2D Simulation of DiscBots

MAE 4900: Independent Research

Vinay Rajur(vsr7)
Advisor: Hod Lipson

December 20, 2013

Creative Machines Lab, Cornell University

TABLE OF CONTENTS:

I. Abstract

This research project is a simulation in programmable matter with potential applications in modular robotics. This project is an initial attempt at developing a simulation for a collection of variable-radius, 2D disc-shaped robots (DiscBots). DiscBots are defined as circular robots with control over a single degree of freedom: radius size. Additionally, DiscBots have an attractive force between them, thus allowing them to form clusters. The goal of this research is to observe any collective group behaviors and gain insight into the feasibility and functionality of potential physical implementations of this system. Studying a particular case where individual DiscBots continuously oscillate radius size at a random frequency, the failure modes of this system were studied, and the uniqueness of these failure modes was observed to yield a particular robustness in the functionality of the ensemble.

II. Goals and Motivation

This research is focused on prototyping and studying the design for an implementation of a Self-Assembling Modular Robot (SAMR) composed entirely of DiscBots. Given that we understand the behavior of the individual DiscBots, we are interested in determining the collective behavior of the entire system. Specifically, this research aims to address the following questions:

1. What limits does this DiscBot-based design place on the behavior of the SAMR? How do the physical limitations of the constituent DiscBots translate into the larger system?
2. What are the problems with this design?
3. How does this design allow the SAMR to respond to damaged parts and other system failures?
4. What potential uses are there for this system and what would be the significance of building an implementation of this robot?

There are many other questions to be answered before this form of SAMR can be fully understood; this project, however, is meant to serve as a starting point and a foundation for future research and study.

III. Background and Literature Review

<u>Programmable Matter:</u>

Programmable matter is an emergent field of study focused on matter with the ability to alter its physical properties in response to both user input and external stimuli in a predefined way. There are various forms of programmable matter, ranging from microscopic sensors and displays to large scale, self-assembling systems; thus, the defining characteristic of this type of matter is its ability to interact with and actively respond to other matter. Fundamentally, this field pertains to the study of materials with the inherent ability to process information.

Although many of the implementations of programmable matter are relatively simple (being composed of elements typically restricted to just a few degrees of freedom), these systems are significant and provide the necessary means in allowing technology and information to cross from the virtual realm into physical or synthetic reality. This crossover allows technology to be simpler to understand and more intuitive to manipulate, and as a result, there are many potential uses for programmable matter. One of the main uses is the concept of Distributed Processing (or Ubiquitous Computing), which is characterized by embedded computers throughout our physical environment, which constantly interact with and process information, yet remain largely invisible to us within the surroundings [7]. The remaining uses for programmable matter fall into three categories, known as the three paradigms of Ubiquitous Computing: Dusts, Skins and Clays [3]. Dusts are generally composed of Micro Electro Mechanical Systems (MEMS), devices such as sensors, displays and processors, which are built on a scale ranging from micrometers up to millimeters. Skins are defined as intelligent surfaces, such as windows, walls, fabrics and other object surfaces, which are embedded with processors, sensors and displays (essentially a 2D extrapolation of Dust). Lastly, clays are 3D ensembles of Dusts that can be used to create arbitrary, intelligent objects. SAMR's in particular generally fall under this last category of programmable matter.

Self-Assembling Modular Robots:

Self-Assembling Modular Robots are autonomous systems with the ability to alter and reconfigure their physical properties. SAMR's are composed of a number of smaller (and simpler) robots, much like the manner in which a body is composed of cells. Within a particular SAMR, there can be several different types of cells (though this number is usually limited to one or two), but the entire system can consist of a vast number of these cells.

SAMR's are of great interest because, despite the simplicity of their constituent cell bots, collectively the ensemble can display relatively complex behavior, which derives from the nature of the cells. With a nearly infinite variety of designs to use for the cell bots, it is clear to see that SAMR's themselves can display an extremely diverse set of behaviors. Previous research on SAMR's has studied various types of cell bot designs including folding chains [1], locomotive, cube-shaped bots [5] and autonomous, reconfigurable bots that allow for self-assembly [6]. A common trait, however—perhaps even limitation—among much of the past research is the lack in range of motion for the particular cell bot design. Although this limitation in mobility diminishes as the system scales, artifacts in the ensemble behavior (such as inherent symmetries in the behavior, etc…) may form as a result of the cell bot limitations; this is a topic that has not been thoroughly studied.

Therefore, this research project proposes a cell bot design that allows for a fully continuous range of 2D motion using a single class of cells, DiscBots. Additionally, the cells in this design are given control over their individual size, a unique parameter that has not as of yet been explored. Thus, by studying this novel SAMR design, this research seeks to answer the questions proposed in Section II relevant to this design, while gathering information applicable to SAMR's in general.

