Probabilistic and Distributed Control of a Large-Scale Swarm of Autonomous Agents

Saptarshi Bandyopadhyay, Soon-Jo Chung *Senior Member, IEEE*, and Fred Y. Hadaegh *Fellow, IEEE*

***Abstract*—We present a distributed control algorithm of simultaneously solving both the stochastic target assignment and optimal motion control for large-scale swarms to achieve complex formation shapes. Our Probabilistic Swarm Guidance using Inhomogeneous Markov Chains (PSG–IMC) algorithm adopts an Eulerian density-control framework, where the physical space is partitioned into multiple bins and the swarm's density distribution over each bin is controlled in a probabilistic fashion to efficiently handle loss or addition of agents. We assume that the number of agents is much larger than the number of bins and that each agent knows in which bin it is located, the desired formation shape, and the objective function and motion constraints. PSG–IMC determines the bin-to-bin transition probabilities of each agent using a time-inhomogeneous Markov chain. These time-varying Markov matrices are computed by each agent in real time using the feedback from the current swarm distribution, which is estimated in a distributed manner. The PSG–IMC algorithm minimizes the expected cost of transitions per time instant, required to achieve and maintain the desired formation shape, even if agents are added to or removed from the swarm. PSG–IMC scales well with a large number of agents and complex formation shapes, and can also be adapted for area exploration applications. We demonstrate the effectiveness of this proposed swarm guidance algorithm by using numerical simulations and hardware experiments with multiple quadrotors.**

***Index Terms*—swarm robotics, multi-agent systems, Markov chains, path planning, guidance.**

I. Introduction

A large-scale swarm of space robotic systems could collaboratively complete tasks that are very difficult for a single agent, with significantly enhanced flexibility, adaptability, and robustness [1]. Moreover, a large-scale swarm of robots (having 10^3–10^6 or more agents) may be deployed for challenging missions like constructing a complex formation shape (see Fig. 1) [2]–[8] or exploring an unknown environment [9]–[14]. The control algorithm for such a large-scale swarm of autonomous agents should be:

- Distributed: The algorithm should not depend on a centralized supervisor or controller.
- Versatile: The algorithm can be easily tailored for multiple applications such as reconfiguring the formation shape or exploring the target area.
- Robust: Since a fraction of agents in the swarm might get lost during the course of an operation or new agents might get added to the swarm, the algorithm should seamlessly adapt to loss or addition of agents. Moreover, the algorithm should effectively deal with sensing and actuation errors and other uncertainties.
- Scalable: The algorithm should scale well with the number of agents and the size of the area.

S. Bandyopadhyay and F. Y. Hadaegh are with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA (e-mail: saptarshi.bandyopadhyay@jpl.nasa.gov and fred.y.hadaegh@jpl.nasa.gov). S.-J. Chung is with the California Institute of Technology, Pasadena, CA 91125, USA. (email: sjchung@caltech.edu). This research was supported in part by AFOSR grant FA95501210193 and NSF IIS 1253758.

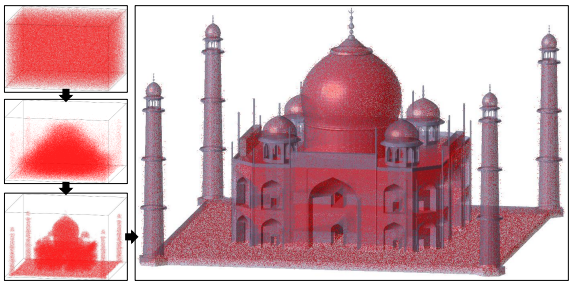


Figure 1. Using PSG–IMC for shape formation, a million swarm agents (shown in red) attain the complex 3-D shape of the Taj Mahal (translucent silhouette shown in gray). The physical space is partitioned into $100\times100\times70$ bins. See the supplementary video (SV1).

In this paper, we lay the theoretical foundations of a distributed, versatile, robust, and scalable algorithm for controlling the shape of large-scale swarms.

One way of categorizing distributed or decentralized control algorithms for swarms is to draw an analogy from fluid mechanics [15], [16]. Whereas each agent's trajectory is generated separately in the individual-agent-based *Lagrangian* framework [2]–[14], a collective property of the swarm, such as its density distribution, is tracked and controlled over time in the *Eulerian* framework. If the Lagrangian framework is used for controlling a large number (10^3–10^6 or more) of agents, the computation cost for generating all the optimal trajectories one by one would be prohibitively expensive. Furthermore, the computational complexity of allocating the optimal target location to each agent (i.e., target assignment) increases at least quadratically with the number of agents for a distributed algorithm [5]–[8] while the well-known centralized assignment algorithm of the Hungarian method possesses a cubic time complexity [17]. In essence, a Lagrangian approach does not scale well with the size of the area and arbitrary formation shapes [2]–[4].

