# Real-Time Flood Navigation of Waterborne Vehicles using MEA*

Given Name Surname
*Department of Computer Science & Engineering*
*Indian Institute of Technology Ropar*
India
email address

***Abstract*—Autonomous navigation and formation control of multi-UAV systems pose a significant challenge for the robotic systems that operate in partially observable, dynamic and continuous environments. This paper addresses the problem of multi-UAV cooperative sensing and coverage of a flood-struck region to identify serviceable paths to critical locations for waterborne vehicles (WBV) in real time. A serviceable path is defined as a location that is obstacle free and has adequate water level for possible movement of WBVs. We develop a deep reinforcement learning model to learn a cooperative multi-UAV policy for real-time coverage of a flooded region. The coverage information gathered by the UAVs captures the presence of obstacles present in the path connecting the start and target/critical locations given by the shortest Manhattan distance. This coverage information is utilized by the path planning algorithm, i.e., MEA$^*$, to minimize the number of expansion nodes and identify a serviceable path quickly. To conserve energy, UAVs initially follow a guided path to explore the optimal route. If obstacles are encountered, the UAVs search nearby areas for an alternate path to reach the critical location(s). The proposed approach, MEA$^*$MADDPG, is compared with other prevalent techniques from the literature over real-world inspired simulated flood environments. The results highlight the significance of the proposed model as it outperforms other techniques when compared over various performance metrics.**

***Index Terms*—Unmanned Aerial Vehicles (UAVs), Deep Reinforcement Learning, Real-Time Path Planning, Disaster Management**

## I. Introduction

Real-time path planning in unknown environments is a non-trivial task, for example, identifying a serviceable path to reach critical locations during floods. In case of floods, a location is said to be critical, where victims are known to be at risk and need evacuation due to high flood water accumulation [1]. In order to perform real-time path planning, access to ground information of the unknown environment is crucial to identify a path from a given start location to the critical location(s). To gather such information about the environment, recent studies have proposed UAV based models to capture the environment dynamics [2]. However, UAVs need to have adaptive control policies to perform complex tasks in unknown environments.

In this paper, our objective is to deploy UAVs in an interleaved formation to identify serviceable paths to critical location(s) in an unknown flood-struck region. This objective is break-down into two sub-objectives, where the first sub-

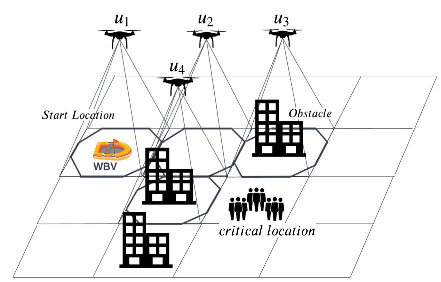


Fig. 1. End-to-end UAV coverage to identify a serviceable path for WBV to reach critical location(s).

objective is that the UAV needs to capture environment-related information in a connected formation so as to have a large Field of View (FoV) of the surface with end-to-end coverage as seen in Figure 1. Secondly, a serviceable path has to be identified by utilizing this sensed information to make the waterborne vehicles (WBVs) reach the critical location(s) to evacuate affected people. For the former sub-objective, there are quite a few control algorithms for UAVs that exist, such as a trajectory optimization based model or a heuristic-based approach such as Particle Swarm Optimization (PSO), etc [3]. However, it is difficult to attain realistic implementations for UAVs using such techniques when the environment is unknown. To address this problem we focus on Reinforcement learning (RL) [4] methods aiding with specially designed frameworks to learn autonomous control policies for UAVs by interacting with the environment in a trial-and-error fashion. The intended control policy is a mapping between the state of the UAV and its feasible actions.

