Geocaching-inspired Resilient Path Planning for Drone Swarms

Michel Barbeau
School of Computer Science
Carleton University
Ottawa, Ontario, Canada
0000-0003-3531-4926

Joaquin Garcia-Alfaro
Institut Polytechnique de Paris
CNRS UMR 5157 SAMOVAR
Telecom SudParis, France
0000-0002-7453-4393

Evangelos Kranakis
School of Computer Science
Carleton University
Ottawa, Ontario, Canada
0000-0002-8959-4428

***Abstract*—A path planning algorithm for drone swarms is presented. From the outset, none of the drones knows the path and final destination. Together, they collectively determine and unravel step-by-step the waypoints and final destination, resolving a localization problem at each step. It is a shared-information path planning algorithm. The algorithm is fault-tolerant and resilient to drones falling victim of attacks to their positioning system. It is shown that correctly functioning drones navigate the path provided that the number of faulty drones is less than $\frac{n-d}{2}$, where n is the total number of drones and d, equal to two or three, is the dimension of the space navigated by the drones. We validate the algorithm with appropriate simulations, implemented over OMNeT++ and GNSSim, which allow building network simulations including GPS attacks (e.g., jamming and spoofing attacks). The OMNeT++ models and GNSSim functions are linked together.**

***Index Terms*—Autonomous aerial vehicle, drone formation control, drone swarm, goal location, quadcopter, information sharing, localization, location, path planning.**

I. Introduction

We present a path planning algorithm for drone swarms. Together they collectively determine and uncover step-by-step the path and final destination, resolving a localization problem at each step. Ignorance about the path and final destination is a desired feature to collectively handle fault-tolerance and resilience to attacks.

Our work finds inspiration in geocaching, a well-known outdoor recreational activity, in which participants use a GPS receiver as well as a variety of scene cues (e.g., clues and references to landmarks) and other navigational techniques in order to collectively hide and seek objects while at the same time navigating a waypoint trajectory.

The algorithm presented allows for shared-information path planning through waypoints and is fault-tolerant and resilient to drone attacks. More specifically, non-faulty drones can correctly navigate the path provided that the number of faulty drones is less than $\frac{n-d}{2}$, where n is the total number of drones and d is the dimension of the space navigated ($d = 2, 3$).

Section II surveys related work. Section III presents our path planning algorithm. Fault tolerance and resilience to malicious attacks are discussed in Section IV. Section V validates our results with simulations. Section VI concludes the paper.

II. Related Work

Path planning for drones involves making them move from one point to another, while avoiding obstacles, following walls or moving consistently as a team, according to a pattern. Radmanesh et al. have published a survey on path planning with obstacle avoidance for drones [1]. Path planning with a coverage goal is discussed in a survey authored by Otto et al. [2]. Zhao et al. survey path planning involving computational intelligence [3]. General path planning approaches find inspiration in ideas developed for classical robotics, such as the ones using artificial potential function [4], random trees [5] or Voronoi diagrams [6]. Path planning may be considered together with team work and formation control [7]. Some approaches have been adapted to quadcopters [8], as well. Similarly to other path planning algorithms, the drones have a common goal, i.e., a location or coverage of an area. In contrast to the others, in our algorithm the drones do not know what the exact goal is, i.e., the final location, until the very end.

In the geocaching game analogy, a participant navigates to a location and tries to find a hidden container. It is given geographical coordinates and details about the location, clues and references to landmarks. In our algorithm, the geocaching game analogy is present when a drone participates to the resolution of a waypoint. For an individual drone, it means finding a new point, which is used with other points found by other drones. It is given a relative starting point and a translation, which, as in geocaching, may also be clues and references to landmarks. For that aspect, scene analysis, landmark search and navigation methods for drones can be used in combination with our algorithm [9]–[11].

Our research is also related to drone formation control. The authors of [12] assume that the signal propagation model of the drone has the shape of a sphere and analyze network capacity allocation in drone-based network infrastructure; they propose a drone formation algorithm that determines the 3D geographic location of each drone. In [13], the authors show how to operate a swarm by human piloting a drone (the leader) while the remaining followers are autonomous; they propose a solution in order to synchronize and orchestrate a swarm of drones, based only on ad hoc communications

to position drones. In [14], the authors formulate the multi-UAV formation reconfiguration problem as an optimal control problem with dynamical and algebraic constraints and provide a hybrid particle swarm optimization and genetic algorithm.

