5-2-2007

The Effect of Collision Avoidance for Autonomous Robot Team Formation

Mark H. Seidman
Rochester Institute of Technology

Shanchieh Jay Yang
Rochester Institute of Technology

Recommended Citation

Mark H. Seidman, Shanchieh J. Yang, "The effect of collision avoidance for autonomous robot team formation", Proc. SPIE 6561, Unmanned Systems Technology IX, 65610M (2 May 2007); doi: 10.1117/12.718071; https://doi.org/10.1117/12.718071

The Effect of Collision Avoidance for Autonomous Robot Team Formation

Mark H. Seidman[a] and Dr. Shanchieh J. Yang[b]

[a]Rochester Institute of Technology, 46 William St Apt C, Newport, RI, USA;
[b]Rochester Institute of Technology, 83 Lomb Memorial Drive, Rochester, NY, USA

ABSTRACT

As technology and research advance to the era of cooperative robots, many autonomous robot team algorithms have emerged. Shape formation is a common and critical task in many cooperative robot applications. While theoretical studies of robot team formation have shown success, it is unclear whether such algorithms will perform well in a real-world environment. This work examines the effect of collision avoidance schemes on an ideal circle formation algorithm, but behaves similarly if robot-to-robot communications are in place. Our findings reveal that robots with basic collision avoidance capabilities are still able to form into a circle, under most conditions. Moreover, the robot sizes, sensing ranges, and other critical physical parameters are examined to determine their effects on algorithm's performance.

Keywords: Autonomous robots, Shape formation

1. INTRODUCTION

Shape formation may be considered as a starting point for the robot team to perform cooperative tasks. Applications of cooperative robot teams include intruder detection and containment of chemical spills or forest fires. A robot team formation problem is one that seeks autonomous actions taken by individual robots so as to form into a specific shape. A typical shape considered in the literature is a circle or n-polygon, where n is the number of robots forming the shape. Existing work on robot team formation focuses on theoretical analysis where real-world factors, such as the size of the robots, are not considered. It is unclear whether proposed theoretical algorithms are valid or perform well in real-world settings. This research reviews an autonomous robot shape formation algorithm and examines the effect of different sensing and collision avoidance technologies as it is critical for robots to maneuver yet not colliding with each other.

Three different collision avoidance scenarios are presented and analyzed when accounting for the robot sizes. Each scenario models robots with different sensing and collision avoidance capabilities. The first model represents the case where the robots, due to their lack of imminent collision detection, need to conservatively calculate potential collisions before they move. The second scenario models the case where the robots can detect collision and stop just before it happens. In the third scenario, explicit communication is also allowed between the autonomous robots to resolve collision situations. The original ideal algorithm, which does not consider the problem of collisions, is compared against the algorithm run with the three collision avoidance scenarios. Note that most theoretical research treats the robots as an infinitesimal point in the simulation and, thus, no collision will ever happen. This research conducts simulation experiments to present a middle ground between theoretical research and real-world implementation.

Our experiments further examine the effect of various factors, including the number of robots, the robot size, the sensing range, and algorithm-specific parameters. These examinations shall provide insights on how one may design robots to facilitate autonomous robot team formation. A baseline autonomous circle formation algorithm, proposed by Andrew Michael,[9] is selected for this research. Michael's algorithm has been shown to have the robots form into a circle successfully for most cases, but with the assumption of zero-size robots. This research will reveal whether those successful formations are still true in the presence of collision avoidance.

Further author information: (Send correspondence to M.H.S.)
M.H.S.: E-mail: Mark.SPIE@SeidmanOnline.net,
S.J.Y.: E-mail: Jay.Yang@rit.edu

Unmanned Systems Technology IX, edited by Grant R. Gerhart, Douglas W. Gage, Charles M. Shoemaker,
Proc. of SPIE Vol. 6561, 65610M, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.718071

2. ROBOT TEAM FORMATION ALGORITHMS

Several different shape formation algorithms have been developed and tested by different researchers. Two main approaches have been followed, centralized and decentralized. In a centralized approach there is an entity that has knowledge of the entire team of robots and will direct the robots where to move. This work focuses on decentralized approaches, where the robots make their own decisions based on the information that they are able to obtain individually. A decentralized approach yields a more robust system but requires more highly intelligent robots.

Several general assumptions were made about the robots and the environment in this research. The robots are all completely identical. All the robots in the simulated system run the same algorithm and none have any discriminating features. The robots are assumed to have a synchronized global clock, the same size, and move at the same speed. The environment is assumed to be obstacle free. If a robot detects an object with its sensors, then that object is assumed to be another robot. This is something that would need to be overcome when implemented in the real world. However, the present study will keep this assumption and focus on the robot's capability to avoid collisions with other robots.

