Two-dimensional swarm formation in time-invariant external potential: Modeling, analysis, and control

Cite as: Chaos **30**, 093145 (2020); https://doi.org/10.1063/5.0019886
Submitted: 26 June 2020 . Accepted: 03 September 2020 . Published Online: 25 September 2020

Yanran Wang, and Takashi Hikihara

Two-dimensional swarm formation in time-invariant external potential: Modeling, analysis, and control

Cite as: Chaos **30**, 093145 (2020); doi: 10.1063/5.0019886
Submitted: 26 June 2020 · Accepted: 3 September 2020 ·
Published Online: 25 September 2020

Yanran Wang[a] and Takashi Hikihara[b]

AFFILIATIONS

Department of Electrical Engineering, Kyoto University, Kyoto 615-8510, Japan

[a]**Electronic mail:** y-wang@dove.kuee.kyoto-u.ac.jp
[b]**Author to whom correspondence should be addressed:** hikihara.takashi.2n@kyoto-u.ac.jp

ABSTRACT

Cluster formation has been observed in many organisms in nature. It has the desirable properties for designing energy efficient protocols for Wireless Sensor Networks (WSNs). In this paper, we present a new approach for energy efficient WSN protocols that investigates how the cluster formation of sensors responds to the external time-invariant energy potential. In this approach, the necessity for data transmission to the Base Station is eliminated, thereby conserving energy for WSNs. We define swarm formation topology and estimate the curvature of an external potential manifold by analyzing the change of the swarm formation in time. We also introduce a dynamic formation control algorithm for maintaining defined swarm formation topology in the external potential.

Energy conservation is a crucial challenge in Wireless Sensor Networks (WSNs). As energy for data transmission is most costly, WSNs' algorithms need to be designed in ways where data transmission, especially to the control center called the Base Station (BS), is minimized. Clustering is a possible mechanism to design energy efficient algorithms for WSNs. In this paper, we combine the idea of swarm intelligence with WSNs and design an algorithm that captures the environmental information through analyzing the change in sensor cluster formations (swarm formation) rather than gathering information directly through individual sensor measurements. In this approach, it is numerically clarified that the necessity for the BS is eliminated, and the formation is controllable based on the obtained information.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) have attracted much attention due to their ability to provide ubiquitous and multi-faceted situational awareness with a host of applications ranging from structural health monitoring, habitat surveillance, and target detection to power system management, smart car parking, and wireless luggage tags.[1–5] A WSN depends on spatially distributed sensor nodes to measure and collect desired environmental data within its sensing range and then transmit them to a control center called the Base Station (BS). The ideal WSN should be autonomous, robust, and scalable and has an extended network lifetime. However, WSNs are usually deployed in hostile environments where energy resources are limited. As energy is constrained and data transmission is most costly, WSN algorithms need to be designed in ways where data transmission, especially to the BS, is minimized. Clustering becomes an often utilized technique in designing energy efficient algorithms for WSNs.

Cluster formations can be readily observed in nature, such as bird flocking and fish schooling. These organism systems express collective motion behaviors and are studied in a relatively novel interdisciplinary field of research, Swarm Intelligence (SI). Individual agents in the swarm are simple agents with limited sensing abilities and computational rules that interact with each other locally. Nevertheless, the swarm as a whole demonstrates emergent global behaviors, which are unknown to the individuals. SI systems have been applied to tackle real-world problems, such as area coverage, with success.[6,7] It has the desirable properties of being distributed, autonomous, scalable, and robust,[8] all of which are key in designing algorithms for WSNs.

Many clustering algorithms have been developed.[9–11] However, besides the main objective of energy conservation, the existing clustering algorithms mainly focus on the optimization of sensor protocol routines to enhance WSNs in scalability, fault-tolerance, data aggregation, load balancing, and network topology stability, where data transmission to the BS is unavoidable. To address the energy efficiency challenge from a new perspective, this paper combines the SI concept with WSNs. We focus on designing the WSN algorithms where the desired environmental information is obtained through analyzing the change in sensor cluster formations (swarm formation) rather than information collected directly through individual sensors, thereby minimizing the energy expenditure in data transmission between sensors and eliminates the necessity for the BS.

We identify the environmental information as an external potential hypersurface M of $(n-1)$-dimensions ($n = 3$ or $n = 4$). Mathematically, M is a Riemannian manifold defined by the set of solutions to a single equation,

$$F(x_1, \ldots, x_n) = 0, \tag{1}$$

where F is a C^∞ function. We introduce a formation analysis algorithm that uses swarm formation in the external potential to estimate the curvature, which is invariant under isometry, of the manifold M.

This paper is organized as follows. In Sec. II, we introduce the topology of swarm lattice formation and demonstrate how to construct such formation in an arbitrary external potential. In Sec. III, we explain the theoretical basis and computations for the formation analysis algorithm. In Sec. IV, we formulate a formation control algorithm that shows how the swarm lattice formation can be obtained by the dynamic control of the individual agents. In Sec. V, we discuss the simulation results of both formation analysis and formation control algorithms.

II. SWARM FORMATION

The topology definition of the swarm formation is inspired by Reynolds[12] and Olfati-Saber.[13] In this paper, we consider line formation. In the line formation, agents are divided into two groups: one head agent and the following agents. The head agent acts as an initial stimuli to the swarm motion. Its dynamics can be predefined and are not affected by the following agents in the swarm.

Define a *path graph* P_n as a pair $(\mathcal{V}, \mathcal{E})$ that consists of a set of vertices $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and a set of edges $\mathcal{E}$ such that $\mathcal{E} \subseteq \{v_i, v_{i+1}\}$, where $i = 1, 2, \ldots, n-1$. Each vertex represents an agent of the swarm, while the edges represent the inter-agent communications. Let $q_i \in \mathbb{R}$ denote the position of the agent v_i for all $v_i \in \mathcal{V}$. The vector $q = (q_1, \ldots, q_n)^T$ is the configuration of all agents of the swarm. The inter-agent distance, that is, the length of edges, is defined to be the geodesic length between two connected vertices over $G(q)$. To maintain an identical inter-agent distance, we consider an algebraic constraint on the edges,

$$dis(\mathcal{E}_{v_i, v_{i+1}}) = d, \quad \forall\, v_i \in V, \quad d \in \mathbb{R}. \tag{2}$$

A configuration q that satisfies the set of constraints in (2) is referred to as a *lattice formation*.

To formulate lattice formation in any arbitrary external potential M, we need to first define the trajectory of the head agent, then construct a representation of edges—a parallel vector field that is metrically orthogonal to the head agent trajectory, and finally, calculate the geodesic deviation vector field. The trajectories of the following agents are the integral curves of the geodesic deviation vector field. The theoretical basis[14] and details of the formulation are discussed as follows.