III. Shared-information Path Planning

A path is represented as a number of waypoints. At every waypoint, the drones collectively determine the next step by solving a localization problem. In the sequel, we discuss the localization problem, waypoint representation and path modelling.

A. Localization Problem

Let n be a positive integer. We define the *localization problem* as the collective solution by n drones of the location of an unknown point Q. In the two-dimensional Euclidean space, the drones find Q on the perimeter of a circle S, see Figure 1. Q is the point on the intersection of the perimeter

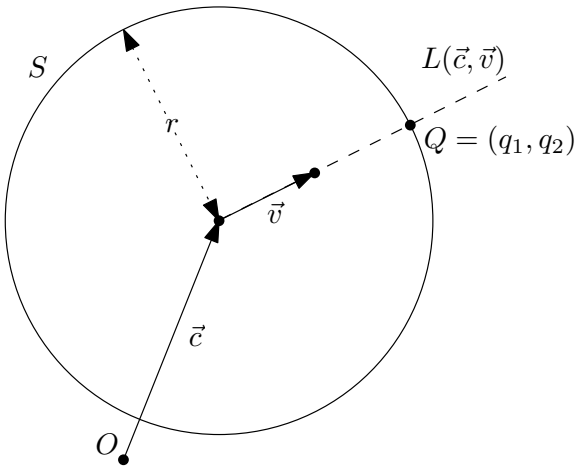


Fig. 1. In Euclidean space with origin O, the point Q is on the intersection of the line of action of vector $\vec{v}$, i.e., $L(\vec{c},\vec{v})$, and perimeter of the circle S.

of circle S and straight line denoted by $L(\vec{c},\vec{v})$. Vector $\vec{c}$ specifies the centre of the circle S. Unit vector $\vec{v}$ is a direction vector originating in the centre of the circle. The location of Q is where the supporting line of $\vec{v}$ intersects with S. It is represented by a pair of coordinates (q_1, q_2).

Initially, the circle S is unknown to the drones. Let i denote a drone index, with $i \in \{1, 2, \ldots, n\}$. Each drone i is given the direction vector $\vec{v}$ and information that determines a point P_i on the perimeter of the circle S. The information about the point P_i is owned only by drone i and not shared with other drones. For brevity, in the sequel, we abbreviate the localization problem as $\mathcal{L}(P_1, \ldots, P_n; \vec{v})$.

Recall from Euclidean geometry that a circle is uniquely determined by three pairwise different points on its perimeter. Therefore if $n \geq 3$, then there is a unique circle S determined by the n points. Upon determination of the points $P_1, \ldots, P_n$, the drones broadcast their positions to each other. They can all determine Q as the point at which the straight line $L(\vec{c},\vec{v})$ intersects with the perimeter of circle S. The localization problem is thus solved.

B. Representing Waypoints

Building upon the localization problem $\mathcal{L}(P_1, \ldots, P_n; \vec{v})$, we show how to represent any desired movement of a drone swarm. To this end, we show for any two different points Q, Q' how to map Q to Q'. Suppose that the drones have

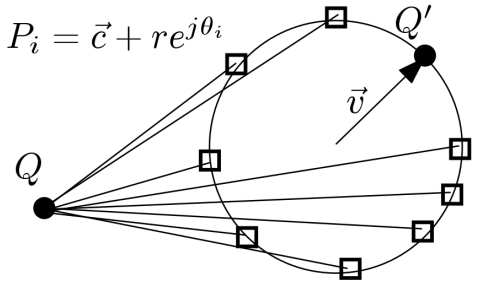


Fig. 2. Given points Q, Q' a unique circle can be determined. It is formed by the new positions of the drones (depicted as squares) in such a way that the point Q' lies on its perimeter.

found the location Q in a previous round, i.e., solution of $\mathcal{L}(P_1, \ldots, P_n; \vec{v})$. The drones resolve a new instance of the localization problem $\mathcal{L}(P_1, \ldots, P_n; \vec{v})$, depicted in Figure 2. The hollow squares represent the new positions that the drones have to find, i.e., $P_1, \ldots, P_n$. In reference to point Q, point P_i results from the transformation $\vec{c}+re^{j\theta_i}$, such that θ_i are n pairwise different angles in the interval $[0, 2\pi]$, $j = \sqrt{-1}$ and r is the radius of the circle. They locate a new point Q'. Each drone needs to know the transformation $\vec{c}+re^{j\theta_i}$ before going on a mission. Assuming drones have scene analysis capability, this transformation may take the form of visual cues, such as landmarks or buildings.

