MULTIDIMENSIONAL FREE-FORM SWARM SHAPE FORMATION INSPIRED ON BIOLOGICAL MORPHOGENESIS

André Luiz Braga, D.Sc*, Ronaldo Ribeiro Goldschmidt,D.Sc*,
Paulo Fernando Ferreira Rosa,D.Sc†

**Instituto Militar de Engenharia*
Rio de Janeiro-RJ, Brazil

Emails: andre.luiz.braga@ime.eb.br, ronaldo.rgold@ime.eb.br, rpauloime@ime.eb.br

Abstract— The morphogenesis process, which allows the full growth of living beings from individual cells, has been very attractive to create algorithms that control agent swarms because, like biological systems, it is distributed, robust and not dependent on external positioning. One of its key issues is how to specify complex swarm three-dimensional shape formation and the paradigm is also able to implement shapes with unlimited canonical dimensions and create Hyper-Surface shapes. In this paper we present an implementation approach of a distributed shape formation algorithm inspired in the morphogenesis paradigm and its performance simulation results against all the factors and parameters of the algorithm. We will show the advantages of this approaches related to the flexibility, robustness and applicability.

Keywords— Swarms, Morphogenesis, Shape Formation, Unmanned Aerial Vehicles, Multidimensional

1 Introduction

Morphogenesis is the biological nature process that describes how billions of individual cells self organize and grow into a precise shape. It inspires several solutions to control swarms of land robots and drones by exploiting its independence of an external global positional reference, shape precision and scalability(Levin, 2012). The mathematical model(Turing, 1952; Setty et al., 2012) allow us to create "Shapes" on any number of canonical dimensions transparently. Aerial Autonomous Swarms, for example, have at least three spatial and rotational dimensions but we could also add others such as "Energy Charge" creating a "Hyper-Shapes" formation process. We implemented these concepts in a sound multidimensional simulation architecture and our major challenge was to adjust several complex parameters, such as the minimum proximity distances and the movement rates and we present here the results which provided very efficient performance for the algorithm.

In the next sections we will describe the morphogenesis concepts and their applications in swarms shape formation. Section four presents the proposed mathematical model implemented, section five the algorithm and implementation and on six the simulation results, further analyzed in the last section.

2 Morphogenesis

2.1 Concepts

Morphogenesis is the biological process(Turing, 1952; Setty et al., 2012) from which individual cells reproduce creating macro-structures (Bhattacharyya, 2006) in multi-cellular organisms(Iber et al., 2016) and living beings(Kerszberg and Wolpert, 2007). The first mathematical model explaining this process was proposed by Alan Turing in 1952, since then, there have been studies on both better understanding it as well as using its principle in computing areas.

The most usual explanation (Kondo and Miura, 2010; Mamei et al., 2004; Guo et al., 2012; Jin, 2013) of this process is the idea that some chemical element, a protein(P), also known as *Morphogen*, diffuses from some cells to its neighborhood and the concentration of P, declining from the original source will translate into positional information. Let's say, for example, that some initial cell would have a concentration of P equals 100 and once it passes to the next neighbor, it declines to 90 and so on. Therefore, if we look at this rate as the gradient of P in some direction, lets say X, we would have a relative coordinate system totally independent of the real world one. This is absolutely necessary for biological cells, since they do not have any means to self-position on space and very interesting paradigm if we want to create algorithms for agents, such as a robot, that is able to position among themselves apart from real-world coordinates.

This concept is very useful for understanding the morphogenesis basic principle, the gradient related positioning, but not enough if we want to build a real working model and algorithms for multiple agents control. If we extend the idea of one morphogen type defines one dimension (ex. X) and another one, other spatial dimensions we would still have a problem: Since the cell has absolutely no spatial orientation with its surrounding environment, how could it know if it is propagating the morphogen on which direction ? This is the same problem a multi-agent robotic system would have if no global positioning system is provided. The real truth is that the main orientation for a coordinate system must be created. The biological systems does that implicitly.

The interaction between cells when propagating the morphogens, according to Turing model and most consensus in bio sciences community, works on an oscillatory feedback loop through reaction-diffusion. Basically, the protein P that represents the morphogen, propagates to the surrounding cells and stimulates the creation of another element, lets name "G"(that actually corresponds to the RNA) that in turn creates more of the protein P that propagates back to the original cell and its surroundings. The protein P activates the creation of its factory (G) but at the same time, inhibits itself, causing it to decay its concentration value. This back and forth mechanism between the cells creates an oscillation pattern of the propagation and decay of the morphogen from its original cells, named "Reaction-Diffusion Model"(Kondo and Miura, 2010). Well, so far we have a stable mechanism that reinforce the consistent gradient decay rate from the source that will define the relative positioning but, in such model, one can realize that the propagation will be the same in all directions and therefore no reference coordinate system can be created.

