Decentralized Coverage Control for Multi-Agent Systems with Limited Connectivity Using Reinforcement Learning

Robert W. Stallworth IV and Erick J. Rodríguez-Seda
Department of Weapons, Robotics, and Control Engineering
United States Naval Academy
Annapolis, MD
m266060@usna.edu, rodrigue@usna.edu

***Abstract*—Multi-agent systems (MAS) can improve efficiency in surveillance and reconnaissance tasks by leveraging redundancy, communication resources, and cooperation. However, most MAS implementations still rely on centralized communication architectures, which can introduce single points of failure, be impractical for agents with limited communication ranges, and be challenging to scale up. While several researchers have studied how to maintain connectivity in decentralized systems, much less attention has been paid to the competing objectives of maintaining connectivity while increasing persistent coverage across large-scale domains. To this end, this paper proposes the use of Reinforcement Learning (RL) to train a group of agents to persistently cover a large geographical area while preserving inter-agent connectivity at all times. The reward function incentivizes the agents to visit each point in the domain regularly and penalizes them whenever global connectivity is lost. To measure coverage and connectivity in training, the function uses the concept of awareness and algebraic connectivity. To demonstrate the proposed approach, a simulation environment with N agents was created. The agents were trained using a Proximal Policy Optimization (PPO) model. It is shown that the group of agents can increase the awareness of the overall domain while also increasing the overall time the agents remained connected.**

***Index Terms*—Coverage Control, multi-agent systems, reinforcement learning, distributed systems**

I. Introduction

Multi-Agent Systems (MAS) have quickly gained popularity and importance in recent years. A MAS consists of multiple decision-making agents that interact in a shared environment to achieve common or conflicting goals [1]. They can range from a highly sophisticated swarm of drones tasked with military offensive missions to a collection of simple sensors used to gather environmental data [2]. Fig. 1 provides an example of a MAS supporting a naval vessel. The intersection of manned surface vessels and unmanned aerial systems can enhance mission capability, awareness, and survival.

One significant use of multi-agent systems is aerial mapping, scanning, and tracking. A swarm of drones with sensors can be used to track targets effectively from the sky with minimal human intervention [3]. The ability of these drones to maneuver and ensure the target remains visible to the sensor is a hallmark of multi-agent systems. If one agent is unable to complete the task, another can provide support, offering a more robust solution to the mission. Another challenging application of multi-agent systems is connecting multiple Artificial Intelligence (AI) models, each with a specific skill set. An AI multi-agent system would enable higher levels of reasoning, as each model can perform its specific operation and share its point of view with other models [4].

Fig. 1: Multi-agent systems supporting naval vessels to enhance mission capability [5].

A major challenge for MAS is the implementation of a decentralized network architecture. A decentralized architecture is one in which agents share information with one another rather than a central knowledge base. This can provide robustness against single-point failures, since the failure of a single agent may not cause the entire system to fail [6]. In addition, decentralized architectures can be easier to scale up since agents only share information with a small subset within the group [7]. In contrast, a centralized network architecture is one in which a global knowledge node shares information with each agent and can directly influence the actions of all other agents. A central node enables more streamlined control of the system at the expense of increased scale-up complexity

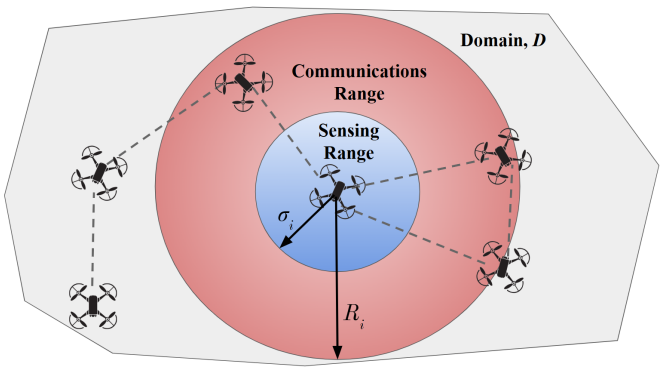
and single-point vulnerabilities [7].

Reinforcement Learning (RL) allows agents to learn optimal behaviors through interaction with the environment [8]. In this work, we employ Proximal Policy Optimization (PPO) [9], a policy-based RL algorithm selected for its stability and implementation efficiency [10]. By combining this modern algorithmic approach with traditional coverage control, we aim to enable agents to rapidly cover a region without losing connectivity. The selection of PPO ensures a simulation that is both computationally efficient and reliable.

