2017 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 13-16, 2017, Miami, FL, USA

Counter UAS Using a Formation Controlled Dragnet

Skyler Tolman and Randal W. Beard[1]

Abstract— **Rapidly developing UAS technology necessitates reliable counter UAS systems. This paper proposes a formation controlled dragnet as a possible solution and compares potential intercept algorithms that can be used in this scenario. Proportional navigation and target-predictive path planning, both existing algorithms, are explored and an original approach, Adaptive Radius Optimal Defense (AROD), is introduced. Simulation results are given and the strengths and weaknesses of each approach are discussed. Based on the simulation results, some advantages that AROD offers over other existing algorithms are listed. Possible improvements and future research directions are also suggested.**

I. INTRODUCTION

With a rise in UAV technology, there is increasing concern for security and defense against new threats [1]. Most proposed counter UAS technologies either utilize some form of ballistics, have risks of collateral damage, or use communication/signal disruption [2]. While signal disruption is a strategy that hold promise, UAS technology is becoming increasingly autonomous and therefore independent of external signals [3] and, accordingly, more immune to this form of counter UAS strategy.

This paper considers a possible solution to the counter UAS problem in which a team of UAS are used to physically intercept an intruding target using a formation controlled dragnet, as shown in Figure 1. This UAV interception strategy is an intriguing concept since it is non-ballistic, allows the intruder to be captured without imposing significant damage, and it offers a robust defensive option against threats that are autopilot controlled and not transmitting electronic signals. Accordingly, our approach to UAS interception is viable for environments ranging from dense urban areas to open rural terrain.

The algorithms explored in this paper are based on a simple scenario in which a single multi-rotor aircraft traverses a path toward the center of a protected area. A team of multi-rotor UAS, beginning at or near the home base, flies in a square formation, carrying a dragnet used to capture the intruding UAV. We use a virtual leader located at the center of the net to determine the formation's desired position and orientation in space. The defending UAS maintain formation about the virtual leader as it is controlled to follow a collision course with the intruding UAV. As the virtual leader collides with or passes the target within close proximity, the four UAS will envelope the target in a dragnet and would then return the intruding UAV back to home.

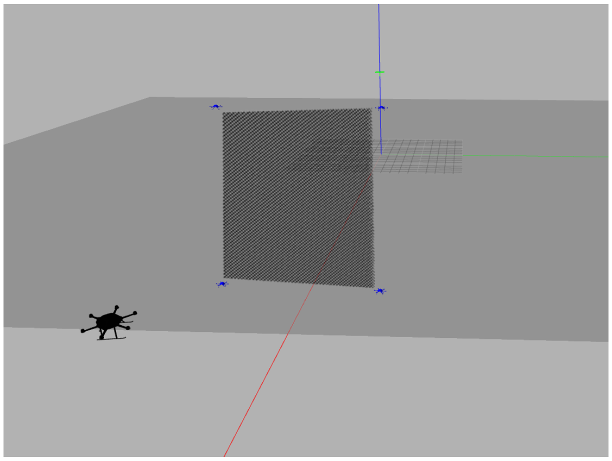

Fig. 1. Proposed counter UAS scenario in which a dragnet controlled by four UAS is used to intercept the intruder.

The primary focus of this paper is to analyze and compare three different possible strategies to minimize the miss distance as the virtual leader attempts a collision with the intruding UAV, thus maximizing the probability of successfully intercepting the target.

The three strategies are:

1) Proportional navigation, an algorithm commonly used for missile guidance systems [4],
2) A basic target-predictive waypoint path planner, and
3) An adaptive radius defense strategy.

The first two approaches are based on well-established techniques with minor modifications adapted to this particular problem. The third approach is an original algorithm developed particularly for the counter UAS intercept problem. This paper will introduce the adaptive radius defense strategy and compare the performances of the three algorithms.

II. PROBLEM STATEMENT

We will refer to the four multi-rotor and net system as "the fleet" and refer to the incoming multi-rotor UAV as "the intruder." The central location that the fleet is defending will be referred to as the "center of defense."

Because this paper is focused on the algorithm used for interception, we make some simplifying assumptions and do not address every aspect of the full system. We use the same assumptions and parameters for each intercept strategy to make the most reasonable comparison of the different intercept approaches, independent of other aspects of the scenario.

[1] S. Tolman is a research assistant, and R. Beard is with the Department of Electrical and Computer Engineering, Brigham Young University, Provo, UT 84602, OH 45435, USA. skyler237 at gmail.com, beard at byu.edu

978-1-5090-4494-8/17/$31.00 ©2017 IEEE

Assumptions:

- We assume there is no wind.
- We disregard the net dynamics and treat each UAS in the fleet as a dynamically independent vehicles, and perform formation control without modeling the physical interconnection that will be imposed through the net.
- We assume, unless otherwise stated, that the fleet of UAS and the intruder UAV are similar in speed and capability.
- We allow aggressive maneuvers for the intruder, but we assume that the intruder is not aware of the fleet, and that it does not intentionally move to avoid interception.
- We do not simulate sensors or estimation of states, but use global truth values for all position and velocity measurements.

