Q-Learning-Based Dynamic Drone Trajectory Planning in Uncertain Environments

Subrahmanya Chandra Bhamidipati*, Adam Maxwell*, Emily Pham*, Johann Zhang*, Sharan Srinivas*, Prasad Calyam*
*University of Missouri, Columbia
{sb5q6, srinivassh, calyamp}@missouri.edu
jam144@uark.edu, pham0579@umn.edu, johann.zhang@tufts.edu

***Abstract*—Unmanned Aerial Vehicles (UAVs) are being increasingly used in delivery services. When the UAV loses communication with the truck, efficient navigation and re-routing in dynamic environments with uncertainties, such as obstacles and traffic congestion, remain as significant challenges. This paper presents a novel Drone Trajectory Planning (DTP) model using Q-Learning to address these challenges. Our model leverages state representations including the truck's location, the drone's position, proximity to congestion zones, and its general direction to adapt to uncertainties. We conducted extensive simulations using a dataset that simulates real-world urban environments. The proposed method demonstrates robust performance in navigating complex environments, optimizing path decisions, and ensuring timely deliveries even when communication with the truck is lost. Add quantitative results.**

***Index Terms*—Keywords: UAV Navigation, Dynamic Decision-Making, Reinforcement Learning, Obstacle Avoidance.**

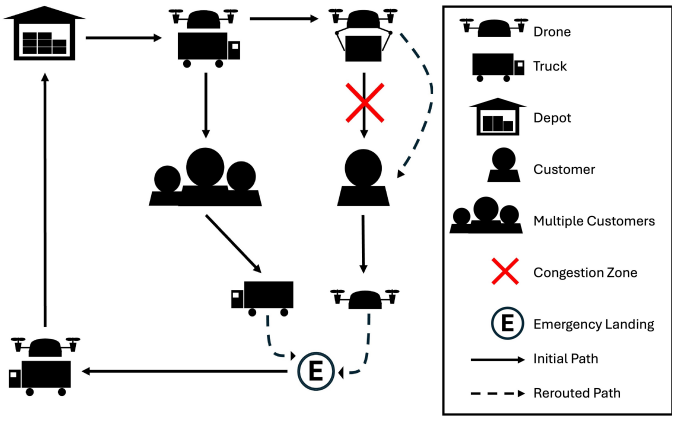

Fig. 1. Route Management System Example

I. Introduction

The rise of autonomous drones in logistics and delivery services has opened new frontiers in efficient and contactless delivery solutions. These drones have shown great potential in applications such as urban delivery, surveillance, and disaster management, providing timely and precise services. However, ensuring the reliability and safety of these drones, particularly when they lose connection with their control centers, remains a significant challenge.

Dynamic decision-making in autonomous drone operations faces several challenges. These include navigating through unpredictable environments with obstacles and traffic congestion, maintaining energy efficiency, and ensuring safe operations during communication loss with control centers. Traditional navigation methods may not adequately address these challenges, leading to inefficient routing and increased risk of collisions or failed deliveries.

This paper addresses the problem of dynamic decision-making for a drone that loses communication with its delivery truck. The drone must autonomously decide whether to continue to the customer's location, return to an emergency spot, or head back to the depot. We propose a Reinforcement Learning-based approach, specifically Q-learning, to enable the drone to navigate a grid environment with obstacles and congestion zones, making optimal decisions based on real-time conditions. Our approach aims to enhance the drone's adaptability and decision-making capabilities in dynamic and uncertain environments.

We conducted extensive simulations to evaluate the performance of our proposed method. These simulations were carried out in a grid environment designed to mimic real-world urban settings, complete with various obstacles and congestion zones. The results demonstrate that our Q-learning based approach significantly improves the UAV's ability to navigate efficiently and safely, even when communication with the delivery truck is lost. Figures showcasing the drone's routing and decision-making process under different scenarios are included to illustrate the effectiveness of our approach.

The remainder of this paper is organized as follows: Section II reviews related work in the field of reinforcement learning for autonomous drone navigation. Section III describes the problem formulation and the proposed Q-learning approach in detail. Section IV presents the simulation setup and results, highlighting the performance improvements achieved by our method. Finally, Section V concludes the paper and discusses potential future work.

II. Related Work

The current state of the art in UAV decision-making primarily focuses on using reinforcement learning (RL) for autonomous navigation. Various RL algorithms, such as Q-learning, Deep Q-Networks (DQN), and policy gradient methods, have been applied to help drones navigate complex,

dynamic environments. These studies emphasize the ability of UAVs to avoid obstacles, optimize their paths, and handle real-time environmental changes like traffic congestion. Many simulations are conducted in grid environments to test and refine the efficiency of UAV decision-making strategies [1]–[3].

Recent research in this area includes the development of energy-efficient multi-drone systems and improved coordination strategies. For instance, studies by Qu et al. and Singh et al. explore RL frameworks that balance obstacle avoidance with energy consumption and coordination among multiple UAVs. However, these approaches typically assume continuous communication between drones and a central controller, relying on consistent data flow for decision-making. This focus on energy efficiency and coordination helps improve UAV operations in collaborative environments but leaves gaps in scenarios where communication is lost [4]–[6].

Our approach introduces a novel focus on UAV decision-making in situations where communication with the delivery truck is lost. Unlike prior work that assumes constant connectivity, our model uses RL to enable the UAV to navigate independently, making real-time decisions based on environmental inputs such as proximity to congestion zones, obstacles, and available energy. By incorporating this disconnection scenario, our method ensures that the UAV can still operate effectively, making it particularly relevant in unpredictable urban environments where network instability is common.

