Detection of Dynamically Changing Leaders in Complex Swarms from Observed Dynamic Data

Christos N. Mavridis(✉), Nilesh Suriyarachchi, and John S. Baras

Department of Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742, USA
{mavridis,nileshs,baras}@umd.edu

Abstract. In this work we consider the problem of defending against adversarial attacks from UAV swarms performing complex maneuvers, driven by multiple, dynamically changing, leaders. We rely on short-time observations of the trajectories of the UAVs and develop a leader detection scheme based on the notion of Granger causality. We proceed with the estimation of the swarm's coordination laws, modeled by a generalized Cucker-Smale model with non-local repulsive potential functions and dynamically changing leaders, through an appropriately defined iterative optimization algorithm. Similar problems exist in communication and computer networks, as well as social networks over the Internet. Thus, the methodology and algorithms proposed can be applied to many types of network swarms including detection of influential malevolent "sources" of attacks and "miss-information". The proposed algorithms are robust to missing data and noise. We validate our methodology using simulation data of complex swarm movements.

Keywords: Leader detection · Anti-UAV defense · Identification of swarm coordination laws

1 Introduction

Air defense systems have been forced to constantly adapt and evolve over time to combat various new types of aerial threats. Today's air defense systems are highly capable of taking out single targets with ever increasing levels of precision. However, the advent and proliferation of the use of Unmanned Aerial Vehicles (UAV's) now poses new challenges to air defense systems. With the increase in computing capabilities in low cost hardware components, it has become feasible for adversarial forces to employ UAV swarms to be used for activities ranging from surveillance to deadly payload delivery and targeted attacks. While modern

This work was partially supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111990027, by ONR grant N00014-17-1-2622, and by a grant from Northrop Grumman Corporation.

Q. Zhu et al. (Eds.): GameSec 2020, LNCS 12513, pp. 223–240, 2020.
https://doi.org/10.1007/978-3-030-64793-3_12

high-precision targeting anti-air defenses are capable of taking down a single UAV, when it comes to a large swarm of UAV's attacking simultaneously, these defences can be rendered ineffective. These problems are even more challenging when in UAV swarms, a few units are managed by humans (we refer to them as "leaders", while most units "follow", leading to very effective management of large swarms. When the role of leaders can be dynamically re-assigned the monitoring and defense against such swarms becomes even more difficult.

The first question that needs to be addressed in creating a defense against a hostile UAV swarm is understanding the control (coordination) and communication laws governing how the drones move and interact with each other. In large swarms it is unlikely that all the interacting drones have independent control and motion planning algorithms (of the kind found in the single robot planning literature [24]). Instead, flocking models have been proposed to study animal flocks and artificial swarm dynamics [1,2,8,9,20,25,27]. The investigation of these biological swarms have provided inspiration and useful modeling abstractions for addressing these challenging problems.

However, when studying complex swarm maneuvers, autonomous models such as the Cucker-Smale model [8,11] or the Boids model [27] cannot capture the behavior of the swarm, and leadership is often incorporated in the flocking model [31]. Having understood the flocking nature of the swarm, one key idea for creating a defense strategy in order to combat the swarm involves accurately identifying the leaders and the underlying dynamics of particle interactions in the swarm. The first step requires the clear identification of the leaders in the hostile swarm. If the leaders can be identified in real time then modern air defense systems such as high precision laser weapons which are aimed at combating UAV's can be used to take out these leaders, thus disrupting the operation of the entire swarm. Recent work into leader detection has looked into the use of Markov Chain Monte Carlo based group tracking methods [6]. However, in order to handle real time leader detection in high particle count swarms this paper proposes a Granger causality based detection method.

In this paper we will use the terms agents and particles interchangeably. Extracting the laws of interaction (or coordination) between agents is the next requirement for creating a defense strategy against large hostile swarms. Understanding the governing dynamics of the hostile swarm will enable the defense system to plan ahead and anticipate how the swarm would react to different strategies such as the focused removal of the agents identified as leaders. Multiple methods exist in order to identify the underlying interactions and dynamics of particle swarms. Statistical [5,16], and, mainly, model-based [8,20,27] learning approaches have been used to infer interaction rules between particles. In [4] symbolic equations are generated from the numerically calculated derivatives of the system variables, in [19] the constitutive equations of physical components composing the system are learned, while in [18] the order of a fractional differential system of equations, which models the system, is estimated. Recently, Matei et al. in [20], and Mavridis et al. in [22] have modeled the networked swarm as a port-Hamiltonian system [29] and have accurately reconstructed the laws of interaction (or coordination) of the swarm and its dynamical properties, from

observed trajectories of the individual agents. Furthermore in these recent works [20,22,23] we have also demonstrated the robustness of the associated algorithms to both noisy observations as well as missing data.

Similar problems are found in many other types of large networked systems, including communication and computer networks, sensor networks, networked cyber-physical systems, biological systems, and social networks over the Internet. In such systems there are corresponding notions of leaders, such as initiators of a malicious attack, or coordinators of malevolent behavior, or initiators of a biological cell-malfunction, or influential sources of miss-information or untrustworthiness [30]. *In all these problems fast identification of the leaders and the associated followers groups (or influence groups) is essential for defending and correcting such malevolent actions and functions. Thus the applicability of the ideas and methods proposed in this work is very broad, with the appropriate modeling and semantic changes for the various domains.*

In this work, we focus on observations of complex swarm maneuvers driven by multiple dynamically changing leaders, and propose a leader detection scheme, based on the notion of Granger causality, that allows for the online estimation of the particle interaction laws through an appropriately defined iterative optimization algorithm. In the learning process, we assume a generalized Cucker-Smale model with non-local repulsive potential functions and dynamically changing leaders [31]. We validate our methodology using simulation data of complex swarm movements. Similar problems exist in communication and computer networks, as well as social networks over the Internet. Thus the methodology and algorithms proposed can be applied to many types of network swarms including detection of influential malevolent "sources" of attacks and "miss-information".