2.1. Related Works

In the arena of decentralized circle formation algorithms, most work done is theoretical in nature; however, a few small scale systems have been implemented in a physical system, such as that by Matarić et al.[1] Their approach uses a video processing technique where each robot is uniquely identified by a color and that color is only used on one robot in the system. Formations are made based on determining a robot's position based on the position of a robot of a specific color, which is called that robot's 'friend' robot. A robot will have only one friend robot, but a robot may be a friend to more than one other robot. Since video processing is used in this work, it is unlikely that it will be able to scale well to a system with many robots.

Suzuki and his research team are considered the pioneers in the field of shape formation in cooperative robotics. Their first work[2] was found to use a somewhat primitive approach. As with all their works, global scanning is required in this algorithm. The algorithm works as follows. A robot, R, will continually scan for the robot farthest from itself, denoted R', and closest to itself, denoted R'', and calculate the distance between those two robots, which is denoted as d. Also, let D represent the desired approximate diameter of the circle to be formed, and δ be a small constant. Robot R will base its movement on the following three cases:

- If $d > D$, then robot R will toward R'.
- If $d < D$ - δ, then R moves away from R''.
- If D-$\delta \leq d \leq D$, then R moves away from R''.

The small constant δ is used to help distribute the robots uniformly on the perimeter. There are some problems with this algorithm. One problem that the authors point out is that it is shown not to work given some specific symmetrical layouts. In some other cases the formation resembles a triangle with slightly curved sides, and not a circle. Despite the use of δ, the robots will still not evenly spread out on the perimeter of the circle. Suzuki and his research team have published other papers that expand from their original work.[3–5]

Prencipe et al have done considerable work in cooperative robotics and pattern formations.[6, 7] This research requires global scanning and requires each robot to have a compass. The robots used here are thought of as "weaker" than the ones used in Suzuki et al's work since the robots do not operate in a synchronous manner. Different algorithms were developed, each giving the robots more knowledge of the environment, to see how much information is needed for the system to still form the desired pattern. They investigate the formation of both symmetrical and asymmetrical pattern formations.

A different approach to the circle formation problem was attempted by Défago et al.[8] The authors use the Smallest Enclosing Circle (SEC) and Voronoi Diagrams in their algorithm. In a field of randomly deployed robots, the SEC is defined by circle that goes through three robots so that the rest of the robots in the field are contained within that circle. Once the SEC is known, then each robot will compute its Voronoi Cell. A robot is then limited to movement within its Voronoi Cell and each iteration the robot will move toward the perimeter of the SEC. Once all of the robots are on the perimeter of the SEC, then the robots will move to spread out and become evenly distributed on the perimeter. Even though it's not mentioned in the paper, collision avoidance is built in to the algorithm. When robots are moving, they cannot collide with another robot since a robot is limited to movement within its Voronoi Cell, which is a unique area for each robot.

2.2. Michael's Algorithm

The base algorithm used for this research was developed by Andrew Michael.[9] One of the main reasons Michael's algorithm was chosen is because it doesn't require a global scanning range unlike most other works in this area. The goal of the algorithm is to have a group of randomly deployed robots autonomously form into a circle. The actual shape the algorithm has the robots form is actually an n-sided polygon. Each robot attempts to form a specified angle, θ, between itself and its two neighbors. This angle varies depending on the total number of robots in the system. When all of the robots obtain the desired angle between their respective neighbors, then the robots will appear to be in the shape of a circle.

2.2.1. System Parameters

There are several variables that have an impact on the robot formation algorithm. They include the robots' size, the scanning range, sr, the step sizes, si and sf, and the number of robots in the system, n. The robot size is assumed to occupy a circular space defined by a radius, r. The scanning range indicates the reachable range of the sensors that will be used to detect other robots. The robot is assumed to be able to detect all robots within the circle defined by its scanning range. The scanning range cannot be too small or the robots will cluster into smaller groups instead of forming a single circle. There are two step sizes used by the base algorithm, an initial step size and a final step size. The initial larger step size is used to help the robots to get to the perimeter, and then the smaller step size is used to help them become evenly dispersed along the perimeter. If a small step size had always been used, the algorithm could still have functioned, but it would have taken many more iterations to do so. It is shown in Michael's work that if the large step size is used for the entire formation operation, the robots will converge to a point and not a circle. However, using the larger step size initially helps to make the robots converge into a circle faster. The number of robots in the simulation needs to be reasonable for the algorithm to work. The algorithm requires that a robot see at least two other robots to be able to compute a new position. If the simulation is run with only a few robots and a small scanning range, the robots would not be able to detect two partners from the very beginning, thus the robots will never be able to converge to a circle since they wouldn't be able to compute any new positions.