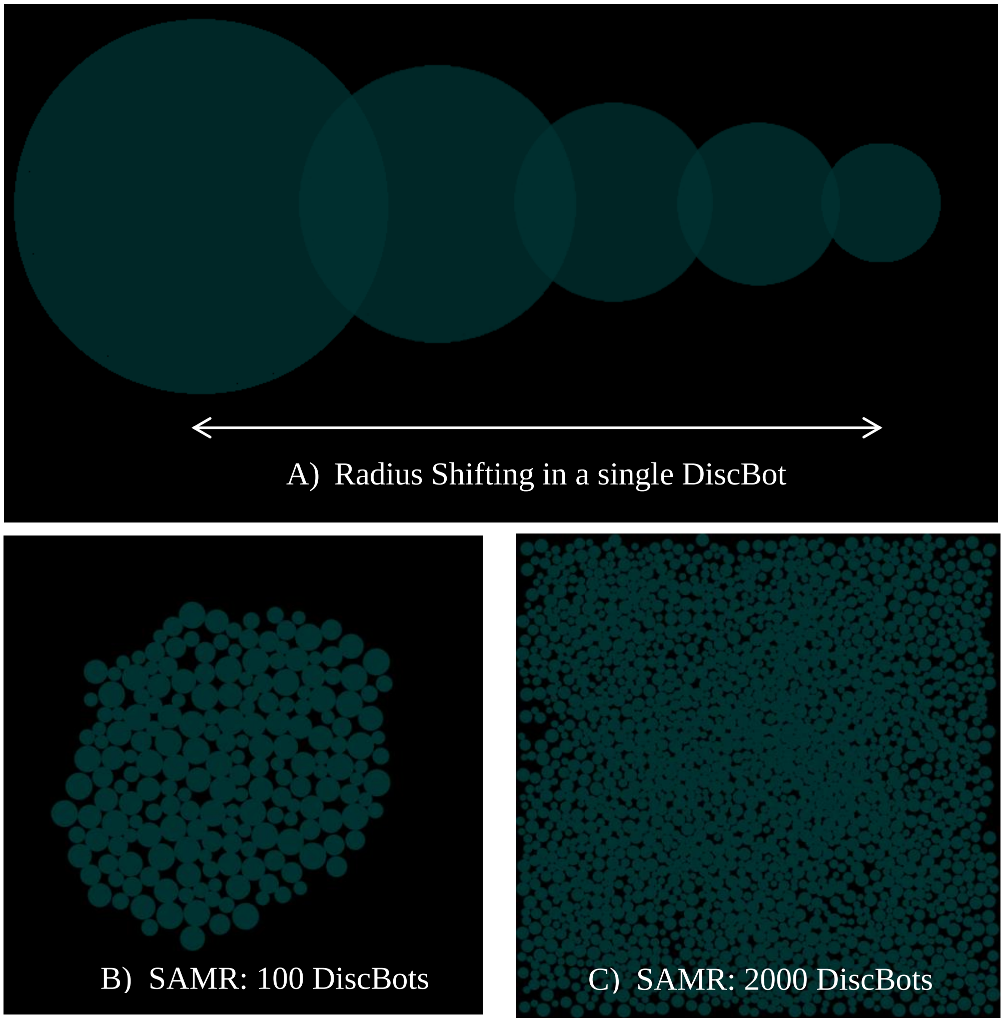

Figure 1: The SAMR studied in this project. A) depicts a single DiscBot and its radius shifting capabilities while B) and C) depict SAMR's of various sizes composed of DiscBots.

IV. Method/Technical Approach

Ultimately the goal of this research is to understand how a large-scale implementation of a SAMR[1] would behave under arbitrarily complex conditions. However, rather than trying to study the SAMR under all conditions (and combinations of conditions) that it could be exposed to, it is much more efficient to determine the range of states that the SAMR can occupy, since

[1] From here on, SAMR will refer the DiscBots-based design previously mentioned in Section III

any response the SAMR can display must lie within one of these states. Additionally, because the SAMR is composed of identical parts, any part of the SAMR can be decomposed into identical smaller components, and so the SAMR is self-similar. This self-similarity simplifies the problem of determining the range of SAMR states, because there exists a fundamental state from which any arbitrary state can be constructed through linear combination: a basis in the state space of the SAMR.

The choice of basis is not unique so there some freedom in defining a sufficient basis state and in this study, the basis state is defined as a collection of 100 DiscBots oscillating sinusoidally with arbitrary frequency and phase[2], where using this state as a basis state is analogous to using sinusoids as a basis for constructing arbitrary functions in Fourier Series analysis. By defining and studying this basis state, we can understand the range of behaviors that the SAMR is capable of.

