FEEDBACK-BASED INHOMOGENEOUS MARKOV CHAIN APPROACH TO PROBABILISTIC SWARM GUIDANCE

Saptarshi Bandyopadhyay[*], **Soon-Jo Chung**[†], **and Fred Y. Hadaegh**[‡]

This paper presents a novel and generic distributed swarm guidance algorithm using inhomogeneous Markov chains that guarantees superior performance over existing homogeneous Markov chain based algorithms, when the feedback of the current swarm distribution is available. The probabilistic swarm guidance using inhomogeneous Markov chain (PSG–IMC) algorithm guarantees sharper and faster convergence to the desired formation or unknown target distribution, minimizes the number of transitions for achieving and maintaining the formation even if the swarm is damaged or agents are added/removed from the swarm, and ensures that the agents settle down after the swarm's objective is achieved. This PSG–IMC algorithm relies on a novel technique for constructing Markov matrices for a given stationary distribution. This technique incorporates the feedback of the current swarm distribution, minimizes the coefficient of ergodicity and the resulting Markov matrix satisfies motion constraints. This approach is validated using Monte Carlo simulations of the PSG–IMC algorithm for pattern formation and goal searching applications.

INTRODUCTION

In swarm robotics, a large number (100s-1000s) of relatively simple agents (like autonomous robots and small satellites) exhibit a desired collective behavior using only local interactions among themselves and with the environment.[1] Swarms of hundreds to thousands of femtosatellites (100-gram-class satellites) are being designed for applications like synthetic aperture radar, interferometry, etc.[2] Similarly, swarms of autonomous robots have demonstrated self-assembly of complex shapes.[3,4,5,6]

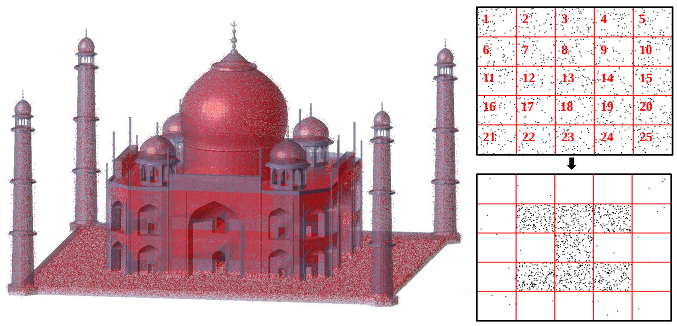
Applications of swarm robotics can be broadly classified into three categories:[1] (i) *Pattern formation*, where the agents only interact among themselves to self-organize into a predefined desired formation.[7,8,9,10] For example, see Fig. 1(a). (ii) *Goal searching*, where the agents mainly focus on the environment to estimate a unknown target distribution (e.g., distribution of chemical plume or oil spill).[11,12,13,14,15] (iii) *Cooperative control tasks*, where the agents have to balance inter-agent interactions and interactions with the environment to successfully perform tasks like cooperative transport, task allocation, surveillance, etc.[16,17,18,19,20,21,22,23] Let us assume that the state space can be partitioned into disjoint bins, as shown in Fig. 1(b), where the size of the bin is determined by the spatial resolution of the desired formation or target distribution. Let us also assume that the number of agents is at least an order of magnitude more than the number of bins. In this paper, we propose an efficient distributed solution approach using inhomogeneous Markov chains for all swarm guidance problems belonging to these three categories that satisfy these two assumptions.

Homogeneous Markov chain (HMC) based swarm guidance algorithms have been used for pattern formation,[9,10] goal searching,[13,14,15] task allocation[21] and surveillance applications.[22,23] Here the Markov matrix encodes the transition probabilities between bins. This swarm guidance approach is probabilistic (as opposed

[*]Graduate Research Assistant, Department of Aerospace Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, 61801, USA

[†]Assistant Professor, Department of Aerospace Engineering and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Urbana, Illinois, 61801, USA

[‡]Senior Research Scientist and Technical Fellow, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California 91109, USA



(a) (b)

Figure 1. (a) A swarm of million agents (shown in red) form the shape of the Taj Mahal (translucent silhouette shown in gray). The PSG–IMC algorithm ensures that the swarm converges to the desired formation and the agents settle down after the desired formation is achieved. (b) The state space is partitioned into 25 disjoint bins. The desired formation π of the letter "I" is given by $\frac{1}{7}[0,0,0,0,0,\ 0,1,1,1,0,\ 0,0,1,0,0,\ 0,1,1,1,0,\ 0,0,0,0,0]$.

to deterministic) because each agent determines its next bin from this Markov matrix using the inverse transform sampling method.[24] These HMC-based algorithms are inherently robust because addition or removal of agents or external damage to the swarm does not affect the Markov matrix, therefore the swarm always achieves its objective with the remaining agents.

If feedback of the current swarm distribution is available, then these HMC-based algorithms cannot incorporate this feedback because the Markov matrix cannot change with time. Intuitively it seems that this feedback can be used to increase the efficiency of these swarm guidance algorithms. Therefore, the objective of this paper is to develop the probabilistic swarm guidance using inhomogeneous Markov chain (PSG–IMC) algorithm to incorporate this feedback while retaining the original robustness properties. We show that these PSG–IMC algorithms guarantee superior performance over existing HMC-based algorithms for pattern formation and goal searching applications. To the best of our knowledge, this is the first paper in the swarm guidance literature that explores the idea of constructing inhomogeneous Markov chains (IMC) "on the fly" based on state feedback. We envisage that the techniques discussed in this paper can be used to boost the efficiency of algorithms in other areas in robotics where HMC-based algorithms are currently used.[25,26]

Paper Contribution

The first contribution of this paper is the novel technique for constructing feedback-based Markov matrices for a given stationary distribution. Our construction technique, discussed in Lemmas 1 and 2, relies on properties of Markov matrices. Each agent's inhomogeneous Markov matrix depends on the desired formation, the feedback of the current swarm distribution, the agent's current bin location, the time-varying motion constraints, and the agent's choice of parameters. We also minimize the coefficient of ergodicity using convex optimization in order to achieve faster convergence of the IMC.

The second contribution of this paper is the PSG–IMC algorithm for pattern formation, which is presented in **Method 1**. In HMC-based algorithm for pattern formation, each agent transitions using a synthesized homogeneous Markov matrix which has the desired formation as its stationary distribution.[10,9] Even after the swarm achieves the desired formation, the agents are not allowed to settle down because the Markov matrix forces the agents to transition. This limitation results in significant wastage of the agent's control effort (e.g., fuel). The proposed PSG–IMC algorithm not only addresses this limitation, but also guarantees better convergence to the desired formation and minimizes the number of transitions for achieving and maintaining the formation. Our core idea, which was first presented in Ref. 27, is that our feedback-based Markov matrix tends to the identity matrix when the current swarm distribution converges to the desired formation. This identity matrix ensures that the agents settle down after the desired formation is achieved, thereby minimizing unnecessary transitions. The PSG–IMC algorithm is numerically compared with HMC-based approaches for pattern formation in Remark 9, where these advantages are elucidated.

We then derive the convergence proofs for the PSG–IMC algorithm based on the analysis of IMC. Unlike the convergence proof for HMC, which is a direct application of the Perron–Frobenius theorem, the convergence proof for IMC is rather involved. We first show that each agent's IMC is strongly ergodic and the unique limit is indeed the desired formation shape. Moreover, we show that the rate of convergence is geometric and provide a conservative bound on this convergence rate. We also provide a conservative bound on the number of agents necessary for ensuring that the error between the swarm distribution and the desired formation is below the given convergence error threshold.

The final contribution of this paper is the PSG–IMC algorithm for goal searching, which is presented in **Method 2**. In the HMC-based goal searching algorithm, which cannot handle motions constraints, each agent randomly explores all the bins and waits in each bin for a time that is directly proportional to the target distribution in that bin.[13,14] After sufficient amount of time, the swarm converges to the target distribution but the agents are not allowed to settle down. We show that the PSG–IMC algorithm for goal searching efficiently incorporates the feedback of the current swarm distribution, satisfies motion constraints, and ensures that the swarm converges to the target distribution and the agents settle down after the target distribution is achieved.

The HMC-based algorithms for task allocation[21] and surveillance[22,23] are exactly similar to the previous HMC-based pattern formation algorithm, with the slight modification that the Markov matrix in this case encodes the transition probabilities between tasks or surveillance landmarks. Similarly coverage over an unknown region can be cast as a goal searching problem. Hence we conclude that all cooperative control tasks can be cast as either pattern formation or goal searching problems. Therefore, in this paper, we only focus on the PSG–IMC algorithms for pattern formation and goal searching.

Each agent determines its higher-level bin-to-bin guidance trajectory using the PSG–IMC algorithm so that the swarm achieves its objective. Each agent also needs a lower-level guidance and control algorithm, which depends on the agent's dynamics, to track this higher-level guidance trajectory in a collision-free manner. For the sake of concise presentation, we do not discuss these lower-level algorithms in this paper. Such lower-level algorithms based on model predictive control or Voronoi partitions are discussed in Refs. 3, 28.

PROBLEM STATEMENT AND PRELIMINARIES

In this section, we first define some key concepts, then state the problem statement for pattern formation, and finally discuss various methods for generating the feedback of the current swarm distribution.

Definition 1. *(Bins $R[i]$)* Let $\mathcal{R} \subset \mathbb{R}^{n_x}$ denote the n_x-dimensional compact physical space over which the swarm is distributed. The region $\mathcal{R}$ is partitioned into n_{cell} disjoint bins represented by $R[i]$, $i = 1, \ldots, n_{\text{cell}}$ so that $\bigcup_{i=1}^{n_{\text{cell}}} R[i] = \mathcal{R}$ and $R[i] \cap R[q] = \emptyset$, if $i \neq q$. The size of the bins is determined by the spatial resolution of the desired formation or target distribution. For example, the state space is partitioned into 25 disjoint bins in Fig. 1(b). □

Definition 2. *(Desired Formation $\boldsymbol{\pi}$ and Recurrent Bins $\boldsymbol{\Pi}$)* Let the desired formation shape be represented by a probability (row) vector $\boldsymbol{\pi} \in \mathbb{R}^{n_{\text{cell}}}$ over the bins in $\mathcal{R}$, i.e., $\boldsymbol{\pi}\mathbf{1} = 1$. Note that $\boldsymbol{\pi}$ can be any arbitrary probability vector, but it is the same for all agents in the swarm. For example, $\boldsymbol{\pi}$ is the shape of the letter "I" in Fig. 1(b). Let us define $\boldsymbol{\Pi}$ as the set of all bins that have nonzero probabilities in $\boldsymbol{\pi}$:

$$\boldsymbol{\Pi} := \{R[\ell] \,:\, \ell \in \{1, \ldots, n_{\text{cell}}\};\ \text{and}\ \boldsymbol{\pi}[\ell] > 0\}\ . \tag{1}$$

The bins in the set $\boldsymbol{\Pi}$ are called recurrent bins while those not in $\boldsymbol{\Pi}$ are called transient bins. Let n_{rec} be the number of bins in $\boldsymbol{\Pi}$. □

Let $m_k \in \mathbb{N}$ agents belong to the swarm at the k^{th} time instant. The algorithms in this paper are most suitable for swarm guidance problems where $m_k \gg n_{\text{rec}}$, because we control the swarm density distribution over the bins. Note that any guidance algorithm can only achieve the best quantized representation of the desired formation $\boldsymbol{\pi}$ using a swarm of m_k agents, where $\frac{1}{m_k}$ is the quantization factor.

