Educational Administration: Theory and Practice
2024, 30(4), 6230-6239
ISSN: 2148-2403
https://kuey.net/

Research Article

Optimizing Pesticide Application Via Drone Navigation: A Reinforcement Learning Framework

Pratyush Goyal*

*Symbiosis International School, Pune, Maharashtra, India.

Citation: Pratyush Goyal, (2024) Optimizing Pesticide Application Via Drone Navigation: A Reinforcement Learning Framework, *Educational Administration: Theory and Practice*, 30(4), 6230-6239
Doi: 10.53555/kuey.v30i4.2379

ARTICLE INFO

ABSTRACT

In modern-day agriculture, efficient and targeted pesticide application is paramount for protection of crops, environment sustainability and safeguarding the farmers' health. Reinforcement learning (RL) algorithms have shown promise in optimizing such tasks by enabling autonomous decision-making. This paper delves into the development of an environment that makes use of Reinforcement Learning techniques, specifically reward shaping, to train autonomous drones for efficient crop spraying. This research aims to investigate whether a Reinforcement Learning-based environment can be designed to effectively guide drones in navigating and spraying crops, effectively answering the critical question. By doing reward shaping, a reward function was discerned that optimizes for different and basic parameters crucial to farmers.

Keywords – drones, reinforcement learning, pesticide application, machine learning, simulation

1. INTRODUCTION

The world population is increasing day by day and projected to reach 9 billion people by 2050, and it's predicted that agriculture consumption will also increase [1]. In order to maintain global food production and ensure food security, agriculture is essential. To sustain high agricultural yields and protect the world's food supply, effective pest control and crop protection are necessary [2]. However, pesticide spraying is harmful for farmers as well as time taking and drones can be very helpful [3],[4]. Estimates suggest that globally 1.8 billion people engage in agriculture and most use pesticides to protect their food and commercial products [5]. Estimates also suggest that 385 million cases of acute pesticide poisoning occur globally every year [6]. In recent years the advancement of drones has shown great promise to address such problems in agriculture contexts, particularly for pesticide spraying [7].

A new development that has attracted a lot of attention is the use of drones in agricultural settings. Drones have the advantage of being able to manoeuvre through any difficult terrain, potentially making pesticide application more accurate and efficient, thus reducing waste and environmental harm [1]. Drones will undoubtedly offer a platform for accurate and focused pesticide application, reducing waste and adverse environmental effects. Effective autonomous control systems are needed, though, to fully leverage the potential of drone-based pesticide spraying [8]. Currently, manually controlled drones require expertise and involve complex control [8].

An emerging subfield under machine learning known as Reinforcement Learning has showed great promise in enabling drones to carry out difficult tasks, on their own [9]. Reinforcement Learning has garnered attention for its capacity to enable autonomous agents to learn and make decisions within complex and dynamic environments. It distinguishes itself from traditional supervised machine learning by enhancing learning through interaction with an environment, rather than relying on labelled data [10]. Reinforcement Learning is about learning from interaction how to behave in order to achieve a goal [10].

Reinforcement learning fosters learning through direct interaction with the environment, as opposed to its opponent, supervised machine learning, which depends on meticulously annotated datasets. This interaction is just as similar to how we humans learn by making mistakes, making it a crucial strategy in the effort to build machines that can learn and adapt completely on their own. The key book "Reinforcement Learning: An

Copyright © 2024 by Author/s and Licensed by Kuey. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Introduction" written by Richard S. Sutton and Andrew G. Barto, serves as a foundational stone for comprehending the fundamental ideas of this area of dynamic interaction.

First, under Reinforcement Learning lies the concept of an "Agent." An agent can be referred to as the “decision-maker” or the “learner”. Agents can be defined as entities capable of acting on the basis of their learned knowledge [10]. These agents interact with its surroundings known as the “Environment” and makes decisions, known as “Actions" with the ultimate goal of achieving a specific objective, which in this scenario would be spraying crops efficiently. The interaction takes place at each of a sequence of discrete time steps t [11]. At every time step, the agent receives a state S_t from the state space S and selects an action A_t from the set of possible actions in the action space A(S_t) [11].

The "environment" in the context of Reinforcement Learning encapsulates the entirety of the external factors, variables, and circumstances that an agent must deal with. It offers the setting that influences the agent's decisions and actions take place. The environment interacts with the agent by using a feedback mechanism called "rewards." Rewards could be positive or negative. These rewards serve as the guide to the agent, reinforcing desirable behaviours and guiding the learning process. One time step later the agent gets a numerical reward, R_{t+1} as a result of the previous action [11]. The agent will then find itself in another state S_{t+1} [11].

