Flocking and Obstacle Response in Boids

Hannah Phillips, Jane Ivanova, and Tony Wang,

***Abstract*—Boid formation and flocking serve as valuable simulations to understand consensus driven behavior in nature. In this paper, we study the cohesion of boid flocking when faced with two types of obstacles: inactive and cyclically moving members. Additionally, we vary the the detection distance, cohesion, and velocity threshold sensitivity of each boid to investigate their effects on the overall flock stability. Finally, the code was implemented on the Robotarium for a live demonstration. Overall, we found that uniform flocking could be achieved among the robots despite the limitations of the field boundaries. Additionally, a higher detection distance correlated to a boid system more robust to disturbances and obstacles.**

***Index Terms*—multi-agent systems, networked control, flocking, boids, formation, separation.**

I. Introduction

FIRST described by Reynolds in 1987 [1], the formation of a flock is caused by the individual actions of each agent, resulting in macro-scale behavioral alignment and consensus. For such behavior to emerge, the robots must follow the "three flocking rules" [1], [2], which are described as "obstacle avoidance", "flock cohesion", and "velocity matching." Additionally, a flock is dynamic; for example, birds can group together in flocks or split off into sub-flocks and therefore can be considered as fission-fusion societies [3].

In a state transition into a flock, these birds will form connections or edges based on their vision, which may be modeled as wedge or disk detection [4]. However, if one bird becomes infected and starts performing irregular motions that are detrimental to the overall destination of the flock, its neighbors should temporarily separate themselves from it, which will result in fission. Once this problem is overcome, the separated birds should be able to regroup with the main flock and continue the journey towards their destination, leaving the infected member behind.

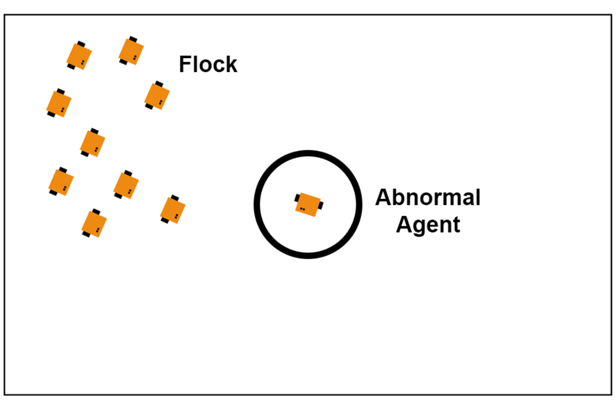

Fig. 1. Our paper seeks to address the response of a flock of boids to an abnormally behaving agent. For the scope of this paper, the abnormal agent could either be stationary (dead) or cycling in a circular manner.

Therefore, our approach to this phenomena will be to simulate it via introducing dysfunctional members (Figure 1) in an existing flock to change its behavior. After flock formation, one of the members will be randomly chosen to be infected. Afterwards, the boid will show two types of actions: becoming stationary/dead or demonstrating abnormal motion, which, for this demonstration, will be cyclic motion. The rest of the flock will then respond to this by either splitting into subflocks, effectively demonstrating fission, or moving around the infected member while keeping its uniformity. The local vision depth as well as a velocity sensitivity threshold, for sensing possible abnormal velocities, of the boids will also be tested to investigate how local behavior can lead to flock stability or instability.

II. Methods

A. Flocking

The three aforementioned "flocking rules" described by Reynolds have been implemented in various ways. Examples include using distance dependent adjacency matrix weights and penalty functions to enforce separation among the boids as well as averaging each boid's velocity with its neighbors for velocity matching. To implement flocking, we use Shalimoon's [5] approach and first define a Δ-disk proximity graph [6], where each boid b_i has a position p_i vector, velocity v_i vector, and a detection radius δ. Therefore, a neighbor boid b_j will belong to b_j's local neighborhood L_i of size k if its Euclidean distance away from b_i is less than δ. This may be mathematically written as the following:

$$L_i = \{b_j \in N, \forall b_j : |p_j - p_i| < \delta\}, \tag{1}$$

where N is the set that all the boids belong to. For this inter-agent sensing, the robots' detection radius was chosen to be 0.3m, as it was sufficiently short enough for each robot to only consider its own neighbors for flocking. If the detection distance was chosen to be the size of the field, 3.2m, boid formation would be trivial and not necessary, as the agents would stay connected from the beginning. For each time step, the positions of all the robots would first be obtained, and the adjacency matrix describing the edge connections would be updated.

