Formations for Improved Connectivity During Agent Search

Joseph Spall IV
School of Electrical and Computer Engineering
Georgia Institute of Technology
jspall3@gatech.edu

Juan Elizondo
School of Electrical and Computer Engineering
Georgia Institute of Technology
jelizondo7@gatech.edu

***Abstract*—Inspired by the kid's game of "blob tag", four networking controls algorithms were developed for finite sensing situations to achieve connectivity of agents. Applications of such algorithms involve low information situations such as search-and-rescue missions. These algorithms were then simulated for various numbers of agents and compared in convergence time and success rate to strong connectivity.**

I. Introduction

Our work was inspired by the kid's game of "blob tag". The game started with having one person or a few people as the start of the "blob". Their goal was to then try to tag others. Once they successfully tag someone, the individuals link arms and continue to tag others. As the game progresses, the blob grows larger and larger. The game ends when all individuals have been tagged and are part of the blob(s).

From this inspiration, we were interested in exploring different styles of achieving connectivity of randomly positioned networked agents. The models could then be applied to search-and-rescue missions in unmapped areas or military air sweeps. Since agents in these situations usually have constrained sensing and communication capabilities, distributed agents can outperform individual ones due to increased coverage and parallelized processing.

II. Controller Algorithms

To focus the modeling, we applied additional constraints, such as unit disk graph [1] representation, matching sensors (or human vision) in the "real-world" having finite sensing distances. Agents that implement our controllers rely on a local reference frame, only knowing distances away from themselves and to other agents (when in the appropriate vision range). For higher level controllers, the information required should be discernible without global system or environment knowledge but can include aspects such as local connectivity.

To accomplish this connectivity task, we implemented several types of algorithms, all of which typically revolved around edge energy functions in some way. However, most had a unique algorithmic modification to the energy function in order to produce the varying behaviours desired. Additionally, some of the algorithms required a separate data structure to track order in the formation because the graphs were not static, but still under the assumption that the information did not rely on global knowledge.

We implemented a leader-follower architecture, with all of the followers using the same algorithm simultaneously. The control strategy of the leaders was simply to travel to random points, irrespective of the location of other agents. The follower's controllers, based on the energy functions, cannot impact the leader's controller besides physical constraints, as further elaborated in section III-A.

A. Min-Max Distance

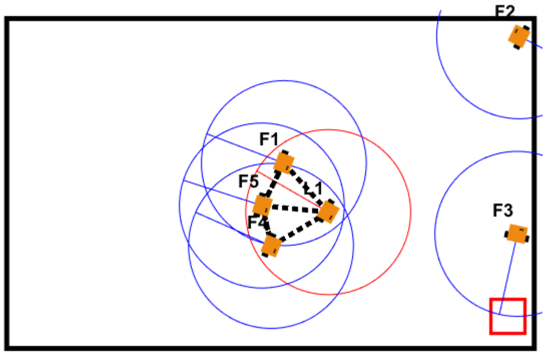

Fig. 1. Simulated run of Min-Max Distance Algorithm with one leader.

The first algorithm, Min-Max Distance, tries to keep agents between a minimum distance of δ and a maximum distance of Δ by combining two energy functions. The formation control energy function from [2] pushes away agents closer than δ while attracting agents that are farther away. With the finite communication distance, maintaining the existing connections within the distance of Δ can be achieved with energy functions from [3]. A combination of these energy functions, as seen in [4], results in the following form:

$$\mathcal{E}_{ij}(x) = \frac{1}{2(\Delta - \delta)} \left(\frac{||x_i - x_j|| - \delta}{\Delta - ||x_i - x_j||} \right)^2 \tag{1}$$

with the use of the gradient descent rule leading to the weight in the weighted consensus protocol to be:

$$\dot{x}_i = -\frac{\partial \mathcal{E}}{\partial x_i} = \sum_{j \in \mathcal{N}_i} w_{ij}(x_i - x_j) \tag{2}$$

$$w_{ij}(x) = \frac{1 - \dfrac{\delta}{||x_i - x_j||}}{(\Delta - ||x_i - x_j||)^3} \tag{3}$$

for the agent $j \in \mathcal{N}_i$ neighbors of agent i being in the vision distance. A note is that the controller does not care if j is a leader or a follower, which leads to possible issues as seen in section III-C. If no other agents were in the vision of i ($\mathcal{N}_i = \emptyset$), then the linear velocity of i was set to $\vec{0}$.