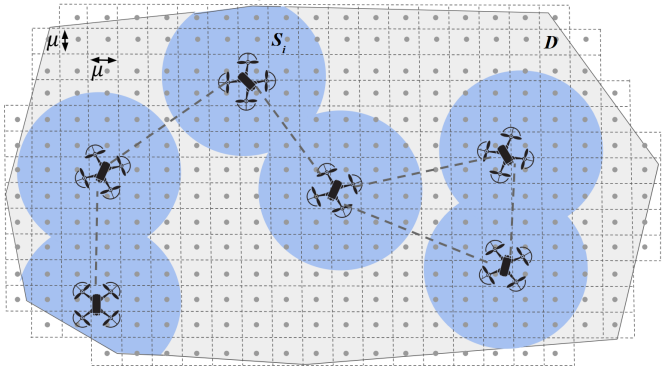
II. Problem Statement

This paper proposes a novel solution to the coordination of N agents tasked with persistently covering a large-scale, convex domain, $D \in \mathbb{R}^2$. To simplify the task, D is discretized into a grid of M $\mu \times \mu$ subregions as illustrated in Fig. 2. Then, the coverage control task can be defined as the group's ability to maintain persistent coverage of all M subregions.

To achieve the coverage objective, it is assumed that each agent has a radially bounded sensing region, $S_i := \{q \in D \mid \|q - q_i\| \leq \sigma_i\}$, where $q_i = [x_i, y_i]^T$ are the Cartesian coordinates of the ith agent and $\sigma_i > 0$ is the sensing range, and that the combined sensing region, $S := \cup_{i=1}^{N} S_i \subset D$, is always significantly smaller than the entire domain D. The latter implies that the agents need to maintain continuous motion to ensure persistent coverage. Moreover, each agent can communicate only with other agents within its communication range, R_i, which is assumed to be larger than the sensing range, as illustrated in Fig. 2a. It is assumed that the agents are working cooperatively and can share coverage information whenever they are within each other's communication range. Without loss of generality, we assume that the communication range is the same for all agents.



(a) Sensing and communication range. The dashed lines represent communication between agents.



(b) Discretization of the domain, D, as a square grid and the combined sensing region, $S = \cup_{i=1}^{N} S_i$, in blue shade.

Fig. 2: Multi-agent coverage control problem.

A. Connectivity Maintenance

It is said that the jth agent is adjacent to, or a neighbor of, the ith agent if they can communicate. The set of all neighbors of the ith agent is known as its neighborhood N_i. This adjacency information is represented by the symmetric adjacency matrix, A, where the ijth element, a_{ij}, of A is 1 if the ith and jth agents are neighbors and zero otherwise. Similarly, the diagonal $N \times N$ degree matrix, Δ, denotes the degree of each agent, that is, the size of its neighborhood, N_i. Then, the Laplacian matrix can be defined as $L = \Delta - A$, which encodes information about the system's connectivity. We said that the agents are connected if there is a communication path between any arbitrary pair of agents. Mathematically, the MAS is connected if the second smallest eigenvalue of L, denoted as λ_2, is non-zero [11]. This value, also known as the Fiedler value, also serves as an indicator of how well connected the group is, with higher values indicating a larger number of communication paths.

B. Awareness Model

To measure how well the MAS covers the domain, the concept of awareness is adopted [12], which is a function of the time agents spend in a subregion. It is assumed that the agents can communicate their awareness values with other agents in their neighborhood. Then, the awareness value that the ith agent has about the ℓth subregion, denoted as $\zeta_{i\ell}(k)$, is given as

$$\zeta_{i\ell}(k+1) = (1-\alpha_\ell) \max_{j \in \mathcal{N}_i} \zeta_{j\ell}(k) + \sum_{j \in \mathcal{N}_i} s_{j\ell}(k), \quad \zeta_{i\ell}(0) > 0 \tag{1}$$

where $\alpha_\ell \in (0,1)$ represents the loss of awareness over time and $\mathcal{N}_i = N_i \cup i$. The gain of awareness over the ℓth subregion is denoted by

