ADVANCEMENTS IN UNMANNED AERIAL VEHICLE PATH PLANNING: DEEP REINFORCEMENT LEARNING APPROACH for ENHANCED NAVIGATION

GUO JINGRUI

MPhil

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

Advancements in Unmanned Aerial Vehicle Path Planning: Deep Reinforcement Learning Approach for Enhanced Navigation

GUO Jingrui

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Philosophy

August 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

GUO Jingrui (Name of student)

Abstract

The main objective of the thesis is to demonstrate a comprehensive study of the autonomous navigation of Unmanned Aerial Vehicles (UAVs) through intricate environments characterized by narrow gaps. Employing advanced Deep Reinforcement Learning (DRL) methodologies, this research introduces two innovative algorithms optimized for real-time UAV path planning. The first algorithm enhances the standard Deep Q-Network (DQN) by integrating a series of improvements that increase its adaptability and decision-making capabilities in complex environments. This enhancement involves the incorporation of a more efficient reward structure and a refined state-action space representation, allowing the UAV to autonomously generate optimized navigation paths. The enhanced DQN framework facilitates rapid adaptation to environmental variations, improving both the learning speed and robustness of the UAV's path planning. This results in more effective navigation, especially in environments with narrow gaps and dynamic obstacles. A key feature of this enhanced algorithm is its ability to map action commands directly from sensor data, thereby improving the UAV's real-time decision-making. Furthermore, by implementing a direction reward function, the algorithm incentivizes the UAV to optimize its trajectory towards target goals while penalizing deviations from the desired path. This approach strengthens the UAV's gen-

eralization ability, allowing it to perform effectively across a range of diverse operational scenarios.

In parallel, this thesis addresses the complex challenge of autonomous navigation for UAVs in real environments. The study employs a sophisticated DRL approach using the Soft Actor-Critic (SAC) algorithm, which is specifically optimized for UAV path planning within a continuous action space. This method utilizes environmental image data to refine the accuracy of flight maneuvers and enhance obstacle avoidance capabilities. The efficacy of our approach has been substantiated through comprehensive simulations in Gazebo and empirical field tests, which demonstrate the algorithm's capability to enable UAVs to adeptly navigate through obstacles using depth maps. Furthermore, the study assesses the robustness of the SAC algorithm by juxtaposing it with conventional DRL methods, highlighting its superior performance in practical applications. This research makes a significant contribution to the advancement of UAV technology, particularly in autonomous motion planning, by incorporating advanced machine learning techniques. The findings and methodologies are accessible via the provided video link: https://www.youtube.com/watch?v=Nd_aMzejNXY.

In general, this research advances UAV technology by integrating cutting-edge machine learning techniques into autonomous motion planning. It enhances the adaptability and efficacy of UAV navigation in narrow-gap environments and contributes significantly to the field by establishing benchmarks for evaluating various DRL algorithms in complex terrains.

Keywords: Unmanned Aerial Vehicles, Deep Q-Network, Soft Actor-Critic.

Publications arising from the thesis

J. Guo, C. Huang, and H. Huang, "A Deep Q-Network-Based Algorithm for Obstacle Avoidance and Target Tracking for Drones," in *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 4530-4535, 2023.

J. Guo, G. Zhou, H. Huang, and C. Huang, "Advancements in UAV Path Planning: A Deep Reinforcement Learning Approach with Soft Actor-Critic for Enhanced Navigation," in *Unmanned Systems*, 2024. DOI: https://doi.org/10.1142/S2301385025500669.

Acknowledgements

I sincerely present my immense appreciation to Dr. HUANG Chao, my supervisor who provide precious instruction, unwavering support, and proficient suggestion to me during my whole MPhil study period. Her profound knowledge and meticulous attention to detail have significantly shaped and refined my research, providing a strong foundation for my thesis.

I wish to express my equal gratitude to Dr. HUANG Hailong, my co-supervisor who support me with his meaningful assistance and expert insights over the past two years. His guidance were pivotal in addressing the intricate challenges of my study, offering both technical support and academic encouragement that were crucial to my studies.

Their combined expertise not only directed me towards rigorous academic inquiry but also taught me the intricacies of systematic research within the field of Unmanned Aerial Vehicles. The lessons learned under their tutelage extend beyond academic knowledge, instilling in me a profound appreciation for thorough and ethical scientific practices.

I am deeply thankful for their patience, mentorship, and unwavering commitment, which have been vital to my academic and personal growth. This thesis would not have reached its fruition without their exemplary guidance and persis-

tent encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Significantly advanced in supporting a wide range of applications can be found in Unmanned Aerial Vehicles (UAVs) across diverse industries, with demonstration of UAVs'pivotal role in boosting operational efficiency and overcoming conventional limitations [48]. These applications include search and rescue missions, structural inspections, geospatial mapping, parcel delivery, and agricultural innovations, where UAVs demonstrate critical advantages in precision, accessibility, and cost-effectiveness [33]. Notably, UAV autonomous system excel in navigating through constricted spaces and narrow apertures [16]. This ability is underpinned by the UAVs' integration of semi-autonomous and fully autonomous operations facilitated by advanced sensory technologies [41].

Moreover, the advent of state-of-the-art UAVs equipped with enhanced GPS navigation, computer vision, and obstacle avoidance systems has revolutionized traditional methods of aerial observation and data collection [33]. These innovations not only address the challenges associated with manned aircraft and satellites, such as high operational costs and limited accessibility but also improve the

spatial resolution and timeliness of data acquisition [42]. By integrating these cutting-edge technologies, UAVs provide a dynamic, efficient, and safer solution for real-time aerial tasks across diverse environments, establishing themselves as pivotal assets in modern technological landscapes [21]. Through continuous advancements in UAV technology, these vehicles increasingly support complex operations, showcasing their versatility and evolving role in critical and emerging sectors [61].

1.1 Study Background and Incentive

The accuracy of path planning is a significant aspect of UAV operations, particularly in challenging terrains such as urban or industrial settings [39]. This process is vital when navigating through narrow openings and intricate landscapes, demanding high accuracy in maneuvering within spatially constrained areas [62]. Advanced object detection frameworks, underpinned by artificial intelligence, play a pivotal role here. Cited extensively in the literature, these frameworks assist UAVs in identifying viable entry points or gaps by utilizing onboard cameras [41]. The critical nature of detecting these gaps allows for the calculation of the most effective angles and trajectories for safe passage, thereby reducing collision risks and enhancing operational efficiency [11].

Developing algorithms that optimize route efficiency and safety, ensuring minimal travel distance while mitigating potential hazards is the main objective of UAV path planning [74]. This aspect is particularly crucial in operations requiring high precision and swift responses, such as in search, rescue, or detailed inspections [36]. The integration of Deep Reinforcement Learning (DRL) [65] has be-

come a cornerstone for enhancing autonomous navigation capabilities in complex, variable environments. Traditional learning paradigms such as supervised and unsupervised learning which rely on pre-labeled data or are suited for data clustering. DRL enables UAVs to autonomously develop optimal navigation strategies [49]. This capacity for self-learning is crucial given the extensive computational resources and significant training durations required for effective implementation [32]. The development of these advanced algorithms emphasize the vital significance of path planning in UAV technology, marking it as an indispensable element in complex operational scenarios [37].

This thesis is motivated by the pursuit of advancing autonomous navigation capabilities for UAVs in environments, a critical challenge in the field of aerial robotics [42]. The core of this research centers on the development and implementation of two sophisticated DRL strategies designed to enhance navigational ability of UAVs.

The first algorithm enhances the standard Deep Q-Network (DQN) [37] by integrating a series of improvements that increase its adaptability and decision-making capabilities in complex environments. This innovative DRL framework is tailored to rapidly adapt to environmental fluctuations and is specifically engineered to accumulate and analyze state and path data across a variety of scenarios. By incorporating a direction-based reward-penalty function into the UAV's reward system, this algorithm substantially enhances the UAV's capacity to perceive its environment and broadens its generalization capabilities. Consequently, this leads to a marked improvement in overall performance, particularly in navigating complex terrains. Simultaneously, this thesis investigates the effectiveness of the Soft Actor-Critic (SAC) algorithm [83], a cutting-edge DRL method opti-

mized for continuous action spaces within UAV path planning. This approach is evaluated against traditional DRL methods to assess its robustness and effectiveness in real-world applications. The SAC algorithm is particularly noteworthy for its ability to maintain optimal policy estimation while navigating through environments that undergo continuous changes, thereby supporting more precise and reliable UAV operations.

Collectively, these research efforts aim to push the boundaries of UAV technology by employing advanced machine learning techniques to refine autonomous path planning. This contribution is crucial for advancing the practical disposition of UAVs across various applications, including surveillance, delivery, and emergency response scenarios. Through this thesis, we seek to establish a benchmark for UAV autonomy that combines state-of-the-art computational intelligence with practical, real-world applicability.

1.1.1 NEWDQN Algorithm

To improve the low success rate and limited environmental generalization capabilities of existing algorithms in dynamic environments for autonomous UAV obstacle avoidance and target tracking, this research proposes an enhanced deep reinforcement learning algorithm named NEWDQN. Firstly, the detection strategy within the DQN algorithm is improved by incorporating an innovative approach called Optimistic Bootstrapping Exploration (OBE), which enables the UAV to explore the environment more effectively. Secondly, a multi-experience pool mechanism is introduced to categorize the collected experience data into successful and unsuccessful experiences. Compared to a single experience pool, this mechanism

improves the quality of sampled data and decrease the possibility of the algorithm from getting stuck in local optima. Additionally, a direction-based reward-penalty function is integrated into the reward system to guide the algorithm into quicker convergence. Moreover, to enhance the UAV's adaptability to the environment, its perception capabilities are augmented, enabling better environmental understanding. Finally, results of simulation verify the efficacy of the proposed approach.

1.1.2 SAC Algorithm

This study employs the Soft Actor-Critic (SAC) algorithm within a consecutive action environment to enhance UAV barrier evasion features. The UAVs are cultivated, by utilizing depth maps and applying SAC with deep learning, to pass through comprehensive emulated environments containing multiple obstacles. This method not only increases more accurate and smooth decisions in action but also illustrates outstanding consistency and reward outcomes during training compared to previous DRL algorithms that cope with LiDAR or location data inputs straightly. Experiments give evidence that employing a delayed update learning method yields better results in UAV gap navigation tasks. Classical DRL approaches tend to proceed direct LiDAR or location data inputs for UAV obstacle navigation tasks. By comparison, the SAC model in this study shows faster convergence and achieves outstanding rewards. This research develops the feasibility for invasive flight maneuvers, such as navigating through narrow slit, which shows significant improvement in UAV maneuverability in scenarios like search-and-rescue operations. Previously, solutions of aggressive flight planning, subjected to the UAV's underactuated dynamics and the comprehensiveness of searching possible paths, depend

on enhancing manually defined loss functions within a constrained framework. Nevertheless, these traditional methodes always predigest the issue via advanced premises, restricting the room for solution.

The emulation circumstance utilized in this research is carefully constructed using Gazebo and the Python PyTorch machine learning mechanism, creating a comprehensive platform for educating a UAV to function as the learning agent. This inventive setting enables the UAV to analyze environmental depth images and execute complex flight and obstacle evasion maneuvers. The conversancy presented by the cultivated UAVs in obstacle avoidance is significant, achieving exceptional success rates that highlight their flexibility and operational effectiveness. Under a governing training context, a remarkable mean successful rate with over 90% has been found in UAVs. Additionally, the experiments achieve commendable successful rates of above 80% and 70% in scenarios where obstacles are either repositioned or entirely redesigned.

The UAVs' ability to achieve a 68% success rate in real-world tests is not merely a performance metric, but an important benchmark indicating the model's robustness and practical viability. This success rate represents a significant accomplishment in navigating through dynamic and unpredictable environments, demonstrating the UAV's capability to adapt and operate effectively despite real-world challenges. Achieving this threshold is not just an isolated goal, but a critical step toward further improving the model's performance under varying operational conditions. The adaptation of policies learned through simulations to real-world UAV applications presents a substantial challenge, primarily due to the stringent error tolerance required for Sim2Real transfer. Sim2Real transfer refers to the process of transferring a model trained in a simulated environment to real-world applications,

a task fraught with difficulties due to the discrepancies between the controlled conditions of simulations and the inherent unpredictability of real environments. In particular, this process often encounters issues such as sensor inaccuracies, environmental variability, and the failure to replicate real-world complexities in simulations. These challenges are compounded by the difficulty of obtaining real flight data, which is typically required to fine-tune models for real-world deployment.

To address these concerns, we propose an innovative technique designed to facilitate the effective transfer of DRL models to actual UAV operations without the need for real flight data. This approach minimizes the risks and logistical complications typically associated with collecting real-world flight data, thereby enhancing the practicality, scalability, and reliability of UAV navigation and obstacle avoidance strategies. By circumventing the need for extensive real-world data collection, our method makes significant strides in improving the transferability of learned policies, ensuring that UAV can more reliably perform autonomous tasks in dynamic, real-world environments.

This study makes significant contributions to the field of UAV autonomous navigation, particularly in the context of obstacle avoidance and real-world application of DRL. The key contributions of this research are as follows:

- Advancing UAV Obstacle Avoidance with the SAC Algorithm: This research introduces the application of the SAC algorithm to UAV obstacle avoidance, a crucial aspect of autonomous navigation in dynamic and complex environments. By utilizing SAC, the study enhances the UAV's decision-making capabilities, allowing for more efficient and reliable path planning in the presence of obstacles.

- Demonstrating Real-World Applicability of the SAC Algorithm: Through a combination of simulated training and real-world testing, the study highlights the efficacy of the SAC algorithm in UAV obstacle avoidance. The approach achieved significant success rates in real-world obstacle avoidance scenarios, demonstrating the algorithm's robustness and practical utility in operational environments.

- Reducing Dependence on Pre-labeled Data: The research leverages environmental image data to train the UAV's navigation model, significantly reducing reliance on large, pre-labeled datasets. This approach ensures that the UAV can effectively learn to navigate through diverse environments by processing visual inputs in real-time, making the learning process more adaptable and scalable.

- Utilizing Continuous Action Spaces for More Fluid UAV Movements: A key innovation in this research is the application of DRL within a continuous action space, as opposed to traditional discrete action models. This methodological shift facilitates smoother, more adaptable UAV movements, enabling more refined control and more precise obstacle avoidance in dynamic environments. The continuous action space allows the UAV to make decisions that involve a range of movement possibilities, rather than being restricted to a limited set of pre-defined actions.

1.2 Thesis Structure

The structure of this thesis is organized as follows. Chapter II presents a literature review on DRL. Chapter III details the construction of the methodology model. Chapter IV discusses the experiment results and analysis of the proposed approach. Finally, Chapter V provides a conclusion to this thesis.

Chapter 2

Literature Review

The introduction of Unmanned Aerial Vehicles (UAVs) has markedly transformed various industries, from agriculture to search and rescue operations, by introducing innovative solutions that significantly enhance both efficiency and safety [5]. As UAV technologies continue to advance, the development of sophisticated autonomous navigation capabilities becomes essential for maximizing their utility in complex and dynamic environments. In these environments, UAVs must contend with transient and unpredictable obstacles, such as moving vehicles, animals, or sudden environmental changes, which necessitate advanced situational awareness and adaptive path planning capabilities [60]. To optimize UAV efficiency and safety in such conditions, robust path planning algorithms are essential. This begins with the UAV's ability to perceive and interpret its surroundings in real time using integrated sensory systems, which typically include visual and depth sensors. The data captured by these sensors must then be processed using advanced artificial intelligence techniques to accurately identify and classify dynamic obstacles.

Considering the inherently limited detection capabilities of UAV sensors, it becomes vitally important for these aerial systems to demonstrate a high degree of precision when navigating through intricate terrains, ensuring a seamless transition through narrow openings. As a result, the discipline of path planning has risen to prominence as an indispensable element for UAVs operating in such challenging environments. The importance of this discipline is further underscored in terrains marked by complex infrastructures, exemplified by urban landscapes or industrial settings, compelling UAVs to operate within spatially restricted areas [40].

Prior to initiating the complex task of path planning, it is imperative for the UAV to precisely identify the aperture or the designated target area [80]. In the quest to achieve this, object detection frameworks, grounded in the principles of artificial intelligence, have been subjected to rigorous academic scrutiny. These advanced frameworks assist UAVs in pinpointing gaps or potential entry points through the use of onboard cameras that continuously monitor their environment [42]. Upon successful detection of a gap, its precise spatial coordinates can be determined based on the parameters of the bounding box. Armed with this information, the UAV then calculates the most appropriate angle and trajectory to safely navigate through the identified opening, thereby minimizing the risk of potential collisions.

Within the realm of UAV operations, the primary objective of path planning is the development of sophisticated algorithms that enable the UAV to chart the most advantageous trajectory from its current position to the designated aperture [54]. It is of utmost importance for the UAV to adeptly navigate around any potential obstacles, ensuring a safe and unobstructed journey. The responsibility of the path planning algorithm is to ascertain the most efficient trajectory, prioritizing the

minimization of both travel distance and associated risks [38]. As such, the chosen algorithm for path planning must be capable of determining the optimal trajectory in the shortest possible time frame, all the while adhering to stringent safety standards. The need for such speed and accuracy becomes paramount in missions that necessitate precision and promptness, such as those related to search, rescue, or detailed inspections [5].

The fundamental aim of path planning in this scenario is to devise algorithms that enable UAVs to autonomously navigate from their starting point to a specified target location in an efficient and secure manner [10]. This involves dynamically adjusting their flight path in response to changes within the environment, thereby minimizing the risk of collisions and optimizing travel time—an imperative in scenarios where timing may be critical, such as in search and rescue operations [82].

As delineated in Table 2.1, a wide array of methodologies, from non-learning-based to learning-based strategies, have been developed within the realm of UAV path planning.

2.1 Basic concepts and definitions

2.1.1 Classical methods

Over recent decades, the broadening scope of UAV applications—ranging from surveillance to cargo delivery—has underscored the need for more advanced path planning techniques [9]. The progression of communication technologies, from first generation (1G) to fifth generation (5G), has significantly enhanced the data

Table 2.1: Summary of UAV path planning methods

Approach	Strengths	Weakness
Classical	1. Good results for path optimization in static environments with simple obstacles. 2. Short run times and low computational resource requirements.	1. Optimal results cannot be guaranteed due to constraints. 2. Complete environments where tasks need to be performed. 3. Poor performance in complex and dynamic environments.
Simple-heuristics	1. Good path optimization results in static environments with constraints on individual UAVs. 2. Moderate response time and moderate computation resource requirements.	1. No guarantee that the result is optimal because of constraints. 2. Poor results in multi-objective path planning tasks and tend to fall into local optima. 3. Performs poorly in complex and dynamic environments.
Meta-heuristics	1. Good path optimization results in complex dynamic environments with multiple UAVs. 2. Reasonable execution time and ease of implementation.	1. No guarantee that the result is optimal because of constraints. 2. Long computation time and high computation cost. 3. No theoretical convergence.
Machine learning	1. Optimal solution. 2. Suitable for complex and dynamic environments with sudden changes.	1. Requires large training data for the environment. 2. Long computation time and high computation cost.

exchange capabilities among UAVs and other connected devices, facilitating more intricate and dynamic operational strategies [8].

The domain of UAV path planning has experienced considerable growth, marked by the introduction of numerous classical methodologies designed to navigate the complex challenges inherent in this field [27]. Prominent among these are the Rapidly-exploring Random Tree (RRT) [63], Visibility Graph (VG) algorithm [47], Voronoi Diagram (VD) [29], Artificial Potential Field (APF) [50], Probabilistic Road Map (PRM) algorithm [26], and the Dijkstra algorithm [73].

These classical methodologies have been lauded for their rapid solution generation and exceptional path optimization capabilities, especially suited to static environments with simple obstacles [12]. However, they require the acquisition of comprehensive and accurate environmental data to develop detailed graphical or model-based representations [79]. This necessity poses a notable challenge and highlights a primary limitation of these methods. Despite their proven efficacy and straightforward implementation in certain scenarios, the practical application of these classical methodologies often hinges on the ability to obtain a detailed and precise understanding of the UAV's operational context.

Moreover, the challenge of UAV path planning extends beyond merely charting a trajectory from an origin to a destination. It involves ensuring that the selected path is free from collisions and aligns with the dynamic and often unpredictable nature of the operational environments [27]. This task also demands adherence to the UAV's physical and kinematic constraints, including considerations for energy consumption and maneuverability, ensuring optimal and safe operations within the designated aerial space [37]. This comprehensive approach to UAV path planning not only boosts operational efficiency but also facilitates the

wider integration of UAVs into increasingly sophisticated application domains.

2.1.2 Heuristic methods

Several heuristic-based algorithms have been developed to enhance UAV path planning in less predictable contexts [2]. The greedy heuristic (GH) algorithm, have demonstrated superiority over genetic algorithms (GA) and multiple population genetic algorithms (MPGA) in terms of execution speed and path optimization [2]. These simple heuristic algorithms, requiring minimal environmental data, perform well in static and simpler dynamic environments.

The adoption of meta-heuristic algorithms has proven to be a formidable approach for addressing the challenges of UAV path planning, especially in managing the complexities of dynamic and multifaceted environments [23]. Techniques such as the improved Genetic Algorithm (GA), neighborhood-based GA, and the Multi-Population Chaotic Grey Wolf Optimization (MP-CGWO) algorithm have demonstrated superior performance in optimizing path length, cost, and convergence speed in multi-UAV operations [2]. These algorithms are crafted to deliver high-quality solutions that effectively adapt to changes in the environment and constraints in UAV operations. Despite their benefits, the deployment of meta-heuristic algorithms in UAV path planning is not widespread, primarily due to their propensity for local optima and the considerable computational demands of their iterative processes.

2.1.3 Machine learning methods

In parallel, the application of machine learning methods, especially Deep Learning (DL). UAV path planning is increasingly recognized as vital, motivated by the escalating demand for these aerial systems to navigate autonomously through evolving and unpredictable environments [30]. Traditional RL methods have proven effective for scenarios with static or nonexistent obstacles [22]. However, these approaches are often inadequate in more complex settings [30].

Deep learning, a transformative force within machine learning characterized by the development of artificial neural networks, gained substantial momentum in the early 2000s, facilitated by advancements in graphical processing units (GPUs) [44]. In the domain of UAV technology, deep learning algorithms are integral to enabling UAVs to perform advanced autonomous tasks, particularly in the areas of obstacle detection, path planning, and precise positioning [81]. These algorithms leverage vast amounts of data to model complex, nonlinear relationships within the environment, thereby enhancing the UAV's ability to perceive its surroundings and make informed decisions in real-time. Deep learning strategies including neural networks for image interpretation [30], sequential information [25] to support decision-making, empower UAVs to automatically recognize obstacles in ever-changing settings, map out efficient flight trajectories, and fine-tune their positioning with exceptional precision. This level of autonomy eliminates the need for constant human intervention, making UAVs more effective in executing tasks such as navigation through cluttered or unfamiliar environments, performing search-and-rescue missions, or conducting precision agriculture operations. The ability of deep learning m-

odels to process sensor data enhances the UAV's situational awareness, allowing for rapid and reliable responses to unforeseen obstacles changes.

The application of supervised learning in UAV navigation involves training models on a well-labeled dataset, which provides high accuracy for navigation tasks in environments similar to the training data [81]. The efficacy of supervised learning largely depends on the availability of extensive, accurately labeled datasets, which are often expensive and labor-intensive to compile. Moreover, supervised models typically face challenges in generalizing to novel, unseen environments that substantially deviate from those represented in the training data.

Unsupervised learning, by contrast, operates without labeled outputs, allowing it to identify hidden patterns and intrinsic structures within data [84]. This makes it particularly valuable in situations where labeled data is scarce or incomplete. While unsupervised learning typically yields less precise predictions than supervised methods [45], it is crucial for exploring and understanding complex datasets and can be instrumental in enhancing the performance of supervised and reinforcement learning algorithms.

In the domain of UAV path planning, the combined use of supervised and unsupervised learning techniques with Deep Reinforcement Learning (DRL) is increasingly recognized as an effective strategy to address the variabilities of dynamic environments [43].