We assume that the agents are *anonymous* and *identical*, i.e., the agents do not have any global identifiers and all agents execute the same algorithm.[29] If the agents are indexed (non-anonymous), then a pseudo-centralized algorithm can be executed using a spanning tree approach,[30] which is not possible in our case. We assume that each agent can sense its global position or bin location in the state space. With the advent of low cost GPS receivers, radio beacons for triangulation, base stations, etc. it is possible for modern-day autonomous robots and small satellites to estimate their global position. Another work-around is to setup a local reference frame for the swarm using some of the agents in the swarm.[4]

Definition 3. *(Current Swarm Distribution $\boldsymbol{\mu}_k^\star$)* Let the row vector $\boldsymbol{r}_k^j$ represent the bin in which the j^{th} agent is actually present at the k^{th} time instant. If $\boldsymbol{r}_k^j[i] = 1$, then the j^{th} agent is inside the bin $R[i]$ at the k^{th} time instant; otherwise $\boldsymbol{r}_k^j[i] = 0$. The current swarm distribution ($\boldsymbol{\mu}_k^\star$) is given by the ensemble mean of actual agent positions:

$$\boldsymbol{\mu}_k^\star := \frac{1}{m_k} \sum_{j=1}^{m} \boldsymbol{r}_k^j \,. \tag{2}$$

Clearly, $\boldsymbol{\mu}_k^\star$ is a probability (row) vector over the bins in $\mathcal{R}$ because $\boldsymbol{\mu}_k^\star \mathbf{1} = 1$. □

Definition 4. *(Motion Constraints $\boldsymbol{A}_k^j$)* If the agent in a particular bin can only transition to some bins but cannot transition to other bins, because of the dynamics or physical constraints, then these (possibly time-varying) motion constraints are specified by the matrix $\boldsymbol{A}_k^j$ as follows:

$$\boldsymbol{A}_k^j[i,\ell] = \begin{cases} 1 & \text{if the } j^{\text{th}} \text{agent can transition to } R[\ell] \\ 0 & \text{if the } j^{\text{th}} \text{agent cannot transition to } R[\ell] \end{cases}, \qquad \text{where } \boldsymbol{r}_k^j[i] = 1, \text{ for all } \ell \in \{1,\ldots,n_{\text{cell}}\} \,. \tag{3}$$

We assume that the matrix $\boldsymbol{A}_k^j$ is symmetric and the graph conforming to the $\boldsymbol{A}_k^j$ matrix is strongly connected. Moreover, an agent can always choose to remain in its current bin, i.e., $\boldsymbol{A}_k^j[i,i] = 1$ for all $i \in \{1,\ldots,n_{\text{cell}}\}$ and $k \in \mathbb{Z}^*$. Moreover, $\boldsymbol{A}_k^j$ must satisfy Assumption 1 in order to ensure that each Markov matrix has a single essential class. □

Let us now state the problem statement. The objectives of the PSG–IMC algorithm for pattern formation are as follows:
(i) Each agent independently determines its trajectory using a Markov chain, which obeys motion constraints ($\boldsymbol{A}_k^j$), so that the overall swarm converges to a desired formation ($\boldsymbol{\pi}$).
(ii) The algorithm reduces the number of transitions for achieving and maintaining the formation in order to conserve control effort (e.g., fuel).
(iii) The algorithm automatically detects and repairs damages to the formation.
The key idea of the proposed PSG–IMC algorithm for pattern formation is that each agent independently determines its trajectory using a feedback-based inhomogeneous Markov chain that tends to an identity matrix as the swarm converges to the desired formation. The pseudo-code for this algorithm is given in **Method 1**.

Generating Feedback of the Current Swarm Distribution

Here we discuss different techniques for generating the feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$. If a central controller observes the entire swarm, then this controller can broadcasts $\boldsymbol{\mu}_k^\star$ to all the agents in the swarm, if such a broadcast is possible. Even if $\boldsymbol{\mu}_k^\star$ is broadcast to a few agents in the swarm, then $\boldsymbol{\mu}_k^\star$ can be rapidly relayed to all the agents using the swarm's communication network topology. Note that the assumption of a central controller observing the entire swarm is valid for many practical scenarios. For example, a ground station can observe a swarm of satellites and can broadcast $\boldsymbol{\mu}_k^\star$ to these satellites.

If such a central controller is not available, then the agents have to rely on each other in order to estimate the current swarm distribution $\boldsymbol{\mu}_k^\star$ in a distributed manner. If the standard consensus algorithm[31,32,33] is used by the agents for estimating $\boldsymbol{\mu}_k^\star$, then the number of iterations necessary for convergence increases exponentially with the number of agents. Instead, the agents should use faster linear-time consensus algorithms.[34,35]

Let the row vector $\boldsymbol{\mu}_k^j$ represent the j^{th} agent's estimate of the current swarm distribution at the k^{th} time instant, which is used henceforth in this paper. If a central controller is available, then $\boldsymbol{\mu}_k^j = \boldsymbol{\mu}_k^\star$ for all agents. If the consensus algorithm is used to estimate $\boldsymbol{\mu}_k^\star$, then $\|\boldsymbol{\mu}_k^j - \boldsymbol{\mu}_k^\star\|_2 \leq \varepsilon_{\text{cons}}$ for all agents where $\varepsilon_{\text{cons}} > 0$ is the convergence error of the consensus algorithm. The effect of this convergence error on the PSG–IMC algorithm is discussed in Remark 10.

Note that our higher-level PSG–IMC algorithm does not need inter-agent communication, but this process of generating feedback or the lower-level guidance and control algorithm might require inter-agent communication.

CONSTRUCTION OF FEEDBACK-BASED INHOMOGENEOUS MARKOV CHAIN

In this section, we first present the construction of a feedback-based inhomogeneous Markov matrix that can incorporate motion constraints. We then present the PSG–IMC algorithm for pattern formation and compare it with existing HMC-based approaches.

Let $\boldsymbol{x}_k^j \in \mathbb{R}^{n_{\text{cell}}}$ denote the row vector of probability mass function (pmf) of the predicted position of the j^{th} agent at the k^{th} time instant, i.e., $\boldsymbol{x}_k^j \mathbf{1} = 1$. The i^{th} element ($\boldsymbol{x}_k^j[i]$) is the probability of the event that the j^{th} agent is in $R[i]$ bin at the k^{th} time instant:

$$\boldsymbol{x}_k^j[i] = \mathbb{P}(\boldsymbol{r}_k^j[i] = 1), \quad \text{for all } i \in \{1, \dots, n_{\text{cell}}\} \,. \tag{4}$$

The elements of the row stochastic Markov matrix $\boldsymbol{M}_k^j \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}$ are the transition probabilities of the j^{th} agent at the k^{th} time instant:

$$\boldsymbol{M}_k^j[i,\ell] := \mathbb{P}\left(\boldsymbol{r}_{k+1}^j[\ell] = 1 | \boldsymbol{r}_k^j[i] = 1\right) \,. \tag{5}$$

In other words, the probability that the j^{th} agent in $R[i]$ bin at the k^{th} time instant will transition to $R[\ell]$ bin at the $(k+1)^{\text{th}}$ time instant is given by $\boldsymbol{M}_k^j[i,\ell]$. The Markov matrix $\boldsymbol{M}_k^j$ determines the time evolution of the pmf row vector $\boldsymbol{x}_k^j$ by:

$$\boldsymbol{x}_{k+1}^j = \boldsymbol{x}_k^j \boldsymbol{M}_k^j, \quad \text{for all } k \in \mathbb{Z}^* \,. \tag{6}$$

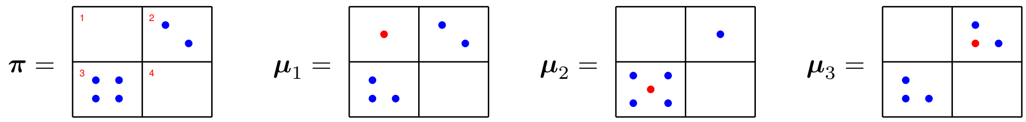
The standard algorithm for constructing a Markov matrix with a given stationary distribution is the well-known Metropolis-Hastings (MH) algorithm.[36,37] In the MH algorithm, the proposal distribution is used to iteratively generate the next sample, which is accepted or rejected based on the desired stationary distribution. Note that there is no direct method for incorporating feedback into the MH algorithm. In this paper, we present a much simpler method for constructing a Markov matrix, where the feedback of the current swarm distribution can be directly incorporated within the construction process.

Let ξ_k^j represent a tuning parameter for the j^{th} agent at the k^{th} time instant, which depends on the feedback of the current swarm distribution $\boldsymbol{\mu}_k^j$ and the desired formation $\boldsymbol{\pi}$. Let $\boldsymbol{\alpha}_k^j \in \mathbb{R}^{n_{\text{cell}}}$ be a positive bounded column vector with $\|\boldsymbol{\alpha}_k^j\|_\infty \leq 1$. The following lemma presents the construction of the Markov matrix $\boldsymbol{M}_k^j$ with the stationary distribution $\boldsymbol{\pi}$.

Lemma 1. *(Construction of Markov Matrix)* For a given tuning parameter ξ_k^j and positive column vector $\boldsymbol{\alpha}_k^j$, the following row stochastic Markov matrix $\boldsymbol{M}_k^j$ has $\boldsymbol{\pi}$ as its stationary distribution (i.e., $\boldsymbol{\pi}\boldsymbol{M}_k^j = \boldsymbol{\pi}$):

$$\boldsymbol{M}_k^j = \boldsymbol{\alpha}_k^j \frac{\xi_k^j}{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j} \boldsymbol{\pi} \operatorname{diag}(\boldsymbol{\alpha}_k^j) + \mathbf{I} - \xi_k^j \operatorname{diag}(\boldsymbol{\alpha}_k^j), \tag{7}$$

where $\boldsymbol{\pi}\boldsymbol{\alpha}_k^j \neq 0$ and $\sup_k \xi_k^j \|\boldsymbol{\alpha}_k^j\|_\infty \leq 1$. This Markov matrix has the following properties:
(i) $\boldsymbol{M}_k^j$ only allows transitions into the recurrent bins, i.e., for $i \neq \ell$ and $\xi_k^j > 0$, $\boldsymbol{M}_k^j[i,\ell] > 0$ if and only if $R[\ell] \in \boldsymbol{\Pi}$,
(ii) the diagonal elements of recurrent bins are positive, i.e., $\boldsymbol{M}_k^j[i,i] > 0$ if $R[i] \in \boldsymbol{\Pi}$ and $\xi_k^j > 0$,
(iii) if $\xi_k^j = 0$, then $\boldsymbol{M}_k^j = \mathbf{I}$, and
(iv) if $\boldsymbol{\alpha}_k^j = \mathbf{1}$, then $\boldsymbol{M}_k^j = \xi_k^j \mathbf{1}\boldsymbol{\pi} + (1 - \xi_k^j)\mathbf{I}$.



(a) Desired distribution (b) Case 1 (c) Case 2 (d) Case 3

Figure 2. The desired distribution (π) is shown in (a), where bins 2 and 3 have 2 and 4 agents respectively. Three cases are shown in (a–c), where one agent (marker in red) is not in its correct bin. The $\mathcal{L}_1$ distances for these cases from π are the same, i.e., $D_{\mathcal{L}_1}(\boldsymbol{\mu}_1, \boldsymbol{\pi}) = D_{\mathcal{L}_1}(\boldsymbol{\mu}_2, \boldsymbol{\pi}) = D_{\mathcal{L}_1}(\boldsymbol{\mu}_3, \boldsymbol{\pi}) = 0.33$. But the Hellinger distances for these cases are different, i.e., $D_H(\boldsymbol{\mu}_1, \boldsymbol{\pi}) = 0.30$, $D_H(\boldsymbol{\mu}_2, \boldsymbol{\pi}) = 0.14$, and $D_H(\boldsymbol{\mu}_3, \boldsymbol{\pi}) = 0.12$.

Proof: For a valid first term in Eq. (7), we need $\boldsymbol{\pi}\boldsymbol{\alpha}_k^j \neq 0$. We first show that $\boldsymbol{M}_k^j$ is a row stochastic matrix. Right multiplying both sides of Eq. (7) with $\mathbf{1}$ gives $\boldsymbol{M}_k^j\mathbf{1} = \xi_k^j \boldsymbol{\alpha}_k^j \frac{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j}{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j} + \mathbf{1} - \xi_k^j \mathrm{diag}(\boldsymbol{\alpha}_k^j)\mathbf{1} = \mathbf{1}$. Next, we show that $\boldsymbol{M}_k^j$ is a Markov matrix with $\boldsymbol{\pi}$ as its stationary distribution, as $\boldsymbol{\pi}$ is the left eigenvector corresponding to its largest eigenvalue 1, i.e., $\boldsymbol{\pi}\boldsymbol{M}_k^j = \boldsymbol{\pi}$. Left multiplying both sides of Eq. (7) with $\boldsymbol{\pi}$ gives $\boldsymbol{\pi}\boldsymbol{M}_k^j = \frac{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j}{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j}\boldsymbol{\pi}\xi_k^j \mathrm{diag}(\boldsymbol{\alpha}_k^j) + \boldsymbol{\pi} - \boldsymbol{\pi}\xi_k^j \mathrm{diag}(\boldsymbol{\alpha}_k^j) = \boldsymbol{\pi}$. In order to ensure that all the elements in the matrix $\boldsymbol{M}_k^j$ are nonnegative, we enforce that $\mathbf{I} - \xi_k^j \mathrm{diag}(\boldsymbol{\alpha}_k^j) \geq 0$ which results in the condition $\sup_k \xi_k^j \|\boldsymbol{\alpha}_k^j\|_\infty \leq 1$.

The off-diagonal term is given by $\boldsymbol{M}_k^j[i,\ell] = \frac{\xi_k^j}{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\pi}[\ell]\right)$ for all $i \neq \ell$. Clearly when $\xi_k^j > 0$; $\boldsymbol{M}_k^j[i,\ell] > 0$ if and only if $\boldsymbol{\pi}[\ell] > 0$. It follows from the definition of recurrent bins that $\boldsymbol{\pi}[\ell] > 0$ if and only if $R[\ell] \in \mathbf{\Pi}$.

The diagonal term is given by $\boldsymbol{M}_k^j[i,i] = \frac{\xi_k^j}{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[i]\boldsymbol{\pi}[i]\right) + \left(1 - \xi_k^j\boldsymbol{\alpha}_k^j[i]\right)$. The second term $\left(1 - \xi_k^j\boldsymbol{\alpha}_k^j[i]\right)$ is always nonnegative because of the condition $\sup_k \xi_k^j \|\boldsymbol{\alpha}_k^j\|_\infty \leq 1$. Hence $\boldsymbol{M}_k^j[i,i] > 0$ if $\boldsymbol{\pi}[i] > 0$, i.e., $R[i] \in \mathbf{\Pi}$. Finally, cases (iii) and (iv) follow from straight-forward substitution into (7). ■

Let us now discuss our choice of the tuning parameter ξ_k^j.