We acknowledge that these assumptions do not capture the full complexity of the problem and limit the direct applicability to a real world implementation. This paper discusses a beginning point for further research and development of applicable algorithms. We discuss our findings thus far with the acknowledgement that further research should be done to more fully establish the real-world effectiveness of the methods discussed.

III. INTERCEPT STRATEGIES

A. ProNav (Proportional Navigation)

Proportional Navigation (ProNav) is a common intercept scheme used in missile guidance systems [5]. In this paper we will use a 3-D implementation of ProNav focused on UAV control [6]. The algorithm produces a desired acceleration perpendicular to the movement of the aircraft which is proportional to the normalized line of sight rate and the closing speed.

Let the position of the fleet be the center of the net represented by $\mathbf{p}_F$. Let the position of the intruder be represented as $\mathbf{p}_I$. The position of the i^{th} fleet vehicle is given by $\mathbf{p}_i = \mathbf{p}_F + \mathbf{h}_i$, where $\mathbf{h}_i$ defines the position in the formation of the i^{th} UAS. Define the line-of-sight vector as $\boldsymbol{\ell} = \mathbf{p}_I - \mathbf{p}_F$ and the line of sight rate as $\dot{\boldsymbol{\ell}} = \dot{\mathbf{p}}_I - \dot{\mathbf{p}}_F$, and let $L = \|\boldsymbol{\ell}\|$. Then the ProNav acceleration command is given by

$$\mathbf{a}_F = N\boldsymbol{\Omega}_\perp \times \mu \mathbf{v}_F,$$

where

$$\boldsymbol{\Omega}_\perp = \frac{\boldsymbol{\ell}}{L} \times \frac{\dot{\boldsymbol{\ell}}}{L},$$

and where

$$\mu = \frac{\|\dot{\boldsymbol{\ell}}\|}{\|\mathbf{v}_F\|},$$

and where $\mathbf{a}_F$ is the commanded acceleration for the fleet. A representation of the geometry is shown in Figure 2.

In order to make the ProNav equations applicable to the counter UAS scenario, we have made the following adaptations.

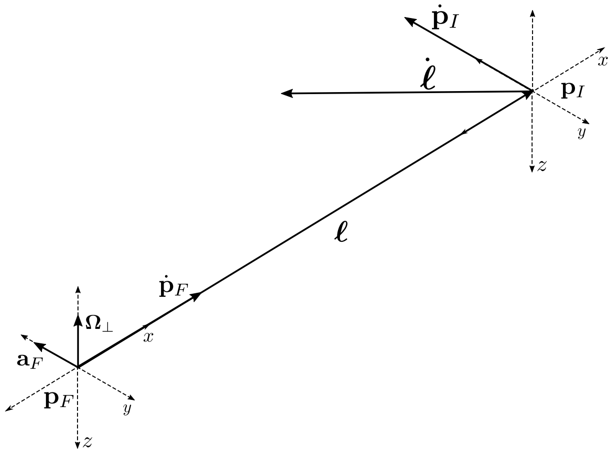

Fig. 2. Example of proportion navigation algorithm geometry.

- **Take-off:** For the ProNav equations to work properly, the fleet velocity must not be zero. To deal with this minor issue, we command the fleet to increase its altitude to a small value and to obtain a small forward velocity before initiating the ProNav algorithm.
- **Independent Rotation:** The ProNav guidance algorithm is used to command the desired position for the fleet's center, but it does not inherently command the desired rotation of the fleet. Therefore, the fleet rotation angles are controlled independently by matching the net's normal vector, $\mathbf{n}$, with the line of light vector $\boldsymbol{\ell}$. Letting $\mathbf{n} = \frac{\boldsymbol{\ell}}{\|\boldsymbol{\ell}\|}$, then the desired angles of the net are

$$\begin{aligned}\phi_d &= \text{atan2}(-n_y, n_z)\\ \theta_d &= \text{atan2}(n_x, n_z)\\ \psi_d &= \text{atan2}(n_y, n_x),\end{aligned}$$

where ϕ_d, θ_d, and ψ_d are the desired roll, pitch, and yaw angles, respectively. The actual commanded roll, pitch, and yaw angles of the fleet are given by

$$\begin{pmatrix}\phi_c\\ \theta_c\\ \psi_c\end{pmatrix} = r(L)\begin{pmatrix}\phi_d\\ \theta_d\\ \psi_d\end{pmatrix},$$

where the ramp function $r(L)$ is

$$r(L) = \begin{cases} 0, & \text{if } L < d_s\\ \frac{d_s - L}{d_s - d_f}, & \text{if } d_s < L < d_f\\ 1, & \text{if } L > d_f\end{cases},$$

and where d_s is the distance at which the formation begins rotating and d_f is the distance at which the formation is fully rotated to the desired angles ϕ_d, θ_d, and ψ_d.

Ramping the rotation in this way allows the net to remain flat for most of the intercept flight thereby allowing for less drag while the fleet is moving into position to intercept the intruder. This same ramped rotation approach is used for each of the algorithms discussed in this paper.