2.2.2. Functionality of Michael's Algorithm

Each robot starts out with a random location and a random orientation. In simulation, these values are assigned by independent random number generators. A robot will scan left and right looking for the first robot it sees on each side. It will then determine from the distance to each robot and the angle between the two robots where it should move so that it can form the desired angle between the two robots. The movements of the robots are limited at each iteration so that a robot will only move a fraction of the distance toward its desired location if it is farther away than the allowed movement distance. This is shown in Fig. 1. It was shown in Michael's research that if the robots were allowed to move directly to its desired position all the time, the group of robots would converge to a single point rather than form the desired circle shape. It should be noted that the exact size of the circle and the exact center of the circle are arbitrary and depend on the given initial deployment of the robots. The size of the circle, if converged, will roughly have a diameter close to the length of the region where the robots are deployed, and the center of the circle is usually in the middle of the deployment region.

3. COLLISION AVOIDANCE SCENARIOS

A total of three collision avoidance scenarios were modeled. Each scenario used progressively more advanced robots. The first scenario involved the simplest robots with sensors that can only take readings while the robots are stationary. The second scenario assumes that the robots have sensors capable of detecting an imminent collision. The final scenario assumes that the same type of sensors that are used in the second scenario are also used in this scenario, but in addition to that, the robots are also capable of working through an imminent collision.

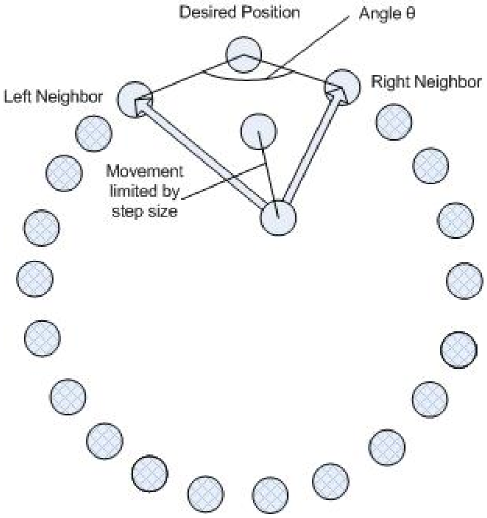


Figure 1. Illustration of the base algorithm, where a robot tries to move towards a desired location where it will form an angle θ with its neighbors.

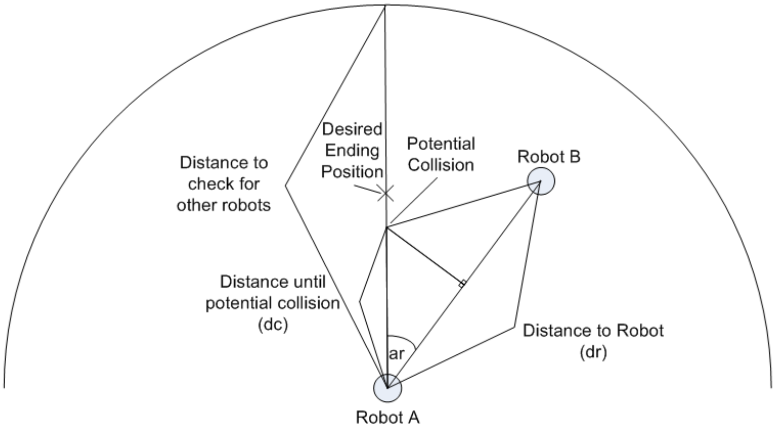


Figure 2. An example of collision addressed in Scenario I.

3.1. Scenario I

In scenario I, the robots are the most conservative and have their step-by-step movement limited the most among the three scenarios. The steps of the original algorithm that are described previously are still followed. In addition to these steps, the robots will each scan $\pm\frac{\pi}{2}$radians from its direction of movement and estimate which robot it might potentially collide with first, based on the assumption that the other robot will move toward its path and collide when the two paths intersect. This is illustrated in Fig. 2. Note that the other robot (Robot B) may or may not move in that exact direction and collide with Robot A, but that information is not available to Robot A and therefore Robot A needs to assume the worst case scenario. If no collision is possible for a given robot, then that robot will move normally like it would have in the original algorithm.

The main downfall of this scenario is that a lot of opportunity for movement is wasted. In most cases the robot would have been able to move further without colliding with any of the robots. The main advantage is that the sensors and other hardware needed for this robot are simpler than what is needed for the other scenarios, and that these robots should be the least expensive and the least complex to build. So, this scenario uses the simplest robots and should be the easiest to implement, from a hardware perspective.

3.2. Scenario II

Scenario II was more efficient than Scenario I, since in this scenario, the robots only alter their final position when a collision is actually going to occur. This scenario, when implemented in a real-world environment, would require sensors that can detect an imminent collision. Despite the more advanced sensors, the overall algorithm complexity should be about the same when implemented in physical robots.

As with Scenario I, the steps of the original algorithm are followed. The robots start moving toward their desired destination. However, if a robot crosses path with another robot, then both robots would stop by detecting each other during the movement and before the collision would occur. This is illustrated in Fig. 3(a). The ending locations of both robots are modified in this scenario, and the exact locations depend on the size of the robots.