The rest of the manuscript is organized as follows: Sect. 2 describes the models used to describe the swarm dynamics, and Sects. 3 and 4 introduce the leader detection algorithm. In Sect. 5 the learning algorithm for the swarm's interaction laws is formulated. Finally, Sect. 6 presents the experimental results, and Sect. 7 concludes the paper.

2 Modeling Complex Swarm Maneuvers

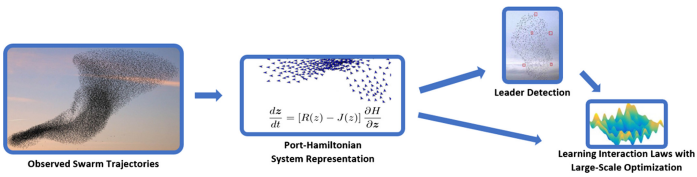

Fig. 1. Reconstructing complex swarm dynamics. The agents' trajectories are observed and used to detect leaders and identify a port-Hamiltonian networked system modeling their interaction rules.

We view the interconnected problems of modeling and learning the interaction laws of a swarm as one problem that can be analyzed in the microscopic scale as a

port-Hamiltonian networked system. We extend existing simulation models, such as the Boids and the Cucker-Smale models, to incorporate interaction, communication and dynamics terms that can capture realistic complex swarm maneuvers and develop corresponding simulation models in the macroscopic domain. Specifically, we introduce

- a scalable simulation algorithm, based on the Boids model, that can capture interaction laws and communication protocols of complex swarm maneuvers, including (a) velocity alignment, (b) spatial cohesion, (c) collision avoidance, and (d) response to dynamically changing leaders.
- a large-scale learning algorithm, based on the generalized Cucker-Smale model and automatic differentiation, designed to work on state-of-the-art deep learning platforms that can identify the interaction laws (a)–(d) by observing particle trajectories of position and velocity (Fig. 1).

2.1 Extended Boids Model

The Boids algorithm is a widely used artificial flocking simulation algorithm based on three basic rules [27].

1. *Cohesion*: Boids are steered in such a way that they move towards the average position (perceived center of mass) of local flockmates. The radius of attraction is a parameter than can be tuned in this section.
2. *Alignment*: Boids are steered towards the average heading and average speed of local flockmates.
3. *Separation*: Boids are steered in such a way that they avoid crowding local flockmates. This acts as a collision avoidance strategy between particles.

The Boids model can be written as a dynamical system:

$$\begin{cases} \dot{x}_i &= v_i \\ \dot{v}_i &= -c\nabla U_c(x) - a\nabla U_a(x,v) + s\nabla U_s(x) \end{cases} \tag{1}$$

where

- $\nabla U_c(x) = x_i - \frac{1}{N_c}\sum_{j\neq i} \mathbb{1}_{[x_i - x_j \leq r_c]} x_j = {}^1\!/_2 \nabla \| x_i - \frac{1}{N_c}\sum_{j\neq i} \mathbb{1}_{[x_i - x_j \leq r_c]} x_j \|^2$, simulates the cohesion rule,
- $\nabla U_a(x,v) = v_i - \frac{1}{N_a}\sum_{j\neq i} \mathbb{1}_{[x_i - x_j \leq r_a]} v_j = {}^1\!/_2 \nabla \| v_i - \frac{1}{N_a}\sum_{j\neq i} \mathbb{1}_{[x_i - x_j \leq r_a]} v_j \|^2$, simulates the velocity alignment rule, and
- $\nabla U_s(x) = \sum_{j\neq i} \mathbb{1}_{[x_i - x_j \leq r_s]}(x_i - x_j)$, simulates the collision avoidance (separation) rule.

In addition to these rules, the interacting agents (boids) may be modeled to have a tendency towards a particular place, by adding an attractive term with respect to a possibly time-dependent potential function

$$-w\nabla U_w(x, x_w) = -w\mathbb{1}_{[x_i - x_w \leq r_w]}(x_i - x_w) = -{}^1\!/_2 w \mathbb{1}_{[x_i - x_w \leq r_w]} \nabla \| x_i - x_w \|^2$$

simulating strong wind or leadership.

Although the Boids model is widely adapted in many simulations due to its simplicity, the fact that the interaction of each particle with its neighbors is local, i.e., the existence of the neighborhood radii r_a, r_s, r_c etc., introduces problems with the differentiability of the cost function of the learning problem. In order to preserve differentiability and be able to utilize existing large-scale optimization frameworks for deep learning to work, we need to replace the indicator function of belonging to a neighborhood with a smooth interaction function ψ that defines the grade of membership of a particle to the neighborhood of another.

2.2 Cucker-Smale Model with Leadership

When focused on the learning algorithm, we model the swarm with the Cucker-Smale model [7,8]. In order to model complex flock maneuvers, we borrow from the theory of flock leadership (see e.g. [31]) and incorporate leadership to the Cucker-Smale model as follows:

Definition 1. *Consider an interacting system of N particles. The leader sets $\mathcal{L}(i),\ 1 \leq i \leq N$ of cardinality $|\mathcal{L}(i)| = 1$ are assigned to each particle representing the index of the leader particle that it is following. Then the Cucker-Smale (CS) model with leadership is defined in the following:*

$$\begin{cases} \dot{x}_i &= v_i \\ \dot{v}_i &= \frac{K}{N} \sum_{j=1}^{N} \psi_{ij}(x(t), v(t)) \end{cases} \tag{2}$$

where

$$\psi_{ij}(x) = \begin{cases} -\nabla U(\|x_i - x_j\|), & j \notin \mathcal{L}(i), j \neq i \\ G(\|x_i - x_j\|)(v_j(t) - v_i(t)) - \nabla U(\|x_i - x_j\|), & j \in \mathcal{L}(i) \end{cases} \tag{3}$$

with a typical choice for the interaction function G that provably results in flocking behavior being $G(r) = \frac{1}{(1+r^2)^\gamma}$ and the potential function usually taking the form $U(r) = -C_A e^{-r/l_A} + C_R e^{-r/l_R}$, with C_A, C_R, l_A, l_R positive scalars.