- **Flyby:** Because ProNav is mainly used as a missile guidance system, there is usually little concern about what to do after the target is reached. In our case, we don't want to stop, but want to continue past the target or ”flyby” in order to ensure effective capture of the intruder.
  We define a flyby unit vector, $\mathbf{v}_{\text{fb}}$, that points in the direction we want the fleet to be traveling and continue to travel when it intercepts the target. Since the ProNav equations are dependent upon the line of sight vector $\boldsymbol{\ell}$, we manually augment $\boldsymbol{\ell}$ with a flyby offset vector
  $$\mathbf{v}_{\text{offset}} = k\mathbf{v}_{\text{fb}}$$
  where
  $$k = \begin{cases} -d_{\text{fb}}, & \text{if } d_{\text{fb}} - L < -d_{\text{fb}} \\ d_{\text{fb}} - L, & \text{if } |d_{\text{fb}} - L| < d_{\text{fb}} \\ d_{\text{fb}}, & \text{otherwise} \end{cases},$$
  and d_{fb} is a predetermined flyby distance. Thus, the augmented line of sight vector becomes $\boldsymbol{\ell}_{\text{fb}} = \boldsymbol{\ell} + \mathbf{v}_{\text{fb}}$.

Pros and Cons: The basic ProNav implementation given here is an optimal guidance law that is efficient for missile guidance, but its optimal performance relies upon some limiting assumptions such as having a non-maneuvering target and little autopilot lag [7]. ProNav's performance degrades when the velocity of the pursuer is less than that of the target, the target performs large acceleration maneuvers close to intercept, or the pursuer has lag in responding to the target's motion. Accordingly, the ProNav algorithm performs reasonable well when the intruder takes a straight path, but it is not very robust at dealing with rapid changes in the intruder's velocity.

B. Target-predictive Smoothed Waypoint Path

The target-predictive algorithm attempts to predict the future position of the intruder and creates a third degree polynomial path to the optimal intercept point.

The digital implementation of this algorithm utilizes a binary search to calculate the intercept point of the fleet and the target. The algorithm assumes a constant velocity model for both the intruder and the fleet, using the current velocity of the intruder and a constant nominal velocity for the fleet. It then plans a smoothed path to the computed intercept point.

The algorithm below is used to calculate the intercept position. This algorithm was adapted from a ball intercept algorithm used for robot soccer [8]:

The algorithm uses the following functions

p_t^+ = **targetPredict**(t^+): Returns the predicted target position at time t^+ into the future

P = **planPath**(p_1, p_2, h): Plans a waypoint path from point p_1 to p_2, such that the direction of the final leg of the path is identical to h.

L = **pathLength**(P): Returns the path length of P.

The path length to the predicted target location is given by

$$L = \textbf{pathLength}(\textbf{planPath}(z, \textbf{targetPredict}(t^+), h))$$

where z is the current location of the UAS fleet. The time it takes the UAS to traverse this path with velocity v is given by

$$T(t^+) = \frac{\textbf{pathLength}(\textbf{planPath}(z, \textbf{targetPredict}(t^+), h))}{v}.$$

Define the function

$$g(T) = vT - \textbf{pathLength}(\textbf{planPath}(z, \textbf{targetPredict}(T), h)).$$

The ideal intercept time is the smallest T^* that satisfies the equation $g(T^*) = 0$. The ideal intercept time can be found using a binary search algorithm. Note that $g(0) < 0$, and that for some $T > 0$, we are guaranteed that $g(T) > 0$. Therefore, a bisection search algorithm will quickly converge to T^*.

Given that T^* is the time to intercept, we can then plan a path to the intercept point using $P = \textbf{planPath}(z, \textbf{targetPredict}(T^*), h)$.

The path is calculated by finding a third degree spline curve that matches the given initial and final positions and velocities. We used the current state of the fleet for the initial position and velocity and use the predicted intercept point and a ”flyby” vector for the final position and velocity.

A smooth path can be generated by using a parameterized path based on τ ranging from $\tau = 0$ at the initial position and $\tau = 1$ at the final position. Such a path can be defined as

$$\begin{pmatrix} \sigma_x(\tau) \\ \sigma_y(\tau) \end{pmatrix} = \begin{pmatrix} m_{11} + m_{12}\tau + m_{13}\tau^2 + m_{14}\tau^3 \\ m_{21} + m_{22}\tau + m_{23}\tau^2 + m_{24}\tau^3 \end{pmatrix}.$$

Defining the matrix

$$M = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \end{pmatrix}$$

and the vector

$$\Phi(\tau) = \begin{pmatrix} 1 \\ \tau \\ \tau^2 \\ \tau^3 \end{pmatrix},$$

then the τ-parametrized path is given by

$$\boldsymbol{\sigma}(\tau) = M\Phi(\tau).$$

Given the initial and final positions and velocities, $\mathbf{p}_0, \mathbf{v}_0, \mathbf{p}_f, \mathbf{v}_f$, then

$$\begin{aligned} \boldsymbol{\sigma}(0) &= \mathbf{p}_0 = M\Phi(0) \\ \boldsymbol{\sigma}'(0) &= \mathbf{p}_0 = M\Phi'(0) \\ \boldsymbol{\sigma}(1) &= \mathbf{p}_0 = M\Phi(0) \\ \boldsymbol{\sigma}'(1) &= \mathbf{p}_0 = M\Phi'(0), \end{aligned}$$

where

$$\Phi'(\tau) = \frac{d}{d\tau}\Phi(\tau) = \begin{pmatrix} 0 \\ 1 \\ 2\tau \\ 3\tau^2 \end{pmatrix}.$$

