

RESEARCH ARTICLE

Adversarial scenarios for herding UAVs and counter-swarm techniques

Bruno L. Mendívez Vásquez^{1,*}  and Jan Carlo Barca²

¹Monash University, Melbourne, Australia and ²Deakin University, Melbourne, Australia

*Corresponding author. E-mail: bruno.mendezvasquez@monash.edu

Received: 22 June 2022; **Revised:** 30 November 2022; **Accepted:** 4 December 2022; **First published online:** 6 January 2023

Keywords: Unmanned Aerial Vehicle, consensus protocol, targeted elimination, herding, adversarial control

Abstract

The present paper aims to design and simulate an adversarial strategy where a swarm of quadrotor UAVs is herding anti-aircraft land vehicles (AALV) that actively oppose the swarm's objective by potentially taking them down. The main strategy is to block the AALVs' line of sight to their goal zone (AALVs' objective), shifting its trajectory so it reaches a kill zone instead (UAVs' objective). The counter-swarm strategy performed by the AALVs consists of taking down the closest aerial units to the goal zone. As a result, a consensus algorithm is executed by the UAVs in order to assess the communication network and re-group. Consensus is based on the propagation of local observations that converge into a global agreement on a communication graph. Re-grouping is done via positioning around the kill zone vector or preferring an anti-clockwise formation to better close gaps. The adversarial strategy was tested in an empty arena and urban setting, the latter making use of a path-planning procedure that re-routes the AALV trajectory based on its current destination. Simulation results show a maximum UAV mission success rate converging to roughly 80% in the empty arena. When targeted elimination procedures are executed, UAV mission performance drops 5%, making no distinction between re-grouping strategies in the empty arena. The urban setting shows lower performance due to navigation complexity but favors the decision to re-group based on a formation that close gaps rather than positioning around the kill zone vector.

1. Introduction

Shepherding behavior refers to external agents (shepherds) influencing the movement of another group of agents (flock) [1]. The forces involved in such influence are repulsive ones, so the flock can move away from the shepherds. By controlling the formation of shepherds, so it “encloses” the flock, the repulsive forces at play can ultimately redirect the motion of the flock toward some designated goal. One form of herding is *herding*, which consists of shepherds steering a flock toward a designated region.

In the realm of multi-agent systems, implementing a herding behavior was inspired by the motion of fish, birds and other animals that live in groups. Subsequent advancements in kinematic modeling allowed artificial swarms to conduct herding tasks such as crowd control, which is another definition for shepherds steering a flock toward a goal. In ref. [2], the author implemented a V-formation control algorithm that can be flexible in terms of the number of shepherds required for successfully cornering the flock, due to the increased effectiveness discussed in ref. [1]. It was later found that shepherd movement based only on the agents' center of mass is effective [3], although cooperation and/or synchronization was not considered. Other non-traditional herding methods involve the abstraction of the flock's geometry as a deformable blob [4], which is useful when the flock has a complex motion or grows considerably in size. In general, herding methods and their outcomes thrive in simulation environments but real applications are intended for swarm robotic systems [5]. Some other examples found in the literature for herding tasks are navigation with unmanned vehicles through challenging terrain, estimation of area scope, guiding birds away from airports, etc. [5].

More complex implementations of artificial herding can be found in ref. [6], where reactive models lead the control methodology. Agents respond to stimuli originating from the environment (e.g., speed between herd and flock, formation of the flock, herd size and obstacle density). Ref. [7] considered a more realistic sensing procedure for the flock, where kinematic forces are governed by the ability of estimating a neighborhood density metric. This improves control stability, and it is more appropriate for robotic implementations. In ref. [6], the discovery of a phase transition is introduced, discussed as an environmental state, which depends on obstacle density and flock size, serving as a boundary for a complexity increase in herding, useful for analyzing the limitations of swarm controllers. In ref. [8], the added complexity to herding behavior consists of the herd collecting rogue flock units back to the formation, as well as leading all of them to the herd's goal zone. Adding noise to the sensed information (actuators only) was done to test a reinforcement learning (RL) procedure. It is thoroughly discussed in refs. [9–11] that AI is needed for cognitive approaches to design herding swarm controllers. An ontology for herding is introduced [10], based on a single architecture with two pillars: decision-making autonomy and contextual awareness for all agents. The focus is on relevant sensing and efficient communication, while the controllers rely on distributed AI algorithms.

A potential military application of herding is studied in this paper, where herders are quadrotor UAVs (Unmanned Aerial Vehicles) driving anti-aircraft land vehicles (AALVs) toward a *kill zone*. That is the main objective for the UAVs; however, AALVs are concerned about completing a patrol in a designated area by reaching a *goal zone*, opposing the decision of the UAVs. Such adversarial scenario creates opportunities for defense strategies adopted by the AALVs, such as counter-swarming via target elimination. The initial UAV response is considered to be the formation's assessment in terms of connectivity. Due to the decentralized nature of UAV controllers, the coordination of motion requires agents to assess their vicinity in order to properly enclose the AALV units on the ground. If one or multiple aerial units are compromised, the swarm must be able to evaluate its structure and reach a decentralized consensus on the current state of the communication topology. Once the consensus is reached, formation changes can be performed so the mission can still be achieved.

This paper focuses on the implementation of an adversarial model for aerial swarms versus land vehicles under different environmental scenarios. The list of contributions is as follows:

- The proposal of a 2-D kinematic model for a swarm of UAVs herding a group of land vehicles with counter-swarm capabilities.
- The proposal of a consensus algorithm that triggers when land units take down elements of the UAV swarm. The ripple effect on the swarm state can help the group to perform formation changes quickly, benefiting the herding goal.
- The study of targeted elimination maneuvers by the land units, considering gaps in the UAV formation in order to calculate which UAV to take down, maximizing the probability of mission failure for the swarm.
- The extension of the 2-D herding model to consider path-planning features when the environmental scenario is an urban setting.
- The experimental evaluation via simulations, in order to assess the best strategies for both the swarm and land units based on mission success rate.

The rest of the paper is organized as follows. Section 2 comprises a literature review of the different concepts explored in this paper's contributions. Section 3 introduces the 2-D formulation for the dynamics of every agent. Section 4 considers the outcome of a targeted elimination procedure and the way UAVs can assess the state of the network afterward. Section 5 proposes algorithms that implement counter-swarm strategies in order to take down aerial units, hoping that their reduced "vision" can be translated into reaching the goal zone more often. In addition, re-grouping techniques are discussed as a response from the UAVs. Section 6 explains the dynamics of a basic kinematic controller for both UAVs and AALVs, when they both execute their adversarial behavior. Section 7 indicates the simulation setup for the proposed adversarial mechanism with the numerical results. Conclusions are given in Section 8.