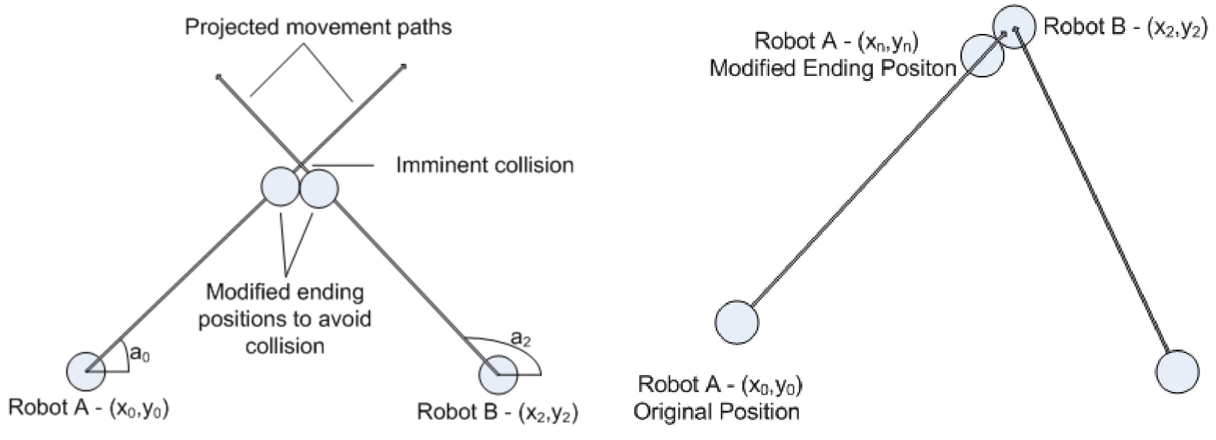
The destinations of the robots in question are only modified when a collision is actually going to occur. This is different from the first scenario where the robot will be limited whenever a collision is estimated to occur. There is still some wasted movement in this scenario. If the collision occurs early on in the robots movement, then a lot of potential movement is wasted for that iteration. This issue will be addressed in the next scenario.

3.3. Scenario III

Scenario III adds even more efficiency into the movement of the robots at each iteration. It improves on Scenario II in that the robots will be able to work through a potential collision as long as the collision does not involve the final stopping positions for the robots. When this happens, a robot will back off so that no more than one robot will occupy the same physical space.

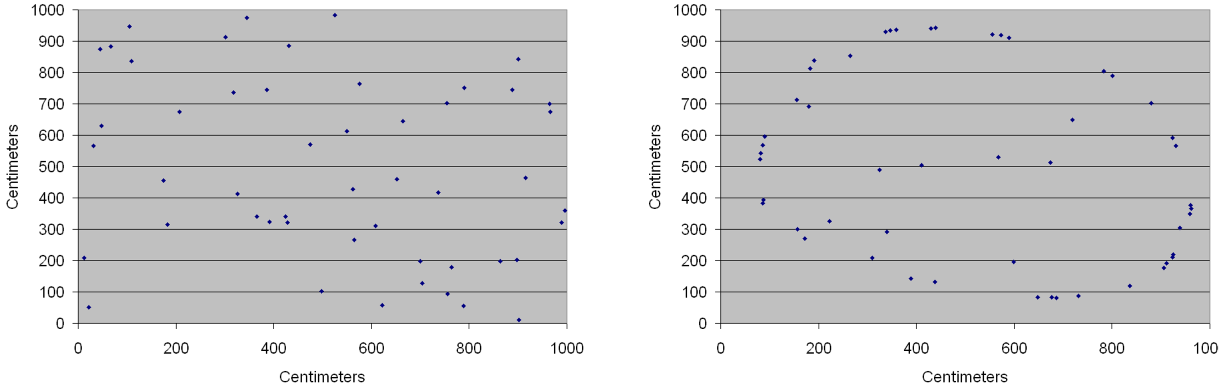
Wireless communication is a viable approach to achieve the type of collision avoidance discussed above. When a collision is about to occur, the robots would need to communicate and determine which robot would stand back and wait for the other robot to cross its path. Unless both robots are trying to occupy roughly the same desired ending position, as is shown in Fig. 3(b), then it will have no affect on the algorithm as to which robot were to back off and wait. The robots will randomly decide which robot may back off as they communicate.