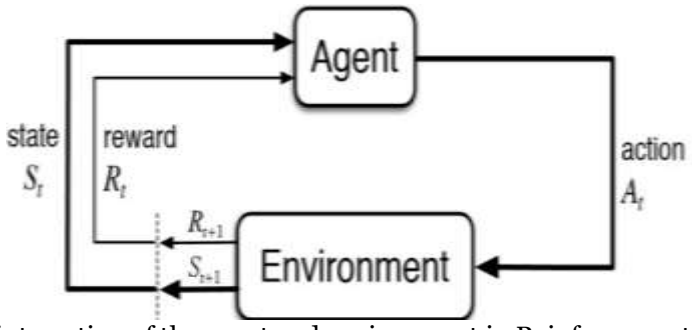


Fig. 1: The interaction of the agent and environment in Reinforcement learning [11]

Overtime, the agent learns through its interactions within the environment and overtime the agent also gauges which of its actions leads to most rewards. The agent attempts to maximise its rewards, and that is the exact final goal of reinforcement learning. The strategy the agent takes to decide which action to take is known as the ‘policy’ [11].

2. RESEARCH OBJECTIVES AND AIMS

Reinforcement learning systems can adjust and improve drone control tactics for effective pesticide spraying by learning from interactions with its environment. Given this background information, this research aims to answer the question: ***Can we design an environment that helps drones learn how to actually navigate an environment to efficiently spray crops?***

The solution to this research question will have a big impact on improving agricultural methods, to ensure efficient and robust spraying of pesticide on crop and assuring sustainable crop output.

Reinforcement learning is best suited for this scenario as the optimal action is not known to us and hence the optimal situation has to be learned using trial and error. If supervised learning were to be used, a labelled data set of the strategies for efficient spraying of pesticides would be required and obtaining this data will be difficult and may be expensive. Supervised learning demands, clear and annotated data for training, which may not be possible in this case. Unsupervised learning, on the other hand, focuses on determining patterns and relationships in absence of labels. It proves very valuable for dimensionality reduction; however, it will not be able to directly devise an efficient strategy for pesticide spraying. Reinforcement Learning allows one to model a scenario as an agent interacting with an environment, hence it is best suited in this scenario.

Having established the importance of drones in agriculture settings and the potential of Reinforcement Learning in enhancing efficiency of drones, it's imperative to delve deeper into the extant literature on the subject. The following Literature Review will examine prior research on reinforcement learning and its integration with drones in. Following this, the Methodology section will elucidate the experimental design and analytical techniques employed to answer the research question. The Results section will then present findings, offering insights into the efficacy of the proposed Reinforcement Learning environment for drones.

3. LITERATURE REVIEW

Due to drones' potential to change conventional agricultural methods, their application in agriculture has attracted growing interest. Numerous studies, in the past, have looked at the use of drones in different agricultural tasks. Some studies have highlighted the potential benefits of drone-based spraying, such as reduced pesticide usage, increased precision, and improved crop health [12][13].

Different algorithms have been researched for drone control in a variety of settings within the context of reinforcement learning. However, little to no research has been done on choosing the best reinforcement learning algorithm for increasing the effectiveness of pesticide spraying.

The following literature review section reviews the following 2 algorithms: DQN and A3C. The following 2 algorithms were reviewed in order to understand their functions and benefits and how they could be employed in this study. Another paper is also reviewed, where in Deep Reinforcement Learning (DRL) architecture is employed to make drones behave autonomously inside a suburb neighbourhood environment.

3.1. "Playing Atari With Deep Reinforcement Learning"

The authors of this paper studied the Deep Q-Network (DQN) algorithm, which demonstrated human-level performance on several Atari 2600 games. It showed how deep neural networks may be used in RL and sparked interest in using deep learning to solve RL issues. The research has paved a path for future research and possible applications in other fields. In this paper, the authors combined deep neural networks with reinforcement learning to create agents that could play several Atari 2600 games. The Q-learning function, Q(s,a) (Q value) predicts the total expected reward for taking an action 'a' in a state 's' and this can be represented in a table. However, when there are billions of such possible pairs, a table becomes too big and tabular methods become impractical use. The DQN algorithm combines deep neural networks with the Q-Learning algorithm to solve this problem.