2. Related work

A general assumption regarding communication capabilities of UAVs is a two-way communication link between agents, only established if they are within a circular communication range [12] (also called situated communication [13]). It can also be assumed that the communication channel is reliable; however, simulating packet losses can be beneficial for measuring the convergence speed of the consensus protocol. When modeling a communication topology under a situated communication scenario, a graph can be used to describe wireless links between agents. The entire topology status can then be measured in terms of total connectivity, or whether or not any two agents can communicate to each other in the network. When it comes to consensus-achieving algorithms, there are many approaches found in the literature, all more or less related to multi-agent scenarios that require a homogeneous assessment of a swarm network.

In ref. [14], a minimum spanning tree was derived from a dynamic and decentralized setting in order to detect damaged sections in aerial vehicles' skins (via detection of connectivity disruptions). Similarly, network nodes can be aware of the overall size in terms of node count when the topology is not known a priori and there is no hierarchy in the node structure [15]. Perhaps the biggest advantage of ref. [15] is the fact that the consensus only relies on active communication with neighbors, notion taken into account for the proposed consensus procedure. It was shown in ref. [16] that the consensus speed strictly increases with the number of neighbors per node, effect that is even more pronounced with the presence of environmental noise. Other methods involve the use of a consensus matrix, which is based on the eigenstructure of the network's topology graph [17].

To assess connectivity this way, the Fiedler value can be used, as seen in refs. [13, 18, 19]. It refers to the second smallest eigenvalue of the Laplacian matrix $L = D - A$, where D is graph degree matrix and A the adjacency one. When the Fiedler value is greater than zero, the graph is connected. The Fiedler value is used in refs. [13, 19] to optimize the robot deployment process, making sure that the motion controller keeps connectivity at all times. Estimation of the Fiedler value determines how the network topology can be reconstructed from local observations. Further applications of connectivity assessment can be found in ref. [13], where a deployment algorithm is developed for robot agents so they can reach tasks at arbitrarily distant locations while satisfying graph connectivity constraints.

However, in ref. [20], such eigenvalue for the Laplacian is estimated in a decentralized and online manner, for the case when the network grows unpredictably. It uses a technique called discrete-time power iteration to reconstruct eigenvalues and their associated eigenvectors, just by sharing local variables among neighboring agents. In order to support the kinematic formulation of agents, the technique is modified to support continuous functions. Nonetheless, the main objective in ref. [20] is to derive only the algebraic connectivity metric (or Fiedler value) from local observations and information sharing, not the actual graph topology. In this paper, the motivation behind estimating a communication network is to support a decision-making process that responds to target-elimination strategies implemented by AALVs on the ground. The Fiedler value can potentially validate the inferred topology, but exchanging information regarding communication links is more valuable for a defense mechanism that aims at specifically controlling the motion of UAVs.

The design of an adversarial strategy that the UAVs can apply when the AALVs are actively trying to escape is considered to be a counter-strategy to the well-studied herding controllers discussed in refs. [1, 4, 5], where the objective is to guide a group of agents toward a certain area. Some of these controllers are bio-inspired herding mechanisms (e.g., sheepdogs guiding a herd of sheep) [2, 3] that can also be applied to swarming agents on both ends, as seen in ref. [21]. In such work, the authors design specific feedback laws to support multiple agents driving a herd of an arbitrary number of members.

In terms of these herd dynamics, it is important that motion and formation is robust enough in order to deal with local flock movement and potential external disruptions. For example, in ref. [22], local laws are used to maintain formation cohesiveness and connectivity when information is limited and node failures are unpredictable. Enhancing cohesion of a swarm by representing it via graph metrics was discussed in ref. [23]. Particle swarm optimization was used to optimize the distance between the flock

and the herd's goal zone, subject to cohesion constraints. It was found that neighborhood awareness is crucial when dealing with limited sensing range. In ref. [24], a flight planning procedure is based on learning about radio propagation characteristics in the environment, with coordination that enables passing messages with routing costs. This achieves robust communication links in the topology while also meeting swarm objectives.

Similarly, a communication backbone can be developed in real time by connecting multiple distant target locations in a decentralized manner [25]. Connectivity is preserved at all times in this proposal, by growing a logical tree based on real physical network links. This is often a distributed procedure since applications require discovery of new routes or nodes in order to complete the mission (e.g., when applied to post-disaster transportation applications like in ref. [25], or searching for a target in an unknown environment [26]). A typical follower-leader relationship can be introduced in order to derive chains of robots that also resemble a topological spanning tree [27]. Connectivity can also be maintained by having a swarm follow biological behaviors such as cohesion, separation and alignment, which were slightly modified so it can fit a general force-based kinematic model [28].

The main contribution of ref. [21] is to use potential fields to model the herd dynamics, which is a feature used in this paper's 2-D formulation proposal. Similar work is seen in refs. [12, 29] where a UAV defense system is used to intercept and capture an enemy swarm via the use of custom formations. The end goal of this approach is to escort the malicious group of UAVs outside a forbidden zone. Previous methods that rely on maintaining a graph connectivity are used in order to prevent losses in communication, due to the fact that the formation procedure is quite granular, considering phases such as deployment, clustering, formation, interception and escort.

In addition, this work focuses not only on the herding mechanism to guide AALVs toward a kill zone, but also takes into account an adversary situation where the mobile unit has an independent objective it is constantly trying to achieve (reaching the goal zone). The defense mechanism is also dynamic, meaning that the AALV group is able to perform a targeted elimination strategy against the UAVs, reducing their influence range and therefore increasing the chances of escaping, ultimately failing the mission of the swarm. Other adversarial methods found in the literature are scarce but they consider swarm-versus-swarm scenarios where UAVs are incapacitated by using numerous strategies such as individual elimination of a target via ad hoc methods, and breaking up a swarm into clusters in order to weaken it [30].

Nonetheless, a basic herding controller is considered, based on ref. [21] and extended to consider the adversarial situation previously described. A simulation setup is implemented, which considers the traversal of an urban setting, with the establishment of path-planning protocols (inspired by refs. [31] and [32]) on top of the proposed adversarial strategies. Path planning can also be considered for herd controllers, directly affecting the implementation of shepherding behavior. In ref. [33], evolutionary-based path planning is used by the herd to encounter the flock and guide it toward the objective. The novelty of this proposal is the application of waypoints, always optimizing sub-goals at every path step while avoiding obstacles. Multiple AI approaches to path planning and motion control can be found in ref. [34]. This survey investigates multiple competences for robot navigation where RL is applied. When it comes to path planning, the overall framework of RL is considered where reactive behaviors implement already established algorithms like Dijkstra or A^* . One example for motion control is to have an action space with a back propagation network so the model converges to the stated objectives in terms of traversal with obstacle avoidance.

