3D Simulation of Evasive Maneuvers for Autonomous Decentralized UAV Swarms

Duke Tran*
College of William & Mary
Williamsburg, VA, USA

Rafer Cooley†
University of Wyoming
Laramie, Wyoming, USA

ABSTRACT

Unmanned Aerial Vehicle (UAV) systems are a promising technological development for the future as they become more readily accessible and affordable to consumers. They have a wide array of applications, ranging from agricultural data collection to natural disaster response to smart cities. With the rise in UAV use comes the increased risk of malicious UAV flight and security threats to legitimate UAV swarms. Depending on the design of individual UAVs, swarms may be vulnerable to wireless attacks such as jamming and data spoofing and/or physical attacks such as corralling or kamikaze attacks. The UAVs within this study were developed using the SHARKS Protocol, a set of algorithms that enable a UAV swarm to operate autonomously without reliance on wireless communications nor a centralized controller authority. Hence, UAVs implementing this protocol are, for the most part, only vulnerable to physical attacks that may affect the movement of the individual UAVs or the formation of the overall swarm. This study focused on implementing a method proven to work in 2D, the Dynamic Distance Ejection technique, to defend against an adversarial corralling attack in 3D. Experimental results indicate the 3D version of the Dynamic Distance Ejection technique is more robust against this particular physical attack.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have become more accessible to many as companies have began producing a variety of UAVs at different price points over the recent years. According to Hildmann et al. [3], civilians and hobbyists can nowadays find themselves affordable UAVs with a wide array of unique and powerful features. Yaacoub et al. [6] note that many commercial UAVs possess live-stream video and image capture capabilities. Since most are small and lightweight, they are more cost-effective than commercial helicopters and small aircraft. This rise in UAV availability is opening the door for widespread use of UAV systems in areas where it might be unsafe or ineffective for humans to work. One such area discussed by Hildmann et al. [3] is agriculture, where detailed information must be gathered in a timely manner as changing conditions could alter the collected data or measured quantities. UAV systems would enable this data collection through aerial means and without hindrances that humans may encounter while traversing by terrain. Coppola et al. and Hildmann et al. [2, 3] describe how UAVs could be deployed to assist in natural disaster response and management, whether it be to deliver resources to incident sites, secure hazardous material, or monitor traffic and urban infrastructure. Another significant application for UAVs lies with the concept of smart cities, which aim to merge innovative technology with urban infrastructure. Cooley, Wolf, and Borowczak [1] outline complex tasks of which swarms can undertake within these smart cities in an efficient manner with minimal human interaction, maximizing human safety. These tasks could include structural maintenance, locating and protecting injured citizens, and crowd control. As for civilian use, Yaacoub et. al [6] describe how UAVs could be applied to a number of areas including cinematography, tourism, and filming commercial ads.

In designing UAV systems, there are several constraints that one must take into account. According to Coppola et al. [2], the necessity for swarms over individual UAVs arises from a given UAV's limited flight time, sensing capabilities, and power capacity. The ability to fulfill complex tasks emerges from cooperation between constituent UAVs within a swarm, yielding efficient objective completion through parallel operation and the possibility for collaborative assignments. Regarding the expectations of a particular swarm, Coppola et al. outline three main facets: robustness, flexibility, and scalability. In other words, a swarm should be relatively stable through the loss or malfunction of individual agents, able to adapt its formation to accomplish different tasks, and capable of adjusting its size to meet the goals of the given task. The algorithms within this study have been developed with these points in mind, ensuring that the resultant UAV swarm is adaptable to different situations and functional through the loss of its constituent UAV. Additionally, UAV systems can be controlled in three different ways, classified by Yaacoub et al. [6] as remote pilot control, remote supervised control, and full autonomous control. The swarms within this study possess full autonomous control, signifying the capability of individual UAVs to make decisions without the need for human intervention. Hence, the UAVs' internal algorithms need to be comprehensive enough to guide them to their objectives, allow them to carry out the mission effectively, and space themselves out appropriately to prevent collisions.

With the rise in commercial UAV availability and the potential for future reliance on UAV systems, the research and development of UAV system security is quintessential to the success of UAV swarms. As discussed by Yaacoub et al. [6], UAVs can be vulnerable to numerous malicious attacks, including hijacking and data interference or interception through wireless ports or injections of malware. UAVs may also be vulnerable to jamming and spoofing if they are remotely piloted and rely heavily on wireless communication. Sensor inputs may also be a target to attackers who could manipulate parameters and send or spoof misleading data to deceive the sensors, potentially leading to manipulation of the UAV's behavior and movement. Worst of all, if an attacker were determined enough, they could send kamikaze UAVs to attempt to crash into and destroy legitimate UAVs. Aside from security issues, UAVs may also be prone to internal issues caused from technical failures, such as a battery depletion or a malfunctioning circuit board. Operational issues may also occur, varying from dysfunctional rotors to broken wings. Natural threats such as inclement weather or strong winds could also present danger to the integrity of UAV flight. Concerning the latter issues, one can only do so much to prevent them by checking their equipment before flights and scheduling missions in fair weather conditions. As for malicious attacks, these can very well be prevented given adequate research and preparation. This study