The key equation in the paper is the Q-learning update, which states that the updated Q-value for a state-action pair is a combination of the immediate reward and the discounted maximum Q-value of the next sate-action pair as follows:

$$Q(s,a) = Q(s,a) + \alpha[r + \gamma \max a' Q(s',a') - Q(s,a)] \text{ [14].}$$

Where:
1. $Q(s,a)$ is the Q value of state 's' and action 'a' [14].
2. α is the learning-rate [14].
3. r is the immediate reward after action 'a' is taken in state 's' [14].
4. γ is the discount factor (value between 0 and 1) which is a parameter used to control how much importance the agent is placing on future rewards as compared to the immediate reward. [14].
5. s' is the next state after action 'a' is taken in state 's' [14].
6. a' is the action selected for the next state 's'' which maximises the Q-value [14].

The authors also show the differentiated loss function with respect to the neural networks' weights so that the gradient for updating the weights can be calculated.

3.2. "Asynchronous Methods For Deep Reinforcement Learning"

The authors of this paper expanded ideas from the aforementioned paper. This paper introduces the Asynchronous Advantage Actor-Critic (A3C) algorithm. This algorithm transforms the training of agents by utilizing asynchronous updates. A3C combines elements of the actor-critic methods with asynchronous updates, enabling many agents to interact with the environment independently and asynchronously update their parameters. The A3C has 2 components; the 'actor' which selects actions and the 'critic' that estimates the states.

The key equation in the A3C algorithm is the advantage function $A(s,a)$. This function depicts the advantage of taking action 'a' in state 's' in comparison to expected value of following the existing policy:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t) \text{ [15].}$$

Where:
1. $Q(s_t, a_t)$ is the Q value of state 's' and action 'a' [15].
2. $V(s_t)$ is the total expected rewards when beginning from state 's_t' and following the current policy [15].

Overall, by using multiple agents and asynchronous updates, A3C boosts the learning time taken by an agent.

3.3. "Drone Navigation And Avoidance Of Obstacles Through Deep Reinforcement Learning"

The authors focuses on using deep reinforcement learning techniques for enhancing navigation abilities of drones. To allow drones to navigate autonomously in a suburban neighbourhood setting, the paper suggests a Deep Reinforcement Learning architecture. Obstacles like trees, cables, parked cars, houses, and even other drones that move pose a threat in the simulated environment. Drones will be taught to recognize and avoid both stationary and moving obstacles.

A virtual geo-fence barrier's distance from the drone, its angle to the objective, and the elevation angle between the goal and the drone are all included in the drone's state. To create the full state, these scalar values are combined with the picture data. The training outcomes for three different training situations are shown in the paper: training with just the learner drone, training with the learner drone and one random drone, and training with the learner drone and two random drones. According to the findings, the drone becomes more adept at avoiding obstacles as training goes on, and it is able to safely achieve its target [11].

4. METHODOLOGY

To address the research question, a virtual simulation environment using OpenAI GYM was developed. The movement of a drone and its pesticide dispersion were simulated in the environment. Then, different reward functions were tried out and they were assessed based on different parameters.

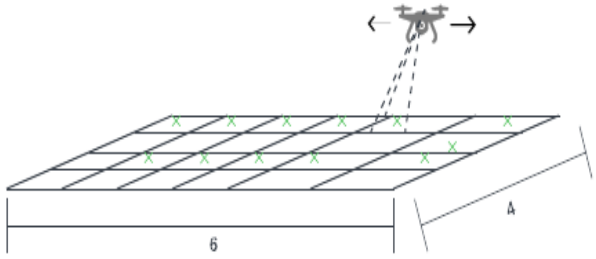
A custom Reinforcement Learning environment was created, where in the agent is the drone, which can learn and improve its performance through trial and error. Here is how the Reinforcement Learning environment was created.

The framework CleanRL, a deep Reinforcement Learning library, was chosen. And, the environment was created using the GYM framework, which is from OpenAI.

The environment grid size was set to 6 by 4, the number of crops were set to 12 and the maximum steps allowed per episode were set to 100.