C. Representing Paths

Using waypoint representation, we determine a sequence of consecutive steps forming a path $Q_0, Q_1, \ldots, Q_{k-1} = Q$. The drones start from an initial configuration and inductively reach a specific destination as a group. The drones start from

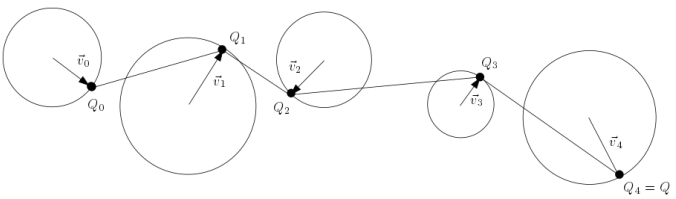


Fig. 3. A path consisting of four hops, as traversed by the drones. The drones start from point Q_0. In each instance, they use a direction vector $\vec{v}$ to compute an intermediate destination point Q_i on the perimeter of a circle. They determine their new positions and again compute the next intermediate destination using the next destination vector. This is repeated until the final destination point Q is reached.

point Q_0 determined by the solution of the initial instance of the localization problem $\mathcal{L}(P_{1,0}, \ldots, P_{n_0,0}; \vec{v}_0)$, see Figure 3.

Given that the solution of the i-th intermediate localization problem $\mathcal{L}(P_{1,i},\ldots,P_{n_i,i};\vec{v}_i)$ is the i-th intermediate point Q_i, they solve the $i+1$st instance of the localization problem $\mathcal{L}(P_{1,i+1},\ldots,P_{n_{i+1},i+1};\vec{v}_{i+1})$ and obtain the $i+1$st intermediate point Q_{i+1}. The final desired destination $Q=Q_{k-1}$ is reached when $i=k-1$.

Observe that in each iteration of the localization procedure some of the drones may malfunction, due to faults or malicious attacks of their positioning system. They may well withdraw from the localization problem. Hence, the number of drones participating in consecutive rounds may change, e.g., from the i-th to the $i+1$-st the number of participating mobiles may change from n_i to n_{i+1}, respectively. In case of an attack, the number of victims depends on the disruption power that the adversary has. Fault tolerance and resilience to attacks is discussed further in the upcoming section.

IV. Fault Tolerance and Resilience to Attacks

We analyze the tolerance to faults and resilience to malicious attacks of the shared-information path planning algorithm. Hardware failures may occur. Besides, drone positioning systems may be attacked, e.g., GPS spoofing attacks can be perpetrated or visual references can be perturbed. Another plausible attack is the capture and reverse engineering of drones by an adversary. In Euclidean space, information from three different non-faulty drones is sufficient to determine the path. The risk of success for this attack is low assuming drone capture and reverse engineering take much longer time than the mission of the drones.

The shared-information path planning algorithm, described in Section III, can be generalized to a d-dimensional space, where d is two or three. From Euclidean geometry, it is known that a sphere in d-dimensional space is uniquely determined by $d+1$ pairwise different points on its outline (perimeter or surface, respectively). Therefore if $n\geq d+1$, then there is a unique geometric object S determined by the n points. When $d=2$ the object is a circle, and when $d=3$ it is a sphere. It follows that if the drones broadcast their own positions to each other then they can all determine Q as the point at which the line denoted by $L(\vec{c},\vec{v})$ intersects with the outline of object S, where $\vec{c}$ is its centre.

Suppose that at most f of the n drones are faulty in the sense that they do not provide their correct location to the rest of the swarm either because they are malfunctioning or their positioning system is the victim of an attack. The remaining $n-f$ drones are reliable and broadcast their correct location to the peers.

> Is it possible for the reliable drone to correctly solve the localization problem and point Q despite the presence of the f faults?