It has been shown in [20] that the Cucker-Smale model with potentials is equivalent to a fully connected N-dimensional network of generalized mass-spring-dampers with appropriately defined Hamiltonian functions, that can be written in a port-Hamiltonian form

$$\dot{\mathbf{z}} = [\mathbf{J}(\mathbf{z}) - \mathbf{R}(\mathbf{z})] \frac{\partial \mathbf{H} z)}{\partial \mathbf{z}} \tag{4}$$

where $z = (q, p)$, with $q, p \in \mathbb{R}^{\frac{N(N-1)}{2}}$ being the vectors of relative distances and momenta between each pair of particles, and the quantities $J = -J^{\mathrm{T}}$, H and R are appropriately defined. The dependence of (5) on the interaction function ψ is introduced by the resistive term $R = R(\psi)$ [20].

It is straightforward to show that the CS model with leadership is equivalent to an input-state-output port-Hamiltonian system of the form

$$\dot{\mathbf{z}} = [\mathbf{J}(\mathbf{z}) - \mathbf{R}(\mathbf{z})] \frac{\partial \mathbf{H}(\mathbf{z})}{\partial \mathbf{z}} + \mathbf{g}(\mathbf{z})\mathbf{u}, \tag{5}$$

where $g(z)$ is appropriately defined, and u is an external control input that affects only the leader particles and is responsible for their trajectories.

The intuitive difference in the interaction function is actually the sole difference between the Boids model and the Cucker-Smale model with potentials. This also justifies why we may use the Boids model to simulate and the CS model to learn, and why approaching the simulation and learning problems with a single dynamical system is important for reconstructing the dynamics of complex swarm maneuvers. The difference in the interaction functions is illustrated in Fig. 2.

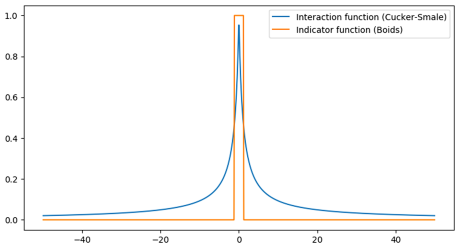

Fig. 2. The indicator "neighborhood" function in Boids model and the interaction function in Cucker-Smale model.

We would like to emphasize that all the models proposed in our work including port-Hamiltonian systems, Boids, CS interaction potentials, are useful abstractions inspired from biology and physics. However the underlying systems do not have to be biological or physical. The validity of the abstraction is measured by the degree with which these abstract models can generate dynamic trajectories very similar to the observed ones (or the observed time varying data series). Therefore these abstractions can be used, and have been used, to model the various networked systems we mentioned earlier.

3 Leader Detection

We adopt a majority vote criterion for leader detection, where each particle i votes for the particle j to be the leader, according to a measure related to the observed trajectories of the particles.

3.1 Granger Causality

Clive Granger in [10] defined a causality relationship based on two principles:

i. The cause happens prior to its effect
ii. The cause has unique information about the future values of its effect.

Given these assumptions, we say that a time series Y Granger-causes X if the past values of Y provide statistically significant information about the future values of X. In other words, we associate the existence of a causal effect of Y on X with the following *Hypothesis Test*:

Definition 2. *Let Y, X be stationary random processes, and consider the following two auto-regression models*

$$x_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i x_{t-i} + \epsilon_t^1, \ t > p \tag{6}$$

$$x_t = \alpha_0 + \sum_{i=1}^{p} \alpha_i x_{t-i} + \sum_{i=1}^{q} \beta_i y_{t-i} + \epsilon_t^2, \ t > \max\{p, q\} \tag{7}$$

where $\epsilon_t \sim N(0, \sigma^2)$ *is white noise. Then the non-causality null Hypothesis:*

$$H_0 : \beta_i = 0, \forall i \in \{1, \ldots, q\}$$

is rejected if model (7) fits the data $\{x_t\}_{t=T}^{T+n}$, $T > \max\{p, q\}$, *in a window of n samples, significantly better than model (6), i.e. if*

$$p \triangleq \mathbb{P}\left[F > \hat{F} | H_0\right] < a$$

for a given confidence level a, e.g. $a \leq 0.05$, *where*

$$\hat{F} = \frac{\frac{\sum_{t=T}^{T+n} \epsilon_t^1 - \sum_{t=T}^{T+n} \epsilon_t^2}{q}}{\frac{\sum_{t=T}^{T+n} \epsilon_t^2}{n - (p + q + 1)}} \tag{8}$$

We note that if (6) and (7) were simple regression models, the random variable F would be defined such that it follows an $F(q, n-(p+q+1))$ distribution. Because of the autoregression nature of (6), (7), it can be shown (e.g. Ch. 8 of [12]), that qF asymptotically follows a $\chi^2(q)$ distribution as $n \to \infty$. In case of non-stationary processes X, Y, one can apply the AR models to the n-th order differences, resulting in ARIMA models.

3.2 Leader Detection Based on Granger Causality

The leader-particle relationship is causal, satisfying both assumptions of Granger Causality. In order to make sure that we capture causality, and not merely correlation, we follow the hypothesis test described in Sect. 3.1 for each pair of particles (i, j).