Next, the action space and the observation space were defined. Observation space is the possible states the agent can perceive and the action space is the possible actions that the agent can take. The action space encompassed 9 discrete movements (8 movement actions and 1 spray action):

1. Spray - This action represents spraying pesticides on the current location of the drone. When the drone performs this action, it interacts with the tree at its current position on the grid (one of the 24 possible positions).
2. Up - This action moves the drone one step up in the grid, with the assumption that the drone is not already at the top boundary of the environment.
3. Right - This action moves the drone one step to the right in the grid, with the assumption the drone is not already at the right boundary of the environment.
4. Down - This action moves the drone one step down in the grid, with the assumption the drone is not already at the bottom boundary of the environment.
5. Left - This action moves the drone one step to the left in the grid, with the assumption the drone is not already at the left boundary of the environment.
6. Up-Right - This action moves the drone one step diagonally up and to the right in the grid, assuming the drone is not already at the top or right boundary of the environment.
7. Down-Right - This action moves the drone one step diagonally down and to the right in the grid, with the assumption the drone is not already at the bottom or right boundary of the environment.
8. Down-Left - This action moves the drone one step diagonally down and to the left in the grid, with the assumption the drone is not already at the bottom or left boundary of the environment.
9. Up-Left - This action moves the drone one step diagonally up and to the left in the grid, with the assumption the drone is not already at the top or left boundary of the environment.



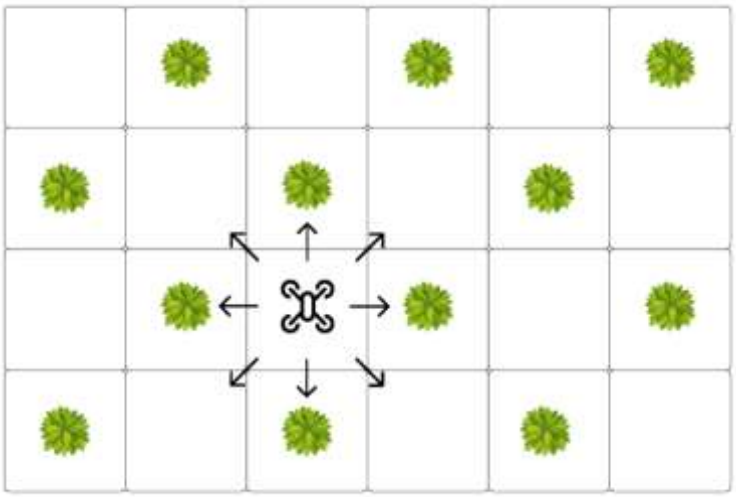


Fig. 2: The Action Space of the drone environment with sample tree positions

With the help of these actions, the drone can move across the grid and interact with the trees by spraying them or moving to the next cells. To learn how to effectively spray the crops in the field, the agent can select these actions at each time step.

The observation space provides a binary representation of the environment grid and it encompasses the positions unsprayed trees, sprayed trees and the drone's current location. This encompassed data is used as an input to the neural network of the reinforcement learning agent, guiding its decision-making process for which action to be taken in each time-step.

The drone is spawned at a random position within the grid at the start of each episode, the positions of the trees are randomized at the beginning of each episode. These initial conditions and randomizations are typical in reinforcement learning environments to encourage the agent to learn a general strategy that works across various scenarios rather than memorizing a specific pattern. It ensures that the learned policy is robust and can generalize well to new situations, which is essential for real-world applications where conditions can change dynamically.

Next, a primary reward function was crafted for the environment, to get the agent to learn the behaviour that is sought. The reward function is designed to stimulate the drone to make optimal decisions. Positive rewards for minimizing pesticide consumption while ensuring proper coverage were given and negative rewards for incorrect sprayings (excess sprays, wrong targets) were given.

Lastly, the DQN algorithm was imported. Given the nature of the environment, the Deep Q-Learning (DQN) algorithm was chosen. DQN is observed to effectively combine Q-learning with deep neural networks, offering a solution for state spaces like this environment. Key features of DQN such as experience replay and the use of a target network offer a stable and good training process. Additionally, DQN has demonstrated its capability in handling complex tasks in various OpenAI Gym environments.

The primary reward function and modified variants of the reward function was trained three times with different random seeds to ensure consistency. Then, the mean and variance of results across the seeds was computed for each reward function variant. Finally, the data was visualised.

5. REWARDS SHAPING AND METRICS

In Reinforcement Learning, reward shaping is crucial because it acts as the cue that directs agent behaviour. Its importance is particularly apparent in fields where agent actions have complex consequences, such as when controlling a drone to effectively spray trees.

The primary experiment employed an initial reward function as follows:

1. Initial reward for any action: For any action taken by the agent, there is a small step cost (penalty) of -0.01. This encourages the agent to complete the task using fewer steps.
2. Spray Action: If the agent chooses to spray and the cell contains an unsprayed tree, the tree is sprayed and the agent receives a reward of 100. However, if the cell does not contain an unsprayed tree (either empty or already sprayed), the agent receives a penalty of -1.

