

Received May 11, 2020, accepted May 31, 2020, date of publication June 8, 2020, date of current version June 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3000774

Biologically Based Control of a Fleet of Unmanned Aerial Vehicles Facing Multiple Threats

SAMI EL-FERIK 

Department of Systems Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

e-mail: selferik@kfupm.edu.sa

This work was supported by King Fahd University of Petroleum and Minerals.

ABSTRACT This paper addresses a set of multiagent, unmanned, aerial vehicles (UAVs) in a mission within a threat-prone environment. Each UAV is considered as a nonholonomic, nonlinear model moving in a two dimensional space. The system is composed of a fleet of UAVs, competing UAVs and a target. The approach proposed in this paper is based on a biological model representing collective behavior in predator-prey systems. The fleet of UAV's (prey) is moving cohesively towards a target and they can come under attack from competing UAV's (predators) which are also expected to avoid a certain area around the target, treating it as an obstacle that they have to avoid. Prey and predators are expected to obey certain social behaviors within their respective species that include coherent motion, separation to avoid collision, and alignment. The prey are also connected through an adaptable network. During attacks, this networked population of prey could be divided into several small groups that are still connected and naturally observe the same social behavior. To identify these groups and their members, the density-based algorithm DBSCAN is used. The aim of this work is to use this biologically inspired model along with a robust feedback linearization controller to achieve both target pursuance and effective evasion from two predators. Simulation results demonstrate the different aspects and features provided in the proposed approach.


INDEX TERMS Nonlinear nonholonomic system, biology-based control, density-based clustering, adapt-then-combine (ATC) algorithm.

I. INTRODUCTION

The use of unmanned aerial vehicle technology is expected to increase and be an integral part of the operation of several commercial industries. Big tech giants - for example, Amazon with their "Amazon Prime Air" (Misener P, 2014, unpublished data) - plan to revolutionize their supply chain systems using unmanned aerial vehicles for home deliveries. The digitization of the economy and the use of artificial intelligence (AI) is catalyzing the development of smarter and more autonomous systems. The increasing use of UAVs continues to make the control design for such systems an active area of research.

Unmanned aerial vehicles belong to the class of nonholonomic systems. A common constraint attributed to this class of models is their inability to be stabilized by a smooth static-

state feedback controller, particularly because such models can usually have an underactuated structure and hence fail to satisfy *Brockett's Condition* [1] and [2]. In the literature, several control strategies have been developed to tackle this problem. Discontinuous feedback control laws [3], [4] and continuous feedback laws [2], [5] have been proposed. Reference [6] discussed the problem of achieving complete tracking of a wheeled robot using a torque nonlinear controller. They proposed two fuzzy controllers for solving this problem. Reference [8] proposed a backstepping control technique for a spherical robot for use in unmanned terrain. The backstepping control technique showed asymptotic tracking convergence to the desired trajectory, which was also verified using simulations. Researchers have also considered sliding-mode control to achieve tracking goals for nonholonomic systems. A non-smooth feedback using sliding-mode control to track the desired state function of a nonholonomic system has been studied in [4].

The associate editor coordinating the review of this manuscript and approving it for publication was Yichuan Jiang .

Several authors have addressed the tracking problem for nonholonomic systems, which is of importance to our research. Reference [9] proposed an approach where the tracking challenge is converted into a two-system stabilization problem by using a transformation and a cascade technique. A linear matrix inequality (LMI) was then designed to stabilize, that is, to track the given reference. Reference [10] proposed an improvement of [9]. The update allowed for exponential convergence to a desired trajectory. In [11], the authors derived an algorithm that allows a nonholonomically constrained rigid body to track a straight line without twirling. The authors derived a function called the steering function by differentiating the trajectory's curvature as a linear combination of the vehicle's position error, orientation error, and current trajectory curvature. A practical approach to solve the problem of tracking and visual servo control for a nonholonomic wheeled robot can be devised using videos of the desired paths [12]. Reference [13] also considered the combined tracking and visual problem; however, they included parameter uncertainties in their analysis. Positioning and distributed path planning can also be formulated as an optimization problem (see, for example, [14]). In [15], distributed path planning for a fleet of UAVs using the particle swarm optimization (PSO) method was developed and validated experimentally.

Also notable is the use of biologically inspired algorithms based on the collective behavior of animal groups to simulate real world scenarios. These biological behaviors have been found to be extremely useful in several engineering applications. Reference [16] harnessed the movement flow model of a school of fish to investigate the output power efficiency of vertical axis wind turbines as opposed to the commonly used horizontal axis wind turbines. Among other biological applications is robot rescue in times of danger, such as fire or natural disasters. Several other biological behaviors can be imitated for other applications. Our work highlights one of these applications. We propose the application of the foraging and evading behaviors of a school of fish to a fleet of mobile UAVs. The idea is easily applicable to rescue missions heading for a target.

The collective behavior of animals is incredibly fascinating [17]. The self-organizing formations of a group of birds in flight [18], schooling fish [19]–[21], or a swarming bees [22] or the hunting techniques used by carnivores in the wild [23] are among some of the behaviors actively being researched. Reference [24] stated that bioinspired schemes could be implemented in robotic systems and incorporated the foraging behaviors of bottlenose dolphins. In [25], a flocking control for systems with many agents has been developed by identifying team leaders and followers. The results were then applied to a group of nonholonomic robots. The work in [26] built on [25] by using a decentralized flocking controller for situations when the target to be tracked is fixed. In all these studies, the systems studied are composed of one specie.

The predator-prey system was considered in [27]. The study defined the term collective "in the sense of the

collective motion of defense or attack often found in behaviors of animal groups." In their assumed nonlinear model, both prey and predator motion is considered in a two-dimensional plane. A game is set between predator and prey such that the birth, death and evolution processes are determined by who is winning or losing the game. Coordinated animal group movements, defined as social behavior, serve as a defensive mechanism against predation. In [28], three simple attack strategies were studied: attacking the nearest victim, a peripheral victim or a split victim (an individual separated from the group). Comparison of these tactics was studied in [29]. The simulations showed that the "attacking the center" strategy is the least successful for the predator. From the prey point of view, social behavior is more advantageous than individualistic behavior. Fuzzy models called synflocks have been used to model fuzzy logic-based bird flocking [29]. The evolution of swarming behaviors has also been studied by [30]. Reference [31] presented a composite attacking tactic in which the predator either uses one of the simplicities mentioned above with a given probability or disperses the group and then attacks the isolated prey. The study assumed a single predator agent and used a genetic algorithm (GA) to maximize the probability of selecting a specific tactic, the distance at which the predator stops dispersing the prey, and the radius within which it will be looking for isolated prey.

In this work, we consider two predators using a simple attacking tactic wherein the nearest prey is selected. We also assume that the predators are not competing with each other. In addition, the prey are always inclined to engage in social behaviors where they obey cohesion, separation, and alignment requirements. We specifically apply the concept of mobile adaptive networks proposed by [32]. Adaptive networks have been used as the backbone of many biologically inspired networks [20]–[22], [33]. In [32], the author wrote a survey on the study of adaptive networks and advances in the field. A summary of different distribution strategies and the best strategy to use for certain applications was provided in [34]. In [35], an adaptive diffusion least-mean-squares algorithm was formulated to ensure cooperation among each node. This work has been extended by adding data-normalized algorithms and a dynamic topology [36]. Reference [37] studied diffusion in concurrent collective strategy diffusion of Multiagents. The study focused on spatial diffusion model. To improve the robustness of diffusion networks in the presence of disturbances, adaptive combiners are added to the networks as in [38]. A similar case of disturbance was analyzed by [39]. More recent advances were investigated in [40]–[42].

The use of biologically inspired control of nonholonomic systems under a single threat has been proposed in [43]. A control law based on backstepping control can introduce system oscillations, and a careful selection of the gains is required. The transformation used in the control design makes the analysis complex. The analysis assumes that the speed of the predator is certain. The case where the pursuers move at

uncertain speeds have been studied in [44]. Reference [45] proposed solutions to the flocking and target interception problems of multiple nonholonomic unicycle-type robots using a distance-based framework. They proposed control laws designed using graph theory and exploited the rigidity properties of a graph modeling the sensing/communication interactions among the robots. The authors proposed an input transformation to prove the stability of the network.

CONTRIBUTION

In this paper, building on the pioneering work of [21], [46] and [47], we propose a biologically inspired algorithm for a fleet of mobile autonomous nonholonomic systems in a multithreat environment.

- 1) **A biologically inspired predator-prey model is used to model the system:** The three actors interact according to norms similar to the collective behavior in animal groups. The fleet of UAVs acts like prey while facing attacks from adversarial or competing UAVs. The latter acts like predators and uses a simple attack technique based on the nearest victim. Both species follow collective social behavior defined as the ability to have cohesive motion, separation to avoid collision and alignment for formation control.
- 2) **The work is a direct extension of [43]:** The context of this study includes three actors: a target, multiple threats, and a fleet of UAVs. Unlike [43], each actor possesses some cognitive skills and abilities. The fleet of multi-agent UAVs can evade, move in a particular formation toward the target, and in the face of a close incoming threat, evade following a planned path. During its movement, the bioinspired fleet follows a predefined navigation trajectory under a designed tracking controller. The fleet is composed of nonholonomic homogeneous UAVs under the Pfaffian constraint, and a local feedback linearization controller is used to ensure path tracking in normal and evasive modes.
- 3) **Different dispersed groups of the original population are identified using density-based clustering:** Within the communication range and in the absence of threats, the strongly connected network, with the UAVs as nodes, moves harmoniously toward the target while maintaining a safe distance between agents. This is called the foraging phase. However, during attacks, this unified network is broken as the UAVs under perceived attack break from the group looking for safety. Several small groups build up during this phase. We propose a density-based technique to identify these groups and apply the same diffusion algorithm to keep them connected, thus forming several clusters of local networks that can reconnect together when they are within the range of communication.
- 4) **The prey can also be the predator:** The target has his own dynamic and considers the fleet of UAVs as predators. Once these predators are within its safe zone, the target follows evading paths. Thus, the role of prey changes to that of predators.

- 5) **Obstacle avoidance:** On the other hand, the threat considers the target as an obstacle. Their role as predators is to attack and hunt each member of the fleet of UAVs. They work together in a non-competitive manner in the sense that if one predator is pursuing a prey, the other predators will have to pursue different predators. Both predators and prey also have to avoid colliding with their own species and must maintain a safe distance between members of their own groups.