III. Background and Problem Formulation

The real-time decision-making ability of a UAV is crucial, as it must continuously adapt to the changing conditions in its surroundings. Autonomous UAVs must decide on their flight paths, avoid obstacles, and balance energy constraints to ensure mission success. Traditionally, UAV decision-making relies on rule-based systems or Model Predictive Control (MPC), which require predefined rules or models of the environment to predict future states. While these techniques work well in predictable settings, they often fall short in environments with high uncertainty or when UAVs lose communication with their control centers, such as delivery trucks in logistics applications.

Maintaining reliable communication between UAVs and ground control is vital for real-time updates on route planning, obstacle avoidance, and environmental changes like congestion zones. Network disruptions or intermittent connectivity can significantly impact the UAV's ability to receive updates on mission-critical data, such as dynamic traffic conditions or changes in delivery locations. When the UAV experiences communication breakdowns due to network instability or obstacles, it must rely on an independent decision-making system to continue its mission. Robust network connectivity, enabled by protocols such as V2X (Vehicle-to-Everything) or MANET (Mobile Ad-hoc Networks), helps mitigate this issue by allowing the UAV to reconnect and resume communication when the connection is restored. However, during periods of disconnection, the UAV must switch to an autonomous mode that allows it to navigate effectively and complete its mission.

In this context, reinforcement learning (RL) has emerged as a promising solution for UAV decision-making in uncertain environments. RL allows UAVs to learn from their experiences and optimize their actions over time through exploration and exploitation. Specifically, Q-learning and Deep Q-Networks (DQN) have been widely used for UAV path planning, allowing UAVs to navigate autonomously and adapt to their surroundings without relying on a central control unit. These RL techniques are well-suited for scenarios where the UAV must navigate through complex environments, avoid obstacles, and adapt to dynamic conditions.

In our research, we use Reinforcement Learning, specifically Q-learning, to enable UAVs to make dynamic decisions in congested and uncertain environments. RL offers several advantages for UAV decision-making:

- **Adaptability**: RL allows UAVs to adapt to changing environments by learning from their interactions and updating their policies accordingly.
- **Optimality**: RL algorithms aim to maximize cumulative rewards, leading to the discovery of optimal or near-optimal policies for navigation and task completion.
- **Scalability**: RL can handle large state and action spaces, making it suitable for complex UAV missions.
- **Autonomy**: RL enables UAVs to operate autonomously, making real-time decisions without requiring constant human intervention.

1) Partially Observable Markov Decision Processes (POMDPs): Given the uncertainty in the UAV's environment, we model our decision-making problem using Partially Observable Markov Decision Processes (POMDPs). A POMDP framework allows the UAV to make informed decisions based on probabilistic estimates of the current state, considering both immediate rewards and long-term outcomes. The POMDP model consists of:

- **State Space (S)**: All possible states the UAV can be in, encompassing its position, heading, energy levels, and proximity to obstacles.
- **Action Space (A)**: All possible actions the UAV can take, including movement in different directions, speed adjustments, and emergency maneuvers.
- **Transition Function (P)**: The probabilities of transitioning from one state to another given a specific action, accounting for environmental dynamics and uncertainties.
- **Reward Function (R)**: The immediate rewards or penalties associated with actions, guiding the UAV towards optimal behavior.
- **Observations (O)**: The information received by the UAV's sensors, which may be noisy or incomplete.
- **Observation Probability (Z)**: The likelihood of receiving specific observations given the actual state of the environment.

A. Drone Trajectory Prediction (DTP) Algorithm

To implement the POMDP framework, we develop a Drone Trajectory Prediction (DTP) algorithm. This algorithm predicts the UAV's future states and processes decisions in advance to handle potential uncertainties such as obstacles and congestion zones. The DTP algorithm is designed to:

- **Detect Path Deviations**: Identify deviations from the planned path ahead of time, allowing the UAV to adjust its course proactively.
- **Predict Trajectories**: Estimate the UAV's trajectory for the next T seconds, considering the current state and possible actions.
- **Maximize Cumulative Rewards**: Optimize the UAV's actions to maximize the cumulative rewards received along its trajectory.

By integrating the DTP algorithm with the RoutePEARL platform, we enhance the UAV's ability to make dynamic decisions and optimize its route in real-time, ensuring efficient and safe deliveries even in complex and uncertain environments. Our proposed method demonstrates robust performance in navigating complex and unpredictable environments, providing a promising solution for UAV operations. By simulating a grid environment with various obstacles and congestion zones, we aim to demonstrate how RL can provide robust solutions for autonomous drone navigation and decision-making under uncertainty.

IV. Networking and Communication

Communication between UAVs and ground stations or delivery trucks is critical to ensure the success of missions in dynamic environments. Reliable communication allows the UAV to receive real-time updates about obstacles, congestion zones, and delivery routes, enabling the drone to adjust its path accordingly. In urban and rural environments, UAVs rely on various networking protocols to maintain robust connectivity throughout the mission.