Algorithm 1 Min-Max Distance

```
$\mathcal{F}$ set of followers
**for** $i \in \mathcal{F}$ **do**
    $j \in \mathcal{N}_i$ set of neighbors of $i$
    $w_{ij} \leftarrow \dfrac{1 - \dfrac{\delta}{||x_i - x_j||}}{(\Delta - ||x_i - x_j||)^3}$
    $\dot{x}_i \leftarrow \sum_{j \in \mathcal{N}_i} w_{ij}(x_i - x_j)$
    **if** $\mathcal{N}_i = \emptyset$ **then**
        $\dot{x}_i \leftarrow \vec{0}$
    **end if**
**end for**
```

B. *Single-File Line*

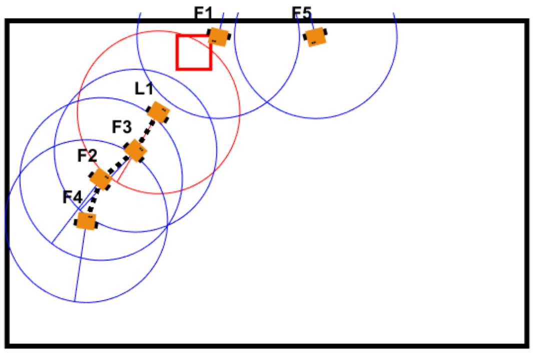

Fig. 2. Simulated run of Single-File Line Algorithm with one leader.

The next algorithm, Single-File Line, was designed to have the agents follow directly behind the leader, which requires a much higher level approach to organization and utilizes a different formation energy function.

We assume that each leader can sense what they are connected, being able to keep track of an ordered list. Additionally, it can be conveyed to an agent in this list that it is the last element in said list.

Each leader $i \in \mathcal{P}$ has a "line" S_i of length N associated with it, which is a permutation of the set of all agents; meaning it is ordered (according to the position in the line) and cannot have repeating values. Each line is initialized with an associated leader.

If a follower agent j sees the agent associated with the end of this line $S_i(N)$ (be it the leader or any other currently following agent) and is not already a member of the line, it joins the line at $S_i(N+1)$. This agent is now labeled as the end of the line. As a note, a single agent can be part of multiple lines if the other conditions are present, which provides compelling behavior for multiple leaders as will be discussed in section III-C. Conversely, if any agent that is part of the line loses vision of the agent in front of it, it and any agents following lose connection to the line.

A distance-based energy function can have the following form:

$$\mathcal{E}_{ij}(x) = (d^2 - ||x_i - x_j||^2)^2 \tag{4}$$

where d is the desired distance between points.

For all members of the line, the distance-based formation controller running is the following:

$$\dot{x}_n = 4(||x_n - x_m||^2 - d^2) \cdot (x_n - x_m) \tag{5}$$

where n is the index in the line, m is the index of the agent in front of n in the line (i.e. $n-1$). This algorithm starts from the end of the line and progresses up to leader, exclusive. A note about this energy function is that it is not globally stable, regardless of connectivity. Instead, it is locally stable for the distances greater than d, which can be enforced with control barrier functions as discussed in section III-A.

If the follower agent was not part of a line, its linear velocity was set to $\vec{0}$. This allowed for the line to not be congested by agents trying to join at "improper" locations, waiting for the back to show up in the view. The behavior also allowed for, as before with the Min-Max Distance controller, velocity being set to $\vec{0}$ if no other agents were seen.

C. *Angled Line*

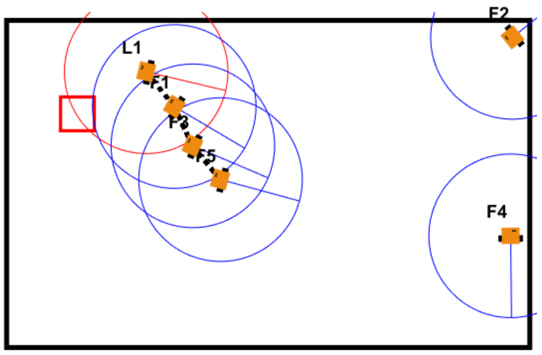

Fig. 3. Simulated run of Angled Line Algorithm with one leader.