Observe that communication may be reliable but data may not be because some drones may have their local system hacked. The main point here is that although the drones know from the broadcast each other's coordinates they do not know in advance which among them are reliable and which are not. In particular, although the drones can broadcast their coordinates to the other drones *correctly* and compute the correct set of coordinates of all the drones, they have no way of verifying their authenticity. Our motivation is inspired from the work of Kleinberg [15] which proposed the "circle" paradigm and looked at the case of non-faulty drones ($f=0$).

We show that the localization problem is correctly solved assuming that the *majority* of the drones are reliable. We make this precise by proving the following theorem.

Theorem 1: Consider a localization problem $\mathcal{L}(P_1,\ldots,P_n;\vec{v})$ for n drones at most f of which are faulty. Let $\vec{v}$ be a direction vector in d-dimensional space known to all the drones.

1) If $n\leq 2f+d$, then there is a d-dimensional object S with center $\vec{c}$ on whose outline at least $f+d$ (which is $\geq n-f$) of the drones can be located in such way that the intersection of the straight line $L(\vec{c},\vec{v})$ and outline of S is not the correct destination point Q.
2) If $n>2f+d$, then there is a unique object S with center determined by $\vec{c}$ on whose outline at least $n-f$ (which is $>f+d$) of the drones are located and the correct destination point Q can be uniquely determined as the intersection of the straight line $L(\vec{c},\vec{v})$ with the outline of S, despite the presence of the f faulty drones.

Proof: We give the proof only for points in the plane ($d=2$). However, the general case in d-dimensional space is merely a reformulation of this special case.

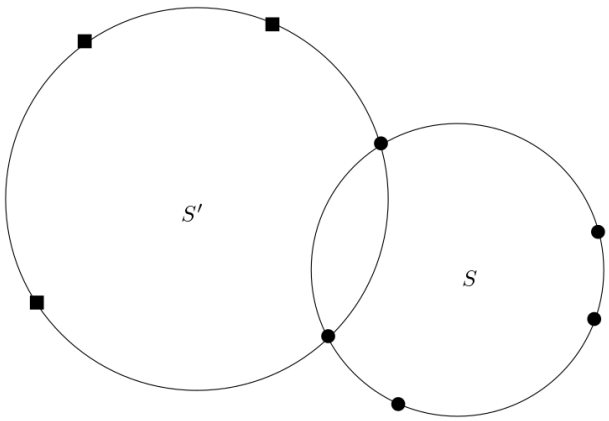


Fig. 4. An arrangement of $n=8$ drones with $f=3$ faulty. Black dots represent reliable drones and black squares faulty drones.

(Part 1) First, let us consider the case where $n\leq 2f+2$ (see Figure 4 for an arrangement of $n=8$ drones with $f=3$ faulty drones; black dots represent reliable drones and black squares unreliable drones). If $n-f\leq 2$ then there is more than one circle passing through the reliable drones and so the point Q cannot be uniquely determined. So, without loss of generality we may assume that $n-f\geq 3$. In this case, the reliable drones determine a unique circle, say S. The remaining f faulty drones may therefore use any two among the $n-f$ drones (a total of $\binom{n-f}{2}$ possibilities) to form a different circle S' that also specifies a different point Q. This gives rise to two circles. The first one S' determined by the $f+2$ drones and the

second S by the $n-f$ drones. Moreover, since $n-f \leq f+2$ the drones cannot determine the circle formed by the reliable drones.

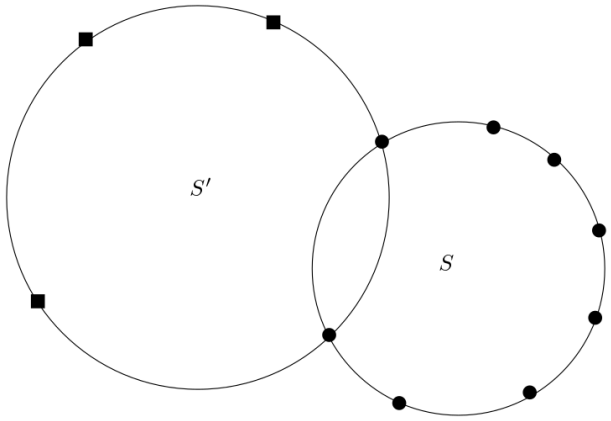


Fig. 5. An arrangement of $n = 11$ drones with $f = 3$ faulty. Black dots represent reliable drones and black squares unreliable drones.