The equations (1) and (2), evolved from Turing's original work (Turing, 1952) in differential equations that describes the oscillatory process.The function $H_s(\mathbf{G})$ in equation (3) express the desired shape over the multidimensional position vector $\mathbf{G}$ to be formed by regulating the gradient decay and $\boldsymbol{f}_l$ is a non-linear function, such as sigmoid, to normalize the system. This computational model assumes a given coordinate system, what in real biology systems is created on the go.

$$\frac{dG_v}{dt} = mP_v - a.\frac{\partial H_s(\mathbf{G})}{\partial G_v}, v = (x, y, z...) \quad (1)$$

$$\frac{dP_v}{dt} = -cP_v - k.\boldsymbol{f}_l(\frac{\partial H_s(\mathbf{G})}{\partial G_v}), v = (x, y, z...) \quad (2)$$

$$H_s(\mathbf{G}) = f(G_x, G_y, G_z...) \quad (3)$$

3 Morphogenic Shape Definition Methods

3.1 Shape Types

Standard Geometric Shapes There are several approaches when specifying standard shapes, such as circles or straight lines(Dang et al., 2016; Yu and Barca, 2015; Seng et al., 2013; Barca et al., 2013), the most usual is to define "goal points" based on the standard shapes functions and calculate agents optimal trajectories to the closest goal points in the shape(Barca et al., 2013; Dang et al., 2016).

Free-Form Shapes There are three major approaches to create free-form shapes. The first is just define straight specify goal points(Barca et al., 2013) to every agent or a lead one with followers. This is a simple and low computational cost approach but not very efficient and does not exploit all the swarm potential. The second is basically to design an ad hoc geometrical algorithm for every different shape(Mamei et al., 2004; Barca and Sekercioglu, 2011; Yeom and Park, 2010; Kar-Han Tan and Lewis, 1996; Yu and Barca, 2015; Navarro and Matía, 2013). This is doubtfully a general solution given that there could be no deterministic algorithm for a certain shape. The third approach, fully exploiting the morphogenesis paradigm, named ***Gene Regulatory Network***(GRN)is capable of create any shape, even standard ones by basically defining the shape function gradient, in analytical form(Jin and Sendhoff, 2008; Guo et al., 2009; Jin et al., 2009) or in pieces using NURBS[1](Piegl and Tiller, 1996), what basically means defining the shape as a combination of parametric functions, what allow us to easily prove the algorithm convergence. The work of Yaochu Jin(Jin and Sendhoff, 2008; Guo et al., 2009; Jin et al., 2009; Guo et al., 2010; Guo et al., 2011; Jin and Meng, 2011; Meng et al., 2013; Ramezan Shirazi et al., 2014; Oh et al., 2016; Guo et al., 2012; Jin, 2013; Oh and Jin, 2014b; Oh and Jin, 2014a; Oh et al., 2017) has been evolving on this direction for many years but not very much going out of the bounds of mathematical simulations and not better explaining the algorithm implementation and reconciliation with the coordinate systems determination.

3.2 Morphgenesis and Swarm Movement

The most usual mistake when trying to implement or understand the morphogenesis process in a multidimensional swarm is ignoring some main differences between cell reproduction and swarm movement. In the former, the "movement" P_x for cells to reach a certain targetH_{xn} (Figure 1(A)) is by reproducing itself in the direction of the target, regarding each dimension. On doing that, the cells diffuses the decay of Morphogen G_x directly by contact on the same reproduction direction thus defining the position of each cell. On the other hand, in order to reach the same target, an agent of the swarm actually moves along the direction and towards the target (Figure 1(B)), therefore without no way to record its own Morphogen G_x decay and position. In the biological Morphogenesis, each cell implicitly knows that it is part of the whole swarm, since its own existence depends on the previous one, but in swarms, each element does not necessary knows the existence or position of other elements. The Morphogenesis

