Cognitive Robots Project Report

Autonomous Drone Defense Against Hostile UAVs

Abstract

In this project, we investigate the deployment of an autonomous drone agent designed to operate within a confined three-dimensional space. Our primary objective is to neutralize hostile entities, represented by Lebanese surveillance drones, which simulate potential real-world threats. The drone agent, capable of omnidirectional movement, is tasked with efficiently eliminating all enemy drones in the shortest possible time and distance while navigating complex and dynamic environments. The project addresses several key challenges, including stochastic/adversarial movement of the hostile drones, and obstacles within the confined operational space, further complicating the agent's path-planning and decision-making processes. Code for our experiments is available at https://github.com/wr0om/cogrobs-project. YouTube video showcasing our simulations is available at https://youtu.be/jioz5pRt4vw.

1 Introduction

Unmanned Aerial Vehicles (UAVs) or drones present an escalating security threat, pushing governments around the world, including Israel, to seek robust countermeasures. A recent response to this challenge is the "Iron Beam" laser defense system, developed by Rafael Advanced Defense Systems and Elbit Systems, anticipated to be operational by 2025 [2]. While promising, the significant cost of such systems—often reaching hundreds of millions of dollars per unit, with multiple units needed for comprehensive coverage—poses a major limitation. In this research, we propose an alternative, cost-effective solution inspired by the Iron Dome system [3]: deploying a friendly drone to intercept and neutralize enemy drones. Our approach leverages a ground-based radar system to detect nearby enemy drones and directs a friendly drone to intercept them through advanced path-planning algorithms. This research aims to address the following question: Can a drone be effectively utilized as a defense mechanism against hostile drones?

1.1 Environment and Agents

The agents in this problem are the friendly drone (in which we control), and the hostile drones. While the friendly drone is designed to act rationally and engage in strategic planning, the hostile drones exhibit varying degrees of randomness in their movement patterns. The behavior of the hostile drones can be categorized into one of three types: remaining stationary, moving within a sphere of fixed radius, or moving within a sphere whose center and radius changes stochastically over time. At the start of each simulation, a movement type is assigned to the hostile drones, and randomness is incorporated into each behavior. For stationary drones, their positions are determined randomly, for drones exhibiting spherical motion both the center and the radius of the sphere are randomly initialized, with dynamic adjustments made over time in the stochastic case. The environment in which we execute the simulations is a bounded three-dimensional space containing both the friendly and hostile drones. A demonstration of this setup is shown in Fig. 1.

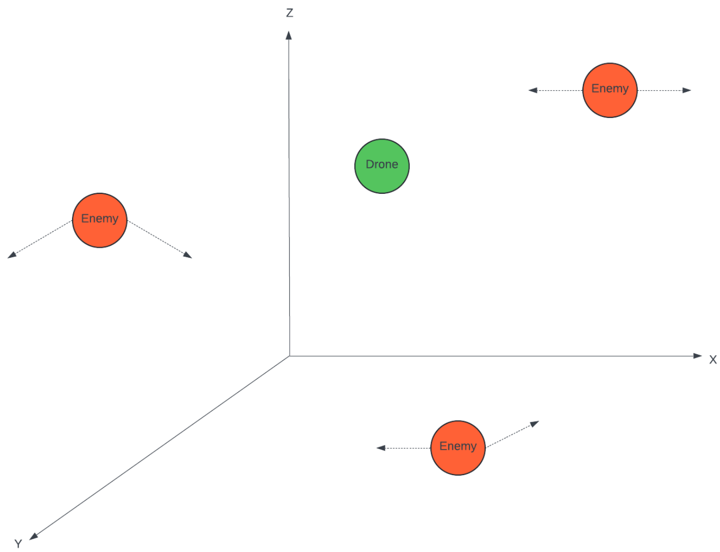

Figure 1: Workspace grid: hostile drones in red, our drone in green.