With the purpose of increasing the coverage of the sweeping agents, we now move to angle the single-file line with respect to the leader. For this algorithm, we assume the agents have a sensor that allows them to "see" the pose of other agents, such as a camera with accompanying AprilTags [5], as well as a simple way of communicating a pair of Boolean variables, such as light indicators. Additionally, the agents are assigned

Algorithm 2 Single-File Line

```
$\mathcal{P}$ set of leaders
$\mathcal{F}$ set of followers
for $i \in \mathcal{R}$ do
    $S_i$ line for leader $i$
    $N$ is size of $S_i$
    for $j \in \mathcal{F}$ do                                ▷ Line addition
        if $j$ in the vision of $S_i(N)$ and not in $S_i$ then
            $S_i(N+1) \leftarrow j$
        end if
    end for
    for $a = 1 : N$ do                                         ▷ Line removal
        $b \leftarrow a + 1$
        if $b$ not in the vision of $a$ then
            Remove $b$ to $S_i(N)$ from $S_i$
        end if
    end for
    for $n = N : -1 : 2$ do                                     ▷ Controller
        $m \leftarrow n - 1$
        $\dot{x}_n \leftarrow 4(||x_n - x_m||^2 - d^2) \cdot (x_n - x_m)$
    end for
end for
for $f \in \mathcal{F}$ do                                     ▷ Controller Cont.
    if $f$ not in any $S_i$ then
        $\dot{x}_f \leftarrow \vec{0}$
    end if
end for
```

an integer number, which can be sensed as well. Leaders are marked with the smallest numbers to assert priority in algorithm 3.

Follower agent i can decide on a neighbor to follow, or target, based on whether some criteria are met.

- It should follow a leader, or an agent who is asserting being part of a following network.
- If multiple options are available, agent i should attempt to follow the agent with number closest to $i-1$, while preferring those less than i.
- A leader not marked as being followed should take highest priority.

However, if agent i has only one neighbor, all the above should be ignored.

For the controller, instead of following the target agent, agent i will follow a point in relation to the target agent, x_a. The angled line of followers will then be formed by specifying x_a as a set distance from the target and an angle in relation to the current pose of the target.

Additionally, the controller imposes a switched system. When agent i is within a maximum distance Δ from the target point, the controller will be identical to algorithm 1. However, if agent i moves outside this distance, but is still connected to agent j, the energy function will be that of the squared second norm:

$$\mathcal{E}_{ij}(x) = \frac{1}{2}||x_i - x_j||^2 \tag{6}$$

$$\dot{x}_i = -\frac{\partial \mathcal{E}}{\partial x_i} = \sum_{j \in \mathcal{N}_i} (x_i - x_j) \tag{7}$$

The Min-Max Distance controller is shown to be stable within the specified Δ. The new specified energy function is shown to converge for static graphs by the Rendezvous problem. Therefore, the switched system should be stable as long as the connection to the target is not lost.

Algorithm 3 Angled Line

```
$\mathcal{F}$ set of followers
$N$ is number of leaders
followed is array for tracking leaders
network is array for tracking agents
for $i \in \mathcal{F}$ do
    $\mathcal{N}_i$ set of neighbors of $i$
    $a$ is the target
    for $j \in \mathcal{N}_i$ do                                ▷ Target Decision
        if $j \leq N$ & followed($j$) is False then
            $a \leftarrow j$
            network($i$) $\leftarrow$ True
            followed($j$) $\leftarrow$ True
            break
        else
            if network($j$) then
                $b \leftarrow (i - j) > 0 \& (a - j) < 0$
                $c \leftarrow (j - a) < 0$
                if $b|c$ then
                    a $\leftarrow j$
                    network($i$) $\leftarrow$True
                else
                    break
                end if
            end if
        end if
    end for
    $x_a \leftarrow \begin{bmatrix} \delta \cos \theta_{line} \\ \delta \sin \theta_{line} \end{bmatrix}$          ▷ Controller
    $T_i$ is a coordinate transformation to $x_i$ frame
    $x_{a \to i} \leftarrow T_i \cdot x_a$
    if $||x_i - x_{a \to i}||^2 < \Delta$ then
        $w_{ia \to i} \leftarrow \frac{1 - \frac{\delta}{||x_i - x_{a \to i}||}}{(\Delta - ||x_i - x_{a \to i}||)^3}$
        $\dot{x}_i \leftarrow w_{ia \to i}(x_i - x_{a \to i})$
    else
        $\dot{x}_i \leftarrow (x_i - x_{a \to i})$
    end if
    if $\mathcal{N}_i = \emptyset$ then
        $\dot{x}_i \leftarrow \vec{0}$
    end if
end for
```