We can stack these quantities as a matrix to get

$$\begin{bmatrix} \mathbf{p}_0 & \mathbf{v}_0 & \mathbf{p}_f & \mathbf{v}_f \end{bmatrix} = M \begin{bmatrix} \Phi(0) & \Phi'(0) & \Phi(1) & \Phi'(1) \end{bmatrix}.$$

Therefore, the coefficient matrix M is given by

$$M = \begin{bmatrix} \mathbf{p}_0 & \mathbf{v}_0 & \mathbf{p}_f & \mathbf{v}_f \end{bmatrix} \begin{bmatrix} \Phi(0) & \Phi'(0) & \Phi(1) & \Phi'(1) \end{bmatrix}^{-1}.$$

In our implementation, we use the spline curve equations to calculate a set of waypoints and use a simple waypoint manager to follow the smoothed waypoint path.

Pros and Cons: This approach is efficient for intercepting predictable targets, but it is ineffective against targets with more unpredictable or aggressive paths. This algorithm performs well when the intruder path is relatively straight and simple, but it is much less effective against an intruder using more agile maneuvers.

C. Adaptive Radius Optimal Defense

The Adaptive Radius Optimal Defense (AROD) algorithm is designed to optimize interception in a defensive situation where an offensive scheme is not necessary. It also takes into account an inherent control delay between the fleet and the intruding UAV.

In this strategy, the fleet deploys to a maximum radius and tracks the intruder position on that radius. The radius on which the fleet resides is dynamically increased based on the chance the fleet has to capture the intruder and is decreased based on how likely it is to miss the intruder. By adapting the fleet radius in this way, we can effectively increase the maximum allowable reaction time in order to respond to a maneuver performed by the intruder.

The geometry for the adaptive radius optimal defense algorithm is shown in Figure 3. The algorithm is explained in more detail below.

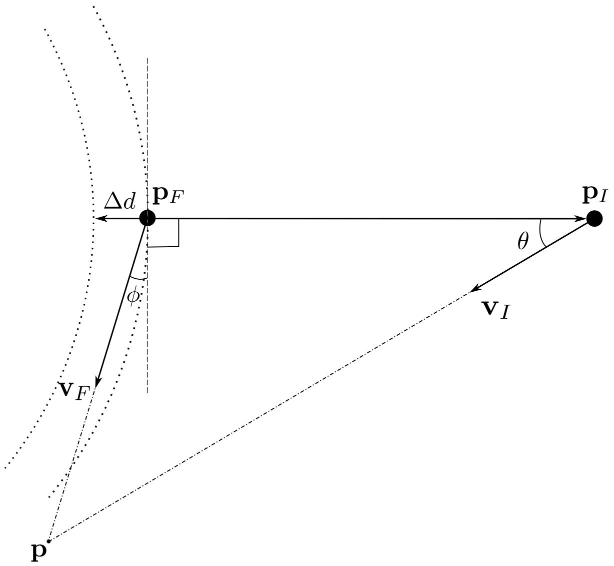

Fig. 3. The geometry associated with the Adaptive Radius Optimal Defense algorithm.

Let d be the distance between the fleet and the intruder and let θ be the angle between the line-of-sight vector and the intruder's velocity $\mathbf{v}_I$. Also, as shown in Figure 3, let ϕ be the deflection angle so that if the formation center moves at velocity $\mathbf{v}_F$, it will intercept the intruder at point $\mathbf{p}$.

Let d_F and d_I be the respective distances of the fleet and the intruder to the intercept point $\mathbf{p}$, and let t_F and t_I be the time it takes each to get to point $\mathbf{p}$. Then

$$t_F = \frac{d_F}{\|\mathbf{v}_F\|} + t_r$$
$$t_I = \frac{d_I}{\|\mathbf{v}_I\|},$$

where t_r is the delay due to reaction time of the fleet. By the law of sines,

$$d_F = d\frac{\sin(\theta)}{\sin(90 - \theta - \phi)}$$
$$d_I = d\frac{\cos(\phi)}{\cos(\theta + \phi)}.$$

Therefore,

$$t_F = \frac{d\sin(\theta)}{v_F\cos(\theta + \phi)} + t_r$$
$$t_I = \frac{d\cos(\phi)}{v_I\cos(\theta + \phi)}.$$

The objective is to solve for the deflection angle ϕ when $t_F = t_I$. To do so we use the bisection search listed in Algorithm 1.

