Barricade: A Target Defending Algorithm on Robotarium

Weilian Fu   
Electrical and *Computer Engineering Department  
University of Virginia*Charlottesville, VA  
wf4fj@virginia.edu

Hongyu Li  
Electrical and *Computer Engineering Department  
University of Virginia*Charlottesville, VA  
hl5uq@virgini.edu

Arman Mottaghi  
*Mechanical and Aerospace Engineering Department  
University of Virginia*Charlottesville, VA  
am5yc@virginia.edu

Cody H. Fleming  
Assistant Professor *Systems and Information Engineering Department*  
*University of Virginia*Charlottesville, VA  
cf5eg@virgini.edus

***Abstract*—Commercially available drones were previously used against U.S. and Russian military personals in both Iraq and Syria. They have also caused numerous incidents and accidents nearby airports or other places with heavy air traffic. This paper proposes a novel algorithm to identify and track enemy drones in the presence of limited number of defender agents.**

***Keywords—Target-defending, Algorithm, Simulate, Optimize***

# Introduction

In today’s rapidly changing world, commercial drones are becoming more popular among youth. Although for the most part drones don’t pose a serious threat, they can jeopardize our national security or the safety of our citizens. There had been a helicopter crash in South Carolina in early February 2018. The National Transportation Safety Board confirmed that a civilian drone triggered the crash. In early 2018, a swarm of drones attacked the Russian Run-Hmeimim base in Syria after traveling roughly 50 km. They primarily relied on their GPS devices to navigate through rough Syrian terrain. Washington Post reported that in several occasions the U.S. Special Operations forces were attacked by the Islamic State drones in near Raqqa in Iraq. “Unlike in Mosul, where U.S. forces have deployed an array of drone-stopping systems, U.S. troops on the ground in Raqqa are operating with fewer resources and have a limited ability to defend against the small, hard-to-spot aircraft,” the officials said to the reporters. Such incidents can easily be prevented by a military-grade UAV capable of flying faster than any commercially-off-the-shelf (COTS) drone or handmade black-market RC airplanes. A robust, efficient and economical approach is needed to eliminate such threats. A system of (defender) UAVs capable of patrolling a specific area and eliminating the enemy drones would be far better than any anti-drone missile due to its relative cost and re-usability.

Although a robust defense system composed of several defender drones could potentially be a viable solution to ensuring the safety and security of our military personnel abroad as well as civilians across the world, such system requires careful planning and design. A reliable defense system shall be capable of protecting its assets in the worst-case scenarios, such as a simultaneous attack by a swarm of a few hundred drones. Thus, to have a robust, reliable, and efficient drone defense system, a careful analysis of its control architecture is not only inevitable but required.

The purpose of this paper is to evaluate and analyze the level of the threat posed by a swarm of enemy drones under different defensive formation of defender agents. In this work, we focus on the problem of finding an optimal enemy assignment algorithm given an initial defender formation policy.

# Relevant Work

In the recent years, the study of drones is getting more and more popular. Most of the study is focusing on the control of agents, trying to make the drones behave more accurately. Of course, as the drone is easily getting by anyone, there is more concern about security. So, some of the authors will try to find a way to defend drone trespassing. However, still, not many of them are dealing with the target choosing.

## Control Laws

Many works are focusing on this area. Works in [1] show that with Kalman filter, a centralized control point can control the UAVs well. Using the dynamic programming, the controller could optimize the coordination of UAV [2]. In [3] and [4], the authors developed ways about controlling multiple agents in the same time. [5] Shows that it is possible to use multiple GPS receivers for tracking drone orientation. With all the paper, we could assume that in large scale, the centralized control of defender agents will be accurate with a global orientation.

B. Radio Jamming

Fewer works are done here because most of the works in this area are about how to build robust UAVs against radio jamming. However, there is still some useful approach to radio jamming small agents. [6] showed a way about building an anti-drone system which consists of a 3D Frequency Modulated Continuous Wave radar and a directional jammer working at 2.4 GHz. [7] describes a spoofing system designed to control drones and outlines the system requirements and features to take control of a common radio-controlled drone. However, still less are done here about the target choosing.