In summary, while deep learning-based algorithms hold transformative potential for enhancing UAV autonomous navigation, the deployment of these technologies must be carefully tailored to the specific demands of each application. The selection between supervised and unsupervised learning methodologies should be informed by factors such as the availability of data, the complexity of the naviga-

tional tasks, and the precision required in navigation outcomes. Balancing these considerations is essential for effectively harnessing the strengths of each learning paradigm to optimize UAV operations in environments characterized by dynamic and unpredictable obstacles.

Recent developments have enhanced the synergy of deep learning with reinforcement learning, offering solutions to some of the inherent challenges of conventional reinforcement learning approaches in dynamic scenarios [13]. These advancements facilitate more robust and adaptive navigation strategies, capable of operating effectively across a broader range of environmental conditions. For instance, Tai and Liu developed a DRL strategy utilizing CNNs, though its application was limited to static scenarios [56]. Fang et al. extended these approaches to environments with low dynamics or sparse obstacles [17], yet these too fall short in highly dynamic situations where obstacles such as moving vehicles, pedestrians, or animals present continuous and unpredictable challenges. Such environments are typical in urban and semi-urban areas where UAVs must navigate at low altitudes amidst a plethora of moving elements.

In response to the evolving needs of UAV navigation, path planning, and obstacle avoidance, various Reinforcement Learning (RL) methods have been refined and adapted. Q-learning, a cornerstone value-based RL approach, updates a Q-table to determine the optimal policy and is particularly effective in scenarios that do not necessitate a model of the environment [65]. However, its application to UAV control is limited due to difficulties in scaling within continuous action spaces, an essential feature for precise UAV maneuvers.

Deep Q-Networks (DQN) extend Q-learning by integrating deep neural networks, thus enabling the approximation of Q-values to handle larger state spaces.

One of the primary issues is the instability and potential divergence during training, as the Q-values can be updated inappropriately, leading to compounding errors without stabilization mechanisms like experience replay and target networks. Additionally, DQNs often suffer from overestimation of Q-values, which can result in suboptimal policies and less effective decision-making, especially in high-variance environments. The algorithm is also sample-inefficient, requiring a large number of interactions with the environment to converge, which can be problematic in real-world scenarios where data collection is costly or impractical. Furthermore, DQN is inherently designed for discrete action spaces, making it unsuitable for continuous control problems such as UAV navigation, where continuous action spaces are necessary. The exploration vs. exploitation dilemma also remains a challenge, as DQN may struggle to balance exploration of new actions with exploitation of known high-reward actions, particularly in complex environments. In addition, DQNs can suffer from high memory usage due to the experience replay mechanism, which becomes impractical in environments with complex states and actions. The algorithm also relies heavily on well-defined reward functions, and poor reward shaping can lead to unintended behaviors, making the reward design process crucial. Long training times are another drawback, as DQNs require substantial time to converge to a good policy, which is often incompatible with real-time applications. Finally, DQNs are black-box models, lacking interpretability and transparency in decision-making, which can be problematic in safety-critical applications where understanding the reasoning behind decisions is essential. These disadvantages highlight the challenges of applying DQN in dynamic, high-dimensional environments, necessitating modifications or the use of alternative algorithms to address these limitations effectively. Moreover, the

substantial computational demands for training can be prohibitive for UAVs with limited onboard processing capabilities [20].

To accommodate continuous action decisions, the Deep Deterministic Policy Gradient (DDPG) method utilizes a model-free [18], actor-critic framework that combines policy gradients with Q-learning. This adaptation allows for smoother integration of continuous action spaces in the training process. Although DDPG is adept at complex control tasks and suited to the nuanced requirements of UAV operations, it remains highly sensitive to hyperparameter settings, is prone to local optima, and tends to overestimate action values, which can adversely affect its performance in dynamically evolving environments.

Further refining the capabilities of RL, the Distributed Proximal Policy Optimization (DPPO) algorithm [3] extends Proximal Policy Optimization (PPO) to a distributed architecture, enhancing the balance between exploration and exploitation through trust-region methods. However, DPPO requires a network of learners for effective policy updates, introducing additional complexity and substantial computational infrastructure demands, which may not be practical for all UAV systems.

These innovations underscore the necessity for continued research to develop more robust, efficient, and adaptable DRL algorithms. Such frameworks are essential for effective operation in the dynamic and often unpredictable environments typical of urban and semi-urban UAV applications. This ongoing research is vital to advancing the capabilities of UAVs in complex operational contexts. Such advancements are crucial for enabling UAVs to perform autonomous operations safely and efficiently amidst a complex array of moving obstacles.

2.1.4 Hybrid methods

The burgeoning integration of advanced machine learning techniques with classical path planning methods represents a substantial evolution in the development of navigation systems for UAVs, particularly as they are increasingly deployed in dynamic and unfamiliar environments [7]. This research proposal aims to tackle the significant challenges presented by environments through an innovative integration of DRL with DL methodologies, thereby enhancing both the adaptability and computational efficiency of UAV path planning [52].

Historically, methods like the Grey Wolf Optimization (GWO) blended with reinforcement learning have shown promising improvements in adaptability for UAV path planning [35]. Similarly, the combination of APF methods with the RRT strategy has proven effective in handling complex static obstacles [35]. However, these methods typically require extensive environmental data and struggle in highly dynamic settings where obstacles and environmental conditions change unpredictably.

The incorporation of DRL with established optimization algorithms such as the Interfered Fluid Dynamical System (IFDS) [57], APF, and Model Predictive Control (MPC) marks a considerable progression in the domain of path planning and maneuver control for autonomous systems. When integrated with IFDS, DDPG can optimize the control policy by learning the optimal actions in a simulation that models fluid dynamics [33]. The IFDS provides a smooth, global trajectory that guides the DDPG's exploration, enhancing the algorithm's efficiency in environments mimicking fluid flows. This synergy enables the system to adapt to moving obstacles.

APF is a renowned method in robotics for obstacle avoidance, characterized by obstacles generating repulsive forces and goals generating attractive forces that influence the movement of the navigating agent [75]. By combining PPO with the APF method, the system can robustly handle real-time changes in the environment [76]. APF provides immediate responses to obstacle proximity through repulsive forces, while the PPO algorithm continuously adjusts the navigation strategy to minimize potential collisions and optimize the trajectory towards the target. This integration is particularly beneficial in crowded environments where dynamic obstacle avoidance is critical, such as in urban UAV navigation or mobile robotics in industrial settings [68].

MPC is a simple control strategy that employs an optimization algorithm to determine the control actions based on the prediction of future states of the system over a defined horizon [19]. This approach allows for detailed and anticipatory control decisions that are essential for the dynamic management of UAVs in complex environments. When combined with MPC, the SAC algorithm can utilize the predictive model of MPC to foresee future states and optimize actions over a receding horizon.

2.2 Prior studies (relevant research)

Each RL algorithm offers unique benefits and challenges in UAV applications. Tailoring them to specific UAV requirements often involves hybrid approaches and careful tuning of parameters to overcome inherent limitations. Recent research demonstrates that using a deep reinforcement learning framework like D3QN PER [15], which incorporates a prioritized experience replay mechanism, can signifi-

cantly improve the planning effectiveness for UAVs in dynamic scenes. This approach outperforms classical methods such as A* [34], RRT, and DQN, mainly due to better handling of real-time changes in the environment [59]. These benefits are still challenged by highly dynamic environments where global state information is incomplete [14]. This limitation can lead to suboptimal path planning and poor convergence in learning-based algorithms. Enhancements have also been made in multi-UAV autonomous path planning. By adopting DRL, UAVs can now perform better in reconnaissance missions even with incomplete information. These benefits need a proper reward structure that can accurately reflect the contribution of each UAV to the collective outcome, which is critical. Studies have pointed out that most existing approaches do not adequately solve the credit assignment problem, which is essential for fostering cooperative behavior [67]. A Q-learning model that integrates environmental feedback in real-time, enhancing UAV navigation in urban landscapes and reducing path deviations [28]. These approaches are developed to deal with consecutive movement and statements. However, they confront issues in applying UAV path planning in the practical world due to the greater complexity of these scenarios compared to those usually studied [1]. These algorithms require a resource-heavy training process, involving important computational capability and vast image datasets to effectively train accurate navigation to the models [6].

In an endeavor to reduce reliance on traditional methods, a subsequent study explored the use of RL to navigate through gaps, signifying a shift away from purely optimal control-based approaches. This innovative method employed a neural network to replicate trajectories that were initially computed using an optimal control solver. The trajectories generated through this approach demonstrated

a greater variety of patterns as opposed to the parabolic curves produced by earlier methods. However, despite these advancements, the initial trajectory for imitation learning was still derived from an optimal control framework, inheriting its inherent limitations and heavy reliance on predefined models. While this use of imitation learning marked a novel approach, it risked converging on locally optimal solutions that merely replicated the demonstrated trajectories, thus not sufficiently exploring other, potentially more efficient, navigational paths.

2.3 Research Questions

The primary objective of this research is to develop a comprehensive solution that addresses the key challenges associated with autonomous UAV navigation in environments characterized by dynamic obstacles. This study seeks to investigate how DRL can be leveraged to enhance UAV navigation, particularly in complex and unpredictable settings. While DRL shows great promise, the transition from controlled simulation environments to real-world applications presents several difficulties that need to be systematically explored. The following research questions are designed to guide the investigation and address these challenges:

- How can DRL algorithms be adapted to effectively handle dynamic obstacles in real-time environments? Given that environmental changes, such as the movement of other UAVs, animals, or humans, are not always well-represented in training data, how can a DRL model be designed to continuously learn and predict obstacle interaction patterns, thereby improving the UAV's adaptability to unforeseen situations [71]?

- How can sensor noise and inaccuracies in real-world data be mitigated to improve the performance of DRL algorithms in UAV navigation? What strategies can be employed to counteract the negative impact of sensor noise, inaccuracies, and environmental conditions (such as variable lighting and weather effects) that distort data and hinder state observation, thus compromising the accuracy of the UAV's decision-making process [51]?

- What methods can be developed to manage high-dimensional state spaces in real-time UAV navigation tasks? Considering the vast amounts of sensor data and the dynamic nature of the environments in which UAVs operate, how can DRL algorithms efficiently process high-dimensional state spaces within the constraints of limited computational resources, ensuring real-time responsiveness and stability [77]?

- How can the sim-to-real transfer problem be effectively addressed to ensure DRL algorithms trained in simulations perform optimally in real-world settings? What techniques can be implemented to bridge the gap between simulated environments and real-world applications, given that models trained in idealized simulation conditions often suffer from performance degradation when deployed in the unpredictable and complex dynamics of real-world scenarios [4]?

- What approaches can ensure real-time decision-making capabilities for UAVs operating in complex environments? How can DRL algorithms be optimized to operate at high speeds with minimal latency, ensuring real-time processing and decision-making in environments characterized by high dimensional data and limited onboard processing power [24]?

- How can algorithm stability and convergence issues in DRL models be mitigated to ensure optimal UAV performance? In continuous action spaces, how can stability and convergence problems be addressed, given the challenges of non-stationary target policies and the variability of reward signals that often lead to slow or unstable training and suboptimal policies [31]?

- What methods can be used to design effective reward functions that guide UAV behavior in dynamic environments? How can reward functions be designed to robustly handle variations in environmental conditions, ensuring that UAVs consistently learn desired behaviors, while avoiding unintended actions that might arise from poorly specified rewards [7]?

This thesis aims to develop a comprehensive solution to address key challenges in autonomous UAV navigation in unknown and dynamic environments. Efforts concentrate on strengthening DRL algorithms to achieve higher robustness and dependability, even when confronted with sensor imprecision, data noise, and challenging conditions such as fluctuating illumination and shifting weather. To mitigate these challenges, the study introduces advanced sensor data processing techniques, enhancing the quality and consistency of the state observations crucial for DRL. Moreover, the research tackles the issue of managing high-dimensional state spaces in UAV navigation, utilizing innovative algorithmic designs and op-timized computational strategies to enable real-time processing in environments characterized by constant changes and complexities. A key aspect of this work is the development of simulation environments that more accurately mirror the dy-namic and unpredictable nature of real-world settings, ensuring that DRL models trained in simulations are adaptable and robust when applied to practical scenarios. A central the real-time decision-mak-

ing capabilities of goal of this research is to advance DRL algorithms for UAV navigation. To achieve this, the study op-timizes the algorithms for higher operational speeds and reduced latency, which are critical for environments requiring immediate and accurate responses. This optimization process involves enhancing the efficiency of both the neural network architectures and the training algorithms, as well as reducing the computational burden through techniques such as model pruning, parallelization, and efficient state-action representation. The aim is to minimize the time delay between input sensing and the corresponding control action, ensuring that the UAV can respond to rapidly changing environments with minimal lag. Furthermore, the thesis addresses the issues of algorithm stability and convergence, with improvements in learning rates and reward structures to stabilize the training process and produce optimal navi-gation policies. Finally, the study contributes to the design of more sophisticated and adaptable reward functions that more precisely reflect the desired behaviors in UAV navigation, thus enabling more effective learning and greater autonomy.

Chapter 3

Methodology

This section delineates the architecture of the policy, the computational models employed, and the training methodologies utilized to develop a control mechanism for navigating a UAV through a complex gap scenario.

3.1 NEWDQN Algorithm

To address the complexities delineated earlier, a circumscribed annular configuration is introduced within a simulated environment, serving as a surrogate for a narrow aperture. Following this, an UAV is maneuvered through this annular design, thereby enabling an exhaustive simulation experiment. Within the purview of UAV operations in simulated settings, the precise delineation of elliptical structures emerges as a critical consideration.

3.1.1 One Circular Crossing Algorithm

This research presents a comprehensive methodological framework designed to develop an effective circular crossing algorithm. Illustrated in Fig. 3.1, entitled "Circular Crossing Detect Algorithm Overview," the methodology initiates with the preprocessing of raw image data using a Gaussian filter. This critical step is fundamental in reducing noise interference, thereby enhancing the clarity of the image and preparing it for subsequent analytical procedures. Following this, the Adaptive Canny edge detection method is employed, renowned for its ability to adjust to varying image conditions and reliably outline boundaries.

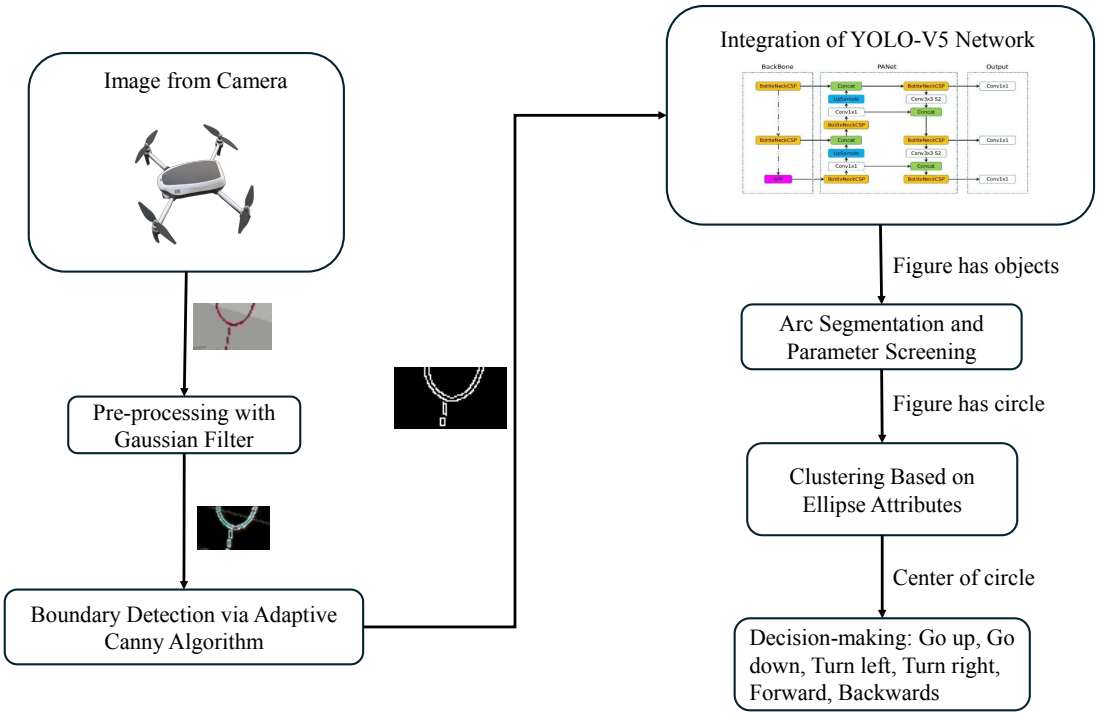


Figure 3.1: Circular crossing detect algorithm overview.

Further strengthening the detection process is the integration of the YOLO-V5 network [69], a leading-edge object detection model. This model, extensively trained on a diverse dataset of ellipses, excels in identifying ellipses under various conditions with exceptional precision and speed. Once the boundaries are established, they are segmented into concave and convex arcs. These segments are rigorously analyzed based on predefined criteria to improve accuracy. From these arc segments, parameters describing the potential ellipse are derived. A validation step follows, aiming to accurately locate the ellipse's centroid.

Subsequently, an ellipse score is calculated to assess the accuracy of the detection. The characteristics of the identified ellipse, including its center, major and minor axes, and angular orientation, are utilized in clustering processes to group similar ellipses. This application of deep learning models like YOLOv5 in UAV imagery for ellipse detection reflects significant advancements in the field, offering enhanced accuracy and reduced processing times, which are crucial for real-time UAV operations.

Relying on the results from the circular recognition detection, the UAV's control system adjusts its navigation based on the precise location of the identified ellipse. It is critical to recognize that as the UAV approaches the target within a meter, the target may become less discernible due to the limitations of the field of view. In such instances, the UAV, guided by its current positional data, continues its path through the target. Once this flight objective is successfully accomplished, the UAV proceeds back to its launch point, thereby concluding the overall operational sequence.

3.1.2 Two Circular Crossing Algorithm

In the pursuit of navigating dual circular structures, the detection of these circular entities adheres to the methodology previously detailed. However, the overarching strategy for global exploration in this context leverages the principles of RL.

The RL agent operates through iterative steps, receiving an observation denoted as o_t and a corresponding reward r_t from the environment at each interval. Based on these inputs, the agent executes an action a_t, driven by a policy π. This policy serves to map states to a probabilistic distribution of potential actions. The implementation of action a_t induces a transition in the environment from its current state s_t to the subsequent state $s_{(t+1)}$, concurrently delivering a new reward $r_{(t+1)}$ and an updated observation $o_{(t+1)}$. It is pertinent to acknowledge that in numerous real-world scenarios, the states of the environment are only partially observable. However, for the purposes of simplification in this model, we assume that $s_t = o_t$.

The overarching objective for the RL agent is to engage in iterative interactions with the environment to ascertain the optimal policy that maximizes the cumulative future reward or return. This return is mathematically formulated as $R_t = \sum_{i=t}^{T} \gamma^{(i-t)} r(s_i, a_i)$, where T represents the terminal time step and γ is the discount factor, which quantifies the decreasing significance of future rewards.

The reward function within this RL framework is meticulously designed, integrating four distinct components to strategically influence the agent's behavior:

1. Termination Reward-Penalty: This component rewards or penalizes the agent upon the completion or termination of a task, encouraging efficient task completion.

2. Step Reward-Penalty: This component is designed to optimize the number

of actions executed by the agent, thereby incentivizing the identification of the most efficient, shortest path available.

3. Direction Reward-Penalty: This aspect of the reward system confers benefits to the agent for sustaining an optimal trajectory towards the target, thereby enhancing direct and efficient navigational practices.

Together, these elements of the reward function are designed to ensure that the RL agent not only reaches its targets but does so in an efficient and effective manner, reflecting the complexities and demands of navigating through dynamic and potentially cluttered environments.

DQN is a seminal algorithm in the field of RL, well-regarded for its utility in complex decision-making scenarios. In this research, the DQN framework is applied to a specific aerial navigational problem where the UAV, functioning as the tracker, must locate and traverse through a target identified as the "circular". This task is set against the backdrop of intricate environments, notably those punctuated by narrow gaps, which present significant challenges for UAV navigation.

The intricacies of such environments often produce a vast quantity of interaction data, which complicates the training of an end-to-end neural network via DRL. This research is dedicated to the development and refinement of an innovative DRL algorithm specifically engineered for real-time UAV path planning in environments characterized by narrow gaps. The proposed algorithm is designed to translate sensor data directly into action commands, thereby enhancing the UAV's autonomous decision-making capabilities in dynamically changing environments.

A pivotal element of this algorithm is its ability to aggregate state and path data from various narrow-gap scenarios, utilizing this information to train a deep neu-

ral network effectively. This training approach is strategically developed to facilitate the swift generation of optimized navigation paths in familiar terrains, while strictly adhering to tight temporal constraints. Moreover, the research will investigate diverse neural network architectures to process different sensor datasets, aiming to establish a robust environmental feature set. This feature set is intended to underpin a UAV path planning methodology that leverages state-of-the-art DRL techniques.

An essential component of this research involves conducting a comparative analysis of the proposed DRL algorithm against traditional navigation strategies. Furthermore, the study will rigorously assess the adaptability and effectiveness of the path planning approach, particularly in environments characterized by narrow gaps, which are central to this innovative algorithm.

In terms of exploration strategies, the DQN algorithm typically utilizes an ε-greedy approach to balance exploration with exploitation effectively. This method is articulated mathematically as follows: the exploration rate ε is incrementally adjusted over iterations according to the formula $\varepsilon_{(i+1)} = \varepsilon_i + \Delta\varepsilon$, constrained within the bounds $\varepsilon_{min} \leq \varepsilon_i \leq \varepsilon_{max}$. The action selection for the tracker, based on this exploration rate, alternates between exploiting the best-known action, determined by $\arg\max_a Q(o_t, a)$, and exploring new actions randomly, dictated by the condition $f_{rand} \geq \varepsilon_{(i+1)}$, ensuring a comprehensive assessment of possible strategies. This ε-greedy exploration strategy is crucial for refining the UAV's navigational commands, optimizing the balance between exploring uncharted paths and exploiting known trajectories to enhance operational efficiency in complex aerial environments.

$$\begin{cases} \varepsilon_{i+1} = \varepsilon_i + \Delta\varepsilon, \varepsilon_{\text{min}} \leqslant \varepsilon_i \leqslant \varepsilon_{\text{max}} \\ a_{i+1}^t = \begin{cases} \text{argmax}_a \, Q\left(o_t, a\right), & f_{\text{rand}} < \varepsilon_{i+1} \\ \text{random}(A), & f_{\text{rand}} \geqslant \varepsilon_{i+1} \end{cases} \end{cases} \tag{3.1}$$

The ε-greedy strategy is fundamental to the operational dynamics of the DQN algorithm, functioning as a critical mechanism for balancing exploration and exploitation during the learning process. This strategy functions by permitting the tracker to explore alternative actions with a probability defined by ε, thereby enabling the discovery of potentially more effective solutions. As learning advances across successive iterations, ε is gradually reduced to increase the tracker's dependence on its accumulated knowledge, progressively transitioning from exploration to exploitation.

The configuration and decay schedule of ε within the DQN framework are meticulously adjusted to align with the particularities of the problem and the model's parameters. Typically, ε is initialized at a high value, often 1.0, to prioritize exploration during the early stages of training. This high level of exploration ensures that the tracker is not prematurely confined to a limited area of the action space, thus avoiding local optima and encouraging a thorough search of the environment.

Over time, ε is methodically decreased according to a predefined decay schedule, such as linear, exponential, or step decay. This gradual reduction is designed to decrease the rate of exploration while correspondingly increasing the rate of exploitation. By adjusting ε, the algorithm progressively focuses more on leveraging the best-known strategies derived from past experiences rather than seeking out new ones.

The utilization of the ε-greedy exploration strategy allows the DQN algorithm

to strike an effective balance between exploring new possibilities and exploiting learned behaviors. This balance is crucial for the algorithm's ability to converge towards the most effective policies while mitigating the risk of converging to sub-optimal solutions. Through this strategic adjustment of ε, the DQN algorithm optimizes its performance, enhancing its capability to navigate complex decision-making environments efficiently.