As a result, a particle i votes for j to be the leader, where each vote takes the value G_{ij}, with $G = \mathbb{1}_{[p<\alpha]}$ indicating Granger Causality, where p is the p-value according to the χ^2 distribution as argued in Sect. 3.1.

However, because of the high correlation between the trajectories of the particles-followers, it is often the case that $G_{ij} \simeq 1$ even between two followers. In order to avoid such confusion, we bypass the last quantization step $G_{ij} = \mathbb{1}_{[p<\alpha]}$, and compare directly the p-values. Going one step further, we can see that the lowest p-value, corresponds to the highest $\hat{F}$-value. Moreover, the profile of the $\hat{F}_{ij}$ values is such that $\hat{F}_{ij}$ is *consistently* higher (i.e. lower variance) for every i, when j is the leader. In other words, even though for a follower j, a set of $\hat{F}_{ij}$ values may be high, indicating that particles with different indices i may be leaders, for each i, a high fluctuation on the observed values $\hat{F}_{ij}$ is indicative of a false positive, i.e. that particle i is not a leader. Therefore, we define the proposed leader detection algorithm to be based on the measure

$$F_{v,j} = \frac{\hat{\mu}_{\hat{F}_{\cdot j}}}{\hat{\sigma}_{\hat{F}_{\cdot j}}} = \frac{\sum_{i \neq j} \hat{F}_{ij}}{\sqrt{N \sum_{i \neq j} \hat{F}_{ij}^2 - \left(\sum_{i \neq j} \hat{F}_{ij}\right)^2}}$$

for each j. The measure $F_{v,j}$ can be thought of as the *inverse coefficient of variation*, and is designed such that particles i with high variation on the observed values $\hat{F}_{ij}$, for different followers-voters j, are not selected as leaders. The detection algorithm is shown in Algorithm 1.

Algorithm 1. F-Based Leader Detection Algorithm

Require: w(big enough), t, λ
  for i `in` $\{1, \ldots, N+1\}$ **do**
    for $j \neq i$ **do**
      `In the window` $[t-w, t]$`:`
      `Compute` $\hat{F}_{ij}$
      `Compute` $F_{v,j} = F_{v,j} = \dfrac{\hat{\mu}_{\hat{F}_{\cdot j}}}{\hat{\sigma}_{\hat{F}_{\cdot j}}}$
    end for
  end for
  `Select the leader:` $L_F \leftarrow \arg\max_j F_{v,j}$

4 Estimating the Number of Leaders

The first question one needs to answer when dealing with leader detection is the number of leaders that the algorithm is trying to find. We view this problem as

a clustering problem given a window of position and velocity observations of the particles, since it is reasonable to assume that particle trajectories will be more 'similar' to each other if they are following the same leader.

However, the number of the clusters is not known a priori, which makes standard clustering algorithms based on Vector Quantization (e.g. k-means) inappropriate for this application. Instead, we need an unsupervised learning algorithm that progressively estimates the number of clusters by adding new clusters only when some measure of distortion is high enough to support this decision. In this regard, the Deterministic Annealing algorithm [28] is a fitting clustering algorithm for estimating the number of leaders and is presented in the next Section.

4.1 Deterministic Annealing

The observation of annealing processes in physical chemistry motivated the use of similar concepts to avoid local minima of the optimization cost. Certain chemical systems can be driven to their low-energy states by annealing, which is a gradual reduction of temperature, spending a long time at the vicinity of the phase transition points.

Deterministic Annealing (DA), proposed by Rose [28], is an annealing optimization method that tries to achieve a good compromise between the world of stochastic relaxation, or simulated annealing [15], and the world of deterministic optimization. On the one hand it is deterministic, meaning that we do not want to be wandering randomly on the energy surface while making incremental progress on the average, as is the case for stochastic relaxation. On the other hand, it is still an annealing method and aims at the global minimum, instead of getting greedily attracted to a nearby local minimum. One can view DA as replacing stochastic simulations by the use of expectation. An effective energy function, which is parameterized by a (pseudo) temperature, is derived through expectation and is deterministically optimized at successively reduced temperatures.

The Optimization Problem. The problem of divergence-based Vector Quantization can be stated as an optimization problem:

Problem 1. Let $X : \Omega \to S$ be a random variable defined in the probability space $(\Omega, \mathcal{F}, \mathbb{P})$, and $d : S \times ri(S) \to [0, \infty)$ be a divergence measure, with $ri(S)$ representing the relative interior of S. Let $V := \{S_h\}_{h=1}^{k}$ be a partition of S with respect to d and $M := \{\mu_h\}_{h=1}^{k}$, such that $\mu_h \in ri(S_h),\ h \in K$, $K := \{1, \ldots, k\}$, and define the quantizer $Q : S \to S$ such that $Q(X) = \sum_{h=1}^{k} \mu_h \mathbb{1}_{[X \in S_h]}$.

Then the problem is formulated as

$$\min_{M,V} \ J(Q) := \mathbb{E}_X \left[d\left(X, Q(X)\right) \right]$$

The distortion function J is typically non convex and riddled with poor local minima. In order to deal with this phenomenon, soft-clustering approaches have been proposed as a probabilistic framework for clustering, where input vectors are assigned to clusters in probability.

For the randomized partition we can rewrite the expected distortion as

$$\begin{aligned} D &= \mathbb{E}\left[d_\phi(X, M)\right] \\ &= \mathbb{E}\left[\mathbb{E}\left[d_\phi(X, M)|X\right]\right] \\ &= \sum_x p(x) \sum_\mu p(\mu|x) d_\phi(x, \mu) \end{aligned}$$

where $p(\mu|x)$ is the association probability relating the input vector x with the codevector μ. At the limit where the association probabilities are hard and each input vector is assigned to a unique codevector with probability one, this becomes identical with the traditional hard clustering distortion.

