Multi-Robots Systems : Project Report

Nathalie Majcherczyk
Joseph Schornak
Nishan Srishankar
RBE 510 F16 - E. Eberbach
Worcester Polytechnic Institute

Fall 2016

Abstract

In this paper, we describe a flocking algorithm implemented on the ARGoS simulator and tested on a 'swarm' of Arcbotics Sparki robots. The algorithm works without global information and without communication between robots. This implementation uses the Lennard-Jones algorithm, as a means of weighting the attraction-repulsion forces between robots, and goal-aware robots to orient them as required. Once flocking was attained, we analyze the ability of the algorithm to maintain connectivity within the flock while moving towards a common goal without the loss of any members. The efficiency of this algorithm was analyzed, and possible improvements were suggested. These were done by testing various flocking scenarios: goal in motion, absence of a goal, tightly-positioned initial swarm, and flock in a non-cohesive initial position

1 Introduction and Background

In recent years, many control algorithms have been proposed for multi-agent systems. While these can be computationally more complex than the control of a single agent (robot), multi-agent control provides advantages in a variety of applications. For instance, a multi-agent system can be used to quickly explore a novel terrain while using less energy than a single-agent system. Multiple robots can be used for task allocation such as the transport of a heavy object in scenarios where a single robot could fail.

1.1 Flocking in biological swarms

Flocking or self-organized flocking is when a group of organisms move cohesively in a common direction towards a common goal. This can be seen in many animal species such as a swarm of locusts, clump of ants, or a flock of birds[1].

All flocking behaviors in nature have three similarities.

1. Separation- Within short ranges, every organism repels each other to avoid overcrowding neighbors.
2. Alignment- All members are oriented towards an average heading of its neighbors.
3. Cohesion- All members of a flock are attracted towards the average position of its neighbors.

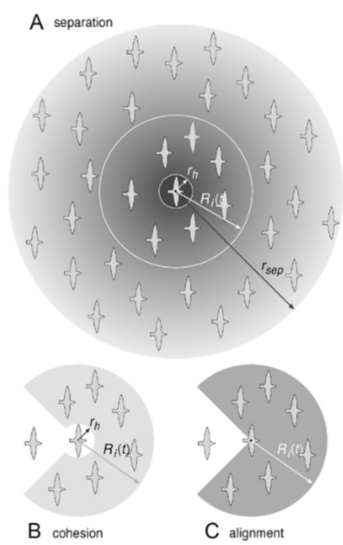


Figure 1: Interactions in a biological swarm

Flocking is an example of emergent behavior in a swarm[2], and manifests as a result of local interactions between organisms in the swarm. As can be seen in Figure 1, a flocking is maintained in a biological swarm through a combination of the three individual behaviors shown above.

1.2 Flocking in swarms of robots

In this project, we look at the possibilities for a system of multi-agent robots to achieve flocking. We define a multi-robot system as a set of autonomous, mobile robots that collaborate to achieve a mission. Specifically, this paper looks at flocking in a swarm of robots which consist of a large number of simple, individually-weak physical agents[3]. One of the overlying parameters that need to be accounted for in any swarm is connectivity. This is because, any single robot by itself has extremely limited capabilities with sensing, localization or processing power. It is only through a collective effort of all of the members in the swarm can a cohesive unit be formed that has decent sensing abilities and processing power. Connectivity, therefore is of vital importance, and a loss of even a single agent can greatly depreciate the effectiveness of the swarm.

A prominent paper in flocking-behavior of a mobile robot swarm by researchers at IRIDIA, Université Libre de Bruxelles breaks down the behavior of every robot in the swarm into three categories, attraction, repulsion, and alignment. This is in keeping with research done into flocking in biological swarms.

The attraction-repulsion behaviors ensures a balance in which robots are kept in proximity with its neighbors, while avoiding being extremely close to each other and potentially colliding with each. Finally, alignment ensures that all individuals match its heading to that of its neighbors, and hence point in the same direction.

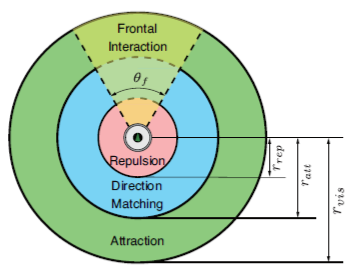


Figure 2: An individual robot's sensing area

Most papers and scenarios implement this scheme using Range-And-Bearing sensors.There are also scenarios in which flocking is achieved without alignment control. In this case, most robots are aware of the direction of a goal or a light-source, broadcast this to the other robots in the swarm, and cohesively move towards its objective[4].