Algorithm 1 Bisection search for ϕ

```
Set $\phi_1 = -\frac{\pi}{2}$ and $\phi_2 = \frac{\pi}{2}$
  do
      $\phi = \frac{\phi_1+\phi_2}{2}$
      $t_F = \frac{d\sin(\theta)}{v_F\cos(\theta+\phi)} + t_r$
      $t_I = \frac{d\cos(\phi)}{v_I\cos(\theta+\phi)}$
      $\Delta t = t_I - t_F$
      if $\Delta t < 0$ then
          $\phi_1 = \phi$
      else
          $\phi_2 = \phi$
      end if
  while $\|\Delta t\| > \epsilon$
```

In addition, the deflection angle ϕ is used to adjust the radius at which the fleet resides by

$$\Delta d = v_{max} * dt * sin(\phi)$$

where dt is the sample rate of the intercept algorithm.

We also allow for a tunable minimum radius at which the fleet's commanded radius will saturate. This is used to prevent the fleet from backing too far from an aggressive target, allowing the target to get closer to the center of defense than desired.

Pros and Cons: The main advantage to this approach is that the fleet can focus its energy and efforts on defending a central location and minimize the chance of missing the target. As a direct result, one disadvantage of this approach is that it is not offensively aggressive implying that the intercept usually occurs closer to the center.

IV. SIMULATION RESULTS

The simulations for this paper were conducted using Gazebo 7 and ROS Kinetic. Unless otherwise stated, the assumed net size for each simulation is 6×6 meters, the initial position of the fleet is the origin (0, 0, 0), which is also the center of defense, and the initial position of the intruder is $(130, 0, 35)$ meters.

We evaluate the performance of each intercept strategy by comparing the accuracy, intercept time, and intercept distance of each approach while varying different simulation parameters. We ran one thousand iteration of each test case to obtain a reasonable sample size so we can observe the average behavior for each case.

- *Accuracy:* Measured in meters as the relative position of the intruder to the center of the fleet net when the intruder passes the fleet. If the intercept point is within the bounds of the net, it is marked as a success. In the simulation results, the intercept points are plotted with $(0, 0)$ being the center of the net.
- *Intercept Time:* Measured in seconds from the beginning of the simulation when the fleet is deployed, to the time when the fleet passes the intruder, whether it was a successful attempt or not.
- *Intercept Distance:* Measured in meters as the total distance from the center of defense to the position of the target when the target and fleet intercept.

A. Straight Path to Center

The simulation results in this section are for the case where the intruder follows a straight waypoint path to the center of defense. The waypoint path follows a steady decline towards the center with no lateral maneuvers. An example straight waypoint path is shown in figure 4.

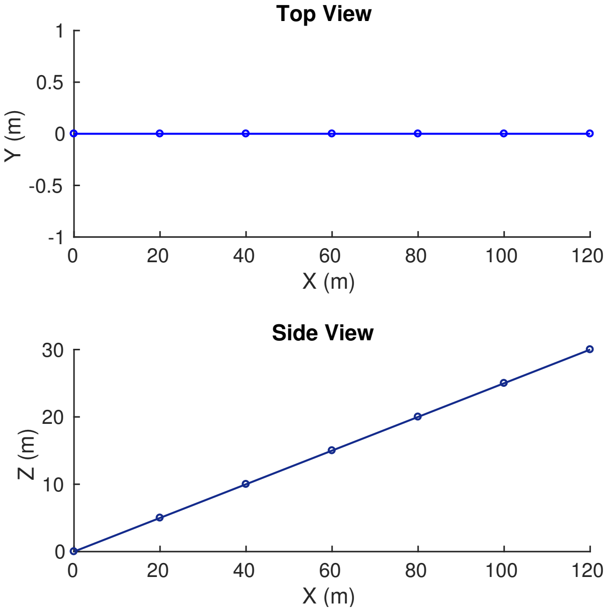

Fig. 4. Top and Side View of Intruder Straight Path to Center

The results for the three tests are summarized in Table I. Corresponding intercept point distribution plots are found in figures 5-7.

TABLE I
STRAIGHT PATH TO CENTER RESULTS

	ProNav	**Predictive Path**	**Adaptive Radius**
# of trials	1000	1000	1000
Successes	818	991	1000
Success rate	81.8%	99.1%	100.0%
Avg. Point	(-0.88, -1.77)	(1.47, 0.21)	(0.80, -0.30)
Avg. Time	12.40	14.87	27.70
Avg. Distance	98.80	83.73	60.46

For a simple straight path scenario, all three of the algorithms perform quite well.

a) ProNav: The ProNav approach was least effective, with a success rate of only 82%. In the simulations, ProNav responded least well to changes in the intruder's velocity. While there was little lateral movement in this scenario, the ProNav intercept algorithm could not respond quickly enough to those changes in the lateral velocity and therefore did not have the accuracy of the other two algorithms.