We seek the distribution that minimizes D subject to a specified level of randomness, measured by the Shannon entropy

$$\begin{aligned} H(X, M) &= \mathbb{E}\left[-\log p(X, M)\right] \\ &= H(X) + H(M|X) \\ &= \mathbb{E}\left[-\log p(X)\right] + \mathbb{E}\left[\mathbb{E}\left[-\log p(M|X)|X\right]\right] \\ &= H(X) - \sum_x p(x) \sum_\mu p(\mu|x) \log p(\mu|x) \end{aligned}$$

by appealing to Jaynes's maximum entropy principle [13] which states: of all the probability distributions that satisfy a given set of constraints, choose the one that maximizes the entropy.

The optimization is conveniently formulated as the minimization of the Lagrangian

$$F = D - TH \tag{9}$$

where F represents the free energy and T is the temperature parameter that acts as a Lagrange multiplier. Clearly, for large values of T we maximize the entropy, and, as T is lowered, we trade entropy for reduction in distortion.

As in the case of Vector Quantization, we form a coordinate block optimization algorithm by successively minimizing with respect to the association probabilities $p(\mu|x)$ and the codevector locations μ. Minimizing F with respect to the association probabilities $p(\mu|x)$ is straightforward and gives the Gibbs distribution

$$p(\mu|x) = \frac{e^{-\frac{d_\phi(x,\mu)}{T}}}{\sum_\mu e^{-\frac{d_\phi(x,\mu)}{T}}}$$

while, in order to minimize F with respect to the codevector locations μ we set the gradients to zero

$$\begin{aligned} \frac{d}{d\mu} D = 0 \implies & \frac{d}{d\mu} \mathbb{E}\left[\mathbb{E}\left[d_\phi(X, \mu)|X\right]\right] = 0 \\ \implies & \sum_x p(x) p(\mu|x) \frac{d}{d\mu} d_\phi(x, \mu) = 0 \end{aligned}$$

Remark 1. If d_ϕ is a Bregman divergence [3,21], such as the Euclidean distance or the Kulback-Leibler divergence, we get $\frac{d}{d\mu} d_\phi(x, \mu) = \frac{d\phi}{d\mu}(\mu)(x - \mu)$, which allows for the direct computation of the optimal solution μ as the convenient centroid form

$$\mu = \mathbb{E}[x|\mu] = \frac{\sum_x xp(x)p(\mu|x)}{p(\mu)}$$

This deterministic optimization procedure takes place for decreasing values of the temperature T such that DA maintains minimum free energy (thermal equilibrium) while gradually lowering the temperature. Adding to the physical analogy, it is significant that, as the temperature is lowered, the system undergoes a sequence of "phase transitions", which consists of natural cluster splits where the cardinality of the codebook (number of clusters) increases. This is a bifurcation phenomenon and provides a useful tool for controlling the size of the clustering model relating it to the scale of the solution. At very high temperature ($T \to \infty$) the optimization yields uniform association probabilities

$$p(\mu|x) = \lim_{T \to \infty} \frac{e^{-\frac{d_\phi(x,\mu)}{T}}}{\sum_\mu e^{-\frac{d_\phi(x,\mu)}{T}}} = \frac{1}{K}$$

and all the codevectors are located at the same point

$$\mu = \mathbb{E}[X]$$

which is the expected value of X (in practice we get the sample mean of the N realizations of X that we observe). As we lower the temperature, the cardinality of the codebook changes. The bifurcation occurs when a set of coincident codevectors splits into separate subsets, which can be traced when the Hessian of F loses its positive definite property. In other words, the effective number of codevector depends only on the temperature parameter which is the Lagrange multiplier of the multi-objective minimization problem (9).

We can approach the bifurcation using perturbation analysis. At each temperature, we can generate a perturbed pair of codevectors for each effective cluster which, after convergence, can either merge together or separate depending on whether a phase transition has occurred.

The Algorithm. A computationally efficient implementation of the DA algorithm for clustering can be constructed in this way. The complete algorithm is shown in Algorithm 2 and constitutes a batch unsupervised learning algorithm that provides the ability to trade complexity for accuracy by progressively increasing the model size (number of efficient clusters) when needed (when a critical temperature has been reached). Furthermore, as argued in Remark 1, when d_ϕ is a Bregman divergence [3,21], such as the Euclidean distance or the Kulback-Leibler divergence, the optimization steps can be solved analytically providing a computationally efficient implementation.

Algorithm 2. Deterministic Annealing Algorithm

Require: Dataset $\mathcal{X}$ ▷ $|\mathcal{X}| = N$
  Set parameters:
    K_{max} ▷ maximum number of codevectors
    T_{max}, T_{min} ▷ maximum and minimum temperatures
  Initialize:
    $K = 1$ ▷ number of codevectors
    $T = T_{max} > 2\lambda_{max}(C_x)$ ▷ temperature
    $\mu_1 = \sum_x xp(x)$, $p(\mu_1) = 1$ ▷ 1^{st} codevector
  while $K < K_{max}$ **and** $T > T_{min}$ **do**
    Replace each μ_i with a perturbed pair $\{\mu_i', \mu_i''\}$
    Update:
    $p(\mu_i') = p(\mu_i'') = p(\mu_i)/2$
    $K \leftarrow 2K$
    repeat ▷ Step (O)
      for $i = 1, \ldots, K$ **do**
        Update:
        $p(\mu_i|x) \leftarrow \frac{p(\mu_i)e^{-\frac{d_\phi(x,\mu_i)}{T}}}{\sum_i p(\mu_i)e^{-\frac{d_\phi(x,\mu_i)}{T}}}$, $\forall x$ ▷ Step (E)
        $p(\mu_i) \leftarrow \sum_{x\in\mathcal{X}} p(x)p(\mu_i|x)$ ▷ Step (M_1)
        $\mu_i \leftarrow \frac{\sum_{x\in\mathcal{X}} xp(x)p(\mu_i|x)}{p(\mu_i)}$ ▷ Step (M_2)
      end for
    until Convergence ▷ $\|\Delta\mu_i\| < \epsilon_c$, $\forall i$
    Keep only effective codevectors:
    if $\|\mu_i - \mu_j\| < \epsilon_n$ **then**
      discard μ_j
      set $p(\mu_i) \leftarrow p(\mu_i) + p(\mu_j)$, $\forall i \neq j$
    end if
    Update K
    Lower the temperature ▷ $T \leftarrow \gamma T$
  end while
  Do one hard-clustering loop ▷ Step (O) with $T = 0$