$$s_{i\ell}(k) = \begin{cases} \beta_i & \text{if } \|q_i(k) - \tilde{q}_\ell\| \leq \sigma_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $\tilde{q}_\ell$ are the coordinates of the center of the ℓth subregion (see Fig. 2b) and $\beta_i > 0$ is the ith agent's sensing capability or strength. Note that the awareness value (1) that the ith agent has about an ℓth subregion increases every time that the subregion is visited by the ith agent or one of its neighbors, and decreases otherwise. Furthermore, note that the awareness metric is bounded by

$$0 < \zeta_{i\ell}(k) \leq \frac{N \cdot \max_i \beta_i}{\min_\ell \alpha_\ell}, \qquad \forall\ k, i, \ell. \tag{3}$$

with higher values representing more recent coverage. Finally, one has that for a connected MAS, the awareness values among agents are equal, i.e., $\zeta_{i\ell}(k) = \zeta_{j\ell}(k)\ \forall i \neq j, \ell \in \{1, \cdots, M\}$ if $\lambda_2(k) > 0$.

In addition to measuring the time and recency of an agent in a given region, it is useful to log the last time an agent visited the region. Therefore, we denote $\tau_{i\ell}$ as the last time that the ith agent visited the ℓth region and

$$\bar{\tau}_{i\ell}(k) = \max_{j \in \mathcal{N}_i} \{\tau_{j\ell}\} \tag{4}$$

as the record that the ith agent has about the last time that the ℓth region was visited by an agent.

C. Vehicle's Kinematic Model

It is assumed that agents can move to a single subregion at each time step and that each agent can move in any cardinal direction or pause. Accordingly, their next position is given by

$$q_i(k+1) = q_i(k) + g_i(k) \tag{5}$$

where

$$g_i(k) \in \left\{ \begin{bmatrix} 0 \\ \mu \end{bmatrix}, \begin{bmatrix} 0 \\ -\mu \end{bmatrix}, \begin{bmatrix} \mu \\ 0 \end{bmatrix}, \begin{bmatrix} -\mu \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\} \tag{6}$$

is the control input.

D. Control Objective

The objective of this research is to develop an AI-based, decentralized, multi-agent control framework capable of coordinating the motion of N autonomous agents to maintain a high level of awareness across the entire domain while remaining connected most of the time. Mathematically, we would like to train the agents such that $\lambda_2(k) > 0$ almost everywhere and that, for some desired threshold performance $\bar{z} > 0$ and after some time $T > 0$, $\zeta_{i\ell}(k) > \bar{z}$ for all $k \geq T$ and $\ell \in \{1, \cdots, M\}$.

III. Reinforcement Learning

As a reinforcement learning agent, the proposed approach uses the PPO algorithm developed by OpenAI [8]. The PPO algorithm seeks to ensure stable policy updates while maximizing the rewards. It constrains large policy updates that can destabilize training by using a clipped objective function. For more information about the PPO algorithm, the reader can refer to [9].

A. Agent

In reinforcement learning, the agent selects an action, $g_i(k)$, from its action space and receives an observation or outcome from the environment, along with a reward, as seen in Fig. 3. The action space for an agent is the set of motions the agent can take: move north, south, west, east, or hold.

B. Environment

The environment consists of three key components: the step function, the observations, and the reward function.

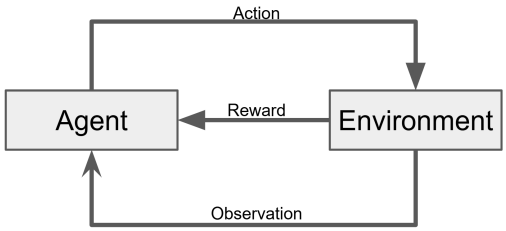


Fig. 3: Flow of reinforcement training process.

1) Step Function: The step function is the core of the environment. It takes in the actions of each agent and the information about the environment, including the positions of the agents, the simulation time, and the awareness values $\zeta_{i\ell}(k)$. The step function takes in the actions of each agent $g_i(k)$ from the discrete action space of $(north, south, west, east, hold)$ and updates the location of the respective agent. After each agent moves, the step function checks for collisions. To resolve collisions during training and avoid the case of two agents occupying the same cell, the location of the ith agent is moved diagonally one-step north-east, unless that would push it out of D, in which case it moves south-west. The agents are penalized for this collision, and this resolution response is not used after training.