Lastly, an important subset of behaviors that this research studies, relates to the system's response to failures, which are important in determining the stability of the SAMR. For this system, there are two classes of failures: A) unresponsive components and B) incorrectly responding components, but for this research only Type A failures are studied. In summary, using this information, the following steps were taken in order to gain insight into the full range of behaviors the SAMR can display:

1. Implement a simulation of 100 DiscBots sinusoidally oscillating radius size.
2. Vary the oscillatory behavior of the DiscBots and study the SAMR behavior.
3. Run tests with "broken" DiscBots (i.e. DiscBots frozen at a specific radius, and observe the system's response).

V. Implementation and Results

<u>Implementation:</u>

The SAMR implementation studied in this research uses DiscBots as the constituent cell bots and this section will discuss the properties of the DiscBots and the simulation in more detail.

DiscBots:
As stated before, there are three defining features attributed to the DiscBots studied in this design:

1. DiscBots are circularly symmetric robots with control over just one degree of freedom: control over radius size.
2. They are free to move in any x and y direction but are constrained to the x-y plane.
3. DiscBots are defined as having some attractive force between them.

In the simulation (see Appendix A for source code), DiscBots are modeled as identical 2D circles (class Ball in the code) with a parameters controlling radius size, oscillation frequency and

[2] Upper bounds were imposed on these values, however, for the sake of simulation.

phase. In the tests run, the radius length for each bot oscillates as a sinusoidal function, with frequency and phase assigned randomly to each bot. Additionally, in the simulation, an attractive force was assumed to follow an inverse square law, where force was of the form: $F = \frac{k}{r^2}$. The parameter k, governs the magnitude of the force and the sign determines the direction, where the force is attractive[3] for $k < 0$. This form of equation was chosen for the force in order to model the simplest type of attractive force that could be used in a physical DiscBot. At the macroscopic scale this force could be implemented using ferromagnetic bars around the bot's circumference, or at the microscopic level, it could be implemented by replacing the magnetic force with an electrostatic one; in both cases, the interaction force between the DiscBots would follow the $\frac{1}{r^2}$ form.

SAMR:

The SAMR tested in this research was implemented as a collection of 100 DiscBots oscillating at random frequencies and with random. These parameters controlling the oscillations in radius were implemented as fixed throughout the simulation, but in future research, location-based radius oscillations and oscillations in response to neighboring DiscBot behavior are two areas this study can be expanded on. Lastly, the SAMR is constrained to a 2D surface (just as the individual DiscBots are), thus, there is no piling up of DiscBots in the z-direction.

Simulation:

The code for this simulation is written in an open source, Java-based language (and IDE) called Processing, which was designed and is maintained by Casey Reas, Benjamin Fry and their development team. Additionally, the example sketch "Bouncy Bubbles" on the Processing website (which is based on code written by Keith Peters) was used as the physics engine modeling the collisions between bots in the simulation. Additional features and capabilities where implemented on top of this framework and these additions include:

- Modifications to the class Ball to add DiscBot functionalities, such as parameters for the radius size, oscillation frequency and offset as well as a broken state. Additional helper functions were added as well.
- Inclusion of extra forces in the mechanics of the bots' movements such as an attractive force, a frictional drag force, and static friction when in contact with the boundaries.
- Synchronous update sequence
- Other user input based code to allow the simulation to be paused, and to enable interaction and modification of the simulation while it is running.

<u>Results:</u>

The results from running the simulation with 100 DiscBots each oscillating at a random frequency and phase are shown in Figure 2 below. The first frame of Figure 2 displays the state of the SAMR when the DiscBots are initialized randomly and it depicts the consolidation

[3] Additionally, this SAMR model can be tested using a repulsive force between DiscBots instead. The resulting system, however, would not be fully cohesive, and is not relevant for our purposes.

process, by which the SAMR reaches its equilibrium state. It is important to note that the DiscBots are initialized with no velocity and that their movement is generated entirely by the