Let $U \subseteq \mathbb{R}^n$ be a non-empty open subset and $F : U \to \mathbb{R}$ a C^∞ function defining the external potential. Let $M \subseteq U \times \mathbb{R}$ be the graph of f. The closed subset M in $U \times \mathbb{R}$ projects homeomorphically onto U with inverse $(x_1, \ldots, x_n) \mapsto (x_1, \ldots, x_n, F(x_1, \ldots, x_n))$ that is a smooth mapping from U to $U \times \mathbb{R}$. M is a closed smooth submanifold of $U \times \mathbb{R}$. Using the standard Riemannian metric on $U \times \mathbb{R} \subseteq \mathbb{R}^{n+1}$, the induced metric g on M at a point $p \in M$ is

$$g(p) = \langle \partial_{q_i}|_p, \partial_{q_j}|_p \rangle_p \, dq_i(p) \otimes dq_j(p), \tag{3}$$

with the coordinate chart $\{q_i\}$ on M. Each $\partial_{q_i}|_p \in T_pM$ can be represented as a linear combination of $\{\partial_{x_i}|_p\} \in T_p(\mathbb{R}^{n+1})$ given as

$$\partial_{q_i}|_p = \partial_{x_i}|_p + \partial_{x_i} f(p) \partial_{x_{n+1}}|_p. \tag{4}$$

Consider the aforementioned graph M as a C^∞ Riemannian manifold. Given a curve, $C : [a, b] \longrightarrow M$, a *vector field* X along C is any section of the tangent bundle TM over C ($X : [a, b] \longrightarrow TM$, projection $\pi : TM \longrightarrow M$, such that $\pi \circ X = C$). If M is a smooth manifold, all vector fields on the manifold are also smooth. We denote the collection of all smooth vector fields on the manifold M as $\mathfrak{X}(M)$. For a Riemannian manifold (M, g), the *Levi–Civita connection* ∇_g on M is the unique connection on TM that has both metric compatibility and torsion freeness. The Christoffel symbols of the second kind are the connection coefficients (in a local chart) of the Levi–Civita connection denoted as

$$\Gamma^a_{bc} = \frac{1}{2} g^{ad} (\partial_c g_{db} + \partial_b g_{dc} - \partial_d g_{bc}). \tag{5}$$

For a Riemannian manifold (M, g), a curve is called *geodesic* with respect to the connection ∇_g if its acceleration is zero, that is, a curve γ where $\nabla_{\dot\gamma} \dot\gamma = 0$. A geodesic curve in an n-dimensional Riemannian manifold can be expressed as a system of second-order ordinary differential equations,

$$\frac{d^2 \gamma^\lambda}{dt^2} + \Gamma^\lambda_{\mu\nu} \frac{d\gamma^\mu}{dt} \frac{d\gamma^\nu}{dt} = 0. \tag{6}$$

All geodesics are the shortest path between any two points on the manifold.

We predefine the trajectory of the head agent to be a geodesic curve on the manifold. Based on (6), for a two-dimensional manifold (M, g), the head agent trajectory $h(t)$ can be expressed as a system of ordinary differential equations,

$$\begin{aligned} \dot h_1 &= h_3, \\ \dot h_2 &= h_4, \\ \dot h_3 &= -\Gamma^x_{xx}(h_3)^2 - 2\Gamma^x_{xy} h_3 h_4 - \Gamma^x_{yy}(h_4)^2, \\ \dot h_4 &= -\Gamma^y_{xx}(h_3)^2 - 2\Gamma^y_{xy} h_3 h_4 - \Gamma^y_{yy}(h_4)^2, \end{aligned} \tag{7}$$

where a basis $\{x, y\}$ is used in the index and $\dot{h}$ is the first derivative with respect to time. Each Christoffel symbol depends entirely on the metric at a certain point in M in terms of the basis $\{x, y\}$. Given the initial conditions, $[x_1, x_2, \dot{x}_1, \dot{x}_2]$, (7) is guaranteed to have a solution according to the Picard–Lindelöf theorem. This choice of the head agent trajectory is made to simplify agent's dynamic control. On the geodesic, the head agent is traveling with a constant velocity given the initial position and velocity; thus, there is no requirement for an outside reference beacon.

The distance of the edges of lattice formation needs to be identical. For the representation of edges, a parallel vector field K that is orthogonal to the head agent trajectory $h(t)$ is constructed as

$$\nabla_K K = 0. \tag{8}$$

Here, K forms a family of geodesics where we require the initial position, and velocity conditions (u_0^j, v_0^j) for the jth geodesic $\gamma_j \in K$ are $u_0^j = h(t_j), v_0^j = v : g(v, \dot{h}(t_j)) = 0, j \in \mathbb{Z}$. The frequency of inter-agent communications is defined by the number of geodesics in K within the given traveling time.

The separation vector $s(t)$ connects a point $\gamma(t)$ on one geodesic to a point $\gamma(t) + s(t)$ on a nearby geodesic at the same time. For the parallel vector field K, we can construct a separation vector field S such that $s \in S$ is the separation vector described above. For the swarm lattice formation, the following agents' trajectories are the integral curves of S. Figure 1 gives a visual example of two-dimensional swarm lattice formation traveling in the external potential.

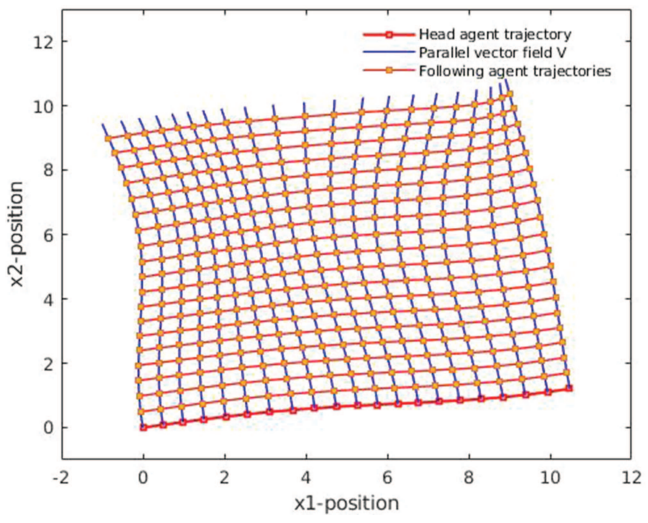

FIG. 1. Two-dimensional visual example of swarm lattice formation in the external potential $F = \sin\left(\frac{x_1}{a}\right) + \cos\left(\frac{x_2}{a}\right)$, $a = 2$. The travel time $t = 20$. The head agent is in red, with the initial condition [0, 0, $\cos(\pi/18)$, $\sin(\pi/18)$], and the following agents are in orange.