When a robot is "turned off" due to having an abnormal velocity, the adjacency matrix is overridden such that other robots do not detect the affected agent. The flock maintains its original dynamics between other members. Next, obstacle avoidance, flocking cohesion, and velocity matching must be implemented. For obstacle avoidance, it is necessary to keep the boids in the local neighborhood L_i a certain distance from

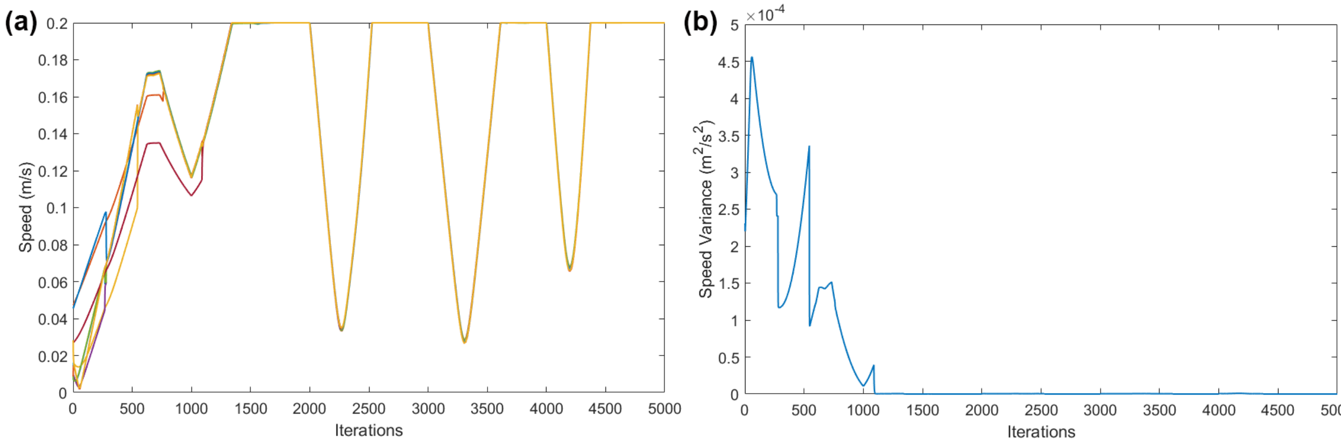

Fig. 2. (a) Speed vs. time-step iteration plot for 10 robots with each color representing a different robot. In the beginning, the robots' positions and velocities are randomized, resulting in asynchronization. The minima in the plots show when the robot slows down when it reaches a corner of the field. Due to pre-programmed reflection, the robot will then bounce off the boundary. To prevent the robots from exceeding actuator limits, a speed magnitude limit of 0.2m/s was placed on the bots, resulting in the flat lines in the plot. (b) The speed variance of all the robots vs. time-step iterations. Sudden increases in variance in the beginning are attributed to subflocks having different speeds. However, velocity matching is achieved at around 1100 time-steps.