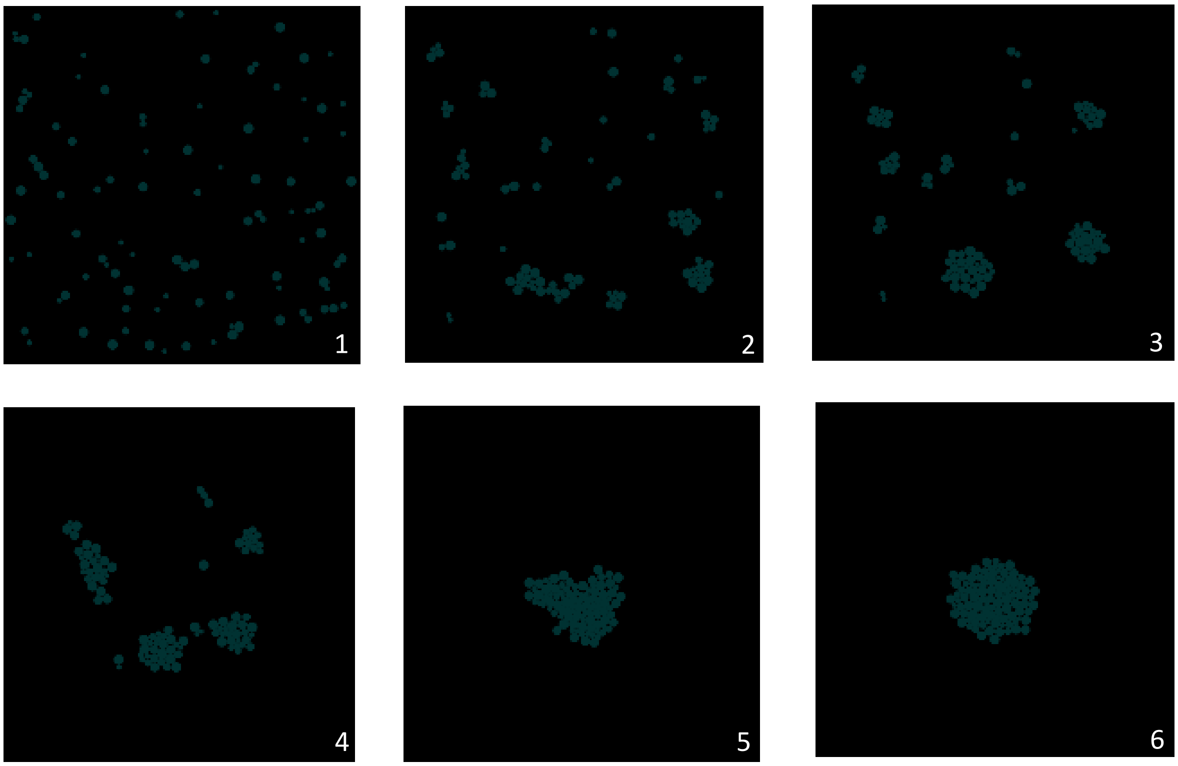

Figure 2: Formation process for a SAMR (frames were captured every 5 seconds). Notice the self-similarity of the initial clumps which compose larger clusters.

attractive forces between the individual bots. A result of this fact is that every state of the SAMR is localized around its center of mass, and additionally that the SAMR has a very limited ability to move on its own.

Although this behavior displayed by the SAMR is relatively simple, this formation process displays important behaviors and properties of this SAMR which provide the foundation for the more complex behaviors that the system is capable of. Specifically, the equilibration process shown in Figure 2 is a prominent example of the self-similarity property inherent to this SAMR: we can clearly see how the final equilibrium state is composed of relatively identical clumps and we can easily extrapolate to see the process by which the clump shown in frame 6 of the figure can itself combine to create even larger clusters. This property of the SAMR is, in fact, very useful, because it gives the SAMR regenerative properties, making it a relatively stable system.

Additional stability in the system comes from the SAMR's response to failures, one form of which is unresponsive components (Type A failures). In this simulation, this was tested by "freezing" particular bots at some radius and observing the system's response. The resulting behaviors of the SAMR are described by Figures 3-5.

A key factor affecting the behavior of the system that became apparent during testing was the size the broken component was stuck at. Specifically, it is the relative size of the disabled component compared to the maximum size a DiscBot can assume which affected the SAMR’s response; the bots stuck at larger radii tended to be expelled towards the boundaries, while bots stuck at smaller radii tended to remain towards the inside of the SAMR. Figure 3 shows the comparison of bots disabled at the minimum radius compared to bots disabled at the maximum radius. For both cases, eight independent simulations of broken DiscBots were run