3. 2-D formulation

A 2-D formulation is considered in order to simplify the kinematics of the herding swarm. Since the adversarial method considers UAVs versus land units, there should be a fixed distance between them for the targeted elimination procedure to be possible. At this stage, it is not considered for an UAV counter-strategy (as a response for targeted elimination procedures) to be the execution of evasion tactics

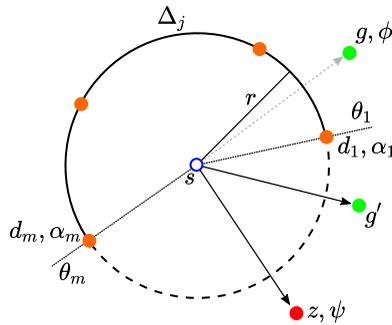


Figure 1. 2-D formulation of the adversarial herding procedure. Point s corresponds to the position of the AALV’s center of mass. Unit s attempts to reach goal g (at angle ϕ), while the UAV formation (units d_j at angles α_j) herds toward the kill zone z (at angle ψ). AALVs and UAVs always maintain a fixed r distance unless UAVs are shot down (targeted elimination counter-strategy). Point s attempts an escape by moving outside of the $[\theta_1, \theta_m]$ boundaries (toward g'). If g is out of reach, s will most likely end up moving toward z .

via changing altitude, due to its trivial nature. Instead, network assessment strategies and re-formation outcomes are proposed, hence the need for simplifying the behavior model and removing the Z axis from the formulation.

Based on Fig. 1, consider m UAVs with positions $d_j \in \mathbb{R}^2$, where $j \in \{1, \dots, m\}$, and AALVs with positions $s \in \mathbb{R}^2$. UAVs are controlled such that they drive the AALVs to a desired kill zone, located at z with angle ψ from position s . By taking the case of just one AALV located at s (or the group’s center of mass), its trajectory toward the kill zone at speed v can be defined as

$$s(t) = s(t - 1) + v \frac{z - s(t - 1)}{\|z - s(t - 1)\|}. \tag{1}$$

This works under the assumption that the AALVs will move toward the kill zone in a straight line. However, and depending on the level of autonomy, the vehicle might plan different escape routes and follow different paths. Nonetheless, since the UAVs are following the AALV group closely, their positions are conditioned to $s(t)$ with additional directional parameters that attempt to influence the orientation of the AALVs so it changes toward the kill zone. The positions for every UAV in the formation are calculated with the following equation

$$d_j(t) = s(t) + r \begin{bmatrix} \cos(\alpha_j) \\ \sin(\alpha_j) \end{bmatrix}, \tag{2}$$

where r is a fixed distance between the AALVs and any UAV, α_j is the angular orientation with respect to ψ and Δ_j the angular separation between UAVs, both defined as

$$\begin{aligned} \alpha_j &= \psi + \pi + \Delta_j, \\ \Delta_j &= \frac{(2j - m - 1)\beta}{m}. \end{aligned} \tag{3}$$

Notice the parameter β included in the previous equation. This is useful to increase or decrease the separation between UAVs in the swarm formation.

In addition, UAV motion contains a noise component in order to simulate wobbling and make the simulation more realistic. It is also a feature that allows modulation of transmission reliability, when the communication range barely covers an agent. The noise component is Perlin 2-D, which takes the form:

$$P(U(0, 1), U(0, 1))/\delta, \tag{4}$$

where δ is a noise modulation factor that controls the deviation from position d_i .

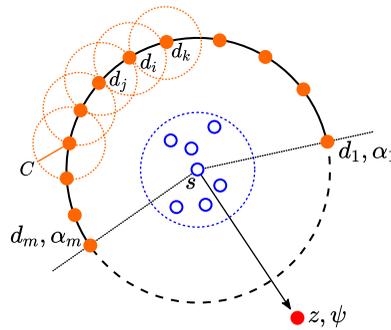


Figure 2. Formation of $m = 12$ UAVs with 3 units d_i, d_j, d_k . Due to a constant communication range C , all nodes can transmit to neighbors left and right. The consensus protocol retrieves neighbors first according to Eq. (5) and then broadcasts the discovered connections plus the learned ones so far. When receiving, only the new links are added to the local inferred topology. The procedure ends when all units have the same node length.

```

1: procedure PING( $d_i, D, C$ )
2:    $N \leftarrow \emptyset$ 
3:   for  $d_j \in D$  do
4:     if  $d_i \neq d_j$  and  $\|d_i - d_j\| \leq C$  then
5:        $N \cup \{d_j\}$ 
6:     end if
7:   end for
8:   return  $N$ 
9: end procedure

```

Figure 3. Ping procedure for individual UAVs, executed to retrieve a list of neighbors that are within the communication range C .

4. Consensus algorithm

Considering a formation of m UAVs, in order to assess connectivity and reconstruct the topology graph, all units detect neighbors by making use of the communication range with radius C . The condition to determine a communication link between two UAV units d_i and d_j is

$$\|d_i - d_j\| \leq C \tag{5}$$

Figure 2 shows a formation of $m = 12$ UAVs, all with the same communication radius C . By applying Eq. (5), UAVs can infer topology links and store them in their own version of the overall global topology. In the figure, agent d_i can infer a link to both d_j and d_k . Same applies to them, since the communication range allows for transmission to d_i . Perlin noise might affect the reach of certain units if the range is insufficient.

Figure 3 shows the algorithm to ping other UAVs in order to retrieve communication links. This procedure is decentralized, meaning it can be executed locally, where the list of current neighbors N is different for every UAV unit. For simulation purposes, the whole set of UAV positions D is available in order to determine neighbors. A realistic ping procedure would retrieve the same results as the geometric alternative shown here. The procedure makes use of Eq. (5) to determine which UAVs are inside of the communication range of local unit d_i . A list of neighbors N is then stored locally.

Figure 4 shows the broadcast procedure, executed after pinging existing neighbors. This is also a decentralized procedure, so every UAV can transmit the discovered link set L to all neighbors in N_i . In order to progressively move toward a consensus, each local copy of the inferred graph topology (G_i in Fig. 4) is sent at every broadcast call. The receive method is then called in order to process the graph estimation. Notice that this series of calls happen in a synchronous way, simplifying the actual process

```

1: procedure BROADCAST( $d_i, N_i, G_i$ )
2:   for  $d_j \in N_i$  do
3:      $L \leftarrow \{(d_i, d_j)\}$ 
4:      $L \leftarrow L \cup G_i$ 
5:     RECEIVE( $d_j, L$ )
6:   end for
7: end procedure

```

Figure 4. Broadcast algorithm for UAVs. Executed after pinging other units and determining active connections. The discovered links are sent to neighbors, along with the locally inferred topology so far G_i . UAV with position d_j then receives topology set L in order to merge it with its own learned graph.

```

1: procedure RECEIVE( $i, L$ )
2:   Let  $G_i$  be the inferred topology so far of UAV  $i$ .
3:    $L' \leftarrow \{l \in L : l \notin G_i\}$ 
4:    $G_i \leftarrow G_i \cup L'$ 
5: end procedure

```

Figure 5. UAV i receives the estimated topology L from another unit and reconstructs its local graph topology G_i based on it. It simply adds edges that are missing from the local version.

of two-way communication in wireless networks. This decentralized synchronous consensus algorithm is developed so correctness is evaluated rather than actual performance.