III. FORMATION ANALYSIS

In Fig. 1, one can notice the change in swarm lattice formation as it travels due to the curvature of the external potential manifold. As an invariant property under isometry, the curvature tensor gives valuable information about the external potential manifold itself. We quantify this change as the acceleration of the separation vectors $s \in S$ along K, which is equivalent to the change in the difference of neighboring agents' velocities. The geodesic deviation equation relates the acceleration of the separation vector between two neighboring geodesic curves to the Riemann curvature tensor. The theoretical elements[14,15] and the analysis method are described in detail as follows.

The Riemann curvature tensor is a $(1, 3)$ tensor defined through the Lie bracket on (M, g) as

$$R(X, Y)Z = \nabla_{[X,Y]}Z - \nabla_X\nabla_Y Z + \nabla_Y\nabla_X Z, \tag{9}$$

where $X, Y, Z \in \mathfrak{X}(M)$ and $R(X, Y)Z$ are vector-valued. $R(X, Y)Z$ can be expressed in a local chart as

$$R^{\rho}_{\sigma\mu\nu} = \partial_\mu \Gamma^{\rho}_{\nu\sigma} - \partial_\nu \Gamma^{\rho}_{\mu\sigma} + \Gamma^{\rho}_{\mu\lambda}\Gamma^{\lambda}_{\nu\sigma} - \Gamma^{\rho}_{\nu\lambda}\Gamma^{\lambda}_{\mu\sigma}. \tag{10}$$

By our definition in Sec. II, the separation vector s can be written as vectors,

$$s(t) = \tilde{\gamma}(t) - \gamma(t), \tag{11}$$

between two nearby geodesics. The separation acceleration field W is

$$W = \nabla_{\dot{\gamma}}\nabla_{\dot{\gamma}} S = \nabla_{\dot{\gamma}} V, \tag{12}$$

where V is the separation velocity field. In a local chart, W and V can be expressed as

$$V^{\rho} = \frac{dS^{\rho}}{dt} + \Gamma^{\rho}_{\mu\nu}\dot{\gamma}^{\mu}S^{\nu}, \tag{13}$$

$$W^{\rho} = \frac{dV^{\rho}}{dt} + \Gamma^{\rho}_{\lambda\sigma}\dot{\gamma}^{\lambda}V^{\sigma}, \tag{14}$$

where $\dot{\gamma} \equiv \frac{d\gamma}{dt}$. Combining (13) and (14) gives

$$W^{\rho} = \frac{d^2S^{\rho}}{dt^2} + 2\Gamma^{\rho}_{\mu\nu}\dot{\gamma}^{\mu}\frac{dS^{\nu}}{dt} + \frac{\partial\Gamma^{\rho}_{\mu\nu}}{\partial\gamma^{\lambda}}\dot{\gamma}^{\lambda}\dot{\gamma}^{\mu}S^{\nu} + \Gamma^{\rho}_{\mu\nu}\ddot{\gamma}^{\mu}S^{\nu} + \Gamma^{\rho}_{\lambda\sigma}\Gamma^{\sigma}_{\mu\nu}\dot{\gamma}^{\mu}\dot{\gamma}^{\lambda}S^{\nu}, \tag{15}$$

where $\ddot{\gamma} \equiv \frac{d^2\gamma}{dt^2}$. Rearranging (11) to be

$$\tilde{\gamma}(t) = \gamma(t) + s(t), \tag{16}$$

where the separation vector $s(t)$ is treated as an expansion parameter. Making use of the fact that both γ and $\tilde{\gamma}$ are geodesics,

we have

$$0 = \frac{d^2S^\rho}{dt^2} + 2\Gamma^\rho_{\mu\nu}\dot{\gamma}^\mu\frac{dS^\nu}{dt} + \frac{\partial\Gamma^\rho_{\mu\nu}}{\partial\gamma^\lambda}\dot{\gamma}^\mu\dot{\gamma}^\nu S^\lambda. \tag{17}$$

Inserting (17) into (15), the first-order geodesic deviation equation is

$$\begin{aligned} W^\rho &= -\left(\frac{\partial\Gamma^\rho_{\mu\lambda}}{\partial\gamma^\nu} - \frac{\partial\Gamma^\rho_{\mu\nu}}{\partial\gamma^\lambda} + \Gamma^\sigma_{\mu\lambda}\Gamma^\rho_{\sigma\nu} - \Gamma^\rho_{\lambda\sigma}\Gamma^\sigma_{\mu\nu}\right)\dot{\gamma}^\mu S^\nu\dot{\gamma}^\lambda \\ &\equiv -R^\rho_{\mu\nu\lambda}\dot{\gamma}^\mu S^\nu\dot{\gamma}^\lambda. \end{aligned} \tag{18}$$

A vector field J along a geodesic γ is called a Jacobi field if

$$\ddot{J} + R(J,\dot{\gamma})\dot{\gamma} = 0, \tag{19}$$

where $\ddot{J} \equiv \nabla_{\frac{d}{dt}}\nabla_{\frac{d}{dt}}J$ and $\dot{\gamma} \equiv \frac{d\gamma}{dt}$. Obviously, the deviation vector field S is a Jacobi field.

Furthermore, the sectional curvature is an equivalent but more geometrical description of the curvature of Riemannian manifolds. Let tangent 2-plane, Π_p, be the two-dimensional subspace in T_pM defined as $\Pi_p \equiv \text{span}\{u, v\}$, with $u, v \in T_pM$. The sectional curvature κ of (M, g) at a point $p \in M$ with respect to the plane Π_p is defined as

$$\begin{aligned} \kappa(\Pi_p) &= \kappa(X_p, Y_p) \\ &= \frac{\langle R(X,Y)X, Y\rangle_p}{|X|^2_p|Y|^2_p - \langle X, Y\rangle^2_p}. \end{aligned} \tag{20}$$

Substituting the vector fields K and S we constructed in Sec. II into (20), we have

$$\kappa(K,S) = \frac{\langle R(K,S)K, S\rangle}{|K|^2|S|^2 - \langle K, S\rangle^2}. \tag{21}$$

Also, S is in fact a Jacobi field along K that is also orthogonal to K. That is, $W \equiv \nabla^2_\tau S = -R(K,S)K$, and $\langle K, S\rangle = 0$. Combining with the fact that the integral curves of K are geodesic sets to have the velocity $|\dot{\gamma}| = 1$, the equation for the sectional curvature is simplified to

$$W = -\kappa(K,S)S. \tag{22}$$

For a two-dimensional Riemannian manifold (M, g), there is only one sectional curvature at each point $p \in M$.

In summary, by recording the change of swarm lattice formation in the external potential, the observer can quantify two variables: S is agents' velocities and W is the second-order difference of nearby agents' velocities. The geodesic deviation equation (22) relates W and S to the sectional curvature κ of the external potential manifold.