A. Networking Protocols for UAVs

Two key networking technologies used in UAV systems are Vehicle-to-Everything (V2X) and Mobile Ad-hoc Networks (MANETs).

a) V2X Communication: V2X enables UAVs to communicate with other vehicles, infrastructure, and the truck in logistics operations. This technology provides a low-latency communication channel that helps the UAV receive up-to-date information on road conditions, traffic congestion, or changes to the delivery route. V2X is especially beneficial in urban environments, where real-time coordination between multiple actors (UAVs, trucks, traffic systems) is crucial.

b) Mobile Ad-hoc Networks (MANETs): MANETs are decentralized networks that enable UAVs to communicate without relying on fixed infrastructure. In rural or remote areas where cellular connectivity might be limited, MANETs allow UAVs to form peer-to-peer networks with nearby devices, including other UAVs and ground units. This decentralized communication enables the UAV to maintain a connection with the truck, even if traditional network coverage is unavailable.

B. Handling Network Disconnections

Although these networking protocols support efficient data exchange, network disruptions due to environmental factors, physical obstructions, or interference can impact the UAV's ability to receive updates. In such cases, UAVs must switch from a network-reliant mode to an autonomous mode, where our proposed model uses reinforcement learning to make decisions independently.

Our model addresses this challenge by implementing a system that allows the UAV to continue its mission even in the absence of communication. When a network disruption occurs, the UAV leverages its local knowledge, sensor data, and RL-based decision-making capabilities to navigate through the environment autonomously. The UAV can assess obstacles, congestion zones, and its own energy levels to make real-time decisions, such as continuing to the customer, returning to the depot, or diverting to an emergency landing spot.

Once the communication link is restored, the UAV resumes receiving real-time data from the truck or control center. This enables the UAV to synchronize its status with updated delivery instructions, adjust its path, and optimize its route based on any new environmental information. This hybrid model ensures that the UAV can operate autonomously during network disconnections while seamlessly reintegrating into a networked environment once communication is reestablished.

V. Decision-Making Framework for Partitioned Drone

Our approach focuses on developing a robust and efficient decision-making framework for UAVs navigating in dynamic and uncertain environments. With Reinforcement Learning, we aim to enable the UAV to adapt to its environment, learn optimal navigation strategies, and make real-time decisions that maximize its cumulative rewards while minimizing risks. This framework leverages a Drone Trajectory Prediction (DTP) algorithm based on the theory of Markov decision processes (MDP). The MDP framework enables the UAV to make informed decisions by predicting future states and handling uncertainties such as obstacles and traffic congestion. This section provides a detailed explanation of the components of the MDP model, including state space, action space, transition probabilities, reward function, and observations.

In this dynamic approach where the drone needs to make reliable decisions to navigate towards the customer, return to the depot, or move to an emergency landing site based on various conditions, we leveraged Q-Learning. The key components of this approach include:

- **Q-value function (action-value function)**: Characterizes the state-action relationship.
- **Bellman equation**: A recursive mathematical formula used in reinforcement learning to express the expected cumulative reward (Q-value) associated with an agent taking

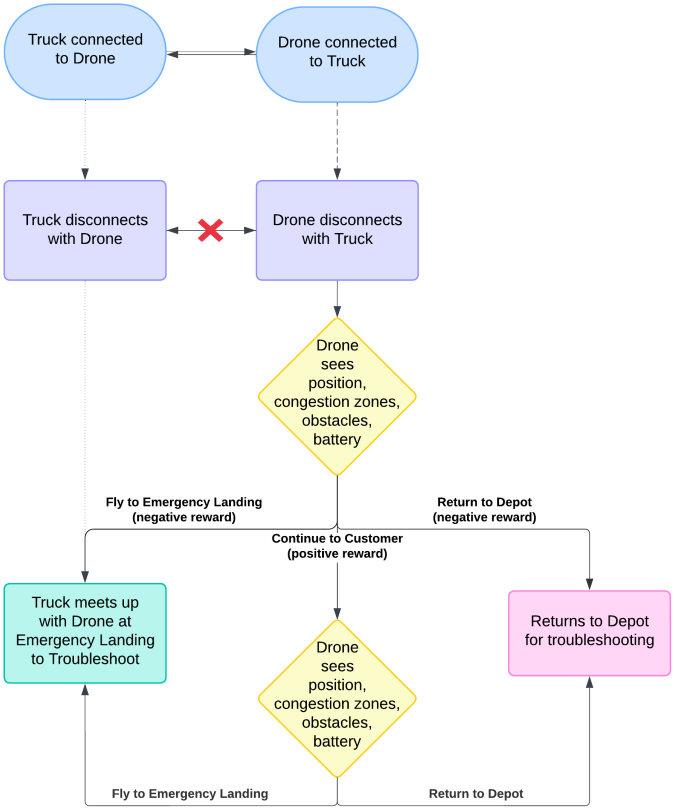

Fig. 2. Decision making framework

a specific action in the current state of the environment. The Bellman equation for Q-learning is given by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

where:

- $Q(s,a)$ is the current Q-value of taking action a in state s.
- α is the learning rate.
- r is the immediate reward received after taking action a in state s.
- γ is the discount factor.
- $\max_{a'} Q(s',a')$ is the maximum Q-value for the next state s' over all possible actions a'.
- s' is the next state resulting from taking action a in state s.

.

- **Q-value function,** $Q(s,a)$: Takes two input parameters:
  - **Current state** s: Represents the current condition or position of the UAV.
  - **Action** a: Represents a specific maneuver or decision taken by the UAV.
- **Cumulative reward prediction**: The Q-value function predicts the cumulative reward an agent can expect to achieve by taking action a in state s and subsequently following the optimal policy.

Here, we detail our state space, action space, and reward function specific to our drone decision making.

