A Scalable and Distributed Approach for Self-Assembly and Self-Healing of a Differentiated Shape

Michael Rubenstein, Wei-Min Shen

Abstract— **As the ability to produce a large number of small, simple robotic agents improves, it becomes essential to control the behavior of these robots in such a way that the sum of their actions gives rise to the desired overall result. These robots are modeled as homogeneous, distributed robots, with only one simple short range sensor. Our simple robots are tasked to form and hold a desired swarm shape, independent of the total number of agents. If this shape is damaged by the removal of some of the robots, the remaining agents will recover the former shape, but on a smaller scale. These shapes can also have a pattern such as a picture or drawing displayed on them by controlling the individual robots color, symbolically representing the differentiation of agents within the swarm. This pattern will resize to fit the existing swarm. With the ability to synchronize in time, the swarm gains the ability to change the pattern displayed, resulting in a moving image.**

I. Introduction

As ability to produce large numbers of simple robots improves, it becomes increasingly important to control the robots behavior. Our goal is to produce a biologically inspired control strategy for a large group of these robots, often called a swarm, to form and hold a differentiated shape that is composed of these small, simple robots, analogous to how cells form the body of an animal. This differentiated swarm shape may empower the group of robots to complete a task as a whole that would be difficult or impossible for them to accomplish as a single agent.

A powerful example of this is the field of modular robotics, whose underlying principal is that a group of cooperating robots is more able as a collective, compared to the sum of the efforts of each robot if working individually. Individually, these modular robots, such as those shown in [1] have limited mobility. However, when many of the robots are joined together, the shape of the swarm gives rise to highly mobile locomotion strategies, for example [2], which are far better than those employed by a single modular robot. Aside from forming a shape, a robotic swarm also allows some interesting additions in functionality, which is not easily possible using a single robot, such as the ability to self-heal. If the shape of a swarm is playing a critical role in its accomplishing a task, and that shape is damaged, the proficiency of the group to complete the desired task may be reduced, or be completely removed. If these swarms can then recover their shape after the damage occurs, then it may also recover its swarm functionality, making self-healing a quintessential part of swarm applications.

If this principal of self-healing was applied to our modular robot example, the swarm of modular robots could regain its functionality after damage occurs. For example, if a hexapod composed of modular robots has a leg removed, then it can no longer perform a “tripod” gait, greatly reducing its mobility. If the modular robots had a way to re-form the missing leg, for example through self-reconfiguration, then it could once again use the “tripod” gait for locomotion.

Other than making a contribution towards the overall shape of the swarm, a robot can contribute towards a swarm task in other ways. One such way it to take on specific functions, depending on the robots location within the swarm, i.e. differentiation. In nature, differentiation is common, where stem cells differentiate into distinct cell types such as skin cells, depending on their location in the body. This arrangement of cells allows skin cells to be located on the outside of the body, where they are most effective at their main job, to protect the inside of the body from invaders. If instead of arranging themselves on the outside, skin cells were just randomly distributed throughout the body without any regard to location, then they would lose their effectiveness. This power from differentiation of cells in a biological organism can be used similarly in a robotic swarm, making it more effective at a given task when it is differentiated.

Going back to the hexapod built with modular robots, each module needs to differentiate its behavior in order to properly contribute to the hexapod gait. In this case, a module in the backbone needs to move and behave as a part of the backbone. If instead, it were to behave as a leg, or exhibit no behavior at all, it would negatively affect the hexapod locomotion. This location dependant specialization of robots along with shape formation and healing will increase the effectiveness of the swarm for its desired task.

In nature, the ability to form and heal a differentiated shape has come about through millions of years of evolution, and comes in two different styles. One style, called epimorphosis, uses cellular reproduction to re-grow damaged parts. With epimorphosis, a salamander, a starfish, or a lizard, can re-grow a lost limb or tail, but their body remains unchanged. This re-grown limb will be the same scale as the original one. A second style of self-healing, called morphallaxis uses cellular movement to reorganize

Manuscript received February 22, 2008.