# Approach

In this approach first, we developed an urgency function (U) to provide a basis for ranking the enemy drones based on their threat level (Equation 1). The value of the urgency function for a given enemy-defender pair is an indication of whether the defender agent should pursue the enemy agent or not. The enemies with a higher value of urgency function are more likely going to be chased by a set of defender drones over those that have low urgency values. Thus, it becomes apparent that enemy selection or assignment is one of the important pillars of this study. Without a consistent and reliable assignment algorithm, evaluation and analysis of the threat posed by a swarm of enemy drones given a defense formation of defender drones become impractical. Therefore, we addressed the problem of assigning the enemy drones to a defender drone by developing an assignment algorithm. Such algorithm aims to find the most optimal selection sequence of enemy agents for a given set of defender agents. The most optimal selection sequence would yield the highest total urgency values for that selection. The problems described above are summarized in the following problem statements.

# Problem Statements

# *Ranking enemy agents on a common scale*

# Definition: Given a system of M defender agents (M > 1) and N enemy agents (N > 2) with a known set of characteristics, such as their absolute position, distance from the protected area, relative distance to enemy or defender agents, what is the best approach to rank the enemy drones based on their threat level?

# Solution: An urgency function (U) is defined as shown below:

Equation 1 - Urgency function

In Equation 1, variablerepresents the distance of enemy i from the defender j and denotes the distance of enemy i from the protected area. They are raised to the power of and respectively.

Once Equation 1 is evaluated for each pair of the enemy I and defender j, the results can be represented by a matrix size ji as shown in Table 1 below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Enemy 1 | Enemy 2 | Enemy 3 |
| Defender 1 | U(1,1) | U(1,2) | U(1,3) |
| Defender 2 | U(2,1) | U(2,2) | U(2,3) |
| Defender 3 | U(3,1) | U(3,2) | U(3,3) |

Table 1 - urgency matrix for a system of I enemy agents and j defender agents

# *Optimal selection sequence*

# Definition: Given a system of M defender agents (M > 2) and N enemy agents (N >2) with a known urgency function value like the one shown in Table 1, what is the most optimal way of assigning the enemy agents to the defender agents?

# Solution: The assignment process shall maximize the following expression:

# 

# Equation 2

The following algorithm is developed in MATLAB to ensure the above condition is always true:

1. Obtain the number of available defenders
2. Find some permutations for the given set of defender agents. For example, given three available defender drones, v = perms([1, 2 3]) would results in a 63 array.
3. For each available permutation, evaluate
4. Select the permutation with the highest value of and assign it to a centralized controller.

# Assumptions

The following assumptions are made during all phases of the project:

* Enemy and defender agent’s velocity ratio is 1.73.
* Enemy agents approach the DZ from random directions, but they all enter the zone at the same time.
* The optimal initial (defensive) formation structure for defender agents is such that the distance between any pair is constant while no enemy agent is entering the DZ airspace.
* If any enemy agent is within the radius of the protected zone, it counts as an attack.
* The initial assignment process begins when all enemy agents enter the DZ airspace.

# Platform

The Georgia Tech’s Roboterium [8] was modified and reconstructed to verify the validity the proposed solutions so that the level of the threat posed by a swarm of enemy drones can be evaluated and analyzed numerically. The following factors are identified as relevant given the set of assumptions:

* Initial distance of defender and enemy agents
* Defender agent’s controller’s x and y velocity gains
* Number of defender agents
* Number of enemy agents
* Detection zone radius
* Protected zone radius
* Time step
* Elimination time
* Urgency function and values
* Number of attacks by enemy agents