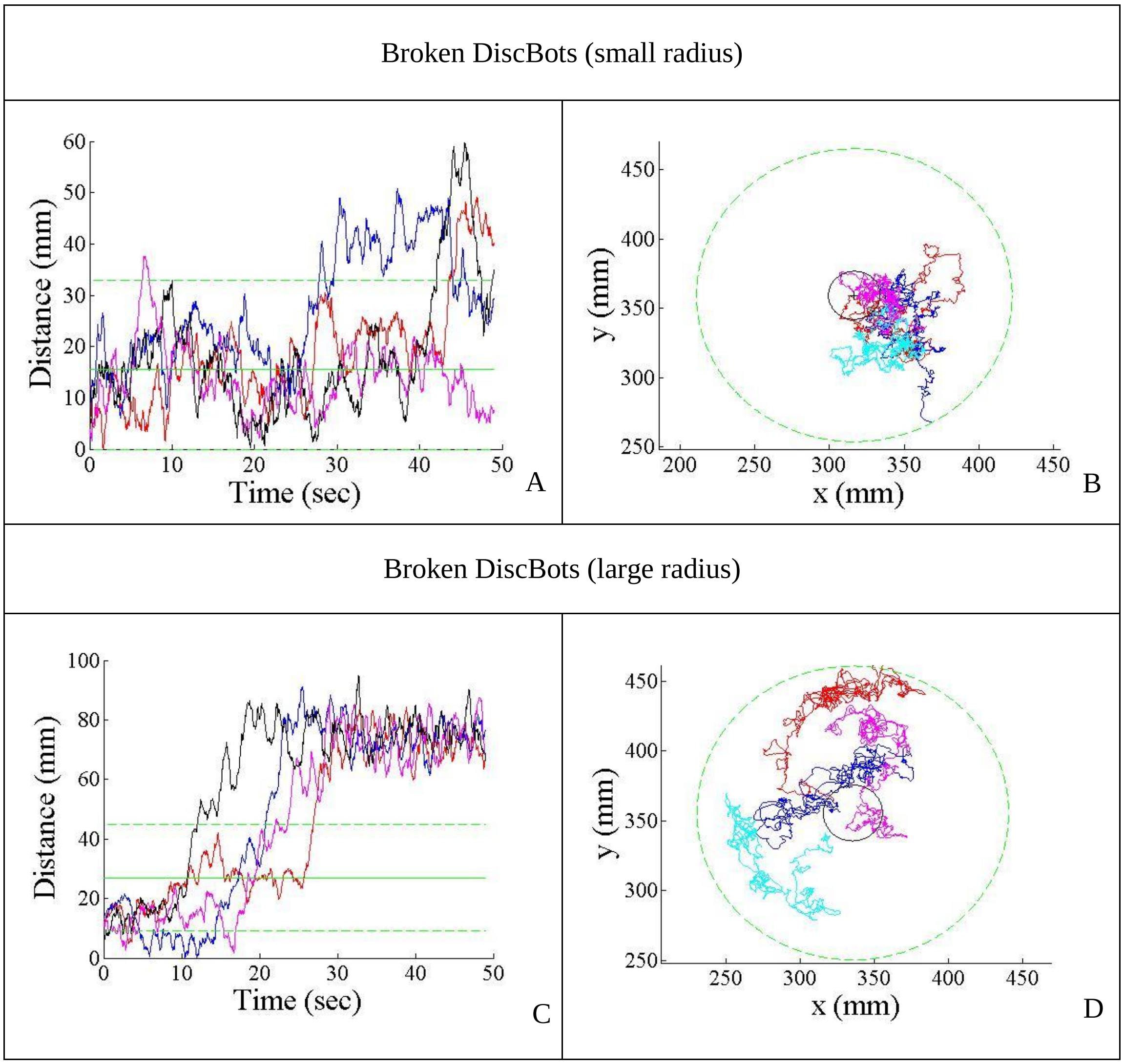

Figure 3: Plots A) and C) depict Distance vs. Time graphs for broken bots (with respect to the center of the SAMR) for both small and large radii, respectively while B) and D) show plots of the different trajectories taken by the bots in the x-y plane. In the Distance vs. Time graphs, the solid green line represents the average distace of a DiscBot (not broken) from the center of the SAMR, while the dashed green lines span a 2σ interval for this average distance, corresponding to a 95% condfidence interval. The black circles in the x-y plots again represent the average distance of working bots from the center, while the green dashed lines represent the maximum upper bound on the distance of any bot from the center.

(depicted by the 4 curves shown in each individual graph). For each simulation, a single bot was frozen (at either a large or small radius) and its position relative to the SAMR was measured. The results of these simulations show that for large radii, there is a statistically significant tendency for the broken bots to be pushed towards the boundaries, as compared to bots frozen at small radii, which tend to remain within the expected range of distances from the center for a normal, functioning DiscBot. In other words, the SAMR filters out larger bots and tends to concentrate smaller bots at its center.

This filtering behavior is due to only two of the factors present in the system, the attractive binding force and the oscillatory behavior, and a simple and effective analogy for understanding this is the concept of buoyancy. In this system, the attractive forces between the DiscBots produce a net radial force directed towards the center of the SAMR, similar to a gravitational force. As a result, the smaller, more compact (i.e. denser) bots are drawn in towards the center, due to the fact that they can align themselves spatially closer to each other than the large bots can. Thus, the larger bots are displaced and “float” towards the outer boundary of the SAMR. The transport process allowing this movement is caused by the oscillation of the DiscBots, which leads to a random walk behavior in bot trajectories that is similar to the Brownian motion underlying the process of diffusion. Thus, by looking at the behavior of these DiscBots through the lens of buoyancy and diffusion, it is easy to see that the attractive force is the driving force behind this behavior and the radius oscillations are the underlying mechanisms enabling it.