Once the agent's position is updated, the adjacency matrix and the algebraic connectivity of the agents are calculated. The final portion of the step function updates the awareness of D according to the model outlined in Section II-B. The step function is described by Algorithm 1.

Algorithm 1 Step Function

```
for Each time step k do
  for Each agent N do
    Obtain the action, g_i
    Update the position
  end for
  Ensure no collisions have occurred
  Calculate A and λ_2
  for Each Agent N do
    Determine Sensed Region, S_i
    Update ζ_iℓ and τ̄_iℓ based on connectivity
    Determine agent rewards, r_i
  end for
end for
```

2) Observation: The observation space is the set of environmental states from which the agent must generate an action. The observation space for each agent contains the ith agent's position and the awareness values $\zeta_{i\ell}$. While the adjacency matrix, A, and the positions of the other agents, q_i, are calculated in the step function, they are not passed as observations to the agents.

3) Reward Function: The goal of the reinforcement model is to maximize awareness, $\zeta_{i\ell}$, of the entire domain D. Higher values of $\zeta_{i\ell}$ indicate a higher awareness, meaning that at least one agent has recently observed the ℓth subregion for a longer

period of time. Therefore, the reward given to the agent must include the awareness value, $\zeta_{i\ell}$. However, since $\zeta_{i\ell}$ has a large range according to (3), it is convenient to normalize the awareness values between 0 and 1. Accordingly, we define the normalized awareness value that the ith agent has about the ℓth subregion as

$$Z_{i\ell}(k) = \frac{\zeta_{i\ell}(k)}{\sqrt[10]{1+\zeta_{i\ell}^{10}(k)}} \tag{7}$$

Then, the goal of the agents is to ensure that the normalized awareness values across the space reach or exceed a threshold value $\bar{Z} = \bar{z}/\sqrt[10]{1+\bar{z}^{10}} > 0$.

In addition to awareness, the reward should encourage exploration, connectivity, and safety (i.e., collision avoidance). Therefore, we propose the following reward function

$$r_{i,A}(k) = h_A \sum_{\ell=1}^{M} (Z_{i\ell}(k) - Z_{i\ell}(k-1)) \tag{8a}$$

$$r_{i,E}(k) = h_E \sum_{\ell=1}^{M} e_{i\ell}(k) \tag{8b}$$

$$r_{i,C}(k) = \begin{cases} h_C, & \text{if } \lambda_2(k) > 0 \\ -h_D, & \text{otherwise} \end{cases} \tag{8c}$$

$$r_{i,B}(k) = \begin{cases} -h_B, & \text{if } q_i(k) \notin D \\ 0, & \text{otherwise} \end{cases} \tag{8d}$$

$$r_{i,S}(k) = \begin{cases} -h_S, & \text{if } \|q_i(k) - q_j(k)\| \leq \mu, \text{ for } i \neq j \\ 0, & \text{otherwise} \end{cases} \tag{8e}$$

$$r_i(k) = r_{i,A}(k) + r_{i,E}(k) + r_{i,C}(k) + r_{i,B}(k) + r_{i,S}(k) \tag{8f}$$

where

$$e_{i\ell}(k) = \begin{cases} 1, \text{ if } \bar{\tau}_{i\ell}(k-1) \leq k - h_T \ \& \ \|q_i(k) - \tilde{q}_\ell\| \leq \sigma_i \\ 0, \text{ otherwise} \end{cases}$$

flags when a region that has not been visited in more than h_T instances is visited by the ith agent and h_A, h_E, h_T, h_C, h_D, h_B and h_S are all positive hyperparameters. Note that $r_{i,A}$ rewards the ith agent whenever the net gain of awareness over D is positive, while $r_{i,E}$ incentivizes the ith agent to explore or revisit a point in a h_T time window. The term $r_{i,C}$ incentivizes the agents to remain connected, while $r_{i,B}$ and $r_{i,S}$ penalize the ith agent whenever it exists D or collides with another agent.

IV. Simulation Results

Each PPO agent was trained over 500 episodes. They have two fully connected layers with 256 nodes, a ReLU layer in between, and a final fully connected layer with 5 nodes and a softmax layer. For training, the learning rate was set to 0.001. The maximum number of steps per episode was 500, with each step lasting 0.1 seconds, leading to a maximum simulation time of 50 seconds. The hyperparameters for all agents were chosen as $h_A = 2$, $h_E = 10$, $h_T = 50$, $h_C = 0.5$, $h_D = 20$, $h_B = 1$, and $h_S = -20$. The parameters for the awareness model were chosen as $\alpha_\ell = 0.1$ and $\beta_i = 1$ $\forall\ i, \ell$. The N agents were each trained simultaneously in the same environment. As a training example, Fig. 4 shows the average reward per agent over time for $N = 4$. Note that the average reward starts to stabilize after 50 episodes.