IV. FORMATION CONTROL

To stay in lattice formation as the swarm traveling through the external potential, the individual agent needs dynamic control of its trajectory. The trajectory of the individual agent can be considered as a dynamic system of the form[16]

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{F}(\mathbf{x}(t)), \tag{23}$$

where $\mathbf{x}$ is the state of the system, which is the position of the agent, and $\mathbf{F}$ is a vector field that depends on the system. In practice, we consider the equivalent discrete-time dynamic system,

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k), \tag{24}$$

where $\mathbf{x}_k$ is the sampling of the agent trajectory in (23) discretely in timesteps Δt. If we can assume that the system is of linear dynamics, then we can work with the form

$$\frac{d}{dt}\mathbf{x} = \mathbf{A}\mathbf{x}, \tag{25}$$

where it admits a closed-form solution

$$\mathbf{x}(t_0 + t) = e^{\mathbf{A}t}\mathbf{x}(t_0). \tag{26}$$

The entire system dynamics is characterized by the eigenvalues and eigenvectors of the matrix $\mathbf{A}$. Using spectral decomposition, one can simplify the dynamic system to

$$\frac{d}{dt}\mathbf{w} = \lambda\mathbf{w}, \tag{27}$$

with $\mathbf{A} = \mathbf{P}\lambda\mathbf{P}^{-1}$ and $\mathbf{w} = \mathbf{P}^{-1}\mathbf{x}$. Using $\mathbf{A}$, we can also predict agents' trajectories in time, thereby controlling the swarm to be in its lattice formation as time evolves.

Even though it is desirable to work with linear dynamic systems, the curvature property of the external potential manifold and thus the trajectories of agents are essentially nonlinear. To analyze a nonlinear dynamics with a linear technique, we utilize the Koopman operator theory.

A. Koopman operator theory and dynamic mode decomposition

Bernard O. Koopman has proved the possibility of representing a nonlinear dynamic system in terms of an infinite-dimensional linear operator acting on a Hilbert space of measurement functions of the state of the system. The basic elements of the Koopman spectral analysis are discussed below.[17–20]

Consider a real-valued measurement function $g : \mathbf{M} \longrightarrow \mathbb{R}$, known as *observables*, which belongs to the infinite-dimensional Hilbert space. The Koopman operator K_t is an infinite-dimensional linear operator that acts on the observable g as

$$K_t g = g \circ \mathbf{F}_t, \tag{28}$$

where $\mathbf{F}_t$ is the system dynamics and $\circ$ is the composition operator. For a discrete-time system with the time step Δt, it becomes

$$g(\mathbf{x}_{k+1}) = K_{\Delta t}g(\mathbf{x}_k). \tag{29}$$

Even though the Koopman operator is linear, it is also infinite-dimensional. Thus, it is important to identify key measurement functions that evolve linearly with the flow of the dynamics. Eigen-decomposition of the Koopman operator can provide us with such a set of measurement functions that captures the dynamics of the system and also behaves linearly in time. A discrete-time Koopman eigenfunction $\varphi(\mathbf{x})$ and its corresponding eigenvalue λ satisfy

$$\varphi(\mathbf{x}_{k+1}) = K_{\Delta t}\varphi(\mathbf{x}_k) = \lambda\varphi(\mathbf{x}_k). \tag{30}$$

Nonlinear dynamics become linear in this eigenfunction coordinate.

In a general dynamic system, the measurement functions can be arranged into a vector $\mathbf{g}$,

$$\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) \\ g_2(\mathbf{x}) \\ \vdots \\ g_m(\mathbf{x}) \end{bmatrix}. \tag{31}$$

Each measurement function may be expanded in terms of eigenfunctions $\varphi_j(\mathbf{x})$; thus, the vector $\mathbf{g}$ can be written as

$$\mathbf{g}(\mathbf{x}) = \sum_{j=1}^{\infty} \varphi_j(\mathbf{x})\mathbf{v}_j, \tag{32}$$

where $\mathbf{v}_j$ is the jth Koopman mode associated with the eigenfunction φ_j. Given this decomposition, we can represent the dynamics of the system in terms of the measurement function $\mathbf{g}$ as

$$\begin{aligned} \mathbf{g}(\mathbf{x}_k) &= K_{\Delta t}^k \mathbf{g}(\mathbf{x}_0) \\ &= K_{\Delta t}^k \sum_{j=0}^{\infty} \varphi_j(\mathbf{x}_0)\mathbf{v}_j \\ &= \sum_{j=0}^{\infty} K_{\Delta t}^k \varphi_j(\mathbf{x}_0)\mathbf{v}_j \\ &= \sum_{j=0}^{\infty} \lambda_j^k \varphi_j(\mathbf{x}_0)\mathbf{v}_j. \end{aligned} \tag{33}$$

The sequence of triples $\{(\lambda_j, \varphi_j, \mathbf{v}_j)\}_{j=0}^{\infty}$ is the Koopman mode decomposition.

Finding such a Koopman mode is extremely difficult even for a system with known governing equations. For systems with an unknown governing equation, such as in our situation, the dynamic mode decomposition (DMD) algorithm is adopted. As one of the modal decomposition techniques, DMD is first introduced in the fluid dynamic community to analyze the high-dimensional fluid state. It provides information about the dynamics of the flow in superposition of modes (similar to eigenmodes), even if the flow is nonlinear. The DMD analysis can be considered an approximation to the Koopman spectral analysis and also provides a numerical method for computing Koopman modes. In this paper, the formation control algorithm is inspired by the online DMD framework from Zhang *et al.*[21]

We consider a sequential set of data vectors $\{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_m\}$, where each $\mathbf{x}_k \in \mathbb{R}^n$ is the system state of time t_0–t_m. These data can be arranged into two matrices,

$$\mathbf{X} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_0 & \mathbf{x}_1 & \cdots & \mathbf{x}_{m-1} \\ | & | & & | \end{bmatrix}, \tag{34}$$

$$\mathbf{Y} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_m \\ | & | & & | \end{bmatrix}. \tag{35}$$

We assume that there exists an operator $\mathbf{A}$ that approximates the nonlinear dynamics of the system as

$$\mathbf{x}_{k+1} \approx \mathbf{A}\mathbf{x}_k. \tag{36}$$

Then, the best-fit operator $\mathbf{A}$ is defined as

$$\mathbf{A} = \arg\min_{\mathbf{A}} \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F, \tag{37}$$

where $\|\cdot\|_F$ is the Frobenius norm. The unique solution to the least-square problem is given by

$$\mathbf{A} = \mathbf{Y}\mathbf{X}^{\dagger}, \tag{38}$$

where $X^{\dagger}$ denotes the Moore–Penrose pseudoinverse of $\mathbf{X}$.