5 Learning the Particle Interaction Laws

For the learning task we model the networked system of interacting agents as a port-Hamiltonian system representing a general Cucker-Smale model (5) [20]. We make use of the position and velocity trajectories of the particles to recover the resistive terms $R(z)$ and the Hamiltonian $H(z)$, which is equivalent to recovering the interaction functions $\psi_{ij}(x, v)$ of a general Cucker-Smale model (2).

The components of the interaction model (resistive element and the spring Hamiltonian) are modeled as neural-networks with one hidden layer, and the following optimization problem with a mean square error (MSE) loss function is formulated

$$\min_w \frac{1}{n}\sum_{i=1}^{n} \|\dot{\mathbf{z}}(t_i) - \dot{\hat{\mathbf{z}}}(t_i; w)\|^2 \tag{10}$$

$$\text{s.t.} \qquad \dot{\mathbf{z}}(t_i) = [\mathbf{J}(\mathbf{z}(t_i)) - \mathbf{R}(\mathbf{z}(t_i))] \frac{\partial \mathbf{H}(\mathbf{z}(t_i))}{\partial \mathbf{z}} + \mathbf{g}(z)\mathbf{u} \tag{11}$$

$$\dot{\hat{\mathbf{z}}}(\mathbf{t_i}; \mathbf{w}) = \left[\mathbf{J}(\mathbf{z}(\mathbf{t_i})) - \hat{\mathbf{R}}(\mathbf{z}(\mathbf{t_i}; \mathbf{w}))\right] \frac{\partial \hat{\mathbf{H}}(\mathbf{z}(\mathbf{t_i}; \mathbf{w}))}{\partial \mathbf{z}} + \mathbf{g}(\mathbf{z})\mathbf{u}, \tag{12}$$

where n is the number of time samples, $w = \{W^{[0]}, b^{[0]}, W^{[1]}, b^{[1]}\}$ is the set of optimization variables, and $\hat{(\cdot)}$ represents quantities estimated by the neural networks.

We approach the solution w^* of (10) with respect to

$$V_p(\theta) := \sum_{\tau=t_0}^{t_f} \|\dot{z}^*(\tau) - \dot{z}(\tau)\|^2$$

with an iterative gradient descent method

$$\theta^{n+1} = \theta^n - \alpha_n(\nabla_\theta V_p(\theta^n)), \ n = 0, 1, 2, \ldots \tag{13}$$

where the iteration maps $\alpha_n : \mathbb{R}^2 \to \mathbb{R}^2$, $n \geq 0$ are defined in accordance with the Adam method of moments for stochastic optimization [14], and the computation of the gradient vectors is implemented using automatic differentiation [17].

The term $g(z)\mathbf{u}$ is not estimated, but, instead, the actual trajectories of the leader particles are used, which incorporate the effect of this term. This requires the knowledge of the leader particles, as well as the followers of each leader. This is provided by the proposed algorithms for leader detection, presented in Sects. 3 and 4. In order to create a scalable learning system, we have focused on the Pytorch [26] deep learning platform that, in addition to automatic differentiation, is endowed with ODE solver capabilities.

6 Experimental Results

6.1 Case of One Leader

We showcase the proposed algorithm in the complex swarm movements shown in Fig. 3 and 5, where the trajectories of the particles are generated by the Cucker-Smale and extended Boids models with one leader, respectively.

We simulated the system of ODEs of the port-Hamiltonian system in (5), with the interaction function as reconstructed by the trained neural network, which resulted in the reconstructed particle trajectories that are depicted in Fig. 4 and 6.

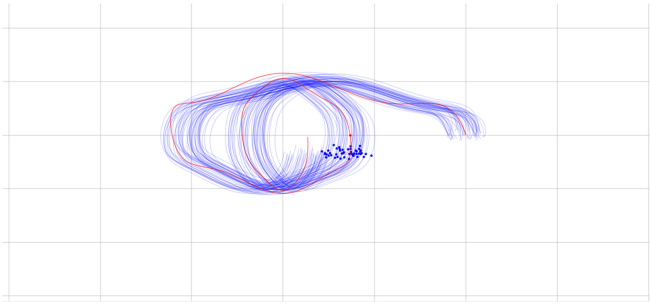

Fig. 3. An example of 2D particle trajectories of a swarm following the dynamics of a Cucker-Smale model with one leader.