In reinforcement learning, the ε-greedy strategy holds a pivotal role by guiding the tracker's gradual move from exploration to exploitation, thereby steadily enhancing its decision-making abilities. However, the unpredictable nature of complex environments can sometimes make initial explo-ration insufficient for the tracker to reliably determine the optimal policy. Further-more, even after identifying what appears to be an optimal policy, the tracker may become trapped in local optima due to inadequate reinforcement signals.

To mitigate these challenges, this research introduces an enhancement to the traditional ε-greedy strategy by incorporating the OBE strategy. This innovative approach is designed to augment the tracker's exploratory capabilities within mul-tifaceted environments. The core concept of the OBE strategy is to dynamically adjust the exploration rate, ε, based on the tracker's performance over a speci-fied period. Specifically, if the tracker repeatedly fails to achieve the desired task within this timeframe, ε is incrementally increased to enhance exploration, thereby enabling the tracker to escape local optima and potentially discover more effective strategies.

The implementation of the OBE strategy is meticulously tailored to the specific requirements of the problem and the intricacies of the model configuration. It involves continuous monitoring of the tracker's performance and employs a sys-

tematic approach to adjust ε based on predefined criteria or performance thresholds. These adjustments can be made at regular intervals or triggered by specific events, thereby ensuring that the tracker consistently maintains an optimal balance between exploring new actions and exploiting established effective strategies.

Integrating the OBE strategy into the exploration framework introduces a more advanced and adaptive exploration mechanism. This integration enables the tracker to recalibrate its exploration activities in response to real-time performance metrics. As a result, the strategy enhances the tracker's ability to navigate complex decision-making environments, effectively addressing the challenges posed by local optima and promoting a more resilient approach to discovering optimal policies.

$$\varepsilon_{i+1} = \begin{cases} \varepsilon_{\mathrm{m}}, & C_{\text{fail}} > C \\ \varepsilon_i + \Delta\varepsilon, & \text{other} \end{cases} \tag{3.2}$$

where the exploration rate, represented as ε_i, is crucial in determining the UAV's capacity to effectively navigate and learn from its environment. The parameter $\Delta\varepsilon$ represents the incremental adjustment to ε in each iteration, which is crucial for adapting the UAV's exploration strategy based on its ongoing performance. Furthermore, C_{fail} quantifies the number of consecutive task failures experienced by the UAV, serving as a critical metric for assessing the necessity of strategic adjustments.

Within this methodological framework, the variable C_{fail} measures the sequential failures of the UAV in completing its designated tasks, while C denotes the maximum threshold of allowable consecutive failures before a forced recalibra-

tion of the exploration rate occurs. If the UAV encounters a number of task failures that meet or exceed this threshold, the exploration rate is reset to $\varepsilon_m = \varepsilon_i$, where i represents the current iteration. This adjustment is intended to significantly improve the UAV's ability to explore its environment more comprehensively, thereby increasing the likelihood of identifying globally optimal solutions.

The specific values and adjustment schedule for $\varepsilon_m = \varepsilon_i$ are carefully tailored to address the unique requirements of the problem and the model's operational parameters. By requiring an increase in the exploration rate following a se-ries of task failures, this approach ensures that the UAV intensifies its exploratory efforts, thereby enhancing its ability to navigate and adapt to its environment effec-tively. This strategic recalibration aims to mitigate the risk of the UAV becoming ensnared in local optima and facilitates a more expansive exploration of potential strategies and states. Such an approach is instrumental in potentially uncovering more efficacious solutions to the challenges encountered during navigation.

Navigating the complex domain of UAVs and the intricacies of their operational algorithms necessitates a profound understanding of the mechanisms underlying their behavior. In the DQN paradigm, the UAV, referred to as the "tracker," is tasked with locating and navigating through a target. The iterative interaction between the tracker and its environment generates a wealth of experiential data, which is accumulated in an experience pool. The effectiveness of the UAV's learning process is fundamentally tied to the quality of the stored data, highlighting the critical importance of strategic data management and algorithmic adjustments in improving overall navigational performance.

In the development of reinforcement learning algorithms for UAVs, a critical challenge frequently encountered is the entrapment of the UAV, or "tracker," in

a local optimum. In such instances, the repetitive reinforcement of the tracker's actions can obscure valuable insights from the experiential data, consequently hindering its ability to escape from these suboptimal solutions. To address this issue, this research introduces a novel algorithm, designated as NEWDQN, which incorporates significant enhancements specifically designed to overcome these limitations.

The NEWDQN algorithm integrates two key modifications aimed at enhancing the robustness and efficacy of the traditional DQN framework: Optimistic Bootstrapping Exploration (OBE) strategy, coupled with the multi-experience pool mechanism, enhances the exploration component of the DQN algorithm. The OBE strategy actively revitalizes exploration by encouraging the selection of exploratory actions. This strategic emphasis on exploration encourages the tracker to engage with a broader array of states and actions, thus increasing the probability of escaping from local optima and facilitating the discovery of more effective navigational strategies.

Concurrently, the multi-experience pool mechanism addresses the shortcomings associated with relying on a single experience pool. This innovative approach categorizes accumulated experiential data into distinct pools based on the outcomes of the tracker's actions, thereby improving the quality of the data sampled for learning. This mechanism is particularly crucial as it enables the maintenance of data integrity and relevance, which are essential for effective learning and adaptation.

The NEWDQN algorithm uniquely integrates three distinct experience pools: a failure experience pool, a success experience pool, and a temporary experience pool. This tripartite structure allows for a nuanced management of experiential

data, where each pool serves a specific purpose in the learning process. The failure experience pool collects data from unsuccessful navigational attempts, offering insights into the conditions and decisions leading to suboptimal outcomes. The success experience pool gathers data from successful missions, providing templates for effective actions and strategies. Lastly, the temporary experience pool serves as a dynamic repository for ongoing assessments and adjustments in strategy, facilitating immediate responsiveness to changing environmental conditions.

This stratified approach to experience management within the NEWDQN algorithm significantly mitigates the risk of continual reinforcement of suboptimal actions and enhances the tracker's overall learning trajectory. By implementing these targeted modifications, the NEWDQN algorithm ensures a more stable convergence process, even in the face of complex tasks that might otherwise predispose the tracker to prolonged periods of exploration without adequate reinforcement. Thus, the algorithm not only enhances the UAV's capability to navigate through challenging environments but also ensures robust learning and adaptation over time.

In the sophisticated domain of UAV pathfinding, the strategic management of experiential data is crucial. The NEWDQN algorithm utilizes a structured experience pool system, consisting of a failure experience pool, a success experience pool, and a temporary experience pool, each designed to optimize learning from distinct operational outcomes.

The failure experience pool collects data from instances where the tracker fails to complete a task, providing valuable insights into ineffective strategies and decision-making processes. This information is essential for identifying and refining approaches that require improvement. In contrast, the success experience

pool collects data from successful task completions or effective navigations, serving as a repository of proven strategies and actions that should be reinforced and emulated.

Additionally, the temporary experience pool acts as a dynamic buffer, temporarily holding data from current operations. When this pool reaches capacity, and the tracker has not yet arrived at a terminal state, it indicates that the decisions made have had a positive impact. The overflow data is then transferred to the success experience pool, assuming these actions have contributed to favorable outcomes. The final allocation of the data remaining in the temporary pool is contingent upon the success or failure of the tracker's ultimate task.

Data utilization from these pools follows a strategic sampling method. Let B represent the proportion of data sampled from the success experience pool, and let β denote the total amount of data obtained through random sampling. The quantities of data drawn from the success and failure experience pools are determined based on B, underscoring the significance of learning from both successful and unsuccessful experiences to refine the tracker's decision-making processes as follows:

$$\begin{cases} b_1 = \begin{cases} D_s, & D_s \leqslant \beta B \\ \beta B, & \text{other} \end{cases} \\ b_2 = B - b_1 \end{cases} \tag{3.3}$$

In the context of the NEWDQN algorithm, data management is meticulously orchestrated to optimize the learning trajectory of UAVs. In this context, b_1 and b_2 represent the quantities of data samples drawn from the success and failure

experience pools, respectively. The variable D_s denotes the total number of data samples currently residing in the success experience pool. During the algorithm's update process, data is systematically sampled from both the success and failure experience pools according to predetermined proportions. This approach ensures that, even when the UAV, referred to as the tracker, finds itself ensnared in a local optimum, it retains the ability to access and learn from the successful experiences stored, thereby facilitating a swifter escape from such suboptimal positions.

However, it is important to acknowledge that improving the tracker's exploration capabilities can introduce substantial fluctuations in the algorithm's convergence process. In the context of particularly challenging tasks, there is a risk that the tracker may persist in a continuous state of exploration without obtaining adequate reinforcement. Such a scenario can impede the convergence process, leading to potential instability and inefficiency.

Fig. 3.2 shows the NEWDQN structure. To mitigate these challenges, this research proposes the adoption of a multi-experience pool mechanism, which categorizes experiential data into three distinct pools: the failure experience pool, the success experience pool, and the temporary experience pool. Such a division not only ensures the preservation of data quality across different contexts but also enhances the robustness of the learning process. By maintaining separate pools, the algorithm can more effectively manage the data relevance and utility, ensuring that the tracker has access to a diverse range of experiences. This arrangement helps to stabilize the convergence of the algorithm and enables the tracker to navigate out of local optima more effectively. In essence, the multi-experience pool mechanism serves as a critical component in fostering a resilient and effective learning environment for UAVs. It ensures that the tracker is not only equipped to han-

dle the complexities of dynamic environments but also improves its overall task performance and navigational capabilities.

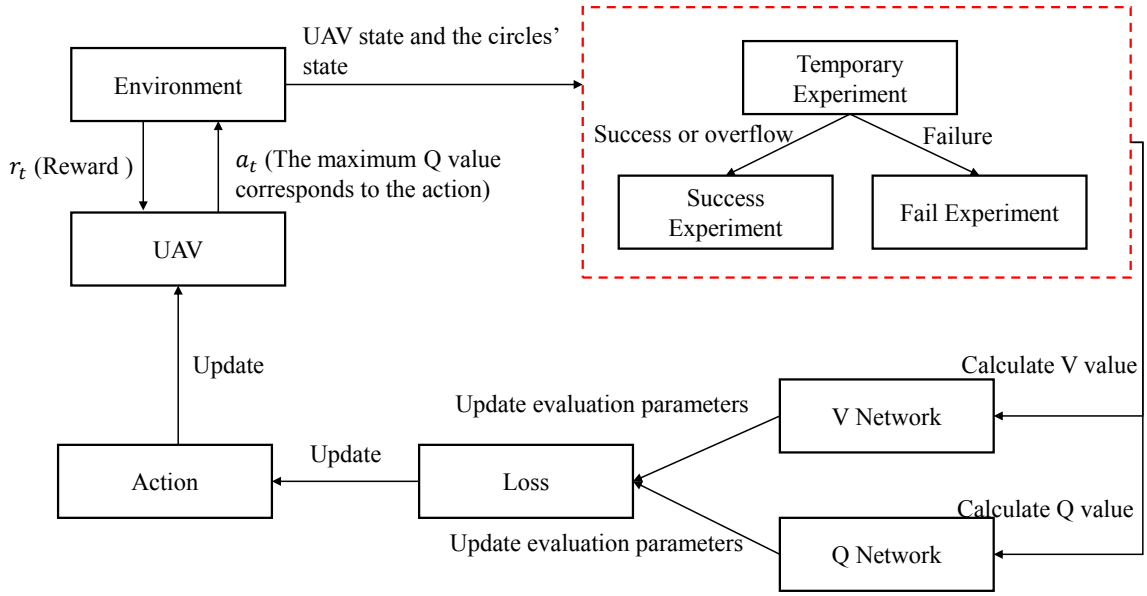


Figure 3.2: NEWDQN structure.

3.2 SAC Algorithm

3.2.1 Problem Statement

3.2.1.1 Dynamics

This section outlines the mathematical framework governing the dynamics of a UAV, incorporating the effects of aerodynamic drag alongside the inherent phys-

ical properties of mass and inertia. The UAV is modeled as a rigid body with a uniform mass distribution and axial symmetry, characterized by constant mass and moment of inertia parameters.

The UAV is modeled as a rigid body with a uniform mass distribution, which simplifies the dynamic equations by ensuring that the mass and moment of inertia remain constant throughout flight. It is assumed that the UAV's center of gravity coincides with its geometric center, a critical assumption that simplifies the analysis by eliminating the need to account for torques and forces resulting from any misalignment between these centers. The lift generated by blade flapping is assumed to occur within the same plane as the center of gravity, implying that the UAV's thickness does not significantly influence its aerodynamics. This assumption further simplifies the model by neglecting any pitch or roll moments that could arise from offset lift surfaces. Additionally, the model disregards friction between the propeller and its motor spindle, as well as aerodynamic drag on the UAV's body. By excluding these factors, the focus remains on the primary forces and torques, without the complexities introduced by mechanical and viscous resistances. Furthermore, the model does not consider the curvature of the Earth or its rotational motion, an approximation typically acceptable for UAVs operating at low altitudes and within small geographical areas, where such factors have minimal impact on the UAV's dynamics.

$$\dot{\mathbf{p}} = \mathbf{v}$$

The dynamics model of the UAV can be expressed as follows [53]:

$$\dot{\mathbf{p}} = \mathbf{v}, \tag{3.4}$$

This equation indicates that the rate of change of the UAV's position, $\dot{\mathbf{p}}$, is equivalent to its velocity, $\mathbf{v}$. It establishes the fundamental principle that the UAV's position evolves in direct relation to its velocity vector.

$$\dot{\mathbf{R}}_g = \mathbf{R}_g \left[\boldsymbol{\omega}_b\right]_\times, \tag{3.5}$$

This equation illustrates how the orientation of the UAV, represented by the rotation matrix $\mathbf{R}_g$, evolves over time in response to its angular velocity $\boldsymbol{\omega}_b$. The skew-symmetric matrix $\left[\boldsymbol{\omega}_b\right]_\times$ is utilized to compute rotational velocities in three dimensions, enabling a precise representation of the UAV's rotational dynamics.

$$m\boldsymbol{v} = m\boldsymbol{e}_3 g + \boldsymbol{R_g}\boldsymbol{e}_3 f_t + \boldsymbol{f}_d, \tag{3.6}$$

This equation describes the net force acting on the UAV, which results in its translational acceleration $\boldsymbol{v}$. The term $m\boldsymbol{e}_3 g$ represents the gravitational force acting on the UAV, where m denotes the UAV's mass, g is the acceleration due to gravity, and $\boldsymbol{e}_3 = [0, 0, 1]^T$ is a unit vector directed vertically (typically upwards). This term effectively accounts for the weight of the UAV. The variable f_t represents the total thrust generated by the UAV's rotors. The rotation matrix $\boldsymbol{R}_g$ is used to convert this thrust from the UAV's body frame to the Earth's frame, allowing for the determination of the UAV's Euler angles (ϕ, θ, ψ). This term captures the upward force generated by the rotors, which enables the UAV to lift off or

move vertically. The term $\boldsymbol{f}_d$ denotes the aerodynamic drag acting on the UAV, which opposes its motion. This force depends on the UAV's velocity components (v_x, v_y, v_z) in the Earth's frame and typically increases with the square of the velocity.

$$\mathbf{J}\dot{\omega}_b = \boldsymbol{\tau}_T + \boldsymbol{\tau}_D - \boldsymbol{\omega}_b \times \mathbf{J}\boldsymbol{\omega}_b, \tag{3.7}$$

This equation relates the rate of change of angular momentum, $\mathbf{J}\dot{\omega}_b$, to the net torque acting on the UAV. In this context, $\boldsymbol{\tau}_T$ represents the thrust torque, τ_D denotes the drag torque, and $\boldsymbol{\omega}_b \times \mathbf{J}\boldsymbol{\omega}_b$ accounts for the gyroscopic effect. This relationship is fundamental for understanding how the UAV's rotational dynamics are influenced by the applied torques and its intrinsic angular momentum.

Incorporating these assumptions into the model simplifies the analysis and enhances its mathematical tractability, allowing for a clearer understanding of the fundamental dynamics governing quadrotor UAV flight. This theoretical framework provides a solid foundation for developing control strategies aimed at improving the UAV's stability and maneuverability in practical applications.

3.2.1.2 Problem Formulation

The objective is to design bold flight paths that enable a UAV to successfully navigate through a slanted aperture, as illustrated in Fig. 3.3. In this depiction, the UAV is represented by a black circle symbolizing its bounding box, while a grey rectangular background represents a wall with an opening, as detailed in Fig. 3.4. The figure exemplifies a scenario where the UAV satisfies the spatial requirements to pass through the gap without altering its trajectory. However, the combined

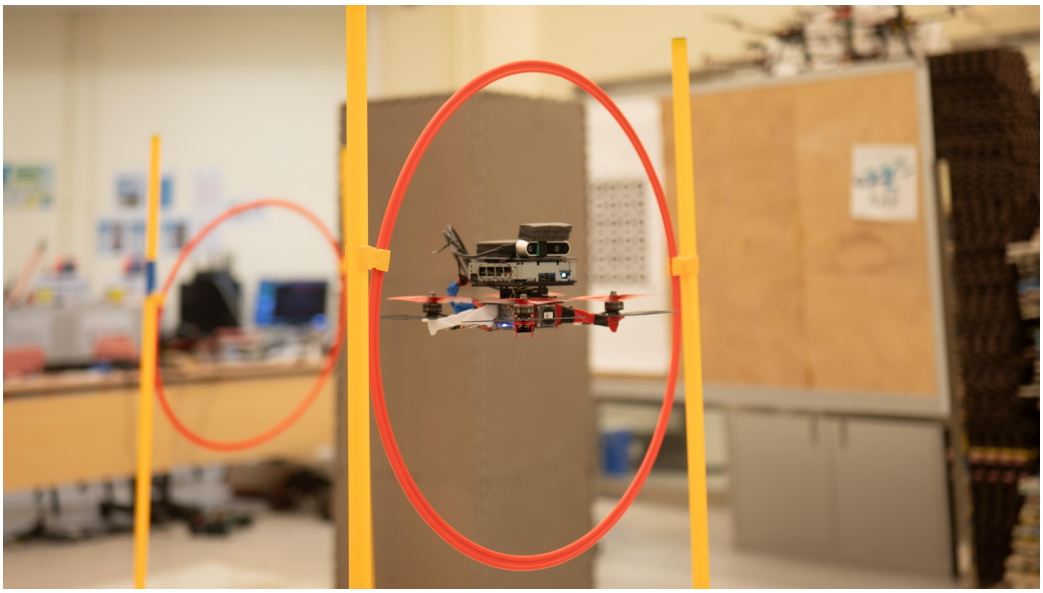


Figure 3.3: The UAV guided by RL is maneuvering through a tilted narrow gap.

forces generated by the UAV's motors and the gravitational pull can induce unintended horizontal movements, increasing the risk of collision, as indicated by the red arrows in the illustrations. Furthermore, employing a forward pitch to enhance speed and maneuverability can expand the UAV's lateral dimensions, thereby reducing the clearance between the UAV and the edges of the gap and diminishing the overall safety margin.

In this context, S denotes a continuous state space, while A represents a continuous action space. At each decision point, given the current state s_t from S, a continuous action a_t from A is selected according to a decision-making policy $\pi(a_t \mid s_t)$. Upon choosing action a_t, the system transitions to a new state s_{t+1} within S, governed by an unspecified probability distribution $p : S \times A \to S$. The process generates a reward, defined by the function $r : S \times A \to [r_{\text{min}}, r_{\text{max}}]$, which is constrained within a predetermined range. The state space S compre-

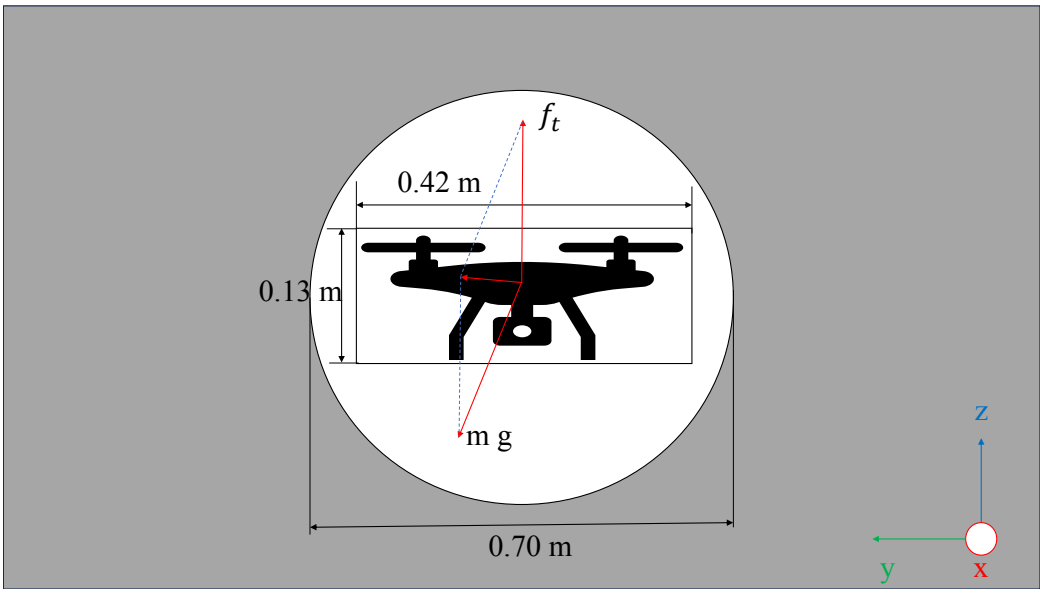


Figure 3.4: One possible traverse.

hensively includes both the UAV's states, denoted by x within the set X, and the positions of the gap (0.70 m), denoted by g within the set G. The primary objective of this algorithmic approach is to develop a control policy $\pi : X \times G \rightarrow A$, which optimizes the UAV's interactions with its environment to facilitate effective navigation and successful task completion.

3.2.2 Simulation Environment

To ensure replicable and verifiable results, our simulation environment was meticulously constructed using the Gazebo, integrated with the PyTorch Machine Learning implemented in Python. Our experiments are conducted within a specifically designed environment in Engine. Each quadrotor UAV, acting as an autonomous learning agent, was configured with specific parameters, including sensor types, sensor noise levels, and initial positioning. These parameters were chosen to

closely mimic real-world flying conditions, with UAVs equipped with standard navigation sensors such as binocular depth camera and IMU. Detailed logging of each simulation run was implemented to capture data on navigation accuracy, obstacle avoidance effectiveness, and computational efficiency, ensuring comprehensive analysis and evaluation of the SAC algorithm's performance under varied environmental conditions. Binocular depth cameras, recognized for their compact size and energy efficiency, play a crucial role in capturing high-resolution depth maps of the environment. These cameras, particularly suitable for scene recognition tasks in conjunction with UAVs, enable the direct acquisition of depth maps within the Gazebo simulation environment. In our experiments, these depth maps are used as inputs for the deep reinforcement learning network, significantly improving the UAVs' navigational accuracy.

The simulation setup includes various slender circles and barriers. Simulation episodes are immediately terminated upon detecting any overlap between the UAV and these elements, indicating a collision. To achieve this, a simple yet effective collision detection algorithm is employed. This algorithm calculates, in real time, the intersection points where the UAV's bounding box meets a wall. A collision is confirmed if any of these intersection points lie outside the designated gap. The simulation considers a traversal successful when the UAV reaches its target position, $\boldsymbol{p}_G$, without any recorded collisions.