*e-mail: dtran@email.wm.edu
†e-mail: rcooley2@uwyo.edu

will examine one potential method for managing a physical (external) rather than wireless (internal) attack. Outlined by Wolf and Borowczak [4], in the case where a UAV piloted or programmed with malicious intent attempts to corral the formation, the UAVs must respond appropriately and prevent the adversary from interfering with their mission and disrupting their movement.

2 Related Works

The 3D implementation of the UAV swarm followed the work of researchers at the University of Wyoming. The autonomous and decentralized UAV system was organized by the SHARKS (Secure, Heterogeneous, Autonomous, and Rotational Knowledge for Swarms) Protocol set forth by Cooley, Wolf, and Borowczak [1]. The protocol is composed of two rules intended to organize the swarm around a central target and prevent individual UAVs from overcrowding or colliding with one another. Together, the rules imposed upon each UAV allow them to compose a cohesive swarm that revolves in unison around an assigned objective or target. The decision to attack, protect, or secure the objective is up to the implementer to specify. The protocol also provides a simple system for collision avoidance with each rule preventing the UAVs from moving to locations that are occupied by other UAVs or objects.

2.1 SHARKS Protocol

The algorithms of the protocol's two rules are outlined in Cooley, Wolf, and Borowczak [1]. The first algorithm, the Center Rule, moves the UAVs towards a central target. The δ parameter specifies the distance the UAVs must maintain from the target. In other words, it represents the radius of the spherical orbital of the swarm in 3D space. The ε parameter specifies the tolerance allowed for a given UAV to drift from the δ. Together, the two parameters form a stability region of $\delta \pm \varepsilon$ units from the target in which the UAVs should remain. The Center Rule is presented in Algorithm 1; let *dist* represent an agent's distance to the target, *loc* represent an agent's position in 3D space, and *c* represent the distance that an agent can move in one epoch to enact the Center Rule.

Algorithm 1 Center Rule Algorithm

```
1:  procedure CENTERRULE
2:      if δ − dist > ε then
3:          if loc − c = empty then
4:              move backwards c units
5:          end if
6:      end if
7:      if δ − dist < −ε then
8:          if loc + c = empty then
9:              move forwards c units
10:         end if
11:     end if
12: end procedure
```

The second rule, the Dispersion Rule, ensures that UAVs do not overcrowd or drift towards one another. It maintains equidistance between UAVs within the swarm and assists with UAV collision avoidance. In each epoch, a UAV will move at an angle of $(180+r)^\circ$ clockwise away from their nearest neighbor. This rotation allows for the UAVs to move apart from one another and consequentially revolve in a counterclockwise manner around the target. The 3D implementation was based on the 3D algorithm derived from Wolf, Cooley, and Borowczak [5] and required a slight modification to account for pitch adjustments. As a result, the rule also accounts for vertical movement in 3D space by checking if its altitude is higher than its nearest neighbor, in which case it would tilt its pitch downwards, and vice versa if it was lower. This keeps the agents at distinct levels of altitude, effectively forming a sphere rather than a band or torus around the target. The Dispersion Rule is presented in Algorithm 2; let *t* represent the angle that an agent should use to adjust its pitch, *d* represent the distance that an agent can move in one epoch to enact the Dispersion Rule, and *r* represent the angle of dispersion that determines the direction and speed of revolution. As aforementioned, *loc* represents an agent's position. The *pitch* value represents an agent's vertical rotation over the *x* axis.

Algorithm 2 Dispersion Rule Algorithm

```
1:  procedure DISPERSIONRULE
2:      Align heading/bearing to face nearest neighbor
3:      Rotate heading/bearing clockwise by 180 + r degrees
4:      if agent is higher than nearest neighbor then
5:          tilt pitch − t degrees
6:      else
7:          tilt pitch + t degrees
8:      end if
9:      if loc + d = empty then
10:         move forwards d units
11:     end if
12: end procedure
```

The two rules of the SHARKS Protocol can be observed in effect in both images of Figure 1. The red sphere at the center acts as the target, and the surrounding gray discs are the legitimate UAVs. The adversarial UAV is also a disc, but its color is white to distinguish from the legitimate UAVs. The white tracers represent vectors that the UAVs used to fulfill the Center Rule, and the red tracers represent vectors used to fulfill the Dispersion Rule. At a given moment in time, a UAV will not be displaying both white and red tracers since they will prioritize the Center Rule before the Dispersion Rule. Therefore, UAVs that display the red tracers are already within the stability region and can focus on dispersing from its nearest agents.