(Part 2) Clearly, the $n-f$ reliable drones lie on the outline of a d-dimensional object (here we consider the case of $d = 2$). We would like to show that if $n - f > f + 2$, then this circle is unique and the faulty drones cannot determine a different circle. If $n > 2f + 2$ (see Figure 5 for an arrangement of $n = 11$ drones with $f = 3$ faulty; black dots represent reliable drones and black squares unreliable drones) then the reliable $n-f$ drones determine a unique circle that involves all the non-faulty drones. This is because $n - f > 2$ and from Euclidean geometry we know that a circle is uniquely determined by three pairwise different points on its perimeter.

The f faulty drones may try to produce a different circle S' so as to mislead the reliable drones. To this end the unreliable drones can only co-operate with at most two reliable drones (without their knowledge) in order to form a different circle that uses $f + 2$ drones. Clearly, the faulty drones cannot co-operate with three or more reliable drones, because we know from Euclidean geometry that three points in the plane determine a unique circle, which is also the circle determined by the $n-f$ reliable drones. However, by assumption $n > 2f+2$. Therefore the correct circle is the one determined by the $n-f$ reliable drones which by our assumption are in the majority since $n - f > f + 2$. ■

V. Simulations

A. Simulation tools

In order to validate our work, the path planning algorithm presented in Section III has been integrated into the OMNeT++ (Objective Modular Network Testbed in C++) framework [16], [17]. The implementation leverages a series of shared libraries over INET [18], an OMNeT++ model suite for the simulation of wired, wireless and mobile networking protocols, including UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, 802.11, MPLS, OSPF, and many others.

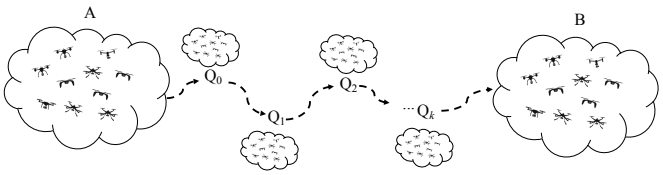
Together, OMNeT++ and INET allow the development of discrete event simulations, in which models are described in the C++ language [19] and network properties are specified in the NED (NEtwork Description) language [20]. In a nutshell, INET simulations consist of networking modules that communicate with each other through message passing (either via OMNeT++ gates, or through direct messages). The use of add-ons, e.g., via shared libraries, makes possible complementing OMNeT++ code with functionality from other simulation engines. This add-on architecture can be used to extend the INET framework for the development of satellite-based communication simulations. As a result, we can use frameworks like OS3 (Open Source Satellite Simulator) [21] and GNSSim (Global Navigation Satellite Simulation System) [22], [23]. Together, they allow the development of OMNeT++ simulations combining satellite and navigation communication protocols, and attacks like GPS jamming and GPS spoofing.

OS3 includes functionality to import real satellite tracks and weather data. This functionality is then used by GNSSim to implement a series of navigation services, such as GPS, GLONASS, Galileo, and BeiDou [24]. The implementation of two GPS related attacks, GPS spoofing and GPS jamming, are also included in the last version of GNSSim [23]. These two attacks are used in our simulation experiments to validate the feasibility of our path planning algorithm to correct the intentional disruption of a series of malicious drones in a swarm, under a linear path mobility experiment. The threat model consists of zombie drones, under the control of a remote adversary conducting a cyber-physical attack [25]–[27]. The zombie drones perpetrate GPS attacks on their neighboring drones of the swarm in order to disrupt their mission. The victims of the attacks are captured by the zombies and taken away from the swarm.