Remark 5. *(Hellinger Distance based Tuning Parameter ξ_k^j)* It is obvious from the property (iii) in Lemma 1 that the tuning parameter ξ_k^j should be based on the distance between the current swarm distribution $\boldsymbol{\mu}_k^j$ and the desired formation $\boldsymbol{\pi}$. For example, if $\boldsymbol{\mu}_k^j = \boldsymbol{\pi}$, then $\xi_k^j = 0$, $\boldsymbol{M}_k^j = \mathbf{I}$, and agents do not move from their current positions. In this paper, we define ξ_k^j to be the Hellinger distance (HD) between $\boldsymbol{\mu}_k^j$ and $\boldsymbol{\pi}$ as follows:

$$\xi_k^j = D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_k^j) := \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{n_{\text{cell}}}\left(\sqrt{\boldsymbol{\pi}[i]} - \sqrt{\boldsymbol{\mu}_k^j[i]}\right)^2}. \tag{8}$$

The HD is a symmetric measure of the difference between two probability distributions and it is upper bounded by 1.[38,39]

We choose the HD, over other popular metrics like $\mathcal{L}_1$ and $\mathcal{L}_2$ distances, because of an important property of HD shown in Fig. 2. The $\mathcal{L}_1$ distances for the cases ($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\mu}_3$) from $\boldsymbol{\pi}$ are the same in Fig. 2. But in Case 1, the wrong agent is in a bin where there should be no agent, hence the HD heavily penalizes this case. If all the agents are only in those bins which have non-zero weights in the desired distribution, then the HD is significantly less. Finally, if an agent is missing from a bin with fewer number of agents in the desired distribution (Case 2), then the HD penalizes it slightly more than the case where an agent is missing from a bin with higher number of agents in the desired distribution (Case 3). These important properties, which are encapsulated in HD, are useful for swarm guidance. □

In Remark 8, the $\boldsymbol{\alpha}_k^j$ vector that minimizes the coefficient of ergodicity of the IMC is found using a convex optimization-based approach.

Incorporating Motion Constraints

The Markov matrix $\boldsymbol{M}_k^j$ from Lemma 1 may not satisfy the motion constraints $\boldsymbol{A}_k^j$ in Definition 4, i.e., the Markov matrix might give non-zero transition probability to a transition that is not allowed by motion constraints. Therefore, in the following lemma, we modify the original Markov matrix $\boldsymbol{M}_k^j$ (7) to satisfy the motion constraints $\boldsymbol{A}_k^j$ (3).

Lemma 2. *(Construction of Markov Matrix that Satisfies Motion Constraints)* Let $\tilde{\boldsymbol{M}}_k^j$ represent the modified Markov matrix that satisfies the motion constraints $\boldsymbol{A}_k^j$, which is obtained from the original Markov matrix $\boldsymbol{M}_k^j$ (7) from Lemma 1. First, set $\tilde{\boldsymbol{M}}_k^j = \boldsymbol{M}_k^j$.

Then, for each transition that is not allowed by the motion constraint (i.e., $\boldsymbol{A}_k^j[i,\ell] = 0$), the corresponding transition probability in $\boldsymbol{M}_k^j$ (i.e., $\boldsymbol{M}_k^j[i,\ell]$) is added to the diagonal element in $\tilde{\boldsymbol{M}}_k^j$ (i.e., $\tilde{\boldsymbol{M}}_k^j[i,i]$). This is written as:

$$\tilde{\boldsymbol{M}}_k^j[i,i] = \boldsymbol{M}_k^j[i,i] + \sum_{\ell \in \left\{1,\ldots,n_{\text{cell}} : \boldsymbol{A}_k^j[i,\ell]=0\right\}} \boldsymbol{M}_k^j[i,\ell]\,. \tag{9}$$

Finally, set $\tilde{\boldsymbol{M}}_k^j[i,\ell] = 0$ if $\boldsymbol{A}_k^j[i,\ell] = 0$ for all $i,\ell \in \{1,\ldots,n_{\text{cell}}\}$. This resulting row stochastic Markov matrix $\tilde{\boldsymbol{M}}_k^j$ has $\boldsymbol{\pi}$ as its stationary distribution (i.e., $\boldsymbol{\pi}\tilde{\boldsymbol{M}}_k^j = \boldsymbol{\pi}$) and satisfies the motion constraints $\boldsymbol{A}_k^j$ (i.e., $\tilde{\boldsymbol{M}}_k^j[i,\ell] = 0$ if $\boldsymbol{A}_k^j[i,\ell] = 0$). This Markov matrix has the following properties:
(i) $\tilde{\boldsymbol{M}}_k^j$ only allows transitions into the recurrent bins $\boldsymbol{\Pi}$ which satisfy motion constraints $\boldsymbol{A}_k^j$, i.e., for $i \neq \ell$ and $\xi_k^j > 0$, $\tilde{\boldsymbol{M}}_k^j[i,\ell] > 0$ if and only if $R[\ell] \in \boldsymbol{\Pi}$ and $\boldsymbol{A}_k^j[i,\ell] = 1$,
(ii) the diagonal elements of recurrent bins are positive, i.e., $\tilde{\boldsymbol{M}}_k^j[i,i] > 0$ if $R[i] \in \boldsymbol{\Pi}$ and $\xi_k^j > 0$,
(iii) if $\xi_k^j = 0$, then $\tilde{\boldsymbol{M}}_k^j = \mathbf{I}$.

Proof: In order to show that $\tilde{\boldsymbol{M}}_k^j$ is row stochastic, let us right multiply the i^{th} row of $\tilde{\boldsymbol{M}}_k^j$ (i.e., $\tilde{\boldsymbol{M}}_k^j[i,:]$) with $\mathbf{1}$:

$$\tilde{\boldsymbol{M}}_k^j[i,:]\mathbf{1} = \tilde{\boldsymbol{M}}_k^j[i,i] + \sum_{\ell \in \left\{1,\ldots,n_{\text{cell}} : \boldsymbol{A}_k^j[i,\ell]=1\right\}} \boldsymbol{M}_k^j[i,\ell] = \boldsymbol{M}_k^j[i,:]\mathbf{1} = 1\,.$$

It follows from above that $\tilde{\boldsymbol{M}}_k^j\mathbf{1} = \boldsymbol{M}_k^j\mathbf{1} = \mathbf{1}$, hence $\tilde{\boldsymbol{M}}_k^j$ is row stochastic. Next, let us left multiply the i^{th} column of $\tilde{\boldsymbol{M}}_k^j$ (i.e., $\tilde{\boldsymbol{M}}_k^j[:,i]$) with $\boldsymbol{\pi}$:

$$\boldsymbol{\pi}\tilde{\boldsymbol{M}}_k^j[:,i] = \boldsymbol{\pi}[i]\tilde{\boldsymbol{M}}_k^j[i,i] + \sum_{\ell \in \left\{\begin{array}{c} 1,\ldots,n_{\text{cell}} : \\ \boldsymbol{A}_k^j[i,\ell]=1 \end{array}\right\}} \boldsymbol{\pi}[\ell]\boldsymbol{M}_k^j[\ell,i] = \sum_{\ell \in \{1,\ldots,n_{\text{cell}}\}} \boldsymbol{\pi}[\ell]\boldsymbol{M}_k^j[\ell,i] = \boldsymbol{\pi}\boldsymbol{M}_k^j[:,i] = \boldsymbol{\pi}[i]\,.$$

Here we used the reversible property of the Markov matrix $\boldsymbol{M}_k^j$, i.e. $\boldsymbol{\pi}[\ell]\boldsymbol{M}_k^j[\ell,i] = \boldsymbol{\pi}[i]\boldsymbol{M}_k^j[i,\ell]$ [40, pp. 14]. Hence the Markov matrix $\tilde{\boldsymbol{M}}_k^j$ indeed has $\boldsymbol{\pi}$ as its stationary distribution. The off-diagonal term of this Markov matrix is given by (i.e., for $i \neq \ell$):

$$\tilde{\boldsymbol{M}}_k^j[i,\ell] = \begin{cases} 0 & \text{if } \boldsymbol{A}_k^j[i,\ell] = 0 \\ \frac{\xi_k^j}{\boldsymbol{\pi}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\pi}[\ell]\right) & \text{otherwise} \end{cases}\,.$$

Clearly, if a transition is not allowed by motion constraints (i.e., $\boldsymbol{A}_k^j[i,\ell] = 0$), then the transition probability is zero (i.e., $\tilde{\boldsymbol{M}}_k^j[i,\ell] = 0$). If a transition is allowed by motion constraints (i.e., $\boldsymbol{A}_k^j[i,\ell] = 1$) and $\xi_k^j > 0$, then the transition probability $\tilde{\boldsymbol{M}}_k^j[i,\ell] > 0$ if and only if $\boldsymbol{\pi}[\ell] > 0$, which implies that $R[\ell] \in \boldsymbol{\Pi}$.

Since $\tilde{\boldsymbol{M}}_k^j[i,i] \geq \boldsymbol{M}_k^j[i,i]$, cases (ii) and (iii) follow from Lemma 1. ■

$$\tilde{\boldsymbol{M}}_k^j = \begin{bmatrix} \overbrace{\begin{bmatrix} \tilde{\boldsymbol{M}}_k^j[1,1] & \cdots & \tilde{\boldsymbol{M}}_k^j[1,n_{\text{rec}}] \\ \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{M}}_k^j[n_{\text{rec}},1] & \cdots & \tilde{\boldsymbol{M}}_k^j[n_{\text{rec}},n_{\text{rec}}] \end{bmatrix}}^{\tilde{\boldsymbol{M}}_{k,\text{sub}}^j \in \mathbb{R}^{n_{\text{rec}} \times n_{\text{rec}}}} & \boldsymbol{0} \in \mathbb{R}^{n_{\text{rec}} \times (n_{\text{cell}} - n_{\text{rec}})} \\ \begin{bmatrix} \tilde{\boldsymbol{M}}_k^j[n_{\text{rec}}+1,1] & \cdots & \tilde{\boldsymbol{M}}_k^j[n_{\text{rec}}+1,n_{\text{rec}}] \\ \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{M}}_k^j[n_{\text{cell}},1] & \cdots & \tilde{\boldsymbol{M}}_k^j[n_{\text{cell}},n_{\text{rec}}] \end{bmatrix} & \begin{bmatrix} \tilde{\boldsymbol{M}}_k^j[n_{\text{rec}}+1,n_{\text{rec}}+1] & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tilde{\boldsymbol{M}}_k^j[n_{\text{cell}},n_{\text{cell}}] \end{bmatrix} \end{bmatrix}$$

Figure 3. Decomposition of the Markov matrix $\tilde{M}_k^j$, where the sub-matrix $\tilde{M}_{k,\text{sub}}^j$ encapsulates the bin transition probabilities between recurrent bins.

For a bin $R[i]$, let us define $\mathcal{A}_k^j(R[i])$ as the set of all bins that the j^{th} agent can transition to at the k^{th} time instant:

$$\mathcal{A}_k^j(R[i]) := \left\{ R[\ell] : \ell \in \{1, \dots, n_{\text{cell}}\};\ \boldsymbol{A}_k^j[i,\ell] = 1 \right\}. \tag{10}$$

Let us now discuss some additional conditions that arise due to the motion constraints, like this trapping problem.

Remark 6. *(Trapping Problem and its Solution)* If the j^{th} agent is actually located in bin $R[i]$ and we observe that $\mathcal{A}_k^j(R[i]) \cap \boldsymbol{\Pi} = \emptyset$ for all $k \in \mathbb{Z}^*$, then the j^{th} agent is trapped in the bin $R[i]$ forever. Let us define $\mathcal{T}_k^j$ as the set of all bins that satisfy this trapping condition:

$$\mathcal{T}_k^j := \bigcup_{i \in \{1,\dots,n_{\text{cell}}\}} \left\{ R[i] \ : \ \mathcal{A}_k^j(R[i]) \cap \boldsymbol{\Pi} = \emptyset \right\}. \tag{11}$$

To avoid this trapping problem, we enforce a secondary condition on the j^{th} agent if it is actually located in a bin belonging to the set $\mathcal{T}_k^j$. For each bin $R[i] \in \mathcal{T}_k^j$, we chose another bin $\Psi_k^j(R[i])$; where $\Psi_k^j(R[i]) \in \mathcal{A}_k^j(R[i])$ and either $\Psi_k^j(R[i]) \notin \mathcal{T}_{k+1}^j$ or $\Psi_k^j(R[i])$ is close to other bins which are not in $\mathcal{T}_{k+1}^j$. The secondary condition on the j^{th} agent is that it will transition to bin $\Psi_k^j(R[i])$ during the k^{th} time step. Moreover, the agents in the transient bins, which are not in the trapping region, will eventually transition to the recurrent bins. We can also speed up this process. The information about exiting the trapping region and the transient bins is captured by the matrix $\boldsymbol{C}_k^j \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}$ given by:

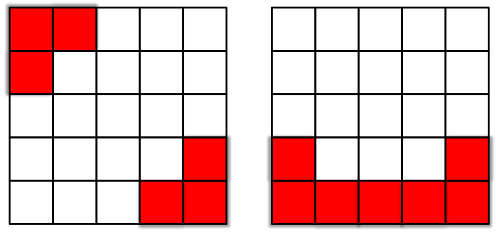
$$\boldsymbol{C}_k^j[i,\ell] = \begin{cases} 1 & \textbf{if } R[i] \in \mathcal{T}_k^j \text{ and } R[\ell] = \Psi_k^j(R[i]) \\ \frac{1}{\left|\mathcal{A}_k^j(R[i]) \cap \boldsymbol{\Pi}\right|} & \textbf{elseif } R[i] \notin \mathcal{T}_k^j,\ R[i] \notin \boldsymbol{\Pi}, \text{ and } R[\ell] \in \mathcal{A}_k^j(R[i]) \cap \boldsymbol{\Pi} \\ 0 & \textbf{otherwise} \end{cases}, \tag{12}$$

which is used instead of the Markov matrix $\tilde{\boldsymbol{M}}_k^j$ in (6). Note that this secondary condition would not cause an infinite loop as the graph conforming to the $\boldsymbol{A}_k^j$ matrix is strongly connected. □

Without loss of generality, the bins are relabeled such that the first n_{rec} bins are recurrent bins and the remaining bins are transient bins. The Markov matrix $\tilde{\boldsymbol{M}}_k^j$ can be decomposed as shown in Fig. 3, where the sub-matrix $\tilde{\boldsymbol{M}}_{k,\text{sub}}^j$ encapsulates the bin transition probabilities between recurrent bins.