Michael Rubenstein is with the Information Sciences Institute, at the University of Southern California, e-mail: mrubenst@usc.edu

Wei-Min Shen is with the Information Sciences Institute, at the University of Southern California, e-mail: shen@isi.edu

the remaining cells in order to reform the structure. This will result in a smaller but complete version of the original shape. We define this as *scalable self-healing.* A popular example of morphallaxis occurs in a small freshwater invertebrate, called the hydra. Using its self-healing abilities, the hydra can recover from many types of seemingly irrecoverable damage. If chopped in half, each half will re-form a new hydra, half the size of the original hydra. It can be divided into 20 pieces, and each piece will form the original hydra shape, just 1/20th the size [3]. The hydra can be scrambled, randomly placing the cells back together, and the cells still move to reform a hydra. If the hydra is prevented from making new cells, it will slowly shrink as the number of alive cells is reduced due to random cell death, always maintaining its shape even as the available number of cells diminishes. Through the use of morphallaxis, the hydra has gained the ability to recover from serious damage, asexually reproduce, and is even suspected to be capable of living forever [4].

Currently most artificial self-healing systems envision using a large swarms of simple, small, distributed, homogeneous robots similar to the robots found in [1][5][6] to form the desired self-healing structure, analogous to how biological cells make up an organism. The challenge faced by most of these systems is to first develop a method for identifying the location of a robot within the swarm or structure, and second, how to determine actions for each robot in the swarm that contributes to forming or healing the desired shape. This process is further complicated by the limitations of the robots. The robots are envisioned to be simple in order to keep cost/complexity down, enabling their production in large numbers. The robots also tend to be homogeneous, again to keep cost and complexity down. To allow swarm recovery from damage to any robot, the algorithms tend to be distributed, preventing the loss of a coordinating leader robot due to damage.

The method for accomplishing self-organization and epimorphosis based self-healing developed in [7] is impressive, in that it can grow and heal any shape. However, it requires the robots to reproduce, and the field of self-replication in artificial systems such as robots is still in its infancy at best [8][9].

Stoy et al. [10] developed a method for self-healing that is capable of forming and healing an impressive number of 3D shapes; it makes use of direct messaging from one agent to another, i.e. a private communication path between connected robots, increasing the complexity of the robots. They also require a starting seed to start formation, a form of centralization.

The work in [11], by Nagpal et al., simulates very simple robots capable of forming and holding 2d shapes; however, they use some global sensors. For example, to detect the orientation of the robots, a "compass" is used. They also cannot adjust the size of the shape formed to optimally use the remaining robots after damage as seen in natural morphallaxis; they keep the shape the same size, just less dense, i.e. not scalable self-healing.

In this paper, we propose a method for self-healing which is geared towards emulating morphallaxis seen in the hydra on a swarm of very simple, low cost, physically realizable, homogeneous, distributed robots. They have only one sensor, and this sensor is used for both distance measurement and inter-robot communication. These agents can form (from any starting configuration) and hold a swarm in a class of shapes, and in the event of damage, reform the shape. This is all done without any centralized functions, such as a seed or leader. This shape is fully given to every robot ahead of time, in the form of the function f_{shape}. The shape formed by the swarm will be correctly sized for the number of agents available by adjusting its scale, i.e. scalable self-healing. They can also demonstrate the ability to differentiate based on location in the swarm, and time, and every robot is given a full description of the differentiation pattern ahead of time, again in the form of a function $f_{pattern}$. This differentiation is shown by the robot displaying a color.

In this paper, we will discuss the model and assumptions used for our simulated robots. We will then go in to detail about the methods used to produce self-healing, self-assembly and pattern formation in this swarm of simulated robots. Next we will show some results of this simulated behavior, discuss some properties of these methods, and then conclude.

II. Model

Our methods for self-healing, self-assembly, and differentiation were tested on a swarm of simulated "puck" robots existing in a simple 2D planar world. The assumptions made in the robot model in this simulation are chosen to closely emulate the properties of living cells as close as possible, keep the cost of the robots to a minimum, and are constrained to be as similar to the capabilities of modern robotics/electronics as possible.

These simulated homogeneous robots are identical to each other in every way possible, and even lack a unique identifier. Each robot has one simple exteroceptive (responds to external stimuli) sensor, capable of directly communicating with nearby neighboring robots, and measuring the distance of that communication path. Simulated noise in the form of a zero mean gaussian [14], is added to these sensor readings, in order to represent the noise found in real sensors. This simple sensor is the only sensor available to the robots. This means that the robots have no global knowledge of their environment, such as a compass, gravity sensor, or GPS. Each robot has two degrees of freedom for moving along the plane, and can move up to a maximum speed. In order to display a pattern, each robot can dynamically change its color to one of its choice.