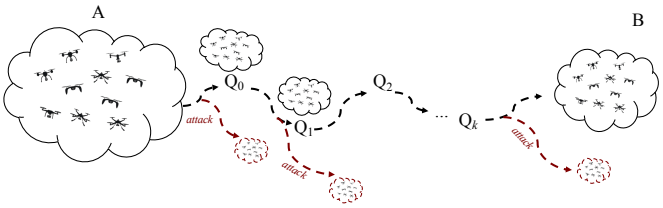
B. Simulation scenario and early results

We report next our simulation scenario and some numeric simulation results. Firstly, we summarize below the main entities and details underlying our simulation scenario:

- Zombie drones are drones from the initial swarm which get hacked by a remote adversary (e.g., a remote attacker conducting a cyber-physical attack [25]–[27]). Zombie drones become perpetrators of GPS attacks (i.e., jamming and spoofing attacks) with the objective of disrupting the remaining drones of the swarm.
- There is a variable number of zombie drones under the control of a remote adversary. Each zombie drone captures a variable number of neighboring drones (i.e., non-zombie drones that are within the coverage range of the GPS attacks of the zombie drones).
- The victims of the zombie drones are pulled out from their normal trajectory, i.e., from the swarm. They are referred to as captured drones.
- Each attack is perpetrated only at a given waypoint, i.e., at each waypoint, new drones are compromised, and may become either zombie drones (i.e., victims of the



(a) Simulation scenario baseline (without attacks)



(b) Defense strategy (under GPS jamming and spoofing attacks)

Fig. 6. Simulation scenario. (a) depicts a swarm of n drones, starting at point A and cooperating (via the path planning algorithm presented in Section III) to reach point B, after visiting k intermediate waypoints (i.e., $Q_0, Q_1, Q_2, \ldots, Q_k$). (b) depicts a series of zombie drones (under the control of the remote adversary) and captured drones (disrupted by GPS jamming and spoofing attacks perpetrated by the zombie drones). Both victim types in (b) fail at reaching the waypoints of the path, and get lost forever. Only a few survivor drones from the original swarm succeed at reaching the final destination.

remote adversary) or captured drones (i.e., victims of the zombie drones). The remaining drones which successfully escape from the remote adversary and zombie drones, are denoted as survivors.

- Each zombie drone leaves the swarm in order to conduct the GPS attacks (both GPS jamming and spoofing), as indicated in [23]. In other words, the zombie drones never recover from their attack. Hence, the attack perpetrated by the zombie drones is self-destructive.

Figure 6 depicts a more elaborated representation of our simulation scenario. In Figure 6 (a), we assume a swarm of n drones. All the drones start the mission at point A. They must cooperate to reach point B as final destination, after visiting k intermediate waypoints (i.e., $Q_0, Q_1, Q_2, \ldots, Q_k$). They execute the path planning algorithm presented in Section III in order to identify the waypoints and complete the mission. Under the absence of the adversary, all the n drones reach point B, in the end.

Parameter	Value
Mobility type of satellites	*SarSGP4Mobility*
Mobility type of drones	*PathPlanningMobility*
Transmitter power	*500 watts*
Packet interval	*0.5 seconds*
Burst duration	*10 seconds*
Sleep duration	*0 seconds*
Position update interval	*1 second*
GPS Jamming attack range	*100 km*
GPS Spoofing attack range	*100 km*
Drone communication range	*80 km*

Fig. 7. GNSSim parameters used in our simulations.

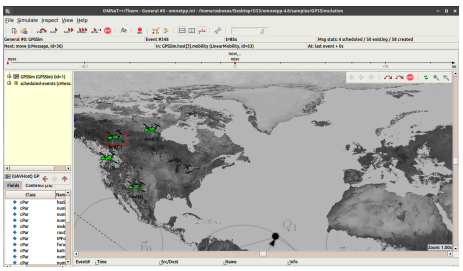


Fig. 8. Sample visualization capture of our ongoing simulation testbed using OMNeT++ [16], OS3 [21] and GNSSim [22], [23]. Some additional videocaptures of our ongoing simulation testbed are available at http://j.mp/gnssimuav.

In Figure 6 (b), the previous scenario changes due to the presence of a remote adversary which is capable of infecting some of the drones in the swarm, with the objective of disrupting their mission. Such drones, i.e., the *zombie drones*, are now under the control of the remote adversary. Commanded by the adversary, the zombie drones perpetrate the GPS attacks (both jamming and spoofing attacks) in order to disrupt their neighboring drones (i.e., drones under the coverage zone of the zombies). The victims of the GPS attacks, i.e., the *captured drones*, get disrupted and also fail at reaching the following waypoint. In the end, they fail their mission. The remaining survivor drones (i.e., those not affected by the adversary or the zombie drones) succeed at unraveling the intermediate waypoints and successfully reach the final destination (point B). The defense strategy depicted in Figure 6 (b) also assumes that the victims of the adversary, i.e., the zombie and captured drones, get lost forever.