3.2.3 Network Design

3.2.3.1 Soft-Actor-Critic Framework

The state space for our neural network model includes comprehensive details pertaining to both the UAV and the narrow gap. Specifically, the state space comprises the UAV's position and orientation, along with the gap's relative position and orientation in relation to the UAV. The relative position of the gap with respect to the UAV is defined by the vector difference between their respective center positions in the global frame, denoted as:

$$\phi_r = \phi_{gm} - \phi_{um}. \tag{3.8}$$

To quantify the orientation differences, the Euler angles of the UAV are subtracted from those of the gap:

$$\theta_\tau = \theta_{gm} - \theta_{um}. \tag{3.9}$$

The Euler angles of UAV provide a comprehensive three-dimensional representation of its orientation in space, encompassing roll, pitch, and yaw. The roll angle, which measures rotation around the UAV's longitudinal axis (x-axis), ranges from -180° to +180°, indicating the tilt of the UAV to the left or right. The yaw angle, representing rotation around the vertical axis (z-axis), governs the UAV's heading and also spans from -180° to +180°, with positive values corresponding to counterclockwise rotation and negative values to clockwise rotation when viewed from above. The pitch angle, defined by rotation around the lateral axis (y-axis), controls the ascent or descent of the UAV and is constrained between -90° and

+90°, with positive values signifying upward tilt and negative values indicating downward tilt.

The observation space is structured to incorporate both the UAV and gap states. To ensure efficient data processing and minimize input dimensionality, the focus is placed on the relative position, represented as a normalized vector. This approach allows for the concise utilization of directional and distance information:

$$p_i^e = \text{sign}\left(p_{gm} - p_{um}\right) \sqrt{|p_{gm} - p_{um}|}. \tag{3.10}$$

This method is applied independently along the x, y, and z axes. The model focuses on the roll and pitch angles for the UAV's orientation, deliberately excluding the yaw angle to streamline the model and maintain a targeted focus. The UAV's motion parameters, including linear velocities (v_x, v_y, v_z) and angular velocities $(\omega_x, \omega_y, \omega_z)$, are critical for executing precise maneuvers toward the gap. Consequently, the state space is composed of 11 dimensions, integrating relative positional and angular information with the UAV's kinematic data. This approach provides a comprehensive yet efficient framework, well-suited for guiding the UAV through complex trajectories.

The following Algorithm 1 presents the SAC-based pseudo code utilizing importance sampling [83].

In the realm of reinforcement learning, the SAC algorithm has emerged as a particularly robust approach, recognized for its sample efficiency and stability. SAC employs a stochastic policy that enhances exploration capabilities, thereby reducing the risk of the agent becoming trapped in local optima. The method Twin Delayed Deep Deterministic Policy Gradient (TD3) can encounter diffcul-

Algorithm 1 Algorithm of SAC

1: Initialization: A given behavior policy $\beta(a \mid s)$. A target policy $\pi\left(a \mid s, \theta_0\right)$ where θ_0 is the initial parameter vector. A value function $v\left(s, w_0\right)$ where w_0 is the initial parameter vector.
2: **repeat**
3: At time step t in each episode, do
4: Generate a_t following $\beta\left(s_t\right)$ and then observe r_{t+1}, s_{t+1}.
5: Update TD error (advantage function):

$$\delta_t = r_{t+1} + \gamma v\left(s_{t+1}, w_t\right) - v\left(s_t, w_t\right)$$

6: Update Critic (value update):

$$w_{t+1} = w_t + \alpha_w \frac{\pi\left(a_t \mid s_t, \theta_t\right)}{\beta\left(a_t \mid s_t\right)} \delta_t \nabla_w v\left(s_t, w_t\right)$$

7: Update Actor (policy update):

$$\theta_{t+1} = \theta_t + \alpha_\theta \frac{\pi\left(a_t \mid s_t, \theta_t\right)}{\beta\left(a_t \mid s_t\right)} \delta_t \nabla_\theta \ln \pi\left(a_t \mid s_t, \theta_t\right)$$

8: **until** maximizing $J(\theta)$.
Output: Search for an optimal policy by maximizing $J(\theta)$.

ies in noisy environments [66]. TD can limit exploration and potentially lead to suboptimal performance. The Double Deep Q-Network (DDQN), while effective, can exhibit high variance during the early phases of training and demands significant computational resources due to its com-plex architecture. PPO may demonstrate inefficiencies when dealing with very large state or action spaces.

The following outlines the fundamental learning objectives of reinforcement

learning in its most basic form [64]:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[\sum_t R\left(s_t, a_t\right) \right]. \tag{3.11}$$

The reinforcement learning framework, when augmented with maximum entropy [58]:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \left[\sum_t R\left(s_t, a_t\right) + \alpha H\left(\pi\left(\cdot \mid s_t\right)\right) \right]. \tag{3.12}$$

This equation introduces randomness into decision-making processes, thereby enhancing the agent's exploratory capabilities.

Previously, the deterministic policy algorithm focused on identifying an optimal trajectory and concluded the learning phase upon its discovery. Currently, our objective is to achieve maximum entropy, which necessitates that the neural network explore every conceivable optimal trajectory [46].

$$J_\pi(\phi) = \mathbb{E}_{s_t\sim\mathcal{D},\varepsilon\sim\mathcal{N}} \left[\alpha \log \pi_\phi\left(f_\phi\left(\varepsilon_t; s_t\right) | s_t\right) - Q_\theta\left(s_t, f_\phi\left(\varepsilon_t; s_t\right)\right) \right]. \tag{3.13}$$

However, due to the dynamic nature of reward variations, employing a static temperature setting in this context is impractical and can result in instability during the training process. It is therefore crucial to adjust the temperature dynamically. As the policy explores new and unknown areas where the optimal action is not yet determined, a higher temperature is essential to enable a more extensive exploration. Conversely, in well-explored regions where the optimal action has been established, it is prudent to reduce the temperature to fine-tune the learning outcomes. The following updated formula incorporates a variable weight for entropy

across different states [78]:

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_t} \left[-\alpha \log \pi_t \left(a_t \mid \pi_t \right) - \alpha \mathcal{H}_0 \right]. \tag{3.14}$$

During the training phase of the Soft Actor-Critic (SAC) method, a systematic strategy is employed that involves interactive engagements with the environment and the meticulous documentation and storage of each interaction's details in a memory buffer [72]. These details encompass the state before an action s_t, the action itself a_t, the resultant reward r_t, and the subsequent state s_{t+1}. Data tuples (s_t, s_{t+1}, r_t, a_t) are subsequently extracted from this buffer to evaluate the effectiveness of the transitions $s_t \rightarrow a_t \rightarrow s_{t+1}$ through the computation of their Q-values. This evaluation acts as a crucial metric for refining our strategy, guiding it towards the maximization of expected rewards. This structured approach is instrumental in optimizing the algorithm's performance over time.

Figure 3.5 presents an overview of the training process. DRL offers significant advantages over traditional control methods for navigating UAVs through narrow obstacles, particularly excelling in complex and uncertain environments where traditional methods may falter due to their dependence on predefined models and parameters. Unlike conventional approaches that necessitate extensive programming and precise adjustments for specific scenarios, DRL empowers UAVs to autonomously learn optimal strategies through trial and error, acquiring experience directly from interactions with the environment. This capability enables DRL-based systems to adeptly adapt to dynamic environmental changes, such as moving obstacles or fluctuating wind patterns, which pose frequent challenges in real-world UAV navigation.

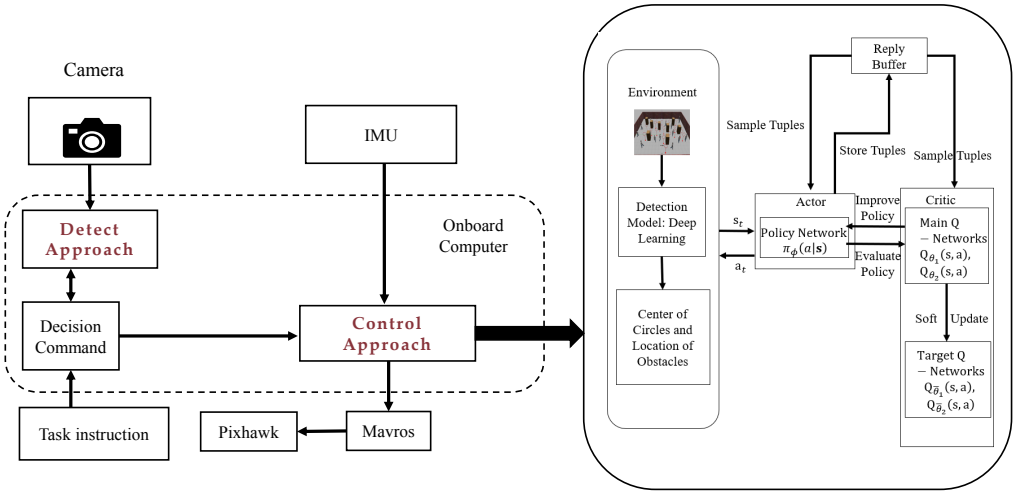


Figure 3.5: Schematic of the complete system training process.

We have simultaneously developed a detection module that utilizes a stereoscopic depth camera to generate temporal images corresponding to depth maps. This module also computes the UAV's position relative to obstacles or rings by analyzing the correlations across successive frames of images.

3.2.3.2 Detection Model

Policy-based DRL methods are particularly suited to handling continuous action spaces. These methods directly generate actions without the need to explicitly compute value functions, thereby simplifying their application to continuous domains. Such approaches require algorithms with high precision and offer a balanced approach to exploration and exploitation. Given that policies are generated directly, these methods can more effectively manage the exploration-exploitation trade-off, enhancing the flexibility in exploration strategies. However, policy-

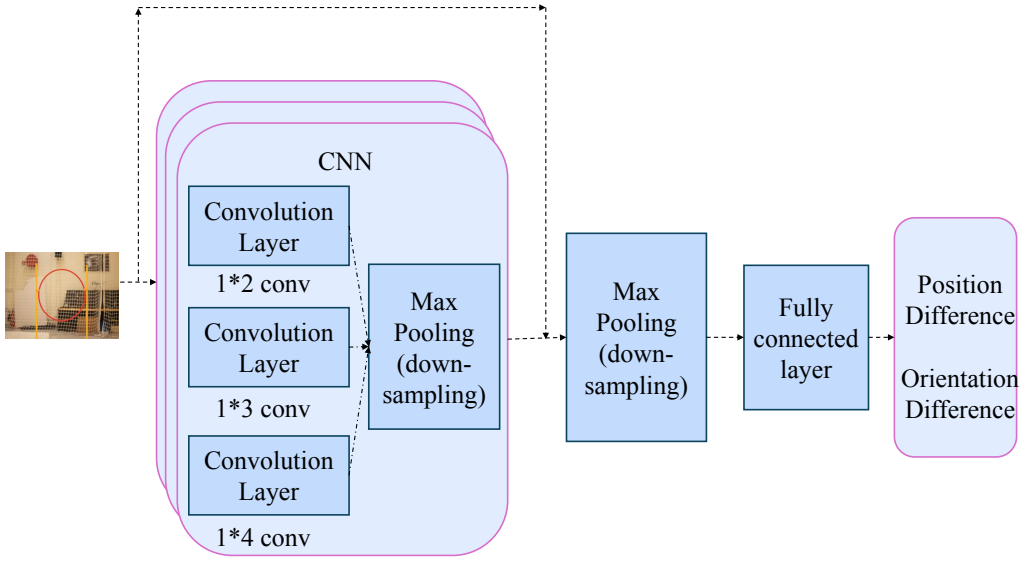


Figure 3.6: The structure of the detection model.

based methods generally demand a greater number of samples and increased computational resources, particularly in high-dimensional state spaces. Consequently, the SAC method is limited to employing shallow neural networks to ensure expedient model fitting. However, shallower neural models frequently face inaccuracies and display restricted robustness in image recognition tasks, as they lack the capacity to thoroughly handle intricate image data. To address this limitation, we have trained a recognition model to concurrently identify im-ages captured by the UAV's onboard stereoscopic camera, serving as inputs to this model. The convolutional neural network (CNN) architecture of this model includes three convolutional layers and one max-pooling layer, the structure of which is detailed in Fig. 3.6.

The initial segment of the model encompasses a CNN section, which consists of three parallel one-dimensional convolutional layers applied to the input sliced

time series. Each convolutional layer is equipped with 32 filters and respective kernel sizes of 2, 3, and 4. The CNN section also features a 2×2 max-pooling layer. The activation function within the CNN layers is the tanh function. The outputs of the CNN section are relayed to the subsequent part of the model via a hidden layer that includes a 0.5 dropout rate to prevent overfitting, particularly in highly expressive networks. This CNN section undertakes convolution and down-sampling of the input images.

The subsequent component of the deep neural network includes max-pooling and fully connected layers. The segmented time series data from the CNN is classified through the fully connected layer after transitioning through a max-pooling layer with a dimensionality of 128. Setting appropriate epochs is essential to avoid unnecessary computation and overfitting, while employing smaller batch sizes during experiments can enhance the model's ability to generalize in solving classification challenges. High batch sizes demand substantial memory resources and may hinder the program's functionality. Conversely, smaller batches tend to yield improved performance on generic models. The loss function calculates the training set loss every 100 batches, using algorithms to adjust neuron weights. The model utilizes a cross-entropy loss function and the Adam optimizer.

The input to the detection model comprises color images and depth maps obtained via a stereo camera setup. This dual-input strategy enriches the model's capacity to interpret the visual field with a more comprehensive contextual understanding, blending texture, color, and spatial depth information crucial for precise object recognition. The model processes this data to identify the pixel coordinates of the center of a ring in the color images, then extracts the corresponding depth information from the depth maps. This procedure accurately determines the

three-dimensional coordinates of the ring's center by integrating both visual and depth data provided by the stereo cameras. Calculating these three-dimensional coordinates is pivotal as it facilitates the determination of both angular and positional discrepancies between the UAV and the circle's center, which are essential for navigation and obstacle avoidance tasks.

This dual-input methodology, which incorporates both color and depth information, fosters a more reliable and robust recognition system. By leveraging depth data, the model can circumvent some limitations inherent in relying solely on visual cues, such as variations in lighting and color ambiguities. The depth data offers an absolute scale reference, enhancing the precision of object localization and enabling accurate spatial judgments necessary for complex navigation tasks. This approach significantly bolsters the UAV's operational reliability, allowing for more precise environmental assessments and more effective decision-making processes based on real-time data analysis. The input image data to the model measures $640 \times 480 \times 6$, and the output quantifies the relative position and angular discrepancy from the UAV to the nearest circle's center.

3.2.3.3 Architecture of the SAC Networks

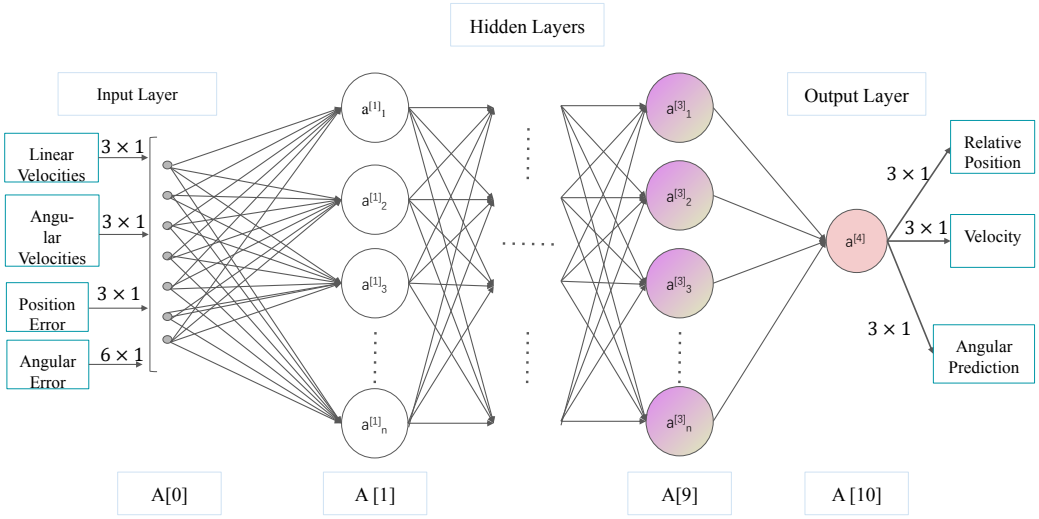
The architecture of the SAC networks is illustrated in Fig. 3.7. The actor network comprises nine fully connected layers, each equipped with 256 nodes, and utilizes the tanh activation function. The actor network receives an input of 15 vectors and generates an output of 9 vectors. These outputs are predictions pertaining to relative position, velocity, and acceleration. In the context of UAV navigation and target tracking, angular error is represented as a 6-dimensional vector, where the first three components correspond to the angular differences between the UAV's

orientation and the center of the target ring, while the remaining three components represent the angular differences between the UAV's orientation and the apex of the target ring. Specifically, the first set of three dimensions quantifies the deviations in roll, pitch, and yaw between the UAV's current heading and the line of sight toward the center of the target ring. These angles are crucial for assessing the UAV's alignment with the target, as they directly influence its trajectory toward the designated location. The second set of three components corresponds to the angular discrepancies between the UAV's orientation and the apex of the target ring, which is typically considered a reference point that defines the precise positioning within the target's vicinity.

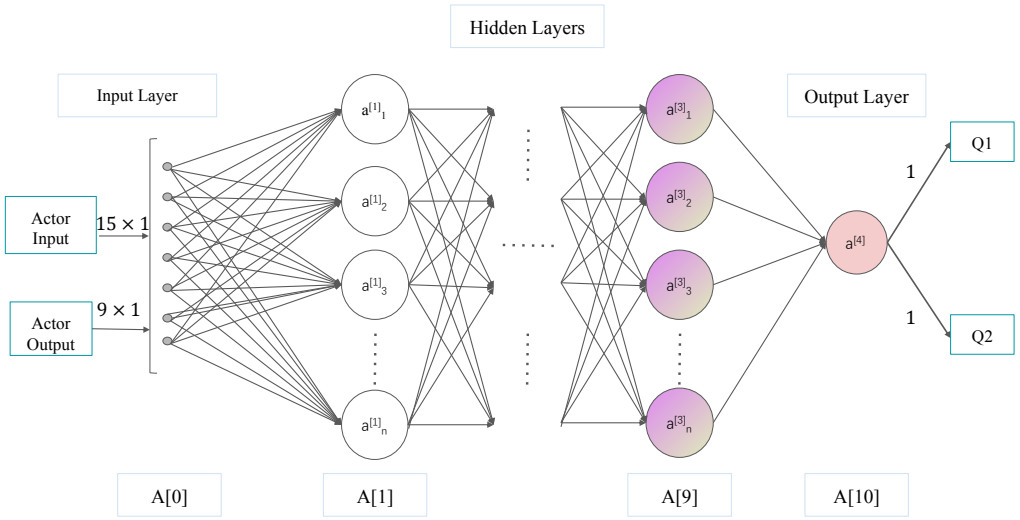
Conversely, the critic network consists of nine fully connected layers, each containing 256 nodes and also employing the tanh activation function. This network receives 24 vectors as input and produces 2 vectors as output. The inputs to the critic network encompass those fed into the actor network as well as the subsequent outputs from the actor network. The primary function of the critic network is to evaluate the value of the actions taken in the current state. As such, its inputs include the vectors from the current moment and the subsequent outputs, which consist of predictions for relative position, velocity, and acceleration. The critic network ultimately calculates two Q-values and consistently selects the smaller of these values.

3.2.3.4 Replay Buffer

The SAC approach in reinforcement learning incorporates a critical storage component known as the replay buffer, which plays a pivotal role in the architecture of many advanced deep reinforcement learning systems, providing several essential



(a) Actor Network.



(b) Critic Network.

Figure 3.7: Structure of the actor network and the critic network.

functions:

- Experience Reuse: The replay buffer archives transitions collected during the agent's interactions with the environment. These transitions, comprising tuples of state, action, reward, subsequent state, and completion status, are stored for later use. By randomly accessing these stored transitions, the replay buffer helps to decorrelate consecutive experiences, thereby enhancing the stability and efficiency of the learning process.

- Enhanced Learning Efficiency: The replay buffer significantly improves learning efficiency by allowing repeated utilization of past experiences to refine policy and value estimations. Without this mechanism, each experience would be used transiently and discarded, leading to suboptimal data utilization.

- Training Stabilization: By providing a diverse mix of past experiences, the replay buffer contributes to a more stable training regimen. This diversity prevents the model from overfitting to a limited range of recent inputs and promotes broader generalization across various environmental conditions.

- Learning from Diverse Policies: As an off-policy learner, SAC benefits from the ability to learn from actions derived from previous policy iterations. The replay buffer maintains a historical catalog of experiences, which expands the learning scope of the agent by exposing it to a variety of behaviors.

The replay buffer in the SAC framework is fundamental to its functionality. It enables experience replay, a mechanism through which the agent reuses past interactions to learn from them multiple times. Additionally, the buffer supports

off-policy learning, allowing SAC to learn from actions outside the current policy. Moreover, by decoupling the learning process from immediate experiences, SAC can sample from a large, diverse batch of experiences, reducing the variance of updates and smoothing the learning curve. This capability is crucial for stabilization in real-world applications where conditions vary unpredictably. Typically, the capacity of the replay buffer in SAC is set to 10^7, accommodating a vast repository of rewards, actions, states, and other relevant data, which fosters a nuanced understanding of the environment and supports the development of a sophisticated agent.

To optimize learning from this extensive dataset, the buffer is strategically divided into two segments. The first segment solely contains data about successful navigations, where the UAV adeptly maneuvers through obstacles to reach its target. The second segment stores all other data, including UAV collisions and standard flight data, which constitutes a substantial portion of the records. This segregation is vital as indiscriminately mixing all types of data could hinder initial learning stages due to the prevalence of collision and normal flight data. Data extraction for model updates follows a 0.2 to 0.8 ratio, ensuring a balanced learning process that encompasses both successful and challenging experiences. This structured approach to data management enhances the development of a robust and efficient agent.

3.2.3.5 Reward Function

The reward function is structured into multiple components to guide the UAV's navigation through narrow gaps. The variables $\boldsymbol{p}(t)$ and $\boldsymbol{p}_T(t)$ represent the UAV's position and the target position at time t, respectively.

$$r_p(t) = -\left\|\boldsymbol{p}(t) - \boldsymbol{p}_T(t)\right\|, \tag{3.15}$$

$$\begin{aligned} r(t) = \lambda_p r_p(t) + \lambda_a \left(r_a(t) + b_a\right) + \lambda_c r_c(t) + \\ \lambda_u r_u(t) + \begin{cases} r_T & \text{if target} \\ -1 & \text{otherwise} \end{cases}, \end{aligned} \tag{3.16}$$

The term $r_u(t)$ quantifies the duration of UAV flight, penalizing behaviors such as crashing, remaining stationary, or bypassing all obstacles to directly reach the final destination. Each of these actions results in a deduction of points. The term $r_c(t)$ indicates the number of circles the UAV successfully navigates through at time t, rewarding the UAV for each successful passage. The coefficients λ_p, λ_a, λ_u, and λ_c are hyperparameters that weigh the importance of various aspects of the UAV's performance within the reward function. Additionally, b_a is a positive offset applied to the relative attitude reward, helping to fine-tune the incentive structure for maintaining optimal orientations relative to obstacles. This formulation enables a comprehensive assessment of the UAV's performance, encouraging efficient, safe, and effective navigation through challenging environments by balancing penalties for undesirable actions with rewards for successful maneuvering.

3.2.4 Update Delay

In traditional actor-critic algorithms, updates to both the actor and critic networks occur at each step during the learning phase, typically facilitating more rapid pro-

gression and convergence. However, empirical observations indicate that this high frequency of modifications throughout ongoing training sessions can excessively alter the action selection strategy, resulting in instability and unpredictable policy behaviors.

To address these issues, a modified strategy has been implemented whereby updates to the networks are postponed until the conclusion of each training epoch. This method ensures that UAV flights are conducted under a consistent policy throughout the duration of the epoch, enhancing the stability of the training process and maintaining policy consistency. This adjustment is aimed at reducing fluctuations in policy behavior and improving the overall effectiveness of the learning algorithm.

Chapter 4

Results and Discussion

4.1 NEWDQN Algorithm

The proposed methodology has been validated through a series of UAV landing simulations conducted under various scenarios utilizing ROS and Gazebo.