In some cases, it is possible that the set $\boldsymbol{\Pi}$ can be decomposed into disjoint subsets, such that any agent cannot transition from one subset of $\boldsymbol{\Pi}$ to another subset of $\boldsymbol{\Pi}$ without exiting the set $\boldsymbol{\Pi}$ due to motion constraints (e.g., see Fig. 4). Since our proposed algorithm suggests that the agents will only transition within the set of recurrent bins $\boldsymbol{\Pi}$ after entering any recurrent bin, the agents in one such subset of $\boldsymbol{\Pi}$ will never transition to the other subsets of $\boldsymbol{\Pi}$. Hence, the proposed algorithms will not be able to achieve the desired formation, as the agents in each subset of $\boldsymbol{\Pi}$ are trapped within that subset. In order to avoid such situations, we need the following assumption on $\boldsymbol{A}_k^j$.

Assumption 1. The motion constraints matrix $\boldsymbol{A}_k^j$ should be such that each agent can transition from any recurrent bin to any other recurrent bin without exiting the set $\boldsymbol{\Pi}$, while satisfying the motion constraints. Let the sub-matrix $\boldsymbol{A}_{k,\text{sub}}^j$ encapsulate the motion constraints between recurrent bins (similar to $\tilde{\boldsymbol{M}}_{k,\text{sub}}^j$ shown in



(a) Not Allowed (b) Allowed

Figure 4. The red region denotes the set Π. The motion constraints are such that the agent in a given bin can only transition to its contiguous bins. In case (a), Π can be decomposed into two subsets at the two corners, such that any agent cannot transition from one subset to another subset without exiting the set Π. In case (b), such a decomposition is not possible. According to Assumption 1, case (a) is not allowed and case (b) is allowed.

Fig. 3). Therefore, the set of recurrent bins $\mathbf{\Pi}$ is strongly connected in the graph conforming to $\boldsymbol{A}^j_{k,\text{sub}}$. This ensures that the matrices $\tilde{\boldsymbol{M}}^j_k$ and $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}$ have a single essential class [41, pp. 12], which is the set $\mathbf{\Pi}$. $\square$

Definition 7. *(Maximum Graph Diameter n_{dia})* Assumption 1 implies that the graph diameter (i.e., longest shortest path) between any two bins in $\mathbf{\Pi}$ in the graph conforming to $\boldsymbol{A}^j_{k,\text{sub}}$ is bounded by n_{rec} for each time instant. Let us define n_{dia} as the maximum graph diameter over all time instants, which is also bounded by n_{rec}. This implies that the matrix $(\tilde{\boldsymbol{M}}^j_{k,\text{sub}})^{n_{\text{dia}}}$ is a positive matrix for all time instants. $\square$

Remark 8. *(Minimum Coefficient of Ergodicity based $\boldsymbol{\alpha}^j_k$ vector)* Here we find the $\boldsymbol{\alpha}^j_k$ vector that optimizes the coefficient of ergodicity of the IMC [41, pp. 137]. Let $\alpha_{\min}$ be some positive constant (i.e., $0 < \alpha_{\min} < 1$). The $\boldsymbol{\alpha}^j_k$ vector is the solution of the following optimization problem:

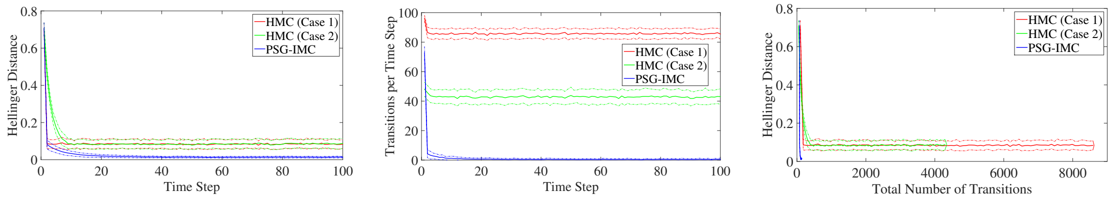
$$\min_{\boldsymbol{\alpha}^j_k} \tau_1\left((\tilde{\boldsymbol{M}}^j_{k,\text{sub}})^{n_{\text{dia}}}\right) ,$$
$$\text{subject to} \quad \alpha_{\min} \le \boldsymbol{\alpha}^j_k[i] \le 1,\ \forall i \in \{1,\dots,n_{\text{cell}}\} . \tag{13}$$

Here $\tau_1(\boldsymbol{P})$ is the proper coefficient of ergodicity defined by $\tau_1(\boldsymbol{P}) := 1 - \min_{i,\ell} \sum_{s=1}^{n_P} \min(\boldsymbol{P}[i,s],\ \boldsymbol{P}[\ell,s])$. Let us first discuss the rationale behind this nonlinear optimization problem. We are interested in optimizing the mixing rate within the set of recurrent bins. If the graph diameter of the graph conforming to $\boldsymbol{A}^j_{k,\text{sub}}$ is greater than two, then there exists bins $i,\ell \in \mathbf{\Pi}$ such that either $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}[i,s] = 0$ or $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}[\ell,s] = 0$ for all $s \in \{1,\dots,n_{\text{rec}}\}$, which implies that $\tau_1(\tilde{\boldsymbol{M}}^j_{k,\text{sub}}) = 1$. In order to avoid this trivial case, we choose to minimize the proper coefficient of ergodicity of the positive matrix $(\tilde{\boldsymbol{M}}^j_{k,\text{sub}})^{n_{\text{dia}}}$. Here, we have retained the structure of the original Markov matrix (7) so that we can take advantage of the properties discussed in Lemmas 1 and 2.

We now show that the nonlinear optimization problem (13) can be efficiently solved using convex optimization. We first show that $\tau_1(\boldsymbol{P})$ is a convex function of the stochastic matrix $\boldsymbol{P}$. We can express $\tau_1(\boldsymbol{P})$ as [41, Lemma 4.3, pp. 139]:

$$\tau_1(\boldsymbol{P}) = \sup_{\|\boldsymbol{\delta}\|_2=1,\ \boldsymbol{\delta}\mathbf{1}=0} \|\boldsymbol{\delta}\boldsymbol{P}\|_1 ,$$

where $\boldsymbol{\delta} \in \mathbb{R}^{n_{\text{cell}}}$ is a row vector. Since $\boldsymbol{\delta}\boldsymbol{P}$ is a linear function and $\|\cdot\|_1$ is a convex function, therefore $\tau_1(\boldsymbol{P})$ is the pointwise supremum of a family of convex functions of $\boldsymbol{P}$, and so it is a convex function of $\boldsymbol{P}$. Hence the nonlinear optimization problem (13) is evidently a convex optimization problem because the constraints are all linear inequalities and the objective function is convex. Therefore standard convex optimization tools can be used to solve this problem efficiently (e.g., interior-point method[42]). $\square$



(a) (b) (c)

Figure 5. In this example, we compare the PSG–IMC algorithm with the HMC-based approaches, where 100 agents start from a random distribution and converge to the desired distribution of the letter "I" shown in Fig. 1(b). The cumulative results, along with 1σ errorbars, of 100 Monte Carlo simulations are shown here for the PSG–IMC algorithm and the two HMC approaches: (Case 1) $M_1 = \mathbf{1}\boldsymbol{\pi}$ and (Case 2) $M_2 = 0.5(\mathbf{1}\boldsymbol{\pi} + \mathbf{I})$. The results show that the PSG–IMC algorithm guarantees much better convergence with significantly smaller number of transitions compared to the HMC algorithms.

PSG–IMC Algorithm for Pattern Formation

The pseudo-code for the PSG–IMC algorithm for pattern formation is given in **Method 1**. The agent first determines its current position (e.g. the j^{th} agent is in bin $R[i]$, therefore $\boldsymbol{r}_k^j[i] = 1$) and the current swarm distribution $\boldsymbol{\mu}_k^j$ (lines 2–3). The agent then checks if it is in a transient or recurrent bin (line 4). If the agent is in a transient bin, then it computes the matrix $\boldsymbol{C}_k^j$ to move closer or into the recurrent bins (line 4). If the agent is in a recurrent bin, then it checks if the number of agents in its current bin is less than (or equal to) the desired number of agents in that bin (line 5). In this case the agent continues to remain in its current bin (line 5). Else, the agent computes the modified Markov matrix $\tilde{\boldsymbol{M}}_k^j$ to transition to other recurrent bins (lines 6–9). Hence the new time evolution of the pmf vector $\boldsymbol{x}_k^j$ can be written as:

$$\boldsymbol{x}_{k+1}^j = \boldsymbol{x}_k^j \boldsymbol{B}_k^j, \text{ where } \boldsymbol{B}_k^j = \begin{cases} \boldsymbol{C}_k^j & \textbf{if } R[i] \notin \boldsymbol{\Pi} \\ \mathbf{I} & \textbf{elseif } \boldsymbol{\mu}_k^j[i] \leq \boldsymbol{\pi}[i] \\ \tilde{\boldsymbol{M}}_k^j & \textbf{otherwise} \end{cases}, \tag{14}$$

where the matrix $\boldsymbol{C}_k^j$ is given by (12) and the Markov matrix $\tilde{\boldsymbol{M}}_k^j$ is given by Lemma 2. Finally, the agent uses inverse transform sampling[24] to select the next bin from the bin transition probabilities (lines 10–11).

Method 1: PSG–IMC algorithm for pattern formation

1:	(one iteration of j^{th} agent during k^{th} time instant)	
2:	Given agent's current position $\boldsymbol{r}_k^j$, e.g. $\boldsymbol{r}_k^j[i] = 1$	
3:	Given feedback of current swarm distribution $\boldsymbol{\mu}_k^j$	
4:	**if** $R[i] \notin \boldsymbol{\Pi}$, **then** Compute $\boldsymbol{C}_k^j$, Set $\boldsymbol{B}_k^j = \boldsymbol{C}_k^j$	} Eq. (12)
5:	**elseif** $\boldsymbol{\mu}_k^j[i] \leq \boldsymbol{\pi}[i]$, **then** Set $\boldsymbol{B}_k^j = \mathbf{I}$	
6:	**else** Compute the tuning parameter ξ_k^j	} Eq. (8)
7:	Compute Markov matrices $\boldsymbol{M}_k^j$ and $\tilde{\boldsymbol{M}}_k^j$	} Lemmas 1 and 2
8:	Compute the $\boldsymbol{\alpha}_k^j$ vector, Set $\boldsymbol{B}_k^j = \tilde{\boldsymbol{M}}_k^j$	} Eq. (13)
9:	**end if**	
10:	Generate a random number $z \in \text{unif}[0;1]$	}Inverse Transform Sampling
11:	Go to bin $R[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{B}_k^j[i,\ell] \leq z < \sum_{\ell=1}^{q} \boldsymbol{B}_k^j[i,\ell]$	

Let us now compare this PSG–IMC algorithm with existing HMC-based approaches for pattern formation.

Remark 9. *(Comparison of PSG–IMC with HMC-based approaches)* For this example, 100 agents start from a random distribution and try to achieve the desired formation $\boldsymbol{\pi}$ shown in Fig. 1(b), in the absence of motion constraints (i.e., $\boldsymbol{A}_k^j = \mathbf{1}\mathbf{1}^T$). The trivial Markov matrix that has $\boldsymbol{\pi}$ as its stationary distribution is given by

$\boldsymbol{M}_1 = \mathbf{1}\boldsymbol{\pi}$. Another such Markov matrix, obtained using property (iv) in Lemma 1, is $\boldsymbol{M}_2 = 0.5(\mathbf{1}\boldsymbol{\pi} + \mathbf{I})$. In the HMC-based approach, the same Markov matrix is used by all the agents for all time steps. In this example, we compare the performance of two HMCs $\boldsymbol{M}_1$ (Case 1) and $\boldsymbol{M}_2$ (Case 2) with the PSG–IMC algorithm. Fig. 5 shows the cumulative results for 100 Monte Carlo simulations.