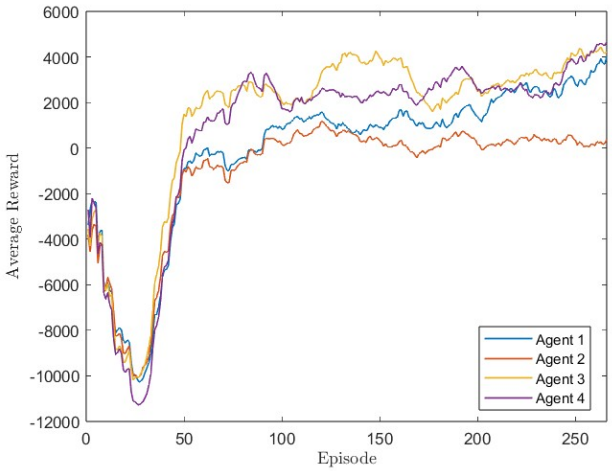


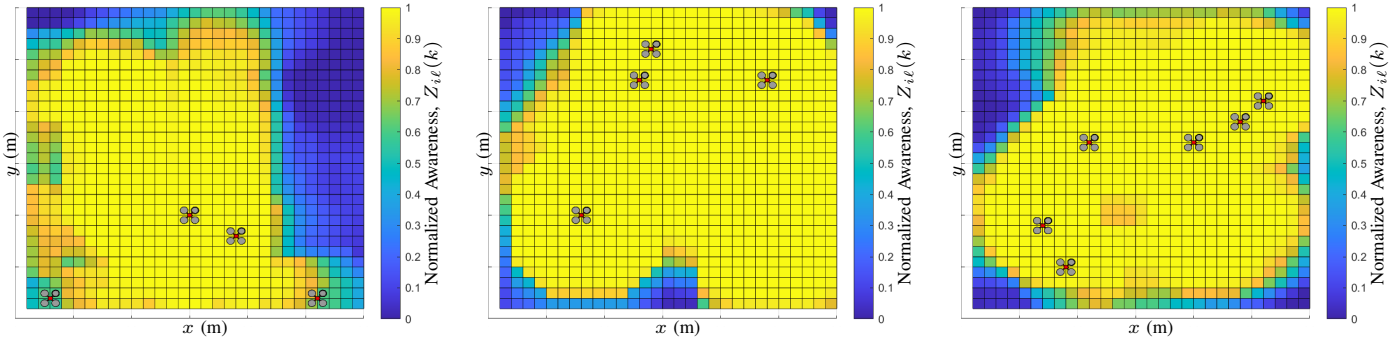
Fig. 4: Average reward per PPO agent during training.

Twelve different configurations were trained under these presets, with N, σ_i, and R_i varied to produce different operating configurations. Table I lists the configuration for each environment's training, the minimum and maximum awareness at the final time step, the mean awareness throughout the simulation, and the connectivity score for each simulation. The reported awareness is that of the agent with the highest overall awareness of the MAS. Each of the aforementioned values is an average from over ten simulations with different initial conditions. The connectivity score is given by

$$\xi = \frac{\sum_{c=1}^{N} (N-c)P(c)}{N-1} \tag{9}$$

and is a function of the number of clusters, c, and the percentage of time that they were observed, $P(c)$. It ranges from 0 to 100, with 0 indicating that agents did not form a single connection and 100 indicating that they were all connected at all times. Note that, in the case that agents do not form a single connected cluster, ξ also increases with a lower number of clusters.

Fig. 5 shows the initial and final awareness for three separate configurations. The three configurations are: four agents with $\sigma_i = 5$m and $R_i = 15$m; four agents with $\sigma_i = 7$m and $R_i = 15$m; and six agents with $\sigma_i = 5$m and $R_i = 15$m. The configurations with a sensing radius of 7m consistently return the highest mean awareness values, regardless of the number of agents. These configurations differ in effectiveness when connectivity is considered. With 2 and 4 agents and R_i equal to 15m, the agents maintained a connectivity score above 87, indicating that most of the time was spent fully connected. The 6-agent configuration has a connectivity score of 86.0, slightly less than it's 2 and 4 agent counterparts. When



(a) Final Awareness of D with Communications Radius equal to 15m, and Sensing Radius equal to 5m, and four agents.