3. Movement Penalty: For movement actions there is a penalty of -0.1. This incentivizes the agent to move less and spray more efficiently.
4. Termination Penalty: If the agent has not sprayed all the trees and the maximum steps is reached, the agent receives a significant penalty of -200.
5. Completion Bonus: If all trees are sprayed before the maximum steps are, the agent receives a large bonus of 500.

The DQN agent was allowed to go through 20,000 steps of interaction ('total timesteps') with the environment during its training process. Agents were trained with a constant learning rate of 0.00105 (the learning rate for the optimizer in the DQN algorithm). A learning rate if 0.00105 implies that in each step of the training process, the weights are adjusted by a small fraction (0.105%) of the gradient of the loss function. Each 'episode' lasted for a maximum of 100 steps unless otherwise terminated earlier by other conditions.

Alternative Reward Functions: This research aimed to explore diverse reward strategies that maximise the efficiency of the drone:

Varying Movement Penalty: Experimenting with different penalties for unnecessary actions to analyse drone exploitation versus exploration.
1. The penalties were increased for moving without spraying to discourage it from unnecessary sprays.
2. A penalty for moving without spraying was introduced to encourage the drone to be direct in its movements and not wander unnecessarily.
3. A large bonus was given to the drone

Consecutive Spray Bonus: Offering a bonus reward for consecutive successful sprays, motivating the drone to locate clusters of trees.

Diminishing Spray Rewards: In order to encourage strategic spraying, reduced spray rewards are included if the drone sprays many trees in quick succession.

Metrics: The results of the experiments of different reward functions were based on the Total Reward accumulated and the Total number of crops sprayed which is a direct reflection of the drones' performance in the environment.