1.2 IDE

For our simulations, we employ the **Webots**[1] Open Source Robot Simulator, specifically version Webots R2023b, to model the desired environments. Additionally, we use the `bitcraze/crazyflie-simulation`[2] package for drone and drone controller simulations. This setup is applied to both our drone and the hostile drones, with necessary modifications tailored to fit the requirements of our experiments. Furthermore, we design custom simulation environments to incorporate the unique challenges relevant to our project.

1.3 Planning

Our approach begins with the implementation of a **Unified Path Framework** (UPF) to guide the friendly drone in neutralizing the hostile drones. This framework is used to define and solve the problem, modeling both the spatial and temporal aspects while accounting for the uncertainty and randomness in the movements of the hostile drones. However, we chose a non-temporal implementation for our problem, as it does not involve any specific constraints. By leveraging this framework, we generate a sequence of actions that not only optimizes the path toward each enemy drone but also adapts to changes in their positions due to their stochastic/adversarial behavior. This ensures that the friendly drone can navigate the environment effectively, taking into account the dynamic nature of the hostile drones' movements.

Throughout the mission, the friendly drone continuously engages in **Motion Planning**, dynamically adjusting its trajectory as it approaches each target. To navigate the environment safely, the drone employs a **Wall Following Strategy** to manage obstacles, maintaining a safe distance from walls or other static barriers to avoid collisions. To address the challenges of dynamic environments and stochastic movements, our approach incorporates a robust **replanning method** that dynamically updates the drone's trajectory during its mission. This method is triggered after the elimination of a hostile drone or upon the expiration of a predefined replanning interval of 6 seconds. By introducing replanning as a core operational feature, the drone ensures it continuously evaluates the environment, adapts to changing circumstances, and refines its path for maximum efficiency. A demonstration of this setup is presented in Fig. 2.

[1] https://cyberbotics.com
[2] https://github.com/bitcraze/crazyflie-simulation

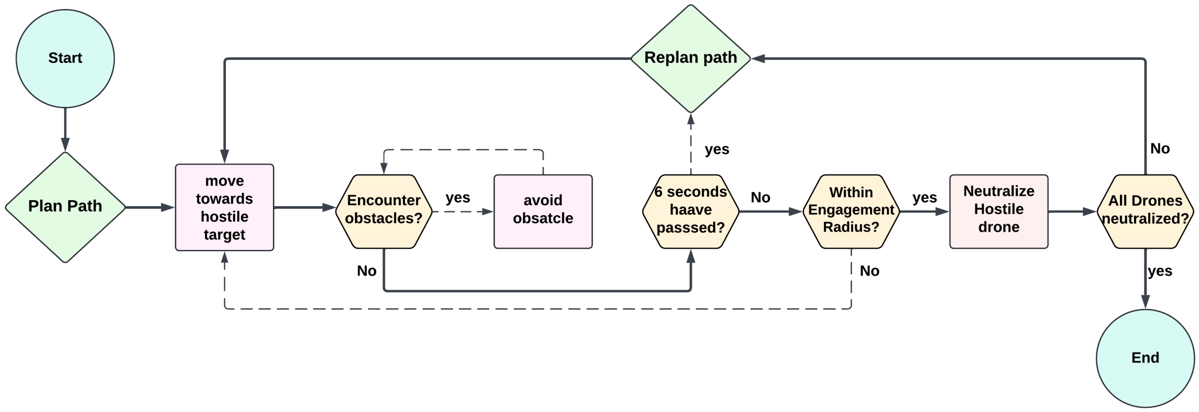

Figure 2: Mission Execution and Engagement Workflow.

2 Experiments

In our simulations, an enemy drone is considered neutralized when the friendly drone reaches within an ϵ-radius of it. In real-world applications, this proximity could represent various countermeasures such as deploying nets, initiating controlled collisions, or employing other active interception methods. While our experiments assume specific movement patterns for the enemy drones, these initial assumptions serve as a foundational framework that can be expanded to test more dynamic and unpredictable behaviors looking forward. We begin by outlining the challenges posed by each experimental scenario in Subsection 2.1, and in Subsection 2.2 we describe the obstacles encountered and the strategies employed to overcome them. Finally, in Subsection 2.3, we detail the adaptive planning techniques used to guide the drone in neutralizing its targets.