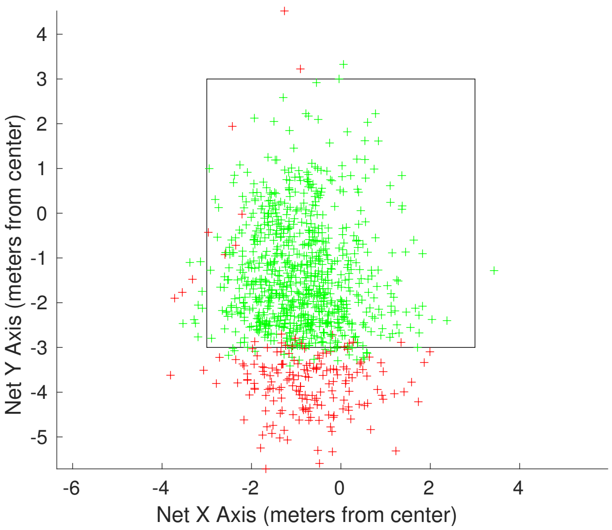

Fig. 5. ProNav Interception Points with Straight Path to Center

b) Target-Predictive Smoothed Waypoint Path: The Predictive Path algorithm performed very well against straight intruder paths. The success rate was near perfect and the average intercept time (14.87 s) was almost half that of the Adaptive Radius approach (27.70 s). The average intercept distance was also about 23 meters further from the center of defense than the Adaptive Radius average distance, which could be a valuable attribute depending on the total radius of the area to be defended.

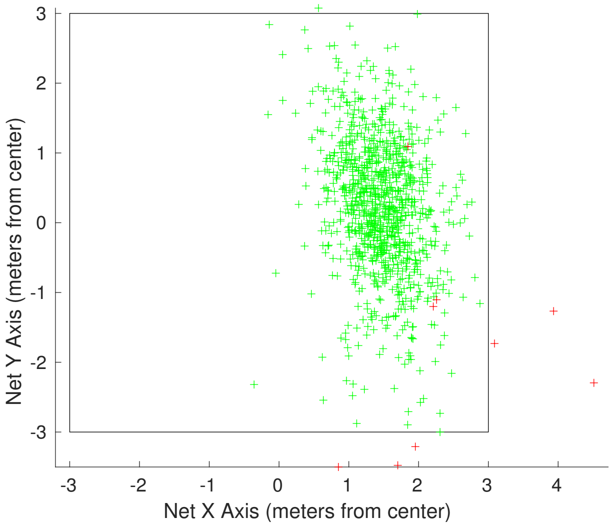

Fig. 6. Target-Predictive Smoothed Waypoint Path Interception Points with Straight Path to Center

c) Adaptive Radius Optimal Defense: The Adaptive Radius approach had a perfect success rate for the 1000 simulations we ran. The intercept points were also more consolidated and more centralized than the other algorithms, suggesting a more robust and accurate approach. The main deficiency of the AROD approach for the straight path scenario is that it lacks the ability to perform a more aggressive or proactive intercept. Rather, it holds back, maintaining a more defensive position to guarantee intercept, but causing the average intercept time to be larger and the average intercept distance to be the smallest of the three. The main trade-offs for this algorithm are time and distance for accuracy and reliability.

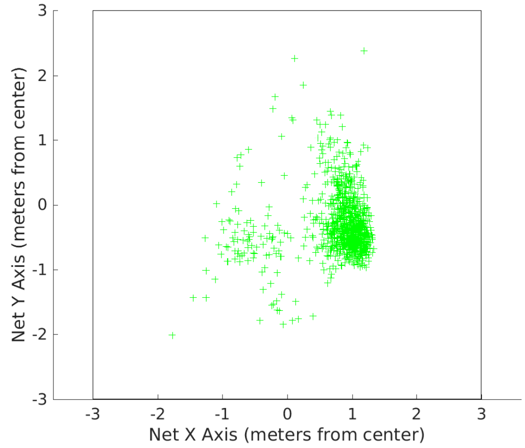

Fig. 7. AROD Interception Points with Straight Path to Center

B. Random Path to Center

For the simulations in this section, the intruder follows a random waypoint path descending to the center of defense. The random path is generated by choosing points on six evenly spaced concentric circles around the center of defense, each at a random angle along that circle. In order to maintain a generally forward-moving path towards the center of defense, we allow a max $\Delta\theta$ of ± 15 degrees between each successive waypoint. This path acts as a stress test on the accuracy and responsiveness of the algorithms by allowing the intruder to maneuver side to side as it moves toward the center of defense. An example random path is shown in figure 8.

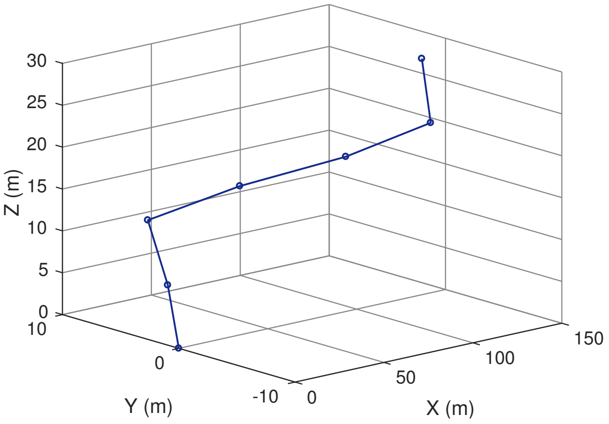

Fig. 8. Sample intruder random path to center.