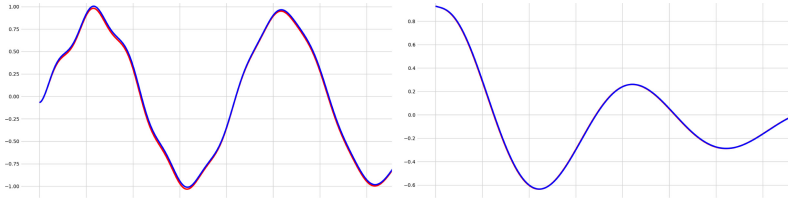

Fig. 4. The actual (blue) and estimated (red) trajectory of the position of a random agent over time for 20s (y-axis in arbitrary units). (left) The x-coordinate. MSE% = 0.0004. (right) The y-coordinate. MSE% = 0.0001. (Color figure online)

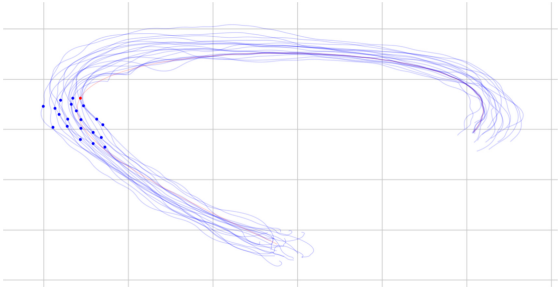

Fig. 5. An example of 2D particle trajectories of a swarm following the dynamics of an extended Boids model with one leader.

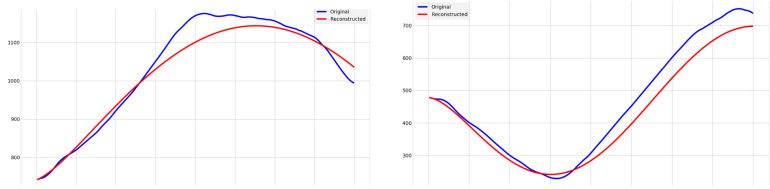

Fig. 6. The actual (blue) and estimated (red) trajectory of the position of a random agent over time for 20s (y-axis in arbitrary units). (left) The x-coordinate. MSE% = 0.1357. (right) The y-coordinate. MSE% = 0.1819. (Color figure online)

We note that the rule-based Boids model generates more jerky trajectories compared with the Cucker-Smale dynamical system and the reconstruction is less than ideal, as expected. This is an indication, however, that the proposed methodology is robust to noisy data generated by a model of unknown form.

6.2 Case of Multiple Leaders

We showcase the proposed algorithm in the complex swarm movement shown in the Fig. 7. where the trajectories of the particles are generated by the CS model with leadership with two leaders.

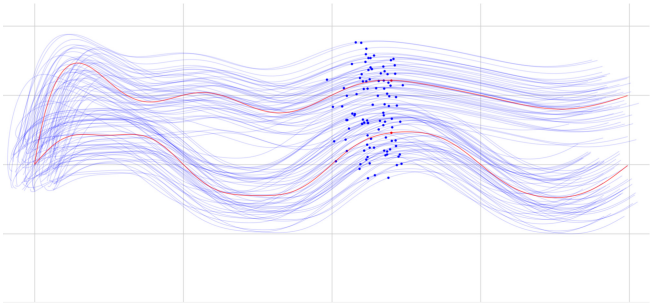

Fig. 7. An example of 2D particle trajectories of a swarm following the dynamics of a Cucker-Smale model with two leaders.

In order to apply our port-Hamiltonian based learning algorithm, we first estimate the sets $\mathcal{L}(i),\ \ 1 \leq i \leq N$ with our leader detection algorithm presented in Sects. 3 and 4. The results of the reconstruction of the interaction function are shown in Fig. 8.

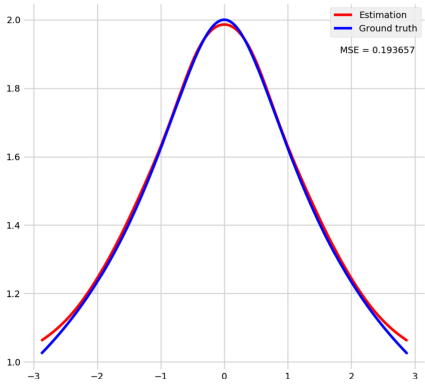

Fig. 8. The actual (blue) and estimated (red) particle interaction function of a swarm following the dynamics of a CS model with two leaders. The x- and y-axes are in arbitrary units. The mean squared error is $MSE = 0.193657$. The x-axis corresponds to the relative distance between a particle and its neighbor. (Color figure online)

7 Conclusion and Discussion

In this work we focus on the problem of defending against adversarial attacks by artificial UAV swarms. The swarms can be driven by multiple dynamically changing leaders and perform highly complex maneuvers. Existing air defense infrastructure is largely inadequate when dealing with the sheer number of agents in the swarm. In this research we propose a method which enables the identification of the leaders of the swarm, as well us the underlying coordination laws. This is the first and most challenging task in the defense strategy against hostile swarm attacks in existing air defense systems.

We develop a leader detection scheme based on the notion of Granger causality, relying on short-time observations of the trajectories of the UAVs. We then proceed with the online estimation of the swarm's coordination laws, modeled by a generalized Cucker-Smale model with non-local repulsive potential functions and dynamically changing leaders, through an appropriately defined iterative optimization algorithm. The proposed methodology is robust to both missing data and noise and is validated using simulation data of complex swarm movements.

While the key focus of this work is related to the defense against hostile UAV swarms, similar problems are found in many other types of large networked systems, including communication and computer networks, sensor networks, networked cyber-physical systems, biological systems, and social networks over the Internet. In such systems there are corresponding notions of leaders, such as initiators of a malicious attack, coordinators of malevolent behavior, initiators of a biological cell-malfunction, or influential sources of miss-information or untrustworthiness. In all these problems fast identification of the leaders and the associated follower groups (or influence groups) is essential for defending and correcting such malevolent actions and functions. Thus the applicability of the ideas and methods proposed in this work is very broad, with the appropriate modeling and semantic changes for the various domains. Important directions of our current and future research include extensions of the framework and algorithms to these broader domains, as well as the utilization of game theoretic methods for their analysis (non-cooperating, cooperating and mean-field games).