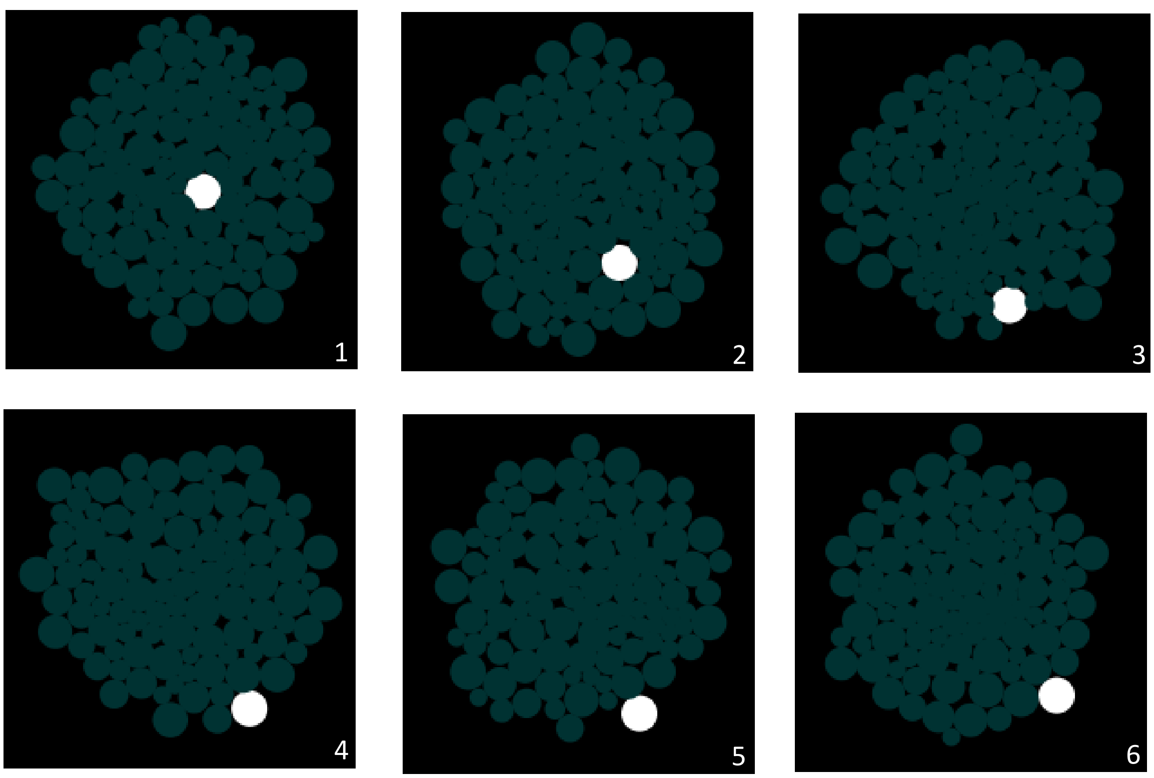

Figure 4: The transport process by which a single broken DiscBot (large radius) diffuses towards the outer boundaries of the SAMR. Frames were captured every 10 seconds.