Having $\mathbf{A}$, computed from data from t_0 up to t_m, we can predict x_{m+1} of the system, that is, controlling agent's trajectory. Since new data are fed into the system as time progresses, $\mathbf{A}$ also changes. Unlike the standard DMD algorithm, the online DMD algorithm updates the operator $\mathbf{A}$ using the new system data, providing a more reliable operator $\mathbf{A}$ for the prediction of future system states. The algorithm does not compute the least-square problem of the whole system once new data have been updated. Alternatively, it computes $\mathbf{A}_{k+1}$ given $\mathbf{A}_k$ and new pairs of data (x_{k+1}, y_{k+1}), on the assumption that $\mathbf{A}_{k+1}$ is close to $\mathbf{A}_k$ in some sense.

Using (38), we have

$$\mathbf{A}_k = \mathbf{Y}_k\mathbf{X}_k^T(\mathbf{X}_k\mathbf{X}_k^T)^{-1}. \tag{39}$$

We define two new matrices $\mathbf{P}_k$ and $\mathbf{Q}_k$,

$$\mathbf{Q}_k = \mathbf{Y}_k\mathbf{X}_k^T, \tag{40}$$

$$\mathbf{P}_k = (\mathbf{X}_k\mathbf{X}_k^T)^{-1} \tag{41}$$

so that $\mathbf{A}_k = \mathbf{Q}_k\mathbf{P}_k$. $\mathbf{P}_k$ is well-defined if we ensure $\mathbf{X}_k\mathbf{X}_k^T$ is invertible. The operator $\mathbf{A}$ at the time t_{k+1} is related to $\mathbf{A}_k$ as

$$\begin{aligned} \mathbf{A}_{k+1} &= \mathbf{Q}_{k+1}\mathbf{P}_{k+1} \\ &= (\mathbf{Q}_k + \mathbf{y}_{k+1}\mathbf{x}_{k+1}^T)(\mathbf{P}_k^{-1} + \mathbf{x}_{k+1}\mathbf{x}_{k+1}^T)^{-1}. \end{aligned} \tag{42}$$

Using the Sherman–Morrison formula, we can express $\mathbf{P}_{k+1}$ as

$$\begin{aligned} \mathbf{P}_{k+1} &= (\mathbf{P}_k^{-1} + \mathbf{x}_{k+1}\mathbf{x}_{k+1}^T)^{-1} \\ &= \mathbf{P}_k - \frac{\mathbf{P}_k\mathbf{x}_{k+1}\mathbf{x}_{k+1}^T\mathbf{P}_k}{1 + \mathbf{x}_{k+1}^T\mathbf{P}_k\mathbf{x}_{k+1}}. \end{aligned} \tag{43}$$

$\mathbf{P}_{k+1}$ can be updated more efficiently, without the computation of inverses. Combining (43) and (42), we obtain the formula

$$\mathbf{A}_{k+1} = \mathbf{A}_k + \frac{(\mathbf{y}_{k+1} - \mathbf{A}_k\mathbf{x}_{k+1})\mathbf{x}_{k+1}^T\mathbf{P}_k}{1 + \mathbf{x}_{k+1}^T\mathbf{P}_k\mathbf{x}_{k+1}}. \tag{44}$$

$\mathbf{A}_{k+1}$ is computed using $\mathbf{A}_k$ and new data pair $\{\mathbf{x}_{k+1}, \mathbf{y}_{k+1}\}$.

B. Formation control algorithm

The aforementioned online DMD algorithm is a data-driven algorithm that predicts nonlinear dynamics. Therefore, even without knowledge of the external energy potential, it is still possible to control the lattice formation of the swarm. The general scheme of

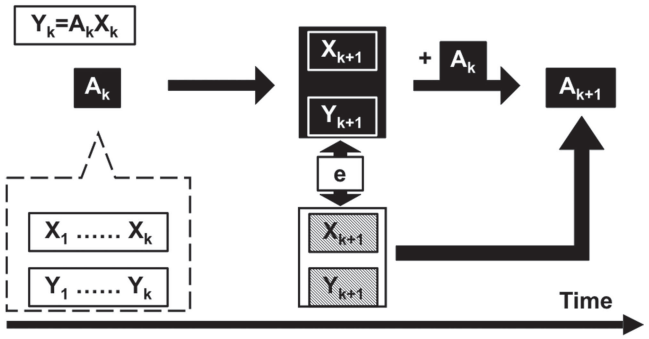

FIG. 2. The dynamic control algorithm scheme. The white box indicates raw system data obtained from measurements, the black box indicates algorithm calculated data, and the strip box indicates corrected data.

the control algorithm is shown in Fig. 2. Notice that if the predicted agent trajectory is not equivalent to the ideal trajectory of the lattice formation, a correction term needs to be added to the prediction. The details of the online DMD algorithm modeled specific to control the lattice formation of swarm are shown next.

Algorithm 1. Online DMD

(1) Collecting system data as it evolves in time. Arrange data into two matrices,

$$\mathbf{X} \equiv [x_0 \; x_1 \; \cdots \; x_{k-1}],$$
$$\mathbf{Y} \equiv [x_1 \; x_2 \; \cdots \; x_k].$$

(2) Compute $\mathbf{A}_k$ and $\mathbf{P}_k$ from (38) and (41).
(3) Predict $\mathbf{y}_{k+1}$ from $\mathbf{y}_{k+1} = \mathbf{A}_k\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{y}_k$.
(4) Correct $\mathbf{y}_{k+1}$ by agent's measurements.
(5) Update $\mathbf{A}_k$ and $\mathbf{P}_k$ using the corrected data pair $(\mathbf{x}_{k+1}, \mathbf{y}_{k+1})$, according to (42) and (43).

This algorithm is scalable to the rest of the following agents in the swarm; therefore, the whole swarm can be dynamically controlled to stay in the lattice formation.

V. SIMULATION AND DISCUSSION

A. Formation analysis algorithm

The formation analysis algorithm in Sec. III is performed on swarm lattice formation in three different external potentials,

1. elliptic paraboloid: $\frac{x_1^2 + x_2^2}{a}$,
2. hyperbolic paraboloid: $\frac{x_1^2 - x_2^2}{a}$, and
3. sinusoidal and cosinusoidal: $\sin\left(\frac{x_1}{a}\right) + \cos\left(\frac{x_2}{a}\right)$,

parameterized by a. All simulations used 100 following agents, with the traveling time $t = 10$.

The algorithm is able to estimate the sectional curvature of all three external potential manifolds with some extent of accuracy. Overall, the estimation accuracy increases with the decrease of curvature, while the frequency of inter-agent communications and inter-agent distance constraints does not significantly affect the algorithm accuracy shown in Figs. 3 and 4. However, the inter-agent distance directly affects the area in which the swarm is sensing. The accuracy of the algorithm is linked to the following features of the sensing area. For the external potential $F(x_1, x_2) = \sin\left(\frac{x_1}{a}\right) + \cos\left(\frac{x_2}{a}\right)$, both features are represented depending on the swarm sensing area.