2.1 Complications

We designed a series of five complications, each introducing a progressively more challenging scenario within the original task of neutralizing enemy drones. In each complication, our objective is for a single friendly drone to neutralize all four enemy drones. Below, we outline each **complication scenario** in order of increasing difficulty:

1. Enemy drones are stationary throughout the scenario.
2. Enemy drones exhibit random movement within a fixed sphere of constant radius. For each drone, the center of this sphere is independently sampled from a uniform distribution: $x, y \sim \text{Uni}[-5, 5]$ and $z \sim \text{Uni}[1, 4]$. The radius of the sphere is also independently sampled, following $r \sim \text{Uni}[1, 5]$. All sampling occurs at the start of the experiment.
3. The characteristics of each sphere (center) / stationary point are resampled stochastically at intervals. Specifically, at each time interval $t \sim \text{Pois}(\lambda = 30)$, the center / point is redrawn using the same distributions as in the second / first scenario.
4. Enemy drones display **adversarial movement** by attempting to evade our drone when it approaches. This is modeled by equipping enemy drones with sensors that detect our drone within a range greater than ϵ. To ensure the success of our drone, we assume it has a matching velocity to that of the enemy drones and an advantage in path planning. While enemy drones start planning their evasive paths only when our drone is within a specific distance, our drone begins path planning from a greater distance, allowing it to accelerate more effectively and neutralize the targets.

5. The environment includes **obstacles** that both our friendly and enemy drones must detect and avoid. Obstacles represent regions that drones cannot pass through. We assume all drones are equipped with suitable sensors to detect obstacles in their path and adjust their routes accordingly.

2.2 Obstacles

To safely navigate between the obstacles we employ a **Wall Following strategy** [1], which involves following the wall until the obstacle ends, much like navigating in a maze. At first, we attempted a strategy of always turning left, but it proved ineffective because the obstacles were sparse and short, leading us to mistakenly think we were in a blocked area, which caused frequent loops. As a result, we adopted the **Polarity method**, where we turn left or right with respect to the polarity. In the polarity method, we first fix the polarity of the drone to go left and then after completing an obstacle avoidance maneuver, we change the drone's polarity to the opposite (from left to right for example) whenever it does not sense any nearby obstacles. This approach worked well in simulations, allowing us to navigate around obstacles, and ultimately became our preferred method.

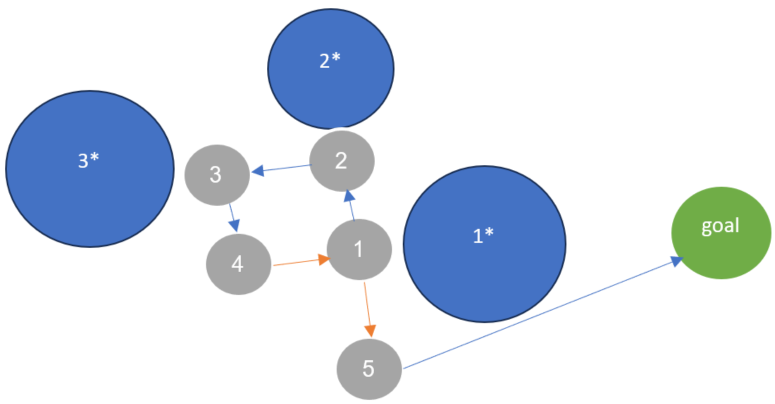

Figure 3: (1) At point 1, the drone encounters obstacle 1* and turns left. (2) After turning left, the drone encounters obstacle 2* at point 2 and turns left again. (3) It then encounters obstacle 3* at point 3 and turns left once more. (4) At point 4, the drone does not detect any obstacles, so it changes polarity and returns to point 1. (5) Upon reaching point 1, the drone changes polarity and turns right to avoid entering an infinite loop. (6) Finally, the drone reaches point 5, where it detects no obstacles and begins moving toward the goal.