For the latter sub-objective, we emphasize on prominent path-planning algorithms that can be employed to identify feasible paths. One such algorithm is the A$^*$ search algorithm [5] which is the most widely used heuristic based path-planning algorithm. It generates the global optimal paths dynamically and can theoretically guarantee the convergence to the globally optimal solution [6]. Such characteristics make it suitable for real-time path planning in dynamically changing maps. Another prominent technique that is widely used for path planning is a sampling-based approach, known as rapidly exploring random tree (RRT) [7]. The RRT algorithm works by constructing a tree-like structure to incrementally explore the unknown environment by randomly sampling coordinates from the map. For the considered problem of identifying a connected path over a grid space, A$^*$ is deemed to be a better-suited approach and hence we opt for an A$^*$ based path planning model. However, as the A$^*$ search algorithm suffers from node expansion resulting in storage memory saturation in large state-space environments [8], it sometimes becomes infeasible to apply the A$^*$ based path planning model. To address this, we propose a technique called minimized node expansion (MEA$^*$). It involves restricting the expansion of nodes by removing unserviceable nodes based on real-time information gathered using UAVs.

To learn an autonomous control policy for the multi-UAV system this paper employs the Multi-agent Deep Deterministic Policy gradient (MADDPG) algorithm [9] as it is well suited to problems with continuous state and action spaces. The reward function is designed in such a manner that the UAVs are able to learn joint actions so as to maintain end-to-end coverage for connected path planning. The information captured by the multi-UAV system is utilized by the path planning algorithm to update its heuristic function and identify a serviceable path quickly and in real time. Real-time path planning corresponds to UAVs taking action in a short and finite amount of time.

## II. Related Literature

### A. Autonomous Multi-UAV Formation Control

The task of autonomous formation control is to align the UAVs (without human intervention/pilot control) in such a manner as to have a larger area under their joint FoV or capture the object of interest without overlapped sensing. Many recent studies have addressed the challenge of formation control using heuristic and RL based approaches while working with multi-UAV systems. In [10], authors proposed a swarm formation control using particle swarm optimization (PSO) with Cauchy mutant (CM) operators for aerial vehicles. CM operators are applied to enhance the PSO algorithm by inspecting varying fitness levels of the local and global optimal solutions for UAV formation control. In another study addressing the multi-UAV formation control problem, authors in [11] proposed a MADDPG based model where the UAVs are tasked with tracking of moving objects while maintaining an optimal formation. A well-designed reward function is formulated to incorporate tracking, formation control and collision avoidance objectives in policy learning. In another RL based study, the authors in [12] discuss a model-free solution for a leader–follower formation control problem of heterogeneous multi-agent systems. The objective of the multi-agent formation control is to guarantee that the agents form the desired formation determined using a set of offset values and track the trajectories generated by the virtual leader. Authors in [13], propose a data-driven solution for formation control of heterogeneous quadrotors in air–ground coordination. The position and heading angle for the quadrotor and the unmanned ground vehicles (UGV) are generated using the local information of themselves and their neighbours. Based on RL theory, an optimal formation controller for the heterogeneous agents is learned without knowing the agent dynamics.

### B. Path Planning

Path planning is another aspect of this study and there have been numerous studies proposing novel methods to achieve real-time path planning. Real-time path planning involves taking control decisions within a limited time frame in a dynamically changing environment. In [14], authors proposed a new method, namely, RRT$^*$-Smart to accelerate the convergence of RRT. The proposed method was tested in a simulated environment with obstacles and compared to the RRT algorithm for performance evaluation. In [15] authors proposed an Improved Artificial Bee Colony algorithm (IABC) for real-time path planning of a multi-UAV system. After the initial path is determined, the IABC algorithm optimizes a certain set of waypoints decided by the cost value of the path, thereby avoiding a random search of the entire space area. In [2], authors proposed a DRL based approach for UAV path planning using global observation information. A set of observed global maps are provided as input to the dueling double deep Q-network (D3QN) algorithm to approximate the value function corresponding to all feasible actions.

In another study [16], authors addressed the problem of path planning for wheeled mobile robots over a partially known terrain. A$^*$ algorithm is employed to track the global path with the available partial information of the environment. Further, Q-learning is used for local planning to avoid nearby detected obstacles to the wheeled robot. In [5], authors analyzed the performance gap between A$^*$ and RRT$^*$ using a realistic simulator to handle the dynamic properties of the robot. In [7] authors proposed a real-time path planning approach, known as an executive rapidly-exploring random tree (ERRT) for path planning in unknown environments. ERRT maintains a waypoint cache and updates it when a path is found. Also, a path cost is taken into account to allow new nodes to be selected more closely to the target waypoint.