[1]Non-Uniform Rational B-Splines

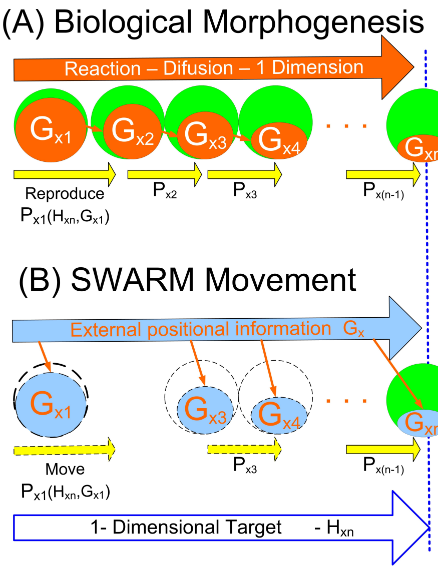

Figure 1: Morphogenesis x Swarm Movement

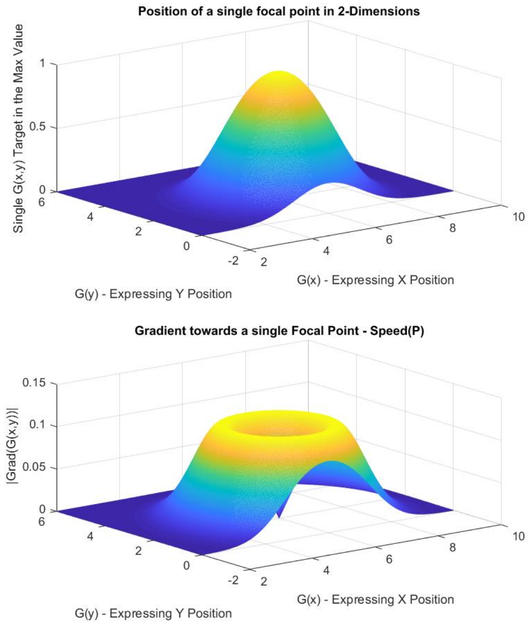

Figure 2: Single Point position and movement

paradigm would still be applied to the gradient descent movement towards the target (P_x) but it is necessary some external form of positional information to be communicated among the swarm elements. There are some methods that could be applied. The simple one is by external positioning systems, such as GPS[2]. Another already exploited solution by some researchers is sending distances among the swarm and reconstruct original positions using a euclidean matrix (Madej, 2009; Dokmanic et al., 2015). The more sound solution, keeping the inspiration on biological morphogenesis is to send directional signals among the elements of the swarm diffusing the morphogen G_x with the proper decay. This requires that the communication between emitters and receivers to be directional for each physical dimension.

4 Proposed Morphogenesis Inspired Free-Shape Formation Approach

The current morphogenesis approach for free-form shape formation(Guo et al., 2012; Oh and Jin, 2014a; Oh and Jin, 2014b; Oh and Jin, 2016) mainly focus on the definition of shapes using NURBS(Piegl and Tiller, 1996), in other words, combination of non-linear curves. Our approach uses focal points, as in (Barca and Sekercioglu, 2011; Barca et al., 2013) in combination with the GRN methods The dimension units presented in the graphs of the next sections are meant just for conceptual understanding and therefore not included

[2]Global Positioning System

4.1 Shapes by Focal Points

In order to better understand the basis of the morphogenesis shape formation, we first will demonstrate how one specific agent is driven towards one point of the shape and them extend the concept to a free-form shape composed by multiple arbitrary points.

Single Focal Point Shape The Morphogenesis model, as presented previously, assumes that the shape is defined by a function H(G(x),G(y)...], where G is the morphogen expressed as a position in every dimension. The gradient of H(s) express the movement or speed of the agent in each dimension. The figure 2(upper) shows an example in 2 dimensions for defining a single point of a shape. Notice that H(s) would really be a circle in 2D space where we assume that the radius is so small that the equations convergence would point to the center. The figure 2(lower), in the same scale shows the module of the gradient vector representing the speed P. Notice that the speed gets to a maximum value around the center point pulling it from the surroundings and decreasing to zero once the agent gets to the target. One has to imagine a N-Dimensional space where an Hyper-Point would be the center of an Hyper-Sphere of attraction.