In Fig. 5(a), the variation of HD with time step shows that the PSG–IMC algorithm converges to a HD (≈ 0.014) that is a significant improvement (≈ 6 times better) over that of the HMC cases (≈ 0.08). In Fig. 5(b), the variation of number of transitions with time step shows that the agents executing the PSG–IMC algorithm only perform transitions when the HD is large and perform very little transitions when the HD is small. This is because the feedback of the current swarm distribution allows to PSG–IMC algorithm to increase or decrease the rate of transitions depending on the HD. On the other hand, the agents executing the HMC algorithms continue transitioning at the same rate for all time step. In Fig. 5(c), the total number of transitions up to a given time step is plotted against the corresponding HD at that time step. This figure shows that not only does the PSG–IMC algorithm converge to a much smaller HD, but also the total number of transitions in the PSG–IMC algorithm (≈ 130) is significantly less ($\approx 33 - 66$ times reduction) than that of HMC Case 1 (≈ 8600) and HMC Case 2 (≈ 4300). Hence, the PSG–IMC algorithm guarantees much better convergence with significantly smaller number of transitions compared to the HMC algorithms.

The key reasons behind the superior performance of the PSG–IMC algorithm over HMC-based algorithms are:
(i) In PSG–IMC, the agents transition out of a recurrent bin if and only if there is an excess of agents in that bin. Agents do not transition out of a bin if there is a deficiency of agents in that bin. This is checked by the condition $\boldsymbol{\mu}_k^j[i] \leq \boldsymbol{\pi}[i]$ in Eq. (14).
(ii) The number of transitions in PSG–IMC is proportional to the HD. This helps in achieving faster convergence (when HD is large) while avoiding unnecessary transitions (when HD is small). This also ensures that the agents settle down after the desired formation is achieved.
These two key reasons that depend on the feedback cannot be incorporated into HMC-based algorithms. □

The convergence analysis of the PSG–IMC algorithm is given in the next section. Let us now discuss the robustness properties of this algorithm.

Remark 10. *(Robustness of PSG–IMC)* The PSG–IMC algorithm satisfies the Markov property, because the action of each agent depends only on its present location and the current swarm distribution. This memoryless property ensures that all the agents re-start their guidance trajectory from their present location during every time instant. Thus the swam continues to achieve its objective even if agents are added or removed from the swarm or the swarm is damaged by external disturbances or some agents have not reached their target bin during the previous time instant.

Even if there is a small error in the agent's estimate of the current swarm distribution (i.e., $\|\boldsymbol{\mu}_k^j - \boldsymbol{\mu}_k^\star\|_2 \leq \varepsilon_{\text{cons}}$), then its effect on the HD is also small (i.e., $|D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_k^j) - D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_k^\star)| \leq \frac{\varepsilon_{\text{cons}}}{\sqrt{2}}$). If we ensure that $\varepsilon_{\text{cons}}$ is sufficiently small (e.g., $\varepsilon_{\text{cons}} < \frac{1}{m_k}$), then its effect on the HD is negligible. □

CONVERGENCE ANALYSIS OF PSG–IMC ALGORITHM FOR PATTERN FORMATION

In this section, we discuss the convergence analysis of the PSG–IMC algorithm for pattern formation given in **Method 1**. In Theorem 3, we show that all the agents leave the transient bins and enter the recurrent bins $\boldsymbol{\Pi}$ in finite time. In Theorem 4, we show that each agent's pmf vector $\boldsymbol{x}_k^j$ converges to the desired distribution $\boldsymbol{\pi}$ and the geometric rate of convergence if given in Theorem 5. In Theorem 6, we provide a lower-bound for the minimum number of agents needed to satisfy the given convergence error threshold between the current swarm distribution $\boldsymbol{\mu}_k^\star$ and the desired distribution $\boldsymbol{\pi}$.

The overall time evolution of the pmf vector $\boldsymbol{x}_k^j$, defined in Eq. (4), from an initial condition $\boldsymbol{x}_0^j$ to the k^{th} time instant is given by the inhomogeneous Markov chain:

$$\boldsymbol{x}_k^j = \boldsymbol{x}_0^j \boldsymbol{V}_{0,k}^j, \quad \text{for all } k \in \mathbb{N}, \tag{15}$$

where the forward matrix product $\boldsymbol{V}_{0,k}^j := \boldsymbol{B}_0^j \boldsymbol{B}_1^j \dots \boldsymbol{B}_{k-1}^j$, and each $\boldsymbol{B}_k^j \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}$ is given by Eq. (14).

Theorem 3. Each agent enters the set of recurrent bins $\mathbf{\Pi}$ in finite time. Once an agent is inside a recurrent bin, it always remains within the set of recurrent bins $\mathbf{\Pi}$.

Proof: If the j^{th} agent at the k^{th} time-instant is in a recurrent bin $R[i] \in \mathbf{\Pi}$, then the matrix $\boldsymbol{B}_k^j$ in Eq. (14) is either $\tilde{\boldsymbol{M}}_k^j$ and $\mathbf{I}$. It follows from Lemma 2 that the j^{th} agent either remains in bin $R[i] \in \mathbf{\Pi}$ or transitions to any other recurrent bin $R[\ell] \in \mathbf{\Pi}$, hence it cannot transition to any transient bin. Therefore, once an agent is inside a recurrent bin, it always remains within the set $\mathbf{\Pi}$.

If the j^{th} agent at the k^{th} time-instant is in a transient bin $R[i] \notin \mathbf{\Pi}$, then the matrix $\boldsymbol{B}_k^j$ in Eq. (14) is given by $\boldsymbol{C}_k^j$ (12). If the bin $R[i]$ is not in the trapping region (i.e., $R[i] \notin \mathbf{\Pi}$ and $R[i] \notin \boldsymbol{\mathcal{T}}_k^j$), then the j^{th} agent will enter a recurrent bin in the next time-instant and subsequently continue to remain within the set $\mathbf{\Pi}$. If the bin $R[i]$ is in the trapping region (i.e., $R[i] \notin \mathbf{\Pi}$ and $R[i] \in \boldsymbol{\mathcal{T}}_k^j$), then $\Psi_k^j(R[i])$ is chosen such that the agent exists the trapping region as soon as possible. This ensures that the number of steps in the trapping region is finite and upper bounded by $(n_{\text{cell}} - n_{\text{rec}})$. Hence each agent enters the set $\mathbf{\Pi}$ in atmost $(n_{\text{cell}} - n_{\text{rec}} + 1)$ time steps. ■

In the next two theorems, we discuss the convergence of each agent's pmf vector to the desired formation and then bound the rate of convergence.

Theorem 4. *(Convergence of Inhomogeneous Markov Chains)* Each agent's time evolution of the pmf vector $\boldsymbol{x}_k^j$, from any initial condition $\boldsymbol{x}_0^j \in \mathbb{R}^{n_{\text{cell}}}$, is given by the inhomogeneous Markov chain (15). If each agent executes the PSG–IMC algorithm, then $\boldsymbol{x}_k^j$ converges pointwise to the desired stationary distribution $\boldsymbol{\pi}$, i.e., $\lim_{k\to\infty} \boldsymbol{x}_k^j = \boldsymbol{\pi}$ pointwise for all $j \in \{1,\ldots,m\}$.

Proof: It follows from Theorem 3 that there exists a time instant $T \leq (n_{\text{cell}} - n_{\text{rec}} + 1)$, such that all agents are always in the set $\mathbf{\Pi}$ from time instant T onwards. Therefore,

$$\boldsymbol{x}_T^j = \boldsymbol{x}_0^j \boldsymbol{V}_{0,T}^j, \quad \text{where } \boldsymbol{x}_T^j[i] = \begin{cases} \geq 0 & \text{if } R[i] \in \mathbf{\Pi} \\ 0 & \text{otherwise} \end{cases},$$

$$\boldsymbol{x}_{k+1}^j = \begin{cases} \boldsymbol{x}_k^j & \text{if } \boldsymbol{\mu}_k^j[i] \leq \boldsymbol{\pi}[i] \\ \boldsymbol{x}_k^j \tilde{\boldsymbol{M}}_k^j & \text{otherwise} \end{cases}, \quad \text{for all } k \geq T\,.$$

Without loss of generality, the bins are relabeled such that the first n_{rec} bins are recurrent bins and the remaining bins are transient bins. Therefore, the pmf vector $\boldsymbol{x}_T^j$ and the desired formation $\boldsymbol{\pi}$ can be decomposed as follows:

$$\boldsymbol{x}_T^j = \Big[\ \underbrace{\boldsymbol{x}_T^j[1],\ldots \boldsymbol{x}_T^j[n_{\text{rec}}]}_{\bar{\boldsymbol{x}}_T^j} \quad 0,\ldots,0\ \Big], \qquad \boldsymbol{\pi} = \Big[\ \underbrace{\boldsymbol{\pi}[1],\ldots,\boldsymbol{\pi}[n_{\text{rec}}]}_{\bar{\boldsymbol{\pi}}} \quad 0,\ldots,0\ \Big],$$

where the probability row vectors $\bar{\boldsymbol{x}}_T^j \in \mathbb{R}^{n_{\text{rec}}}$ and $\bar{\boldsymbol{\pi}} \in \mathbb{R}^{n_{\text{rec}}}$ denote the agent's pmf vector and the desired formation over the set of recurrent bins $\mathbf{\Pi}$ respectively. Note that convergence of $\bar{\boldsymbol{x}}_k^j$ to $\bar{\boldsymbol{\pi}}$, implies the convergences of $\boldsymbol{x}_k^j$ to $\boldsymbol{\pi}$. The Markov matrix $\tilde{\boldsymbol{M}}_k^j$ is also decomposed as shown in Fig. 3, where the submatrix $\tilde{\boldsymbol{M}}_{k,\text{sub}}^j$ encapsulates the bin transition probabilities between recurrent bins. The time evolution of the agent's pmf vector over the set $\mathbf{\Pi}$ is given by:

$$\bar{\boldsymbol{x}}_{k+1}^j = \begin{cases} \bar{\boldsymbol{x}}_k^j & \text{if } \boldsymbol{\mu}_k^j[i] \leq \boldsymbol{\pi}[i] \\ \bar{\boldsymbol{x}}_k^j \tilde{\boldsymbol{M}}_{k,\text{sub}}^j & \text{otherwise} \end{cases}, \quad \text{for all } k \geq T\,. \tag{16}$$

If the current swarm distribution is equal to the desired formation, then $\xi_k^j = 0$ and the PSG–IMC algorithm tells each agent to maintain their present location as no further convergence is necessary. Since this is a special case in a stochastic processes, henceforth we assume that the swarm has not converged and $\xi_k^j > 0$.

Each Markov matrix $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}$ is row stochastic (i.e., $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}\mathbf{1} = \mathbf{1}$) and has $\bar{\boldsymbol{\pi}}$ as it stationary distribution (i.e., $\bar{\boldsymbol{\pi}}\tilde{\boldsymbol{M}}^j_{k,\text{sub}} = \bar{\boldsymbol{\pi}}$). Hence, the sequence of matrices $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}$, $k \geq T$ is asymptotically homogeneous with respect to $\bar{\boldsymbol{\pi}}$ because $\bar{\boldsymbol{\pi}}\tilde{\boldsymbol{M}}^j_{k,\text{sub}} = \bar{\boldsymbol{\pi}}$ for all $k \geq T$ [41, pp. 92, 149]. Moreover, each $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}$ is irreducible because the corresponding motion constraint sub-matrix $\boldsymbol{A}^j_{k,\text{sub}}$ for recurrent bins is strongly connected (see Assumption 1).