3 Evasive Maneuvers

As reported by Wolf and Borowczak [4], the SHARKS Protocol had been shown to be vulnerable to adversarial UAVs. These adversaries would disrupt the dispersion of the legitimate agents, resulting in a corralled formation in which the agents would be clustered more closely rather than being equidistantly separated. To overcome this corralling attack, Wolf and Borowczak developed a technique that involved "ejecting", or maneuvering a UAV outside of the stability region, to provide it with enough space to circumvent the adversaries. With enough epochs, a swarm should theoretically reach equidistant equilibrium with legitimate agents interspersed between adversarial UAVs and maintaining the ideal distance between one another.

3.1 Dynamic Distance Ejection

Wolf and Borowczak [4] developed three ejection techniques to combat adversarial attacks. The basic ejection technique specifies a particular percent chance for each agent to eject and a particular distance an agent can move towards the target seeking to gain enough clearance to traverse around the adversary as it returns to the stability region. The two other techniques are extensions of this basic technique; the Dynamic Distance Ejection technique allows agents to eject further from the stability region based on their errors (*err*) from the ideal distance, and the Dynamic Percentage Ejection technique adjusts the probabilities that the agents will eject based on their errors from the ideal distance. This study primarily focuses on the Dynamic Distance technique since it proved to be the most successful in combating adversarial attacks according to Wolf and Borowczak [4]. The distance to eject is determined using the following equation:

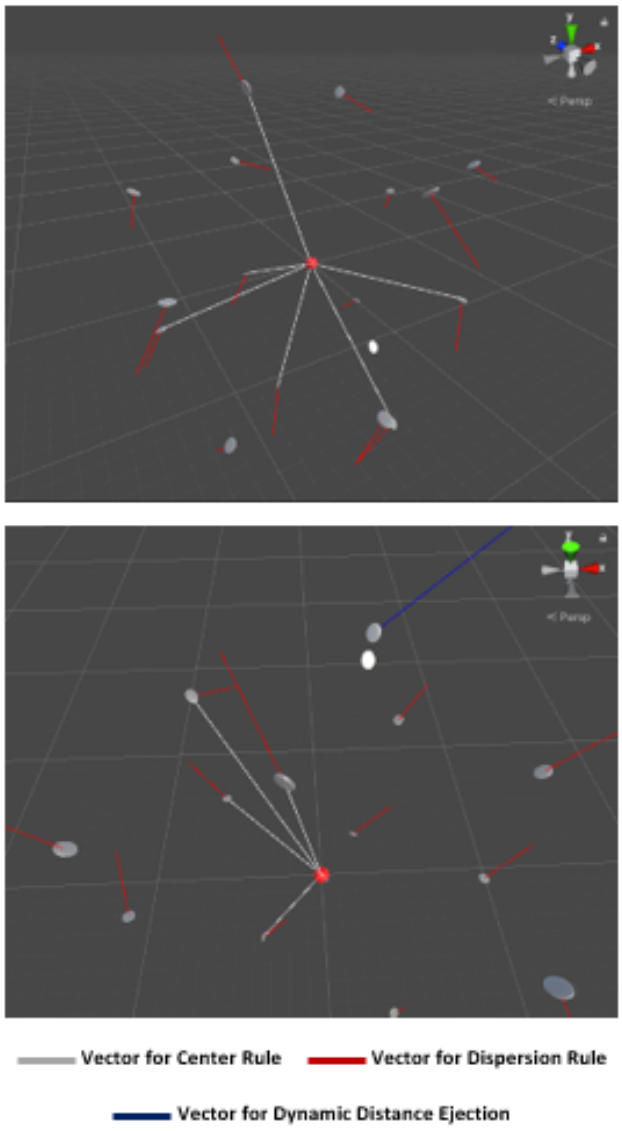

Figure 1: Images of a simulation ran in Unity 3D with a swarm population size of 16, an adversary present, and the Dynamic Distance Ejection technique implemented within the swarm. The top image depicts the adversary pursuing a legitimate agent. The bottom image depicts a UAV ejecting away from the adversary as the legitimate agent detects the adversary's presence.

$$Ejection\_Distance = (\delta/2) * err^{\frac{1}{4}}$$

The error is calculated as the difference between the average actual distances between the UAVs and their ideal distances. The ideal distance between UAVs is calculated using the following equation:

$$Ideal\_Distance = 2 * \delta * sin(180/num\_agents)$$

As defined earlier, δ represents the distance that each agent aims to maintain from the target, or the radius of the spherical orbital of the swarm. Since the error falls within the range of zero and one, Wolf and Borowczak [4] used the $\delta/2$ expression to weight the ejection distance so that it ranges from no ejection to ejecting halfway to the target (half the radius). Additionally, they exponentiated the error by $\frac{1}{4}$ to allow agents to eject further when they are experiencing denser congestion, since the error would be larger when the actual distances between agents are further from the ideal distance. One significant point to note is that the ideal distance equation was intended for a study conducted in 2D. This method of ideal distance calculation may be inaccurate for 3D space.