The downfall of this is that the hardware and algorithm will be much more sophisticated than the previous two scenarios. The same sensors from Scenario II may be used again in this scenario since the robots would need to be able to determine when an imminent collision was about to occur. In addition, wireless communicators may be needed. If a node of a wireless sensor network were affixed to each robot, then the system of robots could function as a mobile wireless sensor network. There are already a large number of networking protocols that have been designed and developed for wireless sensor networks and they could be adopted for this environment. Wireless sensor networks can change over time due to events such as a node failure and have to be dynamic to

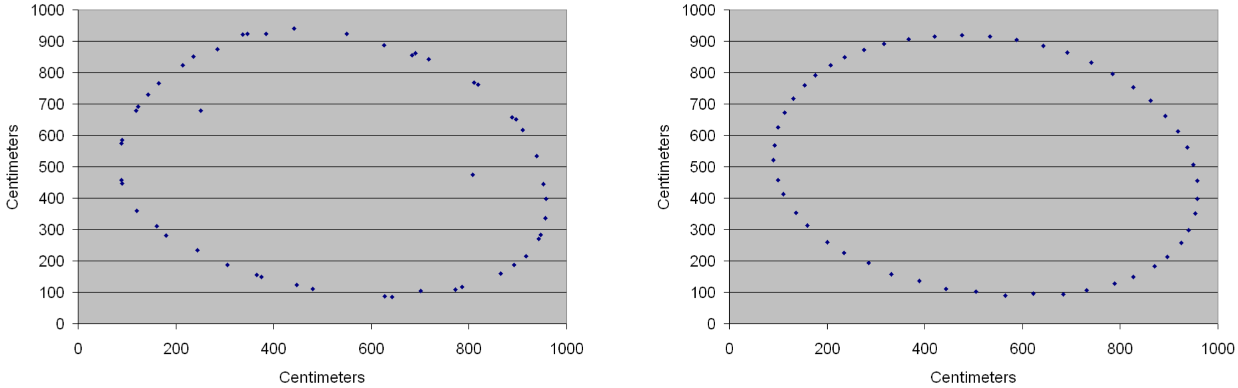


(a) Scenario II (b) Scenario III

Figure 3. Examples of collision addressed in Scenario II and Scenario III.



(a) Original Random Layout (b) Robot Positions after 5^{th} Iteration



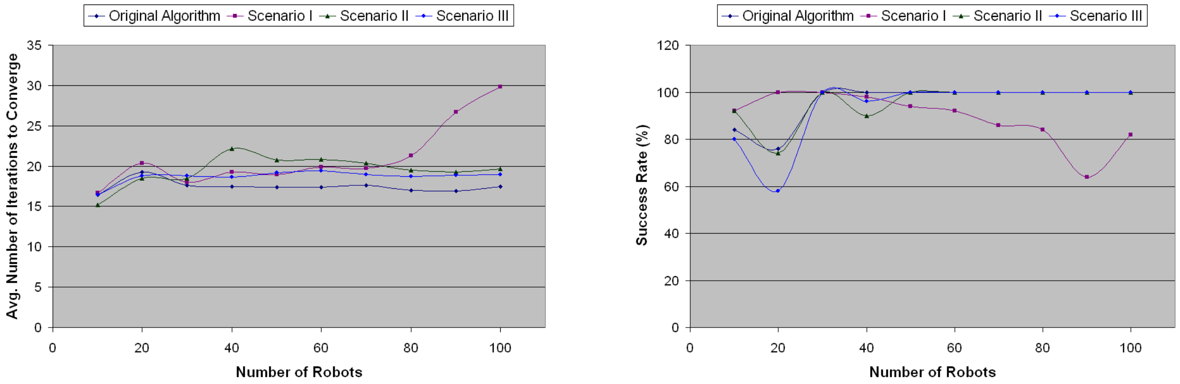
(c) Robot Positions after 10^{th} Iteration (d) Robot Positions after 23^{rd} Iteration

Figure 4. Example Results from Scenario III

account for such changes. Since the mobile network of robots in this system will be changing positions, these networking protocols would be able to develop new links from robot to robot. Another approach if wireless communication in not available is to have the robots wait for a short period of time and then back off at a random interval. Then the other robot that had chosen a longer random interval would now be able to see that there is no longer a collision and move forward. The robot that had backed off would then be able to move forward.

4. RESULTS

An example run of the algorithm using the collision avoidance scheme of Scenario III is shown in Fig. 4. The robots were randomly placed in a 10 meter by 10 meter grid, with the exception that no robots may occupy the same space. The scanning range for the robots was 500 cm, the initial step size was 100 cm, the final step size was 10 cm, and the radius of the robots was 5 cm. Fig. 4(a) shows the initial random deployment of 50 robots. Fig. 4(b) and Fig. 4(c) show a progression of how the group of robots forms a circle while running Scenario III. The final result of the algorithm after 23 iterations is shown in Fig. 4(d) and was run using Scenario III. It can be seen how the robots first move to the perimeter of the circle and then work to become evenly dispersed along that perimeter. The initial larger step size is used to help the robots to get to the perimeter, and then the smaller step size is used to help them become evenly dispersed. The algorithm stopped running after the 23^{rd} iteration since that is when the termination criteria was met. The criteria considers the robots to have converged to a circle when 90% or more of the robots were within ±3% of the angle θ. The termination criterion doesn't just



(a) Number of Iterations to Converge

(b) Success Rate

Figure 5. Original algorithm performance versus the performance of the collision avoidance scenarios.

make sure they are on the perimeter of the circle, but it will also check that the robots are reasonably spread out along the perimeter.