Let η^j_k represent the probability of the event that the j^{th} agent is in a bin with excess agents at the k^{th} time instant:

$$\eta^j_k := \mathbb{P}\left(\cup_{i=1}^{n_{\text{rec}}}(\boldsymbol{r}^j_k[i] = 1 \text{ and } \boldsymbol{\mu}^j_k[i] > \boldsymbol{\pi}[i])\right) .$$

If the swarm has not converged, then $\frac{1}{n_{\text{rec}}} \leq \eta^j_k \leq 1 - \frac{1}{n_{\text{rec}}}$ because at least one bin has excess agents and at least one bin has deficiency in agents. The time evolution of the agent's pmf vector in Eq. (16) can be written as:

$$\bar{\boldsymbol{x}}^j_{k+1} = \bar{\boldsymbol{x}}^j_k \underbrace{\left(\eta^j_k\tilde{\boldsymbol{M}}^j_{k,\text{sub}} + (1-\eta^j_k)\mathbf{I}\right)}_{\boldsymbol{P}^j_k}, \quad \text{for all } k \geq T ,$$

where $\boldsymbol{P}^j_k := \left(\eta^j_k\tilde{\boldsymbol{M}}^j_{k,\text{sub}} + (1-\eta^j_k)\mathbf{I}\right)$. This Markov matrix $\boldsymbol{P}^j_k$ is row stochastic (i.e., $\boldsymbol{P}^j_k\mathbf{1} = \mathbf{1}$), has $\bar{\boldsymbol{\pi}}$ as it stationary distribution (i.e., $\bar{\boldsymbol{\pi}}\boldsymbol{P}^j_k = \bar{\boldsymbol{\pi}}$), is irreducible (because $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}$ is irreducible), and the sequence of matrices $\boldsymbol{P}^j_k$, $k \geq T$ is asymptotically homogeneous with respect to $\bar{\boldsymbol{\pi}}$. The overall time evolution of the agent's pmf vector is given by:

$$\bar{\boldsymbol{x}}^j_k = \bar{\boldsymbol{x}}^j_T\boldsymbol{U}^j_{T,k}, \quad \text{for all } k > T , \tag{17}$$

where the forward matrix product $\boldsymbol{U}^j_{T,k} := \boldsymbol{P}^j_T\boldsymbol{P}^j_{T+1}\dots\boldsymbol{P}^j_{k-2}\boldsymbol{P}^j_{k-1}$. We now show that this matrix product $\boldsymbol{U}^j_{T,k}$ is strongly ergodic and $\bar{\boldsymbol{\pi}}$ is its unique limit.

The matrix $\boldsymbol{U}^j_{T,k}$ is a product of nonnegative irreducible matrices, hence it is also a nonnegative and irreducible matrix. Moreover, each diagonal element $\boldsymbol{U}^j_{T,k}[i,i] \geq \prod_{s=T}^{k-1}\left(\eta^j_s\tilde{\boldsymbol{M}}^j_{s,\text{sub}}[i,i] + (1-\eta^j_s)\right)$, where each $\tilde{\boldsymbol{M}}^j_{s,\text{sub}}[i,i] > 0$ (see Lemma 2). Therefore $\boldsymbol{U}^j_{T,k}$ is a primitive matrix [43, pp. 678].

A number of cross-diagonal elements in $\tilde{\boldsymbol{M}}^j_{k,\text{sub}}$ and $\boldsymbol{P}^j_k$ are equal to zero because those transitions are not allowed by motion constraints. We now find a lower-bound (γ) for the positive elements in $\boldsymbol{P}^j_k$, where the transitions are allowed by motion constraints. If at least one agent is not in the correct location, then $\sum_{R[i]\in\mathcal{R}}|\boldsymbol{\mu}^\star_k[i] - \boldsymbol{\pi}[i]| \geq \frac{2}{m_k}$. Due to the quantization of the pmf by the number of agents m, the smallest positive tuning parameter $\xi_{\min}$ is given by $\xi_{\min} = \frac{1}{2^{\frac{3}{2}}\max_{k\in\mathbb{Z}^*} m_k} \leq \min^+_{j\in\{1,\dots,m\},\, k\in\mathbb{Z}^*}\xi^j_k$. The smallest positive element in the $\boldsymbol{\alpha}^j_k$ vector in Remark 8 is $\alpha_{\min}$. Finally, the smallest positive element in $\bar{\boldsymbol{\pi}}$ is given by $\bar{\pi}_{\min} = \left(\min_{i\in\{1,\dots,n_{\text{rec}}\}}\bar{\boldsymbol{\pi}}[i]\right)$ and $\max(\bar{\boldsymbol{\pi}}\boldsymbol{\alpha}^j_k) = 1$. Hence the lower-bound of the positive element in $\boldsymbol{P}^j_k$ is given by:

$$0 < \gamma = \frac{\xi_{\min}\alpha^2_{\min}\bar{\pi}_{\min}}{n_{\text{rec}}} \leq \min_{i,\ell}{}^+\boldsymbol{P}^j_k[i,\ell] . \tag{18}$$

Therefore, it follows from Theorem 4.15 [41, pp. 150] that the forward matrix product $\boldsymbol{U}^j_{T,k}$ is strongly ergodic. Since all the recurrent bins are strongly connected in all $\boldsymbol{P}^j_k$, there is a single essential class of indices. Therefore, it follows from Theorem 4.12 [41, pp. 149] that the limit vector of $\boldsymbol{U}^j_{T,k}$ is $\bar{\boldsymbol{\pi}}$. Hence, each agent's pmf vector converges to:

$$\lim_{k\to\infty}\bar{\boldsymbol{x}}^j_k = \lim_{k\to\infty}\bar{\boldsymbol{x}}^j_T\boldsymbol{U}^j_{T,k} = \bar{\boldsymbol{x}}^j_T\mathbf{1}\bar{\boldsymbol{\pi}} = \bar{\boldsymbol{\pi}} ,$$

which implies that $\lim_{k\to\infty}\boldsymbol{x}^j_k = \boldsymbol{\pi}$ pointwise for all agents. ∎

Theorem 5. *(Geometric Convergence Rate)* After every n_{dia} time instants, each agent's pmf vector geometrically converges to $\boldsymbol{\pi}$ with a rate greater than $(1 - n_{\text{rec}}\gamma^{n_{\text{dia}}})$, i.e.,

$$\|\boldsymbol{x}^j_{k+n_{\text{dia}}} - \boldsymbol{\pi}\|_2 \leq (1 - n_{\text{rec}}\gamma^{n_{\text{dia}}})\|\boldsymbol{x}^j_k - \boldsymbol{\pi}\|_2\,,\ \forall k \geq T\,,$$

and γ is computed in Eq. (18).

Proof: Here we reuse the notations in the proof of Theorem 4. It follows from Eq. (17) that $\bar{\boldsymbol{x}}^j_{k+n_{\text{dia}}} = \bar{\boldsymbol{x}}^j_k \boldsymbol{U}^j_{k,k+n_{\text{dia}}}$ for all $k \geq T$. Furthermore, it follows from Definition 7 that the matrix $\boldsymbol{U}^j_{k,k+n_{\text{dia}}}$ is a positive matrix. A conservative lower-bound on the off-diagonal terms in the matrix $\boldsymbol{U}^j_{k,k+n_{\text{dia}}}$ is $\gamma^{n_{\text{dia}}}$. Therefore, the upper-bound on the proper coefficient of ergodicity (see Remark 8) is given by $\tau_1(\boldsymbol{U}^j_{k,k+n_{\text{dia}}}) \leq 1 - n_{\text{rec}}\gamma^{n_{\text{dia}}}$. Hence the geometric convergence rate after n_{dia} time instants is $(1 - n_{\text{rec}}\gamma^{n_{\text{dia}}})$, i.e., $\|\bar{\boldsymbol{x}}^j_{k+n_{\text{dia}}} - \bar{\boldsymbol{\pi}}\|_2 \leq (1 - n_{\text{rec}}\gamma^{n_{\text{dia}}})\|\bar{\boldsymbol{x}}^j_k - \bar{\boldsymbol{\pi}}\|_2$, which implies the desired result. ■

Note that this geometric rate of convergence is extremely conservative. The actual convergence rate of the IMC can be improved by minimizing the coefficient of ergodicity, as discussed in Remark 8.

We now focus on the convergence of the current swarm distribution to the desired formation. In practical scenarios, the number of agents is finite, hence the following theorem gives a lower-bound on the number of agents for satisfying the given convergence error thresholds.

Theorem 6. For some acceptable convergence error thresholds $\varepsilon_{\text{conv}} > 0$ and $\varepsilon_{\text{bin}} > 0$, if the number of agents is at least $m_k \geq \frac{1}{4\varepsilon^2_{\text{bin}}\varepsilon_{\text{conv}}}$, then the pointwise error probability for each bin is bounded by $\varepsilon_{\text{conv}}$, i.e., $\mathbb{P}\left(\left|\frac{S^{m_k}_\infty[i]}{m_k} - \boldsymbol{\pi}[i]\right| > \varepsilon_{\text{bin}}\right) \leq \varepsilon_{\text{conv}}$, $\forall i \in \{1, \ldots, n_{\text{cell}}\}$.

Proof: Let $X^j_k[i]$ denote the independent Bernoulli random variable representing the event that the j^{th} agent is actually located in bin $R[i]$ at the k^{th} time instant, i.e., $X^j_k[i] = 1$ if $\boldsymbol{r}^j_k[i] = 1$ and $X^j_k[i] = 0$ otherwise. Let $X^j_\infty[i]$ denote the random variable $\lim_{k\to\infty} X^j_k[i]$. Theorem 4 implies that the success probability of $X^j_\infty[i]$ is given by $\mathbb{P}\left(X^j_\infty[i] = 1\right) = \lim_{k\to\infty} x^j_k[i] = \boldsymbol{\pi}[i]$. Hence $\mathbb{E}\left[X^j_\infty[i]\right] = \boldsymbol{\pi}[i]\cdot 1 + (1 - \boldsymbol{\pi}[i])\cdot 0 = \boldsymbol{\pi}[i]$, where $\mathbb{E}[\cdot]$ is the expected value of the random variable. Let $S^{m_k}_\infty[i] = X^1_\infty[i] + \ldots + X^{m_k}_\infty[i]$. As the random variables $X^j_\infty[i]$ are independent and identically distributed, the strong law of large numbers (cf. [44, pp. 85]) states that:

$$\mathbb{P}\left(\lim_{m_k\to\infty} \frac{S^{m_k}_\infty[i]}{m_k} = \boldsymbol{\pi}[i]\right) = 1\,. \tag{19}$$

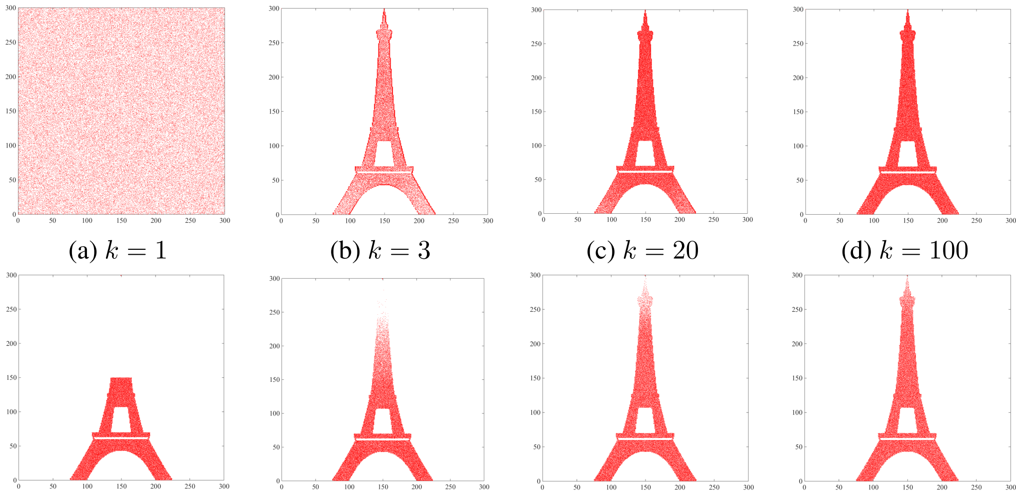
The final swarm distribution is also given by $\lim_{k\to\infty} \boldsymbol{\mu}^\star_k[i] = \frac{1}{m_k}\sum_{j=1}^{m_k} \boldsymbol{r}^j_k[i] = \frac{S^{m_k}_\infty[i]}{m_k}$. Hence (19) implies that $\lim_{m_k\to\infty}\lim_{k\to\infty} \boldsymbol{\mu}^\star_k = \boldsymbol{\pi}$ pointwise.

The variance of the independent random variable $X^j_\infty[i]$ is $\text{Var}(X^j_\infty[i]) = \boldsymbol{\pi}[i](1-\boldsymbol{\pi}[i])$, hence $\text{Var}(\frac{S^{m_k}_\infty[i]}{m_k}) = \frac{\boldsymbol{\pi}[i](1-\boldsymbol{\pi}[i])}{m_k}$. The Chebychev's inequality (cf. [45, Theorem 1.6.4, pp. 25]) implies that for any $\varepsilon_{\text{bin}} > 0$, the pointwise error probability for each bin is bounded by:

$$\mathbb{P}\left(\left|\frac{S^{m_k}_\infty[i]}{m_k} - \boldsymbol{\pi}[i]\right| > \varepsilon_{\text{bin}}\right) \leq \frac{\boldsymbol{\pi}[i](1-\boldsymbol{\pi}[i])}{m_k\varepsilon^2_{\text{bin}}} \leq \frac{1}{4m_k\varepsilon^2_{\text{bin}}}\,. \tag{20}$$

Hence, the minimum number of agents is given by $\frac{1}{4m_k\varepsilon^2_{\text{bin}}} \leq \varepsilon_{\text{conv}}$. ■

In this section, we have proved the convergence of the PSG–IMC algorithm for pattern formation. Extensive Monte Carlo simulations are presented below.



(a) $k = 1$ (b) $k = 3$ (c) $k = 20$ (d) $k = 100$

(e) $k = 101$ (f) $k = 110$ (g) $k = 150$ (h) $k = 200$

Figure 6. These plots show the swarm distribution of 10^5 agents at different time instants, in a sample run of the Monte Carlo simulation. Here the agents execute the PSG–IMC algorithm for pattern formation. Starting from an uniform distribution (a), the swarm converges to the desired formation ($\boldsymbol{\pi}$) of the Eiffel Tower. After 100 time instants, the agents in the top half of the formation ($\approx 3\times10^4$ agents) are removed and the remaining agents reconfigure to the desired formation.