3.2 Algorithm

The 3D implementation of the Dynamic Distance Ejection technique was built off of the algorithm that Wolf and Borowczak [4] designed with one difference being the direction of ejection. Rather than ejecting towards the target, agents would face its nearest adversarial agent and eject backwards and to the left to attempt to evade the adversary. This change accounts for the different angles and positions that an adversary could approach a given agent in 3D space. Since an adversary with malicious intent could try to thwart, or worse, disable, a legitimate agent, it does not necessarily have to stay within the stability region and therefore does not have to be to the relative left or right of a given agent. In addition, the existence of a third dimension brings more unpredictability to the location and movement of an adversarial agent, and therefore, it is impractical to base an agent's ejection trajectory solely on the target. The Dynamic Distance Ejection technique is portrayed in Algorithm 3; let e represent the distance to eject, calculated using the above Ejection Distance equation. As established prior, loc represents an agent's position.

Algorithm 3 Dynamic Distance Ejection Algorithm

```
procedure DynamicDistanceEjection
    Align heading/bearing to face nearest adversary
    Calculate ejection distance
    if loc − e = empty then
        move backwards and to the left e units
    end if
end procedure
```

The Dynamic Distance Ejection technique can be observed in the bottom image of Figure 1. The UAV at the top of the image had detected an adversary in its presence, and therefore, it ejected following the blue vector to distance itself from the adversary.

4 Experiment

The primary subject investigated in this study was the effectiveness of the Dynamic Distance Ejection technique in the presence of an adversarial UAV in 3D space. The experiment consisted of three groups: no adversarial UAV present to act as the control group, an adversarial UAV present with no ejection technique implemented within the swarm, and an adversarial UAV present with the Dynamic Distance Ejection technique implemented within the swarm. The simulations were run for 10,000 epochs in Unity 3D[1] to simulate a 3D space without obstacles, weather, nor complex physical conditions (wind or gravity). Each group was simulated at three swarm population levels of 8, 16, and 32 UAVs. For each level, three trials were ran, yielding a total of 27 simulations[2].

At the start of each simulation, the UAVs and adversary, if present, would be initialized in random positions within a cubic area spanning from (-30, -30, -30) to (30, 30, 30). This area was designated to provide the UAVs with ample space to disperse and move towards the stability region. Moreover, this random initialization allowed the adversary to attempt to interfere with the swarm's formation before it had reached stability.

[1] Within Unity 3D, meters is the default unit for distance, meters/second is the default unit for velocity, and degrees is the default unit for angles.

[2] Once again, we note our strict time constraints that prevented us from running more simulations.

4.1 Parameters

Pertaining to the swarm's parameters, all UAVs possessed the same parameter values relevant to the SHARKS Protocol to ensure that they all functioned with algorithmic homogeneity. Below are the parameters that were suitable for the Unity 3D environment:

- **Orbital Radius (δ):** 12
- **Error Tolerance (ε):** 2
- **Center Rule Distance (c):** 4
- **Dispersion Rule Distance (d):** 3
- **Dispersion Rotation (r):** 20°
- **Dispersion Tilt (t):** 20°

Regarding the Center Rule and Dispersion Rule distances, Cooley, Wolf, and Borowczak [1] showed that a ratio of 4:3 was optimal for the proportion of c to d. In other words, for every 3 units an agent moved to fulfill the Dispersion Rule, it would have to move 4 units to fulfill the Center Rule. This ensured that agents prioritized reaching the specified radius (δ) around the target over dispersing from one another. Hence, they would be able to reach the stability region quicker without being overly concerned about getting further away from its neighboring agents. Similarly, Cooley, Wolf, and Borowczak found the dispersion rotation of 20° to be most effective in helping the agents reach equidistant equilibrium.