Figure 5 shows the procedure for individual UAVs to receive existing versions of topology graphs from surrounding neighbors. It simply reconstructs local versions G_i by examining already detected links in the received payload L , and adding the ones that are new. The basic idea is that after a series of exchanges, the network would be able to agree on a global topology that represents the actual communication graph (ground truth). Nonetheless, this is also dependent on the communication range C , movement noise and, most importantly, the existence of gaps due to target-elimination strategies executed by AALVs on the ground. The next section takes a look at how this algorithm performs under different scenarios that vary these variables.

5. Targeted elimination

Targeted elimination refers to the strategy conducted by AALVs that aims at maximizing the opportunity of reaching the goal location, as opposed to the UAVs’ mission, which consists of pushing the flock toward the kill zone. This is an addition to the basic competitive scenario that both groups follow, where the two missions are executed simultaneously. To further increase the odds of reaching the goal location, AALVs activate their defense mechanisms to take down aerial units so they stop blocking the path toward the goal. While the UAV formation re-groups, the newly formed gap can be an advantage to AALVs, perhaps reaching the goal before the aerial units can apply a countermeasure. Figure 6 shows the targeted elimination procedure in terms of the geometry of the agents. Figure 6(a) identifies the k aerial units closest to the goal zone as targets. The number of targets k is arbitrary and depends mostly on the real-world capabilities of AALVs. After performing the elimination of targets, in Fig. 6(b) UAVs attempt to fix the formation resulting in a gap that the AALVs’ center of mass s can follow toward the goal g .

To assess the capabilities of the proposed targeted elimination procedure, a measurement is performed on how the separation angle between aerial units changes when units are lost. From Eq. (3), the difference between two contiguous units is calculated as follows:

$$\begin{aligned}
 \Delta_j - \Delta_{j-1} &= \frac{2j - m - 1}{m} - \frac{2(j - 1) - m - 1}{m} \\
 &= \frac{2}{m}.
 \end{aligned} \tag{6}$$

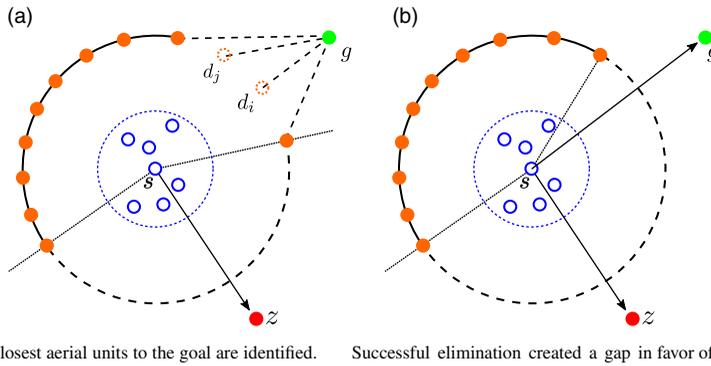


Figure 6. Targeted elimination procedure executed by AALVs. The procedure simply targets those aerial units that are closest to the goal location. In doing so, AALVs expect a gap opening that gives them direct access to the goal. However, the UAV formation might choose to cover the path to the goal when re-grouping, although that might be enough time for the AALVs to escape. Numerical results test this “gap period” and assess if translates to UAV mission failure.

In addition, how the flock coverage is affected by the targeted elimination procedure can also be calculated. Coverage is defined as the angular distance of the flock, from the first to last unit, hence

$$\begin{aligned}
 \alpha_m - \alpha_1 &= (\psi + \pi + \Delta_m) - (\psi + \pi + \Delta_1) \\
 &= \Delta_m - \Delta_1 \\
 &= \beta \frac{(2m - 2)}{m}, \tag{7}
 \end{aligned}$$

where β is the parameter used to increase coverage via larger angular separation between units. The higher the separation, the more coverage of the flock.

5.1. Modified UAV formation procedure

As a response for the targeted elimination strategy performed by the AALVs, the UAV formation always re-groups following an anti-clockwise orientation. This is the default behavior as dictated by Eq. (3), where gaps always appear for the farthest UAV units in terms of the angle α_j . To make things more fair, and to also test a different formation strategy, α_j can be calculated by splitting UAV units in equal parts from the kill zone orientation vector. Figure 7 shows the strategy for this new “fair” formation.

By taking into account the angular positioning equations that conclude with Eq. (3), the coverage equation in (7) for $\beta = 1$ is redefined as

$$\Delta_\alpha = \alpha_m - \alpha_1 = \frac{(2m - 2)}{m}, \tag{8}$$

which is the total angular coverage of the UAV flock. In order to split such region into equal parts, the space is divided among the m UAV units, yielding the angular step defined as

$$\frac{\Delta_\alpha}{m} = \frac{2m - 2}{m^2}. \tag{9}$$

The algorithm to calculate UAV positions d_j is as follows

1. Calculate the last unit’s $j = m$ (i.e., the one with the largest angle from ψ in a counter-clockwise orientation) angular component as

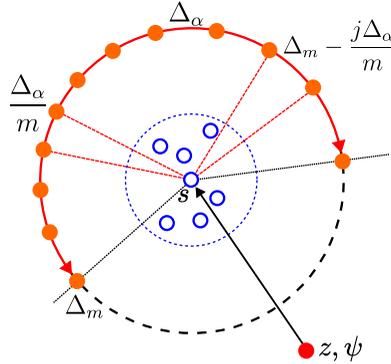


Figure 7. New formation strategy as a response to targeted elimination. The goal is to cover an equal amount of space to the left/right of the $s - z$ vector by starting from Δ_m and decreasing the angle at rate $j\Delta_\alpha/m$ where Δ_α is the total angular coverage by the swarm according to Eq. (3).

$$\Delta_{j=m} = \frac{m - 1}{m}, \tag{10}$$

according to Eq. (3).

2. Calculate the angular position for unit j as

$$\alpha_j = \psi + \pi + \Delta_m - \frac{j\Delta_\alpha}{m}, \tag{11}$$

for $j = 1 \dots m$.

3. Calculate d_j according to Eq. (2).

6. Basic controller for flock and herd

Figure 1 shows the basic 2-D formulation for agents in the herding scenario. The adversary strategy that guides both AALVs and UAVs trajectories was also briefly introduced.

Figure 8 shows the pseudo-code for the adversarial herding procedure. Such procedure is executed at every time step t , with positional inputs from a previous state (lines 2 and 3). The termination criteria consider reaching either the goal g or kill zone z , via proximity to their respective radii (r_g and r_z).