Multiple Focal Points Shape We must now extend the concept applied in the previous section assuming that every free-form shape may be formed by a linear combination of H(s) Hyper-Point functions.H(s) of each focal point is zero for

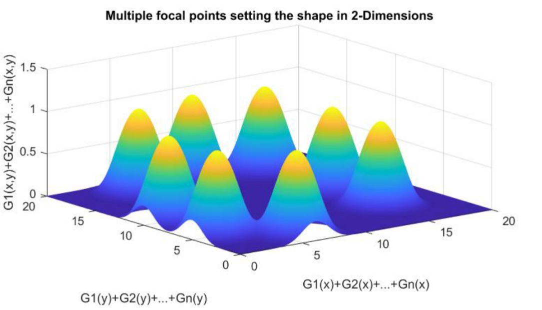

Figure 3: Multi Point Shape Position

every point far enough of each other (figure 3), and therefore, H(s) for a general shape may be written by eq.(4)

$$G_1(x, y, ..) + G_2(x, y, ..) + ... + G_i(x, y, ..) \quad (4)$$

The movement, or speed of the agent towards the shape formation is then expressed by the eq. (5) but we may realize that the multiple parameter adjustments in the gradient descent function could cause, for example, the points to collapse on one unique shape violating our assumptions about the linear combination.

$$\nabla G_1(x, y, ..)+\nabla G_2(x, y, ..)+...+\nabla G_i(x, y, ..) \quad (5)$$

5 Implementation

5.1 Algorithm

The equations (6) through (9) expresses our simulation implementation. The set of agents m in the positions $\overrightarrow{G}_m$ (7) searches for the$n$ targets $\overrightarrow{T}_n$(6) by calculating all the multidimensional euclidean distances $\overrightarrow{D}_m$(8). The combination of the distances for all targets could be combined weighting according to the distances or just picking the closest one. Either way, the movement towards the target, expressed by $\overrightarrow{P}_m$(9) obeys the decay functions $lambda_l$ on each moving dimension l (9). Note that not necessarily all dimensions may be allowed to "change" when moving to the multidimensional "Hyper-Target". If, for example, one of the dimensions is an energy state, we may want the agents to find the targets with the same energy without changing its internal one along the way. The decay vector $\overrightarrow{\lambda}$ would implicitly be setup and express the swarm shape formation characteristics.

$$\overrightarrow{T}_n, n = (1, 2, ...) \text{ - Target shape points} \quad (6)$$

$$\overrightarrow{G}_m, m = (1, 2, ...) \text{ - Agent's position} \quad (7)$$

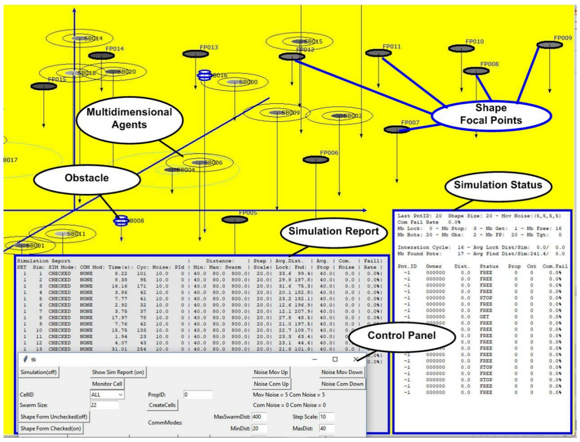

Figure 4: Simulator Main Screen

$$\overrightarrow{D}_m = \overrightarrow{T_n} - \overrightarrow{G}_m|_{\min_n \|\overrightarrow{T_n} - \overrightarrow{G}_m\|} \quad (8)$$

$$\overrightarrow{P}_m = \overrightarrow{\lambda} \cdot \overrightarrow{D}_m, \overrightarrow{\lambda} = \{\lambda_1, \lambda_2, ..\lambda_l\}$$
$$\lambda_l = \left\{ \begin{array}{l} \kappa_l e^{-\alpha t}, l \in \{Moving\ dimentions\} \\ 0, otherwise \end{array} \right\| \quad (9)$$

5.2 The Simulator System

The system proposed architecture act as a proof of concept for the issues and strategies regarding the swarm shape formation. The main screen (figure 4) is just a viewer for internal components, therefore decoupling the relationship with the mathematical elements. It provides the multidimensional aspect of the swarm, such as the three spatial and the 4th represented by the gray shade of the agents. By no means, it has a precise scale or geometric perspective. The simulator generates a unique random set of focal points for all simulation sets, represented as the larger gray shaded filled circles. The Agents of the Swarm are the smaller and surrounded by two others indicating the minimum and maximum distance. It also creates some random obstacles that would not move but has to be avoided by the agents. They do not just move in three-dimensional space but also in the 4th dimension by changing its gray shade level. Each agent performs one step, by cycle and the number of cycles in each simulation gives us the sense of how long it takes to meet the final target. We may manually adjust each parameter for detailed analysis but a predefined number of simulation sets with specific parameters may be fired and stored in data files.