A. State Space (S)

The state space represents all possible states that the UAV can be in during its operation. Each state captures essential information about the UAV's environment and its operational parameters. For our model, the state at time t, denoted as s_t, includes the following components:

- **Drone's Position** ($P_D(n)$): The current coordinates of the UAV in the 2D grid environment.
- **Vehicle's Heading** (ϕ_t): The direction in which the UAV is currently moving.
- **Vehicle's Total Time Capacity (Battery)** (E_t^n): The total time the UAV has been operating since the last recharge or reset.
- **Delivery Location** (L_d): The coordinates of the intended delivery point.
- **In Congestion Zone** (C_z): A boolean variable indicating whether the UAV is currently within a congestion zone.
- **Nearest Congestion Zone** (D_{Cz}): The cardinal direction of the nearest congestion zone within a 100-pixel radius, helping the UAV avoid these areas.

By including these parameters, the state space provides a comprehensive snapshot of the UAV's operational context, allowing the decision-making algorithm to account for both the UAV's and the truck's positions and capabilities.

For example, the state at time t could be represented as:

$$\begin{aligned} s_t = \{ & P_D(n) = (10, 20), \\ & P_T(n) = (30, 40), \\ & \phi_t = 45^\circ, \\ & E_t^n = 80 \,\text{minutes}, \\ & F_t = 40 \,\text{km}, \\ & L_d = (50, 60), \\ & C_z = \text{false}, \\ & D_{Cz} = \text{East}\} \end{aligned}$$

This state captures the UAV's position at coordinates (10, 20), heading at a 45-degree angle, with 80 minutes of battery life remaining. The truck is located at (30, 40) with 40 km of range left. The delivery point is at (50, 60), the UAV is not currently in a congestion zone, and the nearest congestion zone is to the east.

By including these parameters, the state space provides a comprehensive snapshot of the UAV's operational context, allowing the decision-making algorithm to account for both the UAV's and the truck's positions and capabilities.

B. Action Space (A)

The action space defines the set of all possible actions that the UAV can take at any given state. Each action corresponds

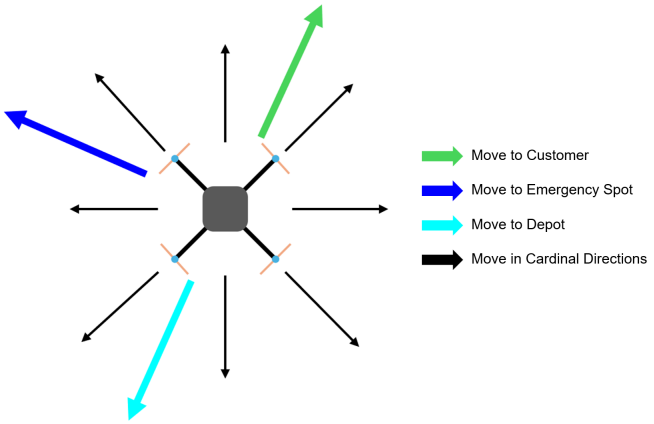

Fig. 3. Actions available for the UAV Will beautify/replace and add legend

to a specific maneuver or decision that the UAV can execute to navigate the environment or adjust its operational parameters. The action set A includes:

- **Move in Eight Directions**: Adjusting the UAV's position to any of the eight cardinal and intercardinal directions (north, northeast, east, southeast, south, southwest, west, northwest) within the 2D grid to navigate towards different locations.
- **Move to Customer**: Directing the UAV to proceed towards the customer's delivery location.
- **Move to Emergency Landing**: Directing the UAV to proceed towards a predefined emergency landing site.
- **Move to Depot**: Directing the UAV to return to the depot or base station.
- **Speed Up**: Increasing the UAV's velocity to accelerate.
- **Speed Down**: Decreasing the UAV's velocity to decelerate.

Each of these actions plays a crucial role in enabling the UAV to navigate efficiently and respond to dynamic changes in its environment:

Move to Customer, Emergency Landing, or Depot: The coordinates of the drone and the respective target (customer, emergency landing site, or depot) are used to calculate the vector pointing towards the target. This vector provides the heading direction for the UAV. By aligning its heading with the calculated vector, the UAV can navigate directly towards its intended destination, whether it's completing a delivery, making an emergency landing, or returning to the base station, without manually navigating using the cardinal directions.

Move in Eight Directions: This action allows the UAV to adjust its position in any of the eight directions within the grid. This flexibility is essential for maneuvering around obstacles and avoiding congestion zones. When the UAV detects a problem area, such as an obstacle or a congestion zone, it can calculate the optimal direction to move away from the hazard, ensuring a safer and more efficient path towards its goal.

Adjust Speed: This action allows the UAV to either increase or decrease its speed, which involves a trade-off between risk and reward. Increasing speed enables the UAV to reach its destination faster, enhancing delivery efficiency. However, this also results in higher energy consumption, causing the battery to deplete more quickly. Conversely, decreasing speed conserves energy, extending the UAV's operational range and ensuring it has sufficient battery life to complete its mission or reach a safe location. Adjusting speed also provides the UAV with flexibility to react to dynamic environmental changes, such as sudden obstacles or congestion zones, allowing for more precise and adaptive navigation.

These actions collectively enable the UAV to make informed decisions, optimizing its route and operational efficiency while handling the complexities of a dynamic and uncertain environment.