Each drone was equipped with 16 sensors - 8 pairs, each pair spaced 45 degrees apart. One of the pair is pointed forward, while the other pointed downward. This arrangement was designed to prevent drones from being recognized as obstacles, as only one of the two sensor beams can detect the drone.

Our friendly drone detects an obstacle in its path in the following cases:

- When an obstacle is detected by a single sensor pair and the drone is in close proximity to it.
- When an obstacle is detected by two adjacent sensor pairs (spaced 45 degrees apart), provided that the sensor located 90 degrees from the current movement direction does not detect an obstacle.

When an obstacle is detected, the following steps are taken: (1) Stop moving and turn according to the polarity (90 degrees left or right from the current direction of movement). (2) Move in the polarity direction until all sensors stop detecting obstacles. (3) Once the path is clear, continue moving towards the original goal and change the polarity.

Since we are moving perpendicularly to the direction of movement, if the obstacle is not also perpendicular at the time of detection, a collision may occur after turning left or right. Therefore, when escaping from an obstacle and approaching a wall, we need to adjust our movement slightly perpendicular to our current direction (based on the polarity) to move away from the wall. If a closer sensor detects no obstacle in another direction, we adjust our movement accordingly.

2.3 Planning

We developed several distinct planning strategies to improve the path-finding capabilities of our friendly drone. These strategies are computed and managed by an offsite ground-based computer system, which has real-time access to the coordinates of each drone via radar or similar tracking systems. This ground-based system calculates optimal paths and transmits the planned routes to the friendly drone. **Re-planning** occurs every 6 seconds (based on experimental results) or whenever an enemy drone is eliminated, ensuring that the friendly drone remains on the correct path. Below, we outline the four planning strategies explored in this project:

1. **Baseline Strategy: Random Target Selection** The friendly drone randomly selects an enemy drone and moves toward it. This serves as a baseline for evaluating the performance of the more advanced strategies.
2. **Optimal Path Strategy: Distance-Based Cost Minimization** This strategy calculates the optimal path by using the Euclidean-Distance between the friendly drone and each enemy drone as the action cost. The path that minimizes this cost is selected, optimizing the time to neutralize the target.
3. **Centroid Estimation with Optimal Planning (Deprecated - did not work in practice)** This approach estimated the likely location of enemy drones by calculating a weighted average of their past positions, with greater weight given to more recent movements. These estimated centroids were then used to guide the optimal path planning strategy. However, this method was deprecated due to its reduced accuracy in our dynamic environment, as it failed to account for sudden changes in the enemy drones' movement patterns, leading to less reliable predictions and suboptimal paths.
4. **Radial Uncertainty-Aware Planning** This strategy accounts for the uncertainty in enemy drone movements when allowed to move only within a specified radius r around their current location. The radius r is estimated dynamically using the largest observed distance from the (also) dynamically estimated centroid. This uncertainty is integrated into the path-planning process, making it more adaptable to the unpredictable behavior of hostile drones.

These strategies were tested and evaluated to assess their effectiveness in improving the friendly drone's ability to neutralize hostile UAVs in various simulated scenarios.

3 Results

For each complication and planning algorithm, we ran the simulation three times using different random seeds and recorded the total time and distance covered by our friendly drone in each execution. A visualization of an example experiment can be seen in Fig. 4 (more visualizations can be seen in Appendix A). We then calculated the average time (in seconds) and distance (in distance units) across all three runs for each complication- results are summarized in Tab. 1. We chose not to present the confidence interval for each experiment, as the results are very setting-dependent.

