115-148. 1995.
- [46] K. Deb and M. Goyal, A Combined Genetic Adaptive Search (GeneAS) for Engineering Design. *Computer Science and Informatics*, Vol 4, No.26, 1996, pp. 30-45.
- [47] J. Borenstein and L. Feng, "Measurement and Correction of Systematic Odometry Errors in Mobile Robots." *IEEE Transactions on Robotics and Automation*, Vol. 12. No. 6. 1996. pp. 869-880.