4.1. Comparison Against the Original Algorithm

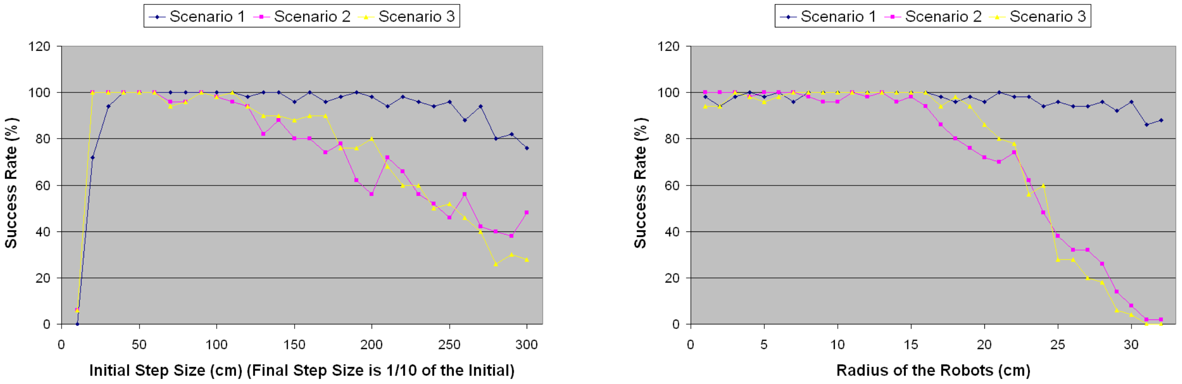
Out of the three collision avoidance scenarios, the performance of Scenario III most closely resembles the performance of Michael's original algorithm, as exhibited in Fig. 5. The original algorithm and the original algorithm using each of the three scenarios were run with the standard parameter values as described earlier. The algorithm exhibits about 100% success rate when there are sufficient robots (≥ 30), except for Scenario I, while using the standard parameter values. In terms of convergence speed, Scenario I also requires more iterations to complete the circle formation than the other scenarios do when there are more than 80 robots. The difference exhibited by Scenario I is due to its conservative calculation to avoid collision. The more robots there are, the more conservative the robots are in Scenario I and thus the harder it is for the system to form into a circle, if it does at all. Interestingly, robots with imminent collision detection (Scenarios II and III) perform very close to the original algorithm even though the original one assumes zero robot size and collision will ever happen.

Additional experiments have also been conducted to examine performance differences when varying different robot and algorithm parameters. We first provide a high level summary of how these parameters affect performance in comparing to the original algorithm. A more detailed discussion and results comparison among the three collision avoidance scenarios will be given in the next subsection.

Clearly there will be a larger discrepancy between the original algorithm and those with collision avoidance if the robots are larger. The smaller the radius of the robots is the closer the systems will perform.

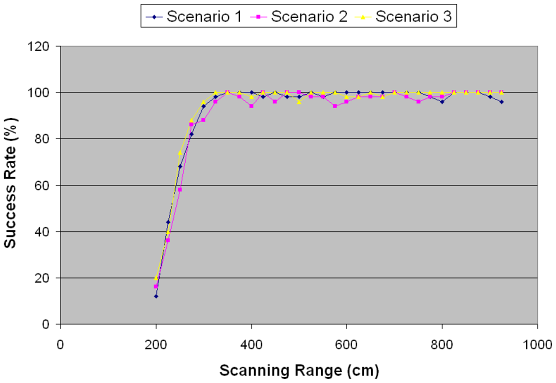
Modifying the step sizes will have a similar effect on both systems. If the step sizes are too large, then the robots will cluster into small groups. Once the robots are clustered together, then they keep scanning and detecting other robots in that group as neighbors and therefore never move out of that group. In the system running the collision avoidance of Scenario III, the robots would just be more spread out in the cluster, whereas in the original algorithm they will be closer together. The exact spacing in the final formation from the system running Scenario III would depend upon the radius of the robots.

Changing the scanning range would affect the robot team circle formation roughly the same way regardless of the existence of collision or the collision avoidance scenario used. If the scanning range is too small then the robots in the simulation are unable to detect at least two other robots, then they will not move and therefore there will be no collisions. Once the scanning range is large enough, then the robots are able to detect other robots and compute their new positions to move toward.