Flocking behavior can also be simplified to collision avoidance, and velocity-matching flock-centering as seen in "Self-organized flocking with agent failure" by Hayes, and Dormiani-Tabatabaei[5]. The robots used in the paper can sense range and bearing of neighbors within a range. This information is then used to compute the Center-of-Mass of the swarm based on the relative neighbor position and heading towards a goal. This in turn helps to form a cohesive flock, with change in the center-of-mass at every time interval ensuring the robots are always aligned with the group.

This paper was of interest to our case, not only because the algorithm was analyzed using around the same number of robots, but because a pseudo-sensor was used. This ”pseudo-sensor” tracks the robots using an overhead camera and broadcasts the readings to robots.

Flocking was chosen as a project because it is a culmination of the content that was learnt in the lab. The lab assignments dealt with cooperation and collaboration when the Sparkis had to collaborate to move an awkwardly placed object, or perform task-allocation when sorting objects (as well as having some semblance of obstacle-detection and avoidance). Flocking takes these behaviors as well as adding on aggregation and following/safe-wandering behaviors.

2 Methodology

The following section deals with the high-level planner of the flocking-algorithm obtained from the list of requirements needed.

2.1 Requirements

For this task, there are some metrics that are assumed. All agents that is to comprise the swarm are homogeneous. That is to say that all robots have the same hardware, and are provided with the same controller. As a result of this, there will be no leaders or followers.

All members of the swarm do not communicate explicitly with each other. Any communication follows virtual stigmergy which involves robots sensing modifications in the environment than direct transmission of messages. In this case, the robots will sense the states of its neighbors.

The controller is completely decentralized. There is no central hub that dictates how members in the swarm would cooperate with each other. A corollary of this is that the robots will only have access to local information that it perceives, and hence make decisions based on this data.

2.2 Flocking Algorithm

The interaction between two neighboring agents in a swarm is modeled using the Lennard-Jones potential equation seen below.

$$V(r) = 4 * \epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^{6} \right] \tag{1}$$

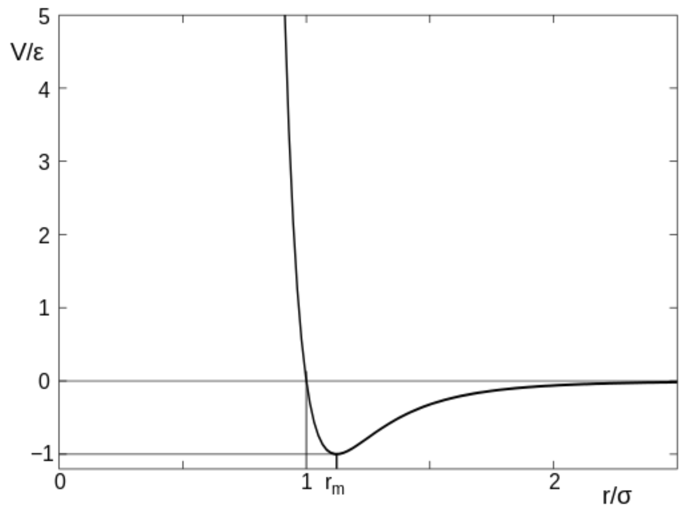


Figure 3: Lennard-Jones potential: Attractive and repulsive forces

In Equation 1, r is the distance between two atoms, while ϵ and σ are constants found experimentally.

As can be seen molecules feel both attractive and repulsive forces. At distances greater than σ, the potential energy is negative, which means the atoms feel an attractive force, whereas at distances below σ, the potential energy increases exponentially due to Pauli's repulsion.

Intuitively, this means that neighbors that are closer than a specified threshold distance will feel a repulsive force and move away from each other, while neighbors that are further away from the threshold (and less than the sensing radii) will feel an increasingly attractive force until the threshold distance is bypassed.