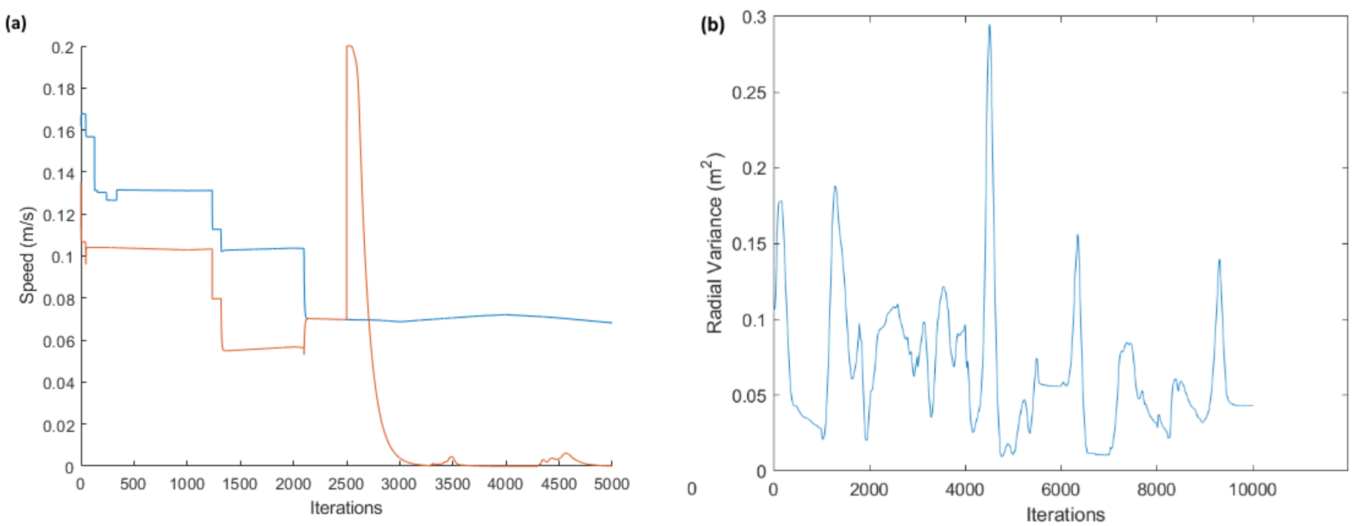

Fig. 3. (a) Velocity v.s. time-step iteration plot for the flock (blue) versus the abnormal agent (orange). Once a robot turns off and leaves the flock, the flock does not stop or try to adjust velocity to take in account the missing robot. (b) The inter-robot distance variance between the flock over time. The robots try to remain within their disk-detection range of 0.3m.

one another to prevent cohesion [5]. We can first mathematically define a separation vector $V_{sep,i}$ for L_i, which relies on the inter-agent distance vector $p_i - p_j$ as:

$$V_{sep,i} = -g_{sep} \sum_{b_j \in L_i} |p_i - p_j| \tag{2}$$

where g_{sep} is a separation gain, and the separation vector is a negative summation of the distance vectors to act as a repulsion force between agents [5]. To keep a flock, obstacle avoidance must be balanced with flocking cohesion, which allows each boid to stay close to the flock center. Each boid attempts to get as near to the centre of the flock as possible. The centre of the flock is based on the relative position of each flock member to its nearest neighbours in the visibility range of a boid. If the boid is surrounded with its neighbours, its flock centring urge is small. On the other hand, if most of its neighbours are afar, then it has a greater urge to get to the centre of the flock. Similarly to obstacle avoidance, it is dependent on the positions of the boids in L_i. We can then mathematically define a flocking cohesion vector $V_{coh,i}$ as:

$$V_{coh,i} = g_{col} c_i - p_i = g_{col} \sum_{b_j \in L_i} \frac{p_j}{k} - p_i \tag{3}$$

where c_i is the centroid of the boids in neighborhood L_i, g_{col} is a collision avoidance gain.

Finally, to move cohesively as a flock, each boid must align its speed and orientation with its neighbors. Here, we can simply take the average of the velocities in L_i and define the velocity matching term $V_{vm,i}$:

$$V_{vm,i} = \sum_{b_j \in L_i} \frac{v_j}{k} \tag{4}$$

To control for each force acting on b_i, we can additionally introduce different gains for separation G_{sep}, cohesion G_{coh}, and velocity matching G_{vm}. As a result, the velocity of each boid for each time step can be written as:

$$v_i = G_{sep}V_{sep,i} + G_{coh}V_{coh,i} + G_{vm}V_{vm,i} \tag{5}$$

Due to the Robotarium already implementing collision avoidance barrier functions for inter-agent distances 0.12m and under and the relatively small detection radius for each robot, a relatively low value for G_{sep} was chosen. Additionally, to prevent the robots from becoming stuck at a corner of the field, a boundary reflector was implemented and checked every 1000 time-steps. If a robot came too close, its velocity would simply be reversed. This in turn would influence its neighbors and change the overall trajectory of the flock. Finally, the maximum speed of the robots was set to 0.2m/s to prevent exceeding actuator limits.

B. Robotarium Environment And Parameters

The simulation was performed in two ways: a simulated environment in MATLAB and a real physical environment with centrally coordinated robots, both developed by the Robotarium [8]. The field of simulation in both cases was defined by the rectangular of size 3.2m x 2m. The robots are around 11cm in diameter. For the safety purposes of the real robots, the Robotarium environment supplies in-built rules for collision and border avoidance both in the simulation mode and in the physical environment using 11cm detection range.