4.1.1 Simulation Environment

In the domain of UAV autonomous systems, particularly focused on rescue and research operations, two simulation scenarios have been devised to assess the UAV's ability to navigate through narrow gaps. Table 4.1 presents a comparative analysis of the simulation scenarios. The first scenario, referred to as Case 1, involves the UAV navigating through a singular circular gap as delineated in the literature [55]. This simulation is conducted on the 'Gazebo' platform, designed with a straightforward environmental setup of one circle, aiming to streamline the model training process by allowing for detailed adjustments to the UAV's navigation algorithms. During this scenario, the UAV utilizes visual data from an onboard camera to navi-

gate the gap, executing a direct 30-degree turn without any translational movement at a maximum speed of 2 meters per second. Impressively, this scenario boasts a success rate of 100% over 1000 trials, demonstrating the UAV's effectiveness in managing simple navigational tasks.

Conversely, Case 2 introduces a heightened level of complexity by requiring the UAV to navigate through dual circular gaps, also referenced in the literature [55]. While the application context remains consistent with Case 1, the presence of two circles increases the environmental complexity. Despite the added challenges, this scenario maintains a focus on refining the UAV's control mechanisms. The performance metrics from this more demanding simulation indicate a success rate of approximately 70% over 1600 trials, highlighting areas where the UAV's navigational algorithms require further enhancements to manage multiple obstacles effectively.

Together, these simulations play a crucial role in providing empirical data that informs the continuous refinement of UAV control algorithms. By systematically varying the complexity of the navigation tasks, the simulations help delineate the capabilities and limitations of current UAV technologies in realistic, controlled environments. The insights gleaned are vital for advancing UAV capabilities, ensuring the technology's effective deployment in real-world applications where precision and adaptability are paramount.

In the comprehensive analysis of UAV navigation within constrained settings, spatial parameters are pivotal in elucidating and evaluating the efficacy of navigational strategies. As depicted in Fig. 4.1, the initial spatial separation between the UAV and the singular narrow gap in the first scenario is quantitatively established at 10 meters. This measurement provides a baseline for evaluating the UAV's ap-

Table 4.1: Comparative analysis of simulation scenarios

Parameters	Case 1: Single Gap Navigation	Case 2: Dual Gap Navigation
Assumption	Navigation through a singular circular gap.	Navigation through dual circular gaps.
Application Context	Primarily tailored for rescue and research operations.	Primarily tailored for rescue and research operations.
Environmental Complexity	Features a singular circular gap.	Comprises two distinct circular gaps.
Simulation Platform	Gazebo	Gazebo
The Rationale for Multiple Cases	To simplify the model training process and facilitate granular adjustments.	To simplify the model training process and facilitate granular adjustments.
Inputs	Visual data.	Visual data.
Environmental Complexity	Features a singular circular gap.	Comprises two distinct circular gaps.
Expected Output	A secure trajectory.	A secure trajectory.
Success Rate	A commendable 100%.	Approximately 70%, indicating the increased complexity of the dual gap scenario.
Number of Trials	1000	1600
Environmental Complexity	Features a singular circular gap.	Comprises two distinct circular gaps.

proach and maneuvering capabilities in a controlled setting.

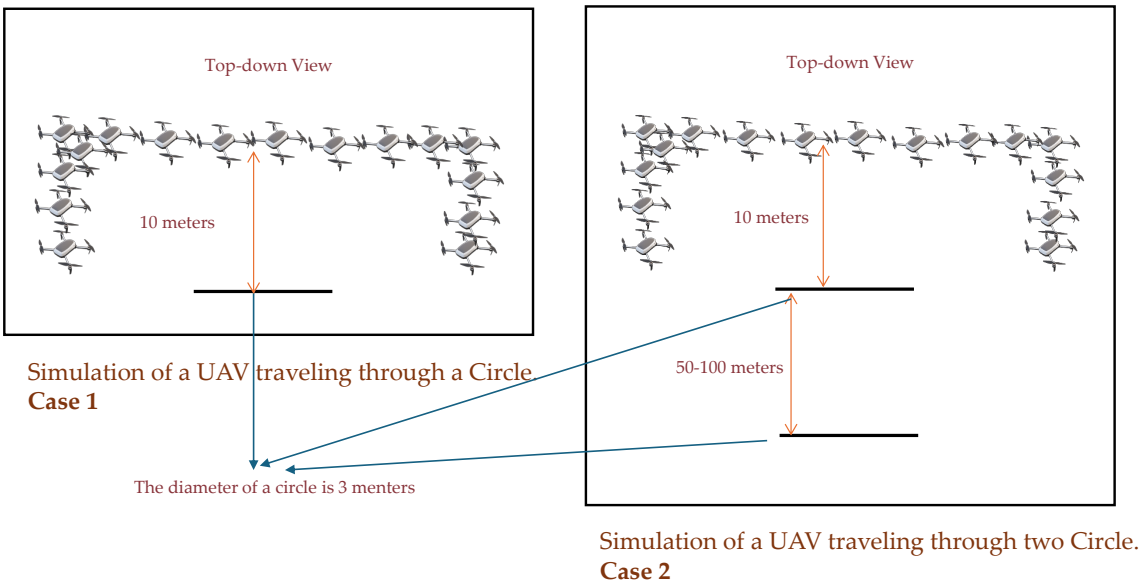


Figure 4.1: View of environment.

Extending this analysis to a more complex configuration, the initial separation between the UAV and the foremost narrow gap in the subsequent scenario is similarly quantified at 10 meters. However, the complexity increases with the introduction of an additional gap, where the interstitial distance between the two narrow gaps varies from 50 to 100 meters. This variable range introduces a significant challenge in path planning and execution, necessitating precise control and advanced navigational algorithms.

Fig. 4.2 offers a visual depiction of the field of view accessible via the on-

board camera of the UAV. This figure is crucial as it illustrates the UAV's visual range, which directly influences its ability to detect and respond to environmental features and obstacles. The observable field of view is a critical factor in the UAV's operational efficiency, especially in scenarios involving multiple obstacles or gaps, as it determines the extent of the environment that can be assessed and navigated at any given moment.

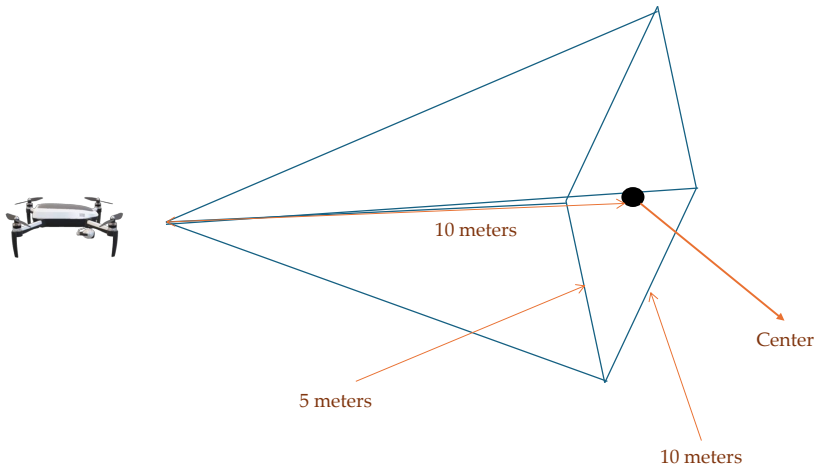


Figure 4.2: View of camera.

Together, these figures and measurements provide a comprehensive framework for analyzing the UAV's performance under varying degrees of environmental complexity. They also serve as fundamental inputs for refining UAV control algorithms, enhancing the UAV's adaptability and efficacy in real-world applications that require precise navigation through narrow and dynamically changing

spaces.

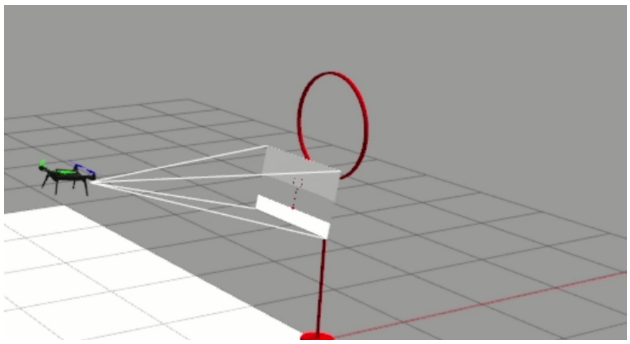
4.1.2 One Circular Crossing Algorithm

Fig. 4.3 presents a graphical representation of a simulated navigation exercise where an UAV maneuvers through a circular gap. The success of this simulation is highlighted by the algorithm's flawless performance, which achieves a 100% success rate in these trials. This impeccable result is primarily attributed to the algorithm's sophisticated capability to precisely identify the center of the circular gap. By accurately determining this central point, the UAV is able to adjust its altitude correspondingly, allowing for optimal alignment. This precise alignment facilitates the UAV's seamless navigation through the gap, demonstrating the algorithm's effectiveness in handling complex navigational tasks within constrained environments. The high success rate in these simulations underscores the algorithm's potential for real-world application, where precise and reliable UAV navigation is critical.

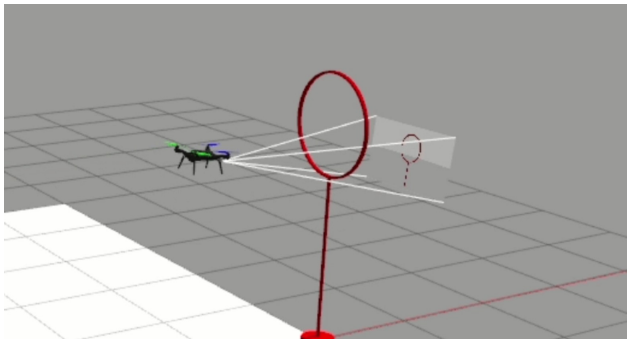
4.1.3 Two Circular Crossing Algorithm

Fig. 4.4 shows the simulation of a UAV through two circulars.

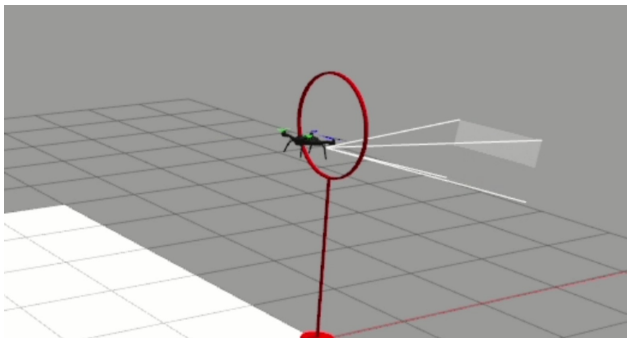
In Fig. 4.5, the cumulative reward curve is illustrated, providing a quantifiable measure of the proposed algorithm's performance over time. A detailed analysis of this curve reveals that the algorithm reaches a state of convergence after approximately 1,000 iterations. This convergence indicates that the algorithm has effectively learned the optimal strategy for the task at hand, as evidenced by the stabilization of the reward accumulation.



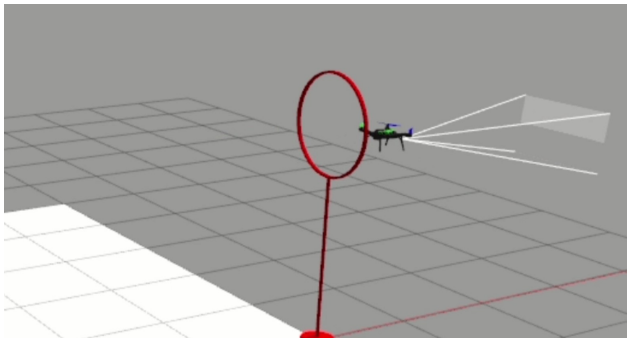
(a)



(b)

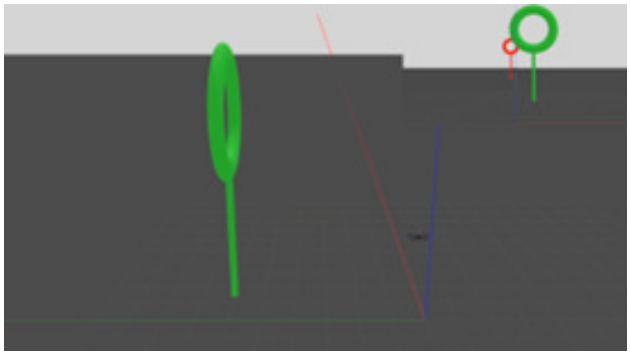


(c)

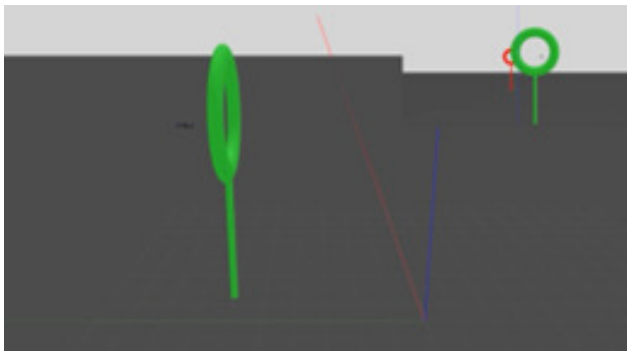


(d)

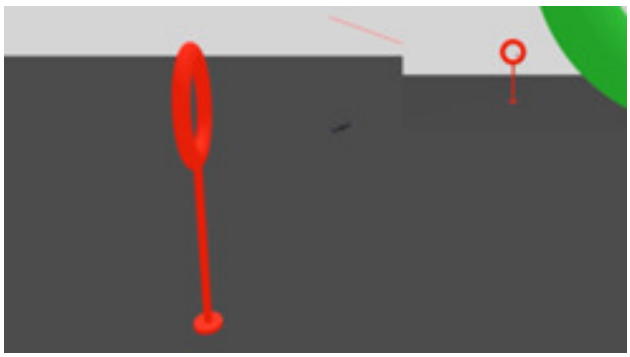
Figure 4.3: Simulation of a UAV traveling through a circulars.



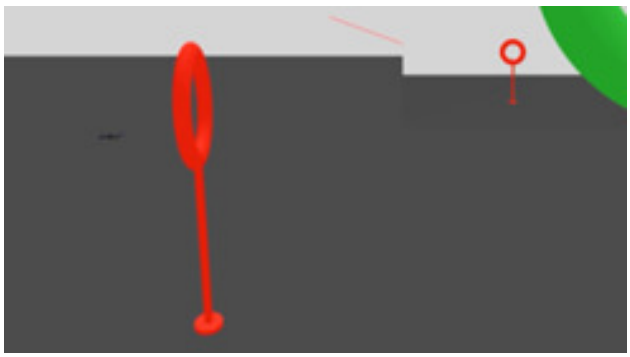
(a)



(b)



(c)



(d)

Figure 4.4: Simulation of a UAV traveling through two circulars.

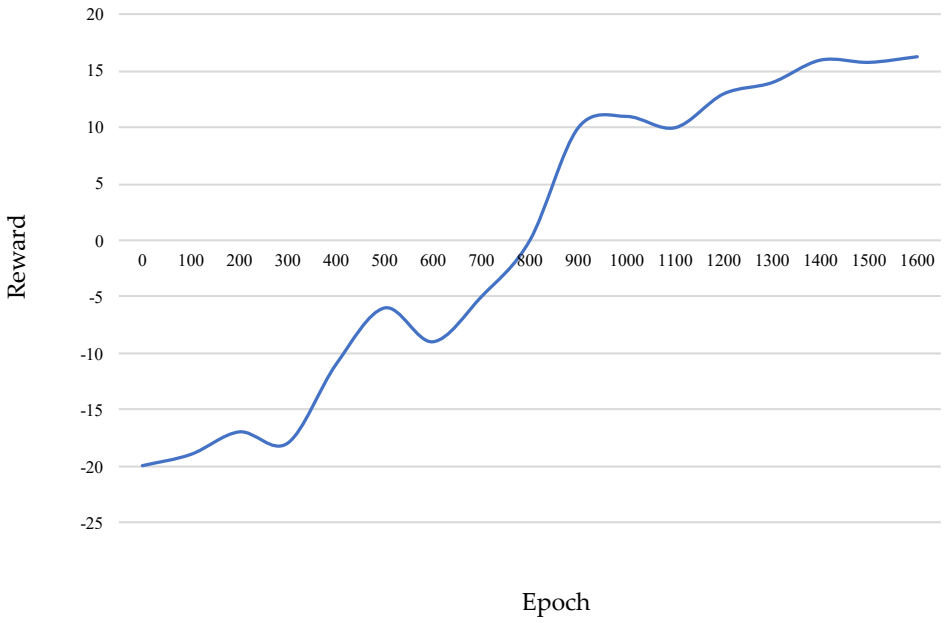


Figure 4.5: Reward function.

Concurrently, Fig. 4.6 displays the success trajectory of the UAV throughout its training phase. This trajectory graphically represents the progression of the UAV's mission success rate over time, expressed quantitatively as a percentage. The increasing trend in this trajectory highlights the effectiveness of the training process, demonstrating a consistent improvement in the UAV's ability to successfully complete its designated tasks.

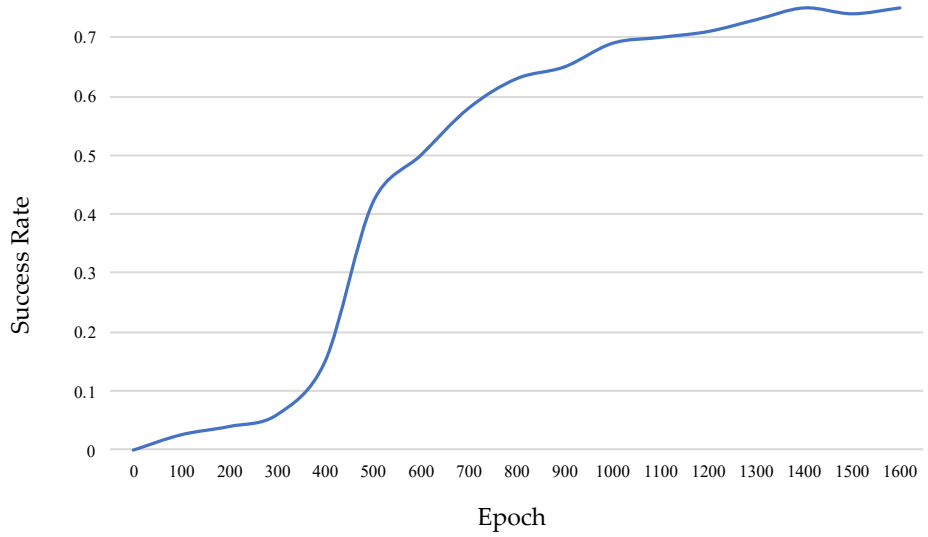


Figure 4.6: Success rate.

4.2 SAC Algorithm

4.2.1 Experiment Results

This section assesses the efficacy of the SAC algorithm in UAV path planning, examining its performance across both simulated environments and actual field applications. The experiments were designed to assess the SAC's efficiency, robustness, and practical utility, aligning with our research objectives to enhance UAV navigational capabilities in dynamic settings. The simulation experiments were executed within an advanced simulation framework meticulously developed

using the Open Dynamics Engine, integrated with Gazebo and the Python-based PyTorch machine learning framework. This configuration enabled a high-fidelity simulation of real-world dynamics, permitting precise control and observation of the UAV's behavior in scenarios involving obstacle avoidance. In this controlled virtual setting, the UAVs were required to navigate through dynamically generated obstacles, with systematic adjustments made to variables such as obstacle density, UAV speed, and response time. The aim was to evaluate the UAVs' capacity to adjust their path planning strategies in response to environmental changes, employing the soft actor-critic algorithm. These simulation trials generated a comprehensive dataset on performance metrics, encompassing navigation precision, collision frequencies, and computational efficiency.

Following the simulation trials, laboratory experiments were conducted to corroborate the findings. These experiments took place in a controlled indoor setting using actual UAVs outfitted with depth cameras and onboard processing units. The laboratory environment facilitated the recreation of the simulation scenarios, providing a platform to assess the real-world effectiveness of the algorithms. The UAVs were subjected to stringent tests in conditions that emulated the complexities observed in the simulated environment, such as obstacle density and unpredictability. Performance metrics analogous to those measured in the simulation experiments were evaluated, with a focus on the UAVs' proficiency in executing the learned navigation strategies. Comparative analysis between the simulated and laboratory results yielded insights into the practical challenges and necessary adaptations for transitioning the algorithms from simulated to real-world applications.

4.2.2 Simulation

In the domain of computer systems and reinforcement learning, the efficacy and success of algorithms are critically dependent on the configuration of various parameters. Optimal parameter settings enhance the utilization of computational resources and precisely calibrate the learning mechanisms of the models employed. This document offers an in-depth overview of key parameters that are crucial in formulating computer configurations and refining reinforcement learning algorithms. These parameters are demonstrated within a simulated environment, as detailed in the accompanying Table 4.2.

Table 4.2: Simulation environment setting

Operating System	Ubuntu 18.04
CPU	Intel(R) Core(TM) i7-12700F
GPU	NVIDIA RTX 3070
RAM	48 GB
CV Library	Gazebo9
Program Language	Python 3.6
ML Library	OpencV 4.4
Simulator	Open Dynamics Engine
Optimizer	Adam
Replay Buffer Size	10^7
Learning Rate	10^{-3}
Batch Size	256
Reward Scale	0.99
Update Times	20
UAV Weight	1203g

In this experiment, an enclosed area was designed measuring 20 meters in all dimensions—length, width, and height—using the Open Dynamics Engine to simulate a flight environment for UAVs. Within this constructed space, twenty

circular hoops were strategically placed, as illustrated in Fig. 4.7. Each occurrence of a UAV successfully navigating through these hoops during a test was recorded as a progression. It was hypothesized that a UAV initiating its flight at one end of the environment and effectively maneuvering through the obstacles to transit through the circles would be considered to have successfully completed the test.

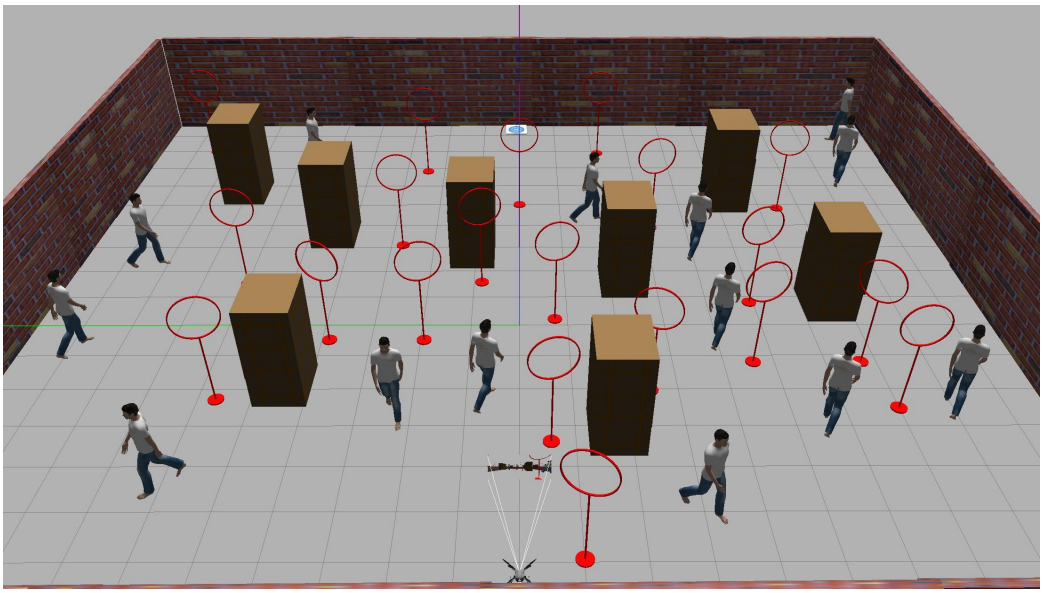


Figure 4.7: Simulation environment.