In summary, the novelty resides in

- the use of biology to address all the behavioral aspects of nonlinear nonholonomic systems composed of fleets of UAVs, multiple attackers, and mobile targets.
- the use of clustering techniques to identify strongly connected groups of agents in a multi-threat environment.
- ensuring robust control of the whole predator-prey-target system during evasion and during foraging while observing social behavior as previously defined.

The paper is organized as follows. Section I is an introduction and highlights the contributions of the paper. The problem is formulated in section II. We introduce the nonholonomic model, and we set up both the problem and the necessary assumptions. In section III, we build on the foundation of the navigation algorithms. First, we discuss the fish-prey algorithm and its application to UAVs. The biological behaviors of foraging fishes are modeled and applied. In section IV, we introduce density-based clustering and the requirement for coherent motion. We then design cognitive evasive strategies for UAVs in the presence of multiple threats and identify the different velocity estimates in section V. The tracking problem is treated elaborately in section VI. Control strategies are designed to track the path planned by the navigation algorithm for both single and multiple UAVs. Section VII shows the simulations, results, and discussions of the research carried out, and finally, we conclude in section VIII.

II. PROBLEM FORMULATION

A. UAV NONHOLONOMIC UNICYCLE MODEL

In this paper, we consider a fleet of multiagent UAV systems, two UAV attackers, and a target of each model as a unicycle moving in the plane. The kinematic model of the unicycle is usually described by a simple nonholonomic nonlinear model, although the methods presented can be extended to the 3-D case [23]. The particularity of this model is that the velocities are constrained and the states are not completely integrable [53]. Each of these elements is equipped with a global positioning system (GPS) and proximity sensors. A communication system connects the fleet of the UAVs, and the information exchanged is limited. The target will be either mobile or stationary, and its position is known to all UAVs. The attackers play the role of predators, are not connected and are assumed to have identifiers that allow them to distinguish themselves as friends and not foes. They are also equipped with proximity sensors and radar to locate the fleet of UAVs.

Assumption 1: The predators do not compete with each other. A particular prey cannot be attacked by a predator j if it is already in pursuit by a predator i . The predators do not collaborate during the hunt and must keep a predefined distance from each other. The predators should also keep a predefined distance from the target and hence treat it as an obstacle that they have to avoid.

This assumption is not restrictive and is in place to fulfill the following objectives:

- Avoid the obvious case where two predators run behind one prey.
- Maximize the hunt, which is a biological behavior.
- Exhibit various social behavior as the predators are assumed to be of different species. An example of such social behavior would be that they maintain a distance between them to avoid colliding with each other, especially during the hunt.

The kinematic model of a unicycle considers the steering angle (orientation or heading) θ and the Cartesian position (x, y) of the UAV moving within a plane. The inputs encompass the linear velocity u_1 and the angular velocity u_2 . The motion of the UAV under this model is constrained by the inability to make a sudden change in its inputs and therefore its positions and heading. These constraints are known as the Pfaffian constraints [54] and can be represented as a nonintegrable first-order differential equation. This puts a limit on the instantaneous maneuvers that the UAV can execute. The kinematic model of the nonholonomic unicycle UAV is as follows:

$$\begin{aligned} \dot{x} &= u_1 \cos \theta_s \\ \dot{y} &= u_1 \sin \theta_s \\ \dot{\theta}_s &= u_2 \end{aligned} \tag{1}$$

Let $q = [x \ y \ \theta_s]^T$ be the state vector of the model comprising the position and orientation states of the UAV. The generalized velocities of the UAV that form the inputs of \dot{q} are coupled and have to satisfy the following Pfaffian constraint [54]: $D(q)\dot{q} = 0$, with $D(q) = [\sin \theta_s \ -\cos \theta_s \ 0]$. This implies that the system has zero lateral velocity, and thus, all available velocities are within the null space of the matrix $D(q)$. For subsequent work, we will define the position in the plane of the UAV, target and predator as x, w^f and w^p , respectively. Each position has two components, one in the x-dimension and one in the y-dimension. Given the vector $v = [v_1 \ v_2]^T$, we define $u(v) = v/||v||$ and $v^\perp = [-v_2 \ v_1]^T$, where $v(1) = v_1, v(2) = v_2$, and $||v||$ is the \mathcal{L}_2 norm. Let \mathcal{N} be the number of UAVs; \mathcal{N}_i^f is the set of UAVs considered neighbors of UAV i ; $w_{l,k}^p$ and w_k^f represent the position of predator l and of the food at time instant k , respectively; r is the safe distance between UAVs; r_{down} is the distance between UAV and predator below which the UAV is declared inactive; r_{pp} is the distance between predators; r_{ft} is the radius of the zone around the target that predators cannot enter; r_{tp} is the sensing range of the target; R_{ps} is the sensing range of the predators; and Δt is the sampling period.

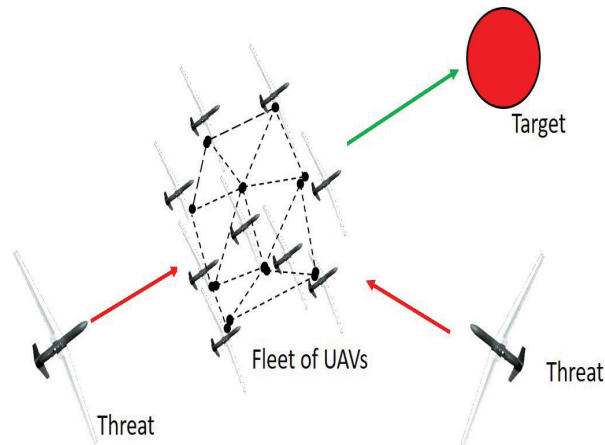


FIGURE 1. UAVs in the presence of multiple attackers.

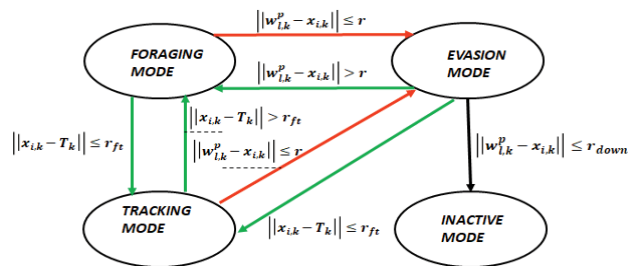


FIGURE 2. State transition diagram for UAV.

B. PROBLEM SETUP

We consider a field where we have a fleet of UAVs navigating toward a target in an environment where they can be subject to multiple attackers (see Figure 10). Each UAV has several modes of operation depending on the realization of the following events: navigation toward the target if no attackers are present within the predefined safe zone, evasion from a pursuing attacker, and hunting the target depending on whether the target is in motion or stationary. During these active modes, the UAV can be completely inactivated by the attackers. Figure 2 illustrates the automata system representing the different UAV navigation and operation modes.

Remark 1: When the UAV is inactivated by the predator or breaks away from the fleet, the network size and configuration change; thus, there is a need to consider Adapt-Then-Combine (ATC) and Combine-Then-Adapt (CTA) diffusion algorithms for the adaptive network (see [21]).

Using biology, there are four phases to this system:

- Foraging phase: The fleet of UAVs, like a school of fish, forage in search of the target. This mode occurs when there is no close threat. Like the school of fish, UAVs should observe social behavior by moving in a harmonious manner, respecting distancing, avoiding collisions and using the instinctive behavior to remain connected.
- Evasive phase: the UAV, like a prey, has to evade attackers and look for safety by breaking away from the school

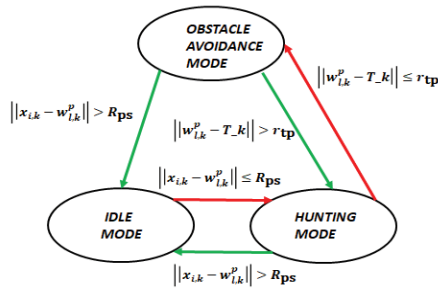


FIGURE 3. Predator state transition diagram.

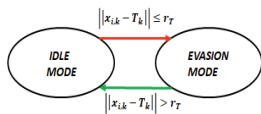


FIGURE 4. Target state transition diagram.

and adopting different velocities and headings. If the evasion is successful, the UAV has to follow a self-organizing behavior phase to reunite with the closet cluster of fish.

- Pursuit phase: If the target can be in motion, the UAV fleet starts behaving like a predator and pursues the target. If the target is stationary, the UAV fleet will gather on the target.
- Inactive phase: If a predator succeed to catch a UAV then it will be inactivated and dropped from the fleet.

The following are the assumptions regarding the system’s capabilities: **Assumptions**

- Each UAV has a GPS.
- Each UAV shares its position and orientation with its local neighborhood.
- The target position is known by all.
- UAVs are not aware of the position and orientation of the attackers until the latter enter the UAV safe zone. The predator-prey behavior is active only then.
- Each UAV is assumed to have physical dimensions [49], with motion governed by physical laws.
- The adaptive network can be in several strongly connected clusters. A cluster can be composed of only one UAV.
- For attacking UAVs assuming the role of predators, the target is an obstacle and has a zone that they cannot access. Figure 5 shows the state transition model and the conditions for moving from one state to another.

In this work, the target will follow two scenarios, either idling or moving. The following Figure 4 illustrates the two modes and the conditions for transitioning from one state to another.