Numerical Simulation of PSG–IMC Algorithm for Pattern Formation

In this numerical example, the PSG–IMC algorithm for pattern formation is used by a swarm of 10^5 agents to achieve the desired formation ($\boldsymbol{\pi}$) of the Eiffel Tower. Monte Carlo simulations are performed to study the performance of this algorithm and the cumulative results from 50 runs are shown in Fig. 7. As shown in Fig. 6(a), each simulation starts with the agents uniformly distributed across the state space which is partitioned into 300×300 bins. Each agent independently executes the PSG–IMC algorithm, illustrated in **Method 1**. During each time instant, each agents gets the feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$. Moreover, each agent is allowed to transition to only those bins which are at most 50 steps away.

As shown in Fig. 6(b), all agents enter the set of recurrent bins within 2 time instants. But the HD is large ($D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_3^\star) = 0.53$ from Fig. 7) because most of the agents are in the boundary bins of $\boldsymbol{\Pi}$. After 20 time instants, the HD has substantially improved ($D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_{20}^\star) = 0.17$ from Fig. 7), but the density of agents in a bin is more in the top half of the formation compared to the bottom half as shown in Fig. 6(c). In 100 time instants, the swarm has reached the desired formation ($D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_{100}^\star) = 0.067$ from Fig. 7) and the density of agents is uniform as shown in Fig. 6(d).

After 100 time instants, all the agents in the top half of the formation (i.e., 3.06×10^4 agents) are removed as shown in Fig. 6(e). The remaining agents from the bottom half slowly make their way to the top half as shown in Figs. 6(f,g). Once again, the swarm reaches the desired formation after another 100 time instants ($D_H(\boldsymbol{\pi}, \boldsymbol{\mu}_{200}^\star) = 0.096$ from Fig. 7) as shown in Fig. 6(h). Thus the repeatability and robustness properties of the PSG–IMC algorithm for pattern formation are evident from these simulation results.

A comparison with the HMC-based goal searching algorithm is also shown in Fig. 7. It is shown that the PSG–IMC algorithm provides ≈ 2 times improvement in convergence and ≈ 10 times reduction in the total number of transitions over the HMC-based algorithm.

PSG–IMC ALGORITHM FOR GOAL SEARCHING

In this section, we present a novel PSG–IMC algorithm for goal searching application, where the objective is to estimate some unknown target distribution over the state space. The HMC-based goal searching

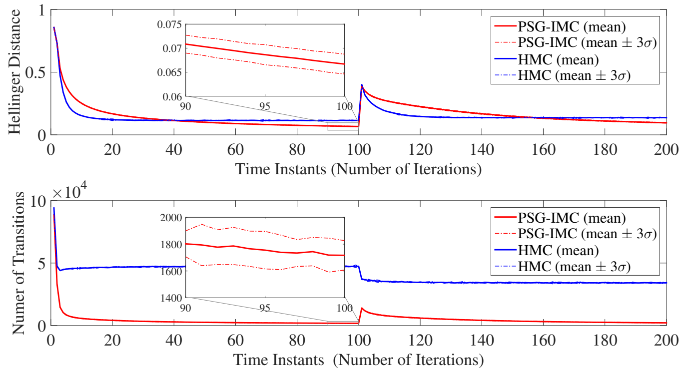


Figure 7. The cumulative results from 50 Monte Carlo simulations of PSG–IMC and HMC-based pattern formation algorithms are shown for (a) Hellinger distance and (b) number of transitions with respect to time instants. For clarity, the inset plots show the resolved data from 90^{th} to 100^{th} time instant. The discontinuity after the 100^{th} time instant is because of the removal of agents from the top half of the formation.

algorithm[13, 14] cannot handle motions constraints. We now present a more efficient PSG–IMC algorithm that can incorporate the feedback of the current swarm distribution to ensure that the swarm converges to the unknown target distribution and the agents settle down after the target distribution is achieved. Let us define the unknown target distribution.

Definition 11. *(Unknown Target Distribution $\boldsymbol{\sigma}$)* Let the unknown target distribution be represented by a probability (row) vector $\boldsymbol{\sigma} \in \mathbb{R}^{n_{\text{cell}}}$ over the bins in $\mathcal{R}$, i.e., $\boldsymbol{\sigma}\mathbf{1} = 1$. We assume that each agent can sense this target distribution in its bin, i.e., the agent in bin $R[i]$ can detect the target distribution $\boldsymbol{\sigma}[i]$. □

The objectives of the PSG–IMC algorithm for goal searching are as follows:
(i) Each agent independently determines its trajectory using a Markov chain, which obeys motion constraints ($\boldsymbol{A}_k^j$), so that the overall swarm converges to the unknown target distribution ($\boldsymbol{\sigma}$).
(ii) The algorithm reduces the number of transitions for achieving and maintaining the target distribution in order to conserve control effort (e.g., fuel).
(iii) The algorithm automatically detects and repairs damages to the formation.
(iv) The algorithm adapts the formation to changes in the target distribution.
In the previous PSG–IMC algorithm for pattern formation, the time-step between time instants is a constant. The key idea of this proposed PSG–IMC algorithm for goal searching is to modify the waiting time in a bin (i.e., the time-step between time instants) based on the target distribution in that bin.

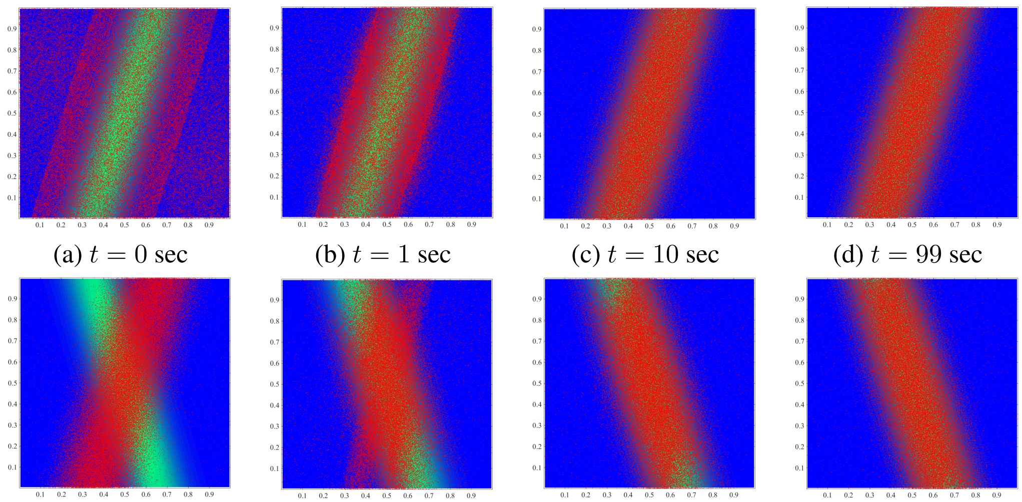
Let t represent continuous time. Let $\boldsymbol{z}_t^j \in \mathbb{R}^{n_{\text{cell}}}$ denote the row vector of probability mass function (pmf) of the predicted position of the j^{th} agent at time t, i.e., $\boldsymbol{z}_t^j\mathbf{1} = 1$. The i^{th} element ($\boldsymbol{z}_t^j[i]$) is the probability of the event that the j^{th} agent is in $R[i]$ bin at time t:

$$\boldsymbol{z}_t^j[i] = \mathbb{P}(\boldsymbol{r}_t^j[i] = 1), \quad \text{for all } i \in \{1, \dots, n_{\text{cell}}\}\,. \tag{21}$$

Let the j^{th} agent be in bin $R[i]$ at time t. The j^{th} agent can detect the target distribution $\boldsymbol{\sigma}[i]$, but it does not know the complete target distribution $\boldsymbol{\sigma}$. We use the following HD-based tuning parameter:

$$\xi_t^j = \begin{cases} 0 & \text{if } \boldsymbol{\mu}_t^j[i] \le \boldsymbol{\sigma}[i] \\ \frac{1}{\sqrt{2}}\left(\sqrt{\boldsymbol{\sigma}[i]} - \sqrt{\boldsymbol{\mu}_t^j[i]}\right) & \text{otherwise} \end{cases}\,. \tag{22}$$

This tuning parameter ensures that agents only transition out of the given bin when the number of agents in that bin is more than the required number of agents in that bin. Let $\boldsymbol{\pi} = \frac{1}{n_{\text{cell}}}\mathbf{1}^T$, $\boldsymbol{\alpha}_t^j = \mathbf{1}$, and the Markov matrices $\boldsymbol{M}_t^j$ and $\tilde{\boldsymbol{M}}_t^j$ are computed using Lemmas 1 and 2 respectively. If the time-step between time



(a) $t = 0$ sec (b) $t = 1$ sec (c) $t = 10$ sec (d) $t = 99$ sec

(e) $t = 100$ sec (f) $t = 105$ sec (g) $t = 110$ sec (h) $t = 120$ sec

Figure 8. These plots show the swarm distribution of 10^5 agents (in red) and the unknown target distribution (background contour plot), in a sample run of the Monte Carlo simulation. Here the agents execute the PSG–IMC algorithm for goal searching. Starting from an uniform distribution (a), the swarm converges to the unknown target distribution. After 100 sec, the unknown target distribution is suddenly changed and the agents reconfigure to the new target distribution.

instants is a constant, then it follows from the previous PSG–IMC algorithm for pattern formation that the swarm convergences to $\frac{1}{n_{\text{cell}}}\mathbf{1}^T$. Then $\lim_{t\to\infty} \boldsymbol{z}_t^j[i] = \frac{1}{n_{\text{cell}}}$ for all $i \in \{1,\ldots,n_{\text{cell}}\}$.

We modify the previous algorithm by making the waiting time in a bin (i.e., the time-step between time instants) directly proportional to the detected target distribution $\boldsymbol{\sigma}[i]$ in that bin, i.e., the waiting time in bin $R[i]$ is equal to $\tau_c\boldsymbol{\sigma}[i]$ where τ_c is the constant of proportionality. Then the agent's pmf vector $\boldsymbol{z}_t^j$ is given by $\lim_{t\to\infty} \boldsymbol{z}_t^j[i] = \frac{\boldsymbol{\sigma}[i]}{n_{\text{cell}}}$ for all $i \in \{1,\ldots,n_{\text{cell}}\}$. This implies that the swarm converges to the target distribution $\boldsymbol{\sigma}$. The pseudo-code for this PSG–IMC algorithm for goal searching is given in **Method 2** and the simulation results are presented below. The convergence, stability and robustness properties of this algorithm are also exactly similar to those of the previous algorithm.

Method 2: PSG–IMC algorithm for goal searching

(one iration of j^{th} agent at time t sec)
Given agent's current position $\boldsymbol{r}_t^j$, e.g. $\boldsymbol{r}_t^j[i] = 1$
Given feedback of current swarm distribution $\boldsymbol{\mu}_t^j$
Detect the target distribution $\boldsymbol{\sigma}[i]$
if $t - t_0 < \tau_c\boldsymbol{\sigma}[i]$, **then** Wait in bin $R[i]$
else Compute the tuning parameter ξ_t^j } Eq. (22)
  Set $\boldsymbol{\pi} = \frac{1}{n_{\text{cell}}}\mathbf{1}^T$ and $\boldsymbol{\alpha}_t^j = \mathbf{1}$
  Compute Markov matrices $\boldsymbol{M}_t^j$ and $\tilde{\boldsymbol{M}}_t^j$ } Lemmas 1 and 2
  Generate a random number $z \in \text{unif}[0;1]$ }Inverse Transform Sampling
  Go to bin $R[q]$ such that $\sum_{\ell=1}^{q-1}\tilde{\boldsymbol{M}}_t^j[i,\ell] \le z < \sum_{\ell=1}^{q}\tilde{\boldsymbol{M}}_t^j[i,\ell]$
  Set $t_0 = t$ **end if**

Numerical Simulation of PSG–IMC Algorithm for Goal Searching

In this numerical example, the swarm of 10^5 agents use the PSG–IMC algorithm for goal searching to estimate the unknown target distribution $\boldsymbol{\sigma}$. The state space $[0,1]\times[0,1]$ is partitioned into 100×100

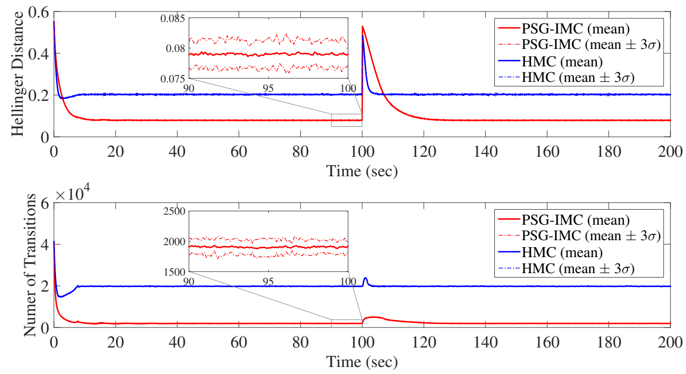


Figure 9. The cumulative results from 50 Monte Carlo simulations of PSG–IMC and HMC-based goal searching algorithms are shown for (a) Hellinger distance and (b) number of transitions with respect to time instants. For clarity, the inset plots show the resolved data from 90 to 100 sec. The discontinuity/change after the 100^{th} sec is because of the sudden change in the unknown target distribution.

disjoint bins. The unknown target distribution for the time between 0 to 100 sec is given by the multivariate normal distribution $f_1(x,y) = \frac{1}{c_1}\boldsymbol{\mathcal{N}}\left(\left[\begin{array}{cc} 0.5 & 0.5 \end{array}\right], \left[\begin{array}{cc} 0.1 & 0.3 \\ 0.3 & 1.0 \end{array}\right]\right)$, where the constant c_1 ensures that $\int_0^1\int_0^1 f_1(x,y)dxdy = 1$. The unknown target distribution $\boldsymbol{\sigma}_1$ is the pmf representation of this distribution $f_1(x,y)$ over the set of bins, as shown in the background contour plots in Fig. 8(a-d). Similarly, the unknown target distribution after 100 sec is given by the multivariate normal distribution $f_2(x,y) = \frac{1}{c_2}\boldsymbol{\mathcal{N}}\left(\left[\begin{array}{cc} 0.5 & 0.5 \end{array}\right], \left[\begin{array}{cc} 0.1 & -0.3 \\ -0.3 & 1.0 \end{array}\right]\right)$, and unknown target distribution $\boldsymbol{\sigma}_2$ is the pmf representation of this distribution $f_2(x,y)$, as shown in the background contour plots in Fig. 8(e-h). Obviously the agents do not know these target distributions and their objective is to estimate them. The performance of the PSG–IMC algorithm is studied using Monte Carlo simulations and the cumulative results from 50 runs are shown in Fig. 9.