To elaborate the restriction of the real-world environment in computer simulation and the flocking algorithm, the reflection rule was introduced and enforced robots to change their velocity upon approaching borders of the simulation area. The decision to reflect was defined by the following conditions:

$$\begin{gathered} x > b_{x_1} - r_x, \\ x < b_{x_2} + r_x, \\ y > b_{y_1} - r_y, \\ y < b_{y_2} + r_y, \\ b_x \in \{-1.6, -1.6\}m \\ b_y \in \{1, 1\}m \\ r_x = r_y = 0.2m \end{gathered} \tag{6}$$

where b_{x_1}, b_{x_2}, b_{y_1}, b_{y_2} - borders' coordinates, and r_x, r_y - offsets.

The initial velocities v_{i0} of the agents were generated randomly by the following function:

$$v_{i0} = (0.2. * randn(2, N)) \tag{7}$$

where N is the number of agents in the experiment.

The centering, g_{sep}, and collision avoiding, g_{col}, gains, were both set to 0.2. The gains for cohesion G_{coh}, separation G_{sep} and velocity matching G_{vm} were set to 0.001, 0.001 and 1 correspondingly for optimal performance.

The abnormal behaving agent was configured to be selected randomly using the following function:

$$rand\_rob = randsample(1 : N, 1) \tag{8}$$

where rand_rob is the index of the robot chosen randomly.

The reaction of other agents was defined by the velocity history tracking. For each boid, the value of the previous velocity was stored at each time-step iteration. Each iteration was approximately 0.033 seconds, which was determined by the Robotarium's time-stepping. Along with the adjacency matrix A, the algorithm was also conditioned by whether the neighbouring boid didn't change its velocity since the last time-step more than the threshold set for abnormality detection:

$$\sum_{v_j \in L_i} \frac{v_j}{k} - t_d < v_i < \sum_{v_j \in L_i} \frac{v_j}{k} + t_d \tag{9}$$

v_j is the velocity of the neighbour j in the neigbourhood L_i of i, where t_d is a abnormality detection threshold. During the experiment, the threshold values 0.01, 0.05, 0.5 were used.

III. Experimental Results

A. Velocity Matching

Analysis on velocity variance of individual robots with comparison to an abnormal agent was performed in Figures 2 and 3.

Early in the simulation, robots move around to begin a network of inter-connected boids determined by a Δ-disk proximity graph. Once they meet they synchronize. As the boids connections are updated by Δ-disk proximity graphs, the boids create smaller sub-flocks displaying velocity matching and centering as seen in Figure 2a. This is maintained until the network is completely connected and final velocity matching is maintained.

The velocities in the flock become synchronized at around time-step 1100. The general shifts in velocity may be attributed to the robots reflecting off the boundaries of the field.

B. Collision Avoidance

The flocks tended to reach a steady state (synchronization) throughout the experiments [1]. The inter-robot distance was maintained using separation penalty functions.

When met with an environmental obstacle, the flock navigated around said obstacle given by its dynamics, using force-field and steer-to-avoid mechanics.

If a robot within the formation was turned off, the boids separated themselves from the abnormal agent and attempted to maintains formation. The boids did not receive updates from the abnormal member. The abnormal member was treated as an obstacle and separate flock of size $N = 1$.

During time-step 2500, a robot was "switched off" as seen in Figure 1 3, and its velocity was changed to redirect it to the center of the field so that the other robots would have to encounter it more frequently.