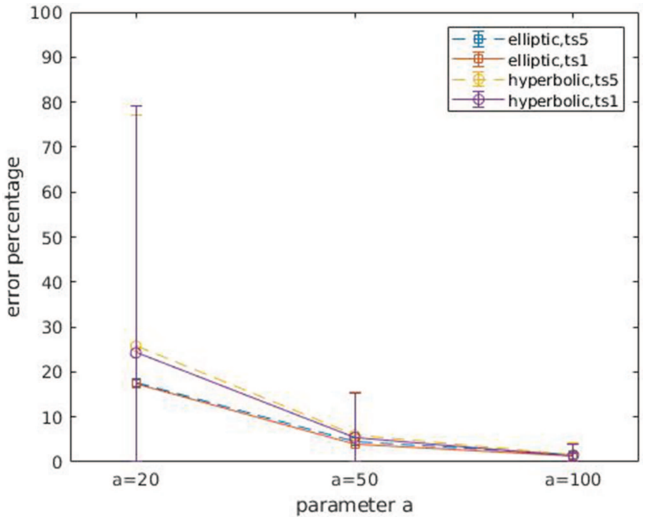

FIG. 3. Percentage error for sectional curvature estimation. All simulations used the same initial conditions for the head agent and have the inter-agent distance constraint $d = 0.1$ and the communication frequency $t_{s1} = 0.1$ and $t_{s2} = 0.5$. The square represents the mean error.

1. Jacobi field conjugate points

For the external potential manifold with a non-negative sectional curvature at each point, such as the elliptic paraboloid potential, there exist conjugate points in the vector field K. Consider $p, q \in M$ are two points connected by a geodesic γ. p, q are conjugate points along γ if there exists a non-zero Jacobi field along γ that vanishes at p and q. Conjugate points are when the geodesic fails, locally, to be the minimum length curve connecting two points. Thus, our geodesic deviation based algorithm also fails. A visual example of the conjugate point is shown in Fig. 5. Additional agent's protocols need to be installed to identify and bypass conjugate points.

2. Striction curve

The external potential two-dimensional manifold with a non-positive sectional curvature, such as the hyperbolic paraboloid, is a saddle surface and thus a ruled surface. For a noncylindrical ruled surface, it always has a parameterization of the form

$$\mathbf{r}(u, v) = \boldsymbol{\sigma}(u) + v\boldsymbol{\delta}(u), \tag{45}$$

where $\boldsymbol{\sigma}$ is called the striction curve. In particular, a hyperbolic paraboloid is a doubly ruled surface that has two striction curves.

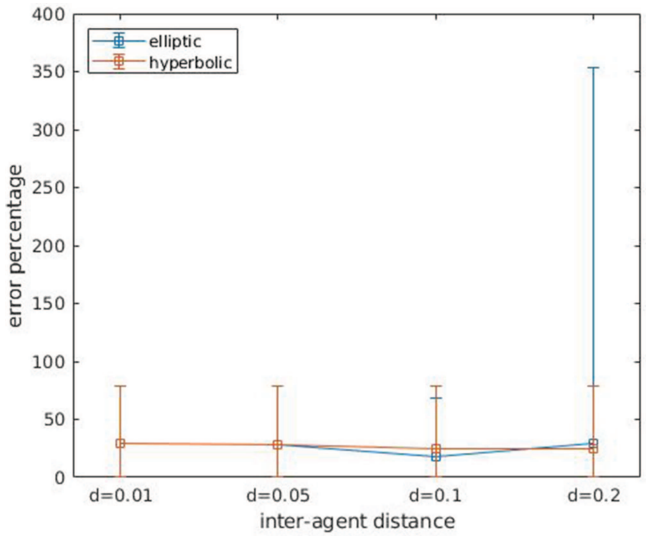

FIG. 4. Percentage error for sectional curvature estimation. All simulations used the same initial conditions for the head agent and the communication frequency $t_s = 0.1$. The square is the mean error.

In our simulation of the hyperbolic paraboloid potential, the two striction curves are $[x_1, \pm x_2]$. Shown in Fig. 6, if the head agent is traveling on striction curves, the swarm is unable to estimate the sectional curvature. The acceleration of the separation vector field is zero, equivalent to a flat space (zero curvature).

With the same number of agents in the swarm, the communication frequency and the inter-agent distance affect the sensing area. For an external potential manifold with feature one, a smaller sensing area is more likely to avoid conjugate points, while for an external potential manifold with feature two, a larger sensing area provides more information about the manifold. The overall shape of the swarm reveals that the external potential is not zero in Fig. 6, even though the curvature estimation is zero.

B. Formation control algorithm

In the simulation for the formation control algorithm in Sec. IV, we assume that the agents can measure the relative distance to their neighbors. The head agent has its own course of trajectory, thus not affected by other agents in the swarm. All the following agents take their right-hand neighbors (closer to the head agent) as a beacon to measure their own relative position. In ideal lattice formation, following agents are traveling in parallel and at a fixed distance to its right-hand neighbor, thus measurable in theory.

Figure 7 shows the velocity trajectory of the following agent that is next to the head agent. In this simulation, we use the time step $\Delta t = 0.1$, the data size $k = 3$, observable $(\mathbf{x}_k, \mathbf{y}_k)$ to be the velocity of the agent. In practice, the data $\mathbf{X}, \mathbf{Y}$ can only be collected by measuring the deviation of $k + 1$ agents traveling on nearby geodesics. Therefore, without any information of the external potential, the initial velocity of these agents can only be approximated in a Euclidean metric, not the external potential metric itself. Figure 7 shows that with a reliable correction step, the algorithm has a reliable prediction of the agent's dynamics and even uses data that are approximated in the Euclidean metric.

However, when agents have a near linear dynamics (constant velocity), the algorithm decreases in accuracy and eventually fails.