Moreover, individual control of each agent in a deterministic fashion does not efficiently handle loss or addition of agents as shall be discussed in Sec. I-A. Consequently, we adopt and systematically combine the Eulerian framework of controlling the density distribution of each partitioned volume with a probabilistic control method. The resulting algorithm simultaneously solves both the stochastic target assignment problem

for achieving a desired shape and generation of optimal bin-to-bin transition trajectories at a coarser spatial resolution, which provides a major computational cost saving.

A. Literature Review

Numerous path planning algorithms within the Lagrangian framework are discussed in the survey papers on swarm robotics [18], [19]. In this subsection, we focus on guidance algorithms that use an Eulerian [20]–[22] and probabilistic approach. For shape formation and reconfiguration applications, the physical space over which the swarm is distributed is first tessellated or partitioned into multiple bins [23], [24]. The bin size is determined by the spatial resolution of the desired formation shape. Assuming that the number of agents is much larger than the number of non-empty bins, the density distribution of the swarm over these bins is controlled to achieve the desired formation shape in a deterministic or probabilistic fashion.

A deterministic target assignment algorithm needs to keep track of the changes in the number of agents and targets [7], [17]. In contrast, a probabilistic approach is more effective and efficient in handling the time-varying number of agents in the swarm, since each agent often lacks the ability to track the number of agents in the swarm. Moreover, our probabilistic approach can also handle measurement uncertainties and actuation errors in a robust manner.

One popular probabilistic method within the Eulerian framework uses a Homogeneous Markov Chain (HMC) for shape formation [25]–[28], area exploration [29], [30], task allocation [31], [32], and surveillance applications [33], [34]. In such an algorithm, the agent's transition probability between bins is encoded in a constant Markov matrix that has the desired formation shape as its stationary distribution. Such an approach is probabilistic, as opposed to deterministic, because each agent determines its next bin location by sampling based on the Markov matrix [35]. The HMC-based algorithms possess the aforementioned benefits of robustness and scalability, because addition or removal of agents from the swarm does not affect the property of convergence to the stationary distribution.

However, the major drawback of these HMC-based algorithms is that they are inherently open-loop strategies which cannot incorporate any feedback. Clearly, the effectiveness of these algorithms can be greatly improved by refining the Markov matrix at each time step using some feedback. Such refinement results in an Inhomogeneous Markov Chain (IMC), which is at the core of our algorithm.

In this paper, we derive the Probabilistic Swarm Guidance using Inhomogeneous Markov Chains (PSG–IMC) algorithm, which incorporates the feedback from the current swarm density distribution at each time step. PSG–IMC is a closed-loop distributed guidance strategy that retains the original robustness and scalability properties associated with a Markovian approach. Another disadvantage of HMC-based algorithms is that they suffer undesirable transitions, i.e., transitions from bins that are deficient in agents to bins with surplus agents. Such undesirable transitions prevent the swarm from converging to the desired formation. The PSG–IMC algorithm suppresses such undesirable transitions between bins, thereby reducing the control effort needed for achieving and maintaining the formation. This benefit also results in smaller convergence errors than HMC-based algorithms.

Swarm guidance can also be formulated as an optimal transport problem [36], [37] (See Remark 12 in Appendix). The optimal transport map for each one-time-step transition from the current swarm distribution to the desired formation is found using an optimization problem. Hence, the estimated current swarm distribution is directly used as a constraint in the optimization problem. If perfect estimation of the current swarm distribution is available, then this optimal transport-based algorithm gives good performance. However, there are two major disadvantages of such an approach. First, we show in this paper that the performance of an optimal transport-based algorithm drops precipitously with estimation errors in the feedback loop. Measurement and estimation errors are routinely encountered in practice and it is often impossible or impractical to generate perfect feedback of the current swarm distribution. Second, the computation time of the optimization problem increases very fast with the increasing number of bins. This is a notable drawback because a large number of bins are necessary for capturing fine spatial details in the desired formation shape. PSG–IMC can overcome both challenges, since it works effectively in the presence of error-prone feedback and scales well with a large number of bins.

Another Eulerian approach is to model the swarm dynamics as a continuum model using a partial differential equation (PDE) [38]–[40]. Since our goal is to achieve arbitrary formation shapes that are not limited to to the equilibrium states of the PDE, we do not consider a PDE-based approach. Our approach is also different from the multiagent Markov decision process approach [41], [42] because the agents do not keep track of the states and actions of other agents.