III. BIOLOGICALLY INSPIRED NAVIGATION CONTROL

A. DIFFUSION ADAPTATION AND THE MEASUREMENT MODEL

Diffusion adaption is concerned with the performance of adaptive networks, which are defined as a collection of nodes

that have the ability to learn and interact with each other. In this study, we consider a collection of \mathcal{N} UAVs (nodes) distributed in space. Let $\bar{u}_{k,i}$ be a random process representing some information or measurements related to node k at time i . Every node $k, k = 1, \dots, N$, evaluates a scalar random process $d_k(i)$ as a $\bar{u}_{k,i}$, where $d_k(i)$ and $\bar{u}_{k,i}$ are correlated. Each node k calculates the unknown and possibly time-varying weights $w_{k,i}$ such that

$$d_k(i) = \bar{u}_{k,i} w_{k,i} + n_k \tag{2}$$

where n_k is a random disturbance or a perturbation, which is usually taken as zero-mean white noise. In the distributed algorithms, a node passes to its neighbors or a subset of the network, which in turn communicates the received data to other subsets of the network. Node \mathcal{N}_k ’s neighbors are a set of nodes immediately connected to it and can be defined as those within its communication range, $\mathcal{N}_k = \{l \in 1, \dots, N, l \neq k, | \|[x_k, y_k]^T - [x_l, y_l]^T\| \leq R\}$. The network aims to compute the parameter w^o such that it minimizes the global objective function [20]. In this work, the Adapt-Then-Combine (ATC) diffusion algorithm proposed by [35] and [39] is used. In this algorithm, a real positive coefficient $a_{l,k}$ is assigned to each communication link between node $k, k = 1, \dots, N$, and its neighbors $l, l \in \mathcal{N}_k$. The weighting coefficients are selected such that

$$\sum_{l=1}^N a_{l,k} = 1, \quad a_{l,k} = 0, \quad \forall l \notin \mathcal{N}_k \tag{3}$$

According to the ATC algorithm, the different weights are updated at each time instant i according to the following relations:

$$\begin{cases} \psi_{k,i} = w_{k,i-1} + \mu_k \bar{u}_{k,i}^T [\hat{d}_k(i) - \bar{u}_{k,i} w_{k,i-1}] \\ w_{k,i} = \sum_{l \in \mathcal{N}_k} a_{l,k} \psi_{l,i} \end{cases} \tag{4}$$

where μ_k is a non-negative step size applied by node k , and $\hat{d}_k(i)$ is the linearly regressed form of $d_k(i)$. Let the vector w^o represent the estimated location of the target. The distance between the UAV k and the target w^o is given by $d_k^o(i)$. UAV k is located at $x_{k,i}$. The unit direction vector from the fish location to the target is given by $\bar{u}_{k,i}^o$ and is a function of θ_k 5, where $\bar{u}_{k,i}^o$ and $d_k^o(i)$ are the exact values. The estimate $\hat{d}_k(i)$ of $d_k(i)$ takes into account any disturbance or measurement errors (see [21]).

$$\begin{aligned} d_k^o(i) &= \bar{u}_{k,i}^o(w^o - x_{k,i}) \\ \bar{u}_{k,i}^o &= [\cos \theta_k(i) \quad \sin \theta_k(i)] \\ \hat{d}_k(i) &\cong d_k(i) + \bar{u}_{k,i} x_{k,i} \end{aligned} \tag{5}$$

B. MOTION CONTROL ALGORITHM:

Using the Euler approximation, each k^{th} UAV coordinate $(x_{k,i}, y_{k,i}, \theta_{k,i})$ in the networked school is updated according to

$$\begin{aligned} x_{k,i+1} &= x_{k,i} + \Delta t. [\cos \theta_{k,i} \quad \sin \theta_{k,i} \quad 0]^T u_{1k,i} \tag{6} \\ \theta_{k,i+1} &= \theta_{k,i} + \Delta t. u_{2k,i} \tag{7} \end{aligned}$$

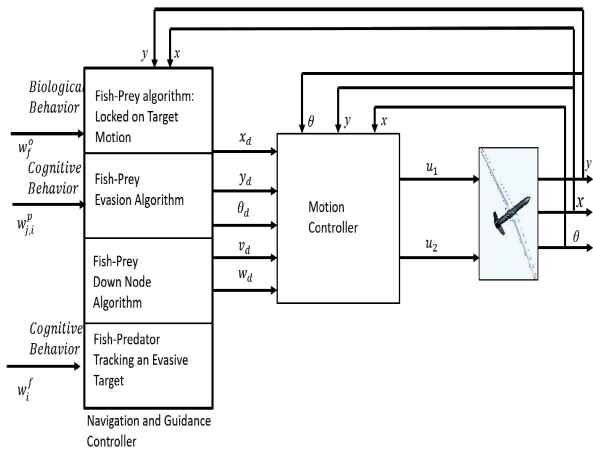


FIGURE 5. Control Schematic - Single UAV.

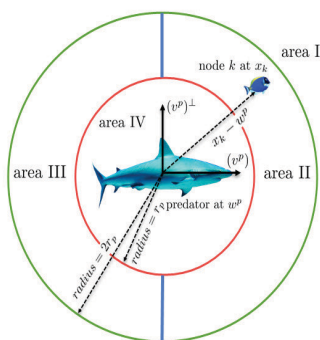


FIGURE 6. Biology-based model and zones around predators [21].

where i is the time instant and Δt represents the sampling time, which is assumed to be uniform in this paper. The inputs $u_{1,k,i}$ and $u_{2,k,i}$ are controlled in a closed loop to drive the UAV to the desired $(x_{k,i}^d, \theta_{k,i}^d)$ that varies depending on the different operational modes. Since the predators are always in motion, each UAV calculates a *local* estimate of the target location $w_{k,i}^p$ and the predator positions $w_{k,i}^p$ in real time. To further illustrate how the UAVs cognitively compute their velocities, Figure 5 gives the control configuration for each UAV.

C. MOVEMENT TO THE TARGET

This mode is valid when no predator is within the safe zone boundaries and the UAV is still active. When a particular UAV is within a predefined distance $r_{down} < r_p$ from a predator, it becomes inactive and is no longer part of the group. For all other UAVs, the scenarios illustrated in Figure 6 and formulated in Algorithm (1) apply. These different scenarios relate to zones I, II, III, and IV (Figure 6) and are used to define the next action by each active UAV.

γ is a design parameter used to balance the attractive forces in both velocity and position. v^f is the velocity of the target, and v^x is the velocity of the UAV. In this work, the target can be in motion. The indicator function I_{v^f} is equal to zero when the target is at rest ($v^f = 0$) and to 1 otherwise. UAVs are

also aware of the speed of the target and therefore adapt their velocities accordingly.

The velocity of the UAV in each of these zones can be defined with respect to the nearest predator as in Algorithm 1: To find the desired position and angular orientation of the

Algorithm 1 Fish Behavior

- 1: **Data required:** Target position, individual UAV status (active or inactive), and active UAV positions.
- 2: **Initialization:** All velocities are set to zero.
- 3: **Distance Predator to UAV:** Let $X_{i,k} = [x_{i,k} \ y_{i,k}]^T$. Compute $\|X_{i,k} - w_{l,k}^p\|$ for $i = 1, \dots, \mathcal{N}$ and $l = 1, \dots, \mathcal{N}_p$
- 4: **UAV inactive:** if $\|X_{i,k} - w_{l,k}^p\| \leq r_{down}$ then UAV i is inactive, else
- 5: **For active UAVs:** Find the nearest predator l to a particular UAV i .
- 6: **Foraging zone I (Figure 6):** if $\|X_{i,k} - w_{l,k}^p\| > 2 * r_p$

$$v_{i,k}^a = u(w_k^f - X_{i,k}) \quad (8)$$

- 7: **Evasion zones II and III (Figure 6):** Let $c_1 = \text{sign}((X_{i,k} - w_{l,k}^p)(w_{l,k}^p - w_{l,k-1}^p)^\perp)$. if $\|X_{i,k} - w_{l,k}^p\| < 2 * r_p$ and $\|X_{i,k} - w_{l,k}^p\| > r_p$

$$v_{i,k}^a = c_1 u((X_{i,k} - w_{l,k}^p)^\perp) \quad (9)$$

- 8: **Evasion zone IV (Figure 6):**

$$v_{i,k}^a = u(X_{i,k} - w_{l,k}^p)(r_p - \|X_{i,k} - w_{l,k}^p\|) \quad (10)$$

target, the following relations are used:

$$\begin{aligned} w_{d_{k+1}}^f &= w_{d_k}^f + \Delta t \cdot V_{d_k}^f \\ \theta_{d_{k+1}} &= a \tan(V_{d_k}^f(2), V_{d_k}^f(1)) \\ w_{d_{k+1}}^f &= \frac{\theta_{d_{k+1}}^f - \theta_{d_k}^f}{\Delta t} \end{aligned} \quad (11)$$

D. SELF-ORGANIZING BEHAVIOR

During evasion maneuvers, the connected network of the fleet is broken, and small groups of UAVs form several local networks (see Figure 10). These small networked groups are copies of the original group and therefore obey the same conditions of social behavior. There is a need to cluster them to recognize where they are and to reconnect them with clusters having edges within the communication zone.

IV. AGENT CLUSTERING USING DBSCAN

Clustering is a technique used in data mining and consists of finding structure in the collected data by grouping objects having similar characteristics. The resulting groups are called clusters. The similarity is often captured through features such as distance in a normed data set. Typical methods in clustering are based on solving the k-means problem. Detailed

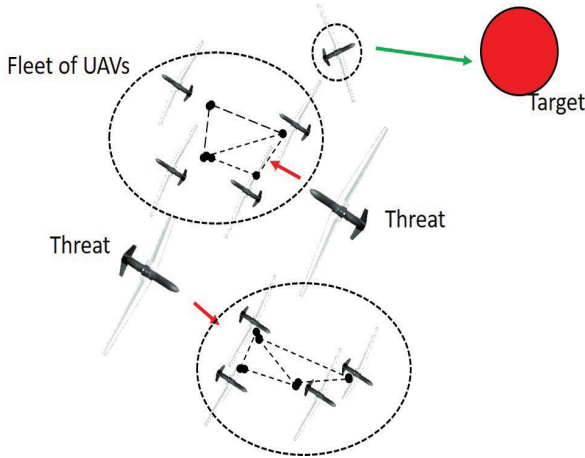


FIGURE 7. Partition of the fleet network due to predator attack.

reviews of popular clustering algorithms are provided in several books (see, for example, [58]). DBSCAN is a density-based clustering algorithm first presented in [59]. Since then, it has been implemented in many real-life applications. In addition to being available in many toolkits, the algorithm has been studied in many publications (see, for example, [57] and [55]). The main idea that governs the DBSCAN algorithm and its extensions and revisions is that points in the dataset belong to the same cluster if they are density reachable from each other [56]. Formally, let D be the dataset and $p \in D$ be a set of points in D . The following definitions are from [59], but the notation and presentation are identical to those of [56].

Definition 1 (ϵ -Neighborhood): The ϵ -neighborhood, $N_\epsilon(p)$, of a data point p is the set of points within a specified radius ϵ around p :

$$N_\epsilon(p) = \{q \in D \mid d(p, q) < \epsilon\} \quad (12)$$

where d is a prescribed distance measure and ϵ is > 0 .