For implementing simplicity, the Lennard-Jones potential equation is simplified to Equations 2,3.

$$d_{norm} = \left(\frac{d_{target}}{d_{dist}} \right)^2 \tag{2}$$

$$V(\epsilon) = \frac{-G}{d_{target}} \left[{d_{dist}}^2 - d_{dist} \right] \tag{3}$$

In this case, G is the gain and d_{target} is the desired distance between robots. These two variables are selected by trial and error according to needed specifications (such as the size of the robots to be utilized in the swarm). Additionally, d_{dist} is the actual distance between robots.

3 Implementation

3.1 ARGoS Simulation

Our flocking algorithm was implemented in the ARGoS simulator that was developed by Professor Pinciroli for swarm robotics. Screenshots in various stages of the swarming are shown below. The swarm starts off tightly-connected, and can detect the goal yellow LED (target position). Throughout the process, the footbots maintain connectivity with each other (from the algorithm), and hence form a flock until the target is reached.

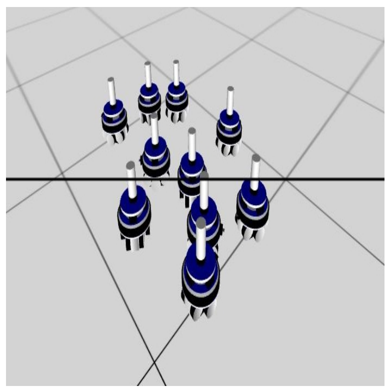


Figure 4: Initial position

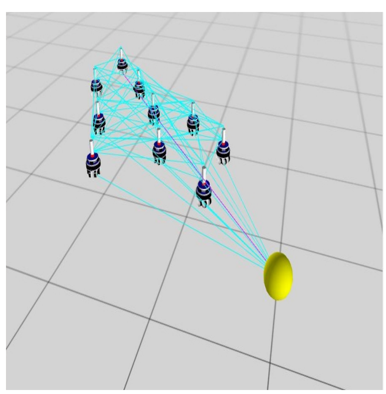


Figure 5: Maintaining connectivity using ranging sensors

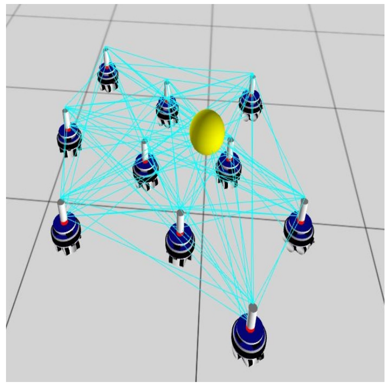


Figure 6: Closer view of how connectivity is maintained

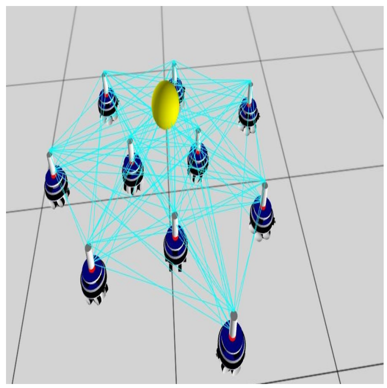


Figure 7: Reaching the goal (Yellow LED)

3.2 Sparki Robots

3.2.1 Differential Steering

The original Sparki Arduino code used a Proportional-Integral-Derivative (PID) controller, and accepted movement commands in terms of speed and sway. While this was useful for lab assignments, direct control of the speeds of the left and right wheels was necessary for the implementation of a flocking algorithm. We modified the Sparki code to accept drive commands in terms of left and right wheel speeds within the range [-1,1].

3.2.2 Server software

The same computer vision and field control server was used as in the previous experiments to perform perspective correction and object detection.

3.2.3 Pseudo-sensor

The original flocking implementation relies on each robot having its own sensors which provide distance and heading to neighboring robots within a range in terms of the robot's local coordinate system. In contrast, the field control server outputs global positions for every robot on the field.

We simulated limited-range individual sensors by creating a pseudo-sensor function. During the control-update step, each robot requests a sensor update through the pseudo-sensor. The pseudo-sensor uses the robot's ID to find its position and orientation in the global field data. It measures the distance from the center of the robot to the center of each neighboring robot within 50 cm, and calculates the heading of each nearby neighbor relative to the robot's forward direction. The pseudo-sensor returns a vector of paired distances and angles for each nearby neighbor.