5.3 Issues and Strategies

1. **Shape Precision** - The shape formation purpose, such as surrounding a target or surveillance, requires different shape precision. We use the average distance of each target even so is allowing multiple agents move

to the same focal point but this is acceptable if we reach the overall shape.

2. **Collision Avoidance** - The morphogenesis algorithms used in Gene Regulatory Networks(GRNs) (Rambabu et al., 2015; Oh and Jin, 2016; Oh et al., 2017; Oh and Jin, 2014b) incorporate a collision avoidance(CA) factor that even proven mathematically sound, it is not reliable on discrete numerical simulations and real world devices. It deals with swarm agents as single points but in real life, it is necessary to keep safe distances to prevent possible damages. We implemented two distance levels, the closer(Min) is used to either indicate danger proximity and target met. The second (Max) act as a warning proximity level and indication of target proximity.

3. **initial Conditions and Competition** - The analysis of GRN equations shows that multiple initial conditions converge to the same shape form point. In other words, swarm agents would compete for the same target, and therefore we implemented the following strategies:

   "Push Away" - This is similar to biological systems and CAM[3](Taylor et al., 2007; Ostergaard et al., 2005), where if the element gets closer to the target it increases its own the collision avoidance range and therefore "repels" other elements. The downside is that if targets fall too much apart, multiple agents would still keep around a taken location. On the hand, it does not requires intense intercommunication among the swarm.

   Locking Target - The agent would "Lock" some specific target to itself and inform all the other agents that the point is no longer available. If competition arises, the agent surrenders the target to a closer agent, on the other hand, it increases the communication cost of exchanging the target lists and owners among the swarm.

4. **Communication** - This critical factor that could cause, not just lack of information but also noise, imprecise distance and data. The communication among agents improve precision but increase the risk of failures, therefore we simulate and measure their effects in the swarm algorithm.

5. **Trajectory** - The gradient descent algorithm does not take into account that real life agents are not points without dimensions and collision avoidance areas causing possible deadlocks or trajectory blocks by other agents.

[3]Cohesion Adhesion Model

Table 1: Simulation Parameters Sample

Sim Set	Nb. Sim	Shape Mode	Av.Stop Dist.	Mov. Noise	Com. Noise
1	100	0	20	10	0
10	150	1	20	15	0
21	150	1	5	6	50

There is no definitive heuristic to find the "best path" to a target blocked by such random moving elements, therefore, we use the following strategies:

Gradient Adjustment(steps) - This is actually part of the GRN gradient descent movement algorithm. It calculates each step to the target dynamically depending on the target distance, therefore, at each new step, a new gradient direction scenario will be evaluated.

Movement Noise - A slight random noise applied to the agent movement increases the chance to find the right path to the target. If the gradient descent step directs the agent to a path passing inside the Collision Avoidance(CA) area to another, the CA algorithm would keep it there forever trying. One small "Push" in some direction places the agent in right straight path to the target. It also helps the "Push Away" mode inducing the agents try new targets when competing.

6 Simulation

6.1 Parameters

There are three types of parameters in the simulation, (**P**)reset ones, which will not change for all sets, the (**S**)imulation ones, which changes for every simulation set and the (**M**)easures collected for every simulation set, as follows (Examples in Table 1[4]):

1. **Simulation Set**(P) - Set identifier
2. **Max. Time out cycle**(P) - Maximum number of run cycles if the goal is not met.
3. **Shape Mode**(S) - Shape formation technique: "push away" / "locking targets".
4. **Avg. Stop Dist.**(S) - Average target swarm met distance.
5. **Mov. Noise**(S) - Step random noise.
6. **Com. Noise**(S) - Random communication fail rate.
7. **Shape Time**(M) - Time to reach shape.