Definition 2 (Point Classes [56]): A point $p \in D$ is classified as

- a core point if $N_\epsilon(p)$ has high density, i.e., $|N_\epsilon(p)| \geq \text{minPts}$ where minPts is a positive integer and the user-specified density threshold,
- a border point if p is not a core point but is in the neighborhood of a core point $q \in D$, i.e., $p \in N_\epsilon(q)$, or
- a noise point otherwise.

Figure 8 illustrates the definition of point classes in density-based clustering algorithms such as DBSCAN.

Definition 3 (Directly Density-Reachable): A point $q \in D$ is directly density-reachable from a point $p \in D$ with respect to $\epsilon > 0$ and minPts if and only if

- 1) $|N_\epsilon(p)| \geq \text{minPts}$ and
- 2) $q \in N_\epsilon(p)$

That is, if p is a core point and q is in its ϵ -neighborhood.

Definition 4 (Density-Reachable): A point p is density-reachable from q if there exists in D an ordered sequence of

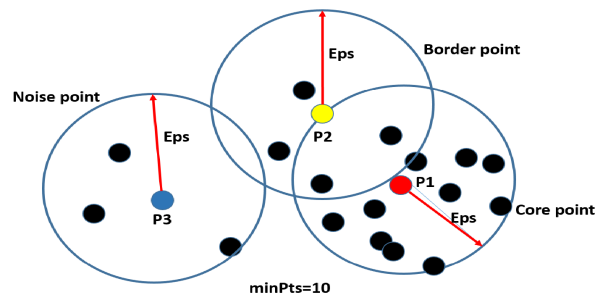


FIGURE 8. Point Classes: Core (P1), border (P2), and noise (P3).

TABLE 1. Estimating UAVs in outer boundaries of clusters.

UAV l 's Position w.r.t. k ($x_{l,i}^k$)	Value	Implication
First Coordinate	$x_{l,i}^k > 0$	UAV l is in front of k
Second Coordinate	$x_{l,i}^k > 0$	UAV l is left of k
	$x_{l,i}^k < 0$	UAV l is right of k

points p_1, p_2, \dots, p_n with $q = p_1$ and $p = p_n$ such that p_{i+1} is directly density-reachable from $p_i \forall i \in 1, 2, \dots, n-1$.

Definition 5 (Density-Connected): A point $p \in D$ is density-connected to a point $q \in D$ if there is a point $o \in D$ such that both p and q are density-reachable from o .

Definition 6 (Density-Based Cluster): A density-based cluster C is a non-empty subset of D satisfying the following conditions:

- 1) **Maximality:** if $p \in C$ and q is density-reachable from p , then $q \in C$.
- 2) **Connectivity:** $\forall p, q \in C$, p is density-connected to q .

The DBSCAN algorithm starts with a random point p and performs a depth-first search (DFS) to identify all its ϵ -neighbors. This will form the first cluster. The algorithm continues to apply the same DFS to expand the cluster, which is complete only if no more core points are found. Another random point is then selected from the remaining points, and the algorithm starts all over again. After all points are processed, points that do not belong to an identified cluster are considered noise points. Selecting the appropriate ϵ and minPts is crucial and may change the identified clusters.

Once a predator attacks the fleet of UAVs, the network is broken into smaller groups as stated previously. For their reunion and biological organization to take place, each UAV at the outer boundaries of the smaller groups must calculate the position of the other groups and advance to them. First, each UAV determines if it is on the edge:

$$x_{l,i}^k = W(v_{k,i})^T(x_{l,i} - x_{k,i}) \quad (13)$$

where $x_{l,i}^k$ is the position of UAV l with respect to k , and $W(v)$ is an orthonormal matrix defined in [21]. Table (1) is used to determine the position of UAV l w.r.t. k . The edge UAV k searches for other UAVs and pulls its related cluster toward

them using

$$v_{k,i+1}^b = \begin{cases} 0, & \text{if } \hat{l} = \text{empty set} \\ \frac{x_{\hat{l},i}^y - x_{k,i}}{\|x_{\hat{l},i}^y - x_{k,i}\|}, & \text{otherwise} \end{cases} \quad (14)$$

Algorithm 2 Network Configuration Procedure

- 1: **Data required:** Previous group average speed, updated list of UAV neighbors, individual UAV status (active or down), list of clusters from DBSCAN.
 - 2: **procedure** Update group average speed
 - 3: **for Each UAV** $i \ i = 1, \dots, N$ **do**
 - 4: **if** UAV is inactive **then** set all velocities to zero.
 - 5: **else**(only active UAVs)
 - 6: **for Each neighbor of UAV** $i \ i = 1, \dots, N$ **do**
 - 7: **Compute** required input velocity to avoid collision and respect safe distance.
 - 8: **end for**
 - 9: **Identify** clusters
 - 10: **for do**Each non-neighbor of UAV i
 - 11: **Compute** distance from non-neighbors.
 - 12: **Identify** possible new (left, right, or front neighbors) from non-neighbors.
 - 13: **Reunite** with new neighbors.
 - 14: **end for**
 - 15: **end if**
 - 16: **end for**
 - 17: **end procedure**
-

A. COHERENT MOTION

For UAVs to adequately follow the social behaviors of a school of fish, they must move collectively to confuse the predator and keep a safe distance r from each other to prevent collision with their neighbors. Therefore, the distance between UAV l and UAV k must satisfy the following equation:

$$r - \varepsilon \leq \|X_k - X_l\| \leq r + \varepsilon \quad \forall l \in \mathcal{N}_k \setminus \{k\} \quad (15)$$

where r is the predefined safe distance required between the UAVs and ε is a very small positive number. The cohesion and anticollision objectives are attained by solving an objective function (see [47]). The solved minimization problem is used to obtain the following potential field forces:

$$\delta_{i,k} = \frac{1}{|\mathcal{N}_i| - 1} \sum_{l \in \mathcal{N}_i \setminus \{i\}} \left(1 - \frac{r}{\|x_{l,k} - x_{i,k}\|} \right) (x_{l,k} - x_{i,k}) \quad (16)$$

The velocity of UAV k due to coherent motion can be defined as $v_{i,k+1}^c = v_{i,k}^g + \gamma \delta_{i,k}$, where $v_{i,k}^g$ is the local velocity value of the network's center of gravity and γ is a nonnegative number. Using the diffusion adaptation technique in equation (4),

the distributed $v_{i,k}^g$ can be estimated as follows:

$$\begin{cases} \psi_{i,k} = (1 - \mu_k^y) v_{i,k-1}^g + \mu_k^y v_{i,k} \\ v_{i,k}^g = \sum_{l \in \mathcal{N}_i} a_{l,k}^y \psi_{l,k} \end{cases} \quad (17)$$

where superscript v shows a relation with v^g and $a_{l,k}^v$ satisfies equation (3).

B. MOTION CONTROL FOR THE TARGET

V. DISTRIBUTED VELOCITY ESTIMATE

Combining all the identified biological behaviors, UAVs will adapt their velocities according to the combined velocity estimate:

$$v_{i,k} = \lambda \cdot I_{i,k} (\alpha v_{i,k+1}^a + \beta v_{i,k+1}^b) + (1 - \lambda \cdot I_{k,i}) v_{i,k}^g + \gamma \delta_{i,k} \quad (18)$$

where γ , β , α and λ are nonnegative weighting elements, while $I_{i,k}$ is a switching function that is either zero or 1 otherwise. To ensure that the distributive velocity estimate meets the nonholonomic constraints of a UAV, a maximum velocity v_{max} is set for the UAVs.

A. THE EVASION PHASE

When the UAVs are attacked by a predator, they will search to evade and look for safety by making acrobatic maneuvers to prevent such attacks from damaging or inactivating them. This scenario has been envisioned in [50]. The author conceived of a fully autonomous UAV that would be capable of executing evasive strategies when challenged by an adversary. This is the essence of the evasion phase. We propose evasive strategies whereby each UAV would track a certain planned straight-line trajectory to maneuver and evade the predator.

1) PATH PLANNING AND TRAJECTORY GENERATION

During evasion, the UAV changes its path by following a trajectory composed of a straight-line segment at an angle θ_d adjacent or perpendicular to the original direction of the UAV. Given a Cartesian plane, the line direction that the UAV would track depends on its current location and the direction from which the predator is coming. To analyze this further, the plane will be partitioned into four trigonometric quadrants to consider all the possibilities.

The UAV computes the distance d_k^p between the predator and itself in real time. At time t_0 , once $d_k^p < r_p$, the UAV senses the danger within the defined safety zone and switches from its planned foraging trajectory and social behavior to an individualistic evasion mode. The position of the UAV ($x(t_0)$, $y(t_0)$) is stored, as it would serve as a reference point during the trajectory generation.

TABLE 2. Direction of evasion path for predators.

Quadrant	Conditions	Desired Evasion Angle
1	$d_k^p < r_p$ and $\theta_p \geq \theta_{np} - 180$	$\theta_d = \theta_p + \theta_c$
2	$d_k^p < r_p$ and $\theta_p < \theta_{np} - 180$	$\theta_d = \theta_p - \theta_c$
3	$d_k^p < r_p$ and $\theta_p \leq \theta_{np} + 180$	$\theta_d = \theta_p + \theta_c$
4	$d_k^p < r_p$ and $\theta_p > \theta_{np} + 180$	$\theta_d = \theta_p - \theta_c$

The straight line for the evasion path is defined using a set of parametric equations:

$$\begin{aligned} x_{d_k} &= x_0 + \hat{\lambda}(n) \cos(\theta_{d_k}) \\ y_{d_k} &= y_0 + \hat{\lambda}(n) \sin(\theta_{d_k}) \\ \hat{\lambda}(n) &= \Delta t + (n - 1)\Delta t \end{aligned} \quad (19)$$

The variable $\hat{\lambda}(n)$ is called the path parameter. It is set as an arithmetic progression that increases to define the tracked trajectory. n is the iteration number during the execution of the evasion maneuvers.

$$I(p_j) = \begin{cases} 1 & \text{if } (x_{k,i} - w_{k,i}^{p_j})^T (x_{k,i} - w_{k,i}^{p_j})^\perp > 0 \\ -1 & \text{otherwise} \end{cases} \quad (20)$$

2) EVASION STRATEGIES

There are several ways to design evasion trajectories. In [43], four quadrants were used to compute the inverse tangent of the angles.