D. Flock

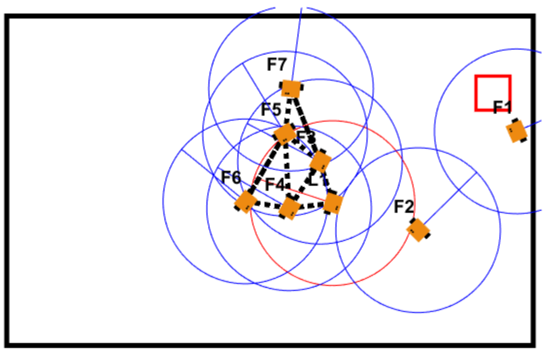

Fig. 4. Simulated run of Flock Algorithm with one leader.

Inspired by bird formations, this algorithm combines two angled lines from algorithm 3 above to achieve a more symmetrical and wider coverage area, centered around the leader.

To accomplish the formation, agent i will now discern between even and odd numbered agents. Even numbered agents will take to the line angled at θ_{line}, and odd numbered agents will take to the line angled at $-\theta_{line}$ (note that this change duplicates the leader being followed indicator).

III. Simulation & Results

A. Setup

Given the algorithmic nature of the controllers, Lyapunov function guarantees of convergence proved infeasible. Instead, we decided to simulate the algorithms over multiple runs and various conditions to profile the behavior. All of the algorithms utilized the simulator from the Robotarium [6].

The motion constraints of the robots utilized by the Robotarium combines two dynamic models, look-ahead and unicycle, into the following:

$$\dot{\tilde{x}} = vcos(\theta) - l\omega cos(\theta) \tag{8}$$

$$\dot{\tilde{y}} = vsin(\theta) + l\omega sin(\theta) \tag{9}$$

which can be inverted into

$$R(\theta) = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{l} \end{bmatrix} R(-\theta) \begin{bmatrix} \dot{\tilde{x}} \\ \dot{\tilde{y}} \end{bmatrix} \tag{11}$$

The control algorithms produced $(\dot{\tilde{x}}, \dot{\tilde{y}}, \theta)$, are passed into a Robotarium function to create the (v, ω) needed by the robots. These differential-drive controls are then also passed through a control barrier function ("CBF") that prevents the collision of robots [7]. The result is then commanded to the actual robots.

We needed to define as many equalizing constraints as possible to make the comparison of algorithms fair. We utilized many of the default characteristics of the physical Robotarium, including the robot diameter of 0.11 m, field size of 3.2 m by 2 m, maximum linear velocity of 0.2 m/s, and communication

Algorithm 4 Flock Algorithm

```
$\mathcal{F}$ set of followers
$N$ is number of leaders
followed is array for tracking leaders
network is array for tracking agents
for $i \in \mathcal{F}$ do
    $\mathcal{N}_i$ set of neighbors of $i$
    $a$ is the target
    for $j \in \mathcal{N}_i$ do                                    ▷ Target Decision
        if $j \leq N$ & followed($j$,mod$(i, 2) + 1$) is False then
            $a \leftarrow j$
            network$(i) \leftarrow$ True
            followed($j$,mod$(i, 2) + 1) \leftarrow$ True
            break
        else
            if network$(j)$ & mod$(i, 2)$ = mod$(j, 2)$ then
                $b \leftarrow (i - j) > 0 \& (a - j) < 0$
                $c \leftarrow (j - a) < 0$
                if $b|c$ then
                    a $\leftarrow j$
                    network$(i)$ $\leftarrow$True
                else
                    break
                end if
            end if
        end if
    end for
    if mod$(i, 2) = 0$ then                                    ▷ Controller
        $x_a \leftarrow \begin{bmatrix} \delta\cos\theta_{line} \\ \delta\sin\theta_{line} \end{bmatrix}$
    else
        $x_a \leftarrow \begin{bmatrix} \delta\cos -\theta_{line} \\ \delta\sin -\theta_{line} \end{bmatrix}$
    end if
    $T_i$ is a coordinate transformation to $x_i$ frame
    $x_{a\to i} \leftarrow T_i \cdot x_a$
    if $||x_i - x_{a\to i}||^2 < \Delta$ then
        $w_{ia\to i} \leftarrow \frac{1 - \frac{\delta}{||x_i - x_{a\to i}||}}{(\Delta - ||x_i - x_{a\to i}||)^3}$
        $\dot{x}_i \leftarrow w_{ia\to i}(x_i - x_{a\to i})$
    else
        $\dot{x}_i \leftarrow (x_i - x_{a\to i})$
    end if
    if $\mathcal{N}_i = \emptyset$ then
        $\dot{x}_i \leftarrow \vec{0}$
    end if
end for
```