Motivated by the current literature, we address the problem of path planning of WBV(s) to reach critical location(s) in an unknown flooded environment using an improved A$^*$ approach. The improvement is introduced in the form of environment-related knowledge captured by the UAVs that helps in minimizing the number of nodes expanded by the A$^*$ algorithm in order to identify a serviceable path from a start location to the critical location(s). Further, the UAVs align themselves in the optimal formation so as to capture a larger FoV and help in planning an end-to-end serviceable path by avoiding coverage gaps as illustrated in Figure 1.

## III. Flood Environment Description

The flood environment consists of an underlying surface seen as a grid with $a_1 \times a_2$ cells with identical side dimensions. The surface is encoded with real-world road-transit network information along with land use data (such as buildings, water bodies, etc.) gathered using Mapbox Streets [17]. Further, to realize flood-like dynamics, a 2D water mask is defined over the environment to simulate the flood using the Mapbox Bathymetry tool [18]. In reference to ASTRALiTe's edge [19] i.e., a small topographic and bathymetric scanning drone lidar used for detecting small underwater objects/infrastructure, and measuring shallow water depths, we categorize a captured location (a location sensed by a UAV) as serviceable or not.

*Serviceable Path:* We aim at identifying serviceable paths for WBV to reach critical location(s). A location $c$ is said to be serviceable $\Lambda^c$ if the underlying surface is clear of underwater objects and the water is not shallow for possible movement of WBV.

$$\Lambda^c = \begin{cases} 1(serviceable) & if\ \mathbb{W}_l^c \geq \mathcal{W}\ and\ O^c == False \\ 0(unserviceable) & otherwise \end{cases} \tag{1}$$

where $O_c$ denotes whether the location $c$ has an underlying obstacle/object(s) and $\mathbb{W}_l^c$ is the water level/depth at location $c$. $\mathcal{W}$ denotes the threshold level below which the water at cell $c$ is said to be shallow (unsafe for WBV movement).

## IV. System Model

In the system model, we have a team of $n$ UAVs ($U$): $\{u_1, u_2, ..., u_n\}$ that are tasked to perform area coverage for serviceable path planning for a waterborne vehicle by sensing the region between the start location of the vehicle and the location of critical areas (the start and critical locations are always known). The objective is to find connected locations (interlinked cells of the environment grid) so as to make the WBV reach the critical location(s). Figure 2 describes the considered scenario where a team of 4 UAVs ($u_1, u_2, u_3, u_4$) are sensing the cells ahead of the WBV in the direction of the closest critical location to identify a serviceable path for the WBV to move (usually in the direction of the critical location). The WBV is perceived in the joint FoV at all times to identify the start location. In this sense, our start location keeps on changing due to the movement of the WBV. Each UAV is equipped with ventral cameras to capture the underlying surface in their FoV. The FoV of each UAV forms a 3D pyramid having a hexagonal field of view as shown in Figure 3. The joint FoV of the UAVs is given by the set $\mathcal{F} = \{f_{u_1}, f_{u_2}, ..., f_{u_n}\}$, where, $f_i$ is the FoV of the $i^{th}$ UAV. The overall objective of the system for maintaining formation amongst UAVs and serviceable path planning is given as:

$$\max \sum_{t=0}^{\infty} \left( \sum_{u_i \in U} \mathbb{T}_t(u_i, \Psi)\Lambda^c + \sum_{f_i, f_j \in F} \mathbb{I}_t(f_i, f_j) + \varphi \Lambda^c \right) \tag{2}$$

where, $t$ denotes time. $\mathbb{T}(.,.)$ is the trajectory alignment reward w.r.t. the guided path ($\Psi$) given by the shortest Manhattan distance from start to critical location. $\mathbb{I}(.,.)$ represents the FoV overlap incentive so that the UAVs learn to maintain end-to-end coverage with no gaps. $\varphi$ is a scalar incentive given to the multi-UAV model if the sensed cells provide a serviceable path. If an obstruction is captured by a UAV the guided path is no longer deemed to be serviceable and the UAVs help in adjusting the WBV trajectory (i.e., the guided path) by sensing neighbouring regions (see Figure 2).