[4]Complete simulation data available in: ***www.github.com/andrelb2000/Phyton/tree/master/MorphSimBot/SIMI***

6.2 Results

The results obtained by the simulations do not have meaningful real units but just for comparison regarding the adjusting of the parameters. The time, number of cycles and coordinate positions of a real swarm would have different nominal values but for the purposes of this analysis, we believe to have properly expressed the real conditions. The purpose of the simulation is the comparative analysis of the influence of several parameters over the shape formation. The number of cycles that the simulation takes to reach the target may seems very reasonable but since the average time for each cycle to complete may change the total simulation time gives us a better perspective. The first information we may realize in the graph on figure 5 is the improvement in the time of the "Push Away" simulations (B1,B6) once we accept less precision on the shape formation. Once we change the strategy to "Lock Targets"(B2) but keep the same shape precision, the time improves and gets more stable. The movement noise could also be reduced not been required so much for trajectory deadlocks. We observe the trade-off between the movement noise and precision in (B3) but when we impose close to zero average distance (B4) the time increases. Communication noise and failures are introduced after the 400 simulations and even so, the shape time gets lower and more stable (B5). This scenario is due to the fact that the communication failure gets back the system closer to the "Push Away" scenario, when the system spends less time exchanging information and the "Push Away" mechanism, still in place, takes care of most of the deadlocks and competition issues. The (B7) set improves a bit more the time but at the expense of less precision, therefore, we may observe an optimal trade-off scenario in (B8) where we have a very good precision, reasonable trajectory noise, Shape Met time and robust to communication failures.

7 Conclusions

We presented the fundamentals of the morphogenesis shape formation paradigm and a proof of concept simulation that could be applied to any number of dimensions but there are several parameters that need to be adjusted for the proper efficiency. Our results demonstrate the major trade-off between the intense communication among the swarm agents and the shape precision when we try to optimize the final shape formation time. In the former, even so all the agents are aware of the data of the others, failures may derail the whole process. On the other hand, the "Push Away" technique proposed works well up to a certain shape precision. The best combination strategy, as our simulation results indicate, keeps the system robust to communication failures using the "Push Away" as a redundant backup in the presence of failures and at the same time keeping high precision by the "Lock Target" technique. We expect to advance the current work by analyzing simulations in more dimensions and further implementation of the algorithm in real micro drone swarm systems.

References

Barca, C. and Sekercioglu, A. (2011). Generating formations with a template based multi-robot system, *Australasian Conference on Robotics and Automation*, Vol. 20, Monash University, Melbourne, Australia, pp. 7–9.

Barca, J. C., Lee, E. E.-J. and Sekercioglu, Y. A. (2013). Flexible Morphogenesis based Formation Control for Multi-Robot Systems, *International Journal of Robotics and Automation (IJRA)* **2**(1): 31–38.

Bhattacharyya, A. (2006). Morphogenesis as an amorphous computation, *Proceedings of the 3rd conference on Computing frontiers - CF '06*, ACM Press, p. 53.

Dang, V.-L., Le, B.-S., Bui, T.-T., Huynh, H.-T. and Pham, C.-K. (2016). A decentralized localization scheme for swarm robotics based on coordinate geometry and distributed gradient descent, *MATEC Web of Conferences* **54**: 02002.

Dokmanic, I., Parhizkar, R., Ranieri, J. and Vetterli, M. (2015). Euclidean Distance Matrices: Essential Theory, Algorithms and Applications, pp. 1–17.

Guo, H., Jin, Y. and Meng, Y. (2012). A morphogenetic framework for self-organized multirobot pattern formation and boundary coverage, *ACM Transactions on Autonomous and Adaptive Systems* **7**(1): 1–23.

Guo, H., Meng, Y. and Jin, Y. (2009). A cellular mechanism for multi-robot construction via evolutionary multi-objective optimization of a gene regulatory network, *BioSystems* **98**(3): 193–203.

Guo, H., Meng, Y. and Jin, Y. (2010). Analysis of local communication load in shape formation of a distributed morphogenetic swarm robotic system, *2010 IEEE World Congress on Computational Intelligence, WCCI 2010 - 2010 IEEE Congress on Evolutionary Computation, CEC 2010*, IEEE, pp. 1–8.