The robots initially are placed at random positions and orientations with respect to world and each other. The robots have no initial knowledge about their location or orientation in this world. They are modeled as a finite sized circle that won't overlap other robots. Each robot has a clock that runs at the same rate as the other robots, however, these clocks are initially started at random times, i.e. not synchronized. The simulation view port is fixed above the 2D plane, looking down. In this viewport, each robot is represented as a pixel of its chosen color, while empty space is shown as black.

The simulation runs in discrete time steps. During each time step, T, each robot will communicate with its neighbors, during which it will receive information from their T-1 time step. The position of each robot is then updated on the 2D plane, along with their displayed color.

III. METHODS

The self healing, self-organization, and differentiation behavior in each robotic agent is separated into three concurrently running tasks. The first task is to setup and maintain a coordinate system that uniquely and correctly identifies the location of each robot. The method used to produce a coordinate system is independent to the rest of the self-healing, and could actually be produced using any method desired; however, the method used, called trilateration, is capable of forming a coordinate system using the simple sensor we envision on the robots, while other methods may require more complicated sensors. Second, each robot makes decisions and movements to form or reform a predefined shape in scalable sense. This task makes use of the coordinates determined in the first task, as well as the given function f_{shape} in a method we call relentless Self-Assembly. Lastly, each robot modifies its color based on $f_{pattern}$, and the coordinates from trilateration, to display a pattern, or time-varying pattern.

A. Trilateration

The task of setting up and maintaining a coordinate system is done using the available range sensor, and communication between neighboring robots through a process called trilateration. Trilateration is an iterative process that is initialized with each robot forming a random initial guess as to its location in a coordinate system $(x_{self},y_{self},z_{self})$. In every iteration the robot will receive neighbor j's estimated values for j's own (x_j,y_j,z_j), for all neighbors j within its sensor range. Simultaneously during each communication with a neighbor, the distance of that communication $d_{j,self}$ is also measured with the sensor. These values of $d_{j,self}$ and (x_j,y_j,z_j) are collected for each neighbor, and are used in the trilateration formula (1). Values for $(x_{self},y_{self},z_{self})$ are then updated using a gradient decent method, in order to minimize (1).

$$\sum_{j=1}^{All\ Neighbors}\left(d_{j,self}-\sqrt{\left((x_{self}-x_j)^2+(y_{self}-y_j)^2+(z_{self}-z_j)^2\right)}\right)^2+|z_{self}| \quad (1)$$

This minimization results in values for $(x_{self},y_{self},z_{self})$ that reduce the error of the guessed distance between $(x_{self},y_{self},z_{self})$ and (x_j,y_j,z_j) when compared to the actual measured distance, $d_{j,self}$, and given the right conditions, over many iterations, the coordinate system for every robot will converge on a solution that uniquely and correctly identifies the location of each robot.

A similar method of trilateration has been used before for localization of robot groups [11], [12], and in a statistical method multi-dimensional scaling [13]. Formula (1) differs from these other trilateration implementations in that it allows three degrees of freedom even though the robots exist on a 2D plane. In order to reduce the problem back from the 3D space to the 2D X-Y plane, (1) contains a forcing term $|z_{self}|$. It has been observed experimentally that this modification allows the swarm to converge on a correct coordinate system quicker and with fewer neighbors than without it.

There is no constraint to the coordinate system that is agreed upon through trilateration, due to the complete lack of global knowledge. Because of this, the origin and axis of the coordinate system could be in any orientation and is equally likely to be left handed or right handed when viewed from the global viewport. To demonstrate this, the top of figure 1 shows a desired shape/pattern, while the squares below it show images taken from multiple simulation runs, with various orientations and handedness.

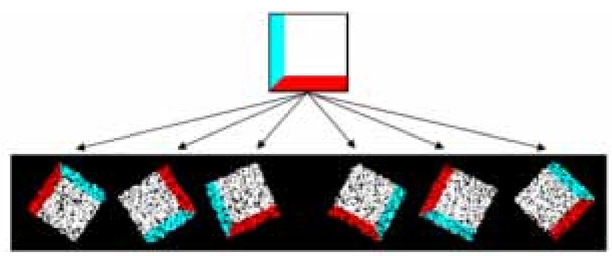

Fig. 1. Demonstrating the arbitrary orientation or the agreed-upon coordinate system discovered through trilateration.