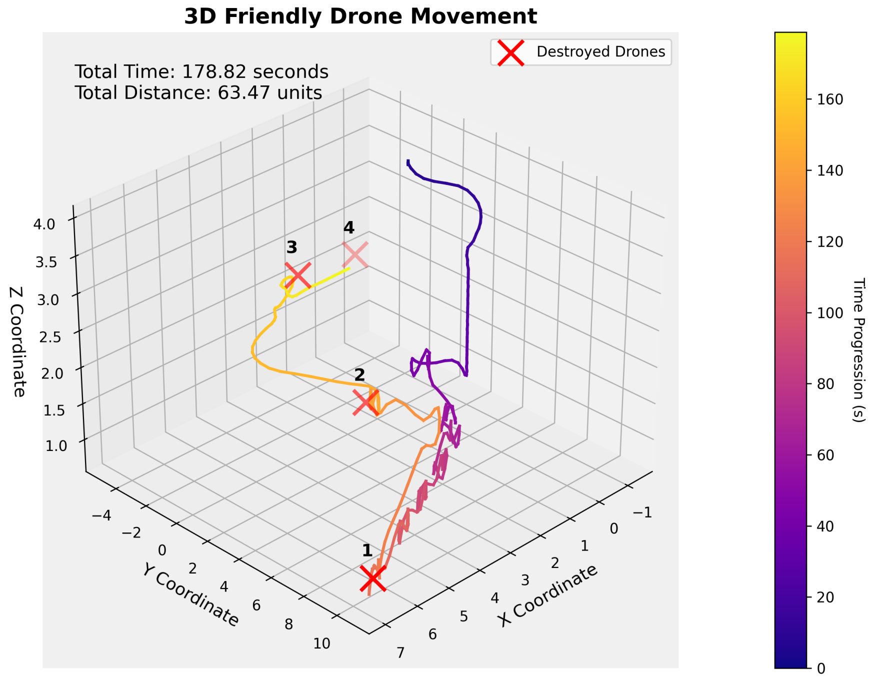

Figure 4: Visualization of our drone movement through time (indicated by color gradient). Drone eliminations are marked with red Xs with their order of elimination above them. This experiment is using complications 1 and 4 (inplace and adversarial), with planning 2 (optimal).

Experiment	Complication					Planning 1		Planning 2		Planning 4	
	1	2	3	4	5	Time	Dist	Time	Dist	Time	Dist
1	✓					76.63	32.9	**69.87**	**28.17**	**69.87**	**28.17**
2		✓				429.42	159.68	191.66	67.67	**181.08**	**64.46**
3	✓		✓			251.5	96.47	191.77	73.09	**153.43**	**60.99**
4		✓	✓			327.55	119.42	**223.94**	**69.18**	289.19	88.37
5	✓			✓		250.73	110.66	**142.49**	**53.27**	259.9	98.02
6	✓				✓	382.5	106.73	**284.86**	**81.78**	302.83	**81.78**
7		✓			✓	843.43	233.11	421.18	113.41	**284.86**	**88.74**
8	✓		✓		✓	402.37	106.5	**167.9**	**49.07**	294.06	86.95
9		✓	✓		✓	380.07	93.65	219.07	60.39	**187.75**	**58.55**
10	✓			✓	✓	577.28	156.42	**292.22**	**90.19**	318.02	95.35

Table 1: Summary of the average time and distance results for each experiment under different complications and planning strategies. The symbol ✓ indicates the presence of a complication.

As expected, Planning 1 (random) performed the worst in each experiment scenario - often by a big margin. Moreover, Planning 4 (radius uncertainty) did not always outperform the regular optimal planning approach.

4 Ablation Study

In this section, we investigate key factors influencing performance metrics, namely time and distance, within a controlled experimental framework. By dissecting these elements, we aim to better understand scalability and computational challenges.

4.1 Impact of Enemy Drone Count on Planning Algorithm Performance

Fig. 5 illustrates how the number of hostile drones affects the average planning execution time across varying enemy drone configurations. This analysis assesses the scalability of the planning algorithm, revealing that execution time grows non-linearly, trending toward exponential increases as the number of drones rises. This pattern underscores the computational complexity of scaling to larger scenarios and highlights the importance of optimization techniques for preserving performance in high-demand situations.