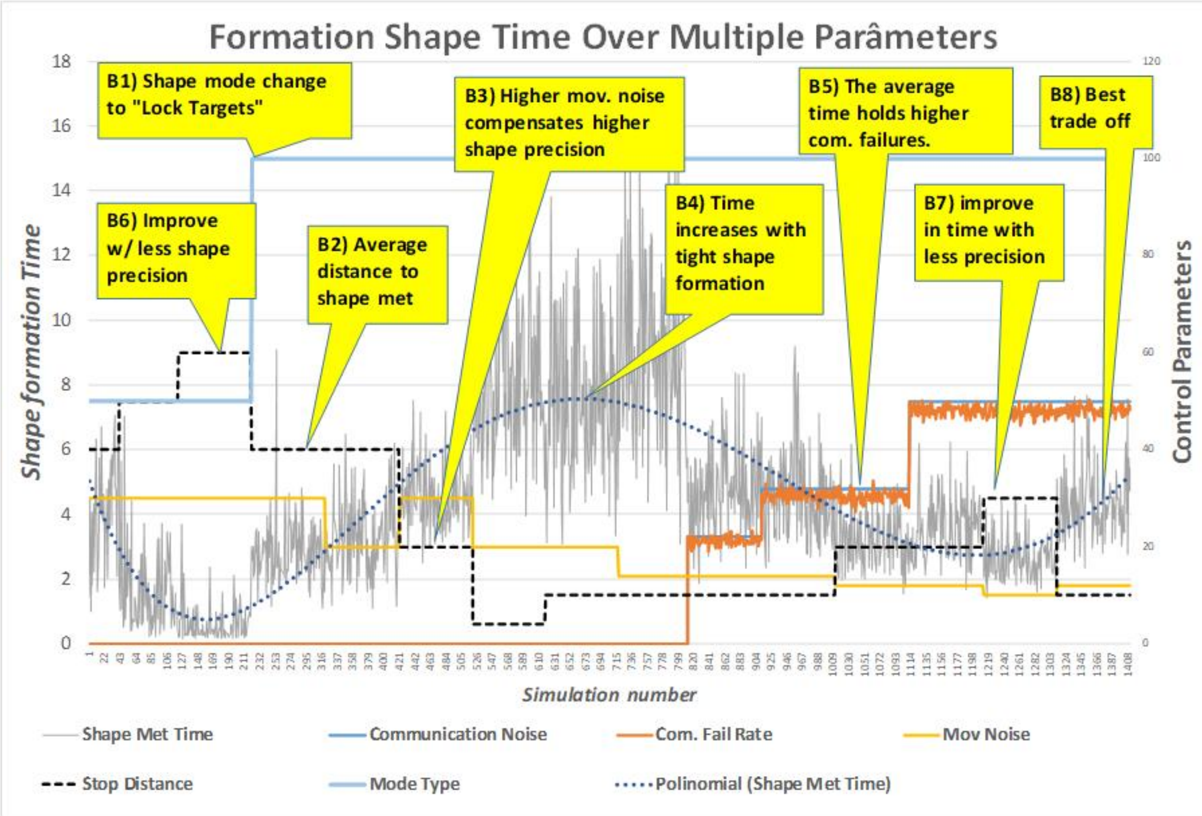

Figure 5: Shape Time Formation by Simulation

Guo, H., Meng, Y. and Jin, Y. (2011). Swarm robot pattern formation using a morphogenetic multi-cellular based self-organizing algorithm, *Proceedings - IEEE International Conference on Robotics and Automation*, IEEE, pp. 3205–3210.

Iber, D., Karimaddini, Z. and Ünal, E. (2016). Image-based modelling of organogenesis, *Briefings in Bioinformatics* **17**(4): 616–627.

Jin, Y. (2013). Morphogenetic Multi-Robot Pattern Formation Using Hierarchical Gene Regulatory Networks, *FOCAS Workshop*, number September, Taomina, Italia.

Jin, Y., Guo, H. and Meng, Y. (2009). Robustness analysis and failure recovery of a bio-inspired self-organizing multi-robot system, *SASO 2009 - 3rd IEEE International Conference on Self-Adaptive and Self-Organizing Systems*, IEEE, pp. 154–164.

Jin, Y. and Meng, Y. (2011). Morphogenetic robotics: An emerging new field in developmental robotics, *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* **41**(2): 145–160.

Jin, Y. and Sendhoff, B. (2008). Evolving in silico bistable and oscillatory dynamics for gene regulatory network motifs, *2008 IEEE Congress on Evolutionary Computation, CEC 2008*, IEEE, pp. 386–391.

Kar-Han Tan and Lewis, M. (1996). Virtual structures for high-precision cooperative mobile robotic control, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96*, Vol. 1, IEEE, pp. 132–139.

Kerszberg, M. and Wolpert, L. (2007). Specifying Positional Information in the Embryo: Looking Beyond Morphogens, *Cell* **130**(2): 205–209.

Kondo, S. and Miura, T. (2010). Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation, *Science* **329**(5999): 1616–1620.

Levin, M. (2012). Morphogenetic fields in embryogenesis, regeneration, and cancer: Non-local control of complex patterning, *BioSystems* **109**(3): 243–261.