While trilateration can help the robot learn its position in the swarm, it cannot directly tell the robot its orientation. To learn its orientation with respect to the agreed-upon coordinate frame, a robot makes two movements, orthogonal to each other, and finds its new coordinates after each movement. After its orientation in the agreed upon coordinate system is known, a robot can move in a desired direction with respect to that coordinate system.

B. Relentless Self-Assembly

We observe that if the swarm can re-form from any type of damage, including complete random shuffling of robots, then this condition is no different than the swarm self-assembling from a random starting pattern. Consequently, if our swarm can heal from a complete random shuffling of robots, then it also has the ability to self-assemble. Using this observation, we propose to solve the problem of self-assembly and the problem of self-healing with a single method, called relentless self-assembly. This method is implemented by having every robot in the swarm continuously move in an attempt to form the required shape.

This movement is determined by the robot using its current position from trilateration converted to polar coordinates (r_{self}, θ_{self}). These position values are plugged into the given function describing the shape, $f_{shape}(r_{self}, \theta_{self})$ and the direction of the negative surface normal is calculated. The robot will then try to move in that direction, while checking to see if a collision occurs, by monitoring if its position values change. If a collision is detected, the robot will just try move in a direction chosen randomly. This process of using the robots current coordinates to determine a movement direction repeats indefinitely.

Currently, the target shape to be formed by the swarm, f_{shape}, is constrained to any shape that can be described as a smooth positive polar function from 0 to 2π.

The iterative movement rule will cause the swarm to form the desired shape in the scale appropriate to the number of agents. This movement can be visualized as the robots existing in a potential field in the form of the desired shape, note: this field is not used in the implementation, and is used only for visualization purposes. The robots will try to move inward in the direction of the field gradient, towards the lower potential. In figure 2, a visualization of the potential field for some sample shapes can be seen, where the darker colors represent lower potential, or a lower multiple of the shape function. Following this rule, the robots will ensure that all the available positions with a potential lower than its current potential will be filled by another robot.

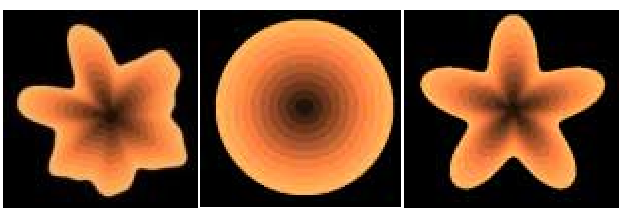

Fig. 2. Potential field for some sample shapes

C. Differentiation

Generally, the specialization that a robot can differentiate into is a property of what physical capabilities the individual robots have. For the simplicity of our robots, this specialization is limited to the capability of changing their color. Each robot chooses this color to properly contribute towards a desired pattern over the shape of the swarm. To properly display this color pattern, each robot must determine the size of the shape. This size is represented as R_{max} , which is the largest scalar multiple of the shape function $f_{pattern}$ that encloses every agent in the swarm. This is calculated iteratively in (2).

$$R_{max} = \max\left\{\sum_{j=1}^{N}\left(\frac{R_{max}^{j}}{N}\right), \quad \frac{r_{self}}{f_{shape}(\theta_{self})}\right\} \quad (2)$$

Where each robot updates its R_{max} value to either the average R_{max} of its N neighbors or to its own scale (r_{self} / $f_{shape}(\theta_{self})$), whichever is larger. However, a small random percentage of the time, a robot will reset its R_{max} value to be its own scale, in order to allow the scale of the shape to shrink in the event of damage. Each robot then uses its current R_{max} to determine its color to display using the function

$$f_{pattern}(r_{self}, \theta_{self}, R_{max}) \rightarrow color$$

This pre-defined function $f_{pattern}$ tells the robot what color it should display for its given position (r_{self}, θ_{self}), and the detected size of the swarm (R_{max}).

If the swarm is synchronized in time, then it can display a differentiation pattern that varies not only in swarm position, but in time as well. To accomplish this, all robots are given a rule (3) to automatically synchronize its internal clock t_{self} with those of its N neighbors.

$$t_{self} = \sum_{j=1}^{N}\left(\frac{t_j}{N}\right) \quad (3)$$

With this iterative rule, every agent's clock, t_{self}, becomes synchronized to that of its neighbors by setting its clock to be the average of its neighbor's clocks. With the time synchronization, the function for determining a robots color is modified as:

$$f_{pattern}(r_{self}, \theta_{self}, R_{max}, t_{self}) \rightarrow color$$

This modified function $f_{pattern}$ tells the agent what color it should display for its given position (r_{self}, θ_{self}) time (t_{self}) and the current scale of the swarm (R_{max}).