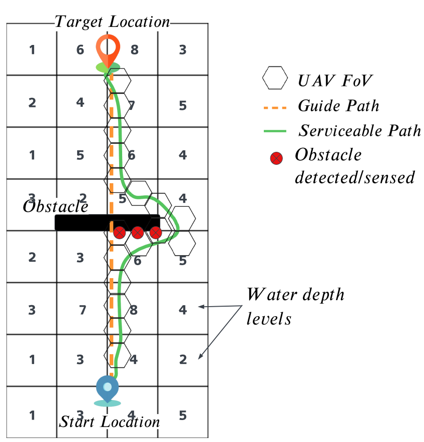


Fig. 2. Multi-UAV coverage for serviceable path planning (depicted by green color) w.r.t. to a guided path (depicted by orange color), considering obstacles and shallow/unserviceable water regions (a cell having a water level of 3ft and less). The cell value denotes the current water level at that location in feet.

## V. Proposed Model

The multi-UAV coverage task is formulated as an MDP $< S, A, T, R, s_0,$
$s_{start}, s_{critical} >$, where $S$ denotes the joint state of the UAVs $(s_{u_1} \times s_{u_2} \times ... \times s_{u_n})$. $s_{u_i} : \{u_{i_t}^c, \psi^c\ \rho_t^{u_i}\}$ depicts the state of the $i^{th}$ UAV, where $u_{i_t}^c$ is the cell occupied by $u_i$ at time $t$ and $\psi^c$ is the cell given by the guide path. $\rho_t^{u_i}$ is the energy level of $u_i$ at time $t$. A denotes the joint action of the multi-UAV system and is given as $(a_{u_1} \times a_{u_2} \times ... \times a_{u_n})$. Each UAV's action $a_{u_i}$ lies within the feasible action set $\Upsilon : \{yaw_{u_i}, pitch_{u_i}, hover_{u_i}\}$ where both $yaw$ and $pitch$ value ranges from [-1,1]. A zero value of $pitch$ denotes the UAV's parallel alignment with the horizontal axis and as the value goes below zero the UAV tilts where the head goes up and the UAV moves in a backward direction. As the value of $pitch$ goes above zero the UAV's head goes down to move in the forward direction. For $yaw$ action a value of zero corresponds to the the head of UAV being in the north direction. A value lower than zero tilts the UAV's head counter-clockwise and a positive value of $yaw$ tilts it in a clockwise direction. $T$ denotes the state transition function and $s_0$ describes the starting configuration of the multi-UAV system. $s_{start}$ denotes the start location and $s_{critical}$ represents the critical location(s) to be reached. $s_{critical}$ can be a set $\{\tau_1, \tau_2, ...\tau_m\}$, if multiple locations are to be reached.

To solve this MDP decision problem and to achieve autonomous and continuous control of the multi-UAV system, we employ a DRL method, known as Multi-agent Deep Deterministic Policy Gradient (MADDPG) [9]. It works in a manner where each UAV is controlled by a separate actor network that is responsible for providing the feasible action based

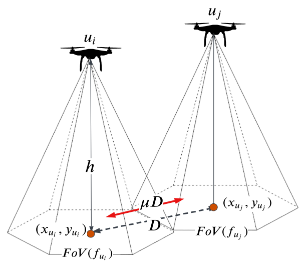


Fig. 3. UAVs $u_i, u_j$ with a joint hexagonal field of view (FoV) and ideal overlap.

on the UAV's current state and a centralized critic network that validates the effectiveness of the actions generated by the actor. Also, a set of target actor and critic networks are used in MADDPG to stabilize policy training. Critic network of a single UAV ($u_i$) takes the state and action of each individual UAV $\{s_{u_1}, a_{u_1}, s_{u_2}, a_{u_2}, ...\}$ at each training time-step $t$ into consideration to realize a state-action value function $Q^{\Pi}(s_{u_{i_t}}, a_{u_{i_t}})$ for UAV $u_i \forall\ i \in U$, given as:

$$Q^{\Pi}(s_{u_{i_t}}, a_{u_{i_t}}) = \mathbb{E}_{a_{u_{i_t}} \sim \Pi}\left[\sum_{k=0}^{\mathbb{H}} \gamma^k R_{t+k+1} | s_{u_{i_t}}, a_{u_{i_t}}\right] \quad (3)$$

where $0 \leq \gamma < 1$ is the discount factor and $\mathbb{H}$ denotes the max. time-step. As we have an estimate of the shortest path leading towards the critical location (as given by the guide path), the UAVs perform joint actions to sense the region over the guide path and the neighbouring regions to identify a serviceable path while avoiding obstructing objects. The guide path is a line connecting the start and critical location corresponding to the shortest Manhattan distance between them, without any knowledge of obstacles present in the environment. The obstacles are sensed by the UAVs in real-time. The reward function $R$ captures the objective of the system and incentivizes the UAVs based on the maintained overlap and their alignment with the guided trajectory:

$$R_t = \sum_{u_i \in U} T(u_i^c, \psi^c) + \sum_{f_i, f_j \in F} I(f_i, f_j) + \varphi \Lambda^c \quad (4)$$

$$\mathbb{T}_t(u_i, \Psi) = \begin{cases} \frac{1}{\underset{u_i \in U}{N}(u_i^c, \psi^c) + e} & if \quad \Lambda^c == 1 \\ 1 & otherwise \end{cases} \quad (5)$$

where $\mathbb{T}_t(.,.)$ denotes the trajectory alignment reward. $\psi^c$ is the set denoting cells/locations covered by the guide path and $u_i^c$ is the cell covered by the UAV $u_i$. $N(.,.)$ is the distance in terms of the number of cells between $\psi^c$ and $u_i^c$, where the diagonally adjacent cells are at the distance of $\sqrt{2}$ and vertically/horizontally adjacent cells are at a distance of 1. $e$ denotes a scalar to prevent ZeroDivisionError. $\Lambda^c$ denoted whether the cell $c$ is serviceable or not (refer to Equation 1).

The reward received by the multi-UAV system for maintaining FoV overlap is given by $\mathbb{I}_t(\upsilon_i, \upsilon_j)$. As seen in Figure 3, if the Euclidean distance $d(.,.)$ between the center of the FoVs of $i^{th}$ $and$ $j^{th}$ UAV is ideal i.e. equal to $D$ or upto allowable threshold $\mu D$, a positive reward is received by the UAVs. Whereas, a larger overlap (i.e. the distance between the center of the FoVs is less than $\mu D$) is observed or if there is a gap in FoVs' i.e. distance greater than $D$, a penalty $\mathbb{I}_t(\upsilon_i, \upsilon_j)$ is incurred by the multi-UAV system, as given in Equation 6.

$$\begin{cases} \omega & if\ \mu D \leq d(f_{u_i}, f_{u_j}) \leq D \\ -\omega \times (d(f_{u_i}, f_{u_j}) - D) & if\ d(f_{u_i}, f_{u_j}) > D \\ -\omega \times (\mu D - d(f_{u_i}, f_{u_j})) & if\ d(f_{u_i}, f_{u_j}) < \mu D \end{cases} \quad (6)$$

where $\omega$ denotes a positive scalar.

### *A. Real Time Path Planning Algorithm*

In the standard A* path planning algorithm, nodes are expanded in the order of their cost $q^c$ (starting from minimum cost), until the critical location is reached. A* takes a large amount of space to store all possible paths and a lot of time to find them. This can be computationally very expensive and sometimes infeasible in environments will large state spaces [8]. Hence, we present an improved A* approach, known as MEA* attaining the minimum number of expanded nodes/cells by performing a look-ahead into the search space and pruning the cells that are unserviceable (either being in a shallow water region or having obstacles).