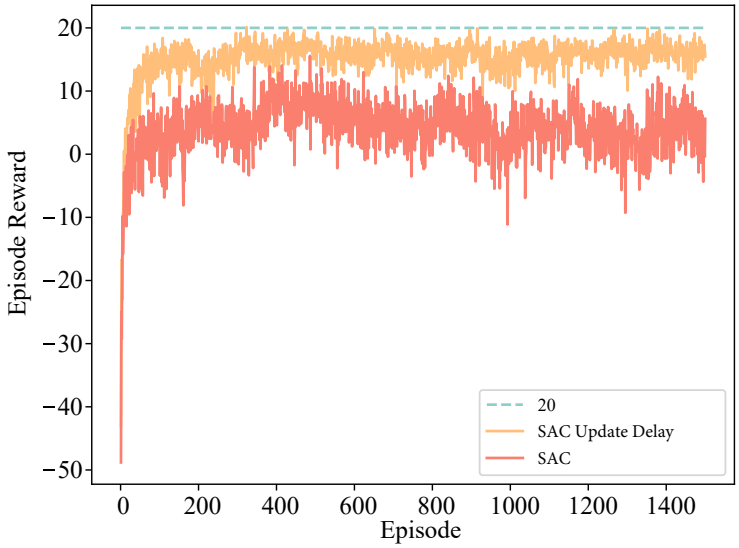
This experiment aimed to assess the SAC network's effectiveness in training UAVs for obstacle navigation. The reward function encouraged UAVs to collect rewards by flying through hoops and avoiding collisions. The total rewards earned during each epoch, whether the UAV reached the destination or crashed, were recorded as the epoch reward. By changing the UAV's starting point every 10 epochs, we built a diverse dataset that captured the UAV's performance under various conditions.

During the testing phase, the UAV commenced each trial from a consistent

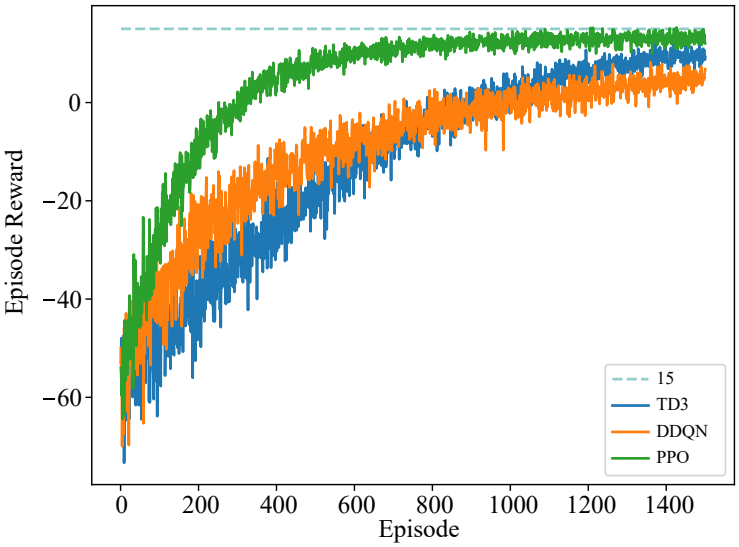
starting position and attempted to navigate through a series of hoops while circumventing obstacles. Success was delineated as the UAV's ability to traverse all hoops and arrive at the final destination unscathed. The intermittent repositioning of the hoops assessed the UAV's adaptability to variations in the environment. The model that demonstrated the highest average success rate across diverse scenarios was chosen for further examination. Subsequently, the UAV, governed by this model, underwent testing in various environments across more than 1,000 epochs. The SAC demonstrated high success rates in navigating through complex simulated environments. In scenarios with positioned obstacles, the highest navigation accuracy success rate that SAC achieved in different scenarios was 90.5%. These findings highlight the SAC algorithm's capacity for adapting to fluctuating conditions, representing a significant advancement for real-time UAV applications in areas such as search and rescue or urban surveillance.

To evaluate the efficacy of the update-delayed version of the SAC algorithm compared to the standard SAC, both variants were subjected to uniform conditions across 1,500 epochs within the same training environment, as depicted in Fig. 4.8(a). The observations indicated that the delayed update strategy not only enhances training stability but also facilitates faster convergence and produces superior reward outcomes relative to the conventional actor-critic method. Fig. 4.8(b) illustrates the outcomes from training various reinforcement learning algorithms in a simulated setting. In our analysis, network updates were implemented following each epoch. Notably, episodic rewards began to incrementally increase after approximately 50 epochs, suggesting that with increased training duration, the magnitude of the rewards also escalates.

To ascertain the superiority of our model, comparative training sessions, each



(a) Both the standard SAC and its update-delayed version were subjected to 1,500 epochs of training for the specified task.



(b) Comparison of 3 RL methods.

Figure 4.8: RL function.

consisting of 1,500 epochs, were conducted between our model and other prominent algorithms—TD3, DDQN, and PPO—within the same experimental framework, focusing on the comparison of reward outcomes. The PPO algorithm excelled by rapidly converging towards the optimal reward value function and adapting effectively to the environmental variables. Initially, the DDQN outperformed the TD3, but subsequently, its performance waned. This variability in performance could be linked to DDQN's initial rapid adaptation, whereas PPO demonstrated sustained efficacy over extended training periods. Among the tested reinforcement learning techniques, the SAC with a delayed learning update approach yielded the highest reward, registering a score of 20. This outcome suggests that incorporating a delay in learning updates can significantly amplify the performance and stability of the SAC model, rendering it highly effective in complex training scenarios where long-term strategic consistency is vital. Fig. 4.8(a) and Fig. 4.8(b) elucidate that in our specific task, the enhanced SAC algorithm markedly surpassed the competing algorithms in terms of reward accumulation rate and peak reward values.

With merely 200 updates, our algorithm exhibited notably commendable performance across both unaltered and altered environments, suggesting that our model effectively addresses the commonly slow convergence issues associated with policy-based DRL algorithms in the context of UAV visual obstacle navigation. Comparing SAC to traditional methods like TD3 and DDQN, SAC consistently offered better performance and quicker adaptation to environmental changes. This comparative analysis not only validates the SAC's superior efficacy but also demonstrates its potential to replace more conventional approaches in advanced UAV path planning tasks. These experimental configurations not only aid in assessing

UAV performance across diverse navigational challenges but also play a pivotal role in advancing algorithmic development for autonomous flight and obstacle negotiation within cluttered settings.

The observed fluctuation in the image data can be attributed to the inherent randomness in the initial conditions and hyperparameter configurations within the training environment. Specifically, the UAV's initial position is randomly initialized at the onset of training, leading to variations in its starting point relative to the environment's spatial layout. This variability, in combination with the stochastic nature of several hyperparameters governing the DRL model such as learning rate, exploration factor, and reward scaling. These random initializations and hyperparameter choices result in a dynamic learning process where the UAV's navigation and decision-making are influenced by varying conditions at each iteration of training. Consequently, the reward function, which serves as a feedback mechanism guiding the UAV's learning process, experiences corresponding fluctuations as the model adapts to different scenarios and conditions. These variations frequently appear in reinforcement learning, especially under complex and changing conditions, reflecting the model's continuous pursuit of optimal actions throughout the state-action space.

4.2.3 Real Environment

Subsequent to the simulated experiments, the UAVs were subjected to real-world testing to evaluate the transferability of their acquired navigation and obstacle avoidance skills. These tests are essential for determining the extent to which training in simulated environments is applicable to real-world settings that may

not be accurately modeled or controlled. The laboratory experiments were specifically designed to validate the efficacy of the SAC algorithm in managing UAV obstacle avoidance under actual conditions. These evaluations were conducted in an indoor flight testing facility outfitted with a range of physical obstacles, such as hoops, barriers, and dynamically moving challenges, to closely replicate the conditions encountered in the simulations. Transitioning to real-world environments, the SAC maintained a high performance, achieving the lowest obstacle avoidance success rate of approximately 68.0% in different environments. The lowest success rate of 68% observed in this study across various test settings represents a notable advancement compared to the success rates of approximately 40% reported in other studies [70]. This slightly lower rate compared to simulation highlights challenges like sensor accuracy and environmental unpredictability, which are not fully replicated in simulations. The results demonstrate the feasibility of deploying UAVs trained with the SAC algorithm in practical scenarios where adaptability and reliability are paramount.

The configuration of the UAV utilized in our real-world testing environment is illustrated in Fig. 4.9. This UAV was equipped with an Intel RealSense D435, which features both a depth camera and a stereo camera, as well as an Intel Tiger Canyon computer for high-performance onboard processing and a Holybro Pixhawk4 Mini flight controller for precision navigation in the controlled laboratory, ensuring that it could accurately interpret and respond to its surroundings in real-time. Once the two-dimensional position of the target is determined using the image obtained from the stereo camera, the vertical distance between the UAV and the target is subsequently calculated with the aid of the depth camera. The stereo camera provides critical information regarding the target's relative position

within the horizontal plane, while the depth camera, which measures the disparity in the depth map, accurately estimates the vertical distance. By combining these data sources, the system can effectively compute the complete three-dimensional spatial relationship between the UAV and the target, thereby enabling more precise navigation and obstacle avoidance in dynamic environments. The experimental framework involved a series of meticulously designed flight paths that the UAV was required to traverse. These paths incorporated strategically placed unexpected obstacles to rigorously evaluate the UAV's agility, situational awareness, and adaptability in learning and maneuvering. This setup aimed to simulate dynamic environmental conditions, thereby enabling us to comprehensively assess the UAV's obstacle avoidance and navigation capabilities under challenging and unpredictable circumstances.

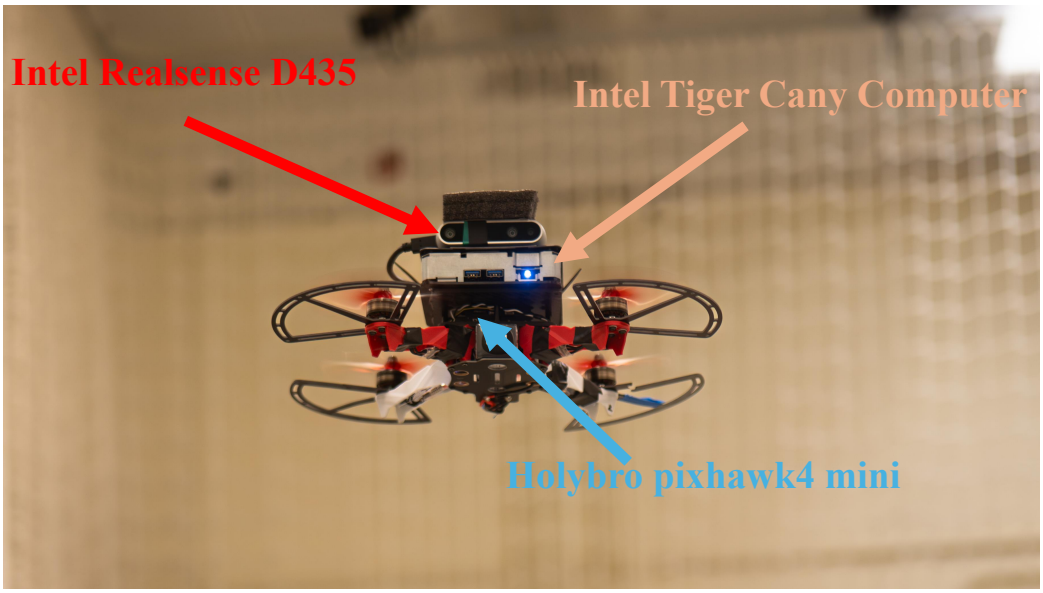


Figure 4.9: Our flying platform.

In preliminary tests, the UAVs exhibited a notably low obstacle avoidance

success rate. However, over time, through repeated trials, the avoidance accuracy across three specified circles improved to 70.4%, as the UAVs adapted to the environment via continuous learning facilitated by the SAC algorithm. UAV exhibited the notably lowest success rate in obstacle avoidance, approximately 68%, when navigating through three challenging scenarios involving two angled circles. Additionally, the onboard computational demand was rigorously monitored. Initial experimental trials revealed that up to 70% of the UAVs' processing capacity was utilized, primarily attributable to the substantial computational demands of the SAC algorithm's actor-critic network. Through targeted optimization of the network parameters, including adjustments to the learning rates and exploration-exploitation balance, the computational load was effectively reduced to approximately 55%. This reduction not only enhanced the operational efficiency of the UAVs but also ensured that the responsiveness of the system to environmental variables remained uncompromised.

In an investigation of UAV navigation capabilities across six distinct environments, various configurations of rings and obstacles were systematically established to evaluate navigational accuracy and obstacle avoidance techniques. From Fig. 4.10 to Fig. 4.15 shows the realistic environment configuration. The configuration details are as follows:

1) Fig. 4.10 illustrates the configuration of Environment 1: A single ring with a diameter of 0.70 m is strategically placed. Positioned 2.90 m from the center of the ring at a vertical distance of 1.64 m, the UAV measures 0.21 m in radius and 0.13 m in height. Additionally, two obstacles are incorporated, including a brown obstacle located 1.40 m from the UAV, which partially

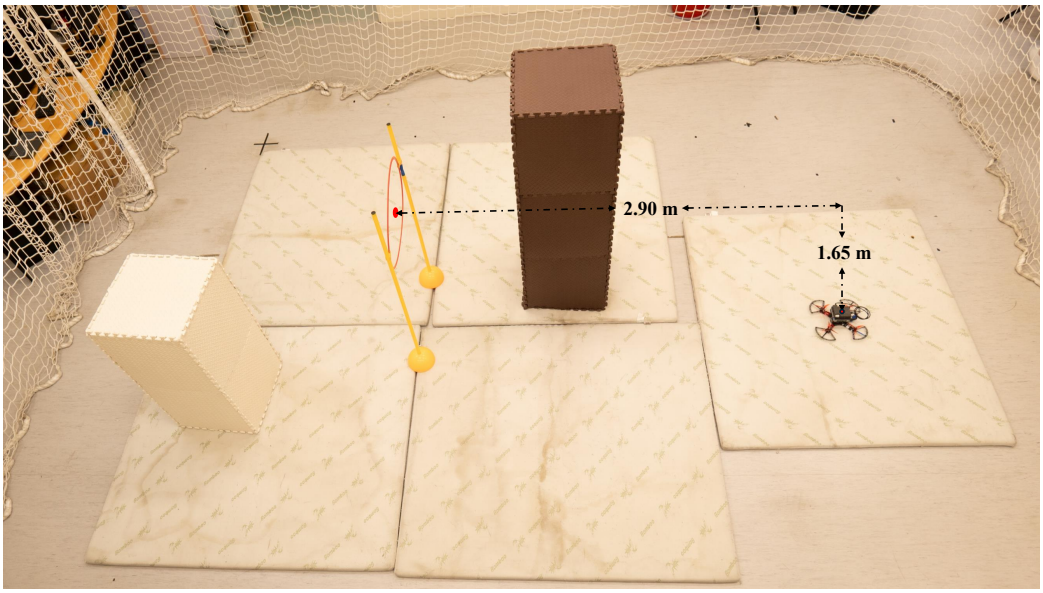


Figure 4.10: Realistic environment configuration: 1 circle.

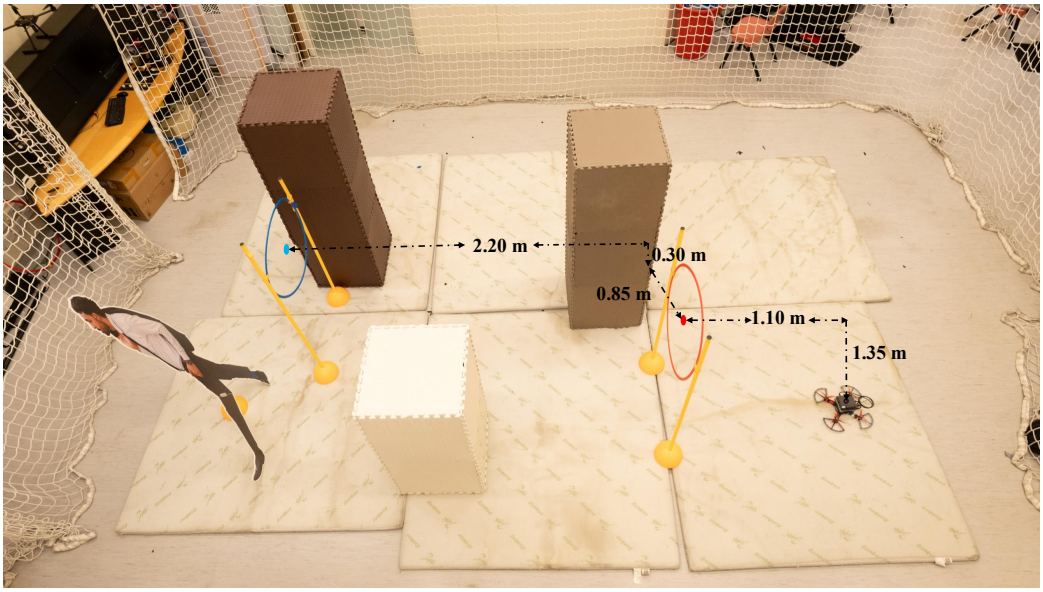


Figure 4.11: Realistic environment configuration: 2 circle.

obscures the ring by one-third, thereby posing a challenge to the UAV's visual navigation systems.

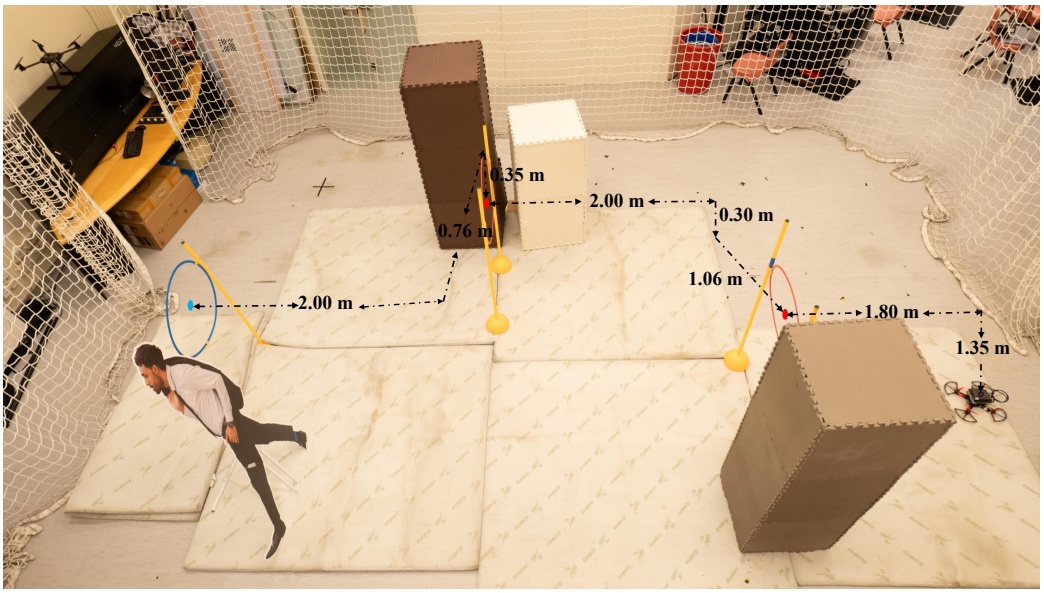


Figure 4.12: Realistic environment configuration: 3 circle.

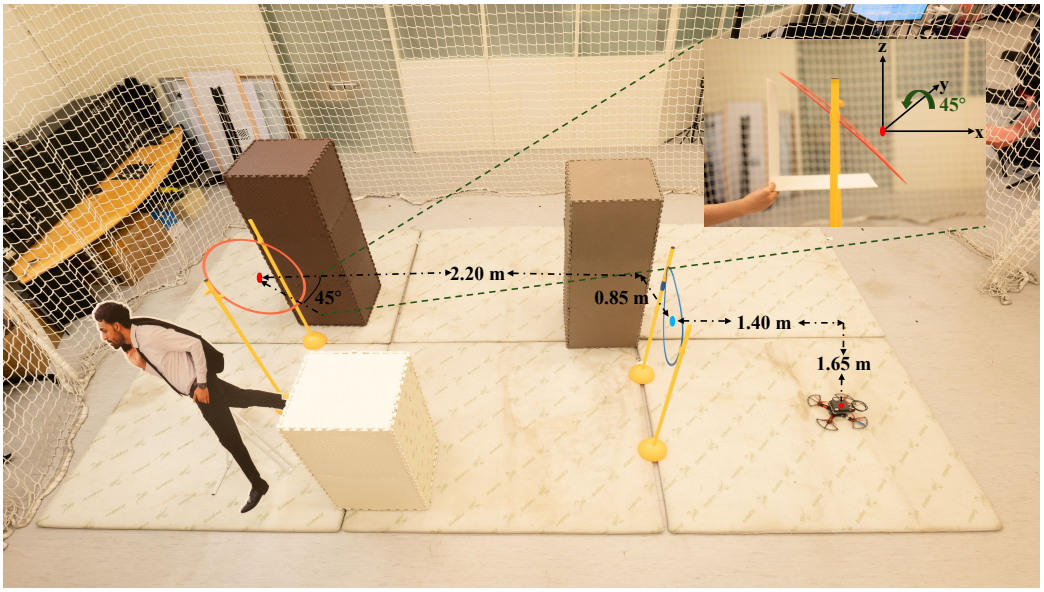


Figure 4.13: Realistic environment configuration: 2 circles (rotate around Y-axis).

2) Fig. 4.11 depicts the configuration of Environment 2, which includes two rings. The UAV is stationed 1.10 m from the first ring's center at a height of

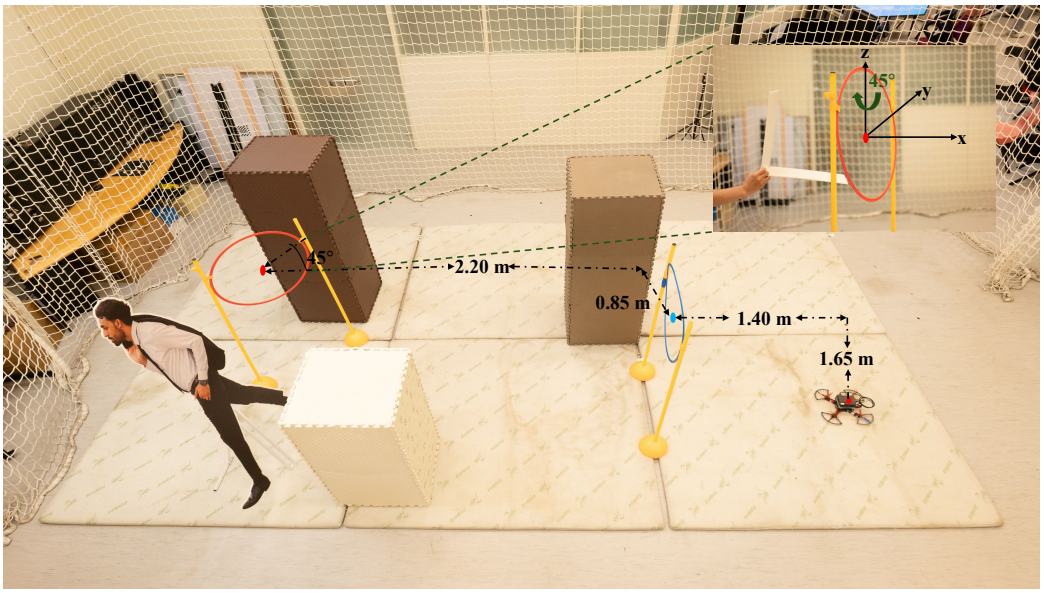


Figure 4.14: Realistic environment configuration: 2 circles (rotate around Z-axis).