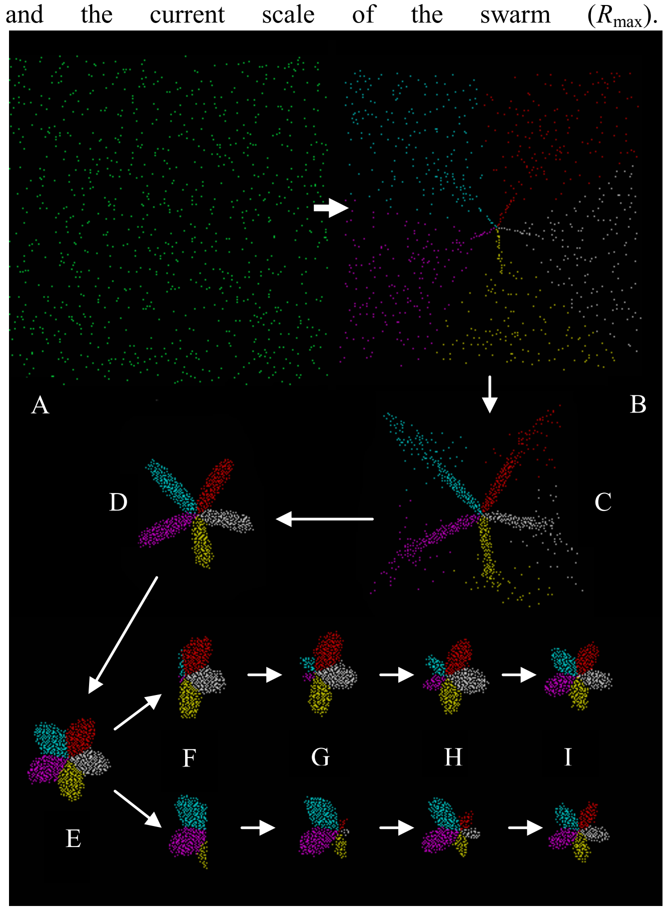

Fig. 3. Self healing starfish shape formation and recovery after damage and separation

IV. RESULTS

The rules described in this paper allow a swarm of simulated robots with the properties described in the model to exhibit relentless self-assembly and differentiation. These rules allow a swarm whose robots are placed in random starting locations (figure 3A) to move to form the desired pre-defined shape, in this example, a five armed "star fish" (figure 3B-E). Between figure 3E and 3F, the swarm is cut in half and the halves separated beyond communication range. The robots continue to move in accordance of the self healing rules, and in figures 3F-I, the separated swarms independently re-form the desired shape, but in a smaller scale, resulting in two smaller but still proportionate "star fish".

Through the use of each robot's individually displayed color, the swarm is capable of differentiating by displaying a pattern upon the shape which is being maintained. This pattern can be a simple coloring scheme; for example, the star fish in figure 3, has a unique color for each arm. In figure 4, there is a more complicated coloring scheme, where the swarm displays a picture of the earth. The pattern can be any arbitrary picture. As shown in both cases, if the swarm is damaged, as the robots move to reform the damaged shape using relentless self-assembly, the differentiation rules will allow the robots to change their individual colors in order to re-form the desired pattern. The new pattern will automatically re-size to fit the original pattern upon the new smaller swarm, as shown in figure 3 and 4.

With the additional capability of time synchronization, it becomes possible not only to display a differentiation pattern on the robots, but also a time varying pattern. This could be in the form of a movie, or as text that scrolls across the swarm of robots. Figure 5A shows an example of the "scrolling text" displayed on a swarm holding a circle shape. As with the self healing pattern, this time varying pattern will also recover from damage, and scale to the size of the swarm, as shown in figure 5B.

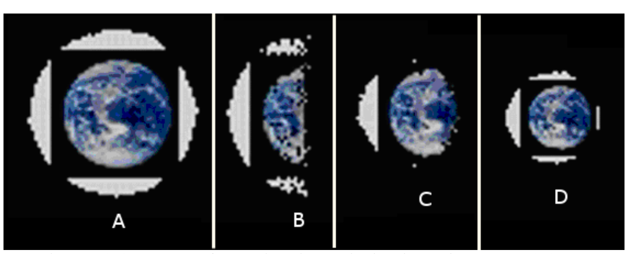

Fig. 4. (A) Pattern formation in a circle shaped swarm, (B) immediately after damage, (C-D) recovering from damage. Note: robots with no color appear to be same color as background

Figure 5. (A) Images of a circular swarm during increasing time, showing the text scrolling across the swarm. (B) Scrolling text swarms before damage (left), after damage once (center), and after damage twice (right).