C. Transition Function (P)

The transition function represents the likelihood of moving from one state to another given a specific action. This function is crucial for modeling the uncertainty inherent in the UAV's environment, such as wind variations or unexpected obstacles. Formally, the transition function from state s_t to state s_{t+1} given action a_t is denoted as $P(s_{t+1}|s_t, a_t)$.

With each action, the state space of the UAV changes, reflecting the new conditions and position of the UAV in the environment:

- **Move in Eight Directions**: By choosing to move in any of the cardinal and intercardinal directions, the UAV's location in the discretized grid space changes. This movement allows the UAV to explore different parts of the grid, enabling it to develop a Q-table with diverse actions in each grid square. As the UAV moves, it encounters various states, which helps in building a comprehensive understanding of the environment and optimizing the decision-making process.
- **Move to Customer, Emergency Landing, or Depot**: Similar to moving in the eight directions, directing the UAV towards the customer, emergency landing site, or depot changes its 2D coordinates. This action aligns the UAV's heading with the vector pointing towards the respective target, resulting in a change in its state. This continuous state change helps the UAV in learning the most efficient paths to reach these critical locations while updating the Q-table with relevant experiences.
- **Adjust Speed**: Adjusting the UAV's speed, whether increasing or decreasing, affects the energy cost or battery drain of each step. Although the physical 2D coordinates of the UAV might remain unchanged by this action, the state of the UAV is still impacted through its energy levels. The interaction of speed adjustments with the energy cost is covered in the reward function section, where the UAV learns to balance speed for efficiency against the need to conserve battery life for completing its mission.

These state changes are integral to the learning process, enabling the UAV to adapt to dynamic environmental conditions and optimize its navigation and decision-making strategies.

D. Reward Function (R)

The reward function quantifies the immediate benefit or cost associated with taking a particular action in a given state. It guides the UAV towards actions that maximize its cumulative reward over time. The reward r_t at time t is defined as follows:

- **Positive Reward for Reaching Customer**: The UAV receives a significant positive reward for successfully reaching the customer's delivery location. This incentivizes the UAV to prioritize completing deliveries.
- **Smaller Positive Reward for Reaching Emergency Landing or Depot**: The UAV receives a smaller positive reward for reaching a predefined emergency landing site or returning to the depot. This ensures that the UAV still gains some benefit from returning to safe locations even without delivering if deemed necessary.
- **Negative Reward for Traversing Congestion Zone**: The UAV incurs a negative reward whenever it traverses through a congestion zone. This discourages the UAV from moving through problematic areas, encouraging it to find alternative, less congested paths.
- **Significant Negative Reward for Collisions or Battery Depletion**: The UAV receives a substantial negative reward if it collides with any obstacle or if its battery is depleted. This penalty emphasizes the importance of avoiding obstacles and maintaining sufficient energy levels to complete the mission.
- **Small Negative Reward for Movement**: There is a small negative reward associated with each movement step, which is influenced by the UAV's velocity. Faster movement incurs a higher penalty, while slower movement results in a lower penalty. This encourages the UAV to balance speed with efficiency, minimizing the distance and time moved while considering energy consumption.

These rewards collectively drive the UAV to optimize its path planning and decision-making processes. Positive rewards for reaching critical locations ensure that the UAV prioritizes its primary objectives, while negative rewards for collisions, energy depletion, and inefficient movement incentivize safe, energy-efficient, and strategic navigation. By balancing these rewards, the UAV learns to navigate complex environments effectively, avoiding hazards, conserving energy, and achieving its delivery goals.

Dr. Sharan wanted to include some algorithms. Maybe here?

VI. Performance Evaluation

A. Grid Environment Setup

The environment is a 800x800 pixel space representing the drone's operating area. The environment includes random obstacles, congestion zones, a start position, a goal position, emergency spots, and a depot. The maze generation algorithm ensures a mix of obstacles and clear paths, while congestion zones represent areas with heavy traffic or high wind speeds and/or delays.

Grid Description: The environment is discretized into a grid with each unit a size of 50x50 pixels which will simplify the action space significantly. The grid is designed to simulate real-world scenarios with varying levels of difficulty. Obstacles are placed randomly to create a dynamic environment that challenges the drone's navigation capabilities.

Require: State space S, action space A, learning rate α, discount factor γ, exploration rate ϵ, episodes N
Ensure: Initialize Q-value table $Q(s, a)$ arbitrarily for all $s \in S$ and $a \in A$
Initialize Q-value table $Q(s, a)$ for all $s \in S$ and $a \in A$
for $episode = 1$ **to** N **do**
  Initialize state s
  while state s is not terminal **do**
    With probability ϵ select a random action a from A
    Otherwise select $a = \arg\max_a Q(s, a)$
    Take action a and observe reward r and next state s'
    Update $Q(s, a)$ as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha\big(r + \gamma \max_{a'} Q(s',a') - Q(s,a)\big)$$