(b) Final Awareness of D with Communications Radius equal to 15m, and Sensing Radius equal to 7m, and 4 agents.

(c) Final Awareness of D with Communications Radius equal to 15m, and Sensing Radius equal to 5m, and six agents.

Fig. 5: Initial and Final awareness of D using a trained PPO.

Case			**Awareness Values**, $Z_{i\ell}(k_{end})$			**Connectivity**
N	σ_i (m)	R_i (m)	Min	Max	Mean	Score
2	5	5	0.0000	0.9966	0.3304	31.000
2	5	15	0.0000	1.0000	0.6808	93.800
2	7	5	0.0000	0.9998	0.6387	46.400
2	7	15	0.0000	1.0000	0.8852	97.200
4	5	5	0.0000	0.9976	0.3286	16.467
4	5	15	0.0000	1.0000	0.6468	89.933
4	7	5	0.0000	0.9999	0.6891	21.933
4	7	15	0.0000	1.0000	0.8474	87.133
6	5	5	0.0000	0.9974	0.3907	43.880
6	5	15	0.0000	1.0000	0.8365	94.960
6	7	5	0.0060	1.0000	0.746	31.880
6	7	15	0.0000	1.0000	0.8008	86.000

TABLE I: Minimum, Maximum, and Mean Awareness Values for Simulated Examples as well as Connectivity Score

considering the region's average awareness, the performance difference between the three configurations is not significant. It should be noted that the system is only considered connected when there is a path between all agents. An agent can continue to communicate with its direct neighbors if the system is not completely connected. When the agents become connected, they will share all information it currently has with their neighbors, increasing awareness of the region.

As seen by the simulation results, the optimal configuration is $N = 2$, $R_i = 15$, and $\sigma_i = 7$, achieving an average awareness of 0.8852, and a connectivity score of 97.200. By being disconnected part of the time, and maximizing the awareness value of the region, this configuration meets both operational goals. Although the number of agents must be considered for practical reasons. In an operational environment, if an agent is lost, awareness will drop significantly. While increasing the number of agents reduces performance in terms of awareness and connectivity, the additional redundancy makes them equally attractive options.

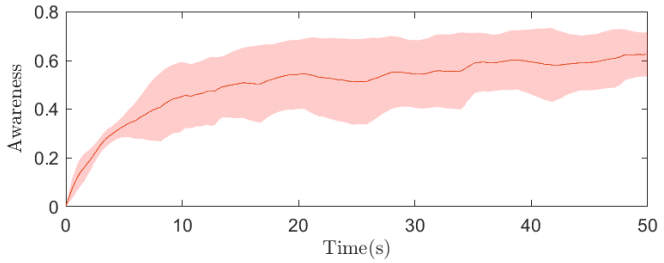
Similarly, Fig. 6 shows the average awareness of D over time for ten separate simulations for the same three configurations mentioned above. The red-shaded area around the trend line indicates one standard deviation. These confirm that the third configuration is the most consistent in maintaining a satisfactory level of awareness over the region and across multiple simulations. The larger deviations in the two-agent configuration highlight concerns about such a small number of agents.

V. Conclusion and Future Work

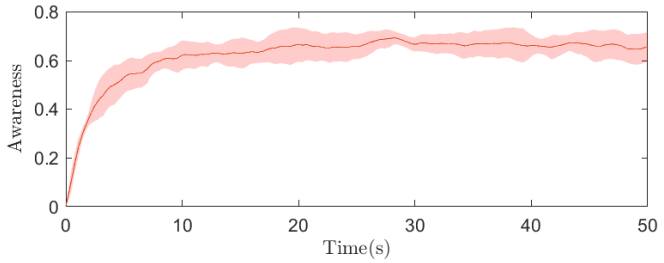
This paper introduces a novel reinforcement learning approach to decentralized coverage control in large-scale domains using the PPO algorithm. The PPO agents successfully learned how to increase the awareness, $Z_{\tilde{q}_l}$ while also ensuring the Fiedler value λ_2 remained greater than zero. The preliminary simulation results indicate that the agents are capable of surveying the region, and maintaining connectivity. Each configuration was able to maintain a steady state level of awareness while maintaining connectivity.