B. Main Contributions

The first contribution of this paper is a novel technique for constructing feedback-based time-varying Markov matrices for a given stationary distribution which represents the desired formation, where the expected cost of transitions at each time instant is minimized. Each Markov matrix satisfies the motion constraints that might arise due to the dynamics or other physical constraints. The Markov matrix converges to the identity matrix when the swarm converges to the desired formation, thereby reducing unnecessary transitions and ensuring that the swarm settles down.

Second, we rigorously derive the convergence proofs of PSG–IMC for shape formation, based on the analysis of IMC, which are more involved than the convergence proofs of HMC. We show that each agent's IMC strongly ergodically converges to the desired formation shape. We also provide a time-varying probabilistic bound on the convergence error as well as a lower bound on the number of agents for ensuring that the final convergence error is below the desired threshold. Furthermore, we present an extension of PSG–IMC for area exploration [43]–[45] to show the versatility of the proposed method.

Using multiple aerial robots, we demonstrate that PSG–IMC, hierarchically combined with a lower-level

Table I
LIST OF FREQUENTLY USED SYMBOLS

Symbol	Explanation
$\boldsymbol{A}_k^j$	Motion constraint matrix of the j^{th} agent (Definition 4)
$B[i]$	Bins, where $1 \leq i \leq n_{\text{bin}}$ (Definition 1)
$\boldsymbol{M}_k^j$	Markov matrix of the j^{th} agent (6)
$\boldsymbol{S}_k^j$	Condition for escaping transient bins (17)
$\boldsymbol{C}_k$	Cost matrix at the k^{th} time instant (Definition 5)
m_k	Number of agents in the swarm (Assumption 1)
$\boldsymbol{r}_k^j$	Actual bin position of the j^{th} agent (Assumption 3)
$\boldsymbol{x}_k^j$	Probability vector of the j^{th} agent predicted position (22)
$\boldsymbol{\Theta}$	Probability vector of the desired formation (Definition 2)
ϵ_{est}	Estimation error between $\boldsymbol{\mu}_k^\star$ and $\boldsymbol{\mu}_k^j$ (3)
$\boldsymbol{\mu}_k^\star$	Current swarm distribution (Definition 3)
$\boldsymbol{\mu}_k^j$	Estimate of $\boldsymbol{\mu}_k^\star$ by the j^{th} agent
ξ_k^j	Hellinger-distance-based feedback error (Definition 6)

collision-free motion planner, can be executed in real time to reconfigure into multiple desired formation shapes. Using numerical simulations, we also show that PSG–IMC yields smaller convergence errors and more robust convergence results than the HMC-based and optimal transport-based algorithms, while significantly reducing the number of transitions in the presence of estimation errors. Thus, PSG–IMC is best-suited for large-scale swarms with error-prone feedback and complex desired formations with a large number of bins.

Compared to our conference paper [46], we have added detailed proofs of the convergence analysis and extensions of our algorithm for shape formation and area exploration applications. We have also added numerical and experimental results to this paper. The paper is organized as follows. The problem statement for shape formation is discussed in Section II. In Section III, the techniques for constructing Markov matrices are given. The PSG–IMC algorithm for shape formation is presented in Section IV. The PSG–IMC algorithm for area exploration is presented in Section V. Results of numerical simulation and experimentation are discussed in Section VI. Concluding remarks are given in Section VII.

C. Notations

The *time index* is denoted by a right subscript and the *agent index* is denoted by a lower-case right superscript (see Table I for important symbols. The symbol $\mathbb{P}(\cdot)$ denotes the probability of an event. Let $\mathbb{N}$ and $\mathbb{R}$ be the set of natural numbers (positive integers) and real numbers, respectively. The matrix $\text{diag}(\boldsymbol{\alpha})$ denotes the diagonal matrix of appropriate size with the vector $\boldsymbol{\alpha}$ as its diagonal elements. Let $\mathbf{1} = [1, 1, \ldots, 1]^T$, $\mathbf{I}$, $\mathbf{0}$, and $\emptyset$ denote the ones (column) vector, the identity matrix, the zero matrix of appropriate size, and the empty set, respectively. Let $\|\cdot\|_p$ denote the ℓ_p vector norm. Let $\min^+$ denote the minimum of the positive elements.

A probability vector $\boldsymbol{a} \in \mathbb{R}^n$ is a row vector satisfying $\boldsymbol{a} \geq 0$ and $\boldsymbol{a}\mathbf{1} = 1$ [47, pp. 92]. The metric $D_{(\cdot)}(\boldsymbol{a}, \boldsymbol{b})$ connotes the distance between probability vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, where the subscript represents the type of metric.

II. PRELIMINARIES AND PROBLEM STATEMENT

After stating the key definitions and assumptions, we give the problem statement for shape formation in Section II-B and present the PSG–IMC algorithm in Section II-C.