- The four-quadrant inverse tangent “ $a \tan 2$ ” is used for the computations of the angles because it gives an angle value between $-\pi$ and π .
- θ_{np} is defined as the actual UAV-to-predator angular direction and is used to decide to which direction the UAV turns to evade the predator.
- θ_d is the desired heading of the UAV path that leads to a safer region. θ_d is computed using the four-quadrant inverse tangent $a \tan 2$; therefore, its value is also between $-\pi$ and π .
- θ_p is the estimate of the predator’s approach angle.

In deciding the orientation that the UAV will choose to evade, we compare the angles θ_p and $\theta_{np} - 180$ and θ_p and $\theta_{np} + 180$ as follows:

a) *Quadrant 1:* In the first quadrant, once a danger approaches, i.e. $d_k^p < r_p$ and $\theta_p \geq \theta_{np} - 180$, the desired evasion angle is given as

$$\theta_d = \theta_p + \theta_c \quad (21)$$

where θ_c is the tangential angle (for example, $\pi/4$) desired for evasion. Here, the adversary UAV approaches at an angle between $-\pi$ and $-\frac{\pi}{2}$; the UAV evades the predator and tracks an orientation between 0 and π . The detailed conditions for evasion and the respective desired angles for each quadrant are given in table (2).

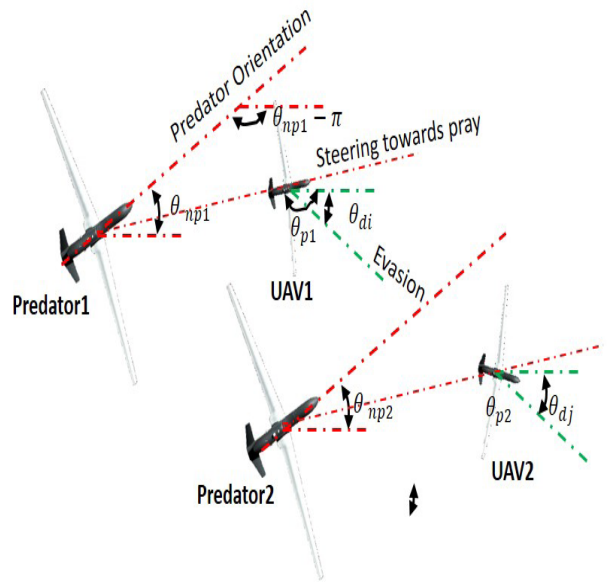


FIGURE 9. Evasion Strategy.

B. ESTIMATING THE GLOBAL VELOCITY & LOCATIONS OF THE ADVERSARY & TARGET

The goal is for the UAV to compute the adversary’s location w^p . In reality, the location of the predator or the target is obtained using a GPS. Using the ATC diffusion algorithm discussed in equation (4), the global location estimates of the predator $w_{k,i}^p$ and the target $w_{k,i}^f$ are calculated with respect to the position of each UAV. In addition, the estimates of the predator and target velocities with respect to each UAV are given by

$$v_{l,k}^p = \frac{1}{\Delta t} (w_{l,k}^p - w_{l,k-1}^p) \quad (22)$$

$$v_k^f = \frac{1}{\Delta t} (w_k^f - w_{k-1}^f) \quad (23)$$

C. PREDATOR BEHAVIORS

The predators are also nonholonomic UAVs, as stated previously. The algorithm below (3) allows computation of the required velocities of the predators. Let V_d^p , $w_{d_i,k}^p$, and $\theta_{d_k}^p$ be the desired velocity, position, and orientation of predator l at instant k , respectively. To estimate the same quantities at $k + 1$, we use the following relations:

$$\begin{aligned} w_{d_{l,k+1}}^p &= w_{d_{l,k}}^p + \Delta t \cdot V_{d_l}^p \\ \theta_{d_{k+1}} &= a \tan 2d (V_{d_k}^p(2), V_{d_k}^p(1)) \\ w_{d_{k+1}} &= \frac{\theta_{d_{k+1}} - \theta_{d_k}}{\Delta t} \end{aligned} \quad (24)$$

$\gamma_3, \gamma_4, \gamma_5$ are positive design parameters that can be selected to balance the different requirements of the velocities.

D. TARGET BEHAVIORS

The target is at rest until a UAV is within a certain distance. Once the target is in motion to flee the pursuing

Algorithm 3 Predator Behavior

- 1: **Data required:** Predators' positions, target position, individual UAV statuses (active or down), active UAV positions.
- 2: **Initialization:** All assignments of velocities are set to zero.
- 3: **Start motion** if $\|w_{l,k}^p - x_j, k\| \leq R_{ps}$, set the predators in motion.
- 4: **Obstacle avoidance:** if $\|w_{l,k}^p - w_{j,k}^p\| \leq r_{pp}$, then

$$\delta^p(k) = (r_{pp} - \|w_{l,k}^p - w_{j,k}^p\|) u(w_{l,k}^p - w_{j,k}^p) \quad (25)$$

- 5: **Collision avoidance:** if $\|w_{l,k}^p - T(k)\| < r_{pp}$, then

$$\delta_k^T = (r_{ft} - \|w_{l,k}^p - T_k\|) u(w_{l,k}^p - T_k) \quad (26)$$

- 6: **Attack mode:** Let

$$\|x_{j,k} - w^p l, k\| = \min_{i=1, \dots, \mathcal{N}} \|x_{i,k} - w^p l, k\| \quad (27)$$

$$v_{hk}^p = u(x_{j,k} - w^p l, k) \quad (28)$$

- 7: **Setting up the desired predator speed:**

$$v_{l,k}^p = \gamma_3 v_{hk}^p + \gamma_4 \delta_k^p + \gamma_5 \delta_k^T \quad (29)$$

- 8: **Idle mode:** if $\|w_{l,k}^p - x_j, k\| > R_{ps}$, then $v_{l,k}^p = 0$

UAV, its desired speed can be determined using algorithm (4):

Algorithm 4 Target Behavior

- 1: **Data required:** Target position, individual UAV statuses (active or inactive), active UAV positions.
- 2: **Initialization:** All velocities are set to zero.
- 3: **Start:** Let $\|x_{j,k} - w_k^f\| = \min_{i=1, \dots, \mathcal{N}} \|x_{i,k} - w_k^f\|$; set the target in motion if $\|x_{j,k} - w_k^f\| < r_{ft}$

$$v_k^f = u(w_k^f - x_{j,k})(r_{ft} - \|w_k^f - x_{j,k}\|) \quad (30)$$

- 4: **Idle mode:** if $\|x_{j,k} - w_k^f\| > r_{ft}$, then $v_k^T = 0$

To find the desired position and angular orientation of the target, the following relations are used:

$$\begin{aligned} w_{d_{k+1}}^f &= w_{d_k}^f + \Delta t \cdot V_{d_k}^f \\ \theta_{d_{k+1}} &= a \tan(V_{d_k}^f(2), V_{d_k}^f(1)) \\ w_{d_{k+1}}^f &= \frac{\theta_{d_{k+1}}^f - \theta_{d_k}^f}{\Delta t} \end{aligned} \quad (31)$$

VI. TRACKING CONTROL OF UAVS

In this section, we will integrate all the concepts proposed in the previous sections by developing control techniques that ensure the desired trajectories planned by the navigation algorithms are followed as expected. We discuss the tracking problem for a single UAV and then explore implementing the

control algorithm on a fleet of UAVs. We also shed more light on the trajectory generation process and trajectory switching conditions. The control schematic for a single UAV closed-loop system is shown in Figure (5).

A. TRAJECTORY GENERATION

Based on the information in Table 2, the trajectory generation algorithms are combined in a concise manner as follows. Let $[x_d \ y_d]$ be the desired states from the trajectory planning, θ_d – the desired steering angle for θ_s , \tilde{v}_d – be the desired UAV linear velocity in the presence of attacks obtained from the evasive control strategy, V_d – be the desired UAV linear velocity in the absence of attacks, and w_d – be the desired UAV angular velocity in either the presence or absence of attacks. A more elaborate definition will be given for the desired velocities in the next subsections based on the presence or absence of predators.

1) ABSENCE OF PREDATORS

When there are no predators within the prescribed safe area, each UAV mimics the behaviors dictated by the bioinspired predator-prey algorithm. The velocity obtained is used to obtain the next desired position of the planned trajectory. It is important to note that this trajectory should also satisfy the nonholonomic constraints of the UAV. Thus, the desired trajectory in the absence of predators is given by:

$$\begin{aligned} x_{d_{k+1}} &= x_{d_k} + \Delta t \cdot \|V_d\| \cos \theta_{d_k} \\ y_{d_{k+1}} &= y_{d_k} + \Delta t \cdot \|V_d\| \sin \theta_{d_k} \\ \theta_{d_{k+1}} &= a \tan(V_{d_{k+1}}(2), V_{d_{k+1}}(1)) \\ w_{d_{k+1}} &= \frac{\theta_{d_{k+1}} - \theta_{d_k}}{\Delta t} \end{aligned} \quad (32)$$

2) PRESENCE OF PREDATORS

In this case, the UAV will track a desired straight line, which was described in section (V-A2), based on its state when it first senses the presence of the predator. The desired trajectories are given in equation (19).

B. SINGLE UAV TRACKING ANALYSIS

Before considering a fleet of UAVs, we first develop a control technique that ensures the tracking of the desired path. A tracking controller is thus developed for the nonholonomic UAV model. Indeed, during evasion, the UAV is trying to escape and therefore is not required to synchronize its movement with that of the other members of the fleet. However, during foraging, the fleet must observe social behavior. In the next two sections, we will describe the controller design for the case of a single UAV and then treat the design for the case of the fleet as a flocking problem.

1) ROBUST FEEDBACK CONTROLLER DURING EVASION MODE

A nonlinear controller based on the backstepping method is designed to achieve asymptotic tracking of the UAV to a

desired trajectory. Let $\tilde{\theta} = \theta - \theta_d$, $X = \begin{bmatrix} x \\ y \end{bmatrix}$, and $\tilde{X} = X - X_d = \begin{bmatrix} x - x_d \\ y_d \end{bmatrix}$.