References

1. Bajec, I.L., Heppner, F.H.: Organized flight in birds. Anim. Behav. **78**(4), 777–789 (2009)
2. Ballerini, M., et al.: Interaction ruling animal collective behavior depends on topological rather than metric distance: evidence from a field study. Proc. Natl. Acad. Sci. **105**(4), 1232–1237 (2008)
3. Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with Bregman divergences. J. Mach. Learn. Res. **6**(Oct), 1705–1749 (2005)
4. Bongard, J., Lipson, H.: Automated reverse engineering of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **104**(24), 9943–9948 (2007)

5. Brunton, S., Proctor, J., Kutz, J.: Discovering governing equations from data by sparse identification of nonlinear dynamical systems. Proc. Natl. Acad. Sci. **113**(15), 3932–3937 (2016)
6. Carmi, A.Y., Mihaylova, L., Septier, F., Pang, S.K., Gurfil, P., Godsill, S.J.: MCMC-based tracking and identification of leaders in groups. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 112–119 (2011)
7. Carrillo, J., Fornasier, M., Toscani, G., Vecil, F.: Particle, Kinetic, and Hydrodynamic Models of Swarming. Birkhäuser Boston, Boston (2010)
8. Cucker, F., Smale, S.: Emergent behavior in flocks. IEEE Trans. Autom. Control **52**(5), 852–862 (2007)
9. Giardina, I.: Collective behavior in animal groups: theoretical models and empirical studies. HFSP J. **2**(4), 205–219 (2008)
10. Granger, C.W.J.: Investigating causal relations by econometric models and cross-spectral methods. Econometrica **37**(3), 424–438 (1969). http://www.jstor.org/stable/1912791
11. Ha, S.Y., Liu, J.G., et al.: A simple proof of the Cucker-Smale flocking dynamics and mean-field limit. Commun. Math. Sci. **7**(2), 297–325 (2009)
12. Hamilton, J.: Time Series Analysis. Princeton University Press (1994). https://books.google.com/books?id=B8_1UBmqVUoC
13. Jaynes, E.T.: Information theory and statistical mechanics. Phys. Rev. **106**(4), 620 (1957)
14. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
15. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**(4598), 671–680 (1983)
16. Lu, F., Zhong, M., Tang, S., Maggioni, M.: Nonparametric inference of interaction laws in systems of agents from trajectory data. arXiv preprint arXiv:1812.06003 (2018)
17. Maclaurin, D., Duvenaud, D., Johnson, M., Townsend, J.: Autograd (2018). https://github.com/HIPS/autograd
18. Mao, Z., Li, Z., Karniadakis, G.: Nonlocal flocking dynamics: learning the fractional order of PDEs from particle simulations. arXiv preprint arXiv:1810.11596 (2018)
19. Matei, I., de Kleer, J., Minhas, R.: Learning constitutive equations of physical components with constraints discovery. In: 2018 Annual American Control Conference (ACC), pp. 4819–4824, June 2018. https://doi.org/10.23919/ACC.2018.8431510
20. Matei, I., Mavridis, C., Baras, J.S., Zhenirovskyy, M.: Inferring particle interaction physical models and their dynamical properties. In: 2019 IEEE Conference on Decision and Control (CDC), pp. 4615–4621. IEEE (2019)
21. Mavridis, C.N., Baras, J.S.: Convergence of stochastic vector quantization and learning vector quantization with Bregman divergences. In: 21rst IFAC World Congress. IFAC (2020)
22. Mavridis, C.N., Tirumalai, A., Baras, J.S.: Learning interaction dynamics from particle trajectories and density evolution. In: 2020 59th IEEE Conference on Decision and Control (CDC). IEEE (2020)
23. Mavridis, C.N., Tirumalai, A., Baras, J.S., Matei, I.: Semi-linear Poisson-mediated flocking in a Cucker-Smale model. In: 24th International Symposium on Mathematical Theory of Networks and Systems (MTNS). IFAC (2021)
24. Mavridis, C.N., Vrohidis, C., Baras, J.S., Kyriakopoulos, K.J.: Robot navigation under MITL constraints using time-dependent vector field based control. In: 2019 IEEE 58th Conference on Decision and Control (CDC), pp. 232–237. IEEE (2019)

25. Okubo, A.: Dynamical aspects of animal grouping: swarms, schools, flocks, and herds. Adv. Biophys. **22**, 1–94 (1986)
26. Paszke, A., et al.: Automatic differentiation in PyTorch (2017)
27. Reynolds, C.: Flocks, herds and schools: a distributed behavioral model. In: ACM SIGGRAPH Computer Graphics, vol. 21, pp. 25–34. ACM (1987)
28. Rose, K.: Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. Proc. IEEE **86**(11), 2210–2239 (1998). https://doi.org/10.1109/5.726788
29. van der Schaft, A., Jeltsema, D.: Port-Hamiltonian systems theory: an introductory overview. Foundations Trends® Syst. Control **1**(2–3), 173–378 (2014). https://doi.org/10.1561/2600000002
30. Theodorakopoulos, G., Baras, J.S.: On trust models and trust evaluation metrics for ad hoc networks. IEEE J. Sel. Areas Commun. **24**(2), 318–328 (2006)
31. Will, T.E.: Flock leadership: understanding and influencing emergent collective behavior. Leadersh. Q. **27**(2), 261–279 (2016). https://doi.org/10.1016/j.leaqua.2016.01.002, http://www.sciencedirect.com/science/article/pii/S1048984316000035. Special Issue: Collective and Network Approaches to Leadership