6. RESULTS

6.1. Primary Reward Function:

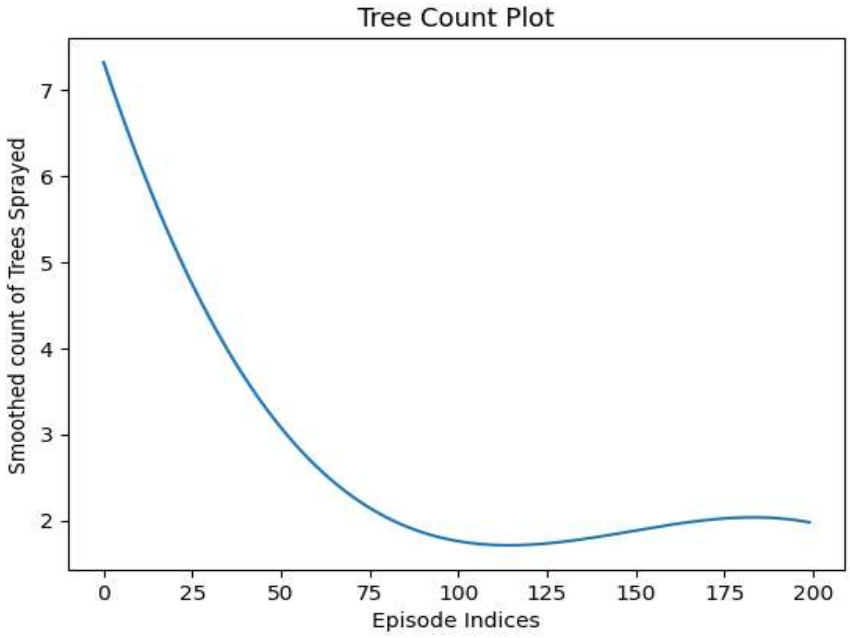


Fig. 3: Tree Count Plot for the Primary Reward Function

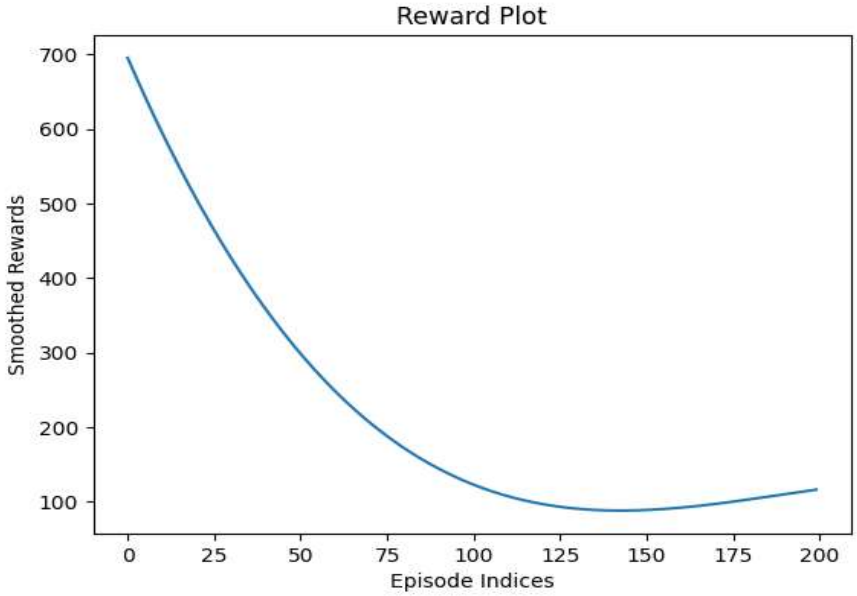


Fig .4: Reward Plot for the Primary Reward Function

6.2. Optimum Reward Function:

1. Initial reward for any action: An initialized reward of -0.01. A small penalty for each step, encouraging the agent to be efficient in its actions.
2. Spraying Actions: If the agent chooses to spray, the tree state changes from unsprayed to sprayed, and the agent receives a reward of 100. This is a significant positive reward for successfully spraying a tree.
3. Movement Actions: If the agent takes a movement action the reward is set to -0.1. This is a penalty for moving, which is larger than the step penalty, likely encouraging the agent to minimize unnecessary movements.
4. If the agent sprays a non-tree cell or an already sprayed tree, it receives a penalty of -1.
5. Termination Penalty: If the episode ends, and there are still unsprayed trees, the agent receives a penalty of -200. This is a significant penalty for not completing the task of spraying all trees within the maximum steps.
6. Completion Bonus: If the agent sprays all trees before reaching the maximum steps, it receives an additional reward of 500. This large bonus encourages the agent to complete its task efficiently.
7. Additional Modification: if the agent sprays a tree, the reward is increased by 100 + (10 * Number of Unsprayed Trees). This implies a higher reward for spraying trees when fewer unsprayed trees are left.

In summary, the reward function is designed to heavily incentivize the agent to spray trees, penalize (to a small extent) unnecessary movements and failure to complete the task within the maximum steps, and provide a significant bonus for completing the task efficiently.