In MEA* the underlying idea is that the sensed nodes (referred to as explored cells) that are in the shallow water region are not expanded further and the nodes that are vertically/horizontally adjacent to obstacles (i.e., an obstacle is present in the region lying ahead of that cell) incur a cost/penalty if expanded. Also, the proposed algorithm works in real-time where the start location changes as WBV moves on the current identified path and no replanning is done. As, the environment is dynamic and the water level of the cells changes, a location deemed to be shallow earlier might become serviceable at a later stage. Such a cell could be visited again if the estimated cost of reaching the critical location from that location plus the cost of returning to that location is less than the estimated cost of going forward from the current location. The cost function for the proposed MEA* is given as:

$$q^c_{MEA^*} = v^c + g^c + o^c \quad (7)$$

$$o^c = \begin{cases} \lambda & if \quad \Lambda^c == 0 \\ 0 & otherwise \end{cases} \quad (8)$$

$o^c$ is an additional cost ($\lambda$) for a node/cell $c$ that is surrounded by an obstacle. The presence of the obstacle lying ahead of the node (i.e. part of the guide path) and the shallow water regions/nodes (that are unserviceable) are captured by the autonomous UAVs deployed in front of WBV for guide path updation to identify a serviceable path. An explored node is different from an expanded node, where any sensed node/cell by a UAV is said to be explored and if shallow water or obstacle is observed at that location that node is not expanded further. The expanded nodes are those that are deemed to be serviceable.

## VI. Experimentation and Results

In this section, we discuss the considered experimental settings and analyze the performance of our proposed model MEA$^*$MADDPG over a 2D grid. For experimentation, a small region of Chennai city's geographic details is encoded using the Mapbox Streets tool to simulate the environment. Further, a 2D water layer is overlayed in reference to the FloodMap tool [20] to mark the regions with shallow flood water (usually the high-elevation regions) and the regions with a large volume of accumulated flood water. During experimentation the shallow water cells are processed as obstacles i.e., they cannot be a part of a serviceable path. We compared our proposed approach MEA$^*$MADDPG with 3 other prevalent techniques from the literature, namely, RRT [7], RRT* [5], [14], A* [5] and real-time path planning algorithm ERRT [7]. Active Pre-training [21] is applied to MEA$^*$MADDPG for 50 episodes to align the UAVs with the guide path. We observe average run-time, number of expanded nodes/cells and average serviceable path length to test our proposed model. The results are observed over 100 episodes with varying initial positions of start and critical location. The maximum run-time of a single episode is set to 50000 time steps.

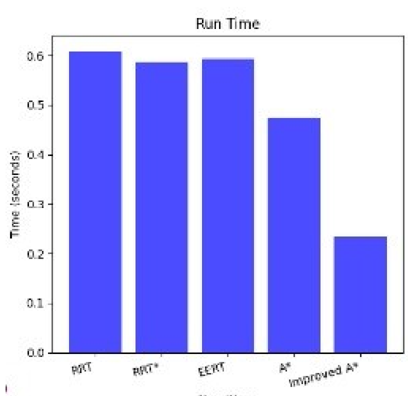


Fig. 4. Performance Comparison over Average Run Time.

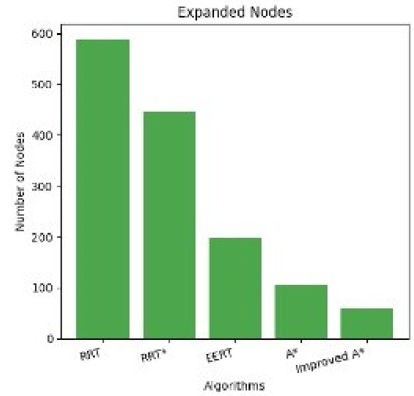


Fig. 5. Performance Comparison over the Average Number of Expanded Cells.

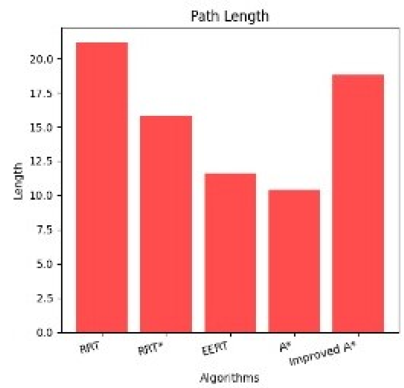


Fig. 6. Performance Comparison over Average Serviceable Path Length

## VII. Conclusion

## References

[1] A. Garg and S. S. Jha, "Directed explorations during flood disasters using multi-uav system," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*, 2022, pp. 2154–2161.