4.2 Security Metric

The level of swarm security was gauged using a metric quantifying the average distance between the UAVs. This metric can then be compared to the ideal distance between UAVs. Since the ideal distance equation has not been proven to be accurate for 3D space, the average distance of the baseline group could be assumed to be the ideal distance to which the experimental groups can be compared. The baseline group did not have an adversarial UAV present, so no external influence could have affected the formation and movement of the swarm. Thus, the average distance obtained from the baseline group is the closest value to a calculated ideal distance. The distance between any two given UAVs can be calculated with the equation below, where *loc* represents the position of a particular UAV:

$$Dist(a,b) = \sqrt{(loc_a^x - loc_b^x)^2 + (loc_a^y - loc_b^y)^2 + (loc_a^z - loc_b^z)^2}$$

Distances between every UAV within the swarm would be calculated using the *Cumulative_Distances* equation shown below, where *num_agents* represents the number of UAVs within the swarm:

$$Cumulative\_Distances = Dist(1, num\_agents) + \sum_{n=1}^{num\_agents-1} Dist(n, n+1)$$

The average distance between UAVs can then be obtained by dividing the cumulative distances by the number of agents within the swarm, shown in the following equation:

$$Average\_Distance = \frac{Cumulative\_Distances}{num\_agents}$$

This metric has its flaws, however, as the average distance of the baseline group can fluctuate due to adjustments made to fulfill the Dispersion Rule. Moreover, due to variability in the formation of the swarm and the UAVs' movement, it would be hard to obtain a consistent and exact number for this supposed ideal distance. A more depictive metric should employ an ideal distance obtained from an equation which would yield a consistent number dependent on the number of agents within the swarm[3].

[3]We were unable to timely formulate a method of calculating ideal distances between agents within the surface of a sphere.

A simple error metric was also calculated to measure the deviation of an experimental group's average distance from the baseline average distance. The error was calculated as follows:

$$Error = \frac{ExperimentalAvgDist - BaselineAvgDist}{BaselineAvgDist} * 100$$

4.3 Adversarial Attack

The focus of this study revolved around a supposed manned UAV attempting to thwart the swarm in 3D. To simulate a UAV piloted with malicious intent, the adversarial UAV was programmed with an algorithm to locate a legitimate agent at random and attempt to impede its movement or, if able, disable the legitimate agent. Let *loc* be the position of a given agent, legitimate or adversarial, and n be the number of epochs to wait before continuing pursuit of a new random legitimate agent. The epochs-to-wait parameter is necessary to prevent the adversarial UAV from endlessly pursuing a given agent and thus causing them to eject too far out from the stability region. While this may be the case in a real-world scenario, it is irrelevant in testing the effectiveness of the ejection technique. The Adversarial Pursuit Algorithm is presented in Algorithm 4; let a be the number of units that an adversarial UAV is capable of moving during a single epoch, which was set to be the maximum distance that a legitimate agent is capable of moving to simulate UAVs with similar capacities.

Algorithm 4 Adversarial Pursuit Algorithm

```
1:  procedure INITIATEADVERSARIALPURSUIT
2:      loc_legitimate = position of random legitimate agent
3:      Align heading/bearing to face legitimate agent
4:      Begin AdversarialPursuit
5:  end procedure
6:  procedure ADVERSARIALPURSUIT
7:      if loc_adversary = loc_legitimate then
8:          wait n epochs
9:          loc_legitimate = position of random legitimate agent
10:     else
11:         align heading/bearing to face legitimate agent
12:         move forwards a units
13:     end if
14: end procedure
```

Note that the algorithm does not account for collision avoidance, since an adversarial agent should be aiming to collide with and disable a legitimate agent if able. This adversarial pursuit would continue for the entire duration of each simulation, starting from when the UAVs are initialized until the simulation ends.

5 Results

The graphs from Figure 2 contained data averaged from the three trials ran per group for all 10,000 epochs. As can be seen in every graph, the beginning 1,000 or so epochs had a large decline in average distance, after which the distance would stabilize and fluctuate as needed. This decline can be attributed to the initialization of the UAVs in random positions. It is very likely that most of the UAVs were situated far apart from each other at the start of a simulation due to their random positioning, hence the large average distances. As they moved towards the target fulfilling the Center Rule, they got closer together and caused the average distance to sharply decline. As for Table 1 and Table 2, the data were only analyzed from 1,000 epochs and beyond to prevent the earlier values from skewing the errors upwards. The 1,000$^{\text{th}}$ epoch was chosen as the starting point following the assumption that the swarms would

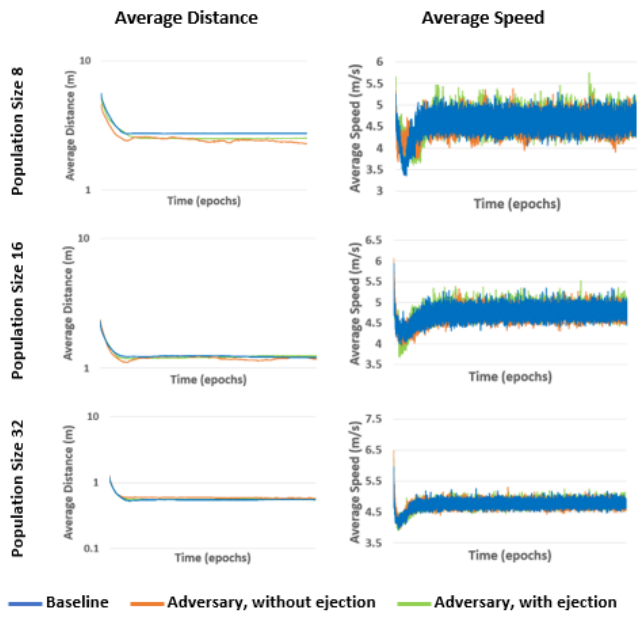

Figure 2: Plots of the average distances and average speeds between UAVs of the three experimental groups for population sizes of 8, 16, and 32.

have reached or been nearing the stability region[4]. This starting point was applied to the mean average distances as well as the mean average speeds to maintain standardization across the analysis.