First, the boundaries $[\theta_1, \theta_m]$ for the UAV formation are calculated. These correspond to the minimum and maximum angles of d_j with respect to s (lines 5 and 6), formally

$$\begin{aligned} \theta_1 &= \min(\{\text{atan2}(d_j - s), \forall j\}) \\ \theta_m &= \max(\{\text{atan2}(d_j - s), \forall j\}). \end{aligned} \tag{12}$$

Any point x is outside the UAV formation boundary if the condition

$$\text{atan2}(x - s) > \theta_m \vee \text{atan2}(x - s) < \theta_1 \tag{13}$$

is satisfied. Then, if g is outside the UAV formation, the AALV center s moves toward g (lines 7 and 8); otherwise, s moves toward the kill zone z . After s moves to its new position, all UAVs are updated in order to maintain the herding formation and push s toward the kill zone z . To achieve this, each angle α_j is updated according to the new orientation of the vector $z - s(t)$ (lines 12 and 13).

Figure 9 shows a simulation example for the adversarial algorithm. It shows that the boundary conditions for the UAV formation indeed force the AALVs to move toward the kill zone when reaching the goal is not feasible. In the absence of obstacles, determining a clear line of sight to any destination is

```

1: procedure ADV-HERDING( $t, s, g, \phi, r_g, z, \psi, r_z, d_j, v$ )
2:    $s \leftarrow s(t-1)$ 
3:    $d_j \leftarrow d_j(t-1)$ 
4:   if  $t > 0$  and  $(\|z - s\| > r_z$  and  $\|g - s\| > r_g)$  then
5:      $\theta_1 \leftarrow \min(\{\text{atan2}(d_j - s), \forall j\})$ 
6:      $\theta_m \leftarrow \max(\{\text{atan2}(d_j - s), \forall j\})$ 
7:     if  $\text{atan2}(g - s) > \theta_m$  or  $\text{atan2}(g - s) < \theta_1$  then
8:        $s(t) \leftarrow s + v \frac{g-s}{\|g-s\|}$ 
9:     else
10:       $s(t) \leftarrow s + v \frac{z-s}{\|z-s\|}$ 
11:    end if
12:     $\alpha_j \leftarrow \text{atan2}(z - s(t)) + \pi + \Delta_j, \forall j$ 
13:    Update  $d_j(t), \forall j$  according to  $\alpha_j$  and (2).
14:  else
15:    Termination due to reach of goal or kill zone.
16:  end if
17: end procedure

```

Figure 8. Herding algorithm that implements adversarial strategies for both AALVs and UAVs in an empty arena.

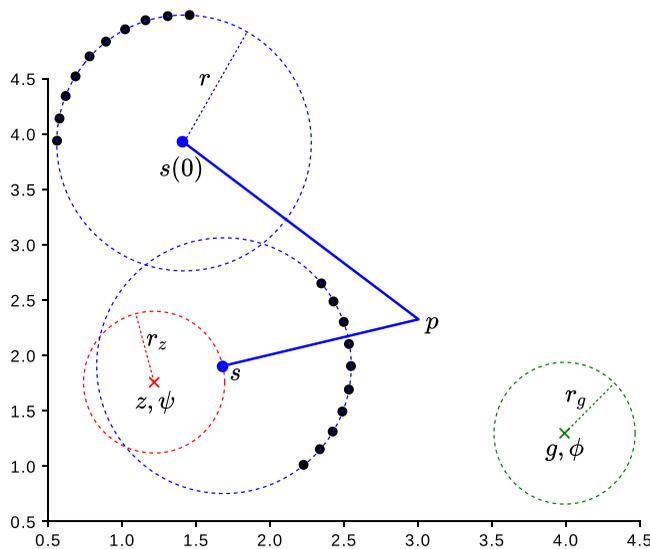


Figure 9. Simulation example with annotations based on the proposed 2-D formulation. The initial AALV center position is $s(0)$, and the trajectory for the entire simulation length is highlighted in blue. Initially, it is oriented toward the goal position g , and as the simulation progresses, the UAV herds s by shifting toward the right, aligning with the vector $z - s$ at every time step. At point p , and due to the execution of the adversarial algorithm, g is blocked by the UAV formation forcing s to move toward the kill zone z .

fairly simple, so a sudden change in direction does not undermine the possibility of reaching the goal or kill zone.

An issue could be the current orientation at the time of changing destination, especially if the distance to any final zone point is fairly short. In such situation, the AALV center depends on its own velocity to determine whether a change in course is feasible or not. Nonetheless, such feature is properly measured in the experiments which directly impacts if the UAV mission is successful or not.

```

1: procedure ADV-HERDING-A*( $s, g, z, R, r_g, r_z$ )
2:   Update  $x$  based on whether  $g$  or  $z$  satisfy (13).
3:   while  $\|z - s\| > r_z$  and  $\|g - s\| > r_g$  do
4:     Break on gridlock conditions.
5:      $P \leftarrow \text{FINDPATH}(R, s, x)$ 
6:     for  $i \in |P|$  do
7:        $p \leftarrow P_i$ 
8:       while  $\|p - s\| > r_g$  do
9:          $s \leftarrow s + v \frac{p-s}{\|p-s\|}$ 
10:        Update  $\alpha_j$  and  $d_j, \forall j$  according to (2).
11:      end while
12:      Update  $x$  based on whether  $g$  or  $z$  satisfy (13).
13:      if  $x$  changes direction then
14:        break
15:      end if
16:    end for
17:  end while
18: end procedure

```

Figure 10. Herding algorithm that implements adversarial strategies for both AALVs and UAVs in a urban setting through A* path planning.

6.1. Adversary herding with path planning

When considering a urban setting for the experiments, the arena becomes a 2-D maze based on real map data. Streets and buildings are represented as paths and obstacles, respectively. The same adversarial herding procedures are applied to this setting with some modifications.

First, locations s (AALV center), g (goal zone) and z (kill zone) are placed randomly on valid paths. Depending on adversarial conditions, the two possible destinations (goal or kill zone) are reached via path planning with the A* algorithm.

Figure 10 shows the pseudo-code for the adversary herding procedure with A* path planning. At any time, the point x refers to the current direction the AALV group is facing. Depending on the UAV formation, it could be either g or z . There is a chance that the algorithm constantly switches between these two positions, generating a gridlock condition which should be detected and avoided.

In the numerical results section, the likelihood of such event for the urban setting experiments is measured. Once a destination has been determined, the A* algorithm is used to obtain a path toward it. If such trajectory is never disrupted by a change of direction (lines 12 and 13), the AALV center s follows each point p in the path P (lines 7 to 11). The UAV formation is also updated at the same time.