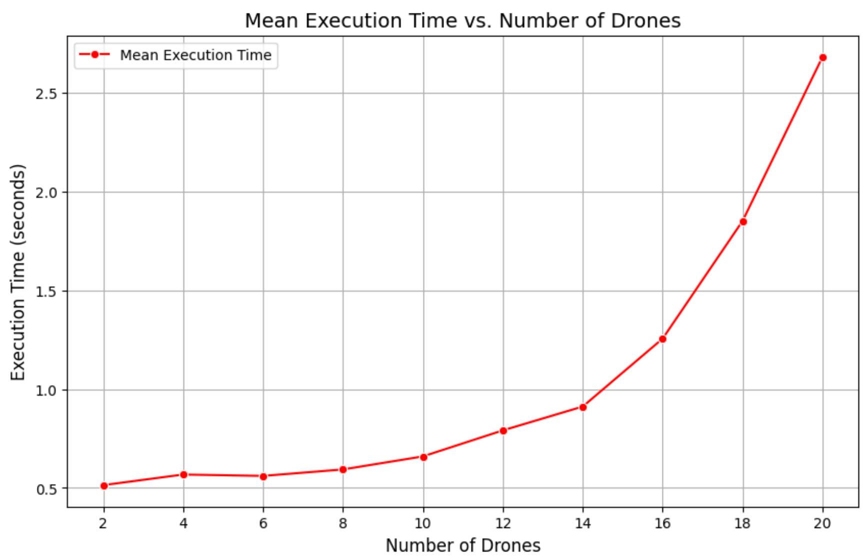

Figure 5: Relationship between the number of enemy drones and planning execution time.

4.2 Influence of Enemy Drone Count on Simulation Performance

Fig. 6 presents the effect of varying enemy drone numbers on two key performance metrics: execution time (left) and distance (right), averaged over three experimental seeds. The data reveals a linear increase in both metrics as the number of enemy drones grows. This trend suggests a promising potential for scaling simulations to accommodate larger numbers of drones while maintaining predictable performance outcomes.

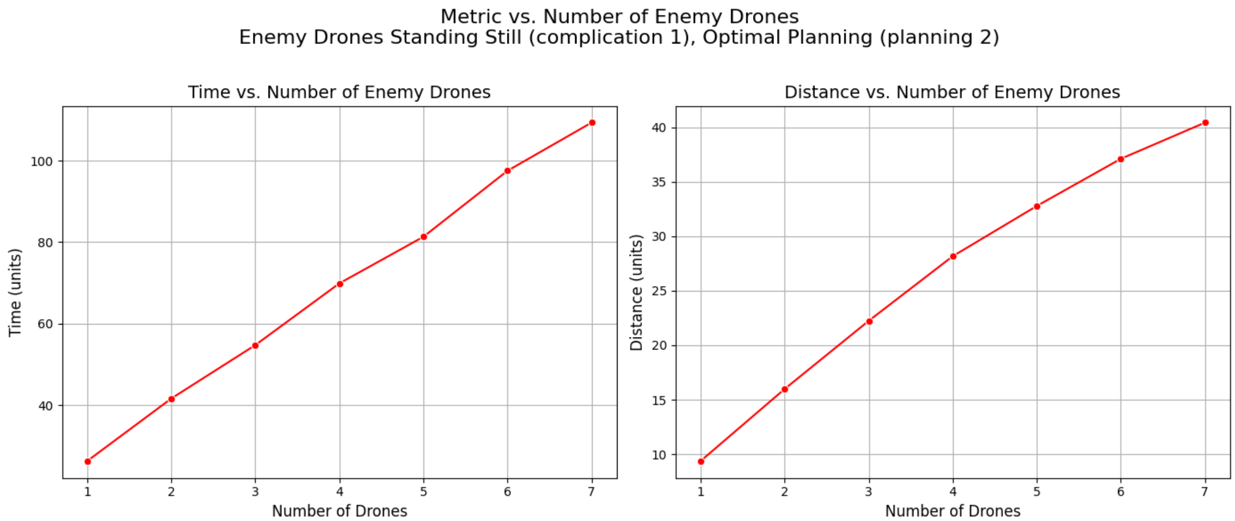

Figure 6: Effect of enemy drone count on performance metrics: time (left) and distance (right), averaged over three seeds. Results are based on an experimental setup using Planning 2 (optimal), with stationary enemy drones (complication 1).