Figures 7 and 8 show some other important parameters used in our OMNeT++ simulations, as well as some captures of the OMNeT++ GUI using OS3 [21] and GNSSim [22], [23]. Some simulation videocaptures are avail-

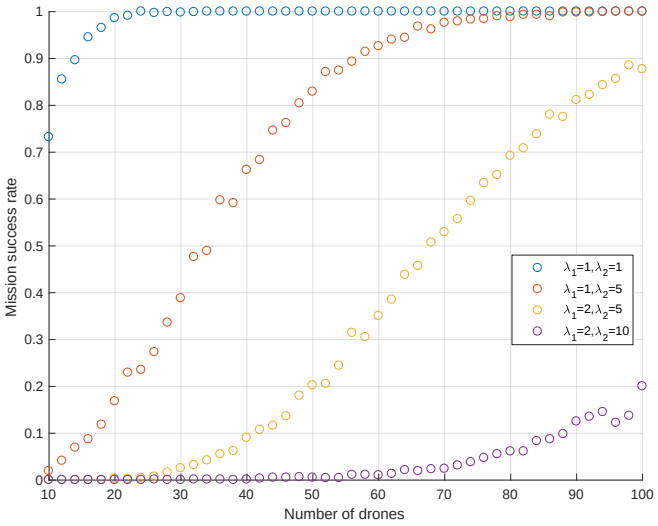


Fig. 9. Simulation assuming the number of zombies per attack follow a Poisson distribution with parameter λ_1 and number of victims per zombie follows a Poisson distribution with parameter λ_2.

able at http://j.mp/gnssimuav. Figure 9 plots the numeric results. At every waypoint, the number of zombies is a random variable that follows a Poisson distribution with parameter λ_1. The number of victims per zombie is also a random variable that follows a Poisson distribution with parameter λ_2. The following λ_1, λ_2 pairs have been simulated: $(1, 1)$, $(1, 5)$, $(2, 5)$ and $(2, 10)$. For a mission to succeed, the number of drones at the end (at point B) must be greater than zero, since no waypoints are assumed to be computed in point B, as depicted in Figure 6. For given values of parameters λ_1 and λ_2, the simulation confirms that the success rate grows consistently with the number of drones; while greater values for the parameters λ_1 and λ_2 translate in higher impact and less chances of mission success.

VI. Conclusion

We have presented an information sharing path planning algorithm for drone swarms. The drones in the swarm do not have any knowledge about the entire path, nor the final destination. They must determine and unravel a series of intermediate steps by conducting a collective process, resolving a localization problem at each step. Likewise, they must work collectively. At each step, the algorithm executed by the drones is geocaching inspired. Geocaching is a well-known outdoor recreational activity in which participants use GPS receivers (or any other alternative navigational technique) as well as a variety of scene cues (e.g., visual clues or references to landmarks) to collectively seek objects while navigating a waypoint trajectory. In our work, the series of waypoints allows the drones to build a shared-information path planning process. The goal is to have a fault-tolerant process, resilient to traditional attacks in drone scenarios, such as GPS jamming and spoofing. Our algorithm have been formalized, analyzed and validated via numeric simulations.

Acknowledgements — The authors gratefully acknowledge financial support from the Natural Sciences and Engineering Research Council of Canada (NSERC), the Cyber CNI Chair of the Institut Mines-Télécom (cf. http://chaire-cyber-cni.fr/), and the European Commission (H2020 SPARTA project, under grant agreement 830892).

References

[1] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, "Overview of path-planning and obstacle avoidance algorithms for UAVs: A comparative study," *Unmanned Systems*, vol. 06, no. 02, pp. 95–118, 2018.

[2] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.

[3] Y. Zhao, Z. Zheng, and Y. Liu, "Survey on computational-intelligence-based UAV path planning," *Knowledge-Based Systems*, vol. 158, pp. 54–64, 2018.

[4] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, 1986.

[5] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Department of Computer Science, Iowa State University, Tech. Rep., 1998.

[6] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[7] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.

[8] A. A. A. Rizqi, A. I. Cahyadi, and T. B. Adji, "Path planning and formation control via potential function for uav quadrotor," in *2014 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, June 2014, pp. 165–170.