Table 1: Computed statistics pertaining to average distance between UAVs of swarms of population sizes of 8, 16, and 32 for the three experimental groups.

Population Size		Baseline	Adversary, without ejection	Adversary, with ejection
8	Mean Average Distance (m)	2.743	2.441	2.525
	Standard Deviation (m)	0.007	0.078	0.052
	Error (%)	n/a	**11.003**	**7.944**
16	Mean Average Distance (m)	1.234	1.198	1.222
	Standard Deviation (m)	0.012	0.033	0.020
	Error (%)	n/a	**2.855**	**0.967**
32	Mean Average Distance (m)	0.548	0.588	0.556
	Standard Deviation (m)	0.004	0.007	0.004
	Error (%)	n/a	**7.386**	**1.581**

5.1 Ejection Efficiency

Based on various perspectives, the ejection technique was more effective in combating an adversarial attack than the absence of one. From Table 1, the mean average distances for the group with the ejection technique implemented within the swarm were closer to those of the baseline than the group without the ejection technique implemented within the swarm. For example, with a population size of 8, the no-ejection group had a mean average distance of 2.441 m, whereas the ejection group's mean average distance was 2.525 m. Of the two, the ejection group's mean average distance was closer to the baseline mean average distance of 2.743 m. Figure 2 also reflected this pattern, with the average distance graphs across all three population sizes depicting the ejection group's line closer to the baseline than that of the no-ejection group. These findings indicated that the ejection technique was more effective than no ejection technique regardless of the swarm size.

[4]A more rigorous approach would note the exact epoch that a given swarm has reached stability and conduct the analysis on that subset of data.

Table 2: Computed statistics pertaining to average UAV speed of swarms with population sizes of 8, 16, and 32 for the three experimental groups.

Population Size		Baseline	Adversary, without ejection	Adversary, with ejection
8	Mean Average Speed (m/s)	4.619	4.540	4.640
	Standard Deviation (m/s)	0.172	0.183	0.203
16	Mean Average Speed (m/s)	4.744	4.706	4.790
	Standard Deviation (m/s)	0.140	0.137	0.143
32	Mean Average Speed (m/s)	4.755	4.747	4.775
	Standard Deviation (m/s)	0.101	0.102	0.108

The errors also corroborated this general trend; the ejection group had a consistently lower error than the no-ejection group. Looking at Table 1, for the population size of 8, the no-ejection group had an error of 11.003%, whereas the ejection group's error was 7.944%. Noting the groups at population size 16, their errors were much lower than those of the other population sizes, with the no-ejection group's error at 2.855% and the ejection group's error at 0.967%. This could have been a result of not conducting enough trials, although it may also be the case that something peculiar was occurring at population size 16 that was difficult to concretely extrapolate. Regardless, those errors still supported the overall trend of the ejection group being more efficient than the no-ejection group at maintaining the mean average distance close to the baseline.

In addition, the mean average distances across the three experimental groups decreased significantly as the population size increased, naturally as a result of more UAVs occupying the limited space within the stability region. As can be seen in Figure 2, the average distances between UAVs converged towards the baseline as the population size increased. Since there were more UAVs within swarms of larger population sizes, the mean average distance would trend towards the baseline, mitigating the effects of the adversary and ejection technique on individual UAVs. This pattern was also exhibited in Table 1 through the consistent decrease in standard deviations within the no-ejection and ejection groups as the population size increased. For example, looking at the no-ejection group, the standard deviation went from 0.078 m for a swarm population of 8 to 0.033 m for 16 to 0.007 m for 32. Given these observations, the ejection technique's effectiveness, while still noticeable, was less significant as the population size grew and the stability region remained the same size (due to the constant radius, or δ).

5.2 Movement Variability

The ejection technique served to reduce the variability in the UAVs' movement, which was reflected in the standard deviations of the mean average distances. Table 1 shows how the standard deviations for the ejection group were markedly lower than those of the no-ejection group at all population sizes. Particularly at population size 32, the standard deviation of the ejection group was at 0.004 m while that of the no-ejection group was at 0.007 m, indicating greater variability in average distance between UAVs within the no-ejection group than the ejection group. The standard deviation of the ejection group was also equivalent to that of the baseline, which was 0.004 m, indicating that their variability in movement matched the level of the baseline group. In fact, the

standard deviations of the ejection group across all population sizes were much closer to the baseline than the no-ejection group. This indicated that the ejection technique successfully enabled the UAVs to adapt quickly to overcome the adversary, whereas the movement of the no-ejection group was more influenced by the presence of the adversary, hence their greater movement variability.