5 Discussions and Conclusions

This research aimed to address a critical question in modern defense systems: can a drone effectively serve as a defense mechanism against hostile drones?

The results of our experiments, summarized in Tab. 1, demonstrate that our planning strategies (Planning 2 and Planning 4) significantly outperform the random planning approach (Planning 1) across all tested scenarios, including those involving obstacles. While the radial uncertainty-aware planning strategy (Planning 4) does not consistently surpass the regular optimal planning strategy (Planning 2), it shows notable improvements in specific cases—particularly in experiments 2, 7, and 9. These scenarios involved enemy drones moving within a defined sphere, where the approximated radius influenced performance and delivered better results across both evaluated metrics. Overall, both Planning 2 and Planning 4 achieved substantially better outcomes compared to the baseline random approach, confirming the effectiveness of strategic path planning. The results underscore the importance of incorporating intelligent path-planning algorithms to enhance interception capabilities, paving the way for more efficient drone defense systems.

Additionally, we conducted ablation studies to examine the effect of increasing the number of enemy drones on planning time and simulation performance metrics, specifically the total time and distance traveled by the friendly drone. The goal was to determine whether the problem could scale to include more drones and to identify whether the bottleneck arises from increased planning time or the need for the friendly drone to spend more time and travel further to reach additional drones.

Our findings in Section 4 reveal that planning time increases almost exponentially with the number of drones, whereas total time and distance in the simulation environment grow linearly. This disparity is likely due to the limited scope of our current simulation experiments, which involved a maximum of 7 drones. Within this range, planning times (Fig. 5) exhibit a nearly linear trend. However, we hypothesize that larger-scale scenarios, such as those involving 14 or more drones, would reveal the exponential growth pattern observed in planning times. These results indicate that while the current simulation framework scales efficiently for small numbers of drones, larger scenarios may face significant computational challenges. Overcoming these limitations is a critical area for future research to ensure the approach's scalability and real-world applicability.

The findings of this research demonstrate that drones can indeed serve as a viable defense mechanism against hostile drones, provided certain technological and operational conditions are met. Through the development and testing of path-planning algorithms and integration with ground-based radar systems, our results indicate that a friendly drone can reliably intercept and neutralize incoming hostile drones in simulated and controlled scenarios. This capability highlights the potential for cost-effective and adaptable solutions that complement or even partially substitute for more expensive defense systems. To transition this approach from simulation to practical applications, future research should focus on optimizing detection algorithms, reducing response latency, and validating the system under real-world conditions.

The project **workload** was distributed evenly among the team members, with each focusing on complementary aspects of the research. Rom and Yarden developed the simulation framework, ensuring accurate drone movement and the integration of environmental complexities. Ben led the planning experiments, testing various strategies, while Eran implemented the obstacle avoidance procedure and contributed largely to the additional complications, ensuring that the drone could adapt to unpredictable stochastic movement. The entire team collaborated on finalizing the submission report, the GitHub repository, and the accompanying YouTube video.

References

[1] Y. Hu, Q. Zhang, L. Qin, and Q. Yin. Escaping depressions in lrts with wall following method. *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 01: 134–138, 2017. URL https://api.semanticscholar.org/CorpusID:11967563.

[2] R. A. D. Systems. Iron beam, 2024. URL https://www.rafael.co.il/system/iron-beam/. Accessed: 2024-11-07.

[3] R. A. D. Systems. Iron dome. https://www.rafael.co.il/system/iron-dome/, 2024. Accessed: 2024-11-07.

A Additional Visualizations of Experiments

This section provides supplementary visualizations from the experiments discussed in the main text.