    $s \leftarrow s'$
  end while
end for

Algorithm 1: Q-learning Algorithm for UAV Navigation

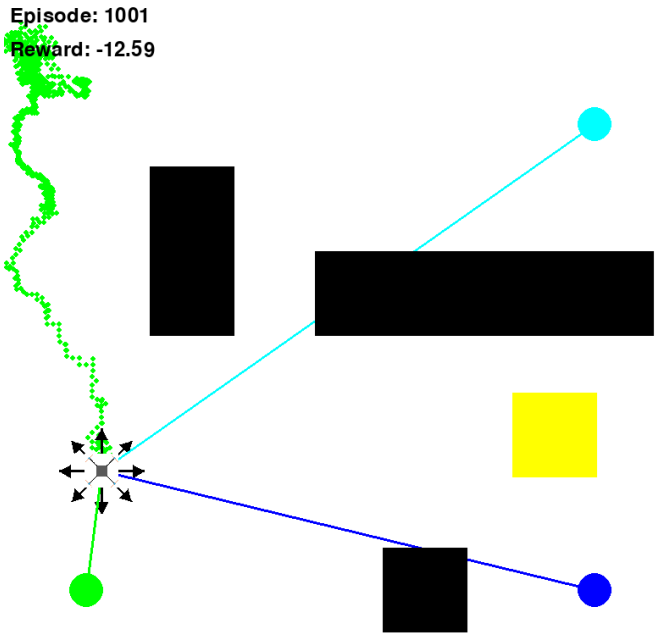

Fig. 4. Environment Setup with Actions

Maze Generation: The maze generation algorithm ensures that there are clear paths from the start to the goal, from the start to the emergency spots, and from the start to the depot. This setup mimics real-world conditions where drones must navigate through unpredictable environments.

Congestion Zones: Congestion zones are randomly generated and represent areas with increased difficulty for navigation, such as high traffic zones or areas with environmental hazards. These zones impact the drone's movement and decision-making process.

B. Simulation and Learning

The learning process involves running the simulation for 200,000 episodes, with the learning rate (alpha), discount

factor (gamma), and epsilon decay carefully tuned to ensure effective learning. The Pygame library visualizes the learning process, displaying the drone's movements, current reward, and Q-values on the screen.

Simulation Parameters: The simulation is configured with a large number of episodes to ensure the drone has sufficient opportunities to explore and learn the optimal policy. The learning rate, discount factor, and epsilon decay are critical parameters that influence the efficiency and effectiveness of the learning process.

TABLE I
HYPERPARAMETERS FOR Q-LEARNING SIMULATIONS SUBJECT TO CHANGE

Hyperparameter	Value
Environment Dimensions	800x800
Grid Dimensions	20x20
Alpha (α)	0.2
Gamma (γ)	0.95
Episodes	100,000
Speed	1 to 20 (variable)
Congestion Zone Size	100x100
Congestion Timer Range	50 to 200 steps
Obstacle Sizes	400x100, 100x200, 100x10
Reward for Reaching Customer	10,000
Reward for Reaching Depot	5,000
Reward for Reaching Emergency Landing Spot	5,000
Penalty for Obstacle collision	-1,000,000,000
Penalty for Congestion Zone	-1,000
Penalty Proportional to Velocity	-0.1 * (abs(vx) + abs(vy))

Learning Process: During each episode, the drone selects an action based on the epsilon-greedy strategy, moves to a new state, receives a reward, and updates the Q-table. This process continues until the drone reaches the goal, an emergency spot, or the depot, or until a maximum number of steps is reached.

Visualization: We use Pygame to visualize the learning process, which enables us to actively monitor the drone's movements, rewards, and evolving Q-values in real-time. This interactive visualization illustrates how the drone learns and adapts to its environment dynamically.

Our experiments demonstrate the Q-learning algorithm's effectiveness in enabling the UAV to navigate a grid environment dynamically and make optimal decisions even in the presence of randomly generated congestion zones and fixed obstacles. The UAV consistently makes real-time decisions, avoiding hazards and ensuring safe navigation.

C. Epsilon Decay

The epsilon decay graph shows the reduction in the exploration rate over time. A lower epsilon value indicates that the drone relies more on its learned policy rather than exploring new actions. The epsilon decay formula used in the Q-learning simulation is given by:

$$\epsilon = \max\left(0.05, 0.9 - \frac{\textit{episode}}{0.95 \times \textit{total\_episodes}}\right)$$

where:

- ϵ is the exploration rate.
- *episode* is the current episode number.
- *total_episodes* is the total number of episodes in the training process.

This formula ensures that the exploration rate starts at 0.9 and decreases linearly over time, but does not go below 0.05 to maintain a minimum level of exploration. This decay in exploration rate helps the UAV to increasingly rely on the learned optimal policy, ensuring efficient navigation and decision-making over time.

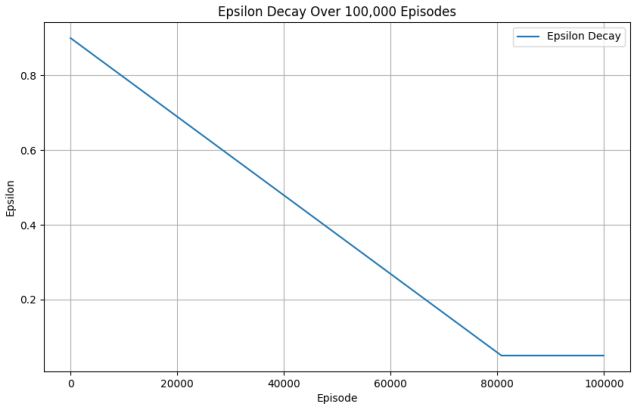

Fig. 5. Epsilon Decay over 100,000 episodes

D. Experimental Results

The UAV's ability to make on-the-fly decisions is showcased through various scenarios where it successfully navigates through dynamically changing environments. The following examples highlight the robustness of the Q-learning algorithm in handling unexpected changes and uncertainties:

Navigating Through Obstacles: As the UAV encounters fixed obstacles, which are colored orange in Figure 6, it dynamically adjusts its path to avoid collisions. This real-time adaptability ensures that the UAV can continue towards its goal without being hindered by static obstacles. By continuously assessing its environment and recalculating the optimal route, the UAV effectively navigates around these obstacles. The green path in the figure illustrates the UAV's ability to maneuver through the environment, showcasing its learned strategies for obstacle avoidance. This capability is essential for real-world applications where the UAV must frequently adapt to physical barriers such as buildings, trees, and other structures, ensuring operational efficiency and mission success.