Theorem 1: Let v_d be given by equation (18) and $A_c \in \mathcal{R}^2$ be a Hurwitz matrix designed according to the dynamics of the desired error. For a single UAV under unknown but bounded disturbances given by equation (38), the following control law ensures that the system is uniformly asymptotically stable.

$$u_1 = \|v\| \cos \tilde{\theta} - \eta \text{sign}(\tilde{X}) \quad (33)$$

$$\theta_d = \text{atan}(v_d(2), v_d(1)) \quad (34)$$

$$u_2 = -k_1 \tilde{\theta} + \dot{\theta}_d \quad (35)$$

$$v = G(\tilde{\theta})^{-1}(v_d + A_c \tilde{X} - \eta \text{sign}(\tilde{X})) \quad (36)$$

$$G(\tilde{\theta}) = \begin{bmatrix} \cos^2 \tilde{\theta} & -\frac{1}{2} \sin 2\tilde{\theta} \\ \frac{1}{2} \sin 2\tilde{\theta} & \cos^2 \tilde{\theta} \end{bmatrix} \quad (37)$$

where the design parameters k_1 and η are positive scalars.

Proof: We will follow in this proof the same procedure as in [45]. Let the system in (1) be written as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} u_1 \cos \theta_s \\ u_1 \sin \theta_s \end{bmatrix} + \Delta \quad (38)$$

$$\dot{\theta}_s = u_2 \quad (39)$$

Let $u_1 = \|v\| \cos \tilde{\theta}$ and $v_d = \begin{bmatrix} v_{dx} \\ v_{dy} \end{bmatrix}$ be the desired velocity set by the guidance and navigation algorithm. v_x and v_y are the corresponding velocities along the x - and y -axes, respectively. This leads to

$$\theta_d = \text{atan}2d(v_y, v_x) \quad (40)$$

in the case where $v_x = 0, \theta_d = 90^\circ$ and in the case where $v_y = 0, \theta_d = 0$. Therefore, $v_x = \|v\| \cos \theta_d$ and $v_y = \|v\| \sin \theta_d$. This allows us to write (38) as

$$\dot{X} = \begin{bmatrix} \|v\| \cos \tilde{\theta} \cos \theta_d \\ \|v\| \cos \theta_d \sin \theta_d \end{bmatrix} \quad (41)$$

Using trigonometric properties and recognizing that $\theta = \tilde{\theta} + \theta_d$, equation 41 leads to

$$\dot{X} = G(\tilde{\theta}) v \quad (42)$$

with

$$\dot{X} = G(\tilde{\theta}) v \quad (43)$$

where

$$G(\tilde{\theta}) = \begin{bmatrix} \cos^2 \tilde{\theta} & -\frac{1}{2} \sin 2\tilde{\theta} \\ \frac{1}{2} \sin 2\tilde{\theta} & \cos^2 \tilde{\theta} \end{bmatrix} \quad (44)$$

$G(\tilde{\theta})$ is nonsingular if $\tilde{\theta} \neq 90^\circ$. In navigation, 90° is always avoided and practically impossible to achieve. We will therefore assume that the desired angle that requests a 90° difference from the actual angle will not be considered. Hence,

the following direct feedback linearizing controller can be proposed:

$$v = G(\tilde{\theta})^{-1}(v_d + A_c \tilde{X}) \quad (45)$$

This allows us to write the dynamic of the error as

$$\dot{\tilde{X}} = A_c \tilde{X} \quad (46)$$

where A_c is a Hurwitz-stable matrix with the desired eigenvalues. Because the system dynamics are subject to an unknown but bounded external disturbance Δ , a robustifying term is required. Using a Lyapunov candidate functional $W = 1/2 \tilde{X}^T \tilde{X}$, the control law in equation (47) can be made robust as follows:

$$v = G(\tilde{\theta})^{-1}(v_d + A_c \tilde{X} - \eta \text{sign}(\tilde{X})) \quad (47)$$

where $\eta > \bar{\Delta}$ and $\bar{\Delta} \max(\Delta)$. This leads to

$$\dot{\tilde{X}} = A_c \tilde{X} - \eta \text{sign}(\tilde{X}) \quad (48)$$

□

2) TRACKING THE FORAGING TRAJECTORY

In this subsection, we will consider the case where the fleet is not under attack. This is the case for multiple nonholonomic UAVs schooling toward a target; it represents a typical flocking or schooling problem except that in this case, because the UAV operation mode changes to foraging after an attack, the original network could be fragmented and composed of several strongly connected groups each with its own network. The clustering technique presented in section IV identifies each subnetwork.

Remark 2: In the case where the network is composed of a cluster containing one UAV, the control law given in theorem 1 will be used.

Let C_k be the number of clusters as identified by DBSCAN. The algorithm also identifies the corresponding UAVs that form a given cluster C_i .

Theorem 2: Let C_i be a cluster of strongly connected networks formed by N_{C_i} UAVs under unknown but bounded disturbances. Let the desired velocity $v_{d,i,k}$ be given by equation (18) and $A_{c_i} \in \mathcal{R}^2$ be a Hurwitz matrix designed according to the dynamics of the desired error. For the i^{th} UAV, the following control law ensures that the system is uniformly asymptotically stable:

$$u_{i1} = \|v_{i,k}\| \cos \tilde{\theta}_{i,k} - \eta_i \text{sign}(X_{i,k}) \quad (49)$$

$$\theta_{d,i,k} = \text{atan}(v_{d,i,k}(2), v_{d,i,k}(1)) \quad (50)$$

$$u_{i2} = -k_{i2} \tilde{\theta}_{i,k} + \dot{\theta}_{d,i,k} \quad (51)$$

$$v_{i,k} = G_i(\tilde{\theta})^{-1}(v_{d,i,k} + A_{c_i} X_{i,k} - \eta_i \text{sign}(X_{i,k})) \quad (52)$$

$$G_i(\tilde{\theta}_i) = \begin{bmatrix} \cos^2 \tilde{\theta}_i & -\frac{1}{2} \sin 2\tilde{\theta}_i \\ \frac{1}{2} \sin 2\tilde{\theta}_i & \cos^2 \tilde{\theta}_i \end{bmatrix} \quad (53)$$

where the design parameters k_{i1} and η_i are positive scalars.

Proof: The proof of this theorem is composed of three parts: the design of the controller and the stability of the multiagent system, the guarantee that the error in positions is according to the required safety distance r , and the consensus on velocity to guarantee coherent motion. For part 1, Theorem (1) can be easily used to show that the system is uniformly asymptotically stable. For part 2, since each cluster C_i is a strongly connected system, the Adapt-Then-Combine diffusion algorithm equations (15)-(18) ensure that each node will respect the distance r given in equation 16. Based on equations (15)-(18), the coherent motion will be such that $v_{i,k}$ will converge to $v_{i,k}^g$ (equation 17), the local speed of the center of gravity of network C_i . \square

Remark 3: • Theorem 2 can also be proven using equations (15)-(18) and graph theory for strongly connected systems, as in [45].

- During path planning for a nonholonomic UAV, the objective of tracking certain trajectories can also be achieved using non-smooth paths. Unfortunately, such non-smooth paths usually occur as a result of a large velocity input, which can be too large and impossible for the nonholonomic model to achieve in real applications due to the nonholonomic constraints. Thus, to attain smooth tracking of the reference paths, a maximum value is used to limit the velocities.
- Using Algorithm (3) and equation (24), Theorem 1 can be used to find the necessary local control law for predators during hunting.
- Using Algorithm (4) and equation (31), Theorem 1 can be used to find the necessary local control law for the target during evasion.

VII. SIMULATION EXAMPLE

We consider several examples to highlight some of the important features in this approach. The simulation requires inputting the initial positions of the UAV fleet, predators and target. It also requires inputting different radii to set up the dynamic motion of the scenario at hand. Figure 10 shows the fleet in motion toward the target. The initial distribution of the group of $\mathcal{N} = 25$ UAVs was such that the following requirements had to be met: the distance between neighboring agents was set according to equation 16) with $r = 4$ and $\epsilon = 0.5$, the initial target position was $w_{k,1}^f = (60, 60)$ and the predators initial positions were set to $w_{k,1}^p = (60, 65)$ for predator 1 and at $(60, 55)$ for predator 2. In all the coming simulations, the simulation time = 200 sec; the sampling time = 0.1 sec. The zone around the target that the predators have to avoid is set to a circle centered at the target with radius $r_{ft} = 12$. Figure 10 illustrates the obstacle avoidance capabilities of the algorithm and how the foraging motion toward the target is harmonious.

Figures 11 and 12 illustrate the model performed with $\mathcal{N} = 9$ and $\mathcal{N} = 4$, respectively. The target is placed at $w_{k,1}^f = (50, 50)$. Figure 11 shows the group initially placed at positions less than the required minimum separation $r = 4$.

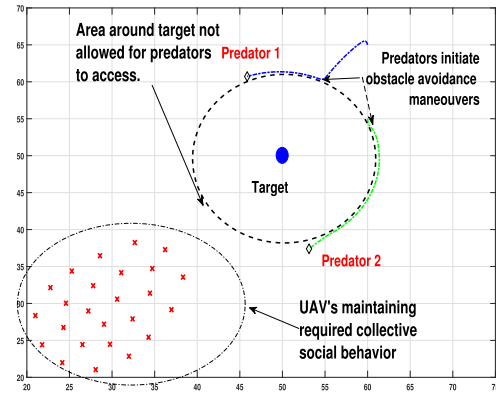


FIGURE 10. Several UAVs in formation and obstacle avoidance.

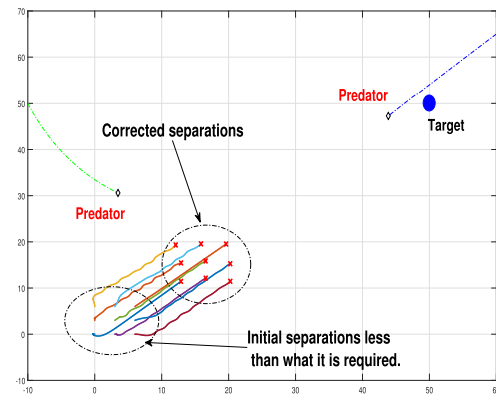


FIGURE 11. Several UAVs starting with initial separation less than what is required as a single cluster.