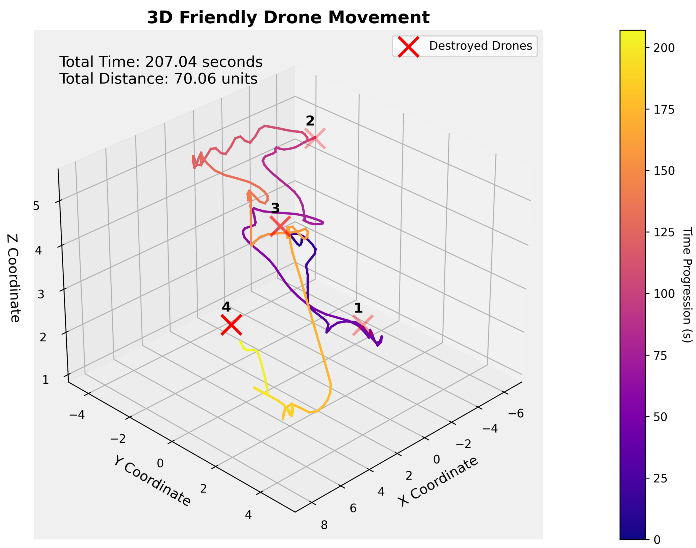

Figure 7: Visualization of an experiment described in Section 2, featuring **Complication 2** (movement within a sphere) and **Planning 2** (optimal).

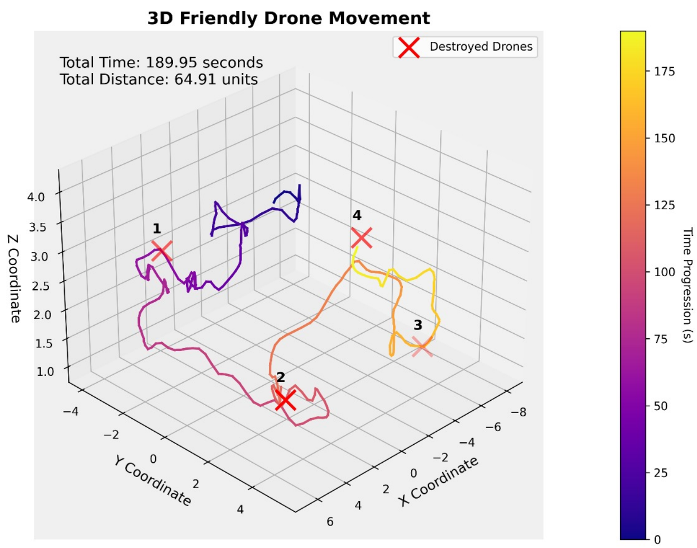

Figure 8: Visualization of an experiment described in Section 2, involving **Complications 2 and 5** (movement within a sphere and obstacles) with **Planning 2** (optimal).

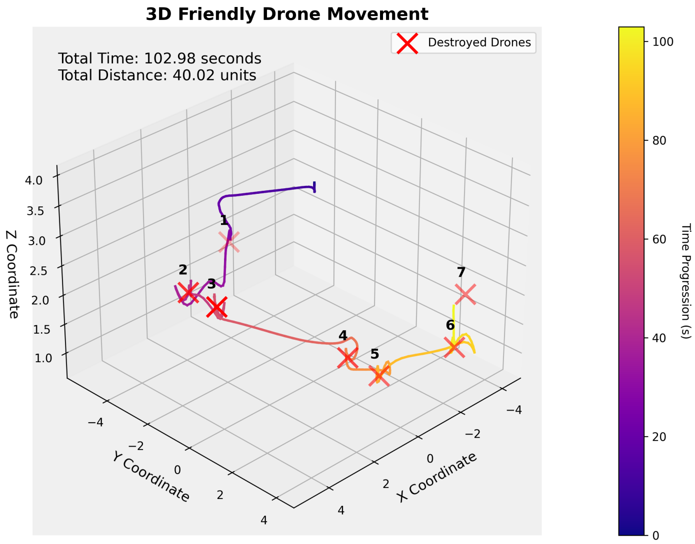

Figure 9: Visualization from the ablation study in Section 4, featuring **7 drones**, **Complication 1** (stationary), and **Planning 2** (optimal).