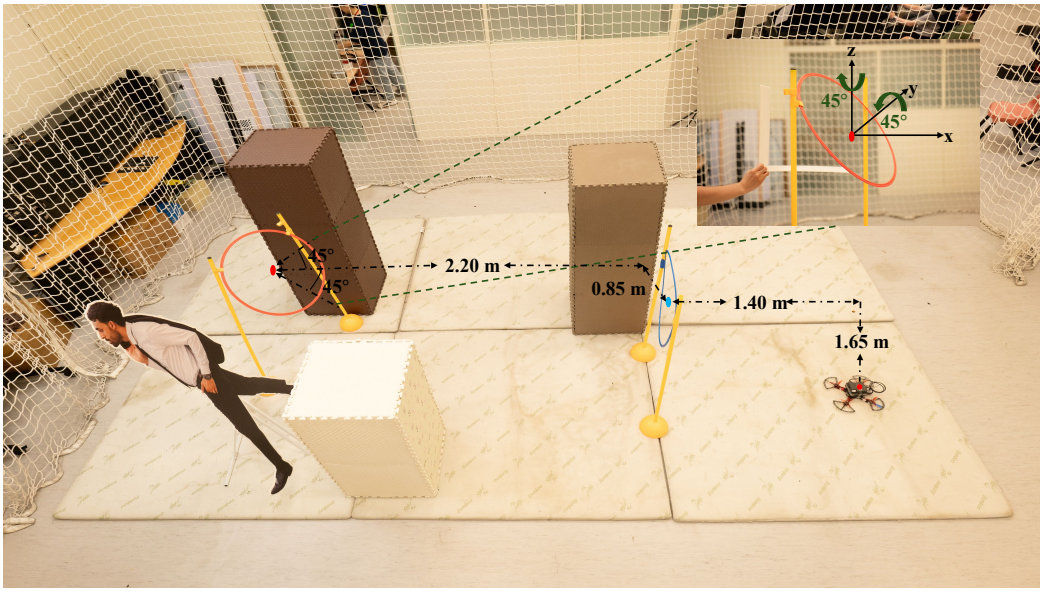


Figure 4.15: Realistic environment configuration: 2 circles (rotate around Y-Z-axis).

1.35 m. The horizontal separation between the rings is 0.85 m, with the first ring positioned 0.30 m lower than the second. The linear distance between the centers of the rings measures 2.20 m. Four obstructions are strategically placed, adding complexity to the UAV's path-finding algorithms and requiring sophisticated maneuvering capabilities.

3) Fig. 4.12 presents the configuration of Environment 3, where three rings are utilized in this setup. The UAV's initial position is 1.80 m horizontally and 1.35 m vertically from the first ring. The subsequent rings are arranged with increasing complexity; the second ring is 1.06 m horizontally from the first and 0.30 m lower in elevation, and the third ring, 0.76 m away horizontally from the second, is 0.35 m lower. The linear distances between the rings remain consistent at 2.00 m. Similar to Environment 2, four obstructions are placed to test the UAV's adaptive response to dynamic spatial challenges.

4) Fig. 4.13 shows environment 4 configuration: This environment is structured similarly to Environment 2, with adjustments made to the orientation of the second ring. While the two rings in Environment 2 are aligned parallel, here, the second ring undergoes a rotation of 45 degrees clockwise around the Y-axis. The initial positioning of the UAV is 1.40 m from the center of the first ring at an altitude of 1.60 m. The horizontal and linear spacing between the centers of the rings remains consistent at 2.20 m. Additionally, four obstacles are strategically placed within this setup to assess the UAV's adaptability and response to altered spatial dynamics.

5) Fig. 4.14 shows environment 5 configuration: This scenario mirrors Environment 4 with a singular variation in the angular disposition of the second

ring. Contrary to Environment 2, in which the rings are parallel, the modification here involves a 45-degree clockwise rotation of the second ring around the Z-axis. All other spatial arrangements, including the positioning of the UAV and the placement of obstacles, are identical to those in Environment 4, facilitating comparative analysis of navigational responses under subtly varied rotational adjustments.

6) Fig. 4.15 shows environment 6 configuration: Similar to Environment 4, this setup introduces a further complex rotational adjustment to the second ring. The modification encompasses a two-stage rotation: initially 45 degrees clockwise around the Y-axis followed by an additional 45 degrees clockwise around the Z-axis. This scenario distinguishes itself from Environment 2 by presenting a compounded rotational challenge while maintaining the same spatial configuration as in Environment 4. This environment is designed to test the UAV's capacity for navigating through increasingly complex angular alterations while managing the same spatial constraints and obstacle configurations.

Table 4.3 presents experimental results comparing success rates of UAV navigation in various configurations both in simulation settings and real-world applications. In a single circle environment, the UAV achieved a success rate of 90.5% in simulation and 80.4% in real-world scenarios, demonstrating a decline when transitioning from controlled to complex real environments. For two and three circles, success rates in simulations are 85.9% and 80.3%, respectively, decreasing in real-world settings to 73.8% and 70.4%. This highlights the increasing challenge posed by additional obstacles. Environments with two circles rotated around the Y-axis

Table 4.3: Experimental results in success rate

Environment	Simulation	Real World
1 circle	90.5 %	80.4 %
2 circles	85.9 %	73.8 %
3 circles	80.3 %	70.4 %
2 circles (rotate around Y-axis)	85.0 %	71.0 %
2 circles (rotate around Z-axis)	84.9 %	70.0 %
2 circles (rotate around Y-Z-axis)	80.5 %	68.0 %

and Z-axis, and both Y and Z axes show varied results. Success rates in simulations are 85.0%, 84.9%, and 80.5%, respectively, while in real-world scenarios, these figures reduce to 71.0%, 70.0%, and 68.0%, indicating a pattern where rotational complexities affect UAV navigation effectiveness.

Fig. 4.16 to Fig. 4.21 illustrate the trajectory of the UAV navigating through six progressively challenging environments.

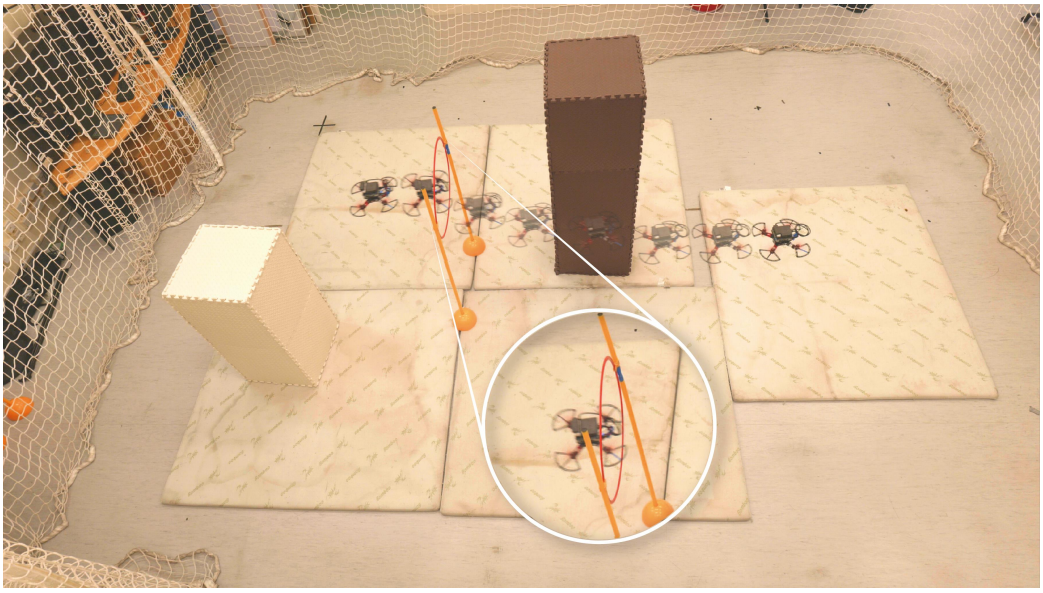


Figure 4.16: Through the 1 circle.

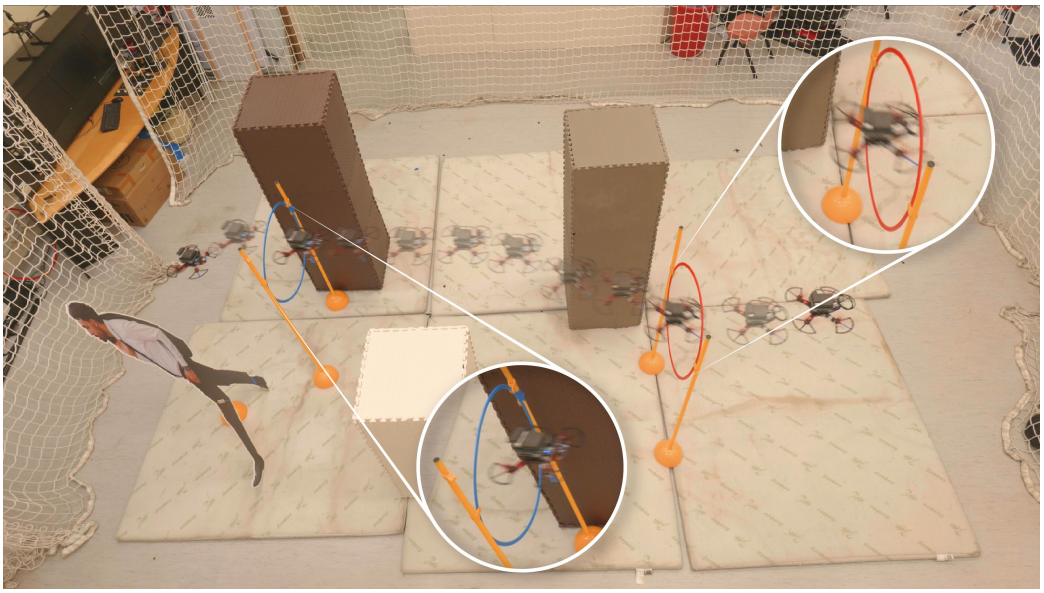


Figure 4.17: Through the 2 circles.

Figure 4.18: Through the 3 circles.

Fig. 4.16 reveals the flight route as the UAV passes through a single ring, offering a relatively straightforward task that tests basic navigation and stability.

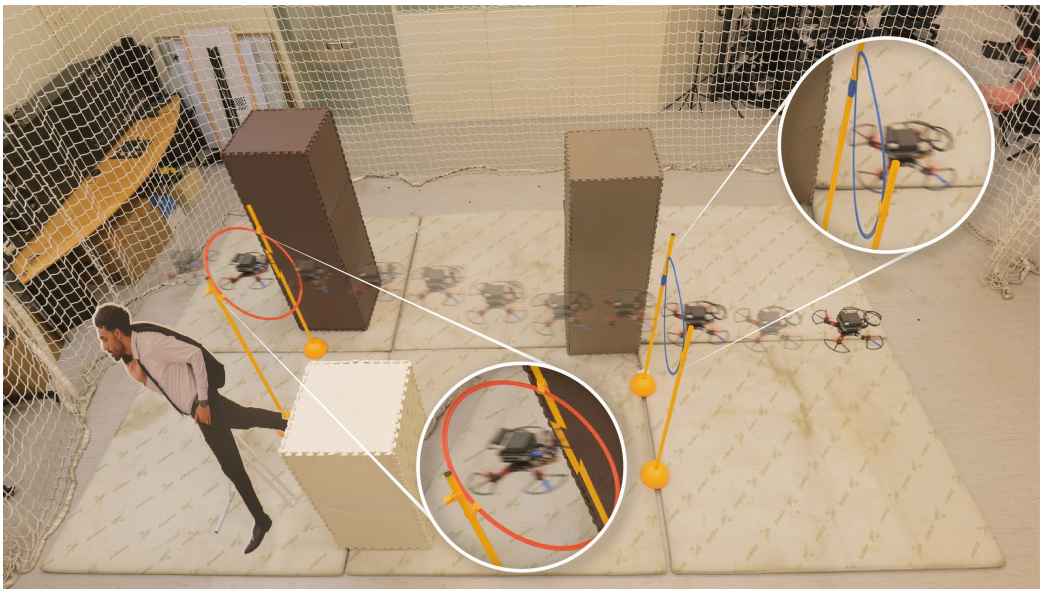


Figure 4.19: Through the 2 circles (rotate around Y-axis).

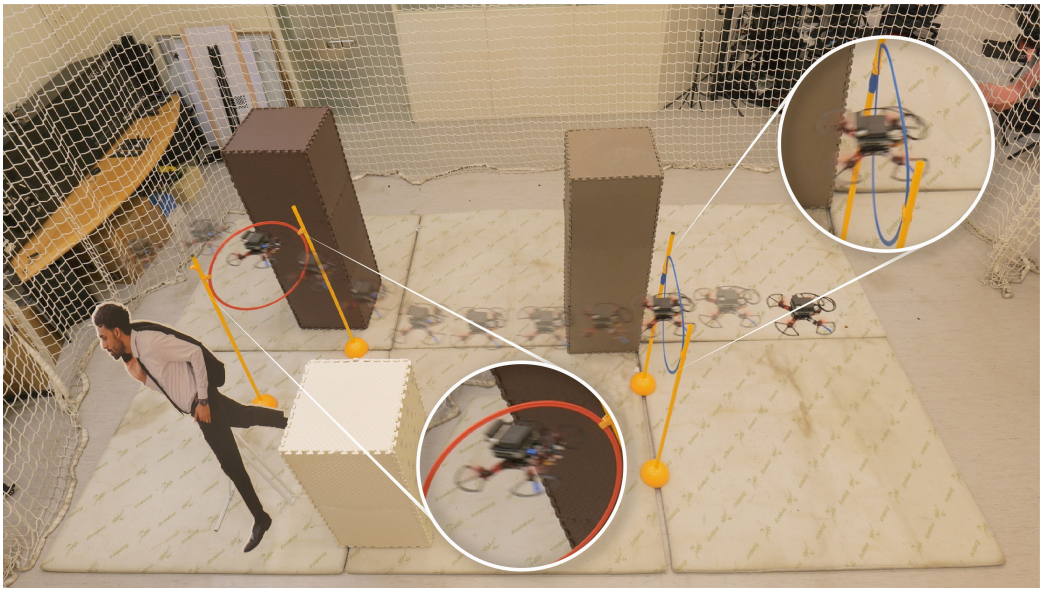


Figure 4.20: Through the 2 circles (rotate around Z-axis).

Fig. 4.17 shows the trajectory when the UAV encounters two rings, requiring it to swiftly adjust its orientation and speed between consecutive targets. Environ-

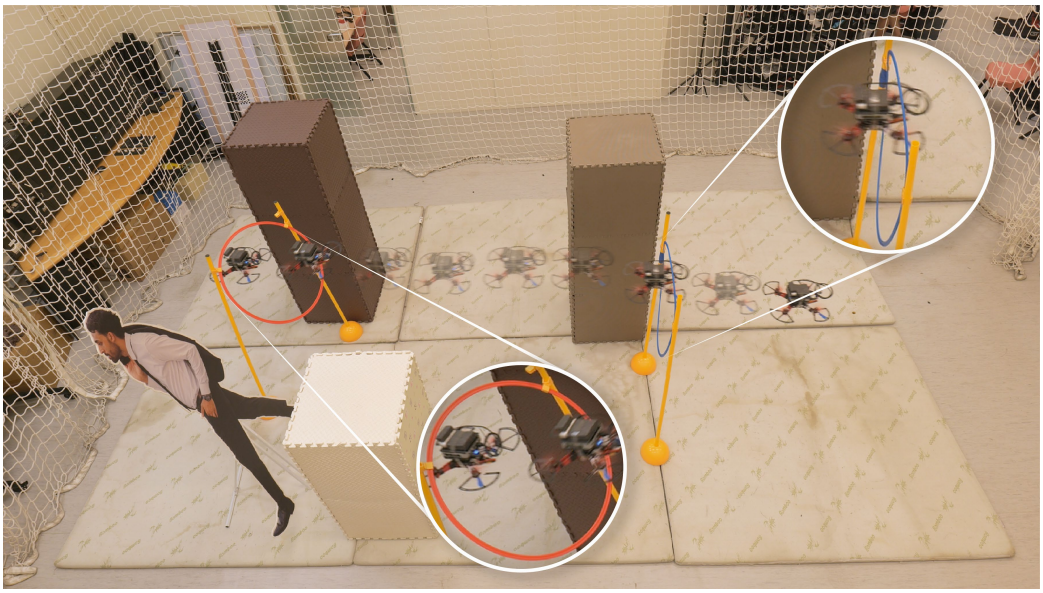


Figure 4.21: Through the 2 circles (rotate around Y-Z-axis).

ment 2 serves as a baseline, which tests the UAV's basic path-finding algorithms and its ability to maneuver through straightforward trajectories with minimal orientation changes. The relatively simpler configuration of this environment allows the UAV to demonstrate core competencies in stable flight dynamics and obstacle avoidance. Fig. 4.18 demonstrates the UAV's ability to maneuver through a sequence of three rings, necessitating continuous orientation adjustments and speed control while maintaining accurate alignment. Environment 2 serves as a baseline, where the UAV navigates through two parallel rings. This scenario tests the UAV's basic path-finding algorithms and its ability to maneuver through straightforward trajectories with minimal orientation changes. The relatively simpler configuration of this environment allows the UAV to demonstrate core competencies in stable flight dynamics and obstacle avoidance. Fig. 4.19 demonstrates a scenario wherein the second ring is rotated by 45 degrees around the Y-axis, presenting a

challenge to the UAV's capacity to adjust its trajectory in response to rotational distortions. This environment tests the UAV's sensor integration and data processing efficacy, requiring a sophisticated understanding of altered geometrical perspectives. The adaptation to this environment suggests a higher level of spatial awareness and computational adaptability, which are crucial for navigating through environments with irregular object orientations. Fig. 4.20 further complicates the navigation challenge by rotating the second ring 45 degrees around the Z-axis. This rotation necessitates an adjustment in the UAV's lateral and vertical control mechanisms, pushing the boundaries of its control algorithms to maintain precision in a dynamically altered navigational context. The successful navigation through this environment indicates an advanced level of robustness in the UAV's control systems, capable of handling changes in the axis of movement which are common in real-world scenarios. Fig. 4.21 presents the most complex scenario with a dual-stage rotation of the second ring, combining rotations around both the Y and Z axes. This environment demands a highly refined integration of sensory and control systems, enabling the UAV to process and respond to multi-axis rotations simultaneously. The UAV's performance in this environment is indicative of its exceptional ability to generalize its learned behaviors to new and more complex situations. The successful navigation through this scenario underscores the UAV's advanced computational algorithms and its robustness in adapting to compounded rotational challenges.

Strategically placed obstacles in each environment simulate real-world challenges that demand precise trajectory refinements and spatial awareness from the UAV. The UAV must navigate around these barriers while aligning accurately through each ring, proving its ability to analyze the environment, predict obsta-

cles, and make real-time adjustments in increasingly complex conditions. The adaptation from basic parallel navigation to managing complex rotations exemplifies significant advancements in the proposed model's generalization capabilities and its robustness. These attributes are essential for applications requiring high reliability and adaptability in dynamically changing environments, such as in urban navigation, disaster response, and other critical real-world applications where unpredictable changes in environmental geometry are prevalent. These trajectories highlight the UAV's agility, accuracy, and adaptability as it navigates through progressively difficult challenges.

Our study employed a deep reinforcement learning strategy that integrates deep learning techniques for preprocessing visual data. This approach enables UAVs to rapidly and effectively learn obstacle avoidance using only a depth camera, eliminating reliance on additional sensory devices. Distinct from other visual obstacle avoidance methods, our technique proficiently navigates obstacles within real-world settings. However, there remains room for further refinement of this methodology. While simulation experiments indicated an obstacle clearance rate through three circles of nearly 80%, the application in real-world conditions demonstrated a marginally lower success rate, averaging approximately 70%. This variance is largely due to the physical constraints and unpredictable dynamics present in real-world environments, which simulations cannot completely replicate. In practical settings, the minimum success rate of the UAV in maneuvering through rings at various angles stands at 68%. The adaptability of the change from parallel rings to rings with different angles shows a significant improvement in the generalization ability and robustness of our model. This improvement underscores the UAV's ability to adjust to geometrically varied configurations and operational

challenges, further validating the practical efficacy and enhanced adaptability of the SAC algorithm in dynamic real-world settings. This achievement not only signifies a critical progression in UAV navigational autonomy but also delineates potential areas for future research aimed at augmenting model reliability and environmental responsiveness. The successful deployment of the SAC algorithm in both simulated and actual environments represents a significant advancement in the autonomous path planning of UAVs. This research sets the stage for more advanced, dependable UAV operations across diverse applications, thereby improving both safety and operational efficiency. While the SAC algorithm exhibited strong adaptability in both settings, the necessity for parameter adjustment was more pronounced during laboratory evaluations. Variations in lighting, obstacle reflectivity, and sensor discrepancies necessitated real-time modifications to ensure optimal performance.

Chapter 5

Conclusion

This thesis investigates the autonomous navigation capabilities of UAVs, particularly in navigating narrow gaps and intricate terrains, which are essential for applications in fields such as rescue operations and environmental research. One of the key advancements of this research lies in its ability to improve UAV path planning and navigation, enabling more efficient traversal through complex and challenging environments. A significant contribution of this work is the effective mitigation of limitations posed by semi-autonomous human control and unpredictable signal transmission. Semi-autonomous human control, while useful in certain situations, often results in inefficiencies and limited responsiveness, particularly in dynamic or narrow-gap environments where precise, real-time decisions are critical. Additionally, unpredictable signal transmission, which is common in cluttered or remote environments, poses a significant challenge to the continuous and reliable operation of UAVs. By addressing these issues, this research reduces reliance on human intervention and enhances the UAV's ability to operate autonomously, even in environments where communication signals may be intermittent or unreliable.

Consequently, the findings of this thesis mark a significant progression in UAV navigation, providing a more reliable and scalable approach for UAVs to operate in complex, unpredictable settings without the need for constant human oversight.

The core of the first research segment focused on leveraging simulated environments to refine the UAV navigation algorithms. This methodology, although successful within a regulated environment, recognizes the constraints associated with simulation-based evaluations, particularly the lack of empirical verification in authentic field conditions. The dynamic and unpredictable nature of real operational environments often introduces variables that are difficult to replicate in simulations, which could affect the UAV's operational efficacy when transitioning from simulation to actual deployment.

In contrast, the subsequent phase of the research deployed a deep reinforcement learning framework that incorporates advanced deep learning methodologies for the preprocessing of visual data. This integration significantly improves the unmanned aerial vehicle's proficiency in mastering obstacle avoidance through the exclusive use of a depth camera. This method diverges from traditional visual obstacle avoidance strategies by demonstrating successful navigation in real-world settings. Nevertheless, divergences were noted between the outcomes derived from simulations and those from real-world implementations—the efficacy in simulated environments reached approximately 80%, whereas in practical applications, it exhibited a modest decline, averaging about 70%. This discrepancy is chiefly ascribed to the physical constraints and unanticipated dynamics that simulations are unable to completely emulate.

Furthermore, the adaptability of the UAV to navigate through rings with varying angles was significantly improved, showcasing the generalization capability

and robustness of the SAC algorithm under dynamically changing conditions. This not only demonstrates a pivotal advancement in UAV navigational autonomy but also underscores the necessity for ongoing enhancements in model reliability and environmental adaptability.

The effective deployment of the SAC algorithm in both simulated and real-world settings highlights the prospects for more advanced and dependable UAV operations across diverse sectors, consequently improving both safety and operational efficacy. The study demonstrates that although the SAC algorithm showed significant adaptability, there was a heightened need for parameter adjustment during controlled experiments. Fluctuations in environmental factors, including lighting conditions, obstacle reflectivity, and sensor discrepancies, required immediate modifications to sustain peak performance.

In conclusion, while the transition from simulated to real-world applications presents challenges, the insights and refinements derived from these tests are invaluable for tailoring UAV algorithms for real-world deployment. The investigation not only propels advancements in the field of UAV navigation but also establishes a benchmark for future studies focused on augmenting the practical effectiveness and versatility of UAV systems within intricate operational environments.

Part of this thesis has been published in the paper, Guo, Jingrui, Huang, Chao, and Huang, Hailong (2023). “A Deep Q-Network-Based Algorithm for Obstacle Avoidance and Target Tracking for Drones”. In: 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 4530-4535. and Guo, Jingrui et al. (2024). “Advancements in UAV Path Planning: A Deep Reinforcement Learning Approach with Soft Actor-Critic for Enhanced Navigation”. In:

Unmanned Systems.