The data on the UAVs' speeds also supported the notion of the ejection technique reducing the movement variability. From Table 2, several patterns emerged regarding the mean average speeds of the no-ejection and ejection groups compared to the baseline group. For one, the ejection group consistently had higher mean average speeds than the baseline, while the no-ejection group had lower mean average speeds than the baseline. The groups at population size 16 demonstrated this pattern with the starkest clarity. Bearing in mind the baseline of 4.744 m/s, the no-ejection group had a mean average speed of 4.706 m/s while the ejection group attained a mean average speed of 4.790 m/s. The difference in the mean average speeds indicated contrasting behaviors between the two experimental groups. For the no-ejection group, their mean average speed was lower than the baseline since the adversary was able to obstruct the movement trajectories of the UAVs. There were also occurrences when the adversary UAV would succeed in colliding with and temporarily "disabling" a UAV by preventing it from moving, causing the average speed of the swarm to momentarily plunge downwards. The adversarial interference resulted in higher movement variability due to their need to take a more labored path to return to equidistant equilibrium. On the other hand, the ejection group's UAVs were able to proactively eject and evade the adversarial UAV before it could impede their movement or paralyze them. The ejection process facilitated the UAVs' swift and timely movement to evade the adversary's interference, and therefore the ejection group's mean average speeds were consistently higher than the baseline. The ejection group also experienced a reduction in movement variability as the UAVs were able to take shorter paths to return to equilibrium following the successful evasion of the adversary.

This proactivity was also reflected in the standard deviations of the UAVs' average speeds, shown in Table 2. Notably for the ejection group, the standard deviations were consistently higher than both the baseline and the no-ejection groups, signifying their quick and drastic changes in speed to eject away from the adversary. The ability to decisively react to the approach of the adversary enabled the UAVs with the ejection technique to maintain the appropriate distances from one another and thus retain their mean average distance close to the baseline, as assessed in Table 1.

The variability in movement and speed shared the same pattern with the mean average distance regarding convergence towards the baseline. While the mean average speed remained relatively stable across the different population sizes, Table 2 presented the standard deviation steadily declining across the three groups. This steady decline represents the inability of the UAVs to drastically change their speed as the stability region becomes more crowded, keeping in mind their requirement to fulfill the Dispersion Rule to maintain the appropriate distance between their nearest agents. Moreover, this decline in variability is observable in the graphs of Figure 2 with the fluctuations in average speed occurring less frequently as the population size increases.

6 Discussion

The simulations and resultant data showed that the Dynamic Distance Ejection technique worked quite efficiently in combating an adversarial UAV attack in 3D space. By ejecting UAVs away from an adversarial UAV within their vicinity, this technique provided the UAVs enough space to evade the adversary. More importantly, it allowed them to continue on their movement trajectories with minimal divergence from the ideal (baseline) distance between their nearest agents. As a result, the Dynamic Distance Ejection technique emerged as a powerful evasive maneuver against corralling attacks and enabled the SHARKS agents to maintain their formation and collective movement without excessive interference.

Within the larger scheme of UAV flight security, the ejection technique serves to defend against an external, physical attack. Since the SHARKS Protocol does not require wireless communications between UAVs, swarms that implement this protocol are essentially immune to wireless network attacks such as spoofing or jamming. The only attacks that may be of concern are physical attacks, such as the corralling attack examined within this study and those of the like, and specific wireless attacks including data spoofing attacks targeting onboard distance sensors. Through the promising results that have been duly collected and analyzed, the ejection technique proved to be a secure method of defense against a physical corralling attack in 3D space.

6.1 Limitations

There were various aspects within this study that were simplified or flawed which could be investigated or improved upon. Most significantly, the mode of data analysis relied heavily on an experimentally-obtained baseline distance between UAVs rather than an ideal distance calculated using a rigorous and well-tested equation. Future work could involve developing an equation to quantify the ideal distance between UAVs utilizing a technique to maximize coverage of a sphere. Due to the time constraints allotted to this study, we were unable to formulate a suitable and accurate equation to represent this ideal distance. We were also restricted in the number of simulations and trials that could be conducted due to the lack of time and researchers on the project. The number of adversarial UAVs was also simplified to one for the sake of time constraints, resulting in an immensely rudimentary investigation into adversarial attacks in 3D space. Further studies could be conducted with more adversarial UAVs, possibly even an adversarial swarm, to truly test the effectiveness of the ejection technique in 3D space. Finally, a large simplifying assumption implemented into this study was that the UAVs would have some capable method of detecting and differentiating an adversarial UAV from legitimate agents. Developing a consistent and thorough method for legitimate UAVs to identify an alien or adversarial UAV would massively benefit the SHARKS Protocol and UAV technology as a whole. One could look at wireless methods to accomplish this or installing and training machine learning algorithms to learn and detect human UAV maneuvering.