The results for the three tests are summarized in Table II. Corresponding intercept point distribution plots are found in figures 9-11.

a) ProNav: The ProNav algorithm performs poorly for a random path. Introducing lateral maneuvers between waypoints causes the ProNav approach to fail about 80% of the time. Proportional Navigation operates best when the interceptor has a velocity greater than the target and when the target maintains a relatively constant velocity. This data shows that ProNav performance is significantly reduced when the velocities are equal and when the target can accelerate rapidly. While it is effective as a missile guidance algorithm ProNav does not perform well as an intercept algorithm for systems in which the intercepting and intruding UAS have similar speeds.

TABLE II

RANDOM PATH TO CENTER RESULTS

	Pronav	Predictive Path	Adaptive Radius
# of trials	1000	1000	1000
Successes	201	420	872
Success rate	20.1%	42.0%	87.2%
Avg. Point	(-0.09, -3.53)	(0.72, 2.46)	(0.09, 0.41)
Avg. Time	11.98	16.04	28.30
Avg. Distance	96.56	82.33	60.13

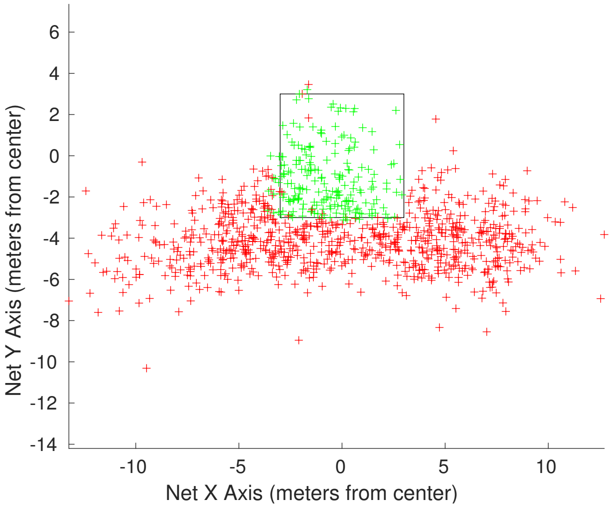

Fig. 9. ProNav Interception Points with Random Path to Center

b) Target-Predictive Smoothed Waypoint Path: The target-predictive approach had a mediocre success rate of 42%, which was more than double that of ProNav, but less than half that of AROD. While it was possibly the best algorithm in overall performance for the straight path scenario, the performance decreased dramatically against a more aggressive intruder. The algorithm was not robust enough to effectively respond to the acceleration of the added maneuvers.

c) Adaptive Radius Optimal Defense: The AROD approach produced a significantly higher success rate against an intruder with a random path than the other two algorithms. Similar to the straight path scenario, this algorithm has the longest time to intercept and the shortest intercept distance from the center of defense, but the performance was accurate, with intercept points centered on the net and almost no attempts resulting in an intercept point more than 2 meters beyond the net. It is possible that with a larger net, this algorithm would have resulted in a near perfect success rate.

In both the straight path and random path simulations, the average distance was very close to 60 meters. This is a result of the minimum radius constraint placed on the fleet. We chose 60 meters as our minimum radius by testing various values and comparing performance and values near 60 meters seemed to perform slightly better. While we kept a static minimum radius for all simulations, a possible future research topic could be to investigate the use of a dynamically changing or more sophisticated selection of a minimum radius.

An example of how the current algorithm adapts to the state of the target is shown in Figure 12.

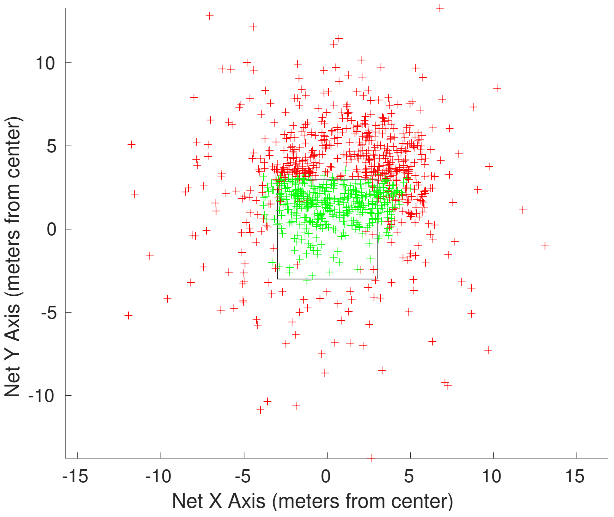

Fig. 10. Target-Predictive Smoothed Waypoint Path Interception Points with Random Path to Center

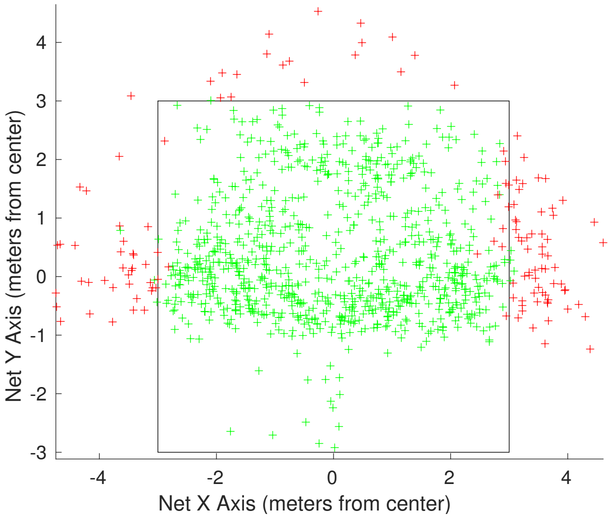

Fig. 11. AROD Interception Points with Random Path to Center

V. CONCLUSION

This paper has introduced three possible intercept algorithms for a counter UAS scenario. In comparing the simulation results for these three algorithms, the ProNav approach was the least effective for this particular scenario. While it had the proactive aggression that AROD lacked, it was much less effective as a defensive approach for UAS. It proved to be a poor choice for UAS intercept schemes where the interceptor and the target have similar velocities.