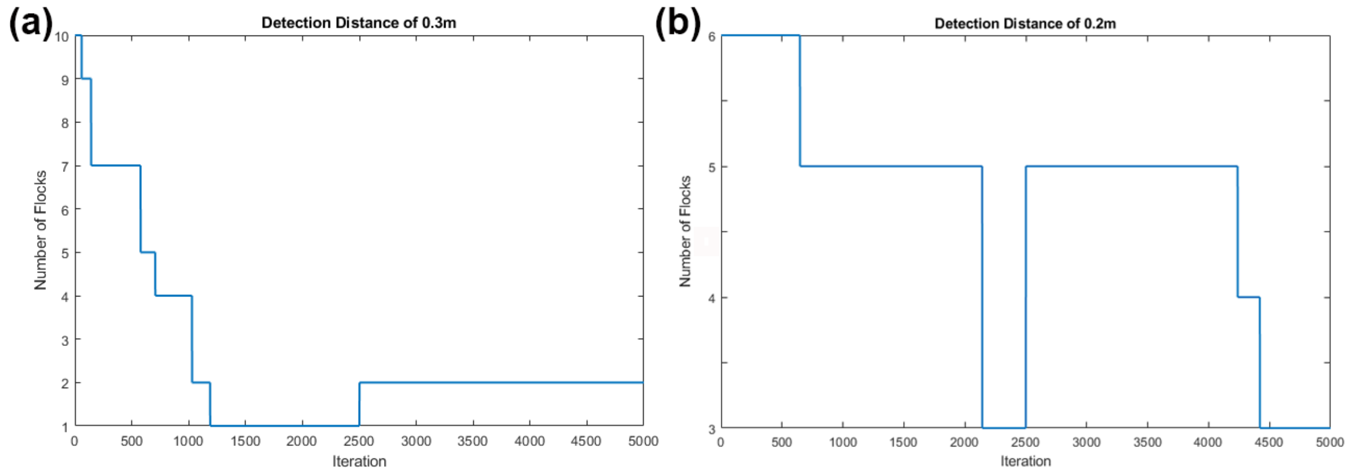

Fig. 4. Number of flocks vs. time-step iteration for (a) δ = 0.3m and (b) δ = 0.2m with a stationary abnormal agent. For both cases, the robot's initial positions and velocities are randomized, and at the 2500th time-step, a randomly selected boid is chosen to move to the origin and stay static for the rest of the time. This can be seen by an increase in the number of flocks at time-step iteration 2500. Furthermore, for δ = 0.2m, the flock became more scattered (increasing the number of flocks) after the dead agent was introduced. This is likely due to connections being more difficult to make for smaller detection radii.

Further investigation into detectability was performed for the circular abnormal agent. Using the values of previous velocity (one time-step back), the boids determined which of their neighbors was abnormal, which had deviation from the average velocity of the flock. This meant that the abnormal agent did not comply with the average velocity of the flock, and thereby had to be ignored. Although the Δ-proximity graph remained the same, the effect of a randomly selected abnormal agent did not break the flocking of other agents. Therefore, the abnormal agent left the flock while the boids continued to move along their original trajectory. The slight deviations from the original trajectory were observed when the connectivity graph of the flock was not full, meaning that some boids had less information about bigger portion of the flock and received the update from the abnormal boid due to the threshold.

Several experiments were set to determine optimal detection threshold, which would allow both avoidance of the abnormal behaviour and keeping the flock together. It was derived that the optimal value for the abnormality detection in the current setting threshold t_d is $0.05m/s$. With this, the boids were able to detect the abnormal member correctly most of the times and still keep the flocking. For the values of threshold $t_d < 0.01m/s$ the boids didn't align the velocities efficiently and the flocks were unstable - side boids were prone to fall out the flock. For the values of threshold $t_d > 0.5m/s$ the boids were following the usual consensus rule - were not excluding abnormal boid and treated it as a leader most of the times.

If the abnormal agent was "turned back on", it returned to the flock and resumed cohesive formation control.

C. Flock Centering

To quantify flock centering, analysis was performed on boid and abnormal agent velocities along with boid inter-agent distance.

In Figure 3, we can see that the boids try to remain close to each other to prevent de-synchronization of velocities. Analysis of the inter-robot distance variance in Figure 3b shows that boids within the flock remains within provided Δ-disk graph's radial bounds of 0.3 m^2.

Determined by boid dynamics, the flock acts as a network with a shifting centroid. As long as a boid is connected to any neighbor within the flock, the Δ-disk proximity graph between flocks updates and the centroid will update to preserve flock dynamics.

D. Abnormal Behaviour

1) Stationary Abnormal Agent

As mentioned beforehand, the abnormally behaving agent was selected randomly and displayed irregular behavior at the 2500th time-step. To have the most impact on flock trajectory, the agent was programmed to move to the center of the stage. Furthermore, the randomly selected agent was removed from the Δ-proximity graph once its velocity crossed the threshold and acted as its own free system. Due to the nature of an undirected graph, the boids were also able to ignore any updates coming from the abnormal agent. Therefore, this study investigated the effect that collision prevention, which was preprogrammed into the Robotarium robots, had on flock fission and fusion.

To quantify flocking fusion, we calculated the number of flocks in the system at each time step. This was done by iteratively by checking each edge in the adjacency matrix and updating the flock count. For example, if two nodes had no edges between them, the number of flocks would increase by 1. Afterwards, the value was recorded into an array and plotted against its corresponding time-step iteration as shown in Figure 4. A higher flock count signified less cohesion while a smaller flock count demonstrated stability and inter-agent communication.