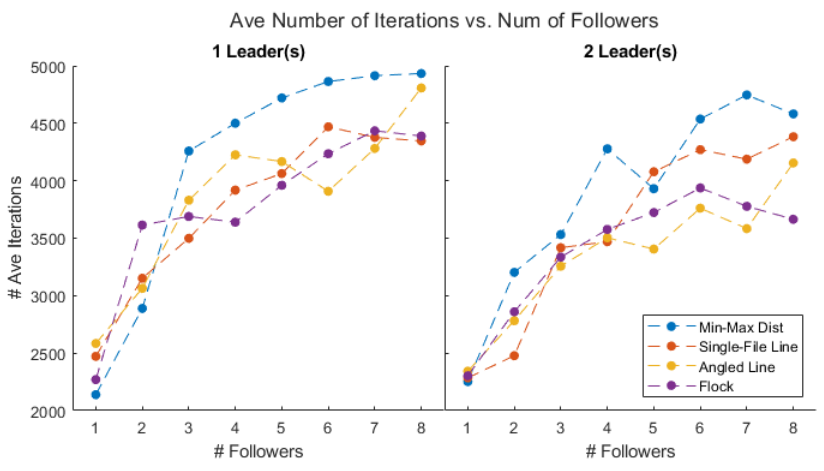

Fig. 5. Plots show the average number of iterations needed for connectivity against the number of followers for each algorithm for both 1 and 2 Leader(s).

time step of 0.033 s. Each agent had the same 360 deg vision distance radius of 0.5 m. All agents are randomly placed at least 0.1 m more than the vision distance away from each other to prevent connections at initialization. The maximum leader linear velocity was 0.08 m/s.

The parameters specific to the algorithms had the following values:

TABLE I
ALGORITHM PARAMETER TABLE

	δ	Δ	d	θ_{line}
Min-Max	0.132m	0.4m	-	-
Single-File	0.132m	0.4m	-	-
Angled Line	-	-	0.5m	-
Flock	0.132m	0.4m	-	$\frac{2\pi}{3}$

For these tests, the leaders randomly traverse the map by picking random valid coordinates. These are simulation constraints, but not agent restrictions and would not require global knowledge for those agents.

B. Simulations

The maximum number of iterations was 5000 loops and each algorithm ran 50 times. A run was considered "successful" if after running a connected component search all agents are at least weakly connected before the maximum number of iterations.

The number of leaders was chosen to be up to two and did not exceed that because the likelihood of all three randomly traversing leaders colliding end up being incredibly low. The number of followers was chosen to provide congestion to the system without breaking the constraint of minimum spacing being at least 0.1 m.

Having one follower for one leader proves as a trivial case for the algorithms since it was just the leader running the random path towards a stationary target, showing the variance present in both the random path and agent placement having an effect on the connection rate.

C. Analysis

The Min-Max Distance algorithm served as a "baseline" algorithm. The largest issue observed during initial testing of this algorithm was the consistent loss of controllability of the leader as the number of followers increased. The first issue is the fact that this algorithm is attempting to control a complete graph, which in theory would not be controllable in the first place. Furthermore, because of the CBFs, the robots cannot overlap, which means that if one robot is trying to get on the other side of another robot, it has to go around. This proves problematic if other robots are in the way on either side in the path around the robot, effectively trapping all robots involved. Because the leader performs the actual "searching" for agents, the lack of controllability prevents the random way-points from being hit and in effect stalling progress and preventing further connections. To this end, the Min-Max Distance seemed to plateau towards the 5000 iterations in 5.

All of the remaining algorithms (Single-file, Angled, and Flock) relied on a "chaining" aspect, in which the followers would trail behind the leader in some geometry, effectively creating a line graph.