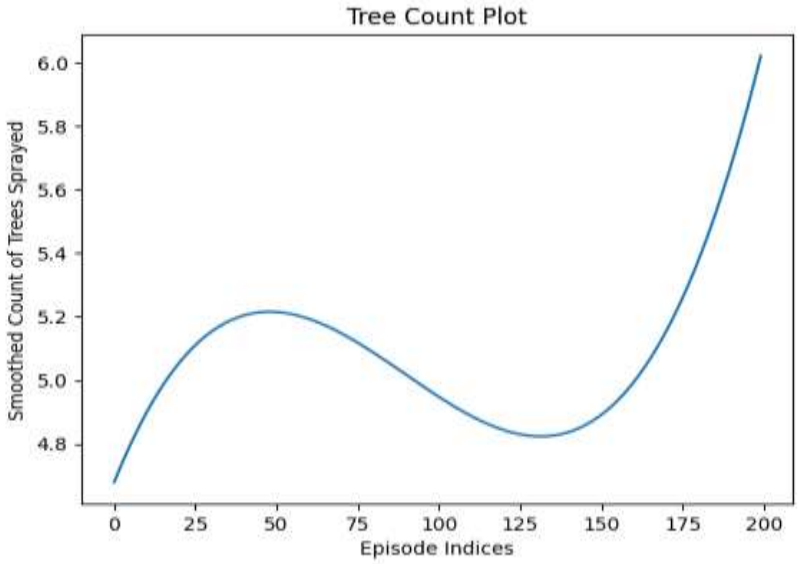


Fig. 5: Tree Count Plot for the Optimum Reward Function

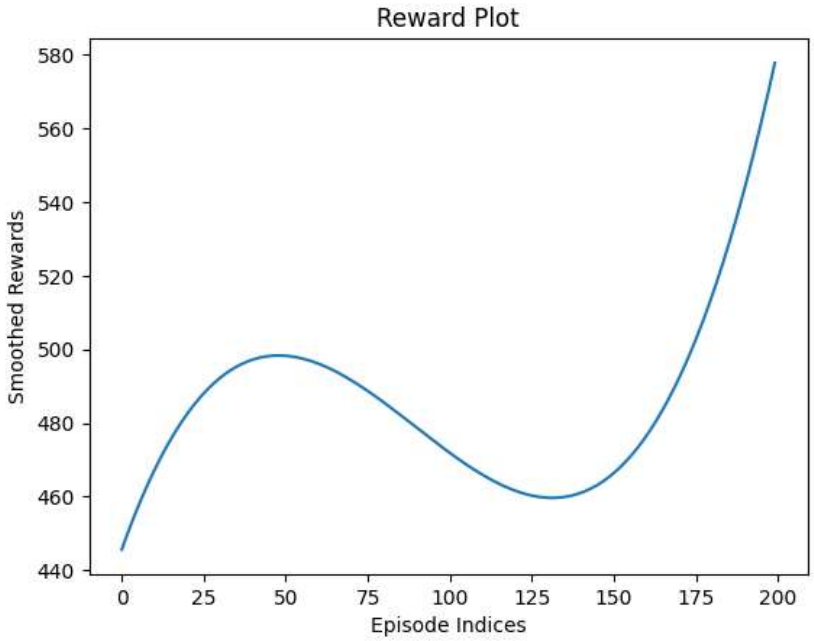


Fig. 6: Reward Plot for the Optimum Reward Function

7. DISCUSSION

7.1. Interpretation Of Results

The Optimum Reward Function's structure played a pivotal role in guiding the drone's learning behaviour within the simulated environment. The initial reward penalty of -0.01 for any action established a baseline where efficiency was paramount, discouraging unnecessary actions by the agent. This is evidenced in the early episodes, where the Tree Count Plot indicates a period of variability as the drone explores the environment to understand the consequences of its actions.

As the episodes progress, the graphs exhibit an upward trend in both trees sprayed and cumulative rewards, suggesting that the drone began to optimize its path to prioritize spraying actions. The Movement Action penalty further reinforced efficiency by incentivizing direct paths to unsprayed trees. This is crucial as it aligns with the practical need for minimizing resource expenditure in real-world crop spraying tasks.

The significant positive reward for spraying, set at 100 points, appears to have been the primary motivator for the agent, encouraging it to seek out and spray unsprayed trees. This is substantiated by the steep incline observed in the latter part of the Tree Count Plot. The additional modification to increase the reward based on the number of unsprayed trees left presents an interesting dynamic where the agent is encouraged to complete the task more swiftly as the opportunity for rewards diminishes, simulating a real-world scenario where efficiency would be rewarded.

7.2. Model Performance

The model's performance, as interpreted from the Reward Plot, indicates a learning curve where the drone's navigation strategy improved over time, maximizing the cumulative reward. This improvement suggests that the drone was able to distil and act upon the environmental feedback effectively, a testament to the well-structured reward function. The Completion Bonus and Termination Penalty are particularly noteworthy. The substantial bonus for completing the task before the maximum steps encouraged the agent to not only complete its objective but to do so in the most time-efficient manner. Conversely, the Termination Penalty discouraged the drone from leaving trees unsprayed, adding urgency to the task.

7.3. Implications

The findings bear promising implications for real-world agricultural applications. If the drone's virtual behaviours can be replicated in actual drones, it could lead to more efficient crop spraying, reduced resource wastage, and improved yields. However, the transition from a simulated to a real-world environment introduces variables not accounted for in the simulation, such as wind, equipment failure, and varying crop density, which must be considered.

7.4. Generalization

While the drone demonstrated an ability to learn and optimize its behaviour in a controlled environment, the real world presents a multitude of unpredictable variables. The early variability in performance may indicate a

learning phase where the model is still developing its strategy.

7.5. Limitations And Challenges

There is an inherent risk of overfitting with any machine learning model. The variability in the drone's early performance could indicate an initial instability in the learning process, perhaps due to an overfitting to specific patterns within the environment. Moreover, the robustness of the learning algorithm under unexpected scenarios, such as a sudden increase in the number of trees or introduction of new obstacles, has not been tested and represents a challenge for future work.

7.6. Further Research:

To further this research, several directions can be explored. Advanced reinforcement learning algorithms, such as Proximal Policy Optimization (PPO) or Trust Region Policy Optimization (TRPO) or with multiple agents may provide better stability and performance. Adjusting the reward structure to reflect a wider range of real-world conditions, such as variable tree density and different types of crops, could yield a more robust and optimum model. Incorporating environmental challenges like weather and terrain variability would also test the adaptability of the drone to real-world conditions.