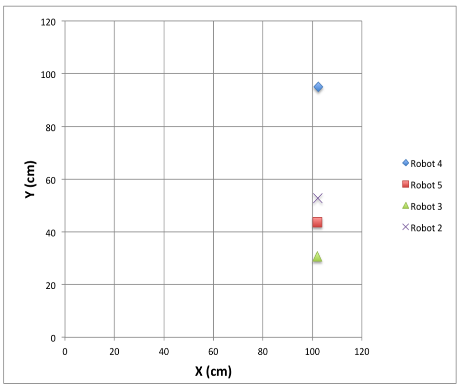
3.2.4 Parameter Tuning

To account for any noise in the system, such as those coming from any inefficiency in the differential-steering controller, or transformation errors from the overhead vision system, as well as to tailor the Lennard-Jones potential for the Sparkis, different variables were modified through trial and error.This involved changing the target distance between robots to be 30 cm. The gain was chosen to be 1000 units, as it scales the attraction-repulsion forces. Finally, the base speed was set to 0.5 units, such that the magnitude of the flocking vectors on both the robots will be smaller than the maximum velocity (normalization).

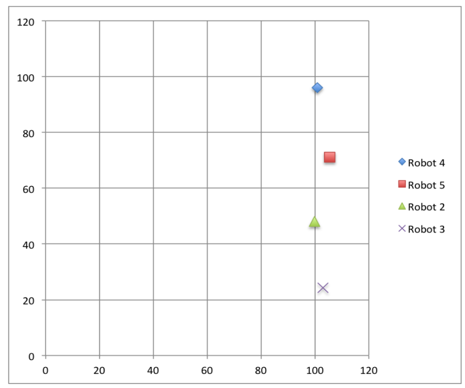
4 Results

We performed and recorded several demonstrations to show that the Sparki swarm was correctly acting under the flocking algorithm.

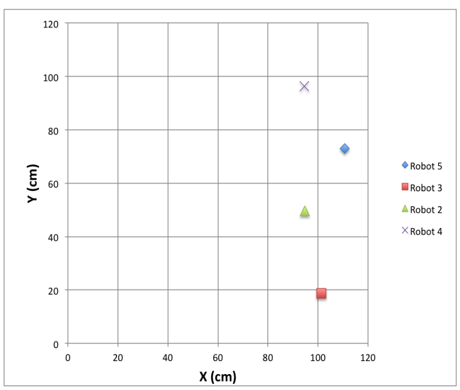
4.1 Line Initial Flock Without Goal



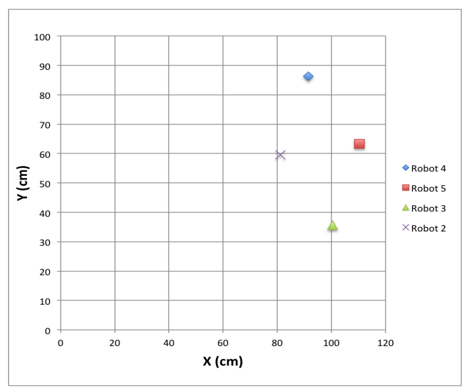
(a) Flock at time step 0



(b) Flock at time step 300



(c) Flock at time step 1200

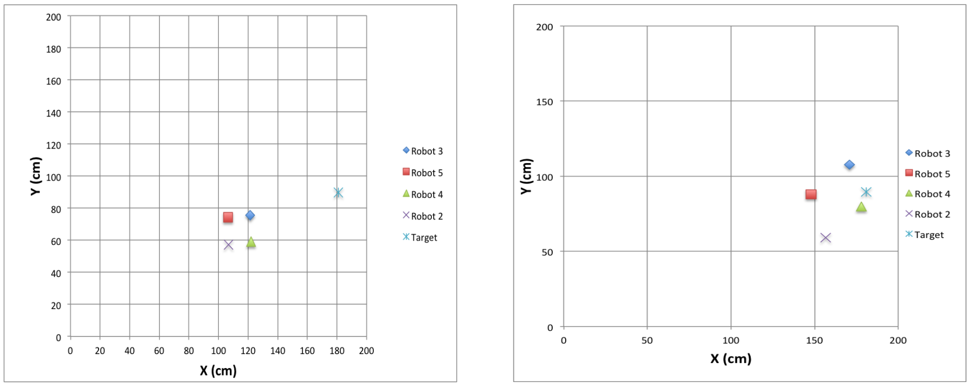


(d) Flock at time step 3000

Figure 8: Flock at different time steps

In Figure 8 the flock starts off parallel to the y-axis, slightly-spaced from each other. While they are close to each other, this is not a flock. As can be seen in the rest of the figures, the robots reposition themselves and circle around to form a diamond-shaped flock.