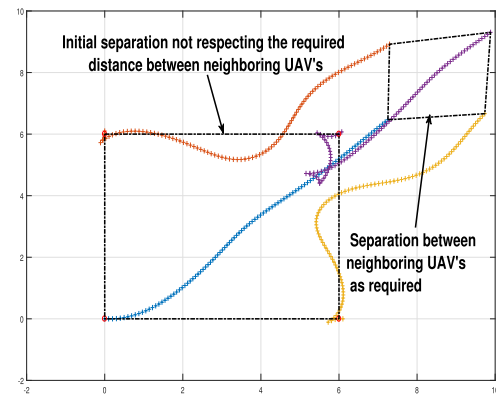


FIGURE 12. Several UAVs starting with initial separation greater than what is required as a single cluster.

The fleet starts by adjusting their distances and repositioning themselves as required. In Figure 12, the initial positions of the group are such that the separation between them is greater than what is required, $r = 3$, with $\epsilon = 0.5$. The figure shows how the individual UAVs correct their positions as required. The time to adjust the positions is not instantaneous, as we are dealing with nonholonomic systems and not points of mass. In both tests, the fleet starts as a single network. In Figure 13, we repeat the same test with $\mathcal{N} = 25$ and $r = 3$ with

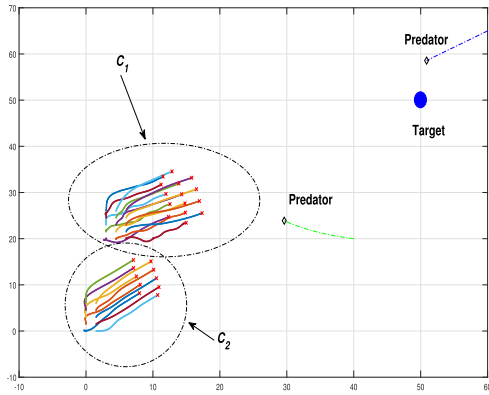


FIGURE 13. Several UAVs starting in two different clusters C_1 and C_2 with initial positions less than what is required.

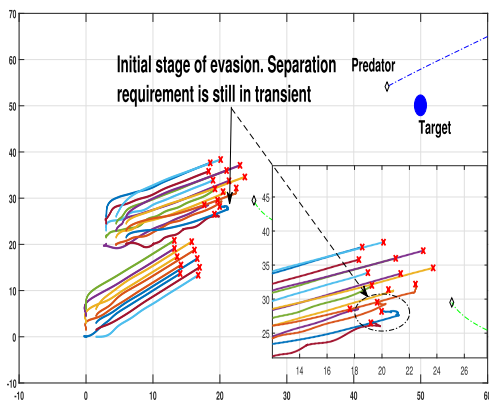


FIGURE 14. Transient of the UAV fleet under attack by predators.

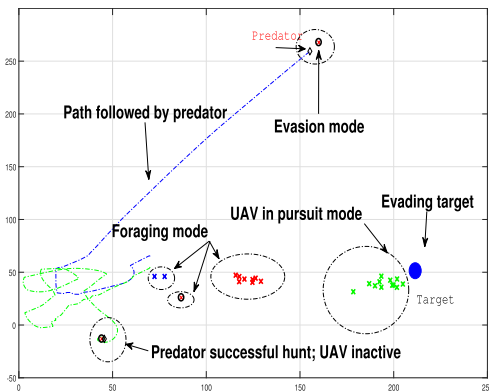


FIGURE 15. Dispersion of the initial population of UAVs to different clusters or social groups.

$\epsilon = 0.5$, but the system initially starts as two different groups. The two clusters start by correcting the positions of the different UAVs to satisfy the distance requirements and then move harmoniously in the foraging phase toward the target. Figure shows the transient UAVs evading the predator. One can see that the distance between two UAVs becomes smaller than what it is required. However, two other UAVs start moving in a direction that accommodates the distance requirement. During attacks, the network can be divided into

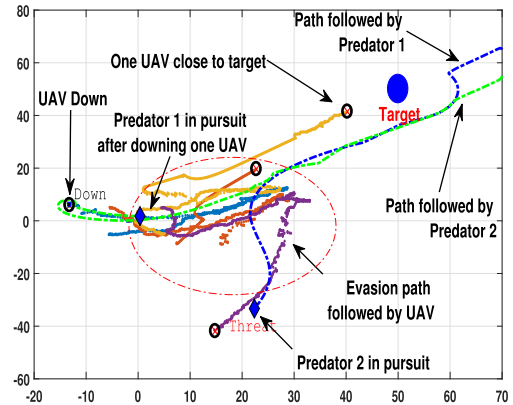


FIGURE 16. Different paths showing evasion and obstacle avoidance.

TABLE 3. Execution time as a function of the UAV fleet size.

	$\mathcal{N} = 4$	$\mathcal{N} = 9$	$\mathcal{N} = 16$	$\mathcal{N} = 25$
Execution time (sec)	9.48	16.64	23.62	28.03
	$\mathcal{N} = 25$	$\mathcal{N} = 49$	$\mathcal{N} = 64$	$\mathcal{N} = 225$
Execution time (sec)	39.11	50.19	64.24	144.95

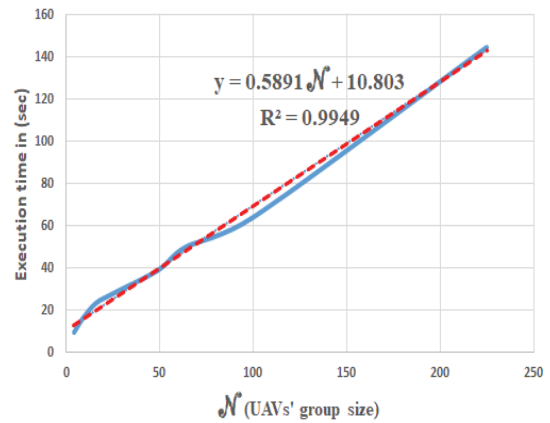


FIGURE 17. Execution time of the algorithm based on the UAV fleet size.

several groups. In Figure 15 and 16, an attack by the predator is already underway. One can see the network divided into several groups, some with only one UAV. The different groups are in different modes. Some UAVs have been inactivated by predators, some are still in evasion mode, others are back to foraging and observing social behavior, and one group is already in pursuit of the target, having initiated its own evasion strategy. The tail of the UAVs in 16 shows the evasion paths the UAV took to evade the predators. The two paths of the predators are also clear and show how they avoided the target by going around it.

Remark 4: One can also see from 16 the presence of an outlier, defined as a UAV performing an evasion that keeps going away from the field and target. It seems natural that the

best choice for the predator is to go back and abandon the chase. In the literature, there are several proposed methods for addressing this phenomenon (see, for example, [31]). However, we did not consider this in our work.

COMPLEXITY OF THE ALGORITHM

The complexity of the algorithm is measured by looking at the execution time as a function of the size of the UAV fleet. Table gives an estimate of the execution time required to complete a simulation that runs in 200 sec. Using EXCEL fitting, we can see from Figure 17 that the complexity increases linearly as a function of the size of the UAV population.

VIII. CONCLUSION

In this work, we used a biological model of a school of fish and of predators such as sharks to control a fleet of nonholonomic UAVs in an environment where they can come under attack by enemy UAVs. The problem we considered was designed in the contexts of a flocking problem, an obstacle avoidance problem, a tracking problem, and a collision avoidance problem. During a predator attack, DBSCAN is used to cluster the divided network. Each cluster can be in a different operational mode. The path tracking controller was based on a feedback linearization approach, and the same type of controller was implemented to control both the predators and the UAVs. The motion of the UAV fleet follows social behavior requirements during foraging, but because each system takes time to adjust, the distance requirement could not be satisfied during transient or evasion modes, where the behavior tends to be individualistic.

For future work, there is a need to quantify the transient period during which distancing is not satisfied. The adopted biological model assumes a simple attack strategy based on nearest prey. There is no priority given in selecting the prey, and all are considered equally important. A better approach could be to assume a game between the predator and the prey with high score values given to those prey that are closer to the target. The motion of the predator-prey system considered in this work is modeled in 2-D. A 3-D model could be used, and therefore, more realistic group behavior can be studied. In addition, better and complex evasion strategies can be designed. In addition, the system developed does not learn. Neither the predators nor the prey can learn each other's moves and plan their actions accordingly. This system could be a good base for the application of machine learning and the use of AI in path planning and evasion countermeasures.