6.2 Future Work

Future work involving the SHARKS Protocol could be taken in various directions. This project involved preliminary work relating to dynamic targets and adaptive deltas. To ensure the robustness of the protocol, the Center Rule and Dispersion Rule must be developed to be able to follow a dynamic, or moving, target or objective. This can be achieved by internally keeping track of the target's location within each individual UAV and adjusting their movement and the eventual swarm formation towards the target. In situations where an adversarial attack with multiple UAVs is launched on a legitimate swarm, it might prove advantageous to adjust the individual UAVs' deltas (orbital radius) before relying on the ejection technique. They would be able to maintain their optimal distances from one another without sacrificing too many resources (i.e. battery power or rotor integrity) towards ejecting and moving back towards the stability region. One important consideration to keep in mind would be when to grow or shrink the delta and how the adversarial UAVs might respond to this change. If protecting the objective is of utmost importance, perhaps this method would be less secure than maintaining the absolute minimum delta from the target. The ability to adjust individual deltas could also potentially help with the problem of overcrowding, allowing the UAVs to spread out to form separate

stability regions rather than a shared one. Improvements could also be made upon the collision avoidance system to upgrade its reliability and sophistication. Rather than only checking if a given UAV would collide with another, a more sophisticated algorithm might also check for environmental obstacles such as walls, trees, or other structures and objects. One might also investigate swarm formations with obstacles within the stability region and adapting the swarm's movement around said obstacles.

7 Conclusion

The SHARKS Protocol had been proven to be effective in assembling and organizing a UAV swarm to revolve around a central target or objective. Within 2D space, the protocol was capable of withstanding an adversarial corralling attack using the Dynamic Distance Ejection technique. This study aimed to replicate those results in 3D space, albeit with a simplified adversarial attack involving a single UAV. Following the results, the Dynamic Distance Ejection technique demonstrated resolute effectiveness in defending against an adversarial corralling attack in 3D space, allowing the UAVs ample space to adjust around the adversary and maintain the ideal distance between neighboring legitimate agents. Moreover, this technique prevented the adversarial UAV from "disabling" or paralyzing any UAVs and ensured that the UAVs could continue their natural revolving movement around the designated target.

Acknowledgments

This study was generously funded and supported by the National Science Foundation (NSF) REU Site HUMANS MOVE joint research program with the University of Wyoming. We wish to thank Rafer Cooley and Hui Hu for providing invaluable mentorship and guidance throughout the duration of the program. We would also like to extend thanks to Dr. Mike Borowczak, who graciously introduced us to the SHARKS Protocol and presented us with various research ideas based on previous studies conducted by his team. Finally, we would like to thank Dr. Amy Banic for organizing and leading the HUMANS MOVE research program.

References

[1] R. Cooley, S. Wolf, and M. Borowczak. Secure and Decentralized Swarm Behavior with Autonomous Agents for Smart Cities. In *2018 IEEE International Smart Cities Conference (ISC2)*, pp. 1–8, 2018. doi: 10.1109/ISC2.2018.8656939

[2] M. Coppola, K. N. McGuire, C. De Wagter, and G. C. H. E. de Croon. A Survey on Swarming with Micro Air Vehicles: Fundamental Challenges and Constraints. *Frontiers in Robotics and AI*, 7:18, 2020. doi: 10.3389/frobt.2020.00018

[3] H. Hildmann, E. Kovacs, F. Saffre, and A. F. Isakovic. Nature-Inspired Drone Swarming for Real-Time Aerial Data-Collection Under Dynamic Operational Constraints. *Drones*, 3(3), 2019. doi: 10.3390/drones3030071

[4] S. Wolf and M. Borowczak. Defensive Strategies for Autonomous Decentralized Lightweight Swarms. In *2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin)*, pp. 1–6, 2020. doi: 10.1109/ICCE-Berlin50680.2020.9352161

[5] S. Wolf, R. Cooley, J. Fantl, and M. Borowczak. Secure and Resilient Swarms: Autonomous Decentralized Lightweight UAVs to the Rescue. *IEEE Consumer Electronics Magazine*, 9(4):34–40, 2020. doi: 10.1109/MCE.2020.2969174

[6] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab. Security analysis of drones systems: Attacks, limitations, and recommendations. *Internet of Things*, 11:100218, 2020. doi: 10.1016/j.iot.2020.100218