5.1 Summary of Contributions

This thesis advances the discipline of UAV navigation through the integration and validation of cutting-edge deep reinforcement learning frameworks designed to improve path planning and autonomous decision-making capabilities in multi-faceted environments. The thesis concentrates on two primary aspects: the development of a DRL framework specifically designed for efficient UAV path planning, and the complexities surrounding sim-to-real transitions.

DRL framework designed specifically for online path planning allows UAVs to navigate narrow and intricate terrains typically found in rescue and research operations. Such environments are challenging due to unpredictable signal transmission and spatial constraints. The framework enhances UAVs' ability to make autonomous decisions without heavy reliance on detailed map data, which is a significant limitation in traditional path planning methodologies. A pivotal aspect of this framework is its ability to directly map action commands from sensor data, facilitating real-time command relay and precise navigation through complex terrains. The SAC Update Delay is the fastest converge than other algorithms in Fig. 4.8. This approach not only streamlines the UAV's decision-making processes but also enables rapid convergence of neural networks, which is crucial for operating effectively in real environments.

Furthermore, the research tackles the perennial challenge of the simulation-to-reality transition that plagues many UAV development processes. By deliberately increasing the complexity of simulated environments beyond that typically

encountered in the real world, the research ensures that UAVs are adequately prepared for various operational conditions. Increasing the complexity of the simulation environment, as opposed to utilizing a high-fidelity simulator, offers several notable advantages, particularly in addressing the challenges associated with sim-to-real transfer in UAV development. This approach significantly reduces the computational resources and costs typically required by high-fidelity simulators, which demand substantial processing power to accurately model real-world physics and sensor systems. By introducing more complex yet computationally manageable simulation scenarios, the research can test a wider variety of environmental conditions and UAV behaviors without the need for expensive hardware or high-performance computing infrastructure. Moreover, this method allows for faster experimentation and iteration, enabling the rapid testing and refinement of algorithms, strategies, and design modifications without the delays inherent in high-fidelity simulation or real-world testing. This method not only facilitates the development of robust reinforcement learning strategies but also enhances the algorithms' transferability from simulated to practical applications.

Moreover, the architectural adjustments in the SAC algorithm reduce the risk of uncontrolled UAV behaviors, ensuring safer and more reliable operations. The strategic implementations detailed in this thesis not only enhance the UAV's operational autonomy and adaptability but also set a robust foundation for future advancements in UAV technology and applications. A holistic strategy tackles key obstacles in UAV navigation and expands the frontiers of drone performance, paving the way for advanced and dependable operations across diverse applications, ultimately boosting both safety and efficiency.

5.2 Future Research

The subsequent stage of this research will concentrate on evaluating the success rate of the DRL model within real-world scenarios that feature dynamic obstacles. This evaluation is crucial as it aims to verify the model's proficiency in guiding UAVs through various obstructions to their intended destinations under evolving conditions. The primary goal is to ascertain the model's ability to adapt to obstacles that differ in size, shape, and trajectory, thus offering a concrete measure of its adaptability and decision-making prowess. Successful navigation in such dynamic environments would mark a significant milestone in the field of autonomous UAV navigation, showcasing the model's capacity to manage situations where traditional algorithms typically fall short.

Achieving consistent performance in these complex settings would considerably expand the practical applications of UAVs, enabling them to autonomously execute missions across diverse domains such as search and rescue, precision agriculture, and infrastructure inspection. Demonstrating effectiveness in these conditions would also bolster confidence in deploying UAVs for tasks that demand high adaptability and real-time decision-making capabilities.

Given the noted discrepancies between the outcomes in simulated versus real-world settings observed in previous experiments, future research will concentrate on refining the SAC algorithm's robustness against physical-world variables. This initiative will encompass optimizing sensor integration and fine-tuning the algorithm to more effectively bridge the gap between simulated training and actual operational environments. These enhancements are anticipated to facilitate greater adoption of UAVs in commercial and scientific fields where manual control is ei-

ther impractical or infeasible.

The forthcoming phase of the research is set to establish UAVs as dependable autonomous entities, adept at performing essential tasks in volatile and dynamically evolving real-world settings with limited human oversight. This progression marks a crucial advancement toward harnessing the complete potential of UAV technology across a broad spectrum of practical implementations, emphasizing the necessity for ongoing enhancement and adaptation of UAV operational algorithms.

References

[1] S. Aggarwal and N. Kumar, “Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges,” in *Computer Communications*, vol. 149, pp. 270–299, 2020.

[2] A. Ait Saadi, A. Soukane, Y. Meraihi, A. Benmessaoud Gabis, S. Mirjalili, and A. Ramdane-Cherif, “UAV path planning using optimization approaches: A survey,” in *Archives of Computational Methods in Engineering*, vol. 29, no. 6, pp. 4233–4284, 2022.

[3] A. Alagha, S. Singh, R. Mizouni, J. Bentahar, and H. Otrok, “Target localization using multi-agent deep reinforcement learning with proximal policy optimization,” in *Future Generation Computer Systems*, vol. 136, pp. 342–357, 2022.

[4] A. M. Ali, A. Gupta, and H. A. Hashim, “Deep reinforcement learning for Sim-to-Real policy transfer of VTOL-UAVs offshore docking operations,” in *Applied Soft Computing*, vol. 162, p. 111 843, 2024.

[5] R. Ashour, S. Aldhaheri, and Y. Abu-Kheil, “Applications of UAVs in search and rescue,” in *Unmanned Aerial Vehicles Applications: Challenges and Trends*, pp. 169–200, 2023.

[6] S. B. Atitallah, M. Driss, W. Boulila, and H. B. Ghézala, "Leveraging deep learning and IoT big data analytics to support the smart cities development: Review and future directions," in *Computer Science Review*, vol. 38, p. 100303, 2020.

[7] Y. Bai, H. Zhao, X. Zhang, Z. Chang, R. Jäntti, and K. Yang, "Towards autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches," in *IEEE Communications Surveys & Tutorials*, vol. 25, no. 4, pp. 3038–3067, 2023.

[8] M. Banafaa, Ö. Pepeoğlu, I. Shayea, A. Alhammadi, Z. Shamsan, M. A. Razaz, M. Alsagabi, and S. Al-Sowayan, "A comprehensive survey on 5G-and-beyond networks with UAVs: Applications, emerging technologies, regulatory aspects, research trends and challenges," in *IEEE Access*, vol. 12, pp. 7786–7826, 2024.

[9] F. Betti Sorbelli, "UAV-based delivery systems: A systematic review, current trends, and research challenges," in *Journal on Autonomous Transportation Systems*, vol. 1, no. 3, pp. 1–40, 2024.

[10] B. Cetinsaya, D. Reiners, and C. Cruz-Neira, "From pid to swarms: A decade of advancements in drone control and path planning (2013–2023)," in *Swarm and Evolutionary Computation*, vol. 89, p. 101 626, 2024.

[11] N. K. Chandran, M. T. H. Sultan, A. Łukaszewicz, F. S. Shahar, A. Holovatyy, and W. Giernacki, "Review on type of sensors and detection method of anti-collision system of unmanned aerial vehicle," in *Sensors*, vol. 23, no. 15, p. 6810, 2023.

[12] X. Chen, J. Tang, Y. Ruan, and J. Zhan, "Path planning methods for UAVs: A survey," in *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, pp. 894–903, 2024.

[13] X. Chen, G. Qu, Y. Tang, S. Low, and N. Li, "Reinforcement learning for selective key applications in power systems: Recent advances and future challenges," in *IEEE Transactions on Smart Grid*, vol. 13, no. 4, pp. 2935–2958, 2022.

[14] Y. Chen, Q. Dong, X. Shang, Z. Wu, and J. Wang, "Multi-UAV autonomous path planning in reconnaissance missions considering incomplete information: A reinforcement learning method," in *Drones*, vol. 7, no. 1, p. 10, 2022.

[15] Y. Deng, I. A. Meer, S. Zhang, M. Ozger, and C. Cavdar, "D3QN-based trajectory and handover management for UAVs co-existing with terrestrial users," in *2023 21st International Symposium on Modeling and Optimiza-tion in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, IEEE, pp. 103–110, 2023.

[16] C. Dinelli, J. Racette, M. Escarcega, S. Lotero, J. Gordon, J. Montoya, C. Dunaway, V. Androulakis, H. Khaniani, S. Shao, *et al.*, "Configurations and applications of multi-agent hybrid drone/unmanned ground vehicle for underground environments: A review," in *Drones*, vol. 7, no. 2, p. 136, 2023.

[17] Q. Fang, X. Xu, X. Wang, and Y. Zeng, "Target-driven visual navigation in indoor scenes using reinforcement learning and imitation learning," in *CAAI Transactions on Intelligence Technology*, vol. 7, no. 2, pp. 167–176, 2022.

[18] X. Gao, L. Yan, Z. Li, G. Wang, and I. M. Chen, "Improved deep deterministic policy gradient for dynamic obstacle avoidance of mobilerobot," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 53, no. 6, pp. 3675–3682, 2023.

[19] J. Ge and L. Liu, "Hierarchical robust model prediction control for a long-endurance unmanned aerial vehicle," in *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 6, pp. 1176–1183, 2023.

[20] M. Gök, "Dynamic path planning via dueling double deep Q-network (D3QN) with prioritized experience replay," in *Applied Soft Computing*, vol. 158, p. 111 503, 2024.

[21] H. J. Hadi, Y. Cao, K. U. Nisa, A. M. Jamil, and Q. Ni, "A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs," in *Journal of Network and Computer Applications*, vol. 213, p. 103 607, 2023.

[22] J. Hao, T. Yang, H. Tang, C. Bai, J. Liu, Z. Meng, P. Liu, and Z. Wang, "Exploration in deep reinforcement learning: From single-agent to multiagent domain," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 7, pp. 8762–8782, 2023.

[23] M. Hooshyar and Y.-M. Huang, "Meta-heuristic algorithms in UAV path planning optimization: A systematic review (2018–2022)," in *Drones*, vol. 7, no. 12, p. 687, 2023.

[24] Y. Hu, Y. Liu, A. Kaushik, C. Masouros, and J. S. Thompson, "Timely data collection for UAV-based IoT networks: A deep reinforcement learning approach," in *IEEE Sensors Journal*, vol. 23, no. 11, pp. 12295–12308, 2023.

[25] Z. Jiandong, G. Yukun, Z. Lihui, Y. Qiming, S. Guoqing, and W. Yong, "Real-time UAV path planning based on LSTM network," in *Journal of Systems Engineering and Electronics*, vol. 35, no. 2, pp. 374–385, 2024.

[26] Q. Jin, Q. Hu, P. Zhao, S. Wang, and M. Ai, "An improved probabilistic roadmap planning method for safe indoor flights of unmanned aerial vehicles," in *Drones*, vol. 7, no. 2, p. 92, 2023.

[27] M. Jones, S. Djahel, and K. Welsh, "Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey," in *ACM Com-puting Surveys*, vol. 55, no. 11, pp. 1–39, 2023.

[28] M. R. Jones, S. Djahel, and K. Welsh, "An efficient and rapidly adaptable lightweight multi-destination urban path planning approach for UAVs using Q-learning," in *IEEE Transactions on Intelligent Vehicles*, 2024.

[29] M. J. Kim, T. Y. Kang, and C. K. Ryoo, "Real-time path planning for unmanned aerial vehicles based on compensated voronoi diagram," in *International Journal of Aeronautical and Space Sciences*, pp. 1–10, 2024.

[30] H. Kurunathan, H. Huang, K. Li, W. Ni, and E. Hossain, "Machine learning-aided operations and communications of unmanned aerial vehicles: A contemporary survey," in *IEEE Communications Surveys & Tutorials*, vol. 26, no. 1, pp. 496–533, 2023.

[31] B. Li, R. Yang, L. Liu, J. Wang, N. Zhang, and M. Dong, "Robust computation offloading and trajectory optimization for multi-UAV-assisted MEC:

A multi-agent DRL approach," in *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 4775–4786, 2023.

[32] C. Li, P. Zheng, Y. Yin, B. Wang, and L. Wang, "Deep reinforcement learning in smart manufacturing: A review and prospects," in *CIRP Journal of Manufacturing Science and Technology*, vol. 40, pp. 75–101, 2023.

[33] J. Li, G. Zhang, C. Jiang, and W. Zhang, "A survey of maritime unmanned search system: Theory, applications and future directions," in *Ocean Engineering*, vol. 285, p. 115 359, 2023.

[34] X. Li, S. Yu, X. Gao, Y. Yan, and Y. Zhao, "Path planning and obstacle avoidance control of UUV based on an enhanced A* algorithm and MPC in dynamic environment," in *Ocean Engineering*, vol. 302, p. 117 584, 2024.

[35] J. X. Lv, L. J. Yan, S. C. Chu, Z. M. Cai, J. S. Pan, X. K. He, and J. K. Xue, "A new hybrid algorithm based on golden eagle optimizer and grey wolf optimizer for 3D path planning of multiple UAVs in power inspection," in *Neural Computing and Applications*, vol. 34, no. 14, pp. 11 911–11 936, 2022.

[36] M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned aerial vehicles for search and rescue: A survey," in *Remote Sensing*, vol. 15, no. 13, p. 3266, 2023.

[37] S. MahmoudZadeh, A. Yazdani, Y. Kalantari, B. Ciftler, F. Aidarus, and M. O. Al Kadri, "Holistic review of UAV-centric situational awareness: Applications, limitations, and algorithmic challenges," in *Robotics*, vol. 13, no. 8, p. 117, 2024.

[38] A. Mannan, M. S. Obaidat, K. Mahmood, A. Ahmad, and R. Ahmad, "Classical versus reinforcement learning algorithms for unmanned aerial vehicle network communication and coverage path planning: A systematic literature review," in *International Journal of Communication Systems*, vol. 36, no. 5, e5423, 2023.

[39] H. Mazaheri, S. Goli, and A. Nourollah, "A survey of 3D space path-planning methods and algorithms," in *ACM Computing Surveys*, 2024. DOI: 10.1145/3673896.

[40] K. Milidonis, A. Eliades, V. Grigoriev, and M. Blanco, "Unmanned aerial vehicles in the planning, operation and maintenance of concentrating solar thermal systems: A review," in *Solar Energy*, vol. 254, pp. 182–194, 2023.

[41] S. Mishra and P. Palanisamy, "Autonomous advanced aerial mobility an end-to-end autonomy framework for UAVs and beyond," in *IEEE Access*, vol. 11, pp. 136 318–136 349, 2023.

[42] S. A. H. Mohsan, N. Q. H. Othman, Y. Li, M. H. Alsharif, and M. A. Khan, "Unmanned aerial vehicles (UAVs): Practical aspects, applications, open challenges, security issues, and future trends," in *Intelligent Service Robotics*, vol. 16, no. 1, pp. 109–137, 2023.

[43] S. Munikoti, D. Agarwal, L. Das, M. Halappanavar, and B. Natarajan, "Challenges and opportunities in deep reinforcement learning with graph neural networks: A comprehensive review of algorithms and applications," in *IEEE Transactions on Neural Networks and Learning Systems*, 2023. DOI: 10.1109/TNNLS.2023.3283523.

[44] R. Muralidhar, R. Borovica-Gajic, and R. Buyya, "Energy efficient computing systems: Architectures, abstractions and modeling to techniques and standards," in *ACM Computing Surveys (CSUR)*, vol. 54, no. 11s, pp. 1–37, 2022.

[45] D. A. Neu, J. Lahann, and P. Fettke, "A systematic literature review on state-of-the-art deep learning methods for process prediction," in *Artificial Intelligence Review*, vol. 55, no. 2, pp. 801–827, 2022.

[46] K. K. Nguyen, T. Q. Duong, T. Do-Duy, H. Claussen, and L. Hanzo, "3D UAV trajectory and data collection optimisation via deep reinforcement learning," in *IEEE Transactions on Communications*, vol. 70, no. 4, pp. 2358–2371, 2022.

[47] J. Ou, S. H. Hong, G. Song, and Y. Wang, "Hybrid path planning based on adaptive visibility graph initialization and edge computing for mobile robots," in *Engineering Applications of Artificial Intelligence*, vol. 126, p. 107110, 2023.

[48] P. Paikrao, S. Routray, A. Mukherjee, A. R. Khan, and R. Vohnout, "Consumer personalized gesture recognition in UAV based industry 5.0 applications," in *IEEE Transactions on Consumer Electronics*, vol. 69, no. 4, pp. 842–849, 2023.

[49] S. Paul and A. Mitra, "A review on applications of machine learning in IoT: Challenges and future prospects," in *Internet of Things*, pp. 299–330, 2023.

[50] J. Rao, C. Xiang, J. Xi, J. Chen, J. Lei, W. Giernacki, and M. Liu, "Path planning for dual UAVs cooperative suspension transport based on artificial

potential field A* algorithm," in *Knowledge-Based Systems*, vol. 277, p. 110797, 2023.

[51] M. R. Rezaee, N. A. W. A. Hamid, M. Hussin, and Z. A. Zukarnain, "Comprehensive review of drones collision avoidance schemes: Challenges and open issues," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6397–6426, 2024.

[52] S. Rezwan and W. Choi, "Artificial intelligence approaches for UAV navigation: Recent advances and future challenges," in *IEEE access*, vol. 10, pp. 26 320–26 339, 2022.

[53] Z. Shen, G. Zhou, H. Huang, C. Huang, Y. Wang, and F. Y. Wang, "Convex optimization-based trajectory planning for quadrotors landing on aerial vehicle carriers," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 138–150, 2023.

[54] J. V. Shirabayashi and L. B. Ruiz, "Toward UAV path planning problem optimization considering the internet of drones," in *IEEE Access*, vol. 11, pp. 136 825–136 854, 2023.

[55] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing with deep reinforcement learning," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 1205–1212, 2021.

[56] L. Tai and M. Liu, "A robot exploration strategy based on Q-learning network," in *2016 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 57–62, 2016.

[57] K. Tamanakijprasart, S. Mondal, and A. Tsourdos, “Dynamic path planning of UAV in three-dimensional complex environment based on interfered fluid dynamical system,” in *AIAA SCITECH 2024 Forum*, p. 2091, 2024.

[58] J. Tang, J. Song, J. Ou, J. Luo, X. Zhang, and K. K. Wong, “Minimum throughput maximization for multi-UAV enabled WPCN: A deep reinforcement learning method,” in *IEEE Access*, vol. 8, pp. 9124–9132, 2020.

[59] J. Tang, Y. Liang, and K. Li, “Dynamic scene path planning of UAVs based on deep reinforcement learning,” in *Drones*, vol. 8, no. 2, p. 60, 2024.

[60] J. Tang, H. Duan, and S. Lao, “Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review,” in *Artificial Intelligence Review*, vol. 56, no. 5, pp. 4295–4327, 2023.

[61] K. Telli, O. Kraa, Y. Himeur, A. Ouamane, M. Boumehraz, S. Atalla, and W. Mansoor, “A comprehensive review of recent research trends on unmanned aerial vehicles (UAVs),” in *Systems*, vol. 11, no. 8, p. 400, 2023.

[62] S. Wandelt, S. Wang, C. Zheng, and X. Sun, “Aerial: A meta review and discussion of challenges toward unmanned aerial vehicle operations in logistics, mobility, and monitoring,” in *IEEE Transactions on Intelligent Trans-portation Systems*, vol. 25, no. 7, pp. 6276–6289, 2023.

[63] J. Wang, W. Chi, C. Li, C. Wang, and M. Q. H. Meng, “Neural RRT*: Learning-based optimal path planning,” in *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1748–1758, 2020.

[64] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, “Deep reinforcement learning based dynamic trajectory control for UAV-

assisted mobile edge computing," in *IEEE Transactions on Mobile Computing*, vol. 21, no. 10, pp. 3536–3550, 2021.

[65] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, "Deep reinforcement learning: A survey," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 4, pp. 5064–5078, 2022.

[66] Y. Wang, Z. Gao, J. Zhang, X. Cao, D. Zheng, Y. Gao, D. W. K. Ng, and M. Di Renzo, "Trajectory design for UAV-based internet of things data collection: A deep reinforcement learning approach," in *IEEE Internet of Things Journal*, vol. 9, no. 5, pp. 3899–3912, 2021.

[67] J. Westheider, J. Rückin, and M. Popović, "Multi-UAV adaptive path planning using deep reinforcement learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, pp. 649–656, 2023.

[68] J. Wu, Y. Ye, and J. Du, "Multi-objective reinforcement learning for autonomous drone navigation in urban areas with wind zones," in *Automation in Construction*, vol. 158, p. 105 253, 2024.

[69] W. Wu, H. Liu, L. Li, Y. Long, X. Wang, Z. Wang, J. Li, and Y. Chang, "Application of local fully convolutional neural network combined with YOLO V5 algorithm in small target detection of remote sensing image," in *PloS one*, vol. 16, no. 10, e0259283, 2021.

[70] C. Xiao, P. Lu, and Q. He, "Flying through a narrow gap using end-to-end deep reinforcement learning augmented with curriculum learning and Sim2Real," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 5, pp. 2701–2708, 2023.

[71] Y. Xue and W. Chen, "Multi-agent deep reinforcement learning for UAVs navigation in unknown complex environment," in *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 2290–2303, 2023.

[72] Z. Xue and T. Gonsalves, "Vision based drone obstacle avoidance by deep reinforcement learning," in *AI*, vol. 2, no. 3, pp. 366–380, 2021.

[73] W. Xushi, "Research on quadrotor UAV control and path planning based on PID controller and Dijkstra algorithm," in *AIP Conference Proceedings*, AIP Publishing, vol. 3144, 2024.

[74] H. S. Yahia and A. S. Mohammed, "Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: A systematic review," in *Environmental Monitoring and Assessment*, vol. 195, no. 1, p. 30, 2023.

[75] L. Yang, P. Li, S. Qian, H. Quan, J. Miao, M. Liu, Y. Hu, and E. Memetimin, "Path planning technique for mobile robots: A review," in *Machines*, vol. 11, no. 10, p. 980, 2023.

[76] P. G. Ye, J. Zheng, X. Ren, J. Huang, Z. Zhang, Y. Pang, and G. Kou, "Optimizing resource allocation in UAV-assisted ultra-dense networks for enhanced performance and security," in *Information Sciences*, vol. 679, p. 120 788, 2024.

[77] Z. Ye, K. Wang, Y. Chen, X. Jiang, and G. Song, "Multi-UAV navigation for partially observable communication coverage by graph reinforcement learning," in *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 4056–4069, 2022.

[78] C. Zhang, S. Liang, C. He, and K. Wang, "Multi-UAV trajectory design and power control based on deep reinforcement learning," in *Journal of Communications and Information Networks*, vol. 7, no. 2, pp. 192–201, 2022.

[79] J. Zhang, S. Xu, Y. Zhao, J. Sun, S. Xu, and X. Zhang, "Aerial orthoimage generation for UAV remote sensing," in *Information Fusion*, vol. 89, pp. 91–120, 2023.

[80] W. Zhang, "An improved DBSCAN algorithm for hazard recognition of obstacles in unmanned scenes," in *Soft Computing*, vol. 27, no. 24, pp. 18 585–18 604, 2023.

[81] J. Zhao, W. Zhao, B. Deng, Z. Wang, F. Zhang, W. Zheng, W. Cao, J. Nan, Y. Lian, and A. F. Burke, "Autonomous driving system: A comprehensive survey," in *Expert Systems with Applications*, vol. 242, p. 122 836, 2024.

[82] X. Zhao, X. Wang, Y. Dai, and Q. Qiu, "Joint optimization of loading, mission abort and rescue site selection policies for UAV," in *Reliability Engineering & System Safety*, vol. 244, p. 109 955, 2024.

[83] Y. Zhou, J. Shu, H. Hao, H. Song, and X. Lai, "UAV 3D online track planning based on improved SAC algorithm," in *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, no. 1, p. 12, 2024.

[84] T. Zoppi, A. Ceccarelli, T. Puccetti, and A. Bondavalli, "Which algorithm can detect unknown attacks? comparison of supervised, unsupervised and meta-learning algorithms for intrusion detection," in *Computers & Security*, vol. 127, p. 103 107, 2023.