(a) Success rate while varying the step sizes

(b) Success rate while varying the radius



(c) Success rate while varying the scanning range

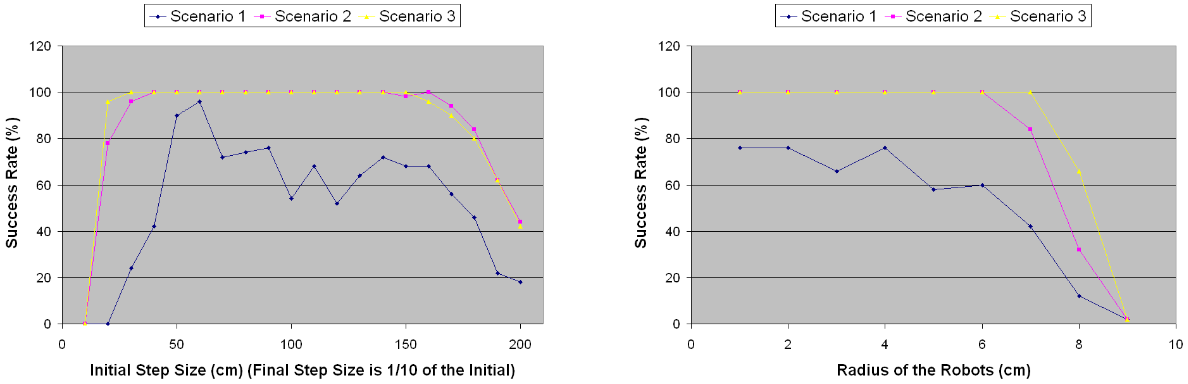
Figure 6. Success rates for the simulations with 30 robots while varying the step sizes, radius, and scanning range.

4.2. Effect of Robot and Algorithm Parameters

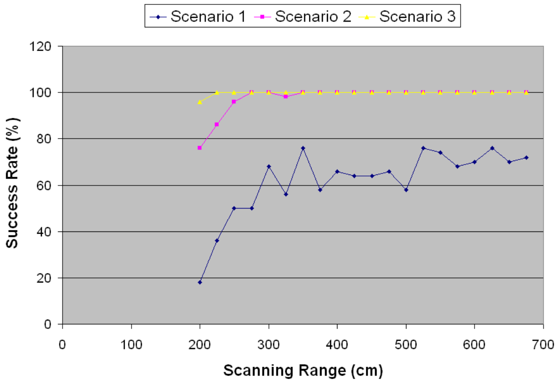
The comparison of the scenarios should involve not only performance comparisons, but also the complexity of the required hardware and software needed to implement the collision avoidance. Scenario I requires the simplest sensors yet it needs to calculate potential collision before every move. Scenario II requires more complex and accurate imminent detection sensors and locomotion control than Scenario I does. Scenario III requires additional hardware and software to enable communications between robots, such as short range wireless communication.

The minimum scanning range that worked for most simulations was around 200 cm to 220 cm. However, the systems with 30 robots and less required a larger scanning range to successfully form a circle, as illustrated in Fig. 6(c). For example, the simulations with 20 robots required a scanning range of about 400 cm to successfully work. The reason for this is simple: with fewer robots in the simulations, it became more difficult for a robot to detect two other robots so the robot could not compute a location to move toward and would stay in the same position. It is also possible that the robots can become divided into multiple smaller groups in the field. When this occurs, the robots in one group are not able to scan the robots in another group because they are out of the range of the sensors.

One trend that was evident throughout all of the simulations is that as the number of robots in the simulations decreased, the robots were able to increase in size and the algorithm was able to still converge. In the simulations with 100 robots, the success rate started to fall off when the radius reached a size of approximately 7 cm, shown in Fig. 7(b), while in the system with 30 robots such a drop off occurred at approximately 16 cm, shown in Fig. 6(b). For the simulations with 10 robots, this trend is even more evident when the drop off occurs when



(a) Success rate while varying the step sizes (b) Success rate while varying the radius



(c) Success rate while varying the scanning range

Figure 7. Success rates for the simulations with 100 robots while varying the step sizes, radius, and scanning range.

the radius is over 40 cm. Since there are fewer robots deployed in the same size field, it makes sense that larger robots are able to successfully form into a circle.

In the scenarios with a larger number of robots, Scenario I performed the worst. Scenario II and III were close, but Scenario III did on average converge faster. In addition, the success rates for Scenarios II and III had a span were it was 100%, whereas the success rate for Scenario I over all three sets of simulations for the 100 robot system only briefly exceeded 80%, as illustrated in Fig. 7(a). Despite the varying number of robots in the simulated system, when the success rate was high the system took around 20 iterations to converge using the criteria previously described. As the success rate fell, the average number of iterations to converge increased and fluctuated more.