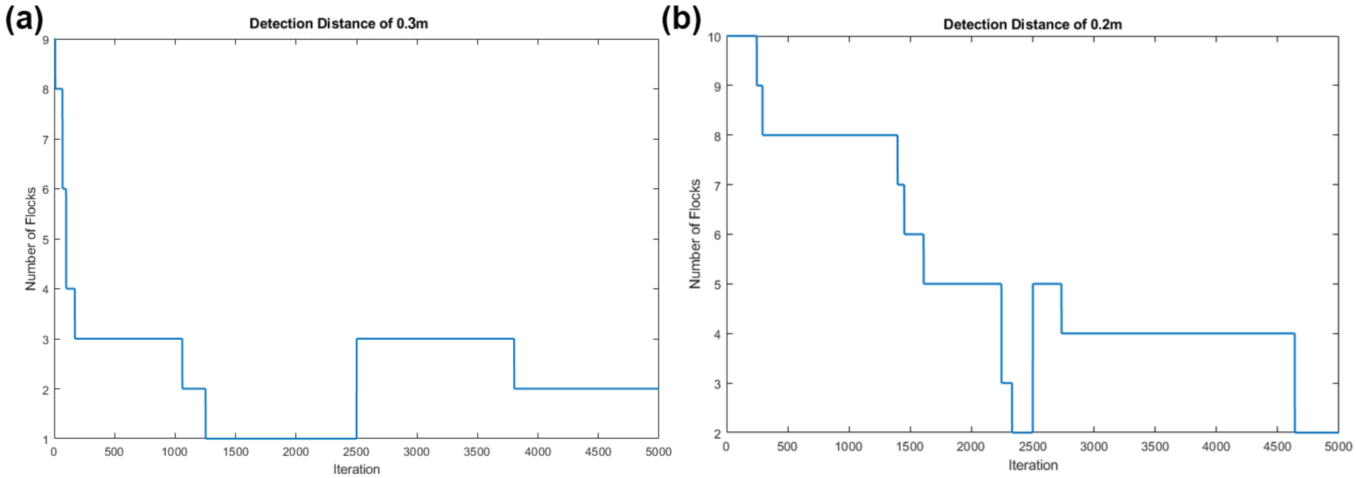

Fig. 5. Number of flocks vs. time-step iteration for (a) δ = 0.3m and (b) δ = 0.2m with a rotational cycling abnormal agent. For both cases, Similarly to the static case, at the 2500th time-step, a randomly selected boid is chosen to move to the origin and rotate instead of moving with the flock. The same relationship as the previous can be seen here; a smaller detection radius will lead to more divisive flocking.

Figure 4 shows the change in the number of flocks before and after introducing the abnormal agent. For the two plots, the detection distance was varied between 0.2m and 0.3m to study how local visibility and flock formation was affected. As shown in Figure 4a, for δ = 0.3m, despite being given randomized locations and velocities, the boids are able to form one unifying flock at around the 1200th time-step. After the abnormal agent was introduced at the 2500th time-step, the number of flocks only increased by 1, which was expected since the dead agent and the rest of the boids formed their own groups. Furthermore, even when moving past the stationary agent, the number of flocks remained the same, demonstrating that the detection distance was sufficient enough to withstand the obstacle.

On the other hand, as shown in Figure 4b, a slightly lower detection distance of 0.2m increased the number of flocks from one to five by the 1200th time-step. Due to their collision/safety radius being only slightly lower at 0.12m, the robots struggle significantly more to establish connections in their local neighborhood according to Equation 1. Despite briefly merging to three flocks, the introduction of a stationary abnormal agent was enough to increase the number of flocks by 2. This result demonstrated that a robust flocking system must have a sufficient yet cost-efficient detection radius so that obstacles will not scatter it.

2) Cycling Abnormal Agent

Although a stationary obstacle can influence flocking, a moving obstacle that actively pushes boid members can further test the limits of the system. For implementation, similarly to the previous case, the abnormal robot was moved to the center of the field and disconnected from its neighborhood. Afterwards, it followed a circular motion of 0.1Hz. During this time, the rest of the robots were actively moving together and bouncing off the field bounds, allowing for interagent deflections to occur. Additionally, the detection distances were chosen to be 0.3m and 0.2m again to study their effect on flock cohesion and robustness.