Each agent executes the PSG–IMC algorithm for goal searching that is illustrated in **Method 2**. Starting from a random distribution (shown in Fig. 8(a)), the agents quickly converge to the unknown target distribution within 10 sec as shown in Figs. 8(b,c) and the corresponding HD is $D_H(\boldsymbol{\sigma}_1, \boldsymbol{\mu}^\star_{10}) = 0.085$ (from Fig. 9). After 100 sec, the unknown target distribution is suddenly changed as shown in Figs. 8(d,e). Once again, the swarm converges to the new target distribution within 20 sec and the corresponding HD is $D_H(\boldsymbol{\sigma}_2, \boldsymbol{\mu}^\star_{120}) = 0.087$ (from Fig. 9). Thus the repeatability and robustness properties of the PSG–IMC algorithm for goal searching are evident from these simulation results.

A comparison with the HMC-based goal searching algorithm is also shown in Fig. 9. It is shown that the PSG–IMC algorithm provides ≈ 3 times improvement in convergence and ≈ 10 times reduction in the total number of transitions over the HMC-based algorithm.

CONCLUSION

A novel technique for boosting the performance of HMC-based swarm guidance algorithms, using IMCs in the presence of feedback, is presented in this paper. The PSG–IMC algorithm performs much better than existing HMC-based algorithms because the agents only transition out of bins with surplus agents and the number of transitions is proportional to the tuning parameter. In case of pattern formation applications, this tuning parameter is the HD between the current swarm distribution and the desired formation. While in case of goal searching applications, this tuning parameter is HD-based difference between the current swarm distribution and the target distribution in the agent's present bin. The PSG–IMC algorithm relies on the generation of Markov matrices with a given stationary distribution using this tuning parameter. These

Markov matrices satisfy motion constraints and are designed to minimize the coefficient of ergodicity of the IMC. Moreover the Markov (memoryless) property of the PSG–IMC algorithm ensures that swarm is robust to external disturbances or damages or addition/removal of agents.

We have shown using extensive Monte Carlo simulations that the PSG–IMC algorithms for pattern formation and goal searching applications simultaneously provides $2-6$ times improvement in convergence and $10-60$ times reduction in the total number of transitions over HMC-based algorithms. Moreover, cooperative control tasks can also be formulated as pattern formation or goal searching applications. Hence, this paper presents an efficient solution approach for any swarm guidance problem, where the state-space can be partitioned into disjoint bins and the number of agents is at least an order of magnitude more than the number of bins. We envisage that these techniques for incorporating feedback into HMC-based algorithms can also be used in other areas in robotics.

ACKNOWLEDGMENT

This research was supported in part by AFOSR grant FA95501210193. This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] Şahin, E. and Spears, W. M., *Swarm Robotics*, Vol. 3342, Springer Berlin Heidelberg, Berlin, 2005.

[2] Hadaegh, F. Y., Chung, S.-J., and Manohara, H. M., "On development of 100-gram-class spacecraft for swarm applications," *IEEE Syst. J.*, 2014, in press, DOI: 10.1109/JSYST.2014.2327972.

[3] Morgan, D., Subramanian, G. P., Bandyopadhyay, S., Chung, S.-J., and Hadaegh, F. Y., "Probabilistic guidance of distributed systems using sequential convex programming," *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, Sept. 2014, accepted.

[4] Rubenstein, M., Cornejo, A., and Nagpal, R., "Programmable self-assembly in a thousand-robot swarm," *Science*, Vol. 345, No. 6198, 2014, pp. 795–799.

[5] Gross, R., Bonani, M., Mondada, F., and Dorigo, M., "Autonomous Self-Assembly in Swarm-Bots," *IEEE Trans. Robotics*, Vol. 22, No. 6, Dec. 2006, pp. 1115–1130.

[6] Kushleyev, A., Mellinger, D., Powers, C., and Kumar, V., "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, Vol. 35, No. 4, 2013, pp. 287–300.

[7] Barnes, L., Alvis, W., Fields, M.-A., Valavanis, K., and Moreno, W., "Heterogeneous Swarm Formation Control Using Bivariate Normal Functions to Generate Potential Fields," *IEEE Workshop Distributed Intell. Syst.: Collective Intelligence and Its Applicat.*, 2006, pp. 85–94.

[8] Cheng, J., Cheng, W., and Nagpal, R., "Robust and self-repairing formation control for swarms of mobile agents," *AAAI Proc. Nat. Conf. Artificial Intell.*, Vol. 1, Pittsburgh, PA, July 2005, pp. 59–64.

[9] Açikmeşe, B. and Bayard, D. S., "A Markov Chain Approach to Probabilistic Swarm Guidance," *Amer. Control Conf.*, June 2012, pp. 6300–6307.

[10] Chattopadhyay, I. and Ray, A., "Supervised self-organization of homogeneous swarms using ergodic projections of Markov chains," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, Vol. 39, No. 6, 2009, pp. 1505–1515.

[11] Bruemmer, D. J., Dudenhoeffer, D. D., McKay, M. D., and Anderson, M. O., "A Robotic Swarm for Spill Finding and Perimeter Formation," *Spectrum*, Reno, NV, 2002.

[12] Hurtado, J. E., Robinett, R. D., Dohrmann, C. R., and Goldsmith, S. Y., "Decentralized control for a swarm of vehicles performing source localization," *J. Intell. Robotic Syst.*, Vol. 41, No. 1, 2004, pp. 1–18.

[13] Hereford, J. M., "Analysis of a new swarm search algorithm based on trophallaxis," *IEEE Congr. Evol. Computation*, Barcelona, Spain, July 2010.

[14] Hereford, J. M., "Analysis of BEECLUST swarm algorithm," *IEEE Symp. Swarm Intell.*, Paris, France, April 2011.

[15] Mesquita, A. R., Hespanha, J. P., and Åström, K., "Optimotaxis: A stochastic multi-agent on site optimization procedure," *Proc. Hybrid Systems: Computation and Control*, St. Louis, MO, Apr. 2008.

[16] Napp, N. and Klavins, E., "Load Balancing for Multi-Robot Construction," *IEEE Inter. Conf. Robotics Automation*, Shanghai, China, May 2011.

[17] Correll, N. and Martinoli, A., "Robust Distributed Coverage using a Swarm of Miniature Robots," *IEEE Inter. Conf. Robotics Automation*, Rome, Italy, Apr. 2007.

[18] Ferrante, E., Turgut, A. E., Huepe, C., Stranieri, A., Pinciroli, C., and Dorigo, M., "Self-organized flocking with a mobile robot swarm: a novel motion control method," *Adaptive Behavior*, Oct. 2012.

[19] Cortés, J., Martinez, S., Karatas, T., and Bullo, F., "Coverage control for mobile sensing networks," *IEEE Trans. Robotics and Automation*, Vol. 20, No. 2, April 2004, pp. 243–255.

[20] Liu, W., Winfield, A., Sa, J., Chen, J., and Dou, L., *Swarm robotics*, chap. Strategies for energy optimisation in a swarm of foraging robots, Springer, 2007, pp. 14–26.

[21] Berman, S., Halasz, A., Hsieh, M. A., and Kumar, V., "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Trans. Robotics*, Vol. 25, No. 4, 2009, pp. 927–937.

[22] Grace, J. and Baillieul, J., "Stochastic strategies for autonomous robotic surveillance," *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, IEEE, 2005, pp. 2200–2205.

[23] Agharkar, P., Patel, R., and Bullo, F., "Robotic surveillance and Markov chains with minimal first passage time," *Proc. IEEE Conference on Decision and Control*, 2014.

[24] Devroye, L., *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, NY, 1986.

[25] Yu, D.-m., Tang, Y., Fang, J., Zhou, Y.-p., and Sun, M.-y., "Homogeneous Markov chain for modeling emotional interactions," *Inter. Conf. Advanced Commun. Technology*, Vol. 1, IEEE, 2008, pp. 265–269.

[26] König, L., Mostaghim, S., and Schmeck, H., "A Markov-chain-based Model for Success Prediction of Evolution in Complex Environments." *IJCCI (ECTA-FCTA)*, 2011, pp. 90–102.

[27] Bandyopadhyay, S., Chung, S.-J., and Hadaegh, F. Y., "Inhomogeneous Markov Chain Approach to Probabilistic Swarm Guidance Algorithm," *5th Int. Conf. Spacecraft Formation Flying Missions and Technologies*, Munich, Germany, 2013.

[28] Bandyopadhyay, S., Chung, S.-J., and Hadaegh, F. Y., "Probabilistic Swarm Guidance using Optimal Transport," *IEEE Multi-Conf. Sys. Control*, Antibes, France, Oct. 2014.

[29] Olshevsky, A., *Efficient Information Aggregation Strategies for Distributed Control and Signal Processing*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2010, p. 99.

[30] Habibi, G. and McLurkin, J., "Maximum-Leaf Spanning Trees for Efficient Multi-Robot Recovery with Connectivity Guarantees," *Distributed Autonomous Robotic Systems*, Springer, 2014, pp. 275–289.

[31] Olfati-Saber, R. and Murray, R., "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, Vol. 49, No. 9, 2004, pp. 1520 – 1533.

[32] Tsitsiklis, J., Bertsekas, D., and Athans, M., "Distributed Asynchronous Deterministic and Stochastic Gradient Optimization Algorithms," *IEEE Trans. Autom. Control*, Vol. 31, No. 9, 1986, pp. 803 – 812.

[33] Bandyopadhyay, S. and Chung, S.-J., "Distributed Estimation using Bayesian Consensus Filtering," *Amer. Control Conf.*, Portland, OR, June 2014, pp. 634–641.

[34] Yuan, Y., Stan, G.-B., Shi, L., Barahona, M., and Goncalves, J., "Decentralised minimum-time consensus," *Automatica*, Vol. 49, No. 5, 2013, pp. 1227–1235.

[35] Olshevsky, A., "Average Consensus in Nearly Linear Time and Its Implications for Decentralized Optimization and Multi-Agent Control," *arXiv preprint arXiv:1411.4186*, 2014.

[36] Chib, S. and Greenberg, E., "Understanding the metropolis-hastings algorithm," *The american statistician*, Vol. 49, No. 4, 1995, pp. 327–335.

[37] Billera, L. J. and Diaconis, P., "A geometric interpretation of the Metropolis-Hastings algorithm," *Statistical Science*, 2001, pp. 335–339.

[38] Torgerson, E., *Comparison of Statistical Experiments*, Cambridge University Press, 1991.

[39] Cha, S., "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions," *Int. J. Math. Models and Methods in Appl. Sci.*, Vol. 1, No. 4, 2007, pp. 300–307.

[40] Levin, D. A., Peres, Y., and Wilmer, E. L., *Markov chains and mixing times*, American Mathematical Soc., 2009.

[41] Seneta, E., *Non-negative Matrices and Markov Chains*, Springer-Verlag, New York, NY, 2006.

[42] Boyd, S. and Vandenberghe, L., *Convex optimization*, Cambridge university press, 2009.

[43] Meyer, C. D., *Matrix analysis and applied linear algebra*, Siam, 2000.

[44] Billingsley, P., *Probability and Measure*, J. Wiley & Sons, New York, 1995.

[45] Durrett, R., *Probability: Theory and Examples*, Thomson Brooks, 2005.