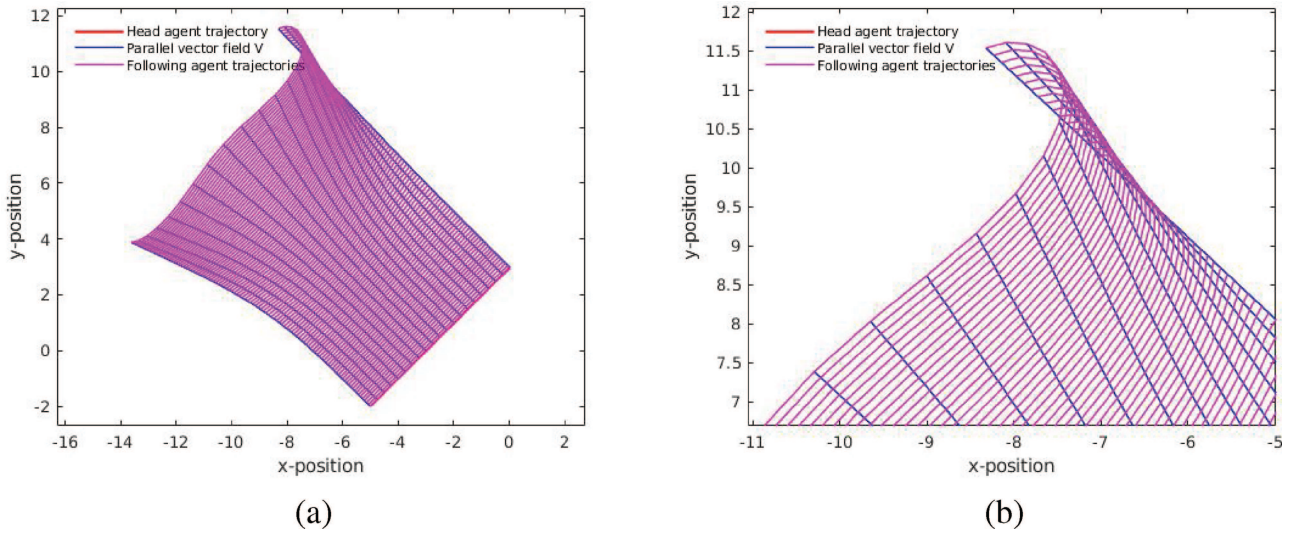

FIG. 5. Visual example of a conjugate point. The external potential is $F(x_1, x_2) = \sin\left(\frac{x_1}{a}\right) + \cos\left(\frac{x_2}{a}\right)$, $a = 2$, and the head agent's initial condition is $[-5, -2, \cos(4/\pi), \sin(4/\pi)]$. (a) Two-dimensional view of the swarm trajectory. The head agent trajectory is in red, and following agents' trajectory is in magenta. (b) Enlargement of (a) around the conjugate point.

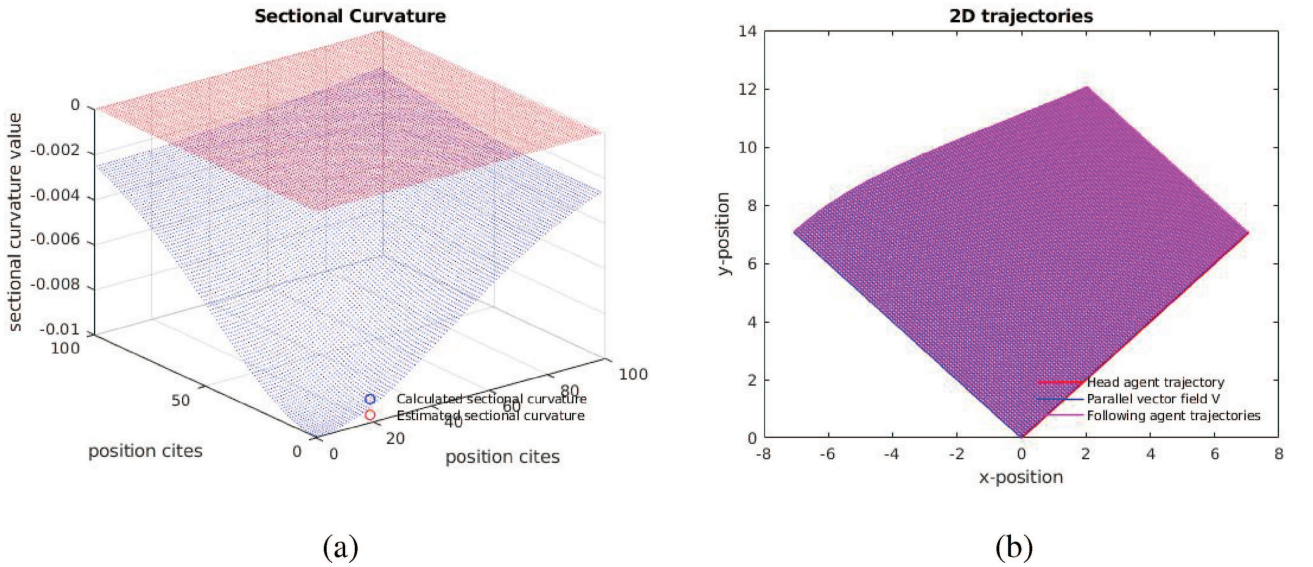

FIG. 6. Sectional curvature estimation and two-dimensional swarm trajectories. The external potential $F(x_1, x_2) = \frac{x_1^2 - x_2^2}{a}$, $a = 20$. The head agent is traveling on a striction curve $[x_1, x_2]$. (a) Sectional curvature estimation. The estimated curvature is in red, and the calculated curvature is in blue. (b) Two-dimensional view of swarm trajectories. The head agent's trajectory is in red, and following agents' trajectory is in magenta.

This is expected as the algorithm is based on the linear regression method. Therefore, an additional control protocol needs to be implanted when agents have linear dynamics, and the agents should have an autonomous decision on when to switch the dynamic control protocols.

C. Margin of algorithms

In formation analysis algorithm simulations, we have discussed for the same number of agents by changing the communication frequency and the inter-agent distance constraint, how the sensing

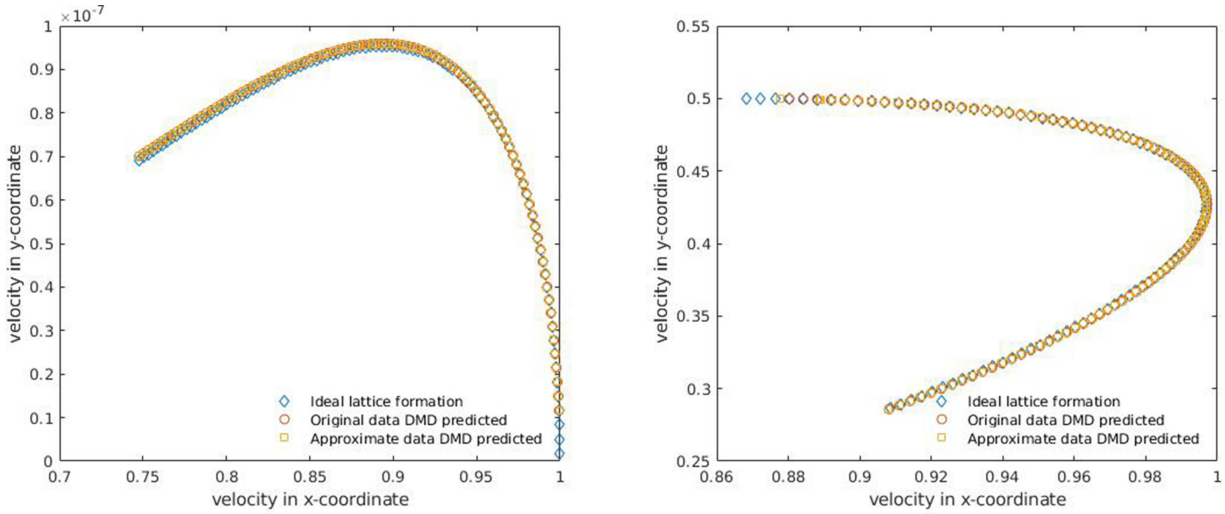

FIG. 7. Velocity trajectory of the first following agent. The ideal lattice formation velocity is in blue, the original data DMD predicted velocity is in red, and the approximate data DMD predicted velocity is in yellow. The external potential $F(x_1, x_2) = \frac{x_1^2 + x_2^2}{a}$, $a = 20$. The head agent's initial condition is [0, 0, 1, 0] (left) and $[-5, 0, \cos(6/\pi), \sin(6/\pi)]$ (right).