Again, the CBF impacted the performance for much higher follower counts because these chains would be broken if a leader was surrounded, losing connectivity. This served as a trade-off because the leader remained controllable but may lose the already "found" followers if the maintained followers do not re-acquire them.

Additionally, the developed formations' lack of rigidity acted as a double-edged sword. In some cases, the flexibility in the formation allowed the followers to move out of the way of leaders since the leaders did not take into account other agents' positions for path planning. However, near the edges of the valid traversable areas, this flexibility could result in agents getting tangled as they tried to avoid collisions with walls or other agents.

As can be seen in Figure 5, the average number of iterations needed for convergence increases with the number of follow-

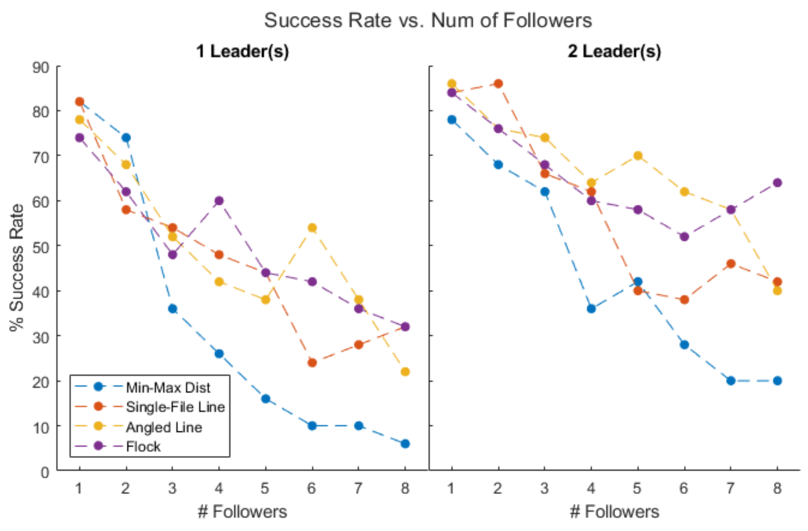

Fig. 6. Plots show the percentage of successful runs against the number of followers for each algorithm for both 1 and 2 Leader(s).

ers. Even though there is more coverage with a higher number of agents, the reduction in controllability of the system and the higher likelihood of breaking connections makes finding all agents an increasing challenge. The same conclusion can be reached from the results in Figure 6 since the success rate of the algorithms seems to decrease with increasing numbers of agents in most cases.

For both figures, comparing the Min-Max Distance to the formation indicate having the leader remaining controllable proved more advantageous over the possible temporary loss in connections.

The simulations with multiple leaders had increased performance for most algorithms. This is on one hand, due to having multiple agents performing the searching of agents, increasing the probability of finding all of them within the iteration limits. Furthermore, all algorithms, excluding Min-Max Distance, had some prioritization that allowed the hand-off of most agents to a single leader. This allowed at least one leader to have greater controllability and more opportunities to gather remaining agents.

IV. Conclusion

The initial goal was to increase likelihood of connectivity during random search and decrease the convergence time of agents starting in randomized positions. The results seem to show that all newly developed algorithms have increased performance against the Min-Max Distance controller in both of these aspects. However, the convergence rate for all algorithms decreases with increased agents due to the reduced controllability in complex systems.

Possible ways to improve the results should include implementing smarter search algorithms on the leaders, taking into account positions of neighbors to avoid barrier function collisions, and taking into account previously traveled areas to prevent traveling through the same points multiple times. Further research could also be done in making the developed formations more rigid; however, the controllability aspect would not be improved without the implementation of a better leader controller.

References

[1] W. Hale, "Frequency assignment: Theory and applications," *Proceedings of the IEEE*, vol. 68, no. 12, pp. 1497–1514, 1980.

[2] M. Mesbahi and M. Egerstedt, "Graph theoretic methods in multiagent networks," in *Princeton Series in Applied Mathematics*, 2010.

[3] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[4] J. Cortés and M. Egerstedt, "Coordinated control of multi-robot systems: A survey," *SICE Journal of Control, Measurement, and System Integration*, vol. 10, no. 6, pp. 495–503, 2017.

[5] J. Wang and E. Olson, "AprilTag 2: Efficient and robust fiducial detection," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, October 2016.

[6] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, vol. 40, no. 1, pp. 26–44, 2020.

[7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," 2019.