Scenarios II and III continuously performed well over all of the systems within a reasonable range of the parameters. When looking at all of the simulations, the results for these two scenarios were about the same. However, Scenario III did perform slightly better overall in terms of the average number of iterations to converge. The success rates for Scenarios II and III were both very close throughout the simulations. These scenarios did not perform well if the radius or step sizes became too large or if the scanning range was too small. The exact values of the successful parameters varied depending on the number of robots in the system. Scenarios II and III performed better because they allowed the robots to move more efficiently. A robot's position was only modified when it needed to be, so the problem of deadlocked robots did not occur in these scenarios, as it had in Scenario I.

The success rate for Scenario I remained high as Scenarios II and III dropped off for larger step sizes and radii for systems with 30 robots or less. This can be seen in Figs. 6(a) and 6(b). As the radius and step sizes increase, the situation of deadlocking robots reverses itself as they occur in Scenarios II and III and not in Scenario I. The collision avoidance scheme in Scenario I counteracts the larger step size and radius and keeps the robots further away from each other. In Scenarios II and III the robots become deadlocked since the collision avoidance has the robots group together when collisions occur. Unlike the success rate for the step sizes and radii, the success rate while varying the scanning range had a similar effect across the three scenarios, which is shown in 6(c).

The success rates for the simulations with 100 robots are shown in Figs. 7(a), 7(b), and 7(c). These graphs show how Scenario I did not perform nearly as well as Scenarios II and III when there was a larger number of robots in the system, regardless of which parameters were varied. The graphs also show that for a larger number of robots the performance of Scenario II and III were about the same.

5. CONCLUSION

The goal of this work was to introduce robot size and collision avoidance into an existing autonomous robot team formation algorithm. A total of three different collision avoidance schemes were introduced each requiring a different level of capabilities from the robots.

Previous systems that deal with a large number of robots have been done in simulation with ideal assumptions. Some basic collision avoidance was considered in systems using global scanning, but none were found that deal with local scanning. This work advances the understanding of Michael's algorithm by incorporating robot size and collision avoidance schemes.

The results of the different collision avoidance schemes were compared, each having its own advantages and disadvantages. The choice of which collision avoidance scheme to use was found to be dependent upon the system it is to be used on. It is the hope that that this research has taken the algorithm one step closer to implementation in a real world system. In the system with under 40 robots, Scenario I looks like the best choice. This is because it requires the simplest hardware and should therefore be the least expensive to implement and the performance for all three scenarios were about the same. For the systems with 40 or more robots, the performance of Scenario II and Scenario III are about the same. If the robots in the system have hardware that makes it easy for the robots to work out a back off scheme, then Scenario III should be used. If cost is a driving factor in the system, then Scenario II should be used since Scenario II would probably be the simplest of the three scenarios to be realized from the software development perspective. Even though Scenario I implements the most conservative form of collision avoidance, the calculation to determine the earliest collisions might require a floating point processor. For Scenarios II and III, is it likely that a fixed point processor is all that would be required. However, Scenario III would also require some form of communication which adds some complexity and cost over that of Scenario II.

Much of the research done in cooperative robots is theoretical work. An important step in the research on cooperative robotics is to introduce more real life factors into the algorithms. Such research will help to identify problems so they do not occur when the algorithm is to be implemented in a real life system.

REFERENCES

1. J. Fredslund and M. J. Matarić, "A general algorithm for robot formations using local sensing and minimal communication," *IEEE Transactions on Robotics and Automation* **18**, pp. 837–846, Oct. 2002.
2. K. Sugihara and I. Suzuki, "Distributed motion coordination of multiple mobile robots," in *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pp. 138–143, 1990.
3. H. Ando, I. Suzuki, and M. Yamashita, "Formation and agreement problems for synchronous mobile robots with limited visibility," in *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 453–460, Aug. 1995.
4. Y. Oasa, I. Suzuki, and M. Yamashita, "A robust distributed convergence algorithm for autonomous mobile robots," in *1997 IEEE International Conference on Systems, Man and Cybernetics*, pp. 287–292, Oct. 1997.
5. I. Suzuki and M. Yamashita, "Distributed anonymous mobile robots: Formation of geometric patterns," *SIAM Journal of Computation* , pp. 1347–1363, 1999.

6. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, "Hard tasks for weak robots: The role of common knowledge in pattern formation by autonomous mobile robots," in *10th Int. Symp. on Algorithm and Computation (ISAAC '99)*, pp. 93–102, 1999.
7. P. Flocchini, G. Prencipe, N. Santoro, and P. Widmayer, "Distributed coordination of a set of autonomous mobile robots," in *Proceedings of the IEEE Symposium on Intelligent Vehicles*, pp. 480–485, 2000.
8. X. Défago and A. Konagaya, "Circle formation for oblivious anonymous mobile robots with no common sense of orientation," in *Proceedings of the second ACM international workshop on Principles of mobile computing*, pp. 97–104, 2002.
9. A. M. Michael, "Circle fromation algorithm for autonmous agents with local sensing," Master's thesis, Electrical Engineering Department, Rochester Institute of Technology, July 2004.