In this platform, the positions are measured from the origin (0,0). T Table 2 illustrates a summary of simulation variables with their respective values:

|  |  |  |
| --- | --- | --- |
| **Factors** | **Average Values** | |
| **Simulation** | **Reality** |
| Defender Velocity Limit | 0.046 / iteration | 100 km/h |
| Enemy Velocity Limit | 0.8 / iteration | 60 km/h |
| Detection Zone Radius | 1 - 4 | 1 - 4 km |
| Protected Area Radius | 1 | 1 km |
| Time Step | 0.6 | See Section III.D |
| Elimination Time | Ten iterations | See Section III.D |
| Number of Enemies | 16 |  |
| Number of Defenders | 4 |  |
| Defender X-Velocity Gain | 2 | See Section III.D |
| Defender Y-Velocity Gain | 2 | See Section III.D |
|  | 0-4 |  |
|  | 1-5 |  |

Table 2 - Simulation variables

# Limitations and Future Work

The single integrator controller was used in the Roboterium environment. The use of a single integrator controller can be a limiting factor considering its simplistic structure as well as the relationship between the simulation variables (such as as-as time step, velocity limits, capturing time). For the most part, the simulations were executed using a large time step (~0.6) which affects the quality of the analysis. The Roboterium uses a two-step process to determine the next position of an agent. First, the velocity values (are generated by the controller for each agent. In the controller, the values are normalized according to the specified velocity limits (0.048 and 0.08).

Although the velocity ratio of defender and enemy agents match with what we assumed previously, the velocity magnitudes are not a mirror of the reality. It becomes an issue for the defender during the randevusuz process as shown in Figure 2.

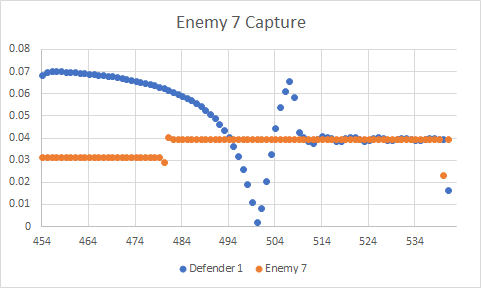


Figure 1

In the current model, the defender agent shall be less than 0.22 simulation unit away from the enemy agent for the capturing countdown process to begin. Thus, if the assumed capture time is ten iteration, it takes about 60 iterations for the defender agent to capture the enemy agent. During this time, the agents have traveled roughly 2.4 simulation distance unit which is unrealistic.

Therefore, the time step, agent’s velocity limits, defender agent’s velocity gains as well as the dimension of the simulation shall be carefully tuned to values such that it would yield in more realistic results. Additionally, more sophisticated controllers such as double-integrator or nonholonomic integrator shall be used to test and verify the validity of the findings of this paper.

# Results

1. Round 1: Finding optimal and values

In the first round, the four defender agents were placed at various distances away from the center of the protected zone[[1]](#footnote-1). For each distance, six combinations of and values were considered and tested 40 times. The 1200 simulations were conducted, and the results were analyzed using Minitab 12. The results are shown below:

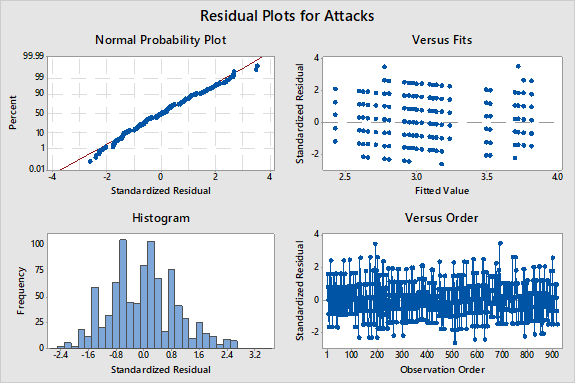


Figure 2 - Residual plots for round 1 simulations

The above plots indicate the normality of the values (number of the attacks by the enemy agents) obtained from the first round of simulations. Since the values are normal, the results can be further analyzed using the factorial design analysis toolkit.

The factorial design analysis indicates that the value of lambda is more significant than gamma when at evaluating the total number of attacks by the enemy agents (Figure 3).