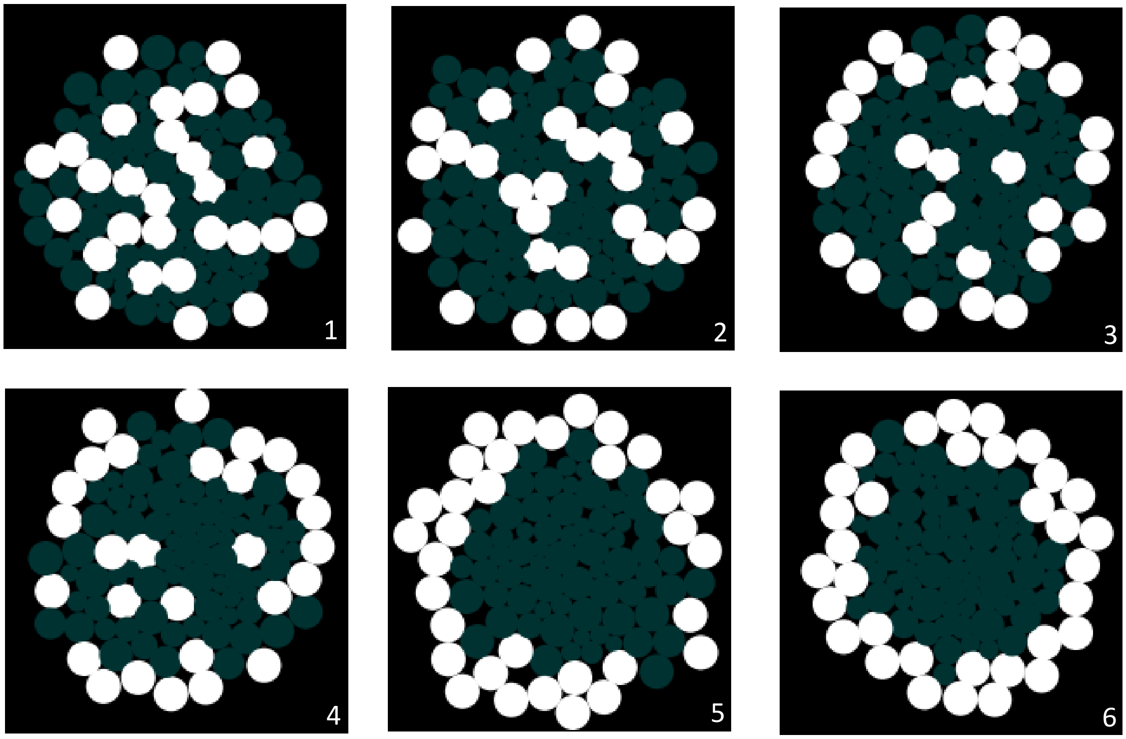

Figure 5: The diffusion transport process for a collection of many disabled DiscBots, and the resulting exoskeleton like equilibrium state. Frames were captured every 5 seconds.

Lastly, another interesting observation, which follows from this diffusion-like transport, is that the filtering process works even when multiple bots are disabled (so long as a sufficient number of bots are still oscillating[4]). The result is the formation of an exoskeleton or "skin" along the periphery of the SAMR (see Figure 5 above). This behavior has potential uses in shielding the SAMR and also provides a means of recycling damaged or dead DiscBots (similar to the way skin forms from cells). Again, however, this behavior is reliant on the same two factors (an attractive force and radius oscillations) in order to enable this type of response.

VI. Conclusions

The primary goal in conducting this research is to study the behavior of a DiscBot based SAMR design, and essentially to provide an overall assessment on the feasibility and functionality of this design. Thus, the fundamental basis state for this SAMR design was studied, which was defined as a collection of 100 DiscBots oscillating in size with arbitrary frequency and phase.

[4] From running various simulations, $\frac{1}{6}$ the total number of bots in the system was empirically determined as a lower bound for the number of functioning DiscBots to be present in order for the diffusion process to work. Below this value, isolated pockets of functioning bots begin to form, which inhibits diffusion from occurring.

Starting the analysis with the limitations of this design, there are several to consider. The primarily limitation of this design is the lack of an effective form of locomotion, deriving from the lack of mobility present in the individual DiscBots. Although potential forms of motion can be imagined (such as by expanding and pushing off surfaces), the practicality and efficiency of these forms of movement are yet to be determined, and with respect to this study, the SAMR is effectively immobile. Another limitation of the SAMR is on its range of physical configurations, which, again, is limited by the DiscBots' lack of mobility in addition to their inability to control the binding force between them. As a result of these handicaps, the SAMR will always tend towards rotationally symmetric alignments (which generally manifests as a hexagonal, honeycomb shape due to the circular nature of the DiscBots), reducing the complexity of available states the SAMR can occupy. Essentially, the simplicity of the DiscBots translates into reduced degrees of freedom in the collective group.

This perhaps oversimplicity in design could also be seen as a potential issue as well, reducing the SAMR's responsiveness to external stimuli. Lack of mobility is another potential issue, as is the individual strength of DiscBots[5]. Despite these problems and limitations, however, the SAMR itself is still relatively adaptable and its simplicity, in fact, enhances its stability.

One form of the SAMR's stability comes from its regenerative abilities, due to its self-similarity, however, many of its more fundamental forms of stability come from the SAMR's ability to respond to damaged parts and system failures. For Type A failures with unresponsive bots, the SAMR has the inherent ability to expel these bots towards the boundaries simply by the Brownian motion of the functioning DiscBots. There are issues to consider, however, since this process only works for bots that are disabled at relatively large sizes. Additionally, the process is statistical and so the time to the boundary for a broken bot is not certain. The process is also dependent a relatively strong attractive force and relatively high oscillation frequencies to keep the transport cycle sufficiently fast. Despite these caveats, however, this behavior is still important because it allows for the formation of an exoskeleton of disabled and damaged bots, which can serve protective purposes for the SAMR. Additionally, one way in which the radius size issue could be solved would be by implementing a default trigger that forces a broken bot to expand to its maximum size, thereby allowing the transport process to occur.