While the simulation with 4 agents and a discretized action space worked to show that coverage control of a region using this approach is possible, there are ways to improve the effectiveness of the approach. The reward function can be improved to better account for the number of agents that cover a given area, incentivize shorter travel distances, and better account for regions with the lowest awareness. As shown by the simulation results, there is no incentive for the agents to explore low-awareness regions under the current reward function. Additionally, creating a continuous action space could allow for more flexibility within the agent's actions. Improvements to the reward function can also work to increase the awareness of D.

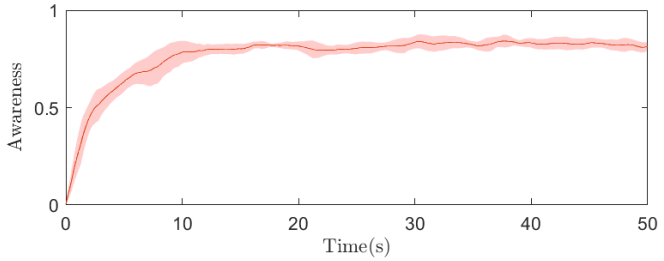
Future work would seek to increase the complexity of the environment and agents themselves. The current model assumes vehicles with perfect sensing capabilities. Changing agent kinematics and sensing capabilities would add complexity more akin to the real-world problem. Additionally, adding static obstacles or increasing the complexity of D would challenge the PPO model and further increase the effectiveness of this model in real-world scenarios. Lastly, applying this problem to a group of physical robots moving in two dimensions would provide a physical proof of concept for this approach to coverage control.



(a) Average Awareness over time for 10 separate simulations of 2 agents with Communications Radius equal to 15m and Sensing Radius equal to 5m.



(b) Average Awareness over time for 10 separate simulations of 4 agents with Communications Radius equal to 15m and Sensing Radius equal to 5m.



(c) Average Awareness over time for 10 separate simulations of 6 agents with Communications Radius equal to 15m and Sensing Radius equal to 5m.

Fig. 6: Average Awareness over time for 3 different simulations using the same PPO Model.

REFERENCES

[1] Z. Feng, X. Dong, G. Hu, and J. Lyu, *Resilient Cooperative Control and Optimization of Multi-Agent Systems*. Elsevier, 2025.

[2] A. Freitas, C. A. Rabbath, C. Williams, N. Lechevin, and S. Givigi, "Multi-unmanned aerial vehicles cooperative tactics: A survey," *IEEE Access*, vol. 13, pp. 119 897–119 921.

[3] D. Hambling, "U.S. drones team up to find and track targets," *Forbes*, Mar. 2025, [Online]. Available: https://www.forbes.com/sites/davidhambling/2025/03/03/us-drones-team-up-to-find-and-track-targets/.

[4] F. Tian, A. Luo, J. Du, X. Xian, R. Specht, G. Wang, X. Bi, J. Zhou, A. Kundu, J. Srinivasa, *et al.*, "An outlook on the opportunities and challenges of multi-agent AI systems," *arXiv preprint arXiv:2505.18397*, 2025.

[5] T. Rogoway, "The compelling case for arming u.s. navy warships with drone swarms," The War Zone, Apr. 2024, [Online]. Available: https://www.twz.com/sea/the-compelling-case-for-arming-u-s-navy-warships-with-drone-swarms.

[6] L. Poudel, S. Elagandula, W. Zhou, and Z. Sha, "Decentralized and centralized planning for multi-robot additive manufacturing," *Journal of Mechanical Design*, vol. 145, no. 1, p. 012003, 2023.

[7] A. Dimakos, D. Woodhall, and S. Asif, "A study on centralised and decentralised swarm robotics architecture for part delivery system," *Academic Journal of Engineering Studies*, vol. 2, no. 3, 2021.

[8] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE signal processing magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[9] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[10] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep RL: A case study on PPO and TRPO," in *International Conference on Learning Representations*, 2019.

[11] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[12] Y. Wang and I. I. Hussein, "Awareness coverage control over large-scale domains with intermittent communications," *IEEE Transactions on Automatic Control*, vol. 55, no. 8, pp. 1850–1859, 2010.