V. COORDINATE SYSTEM ANALYSIS

A critical component to our relentless self-assembly is the ability for all robots to agree upon a common coordinate system, using trilateration. While trilateration is a relatively simple process, there are two main factors that affect its performance, sensor noise and sensor range. In order to measure the effects that these factors have on coordinate system agreement quantitatively, we use the total trilateration error of the entire swarm. This is calculated by summing the magnitude of the difference between the actual distance separating two robots, and the distance of their guessed locations (x_{self},y_{self},z_{self}), for all possible pairs of robots in the swarm. The lower this total trilateration error becomes, the more agreement there is between the robots guessed locations. An indication of perfect agreement between guessed locations and actual distances occurs when this total error converges to the value zero.

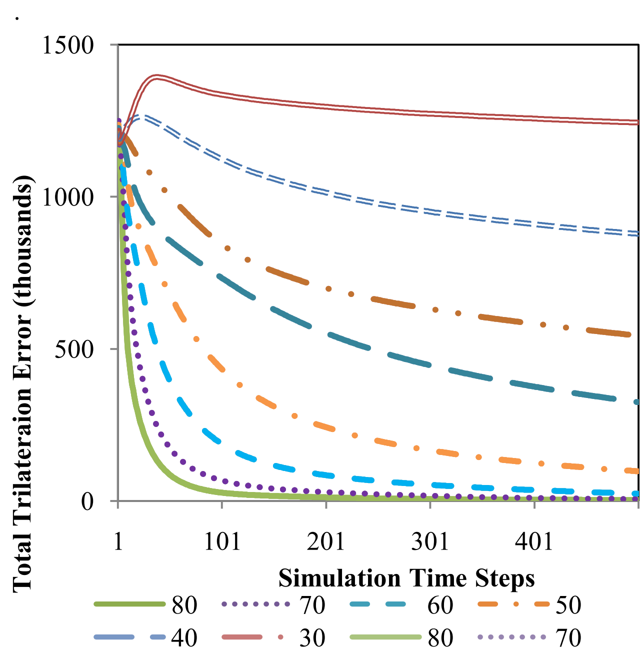

Fig. 6. Rate of Trilateration convergence for varying communication ranges.

To measure how communication range affects this convergence, we measured the total trilateration error for a swarm of 75x3 uniformly spaced robots (225 robots total) over the period of 500 time steps. This was done in sets of 20 experiments, all with the same setting for communication range. The total trilateration error was then averaged for each set. Eight sets were simulated, with communication ranges from 80 to 10. To give some meaning to these values, the communication range is in units of robot diameter, and the robots are uniformly spaced 1 robot diameter away from each other.

The results of this experiment can be seen in Fig. 6. This shows that the trilateration will converge to low values of total error, however this convergence slows as the communication range is reduced. In the case of this experiment, if the communication range is reduced below 20, then the trilateration process does not converge

To look at how sensor noise affects trilateration, we again measured the total trilateration error for a swarm of 75x3 uniformly spaced robots, this time over the period of 150 time steps. In these experiments, the sensor noise was modified by changing the standard deviation of the normal distribution that is added to the actual robot-to- robot range measurements. This was done in sets of 20 experiments, and then these 20 experiments were averaged together. All experiments within the same set had the same setting for sensor noise. Three sets were run, with a value for the sensor noise standard deviation of 0, 10, or 20 robot diameters. The results of this test can be seen in fig. 7. It shows that the trilateration process is robust to small values of sensor noise, but as the sensor noise becomes large, on the order of 20 robot diameters it will start to affect the convergence of the total error. However, this noise in the sensor model is on the order of the communication range, which is much larger than that of most sensors.