Path planning in the urban setting is used to challenge the land units so it is not so easy for them to reach the goal zone. Since the UAVs are herding from above, they are considered to have a clear line of sight toward the AALVs, always re-grouping and changing their motion based on the position of the AALVs' center of mass s . A similar situation is considered for the targeted elimination procedure in the urban setting. The height of buildings and other structures that serve as obstacles for the AALVs do not interfere with the strategy that chooses which UAV to take down. As mentioned before, the urban setting is a limitation that only applies to land units trying to find a way toward the goal zone g , so it only affects the 2-D components of s and ignores any line of sight obstruction that could arise due to the maze design of the urban setting.

7. Results

To test the consensus algorithm, it is a requirement to understand the UAV formation in terms of the number of agents and the radius between the herd and the flock. Naturally, the larger the separation is between the center of the flock and the UAVs, the more space will be available between the aerial agents. This directly affects the communication range, especially if it is fixed for a particular number of UAVs and separation radius. In other words, attempts at modifying the formation due to elimination strategies

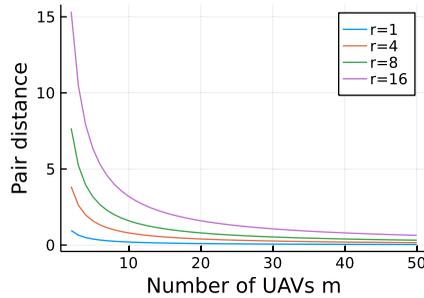


Figure 11. As the number of units increase, the formation is more dense and the pair distance drops. In contrast, an increasing radius or a decrease in number of agents (both resulting from targeted elimination from the ground) tends to increase the pair distance quite sharply, affecting the communication radius if the tolerance is not high.

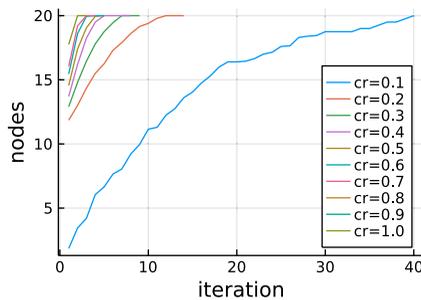


Figure 12. Consensus speed in terms of number of iterations for a formation of $m = 20$ units and flock-herd radius $r = 1$. Consensus speed decreases as the communication range gets larger.

launched from the ground will probably break the topology since gaps are now bigger. Such scenario would probably require much higher tolerances for fixed communication ranges.

Figure 11 shows the relationship between the distance between UAVs in the formation and the number of units when the flock separation radius is varied. This result is a direct measurement from the 2-D formulation in order to assess how the angular orientation α_j responds to flock-herd radius r and number of UAVs m . The overall observation is that targeted elimination strategies from the ground can cause the formation to lose agents or retreat, decreasing the number of units and increasing the flock-herd radius, respectively.

Such changes can generate a sharp increase in pair distance, meaning that gaps between UAVs increase, in detriment of the communication radius, potentially breaking the topology and any chance at a consensus result. Assuming the communication radius is fixed, a particularly high tolerance is needed in order to respond to any counter-swarm strategy launched from the ground flock of AALVs.

7.1. Consensus speed

The synchronous distributed algorithm for reaching consensus on the communication topology is limited by how far UAV units can reach to each other (communication radius). Being a constant parameter, the communication radius is set and it remains the same for the entirety of the simulation. When the topology is compromised, it is the task of the aerial agents to improve consensus by changing the formation.

The consensus speed for different communication radii ($C \in [0.1, 1]$), in a swarm of 20 UAVs is now considered. The flock-herd radius is set to $r = 1$ and the Perlin noise parameter is $\delta = 50$. The goal is to agree on the number of UAVs which is $m = 20$ (ground truth). Figure 12 shows the convergence speed (in terms of number of iterations) for different communication radii.

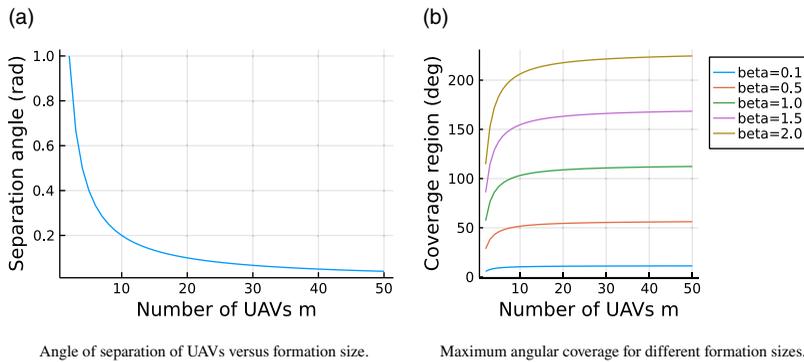


Figure 13. Targeted elimination procedure.

An iteration is conformed by all aerial units completing the ping, broadcast and receive routines and updating their knowledge of the topology graph. It is clear that consensus speed (number of iterations required) drastically improves with the communication range. The more units it covers, the faster a consensus can be reached.

As mentioned before, targeted elimination procedures disrupt the UAV formation, forcing them to close gaps so the communication range, and therefore consensus speed, remains unaffected. It will then be shown how the consensus algorithm performs when units are randomly removed.

7.2. Targeted elimination

In order to assess the performance of the targeted elimination procedure, it is important to measure the impact of the UAV herd size to the region of influence that affects the AALV flock.

Figure 13(a) shows the separation angle between UAVs changing with the formation size. The overall behavior shows an asymptotic increase of angular separation with the number of UAVs. This is considered to be an advantage for the UAVs, since a targeted elimination strategy will need to take down multiple units (e.g., potentially reducing the formation size to less than 10 units) in order to significantly increase the separation angle and therefore gaps in the formation.

Furthermore, the parameter β is directly proportional to the total angular coverage of the UAV formation. Nonetheless, the asymptotic behavior is also shown in Fig. 13(b), where the coverage region is almost constant for any number of UAVs. The formation would have to take significant damage in order to reduce the coverage region so gaps can appear.

7.3. Adversarial experiments

To test the effectiveness of the proposed algorithms, the experimental setup considers both an empty arena and a 2-D maze. These experiments were conducted using the SCRIMMAGE simulator [35] and the arena spans in length from 0 to 20 units in both axes.

Initially, the arena does not consider any particular terrain or obstacles. Later, for the urban setting, real map data are used from downtown Adelaide, Australia (Fig. 14). The AALV group will traverse these environment, using the path-planning algorithm, while the UAVs follow in the air. The base setup is comprised of random positions for the goal g , kill zone z and AALV units (whose center of mass is $s(0)$).

For the urban setting, $s(0)$ must be valid and occur on streets only. Simulations were repeated 100 times, setting the number of UAVs to $m = 20$ for the empty arena and $m = 10$ for the urban setting. The speed was set to $v = 0.01$, main radius $r = 1$ (distance between AALVs and UAVs) and both goal zone and kill zone radius to $r_g = r_z = 0.5$ (area that determines the mission outcome).