Madej, W. (2009). Computations of Coordinates in a Multidimensional Digital Prediction System, *Defence Science Journal* **59**(5): 530–536.

Mamei, M., Vasirani, M. and Zambonelli, F. (2004). Experiments of morphogenesis in swarms of simple mobile robots, *Applied Artificial Intelligence* **18**(9-10): 903–919.

Meng, Y., Guo, H. and Jin, Y. (2013). A morphogenetic approach to flexible and robust shape formation for swarm robotic systems, *Robotics and Autonomous Systems* **61**(1): 25–38.

Navarro, I. and Matía, F. (2013). A survey of collective movement of mobile robots, *International Journal of Advanced Robotic Systems* **10**(1): 73.

Oh, H. and Jin, Y. (2014a). Adaptive swarm robot region coverage using gene regulatory networks, *in* M. Mistry, A. Leonardis, M. Witkowski and C. Melhuish (eds), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 8717 LNAI, Springer International Publishing, Cham, pp. 197–208.

Oh, H. and Jin, Y. (2014b). Evolving hierarchical gene regulatory networks for morphogenetic pattern formation of swarm robots, *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014*, number 601062, IEEE, pp. 776–783.

Oh, H. and Jin, Y. (2016). Evolving H-GRNs for Morphogenetic Adaptive Pattern Formation of Swarm Robots, *in* H. Iba and N. Noman (eds), *Evolutionary {Computation} in {Gene} {Regulatory} {Network} {Research}*, John Wiley & Sons, Inc., Hoboken, NJ, USA, pp. 327–361.

Oh, H., Ramezan Shirazi, A., Sun, C. and Jin, Y. (2017). Bio-inspired self-organising multi-robot pattern formation: A review, *Robotics and Autonomous Systems* **91**: 83–100.

Oh, H., Shiraz, A. R. and Jin, Y. (2016). Morphogen diffusion algorithms for tracking and herding using a swarm of kilobots, *Soft Computing* pp. 1–12.

Ostergaard, E. H., Christensen, D. J., Eggenberger, P., Taylor, T., Ottery, P. and Lund, H. H. (2005). HYDRA: From Cellular Biology to Shape-Changing Artefacts, *in* D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, W. Duch, J. Kacprzyk, E. Oja and S. Zadrony (eds), *Artificial {Neural} {Networks}: {Biological} {Inspirations}*, Vol. 3696 of *{ICANN} 2005*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 275–281.

Piegl, L. and Tiller, W. (1996). *The NURBS Book*, Vol. 28 of *Monographs in visual communications*, 2nd ed edn, Springer, Berlin ; New York.

Rambabu, R., Bahiki, M. R. and Md Ali, S. A. (2015). Relative position-based collision avoidance system for swarming UAVS using multi-sensor fusion, *ARPN Journal of Engineering and Applied Sciences* **10**(21): 10012–10017.

Ramezan Shirazi, A., Oh, H. and Jin, Y. (2014). Morphogenetic self-organization of collective movement without directional sensing, *in* M. Mistry, A. Leonardis, M. Witkowski and C. Melhuish (eds), *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 8717 LNAI, Springer International Publishing, Cham, pp. 139–150.

Seng, W. L., Barca, J. C. and Sekercioglu, Y. A. (2013). Distributed formation control in cluttered environments, *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics: Mechatronics for Human Wellbeing, AIM 2013*, IEEE, pp. 1387–1392.

Setty, Y., Cohen, I. R. and Harel, D. (2012). Executable modeling of morphogenesis: A Turing-inspired approach, *Fundamenta Informaticae* **118**(4): 403–417.

Taylor, T., Ottery, P. and Hallam, J. (2007). An approach to time-and space-differentiated pattern formation in multi-robot systems, *Proceedings of Towards Autonomous Robotic Systems (TAROS 2007)* pp. 160–167.

Turing, A. M. (1952). The Chemical Basis of Morphogenesis, *Philosophical Transactions of the Royal Society of London* **237**(641): 37–72.

Yeom, K. and Park, J. H. (2010). Artificial morphogenesis for arbitrary shape generation of swarms of multi agents, *Proceedings 2010 IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA 2010*, IEEE, pp. 509–513.

Yu, S. and Barca, J. C. (2015). Autonomous formation selection for ground moving multi-robot systems, *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, Vol. 2015-Augus, IEEE, pp. 54–59.