[2] C. Yan, X. Xiang, and C. Wang, "Towards real-time path planning through deep reinforcement learning for a uav in dynamic environments," *Journal of Intelligent and Robotic Systems*, vol. 98, no. 2, p. 297–309, 2019.

[3] A. Zulu and S. John, "A review of control algorithms for autonomous quadrotors," *Open Journal of Applied Sciences*, vol. 04, pp. 547–556, 01 2014.

[4] R. S. Sutton and A. Barto, *Reinforcement learning: an introduction*. The MIT Press, 2018.

[5] J. a. Braun, T. Brito, J. Lima, P. Costa, P. Costa, and A. Nakano, "A comparison of a* and rrt* algorithms with dynamic and real time constraint scenarios for mobile robots," in *Proceedings of the 9th International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, 2019, p. 398–405.

[6] K.-l. Wang, S.-w. Dang, F.-j. He, and P.-z. Cheng, "A path planning method for indoor robots based on partial a global a-star algorithm," *Proceedings of the 5th International Conference on Frontiers of Manufacturing Science and Measuring Technology (FMSMT)*, 2017.

[7] W. Yang, G. Wang, and Y. Shen, "Perception-aware path finding and following of snake robot in unknown environment," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5925–5930.

[8] A. Felner, M. Goldenberg, G. Sharon, R. Stern, T. Beja, N. Sturtevant, J. Schaeffer, and R. C. Holte, “Partial-expansion a* with selective node generation,” in *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*. AAAI Press, 2012, p. 471–477.
[9] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, “Multi-agent actor-critic for mixed cooperative-competitive environments,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 6382–6393.
[10] Z. A. Ali and H. Zhangang, “Multi-unmanned aerial vehicle swarm formation control using hybrid strategy,” *Transactions of the Institute of Measurement and Control*, vol. 43, no. 12, p. 2689–2701, 2021.
[11] P. Kaushik, A. Garg, and S. S. Jha, “On learning multi-uav policy for multi-object tracking and formation control,” in *2021 IEEE 18th India Council International Conference (INDICON)*, 2021, pp. 1–6.
[12] H. Liu, F. Peng, H. Modares, and B. Kiumarsi, “Heterogeneous formation control of multiple rotorcrafts with unknown dynamics by reinforcement learning,” *Information Sciences*, vol. 558, p. 194–207, 2021.
[13] W. Zhao, H. Liu, Y. Wan, and Z. Lin, “Data-driven formation control for multiple heterogeneous vehicles in air–ground coordination,” *IEEE Transactions on Control of Network Systems*, vol. 9, no. 4, pp. 1851–1862, 2022.
[14] J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, and M. S. Muhammad, “Rrt*-smart: A rapid convergence implementation of rrt*,” *International Journal of Advanced Robotic Systems*, vol. 10, no. 7, p. 299, 2013.
[15] G. Tian, L. Zhang, X. Bai, and B. Wang, “Real-time dynamic track planning of multi-uav formation based on improved artificial bee colony algorithm,” *37th Chinese Control Conference (CCC)*, 2018.
[16] B. Zhang, G. Li, Q. Zheng, X. Bai, Y. Ding, and A. Khan, “Path planning for wheeled mobile robot in partially known uneven terrain,” *Sensors*, vol. 22, no. 14, p. 5217, 2022.
[17] “Mapbox streets,” 2023, accessed: 2023-1-1. [Online]. Available: https://docs.mapbox.com/data/tilesets/reference/mapbox-streets-v8/
[18] “Mapbox bathymetry,” 2023, accessed: 2023-1-1. [Online]. Available: https://docs.mapbox.com/android/legacy/maps/examples/bathymetry/
[19] Geo-matching, “Astralite edgetm lidar,” 2023, accessed: 2023-1-1. [Online]. Available: https://geo-matching.com/airborne-laser-scanning/astralite-edge-lidar
[20] (2023) Flood map: Water level elevation map. Accessed: 2023-1-1. [Online]. Available: https://www.floodmap.net/
[21] H. Liu and P. Abbeel, “Behavior from the void: Unsupervised active pre-training,” in *Advances in Neural Information Processing Systems*, 2021.