As shown in Figure 5, the boids showed similar trends as with the previous case; a smaller detection distance generally led to more instabilities and a higher subflock count. The active pushing of the abnormal agent also increased the number of flocks more than the stationary case; however, for both detection distances, the flock count was able to return to two. The limited space in the Robotarium field could have contributed to this; due to constantly bouncing off the boundaries, each robot had a higher chance of regrouping with others. Nevertheless, both of these simulation results show that the detection distance must be chosen carefully if the boids are expected to encounter nonidealities in their environment.

IV. Conclusion

In this paper, we have simulated Reynold's boid flocking to investigate fission-fusion societies. It was determined our simulations met the three requirements to be considered boid flocking: velocity matching, collision avoidance, and flock centering. Based on these goals, we have selected gains and boundary conditions to achieve our objective. An introduction of an abnormal agent was then used to analyze flock cohesion and stability, and it was found that a higher detection distance from each boid led to better connectivity in the flock. Additionally, moving obstacles that actively pushed members away from their neighborhood were found to be more disruptive to the formation. Future research will be conducted in multi-obstacle testing, multi-abnormal agent, and multi-flock interaction.

The analysis of the abnormality detectability experiments show that the exclusion of the abnormally behaving boid can be achieved by setting a threshold for velocity update. A formal mathematical analysis could be a good basis for further investigation. While we used the same constant threshold value for all the boids in these settings, it will be interesting to have further analyses of the effect of each boid having its own threshold value. As well as to see how the time-changing or other factors like the time the boid was assigned to the flock can affect the stability of flocking and abnormality exclusion.

APPENDIX A
CODE

The code extensions referenced and used in this article are attached to this project as followed:

Title
ECE6563FinalProjectNoObstacles.m
ECE6563FinalProjectCenterAgent.m
ECE6563FinalProjectCyclicSubFlock.m
ECE6563FinalProjectCircularDetection.m

APPENDIX B
VIDEOS

The multimedia extensions referenced and used in this article are attached to this project as followed:

Title
ECE6563FinalProjectNoObstacles1.avi
ECE6563FinalProjectNoObstacles2.avi
ECE6563FinalProjectNoObstacles3.avi
ECE6563FinalProjectAbnormalAgent1.avi
ECE6563FinalProjectAbnormalAgent2.avi
ECE6563FinalProjectAbnormalAgent3.avi
ECE6563FinalProjectCircularAgent1.avi
ECE6563FinalProjectCircularAgent2.avi
ECE6563FinalProjectCircularAgent3.avi
ECE6563FinalProjectAgentRejoins1.avi
ECE6563FinalProjectAgentRejoinst2.avi
ECE6563FinalProjectOptimalDetection.avi
ECE6563FinalProjectLowThresholdDetection.avi
ECE6563FinalProjectHighThresholdDetection.avi

ACKNOWLEDGMENT

The authors would like to thank Dr. Chaouki Abdallah for teaching the course and guidance on final project. Sean Wilson for assistance with Robotarium and Robotarium MATLAB code. Finally, Babak Taheri for assistance with understanding the material and teaching support.

REFERENCES

[1] C. W. Reynolds, “Flocks, herds and schools: A distributed behavioral model,” ACM SIGGRAPH Computer. Graphics., vol. 21, no. 4, pp. 25–34, 1987.
[2] R. O. Saber, “A Unified Analytical Look at Reynolds Flocking Rules,” 2003.
[3] I. D. Couzin, “Behavioral ecology: Social organization in fission–fusion societies,” Current Biology, vol. 16, no. 5, 2006.
[4] K. Laventall and J. Cortés, “Coverage control by multi-robot networks with limited-range anisotropic sensory,” International Journal of Control, vol. 82, no. 6, pp. 1113–1121, 2009.
[5] A. Shalimoon, “Cooperative control of multi-agent systems: Consensus, flocking and Formation Control,” thesis, 2018.
[6] S. Martinez, J. Cortes, and F. Bullo, “Motion Coordination with Distributed Information,” IEEE Control Systems Magazine, vol. 27, no. 4, pp. 75-88, 2007.
[7] Robotarium, Georgia Institute of Technology, Atlanta, GA, 30332 USA https://www.robotarium.gatech.edu/
[8] Simulation video on Robotarium environment. https://robotzoo-video.ecs.gatech.edu/owncloud/index.php/s/vvOYhsFoqWpgay7