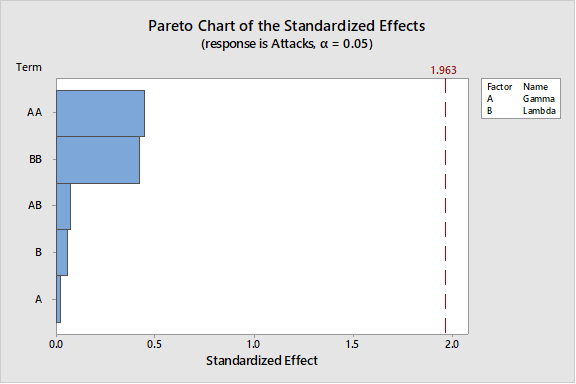


Figure 3

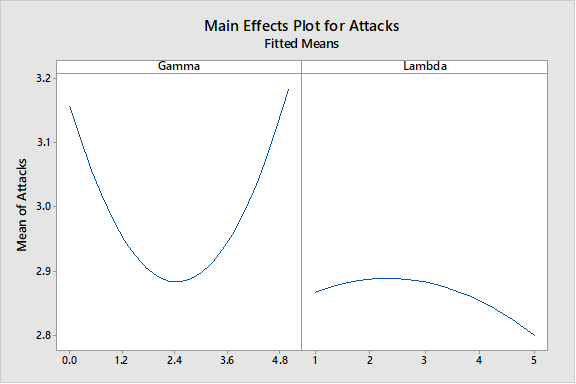


Figure 4

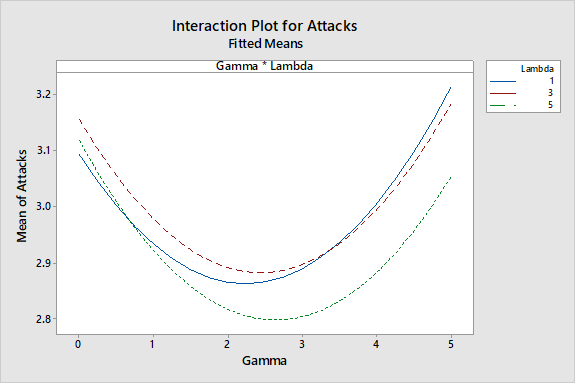


Figure 5

However, the value of gamma is not insignificant as illustrated in Figures 4. As shown in Figure 5, and values are associated with the lowest number of attacks. In this case, the defender agent is more likely to pursue the closest enemy agent rather than chasing the one that is farther away, but closer to the Protection Zone. The contour plot shown in Figure 6 confirms that when enemy agents enter the DZ airspace simultaneously, their relative distance from the individual defender agents become more important during the process in which the enemy agents are assigned to the defender agents.

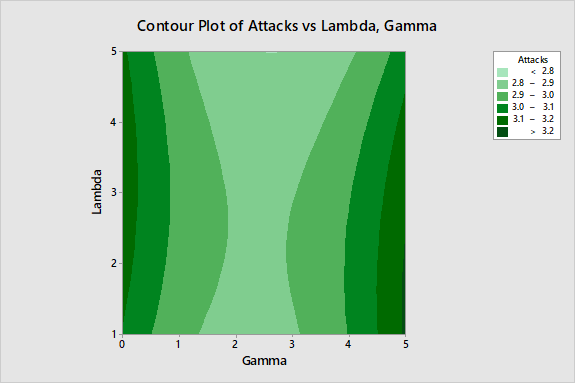


Figure 6

The above simulations were conducted using the values specified in Table 2.

B. Round 2: Relationship between the Radius of Detection Zone and the Initial Position of Defender Agents

Using the values of and obtained in the previous round, a second set of simulations (160 simulations) were conducted to explore the relationship between the initial radial position of defender agents and the Detection Zone radius. The preliminary analysis of the results indicate that the obtained values are normal (Figure 7).