VI. CONCLUSION

The model described in this paper is motivated to have minimal artificial requirements and to be consistent with the conditions found in nature. It shows that even with simple sensors, and no global information, it is possible to control, organize, and differentiate a large swarm of simple agents. It may help us to better understand how natural organisms self-form and self-heal, and provide new insights for developing large scale artificial distributed organizations composed of simple robotic and/or software agents. Our future work includes the generalization of shape functions to cover a larger range of spatial patterns. We would also like to modify the existing model assumptions to match that of existing robotic hardware, or hardware of our design. Modification of the model assumptions could also be geared towards better emulating biological self-healing and pattern formation, in order to better understand natural phenomena. For more information, and videos of this work visit www.isi.edu/robots

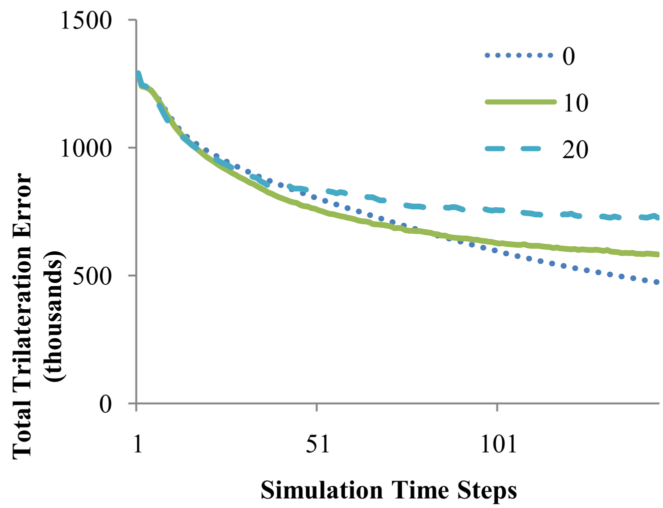

Fig. 7. Rate of Convergence with varying standard deviation for sensor noise model for communication range=60.

ACKNOWLEDGMENT

We would like to thank the students at the Polymorphic robotics lab for their assistance with this work.

REFERENCES

[1] M. Yim, W. Shen, B. Salemi, D. Rus, M. Moll, H. Lipson, E. Klavins, G. Chirikjian. Modular Self-Reconfigurable Robot Systems -- Challenges and Opportunities for the Future. IEEE Robotics and Autonomation Magazine, March():43–53, 2007.

[2] W. Shen, M. Krivokon, H. Chiu, J. Everist, M. Rubenstein, and J. Venkatesh. Multimode Locomotion for Reconfigurable Robots. Autonomous Robots, 20(2):165–177, 2006.

[3] H. Bode. Head Regeneration in a Hydra, Developmental Dynamics 226:225-236, 2003

[4] D. Martinez. Mortality Patterns Suggest Lack of Senscence in Hydra, Experimental Gerontology, Vol. 33, No. 3, pp. 217-225, 1998.

[5] M. Tolley,V. Zykov, H. Lipson,D. Erickson. Directed Fluidic Self-Assembly of Microscale Tiles. Micro-Total Analysis Systems (uTAS), Tokyo, Japan, Oct.2006

[6] J. Bishop, S. Burden, E. Klavins, R. Kreisberg, W. Malone, N. Napp, and T. Nguyen. Self-organizing programmable parts. In International Conference on Intelligent Robots and Systems. IEEE/RSJ Robotics and Automation Society, 2005.

[7] A. Kondacs, *Biologically-inspired Self-Assembly of 2D Shapes, Using Global-to-local Compilation*, International Joint Conference on Artificial Intelligence (IJCAI), 2003.

[8] J. Suthakorn, B. Cushing, G. Chirikjian. *An Autonomous Self-Replicating Robotic System.* Proceedings of the 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, 2003

[9] H. Lipson. (2005) *Self-reproducing machines*, Nature Vol. 435 No. 7038, pp. 163-164

[10] K. Stoy, R. Nagpal. *Self-Repair Through Scale Independent Self-Reconfiguration.* International Conference on Robots and Systems, (IROS), Sendai, Japan. 2004.

[11] J. Cheng, W. Cheng,R. Nagpal, *Robust and Self-repairing Formation Control for Swarms of Mobile Agents*, National Conference on Artificial Intelligence (AAAI '05), July 2005.

[12] R. Grabowski, P. Khosla. *Localization Techniques for a Team of Small Robots*. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01).

[13] I. Borg, *Modern Multidimensional Scaling: Theory and Applications.* Springer Series in Statistics. 2005.

[14] G. Box, M. Muller, *A Note on the Generation of Random Normal Deviates.* The Annals of Mathematical Statistics, Vol. 29, No. 2 pp. 610-611.