Avoiding Congestion Zones: The UAV identifies and avoids randomly generated congestion zones within the environment, marked yellow in Figure 7. By recognizing the presence of congestion zones and adjusting its route, the UAV minimizes delays and optimizes its travel time. This proactive approach involves real-time assessment of the environment to detect congestion zones and reroute accordingly. In some scenarios, the UAV may decide that it is more efficient or safer to forego the delivery to the customer and proceed

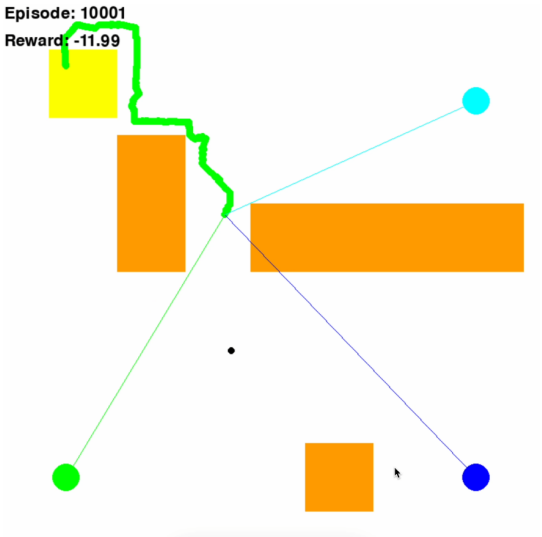

Fig. 6. UAV dynamically navigating through obstacles

directly to the depot, shown as cyan in the environment. This decision ensures that the UAV can avoid heavily congested areas, thereby reducing the risk of delays and ensuring timely mission completion.

Fig. 7. UAV dynamically avoiding congestion zones, by not delivering to customer and returning to depot

Reaching the Goal: The UAV demonstrates its capability to reach the goal efficiently despite the dynamic nature of the environment, as shown in Figure 8. In this scenario, the UAV successfully navigates to the customer's location, marked in green. Whether navigating to the customer's location, an emergency landing spot, or returning to the depot, the UAV consistently makes optimal decisions to ensure successful mission completion. The real-time decision-making process enables the UAV to adapt to changing conditions and obstacles, ensuring it reaches its destination effectively.

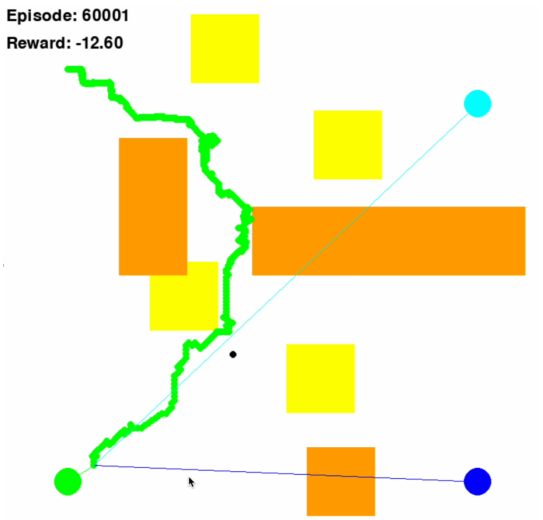

Fig. 8. Real-time decision-making by the UAV, reaching the customer

After successfully reaching the customer in Figure 8, the UAV demonstrates its robust decision-making capabilities by promptly adapting to new directives. Following the customer delivery, the UAV assesses the situation and decides to proceed to an emergency landing spot, as illustrated in Figure 9. This decision underscores the UAV's ability to prioritize safety and adapt to unforeseen circumstances. Subsequently, the UAV determines that the optimal next step is to return to the depot, ensuring it is ready for future missions. This final decision is depicted in Figure 10, showcasing the UAV's efficiency in mission completion and readiness for subsequent tasks.

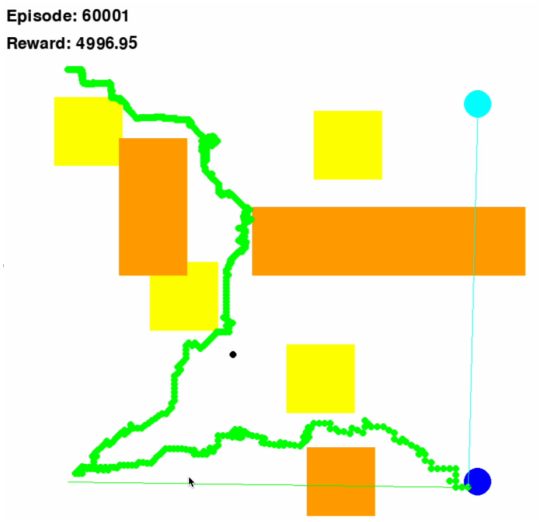

Fig. 9. Real-time decision-making by the UAV, reaching the emergency landing spot