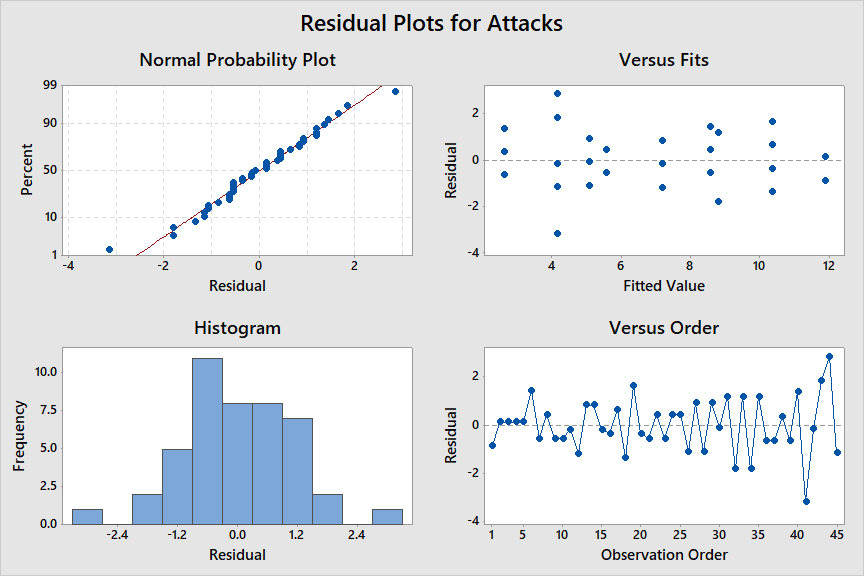


Figure 7 - Residual plots for some attacks in the second round of simulations

Likewise, the second round’s result was analyzed using the full-factorial analysis in Minitab. The results are shown in Figures 8 and 9.

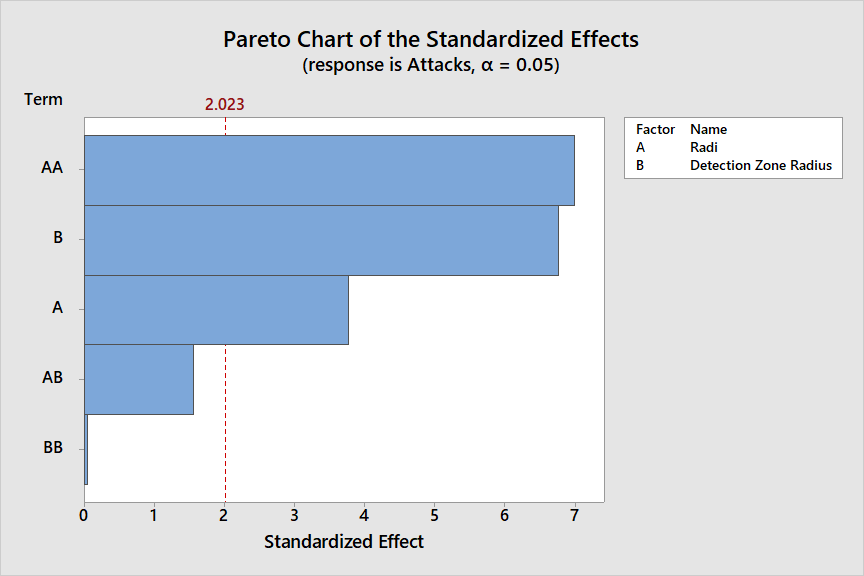


Figure 7 - The second round Pareto chart of effects

The results indicate that the Detection Zone radius has a more significant impact on lowering the total number of attacks than the initial position of the defender agents. Although the interaction factor, AB, shown in Figure 8 might not be significant, it is still impactful. The contour plot shown in Figure 9 suggests that increasing the DZ radius by itself is not sufficient to minimize the total number of attacks made by enemy agents. The initial defensive formation of defender agents is also important to ensure the safety of the Protected Zone. The results indicate that placing the defender agents at half of the DZ radius away from the center point result in the lowest number of attacks.