[9] J. P. Fuentes, D. Maravall, and J. de Lope, "Entropy-based search combined with a dual feedforward-feedback controller for landmark search and detection for the navigation of a UAV using visual topological maps," in *ROBOT2013: First Iberian Robotics Conference*, M. A. Armada, A. Sanfeliu, and M. Ferre, Eds. Cham: Springer International Publishing, 2014, pp. 65–76.

[10] L. Jayatilleke and N. Zhang, "Landmark-based localization for unmanned aerial vehicles," in *2013 IEEE International Systems Conference (SysCon)*, April 2013, pp. 448–451.

[11] Z. B. Tariq, D. M. Cheema, M. Z. Kamran, and I. H. Naqvi, "Non-GPS positioning systems: A survey," *ACM Comput. Surv.*, vol. 50, no. 4, pp. 57:1—57:34, Aug. 2017. [Online]. Available: http://doi.acm.org/10.1145/3098207

[12] S. Park, H. Kim, K. Kim, and H. Kim, "Drone formation algorithm on 3d space for a drone-based network infrastructure," in *IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Setember 2016, pp. 1–6.

[13] O. Shrit, S. Martin, K. Alagha, and G. Pujolle, "A new approach to realize drone swarm using ad-hoc network," in *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, June 2017, pp. 1–5.

[14] H. Duan, Q. Luo, Y. Shi, and G. Ma, "?hybrid particle swarm optimization and genetic algorithm for multi-uav formation reconfiguration," *IEEE Computational Intelligence Magazine*, vol. 8, no. 3, pp. 16–27, August 2013.

[15] J. Kleinberg, *E Pluribus Unum, in* "This Will Make You Smarter: New Scientific Concepts to Improve Your Thinking" *(J. Brockman, editor)*. Harper Perennial, 2012.

[16] A. Varga and R. Hornig, "An overview of the OMNeT++ simulation environment," in *1st International conference on Simulation tools and techniques for communications, networks and systems & workshops (Simutools)*, 2008.

[17] The OMNeT++ Network Simulation Framework, Last Access: 2014, available at http://www.omnetpp.org/.

[18] The OMNeT++/INET Framework, Last Access: 2014, available at http://inet.omnetpp.org/.

[19] J. Heijmans, A. Paalvast, and R. V. der Leij, "OMNeT++ Discrete Event Simulation System — User Manual," Last Access: 2014, available at http://www.ewh.ieee.org/soc/es/Nov1999/18/manual/usman.htm.

[20] ——, "NED language overview," Last Access: 2014, available at http://www.ewh.ieee.org/soc/es/Nov1999/18/ned.htm.

[21] B. Niehoefer, S. Šubik, and C. Wietfeld, "The CNI open source satellite simulator based on OMNeT++," in *6th International ICST Conference on Simulation Tools and Techniques*, 2013, pp. 314–321.

[22] F. Jahan, A. Y. Javaid, W. Sun, and M. Alam, "Gnssim: An open source gnss/gps framework for unmanned aerial vehicular network simulation," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 2, no. 6, pp. 1–13, 2015.

[23] A. Y. Javaid, F. Jahan, and W. Sun, "Analysis of global positioning system-based attacks and a novel global positioning system spoofing detection/mitigation algorithm for unmanned aerial vehicle simulation," *Simulation*, vol. 93, no. 5, pp. 427–441, 2017.

[24] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[25] J. Rubio-Hernan, L. De Cicco, and J. Garcia-Alfaro, "Revisiting a Watermark-Based Detection Scheme to Handle Cyber-Physical Attacks," in *2016 11th International Conference on Availability, Reliability and Security (ARES)*. IEEE, August 2016, pp. 21–28.

[26] ——, "On the use of watermark-based schemes to detect cyber-physical attacks," *EURASIP Journal on Information Security*, vol. 2017, no. 1, p. 8, 2017. [Online]. Available: http://dx.doi.org/10.1186/s13635-017-0060-9

[27] ——, "Adaptive control-theoretic detection of integrity attacks against cyber-physical industrial systems," *Trans. Emerging Telecommunications Technologies*, vol. 32(09), 2017. [Online]. Available: http://dx.doi.org/10.1002/ett.3209