4.2 Tight Initial Flock With Distant Goal

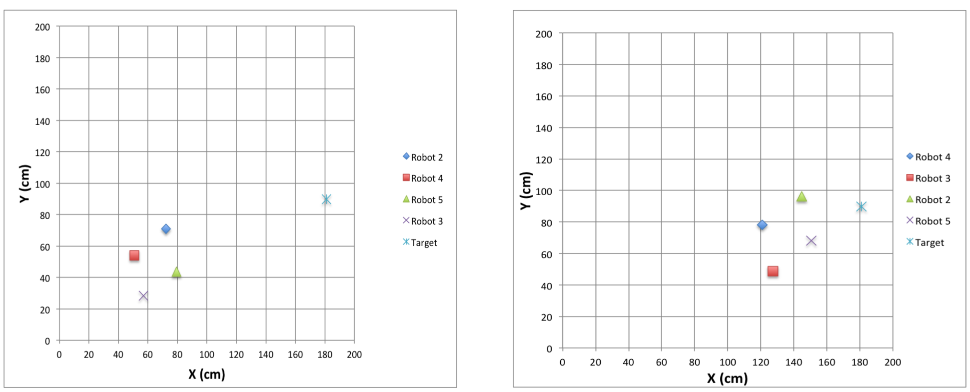


(a) Flock at time step 0 (b) Flock at time step 300

Figure 9: Flock at different time steps

In Figure 9, the robots start off almost touching each other. When the algorithm is run, the Lennard-Jones potential is extremely high, hence there is a repulsive force between each robot (to take collision avoidance into effect). Following this, the swarm flock to a distant goal, and end up in a similar pattern to the previous example.

4.3 Loose Randomly-Oriented Initial Flock With Distant Goal



(a) Flock at time step 0 (b) Flock at time step 25

Figure 10: Flock at different time steps

The experiment run in Figure 10 is similar to that in Figure 9, except the robots are initially spread-out and have random orientations. The end result is the same.

5 Further Improvements

Possible future improvement could include the addition of more Sparki robots using a larger field setup to better understand and study individual interactions between the robots.

We could also test methodologies for modifying an existing swarm, such as splitting swarms if the size gets too large, or see the interactions between swarms and checked against Buzz, a programming language which allows for swarm behaviors to be tested using primitives. Another possible direction would be to use a heterogeneous swarm such as land-based Sparkis and an aerial drone..

6 Conclusion

The implementation of the flocking algorithm on the Sparki robots matches what is obtained via the ARGoS simulator. The Sparkis flock correctly under a variety of conditions and starting configurations. All members of the flock start with random orientations and positions. Once the flocking algorithm is run, they maintain a common heading, in order to attain a necessary goal. When a goal is not specified, the robots form a 'circular' pattern, with each robot's heading oriented towards the centroid of the swarm.

For the future, the effectiveness of the flocking algorithm can be further tested by increasing the number of Sparkis used and seeing if connectivity, obstacle avoidance, and clustering is still maintained.

References

[1] Craig W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

[2] Maja J. Mataric. Designing emergent behaviors: From local interactions to collective intelligence. *From animals to animats 2*, 2:432–441, 1993.

[3] Manuele Brambilla, Eliseo Ferrante, Mauro Birattari, and Marco Dorigo. Swarm robotics: A review from the swarm engineering perspective. *Swarm Intelligence*, 7(1):1–41, 2013.

[4] Toshiyuki Yasusa, Akitoshi Adachi, and Kazuhiro Ohkura. Self-organized flocking of a mobile robot swarm by topological distance-based interactions. *IEEE/SICE International Symposium on System Integration*, pages 106–111, 2014.

[5] A. Hayes and P. Dormiani-Tabatabaei. Self-organized flocking with agent failure: Off-line optimization and demonstration with real robots. *IEEE International Conference on Robotics and Automation*, pages 3900–3905, 2002.