area of the swarm is beneficial in some cases, while in other cases, it affects the accuracy of the swarm formation analysis. Moreover, for the same sensing area, the number of agents has a positive correlation to the accuracy of the formation analysis. The size of the swarm also plays a role in swarm formation control. In formation control algorithm simulations, the control algorithm requires agents to be able to communicate regarding their relative distance and angle in terms of the external potential manifold metric. In practice, these types of sensing are extremely difficult. Approximation of this inter-agent distance can be made in an Euclidean metric by on-board agent sensories. We notice that when using the same number of agents while increasing the inter-agent distance or the communication frequency to enlarge the sensing area, the Euclidean metric approximately increases its error.

In summary, the smaller swarm size is more controllable but with lower accuracy in external potential estimation and vice versa. This conflict is due to the fact that for formation analysis, we utilize the nonlinearity in agents' trajectory to estimate a nonlinear property, namely, the external potential manifold curvature, while in formation control, we rely on linear approximation in a Euclidean metric to correct the agent's trajectory predictions. Therefore, the balance between the number of agents, the communication frequency, and the inter-agent distance is crucial in optimizing our approach of energy efficient WSN algorithms.

VI. CONCLUSION

We introduced a new approach for designing energy efficient WSN algorithms inspired by swarm intelligence. In this approach, we identify the clustering WSN as swarm and time-invariant external potentials as manifolds. By observing the change of swarm lattice formation in the external potential, we are able to estimate the curvature of the manifold, which is valuable information of the external potential. To maintain lattice formation in the external potential, the Koopman operator theory and DMD are used to predict the state of the nonlinear autonomous dynamic system of individual following agents, thus guiding the trajectory of the following agents. The concepts presented in this paper can also be extended to the time-variant external potential, where an individual agent is a nonlinear non-autonomous dynamic system.

ACKNOWLEDGMENTS

Y.W. is supported by the Japanese Government MEXT Scholarship program. Y.W. thanks V. Putkaradze for advice and encouragement. T.H. also thanks I. Mezic and Y. Susuki for helpful discussions.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] M. Erol-Kantarci and H. T. Mouftah, "Wireless sensor networks for cost-efficient residential energy management in the smart grid," IEEE Trans. Smart Grid **2**, 314–325 (2011).
[2] Y. Gao and R. Jin, "A novel wireless sensor networks platform for habitat surveillance," in *CSSE: International Conference on Computer Science and Software Engineering* (IEEE, 2008), Vol. 4, pp. 1028–1031.
[3] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," IEEE Signal Process. Mag. **19**, 17–29 (2002).
[4] Q. Ling, Z. Tian, Y. Yin, and Y. Li, "Localized structural health monitoring using energy-efficient wireless sensor networks," IEEE Sens. J. **9**, 1596–1604 (2009).
[5] S. Srikanth, P. Pramod, K. Dileep, S. Tapas, M. U. Patil, and C. B. N. Sarat, "Design and implementation of a prototype smart parking (SPARK) system using wireless sensor networks," in *2009 International Conference on Advanced Information Networking and Applications Workshops* (IEEE, 2009), pp. 401–406.
[6] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," Swarm Intell. **7**, 1–41 (2013).
[7] A. Schroeder, S. Ramakrishnan, M. Kumar, and B. Trease, "Efficient spatial coverage by a robot swarm based on an ant foraging model and the Lévy distribution," Swarm Intell. **11**, 39–69 (2017).
[8] C. Blum and X. Li, "Swarm intelligence in optimization," in *Swarm Intelligence Introduction and Applications* (Springer-Verlag, Berlin, 2008).
[9] M. Gomathi and P. Sivanesan, "Routing protocols based on cluster in wireless sensor networks: A survey," in *2019 International Conference on Intelligent Sustainable Systems (ICISS)* (IEEE, 2019), pp. 1–6.
[10] Y. Jin, L. Wang, Y. Kim, and X. Yang, "EEMC: An energy-efficient multi-level clustering algorithm for large-scale wireless sensor networks," Comput. Netw. **52**, 542–562 (2008).
[11] M. Mehdi Afsar and M.-H. Tayarani-N, "Clustering in sensor networks: A literature survey," J. Netw. Comput. Appl. **46**, 198–226 (2014).
[12] C. Reynolds, "Flocks, herds and schools: A distributed behavioral model," ACM SIGGRAPH Comput. Graph. **21**, 25–34 (1987).
[13] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: Algorithms and theory," IEEE Trans. Automat. Contr. **51**, 401–420 (2006).
[14] J. M. Lee, *Riemannian Manifolds: An Introduction to Curvature* (Springer-Verlag New York, Inc., New York, 1997).
[15] V. G. Ivancevic and T. T. Ivancevic, *Applied Differential Geometry: A Modern Introduction* (World Scientific Publishing Co. Pte. Ltd., Singapore, 2007).
[16] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Studies in Nonlinearity (Addison-Wesley, Reading, MA, 1994).
[17] S. Brunton and J. Kutz, *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control* (Cambridge University Press, Cambridge, 2019).
[18] A. Mauroy, I. Mezic, and Y. E. Susuki, *The Koopman Operator in Systems and Control Concepts, Methodologies, and Applications: Concepts, Methodologies, and Applications* (Springer, 2020).
[19] Y. Susuki, I. Mezic, F. Raak, and T. Hikihara, "Applied Koopman operator theory for power systems technology," Nonlinear Theor. Appl. IEICE **7**, 430–459 (2016).
[20] I. Mezić, "On applications of the spectral theory of the Koopman operator in dynamical systems and control theory," in *2015 54th IEEE Conference on Decision and Control (CDC)* (IEEE, 2015), pp. 7034–7041.
[21] H. Zhang, C. Rowley, E. Deem, and L. Cattafesta, "Online dynamic mode decomposition for time-varying systems," SIAM J. Appl. Dyn. Syst. **18**, 1586–1609 (2019).