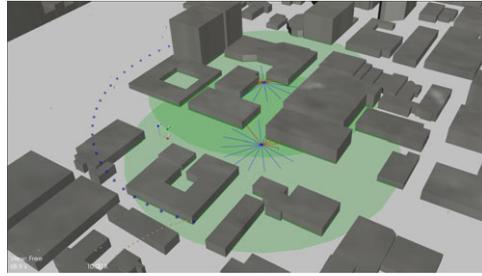


Figure 14. SCRIMMAGE simulation for the urban setting. Real map data were used from downtown Adelaide to position the AALVs and both kill and goal zones. UAVs follow while applying the proposed adversarial algorithm.

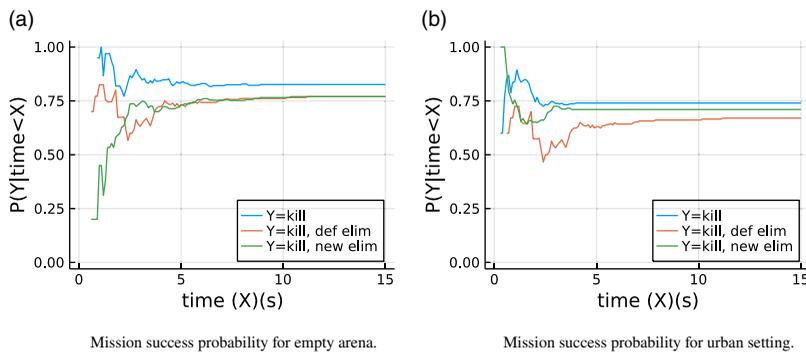


Figure 15. Results for adversarial experiments in both the empty arena and urban setting. Probability of a successful mission for the UAVs (kill) while including targeted elimination strategies performed by the AALVs plus the UAVs' modified re-grouping procedure.

Figure 15(a) shows the probability of a UAV successful mission (i.e., moving flock toward the kill zone) vs time it takes to complete it, for the empty arena scenario. Part (b) shows the same plot but for the urban setting.

A comparison was made between the adversarial execution of the scenarios without the AALVs eliminating targets, and one with the elimination procedure present. In addition, the targeted elimination was performed with two re-grouping strategies by the UAVs, the default one where units arrange themselves according to Eq. (3), and one where the formation changes according to Eq. (11) (i.e., fair positioning).

Results in Fig. 15(a) show that the targeted elimination strategy increases the probability of mission failure to approximately 5%, making no distinction on how the UAVs re-group themselves after one unit is eliminated. Similarly, the urban setting shows a similar trend in terms of mission failure probability, although it seems like the increase is more substantial for the default re-grouping strategy, while the fair positioning is not as effective. This can be explained by the limited “vision” of the UAV formation when dealing with narrow streets in the urban setting. The default strategy seems to indicate an advantage for the AALVs, due to the generation of wider “blind spots” in the UAV formation.

8. Conclusion

This paper proposed an adversarial algorithm as a response to the problem of herding AALV with a swarm of quadrotor UAVs. As opposed to common herding algorithms found in the literature, the agent being influenced is capable of focusing on its own objective (reaching a goal zone) as well as shooting down agents in the swarm, which opposes the AALV units by pushing it toward a kill zone.

Experiments were conducted by considering an empty arena and a urban setting where the AALVs can perform a targeted elimination strategy. Results confirm the predominance of the UAV swarm over the ALLVs when the adversarial objective is to block the AALVs' line of sight to the goal zone.

The success rate drops slightly when the targeted elimination strategy is in effect; however, it is also shown that the A^* path-planning algorithm successfully re-routes the trajectory of the AALVs when the swarm is able to block its line of sight to the goal.

The overall effectiveness of the targeted elimination strategy depends on the UAV formation size and parameters that define the unit separation angle and total coverage region. For example, due to the separation angle increasing asymptotically with the UAV formation size, AALVs have to take down multiple units in order to produce significant gaps that can lead them to the goal zone. Similarly, coverage region shows that asymptotic behavior with formation size, being almost constant for more than 10 aerial units, again making it difficult for AALVs to significantly reduce UAV mission performance.

UAV re-grouping strategies were also analyzed, focusing on the way units are positioned in the formation. For the empty arena scenario, the choice of strategy made no difference in mission performance, while the urban setting shows that aerial units balancing around the kill zone vector is a better choice for maintaining small drops in performance after the targeted elimination attempts.

Since the current adversarial strategy only considers an angular positioning action space, future work will focus on making the whole swarm coverage area affect the decisions of the AALV group. Specifically, the way formation gaps and goal zone coverage influence the attempt of the AALVs to "escape" the UAVs.

In addition, to further test the robustness of the proposed adversarial algorithms, the group of AALVs could improve the targeted elimination procedure to actively reduce the number of aerial units, by fragmenting the swarm into separate chunks to further limit their communication capabilities.

Acknowledgements. The authors want to thank Craig Eales (Defence Science and Technology Group) for useful discussions.

Author contributions. Jan Carlo Barca conceived the study. Bruno L. Mendívez Vásquez designed, conducted and analyzed the numerical experiments. Bruno L. Mendívez Vásquez and Jan Carlo Barca wrote the article.

Financial support. The research was supported by internal grant from the Department of Defence, Australian Government, Defence Science and Technology Group.

Conflicts of interest. The authors declare none.

References

- [1] J. M. Lien, O. B. Bayazit, R. T. Sowell, S. Rodriguez and N. M. Amato, "Shepherding Behaviors," *In: IEEE International Conference on Robotics and Automation (ICRA)*, vol. 4 (2004) pp. 4159–4164.
- [2] K. Fujioka and S. Hayashi, "Effective Shepherding Behaviours Using Multi-Agent Systems," *In: IEEE Region 10 Conference (TENCON)* (2016) pp. 3179–3182.
- [3] K. Fujioka, "Effective herding in shepherding problem in V-formation control," *Trans. Inst. Syst. Control Inf. Eng.* **31**(1), 21–27 (2018).
- [4] J. F. Harrison, C. Vo and J.-M. Lien, "Scalable and Robust Shepherding via Deformable Shapes," *In: International Conference on Motion in Games* (2010) pp. 218–229.
- [5] N. K. Long, K. Sammut, D. Sgarioto, M. Garratt and H. A. Abbass, "A comprehensive review of shepherding as a bio-inspired swarm-robotics guidance approach," *IEEE Trans. Emerg. Top. Comput. Intell.* **4**(4), 523–537 (2020).
- [6] H. El-Fiqi, B. Campbell, S. Elsayed, A. Perry, H. K. Singh, R. Hunjet and H. A. Abbass, "The limits of reactive shepherding approaches for swarm guidance," *IEEE Access*, **8**, pp. 214658–214671 (2020).
- [7] H. El-Fiqi, B. Campbell, S. Elsayed, A. Perry, H. K. Singh, R. Hunjet and H. Abbass, "A Preliminary Study Towards an Improved Shepherding Model," *In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion* (2020) pp. 75–76.
- [8] E. Debie, H. Singh, S. Elsayed, A. Perry, R. Hunjet and H. Abbass, "A Neuro-Evolution Approach to Shepherding Swarm Guidance in the Face of Uncertainty," *In: IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2021) pp. 2634–2641.
- [9] H. A. Abbass and R. A. Hunjet. *Shepherding UxVs for Human-Swarm Teaming: An Artificial Intelligence Approach to Unmanned X Vehicles* (Springer, Cham, 2021).