REFERENCES

- [1] R. W. Brockett, "Asymptotic stability and feedback stabilization," in *Differential Geometric Control Theory*, vol. 27, no. 1, R. W. Brockett, R. Millmann, and H. Sussmann, Eds. 1983, pp. 181–191.
- [2] M. Sampei, H. Kiyota, H. Koga, and M. Suzuki, "Necessary and sufficient conditions for transformation of nonholonomic system into time-state control form," in *Proc. 35th IEEE Conf. Decis. Control*, vol. 4, Dec. 1996, pp. 4745–4746.
- [3] E. Valtolina and A. Astolfi, "Local robust regulation of chained systems," *Syst. Control Lett.*, vol. 49, no. 3, pp. 231–238, Jul. 2003.
- [4] A. Bloch and S. Drakunov, "Tracking in nonholonomic dynamic systems via sliding modes," in *Proc. 34th IEEE Conf. Decis. Control*, vol. 3, Singapore, Dec. 1995, pp. 2103–2106.
- [5] Y.-P. Tian and S. Li, "Exponential stabilization of nonholonomic dynamic systems by smooth time-varying control," *Automatica*, vol. 38, no. 7, pp. 1139–1146, Jul. 2002.
- [6] J. Keighobadi and M. B. Menhaj, "From nonlinear to fuzzy approaches in trajectory tracking control of wheeled mobile robots," *Asian J. Control*, vol. 14, no. 4, pp. 960–973, Jul. 2012. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/asjc.480/abstract>
- [7] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, vol. 96, no. 34, pp. 226–231, 1996.
- [8] Z. Qiang, L. Zengbo, and C. Yao, "A back-stepping based trajectory tracking controller for a non-chained nonholonomic spherical robot," *Chin. J. Aeronaut.*, vol. 21, no. 5, pp. 472–480, Oct. 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1000936108600618>
- [9] Y.-P. Tian and K.-C. Cao, "An LMI design of tracking controllers for nonholonomic chained-form system," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2007, pp. 4512–4517.
- [10] K.-C. Cao, "Global-exponential trackers for nonholonomic chained-form systems based on LMI," *Int. J. Syst. Sci.*, vol. 42, no. 12, pp. 1981–1992, Dec. 2011.
- [11] Y. Kanayama and F. Fahroo, "A new line tracking method for non-holonomic vehicles," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 4, Apr. 1997, pp. 2908–2913.
- [12] W. MacKunis, N. Gans, A. Parikh, and W. E. Dixon, "Unified tracking and regulation visual servo control for wheeled mobile robots," *Asian J. Control*, vol. 16, no. 3, pp. 669–678, May 2014. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/asjc.826/abstract>
- [13] M. Ou, S. Li, and C. Wang, "Finite-time tracking control for non-holonomic mobile robots based on visual servoing," *Asian J. Control*, vol. 16, no. 3, pp. 679–691, May 2014. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/asjc.773/abstract>
- [14] A. Belkadi, L. Ciarletta, and D. Theilliol, "Particle swarm optimization method for the control of a fleet of unmanned aerial vehicles," *J. Phys., Conf. Ser.*, vol. 659, Nov. 2015, Art. no. 012015.
- [15] A. Belkadi, H. Abaunza, L. Ciarletta, P. Castillo, and D. Theilliol, "Design and implementation of distributed path planning algorithm for a fleet of UAVs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 2647–2657, Dec. 2019.
- [16] R. W. Whittlesey, S. Liska, and J. O. Dabiri, "Fish schooling as a basis for vertical axis wind turbine farm design," *Bioinspiration Biomimetics*, vol. 5, no. 3, Sep. 2010, Art. no. 035005. [Online]. Available: <http://iopscience.iop.org/1748-3190/5/3/035005>
- [17] S. Camazine, J. L. Deneubourg, N. R. Franks, J. Sneyd, E. Bonabeau, and G. Theraula, *Self-Organization in Biological Systems*, vol. 7. Princeton, NJ, USA: Princeton Univ. Press, 2003.
- [18] F. Cattive and A. H. Sayed, "Self-organization in bird flight formations using diffusion adaptation," in *Proc. 3rd IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, Amsterdam, The Netherlands, Dec. 2009, pp. 49–52.
- [19] S.-Y. Tu and A. H. Sayed, "Foraging behavior of fish schools via diffusion adaptation," in *Proc. 2nd Int. Workshop Cognit. Inf. Process.*, Elba, Italy, Jun. 2010, pp. 63–68.
- [20] S.-Y. Tu and A. H. Sayed, "Mobile adaptive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 649–664, Aug. 2011.
- [21] S.-Y. Tu and A. H. Sayed, "Tracking behavior of mobile adaptive networks," in *Proc. Conf. Rec. 44th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, Nov. 2010, pp. 698–702.
- [22] J. Li, S.-Y. Tu, and A. H. Sayed, "Honeybee swarming behavior using diffusion adaptation," in *Proc. Digit. Signal Process. Signal Process. Edu. Meeting (DSP/SPE)*, Sedona, AZ, USA, Jan. 2011, pp. 249–254.
- [23] Y. H. Chang, C. Tomlin, and K. Hedrick, "Biologically-inspired coordination of multiple UAVs using sliding mode control," in *Proc. Amer. Control Conf.*, Jun. 2011, pp. 4123–4128.
- [24] M. A. Haque, A. R. Rahmani, and M. B. Egerstedt, "Biologically inspired confinement of multi-robot systems," *Int. J. Bio-Inspired Comput.*, vol. 3, no. 4, pp. 213–224, 2011.
- [25] Q. Li and Z. P. Jiang, "Flocking of decentralized multi-agent systems with application to nonholonomic multi-robots," in *Proc. 17th World Congr. Int. Fed. Autom. Control*, 2008, pp. 9344–9349.

- [26] Q. Li and Z.-P. Jiang, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *Kybernetika*, vol. 45, no. 1, pp. 84–100, 2009.
- [27] S. I. Nishimura and T. Ikegami, "Emergence of collective strategies in a prey-predator game model," *Artif. Life*, vol. 3, no. 4, pp. 243–260, Oct. 1997.
- [28] I. S. Nishimura, "A predator's selection of an individual prey from a group," *Biosystems*, vol. 65, no. 1, pp. 25–35, 2002.
- [29] J. Demšar and I. L. Bajec, "Simulated predator attacks on flocks: A comparison of tactics," *Artif. Life*, vol. 20, no. 3, pp. 343–359, Jul. 2014.
- [30] R. S. Olson, D. B. Knoester, and C. Adami, "Evolution of swarming behavior is shaped by how predators attack," *Artif. Life*, vol. 22, no. 3, pp. 299–318, Aug. 2016.
- [31] J. Demšar, C. K. Hemelrijk, H. Hildenbrandt, and I. L. Bajec, "Simulating predator attacks on schools: Evolving composite tactics," *Ecological Model.*, vol. 304, pp. 22–33, May 2015.
- [32] A. H. Sayed, "Adaptive networks," *Proc. IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.
- [33] F. S. Cattivelli and A. H. Sayed, "Modeling bird flight formations using diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 59, no. 5, pp. 2038–2051, May 2011.
- [34] A. H. Sayed, S.-Y. Tu, J. Chen, X. Zhao, and Z. J. Towfic, "Diffusion strategies for adaptation and learning over networks: An examination of distributed strategies and network behavior," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 155–171, May 2013.
- [35] C. Lopes and A. Sayed, "Diffusion least-mean squares over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [36] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Las Vegas, NV, USA, Mar. 2008, pp. 3285–3288.
- [37] Y. Jiang, "Concurrent collective strategy diffusion of multiagents: The spatial model and case study," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 39, no. 4, pp. 448–458, Jul. 2009.
- [38] N. Takahashi, I. Yamada, and A. H. Sayed, "Diffusion least-mean squares with adaptive combiners: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 58, no. 9, pp. 4795–4810, Sep. 2010.
- [39] F. S. Cattivelli and A. H. Sayed, "Diffusion LMS strategies for distributed estimation," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1035–1048, Mar. 2010.
- [40] J. Chen, C. Richard, and A. Sayed, "Diffusion LMS for clustered multitask networks," 2013, *arXiv:1310.8615*. [Online]. Available: <http://arxiv.org/abs/1310.8615>
- [41] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.
- [42] M. R. Gholami, E. G. Ström, and A. H. Sayed, "Diffusion estimation over cooperative networks with missing data," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 411–414.
- [43] S. El Ferik and R. O. Thompson, "Biologically inspired control of a fleet of Uavs with threat evasion strategy," *Asian J. Control*, vol. 18, no. 6, pp. 2283–2300, 2016, doi: [10.1002/asjc.1324](https://doi.org/10.1002/asjc.1324).
- [44] F. Yan, J. Jiang, K. Di, Y. Jiang, and Z. Hao, "Multiagent pursuit-evasion problem with the pursuers moving at uncertain speeds," *J. Intell. Robot. Syst.*, vol. 95, no. 1, pp. 119–135, Jul. 2019.
- [45] M. Khaledyan, T. Liu, V. Fernandez-Kim, and M. de Queiroz, "Flocking and target interception control for formations of nonholonomic kinematic agents," *IEEE Trans. Control Syst. Technol.*, early access, May 2019, doi: [10.1109/TCST.2019.2914994](https://doi.org/10.1109/TCST.2019.2914994).
- [46] S.-Y. Tu and A. H. Sayed, "Cooperative prey herding based on diffusion adaptation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Italy, May 2011, pp. 3752–3755.
- [47] S.-Y. Tu and A. Sayed, "Mobile adaptive networks with self-organization abilities," *IEEE J. Sel. Topics Signal Process.*, 2010, pp. 379–383.
- [48] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Coordination and control of multiple UAVs," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2002, Paper AIAA 2002–4588.
- [49] O. Albayrak, "Line and circle formation of distributed autonomous mobile robots with limited sensor range," Ph.D. dissertation, 1996.
- [50] J. Sprinkle, J. M. Eklund, H. J. Kim, and S. Sastry, "Encoding aerial pursuit/evasion games with fixed wing aircraft into a nonlinear model predictive tracking controller," in *Proc. 43rd IEEE Conf. Decis. Control (CDC)*, vol. 3, Dec. 2004, pp. 2609–2614.
- [51] J. M. Eklund, J. Sprinkle, and S. Sastry, "Implementing and testing a nonlinear model predictive tracking controller for aerial pursuit/evasion games on a fixed wing aircraft," in *Proc. Amer. Control Conf.*, Jun. 2005, pp. 1509–1514.
- [52] J. M. Eklund, J. Sprinkle, and S. S. Sastry, "Switched and symmetric Pursuit/Evasion games using online model predictive control with application to autonomous aircraft," *IEEE Trans. Control Syst. Technol.*, vol. 20, no. 3, pp. 604–620, May 2012.
- [53] H. Yuan, "Control of nonholonomic systems," Ph.D. dissertation, Univ. Central Florida, Orlando, FL, USA, 2009.
- [54] S. LaValle, *Planning Algorithm*, 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2006.
- [55] H. Jiawei, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2011.
- [56] M. Hahsler, M. Piekenbrock, and D. Doran, "dbscan: Fast density-based clustering with R," *J. Stat. Softw.*, vol. 25, pp. 409–416, 2019.
- [57] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 2, pp. 19–40, 2017, doi: [10.1145/3068335](https://doi.org/10.1145/3068335).
- [58] A. Anant and A. Solanki, "An improved data clustering algorithm for outlier detection," *Int. Acad. Ecol. Environ. Sci.*, vol. 3, no. 4, pp. 121–139, 2016.
- [59] E. Martin, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, 1996, vol. 96, no. 34, pp. 226–231.



SAMI EL-FERIK received the B.Sc. degree in electrical engineering from Laval University, Quebec City, Canada, and the M.S. and Ph.D. degrees in electrical and computer engineering from the École Polytechnique, University of Montreal, Montreal, QC, Canada. His Ph.D. work on flexible manufacturing system modeling and control was co-supervised by the Department of Mechanical Engineering.

After completion of his Ph.D. degree and post-doctoral positions, he worked with Pratt and Whitney Canada as a Staff Control Analyst with the Research and Development Center of Systems, Controls, and Accessories. He is currently an Associate Professor in control and instrumentation with the Department of Systems Engineering, King Fahd University of Petroleum and Minerals (KFUPM). His research interest includes sensing, monitoring, and control with strong multidisciplinary research and applications. His research contributions include the control of drug administration, process control and control loop performance monitoring, the control of systems with delays, the modeling and control of stochastic systems, the analysis of network stability, condition monitoring, and condition-based maintenance.

...