8. CONCLUSION

This research emphasized the critical importance of clever reward shaping in Reinforcement Learning, particularly for complex and nuanced tasks like drone navigation and spraying. The performance-enhancing tactics through careful experimentation has been highlighted, providing a framework for further work in this area. This study provides evidence that a carefully designed simulated environment, coupled with a strategic reward function, can effectively train a drone to navigate and perform tasks such as crop spraying efficiently. The learning progression of the drone, as captured in our reward and tree count plots, underscores the potential of reinforcement learning in the field of precision agriculture. While more work is needed to ensure these behaviours transfer effectively to real-world scenarios, this research lays some direction for developing autonomous systems capable of supporting agriculture.

References

1. S. Ahirwar, R. Swarnkar, S. Bhukya, and G. Namwade, “Application of Drone in Agriculture,” International Journal of Current Microbiology and Applied Sciences, vol. 8, no. 01, pp. 2500–2505, Jan. 2019, doi: https://doi.org/10.20546/ijcmas.2019.801.264.
2. E.-C. Oerke and H.-W. Dehne, “Safeguarding production—losses in major crops and the role of crop protection,” Crop Protection, vol. 23, no. 4, pp. 275–285, Apr. 2004, doi: https://doi.org/10.1016/j.cropro.2003.10.001.
3. R. Desale, A. Chougule, M. Chowdhary, V. Borhade, and S. N. Teli, “Unmanned Aerial Vehicle For Pesticides Spraying,” International Journal for Science and Advance Research in Technology, vol. 5, no. 4, pp. 79–82, Apr. 2019.
4. F. G. Palis, R. J. Flor, H. Warburton, and M. Hossain, “Our farmers at risk: behaviour and belief system in pesticide safety,” Journal of Public Health, vol. 28, no. 1, pp. 43–48, Mar. 2006, doi: https://doi.org/10.1093/pubmed/fdi066.
5. M. C. R. Alavanja, “Introduction: Pesticides Use and Exposure, Extensive Worldwide,” Reviews on Environmental Health, vol. 24, no. 4, Jan. 2009, doi: https://doi.org/10.1515/reveh.2009.24.4.303.
6. W. Boedeker, M. Watts, P. Clausing, and E. Marquez, “The global distribution of acute unintentional pesticide poisoning: estimations based on a systematic review,” BMC Public Health, vol. 20, no. 1, Dec. 2020, doi: https://doi.org/10.1186/s12889-020-09939-0.
7. M. F. Aslan, A. Durdu, K. Sabanci, E. Ropelewska, and S. S. Gültekin, “A Comprehensive Survey of the Recent Studies with UAV for Precision Agriculture in Open Fields and Greenhouses,” Applied Sciences, vol. 12, no. 3, p. 1047, Jan. 2022, doi: https://doi.org/10.3390/app12031047.
8. Aleksandar Ivezić et al., “Drone-Related Agrotechnologies for Precise Plant Protection in Western Balkans: Applications, Possibilities, and Legal Framework Limitations,” Agronomy, vol. 13, no. 10, pp. 2615–2615, Oct. 2023, doi: https://doi.org/10.3390/agronomy13102615.
9. Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, “Autonomous Drone Racing with Deep Reinforcement Learning,” 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2021, Available: https://doi.org/10.1109/IROS51168.2021.9636053
10. R. S. Sutton and A. Barto, Reinforcement learning : an introduction. Cambridge, Ma ; Lodon: The Mit Press, 2018.
11. E. Çetin, C. Barrado, G. Muñoz, M. Macias, and E. Pastor, “Drone Navigation and Avoidance of Obstacles Through Deep Reinforcement Learning,” IEEE Xplore, Sep. 01, 2019. https://ieeexplore.ieee.org/document/9081749 (accessed Sep. 26, 2022).

12. M. H. M. Ghazali, A. Azmin, and W. Rahiman, “Drone Implementation in Precision Agriculture – A Survey,” International Journal of Emerging Technology and Advanced Engineering, vol. 12, no. 4, pp. 67–77, Apr. 2022, doi: https://doi.org/10.46338/ijetae0422_10.
13. J. Kim, S. Kim, C. Ju, and H. I. Son, “Unmanned Aerial Vehicles in Agriculture: A Review of Perspective of Platform, Control, and Applications,” IEEE Access, vol. 7, pp. 105100–105115, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2932119.
14. Volodymyr Mnih et al., “Playing Atari with Deep Reinforcement Learning,” arXiv (Cornell University), Dec. 2013, doi: https://doi.org/10.48550/arxiv.1312.5602.
15. V. Mnih et al., “Human-level Control through Deep Reinforcement Learning,” Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015, doi: https://doi.org/10.1038/nature14236.