- [10] H. A. Abbass and R. A. Hunjet, "Smart Shepherding: Towards Transparent Artificial Intelligence Enabled Human-Swarm Teams," *In: Shepherding UxVs for Human-Swarm Teaming* (2021) pp. 1–28.
- [11] Y. Tan and Y. Shi. *Advances in Swarm Intelligence: 12th International Conference, ICSI 2021, Qingdao, China, July 17–21, 2021, Proceedings, Part II*, vol. 12689 (Springer, Cham, 2021).
- [12] M. R. Brust, G. Danoy, D. H. Stolfi and P. Bouvry, "Swarm-based counter UAV defense system," *Discov. Internet Things* **1**(1), 1–19 (2021).
- [13] N. Majcherczyk, A. Jayabalan, G. Beltrame and C. Pinciroli, "Decentralized Connectivity-Preserving Deployment of Large-Scale Robot Swarms," *In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2018) pp. 4295–4302.
- [14] M. Prokopenko, P. Wang, M. Foreman, P. Valencia, D. Price and G. Poulton, "On connectivity of reconfigurable impact networks in ageless aerospace vehicles," *Robot. Auton. Syst.* **53**(1), 36–58 (2005).
- [15] A. Saha, J. A. Marshall and A. Reina, "Memory and communication efficient algorithm for decentralized counting of nodes in networks," *PLoS ONE* **16**(11), e0259736 (2021).
- [16] Y. Shang and R. Bouffanais, "Influence of the number of topologically interacting neighbors on swarm dynamics," *Sci. Rep.* **4**(1), 1–7 (2014).
- [17] T. M. D. Tran and A. Y. Kibangou, "Distributed Network Topology Reconstruction in Presence of Anonymous Nodes," *In: European Signal Processing Conference (EUSIPCO)* (2015).
- [18] S. A. Amraii, N. Chakraborty and M. Lewis, "Studying Direct and Indirect Human Influence on Consensus in Swarms," *In: AAAI Fall Symposium Series* (2012).
- [19] J. Panerati, L. Gianoli, C. Pinciroli, A. Shabah, G. Nicolescu and G. Beltrame, "From Swarms to Stars: Task Coverage in Robot Swarms with Connectivity Constraints," *In: IEEE International Conference on Robotics and Automation (ICRA)* (2018) pp. 7674–7681.
- [20] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica* **46**(2), 390–396 (2010).
- [21] A. Pierson and M. Schwager, "Controlling noncooperative herds with robotic herders," *IEEE Trans. Robot.* **34**(2), 517–525 (2017).
- [22] D. Krupke, M. Ernestus, M. Hemmer and S. P. Fekete, "Distributed Cohesive Control for Robot Swarms: Maintaining Good Connectivity in the Presence of Exterior Forces," *In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2015) pp. 413–420.
- [23] R. E. Mohamed, S. Elsayed, R. Hunjet and H. Abbass, "A Graph-Based Approach for Shepherding Swarms with Limited Sensing Range," *In: IEEE Congress on Evolutionary Computation (CEC)* (2021) pp. 2315–2322.
- [24] W. L. Teacy, J. Nie, S. McClean and G. Parr, "Maintaining Connectivity in UAV Swarm Sensing," *In: IEEE Globecom Workshops* (2010) pp. 1771–1776.
- [25] A. Kurt, N. Saputro, K. Akkaya and A. S. Uluagac, "Distributed connectivity maintenance in swarm of drones during post-disaster transportation applications," *IEEE Trans. Intell. Transp. Syst.* **22**(9), 6061–6073 (2021).
- [26] Y. Luo, J. Guo, G. Ye, Y. Wang, L. Xie, X. Wang and X. Yan, "Toward target search approach of swarm robotics in limited communication environment based on robot chains with elimination mechanism," *Int. J. Adv. Robot. Syst.* **17**(3), 172988142091995 (2020).
- [27] R. Maeda, T. Endo and F. Matsuno, "Decentralized navigation for heterogeneous swarm robots with limited field of view," *IEEE Robot. Autom. Lett.* **2**(2), 904–911 (2017).
- [28] E. Mathews, T. Graf and K. S. Kulathunga, "Biologically Inspired Swarm Robotic Network Ensuring Coverage and Connectivity," *In: IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2012) pp. 84–90.
- [29] M. R. Brust, G. Danoy, P. Bouvry, D. Gashi, H. Pathak and M. P. Gonçalves, "Defending against Intrusion of Malicious UAVs with Networked UAV Defense Swarms," *In: IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)* (2017) pp. 103–111.
- [30] L. Strickland, M. A. Day, K. DeMarco, E. Squires and C. Pippin, "Responding to Unmanned Aerial Swarm Saturation Attacks with Autonomous Counter-Swarms," *In: Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR IX*, vol. 10635 (2018) pp. 247–263.
- [31] L. E. Parker, "Path Planning and Motion Coordination in Multiple Mobile Robot Teams," *In: Encyclopedia of Complexity and System Science* (2009) pp. 5783–5800.
- [32] M. Fu, W. Song, Y. Yi and M. Wang, "Path Planning and Decision Making for Autonomous Vehicle in Urban Environment," *In: IEEE 18th International Conference on Intelligent Transportation Systems* (2015) pp. 686–692.
- [33] S. Elsayed, H. Singh, E. Debie, A. Perry, B. Campbell, R. Hunjel and H. Abbass, "Path Planning for Shepherding a Swarm in a Cluttered Environment Using Differential Evolution," *In: IEEE Symposium Series on Computational Intelligence (SSCI)* (2020) pp. 2194–2201.
- [34] L. C. Garaffa, M. Basso, A. A. Konzen and E. P. de Freitas, "Reinforcement learning for mobile robotics exploration: A survey," *IEEE Trans. Neural Netw. Learn. Syst.* 1–15 (2021).
- [35] K. DeMarco, E. Squires, M. Day and C. Pippin, "Simulating Collaborative Robots in a Massive Multi-Agent Game Environment (SCRIMAGE)," *In: International Symposium on Distributed Autonomous Robotic Systems* (2019) pp. 283–297.