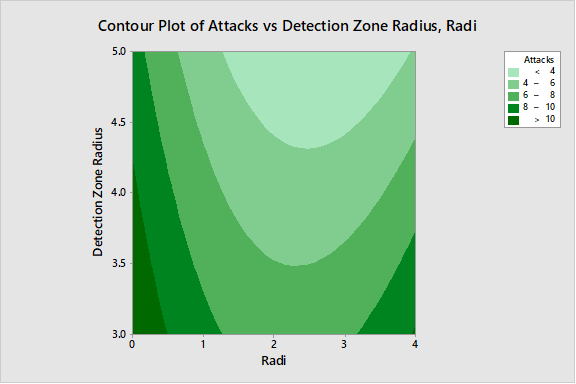


Figure 9

# Conclusion

In this paper, we first focused on the problem of finding an optimal assignment policy algorithm for a multi enemy-defender agent system. The urgency function (U) was developed to provide a reference value in which all enemy agents can be ranked too. The value is unique for every single enemy-defender agent pair. For a given defender j, the enemy with a higher value of urgency function is more likely going to be chased. The value of urgency function for a given pair of enemy-defender agent depends on two factors: the absolute distance of the enemy agent from the Protected Zone and its relative distance from the defender agent. Those factors are raised to the power of and respectively. The Roboterium environment was used to find the optimal and values for a given system with 4 defender and 16 enemy agents. The results suggest that when enemy agents enter the Detection Zone simultaneously, a higher lambda value associated with the relative distance of enemy agents from a defender agent become more important during the enemy assignment process. The results also indicate that increasing the DZ radius by itself is not sufficient to minimize the total number of attacks made by enemy agents. The initial defensive formation of defender agents is also important to ensure the safety of the Protected Zone. Thus, a lambda to gamma ratio of 2 along with placing the defender agents halfway to Detection Zone border will make the Protection Zone safer by minimizing the total number of enemy attacks.

##### References

[1] Zhan, P., Casbeer, D. W., & Swindlehurst, A. L. (2005, October). A centralized control algorithm for target tracking with UAVs. In Signals, Systems and Computers, 2005. Conference Record of the Thirty-Ninth Asilomar Conference on (pp. 1148-1152). IEEE.

[2] Quintero, S. A., Papi, F., Klein, D. J., Chisci, L., & Hespanha, J. P. (2010, December). Optimal UAV coordination for target tracking using dynamic programming. In Decision and Control (CDC), 2010 49th IEEE Conference on (pp. 4541-4546). IEEE.

[3] Li, X. R., & Jilkov, V. P. (2005). Survey of maneuvering target tracking. Part V. Multiple-model methods. IEEE Transactions on Aerospace and Electronic Systems, 41(4), 1255-1321.

[4] Blackman, S. S. (2004). Multiple hypotheses are tracking for multiple target tracking. IEEE Aerospace and Electronic Systems Magazine, 19(1), 5-18.

[5] Mahanth Gowda, Justin Manweiler, Ashutosh Dhekne, Romit Roy Choudhury, and Justin D. Weisz. 2016. Tracking Drone Orientation with Multiple GPS Receivers. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom ’16).

[6]T. Multerer *et al*., "Low-cost jamming system against small drones using a 3D MIMO radar-based tracking," *2017 European Radar Conference (EURAD)*, Nuremberg, 2017, pp. 299-302.

[7]M. Donatti, F. Frazatto, L. Manera, T. Teramoto and E. Neher, "Radiofrequency spoofing system to take over law-breaking drones," 2016 IEEE MTT-S Latin America Microwave Conference (LAMC), Puerto Vallarta, 2016, pp. 1-3.

[8]Pickem, D.; Glotfelter, P.; Wang, L.; Mote, M.; Ames, A.; Feron, E. & Egerstedt, M., ["The Robotarium: A remotely accessible swarm robotics research testbed"](https://arxiv.org/abs/1609.04730) 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore

1. https://youtu.be/lp50UeJfRJ4 [↑](#footnote-ref-1)