The other form of failure, Type B failures, consists of bots that, rather than not responding, respond incorrectly or unpredictably. This form of failure was not specifically addressed in this research; however, some generalizations about the system response can be made due to the existence of two limiting factors that can diminish the scope of the impact from both types of failures. The first diminishing factor is the relative radius of the broken bots compared to the radius of the working bots, where a smaller radius size reduces the impact of the broken bot. This can be seen from the Type A failures, where even though smaller broken bots remain within the center of the SAMR, they do not significantly alter the behavior of the system (whereas the exoskeleton forming larger bots have noticeable impact on the behavior of the system). The second diminishing factor is the relative number of damaged bots compared to the relative number of working bots, where statistical effects will generally allow for a functioning SAMR, even with an appreciable number of broken bots. Thus, as we can see the relative

[5] Due to the primarily expanding-related functionality of the SAMR, the system of DiscBots is susceptible to scenarios with high pressures. Thus, DiscBots will need to be structurally strong and robust enough to cope with these strains.

simplicity and lack of highly ordered states is actually a key factor in the robustness of this design and the source of its resistance to malfunctions.

Thus, a DiscBot-based implementation of a SAMR is a feasible concept due to its stable and robust nature. Furthermore, with a physical implementation of this design, the potential applications of this design could be explored: for example, in 3D, this SAMR could be used to apply an expansive force inside arbitrarily shaped cavities (like a jack or a lever), or in 2D, this SAMR could act as a dynamic coating or surface (which could be coupled with additional features such as displays to make screens). These uses are still relatively far off, however, and only with additional advances in the study of SAMR's, programmable matter and nanotechnology can this concept be implemented in its fullest form. Though the technology is not yet ready, the feasibility and functionality of this design looks promising based on the results of the analyses suggested in this study.

VII. Future Work

This study was meant to serve simply as an initial attempt at simulating and studying the design of this DiscBot based design for a Self-Assembling Modular Robot, and there many important aspects of this topic that were not covered by this current study. In particular one major topic that was not covered was the behavior of this system when it is governed by location-based rules or neighbor-based rules, and this is the next major point I hope to address. A second point, which I did not have time to cover with this research was the implementation of fixing strategies to solve Type A errors. Accordingly, another important topic to study is Type B failures and to observe specifically any solutions available that the system can provide. Lastly, there are elements of the simulation itself that could be improved and made more realistic, for example including adding random noise (error) during the DiscBot update sequence.

VIII. Bibliography

[1] Knaian, Ara N., Kenneth C. Cheung, Maxim B. Lobovsky, Asa J. Oines, Peter Schmidt-Neilsen, and Neil A. Gershenfeld. "The milli-motein: A self-folding chain of programmable matter with a one centimeter module pitch." In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 1447-1453. IEEE, 2012.

[2] Toffoli, Tommaso, and Norman Margolus. "Programmable matter: concepts and realization." *Physica D: Nonlinear Phenomena* 47, no. 1 (1991): 263-272.

[3] Poslad, Stefan. *Ubiquitous computing: smart devices, environments and interactions*. Wiley. com, 2011.

[4] Goldstein, Seth C., and Todd C. Mowry. "Claytronics: A scalable basis for future robots." (2004).

[5] Yim, Mark, David G. Duff, and Kimon D. Roufas. "PolyBot: a modular reconfigurable robot." In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 1, pp. 514-520. IEEE, 2000.

[6] Murata, Satoshi, Haruhisa Kurokawa, and Shigeru Kokaji. "Self-assembling machine." In *Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on*, pp. 441-448. IEEE, 1994.

[7] Weiser, Mark. "Hot topics-ubiquitous computing." *Computer* 26, no. 10 (1993): 71-72.

IX. Acknowledgments

I would like to thank Professor Hod Lipson for being my advisor and helping me work through the process of developing this simulation, guiding me through tough spots and troubleshooting the issues with me. Additionally, I would like to thank Casey Reas, Benjamin Fry and the entire Processing development team for their impressive amount of effort in developing a simple and easy to use graphics development library and programming environment. Lastly, I would like to acknowledge the work of Keith Peters, whose code provided a simple and realistic physics engine for me to develop my simulation on.

X. Appendix

A. Source Code Listings:

Source Code: broken.pde

This code simulates a collection of 100 DiscBots with broken states that can be toggled by clicking the bot or setting its broken field to true. The code written in Processing and simulates the interactions of the collection of DiscBots under attractive forces, frictional forces and collisions with other bots. The code also can record and track the positions of broken bots (best to track one at a time) as well as the positions of every bot, which are output to several files upon closing (this information was processed in Matlab to produce the figures shown in the report).

The following is the basic algorithmic pseudocode for broken.pde:

```
Initialize array of balls with location, radius size, frequency and phase
Draw() {        //Analogous to main() in Java/C++
        For each ball, B
                If B is not broken, calculate its size (according to sinusoidal function)
                Else set the size of B to the size of a broken bot

                Calculate the net force on B (attractive forces, friction, collision, etc…)

        For each ball, B
                Update position of B
                Draw ball B
                If B is broken record its location

        For each ball B,
                Record B’s location
}
```