The Target-Predictive Waypoint Path method had good performance for simple, predictable situations, but its performance degraded quickly as the complexity of the intruder path increased. The algorithm, as implemented here, would serve as a mediocre intercept method. It is possible, however, that a more sophisticated prediction scheme would produce enhanced results.

The Adaptive Radius Optimal Defense approach had the best performance. It was much more consistent at successfully intercepting the intruder in the dragnet. Its main drawback however was its lack of aggression in more simple situations. While it outperformed the other algorithms in accuracy, it was slower to intercept and allowed the intruder to come closer to the center of defense. Despite this trade-off, Adaptive Radius Optimal Defense provides a superior intercept success rate to some existing approaches such as ProNav and target-predictive intercept. We propose that this algorithm merits further research and consideration as a viable option for counter UAS schemes using a formation controlled dragnet.

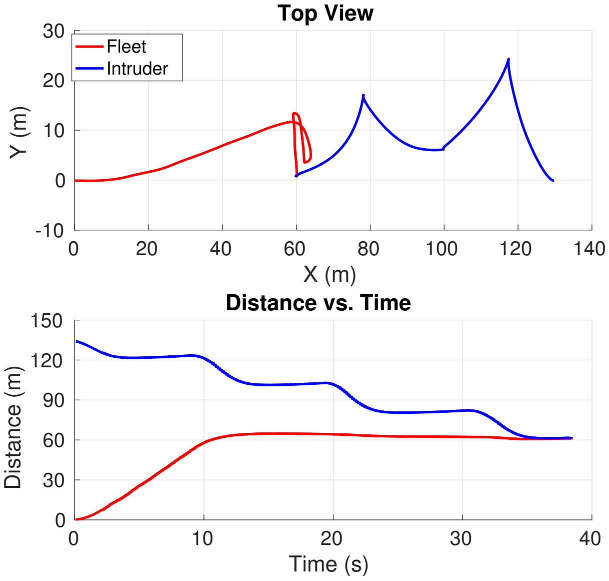

Fig. 12. Example simulation showing how AROD adapts to intruder movements.

A possible topic for further search could be to investigate a fusion of these methods. A dynamic approach, which could be more aggressive when the intruder path is predictable and a more defensive posture when the risk of failure is high, may provide superior results than AROD alone.

ACKNOWLEDGMENT

This research was conducted in the Center for Unmanned Aircraft Systems (C-UAS) with support from the National Science Foundation I/UCRC program grant number IIP-1161036 and C-UAS Industry Advisory Board members.

REFERENCES

[1] M. Turan, F. Gunay, and A. Aslan, “An analytical approach to the concept of counter-ua operations (cuaops),” *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1, pp. 73–91, 2012, iD: Turan2012. [Online]. Available: http://dx.doi.org/10.1007/s10846-011-9580-6

[2] R. Wallace and J. Loffi, “Examining unmanned aerial system threats & defenses: A conceptual analysis,” *International Journal of Aviation, Aeronautics, and Aerospace*, 2015.

[3] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein, “Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft,” *Journal of Field Robotics*, vol. 30, no. 3, pp. 415–438, 2013. [Online]. Available: http://dx.doi.org/10.1002/rob.21454

[4] G. M. Siouris, *Missile Guidance and Control Systems*. New York: Springer-Verlag, 2004.

[5] P. Zarchan, *Tactical and Strategic Missile Guidance*, ser. Progress in Astronautics and Aeronautics. Washington DC: American Institute of Aeronautics and Astronautics, 1990, vol. 124.

[6] R. W. Beard, J. W. Curtis, M. Eilders, J. Evers, and J. R. Cloutier, “Vision Aided Proportional Navigation for Micro Air Vehicles,” in *Proceedings of the AIAA Guidance, Navigation and Control Conference*. Hilton Head, North Carolina: American Institute of Aeronautics and Astronautics, aug 2007.

[7] N. F. Palumbo, R. A. Bauwkamp, and J. Lloyd, "Modern homing missile guidance theory and techniques," *Johns Hopkins Apl Technical Digest*, vol. 29, no. 1, pp. 42–59, January 2010. [Online]. Available: https://www.researchgate.net/publication/265487706_Modern_Homing_Missile_Guidance_Theory_and_Techniques

[8] R. W. Beard, "Notes on robot soccer," February 27, 2017. [Online]. Available: http://rwbclasses.groups.et.byu.net/lib/exe/fetch.php?media=robot_soccer:material:robot_soccer_notes.pdf