These visualizations illustrate the UAV's proficiency in making real-time adjustments to its path, showcasing the effectiveness of the Q-learning algorithm in dynamic and uncertain environments. The algorithm's ability to handle random changes and make informed decisions on the fly is

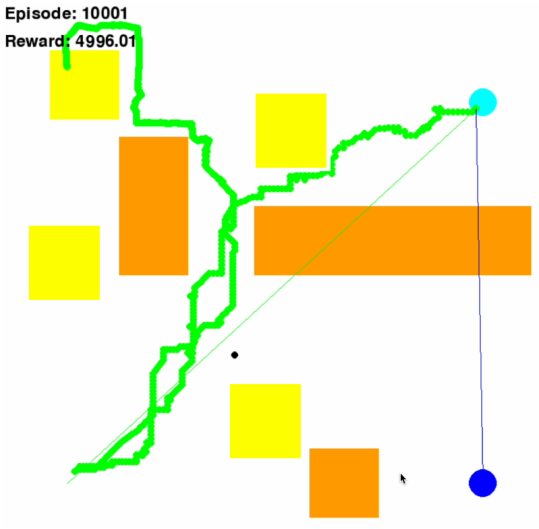

Fig. 10. Real-time decision-making by the UAV, returning to the depot

crucial for ensuring safe and efficient UAV operations in real-world scenarios.

E. Experimental Results

Will add Quantitative results here once they're ready.

The experimental results demonstrate the robustness and adaptability of the Q-learning algorithm in dynamic and uncertain environments. The UAV's ability to navigate through obstacles and avoid congestion zones showcases its real-time decision-making capabilities. As illustrated in Figure 6, the UAV effectively adjusts its path to avoid fixed obstacles, marked in orange, ensuring a smooth and uninterrupted journey. This dynamic path adjustment is crucial for operational efficiency and safety, particularly in environments with static barriers such as buildings and trees.

The UAV's proactive approach to avoiding congestion zones, marked in yellow in Figure 7, further emphasizes its adaptability. By recognizing and rerouting around these zones, the UAV minimizes delays and optimizes its travel time. This capability is essential in urban environments where traffic congestion can significantly impact delivery efficiency.

In Figure 8, the UAV demonstrates its ability to reach the customer's location, marked in green, despite the dynamic nature of the environment. This successful delivery highlights the UAV's proficiency in navigating complex scenarios and making optimal decisions to ensure mission completion.

The UAV's decision-making extends beyond reaching the customer. After completing the delivery, as shown in Figure 8, the UAV decides to proceed to an emergency landing spot, illustrated in Figure 9. This decision underscores the UAV's ability to prioritize safety and respond to unforeseen circumstances effectively. Finally, the UAV's choice to return to the depot, depicted in Figure 10, ensures it is prepared for future missions, demonstrating efficiency in mission turnover.

Overall, the Q-learning algorithm enables the UAV to adapt to various challenges, making informed decisions that optimize operational efficiency and ensure mission success. The visualizations highlight the UAV's capability to handle real-time changes, avoid hazards, and achieve its objectives in a dynamic environment. These results suggest promising applications for UAVs in logistics, surveillance, and other fields requiring autonomous navigation and decision-making.

VII. Future Work

Advanced RL Algorithms: Exploring advanced RL algorithms, such as deep Q-networks (DQN), could improve the drone's decision-making capabilities and handle more complex environments.

Real-World Testing: Conducting real-world tests to validate the effectiveness of the Q-learning approach in practical scenarios. This would involve deploying drones in real environments and observing their performance.

Multi-Agent Systems: Extending the approach to multi-agent systems, where multiple drones collaborate to complete tasks. This would require coordination and communication strategies to ensure efficient operation.

Will update this more accordingly

VIII. Conclusion

Need to add Quantitative results. Comparison with Greedy Algorithm

Improving this

This paper presents a novel approach to UAV decision-making using Q-learning, specifically designed to handle situations where communication with the delivery truck is lost. By integrating reinforcement learning and a Drone Trajectory Prediction (DTP) algorithm, the UAV is able to make real-time decisions autonomously, navigating around obstacles, avoiding congestion zones, and optimizing its route based on environmental uncertainties.

Our approach demonstrates the effectiveness of autonomous decision-making under disconnection scenarios, enabling UAVs to complete their missions efficiently. The use of reinforcement learning allows the UAV to continuously learn from its environment, improving decision-making capabilities over time. The integration of networking protocols, such as V2X or MANETs, further ensures that the UAV can quickly resume communication with the truck when network stability is restored, offering a robust solution for real-world dynamic environments.

Acknowledgments

The project was made possible through the support of the National Science Foundation (NSF). WIll update this more

References

[1] J. Silva *et al.*, "Reinforcement learning framework for autonomous drone navigation," *Journal of Autonomous Systems*, 2018.
[2] L. Yang *et al.*, "Optimizing drone delivery routes using q-learning," *Urban Systems Journal*, 2019.
[3] S. Lee and J. Kim, "Safe landing strategies for drones using reinforcement learning," *Safety Systems Review*, 2020.

[4] C. Qu *et al.*, “Obstacle-aware and energy-efficient multi-drone coordination using rl,” *IEEE CNSM*, 2021.
[5] ——, “Energy-aware multi-drone networks for task efficiency,” *IEEE TNMS*, 2023.
[6] R. Singh *et al.*, “Multi-uav coordination in complex environments,” *Journal of UAV Systems*, 2021.