Definition 1 (*Bins* $B[i]$). The compact physical space over which the swarm is distributed is partitioned into n_{bin} disjoint bins. Each bin is denoted by $B[i]$, $i \in \{1, \ldots, n_{\text{bin}}\}$. The size of the bins is determined by the spatial resolution of the desired formation shape (e.g., $n_{\text{bin}} = 25$ in Fig. 2). □

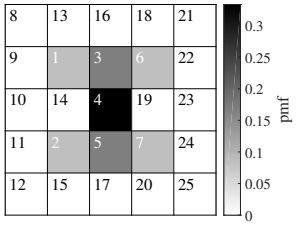


Figure 2. In this example, the desired formation $\boldsymbol{\Theta} = [\frac{1}{12}, \frac{1}{12}, \frac{1}{6}, \frac{1}{3}, \frac{1}{6}, \frac{1}{12}, \frac{1}{12}, \mathbf{0}^{1\times 18}]$. The bins 1 to 7 are recurrent bins.

Definition 2 (*Desired formation* $\boldsymbol{\Theta}$ *and Recurrent Bins*). The desired formation shape $\boldsymbol{\Theta}$ is a probability vector in $\mathbb{R}^{n_{\text{bin}}}$. Each element $\boldsymbol{\Theta}[i]$ represents the desired swarm density in the corresponding bin $B[i]$. The bins that have nonzero elements in $\boldsymbol{\Theta}$ are called recurrent bins. Let n_{rec} denote the number of recurrent bins. The remaining bins, with zero elements in $\boldsymbol{\Theta}$, are called transient bins. Without loss of generality, we re-label the bins such that the first n_{rec} bins are recurrent bins (i.e., $\boldsymbol{\Theta}[i] > 0$ for all $i \in \{1, \ldots, n_{\text{rec}}\}$). Then, the remaining bins are transient bins (i.e., $\boldsymbol{\Theta}[i] = 0$ for all $i \in \{n_{\text{rec}} + 1, \ldots, n_{\text{bin}}\}$). For example, see Fig. 2. □

Note that representing the desired formation as a distribution over bins is analogous to representing a 2-D image using pixels or a 3-D shape using voxels. The complex desired formation shapes of the Taj Mahal in Fig. 1 and the Eiffel Tower in Fig. 6 are generated in this manner.

Assumption 1. Let the scalar $m_k \in \mathbb{N}$ denote the number of agents in the swarm at the k^{th} time instant under the assumption of $m_k \gg n_{\text{rec}}$. The agents do not keep track of m_k. In Section IV-A, a lower-bound on m_k is computed for achieving the desired convergence error. Moreover, we can only achieve the best quantized representation of $\boldsymbol{\Theta}$ using m_k agents, due to the quantization error of $\frac{1}{m_k}$. For example, if $\boldsymbol{\Theta} = [\frac{1}{3}, \frac{2}{3}]$ and $m_k = 10$, then the best-quantized representation of $\boldsymbol{\Theta}$ that can be achieved is $[0.3, 0.7]$. □

Assumption 2. Each agent is assumed to be *anonymous* and *identical* without any global identifier or index. Hence, all agents execute the same algorithm [48]. Efficient algorithms for indexed agents (e.g., a spanning-tree-based algorithm [49]) are not applicable in this paper. □

Assumption 3. Each agent is assumed to be able to sense which bin it belongs to. This requirement is less stringent than having to know the precise location in a global frame. The row vector $\boldsymbol{r}_k^j \in \mathbb{R}^{n_{\text{bin}}}$ indicates the bin position. If $\boldsymbol{r}_k^j[i] = 1$, then the j^{th} agent is presently inside the bin $B[i]$ at the k^{th} time instant; otherwise $\boldsymbol{r}_k^j[i] = 0$. For example, spacecraft in low Earth orbit and outdoor robots can use the Global Positioning System

(GPS) or other less-precise navigation technologies (e.g., cell towers, radio beacons). See Assumption 4. □

Definition 3 (*Current swarm distribution* $\boldsymbol{\mu}_k^\star$). The current swarm distribution $\boldsymbol{\mu}_k^\star$ is a probability vector in $\mathbb{R}^{n_{\text{bin}}}$, given by the ensemble mean of actual bin positions of the agents:

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

Each element $\boldsymbol{\mu}_k^\star[i]$ gives the swarm density in the corresponding bin $B[i]$ at the k^{th} time instant. □

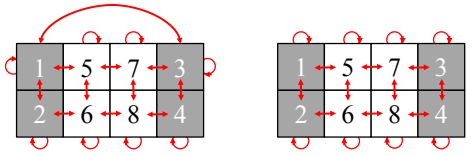
Definition 4 (*Motion constraints*). Each agent can transition to only certain bins because of the dynamics or physical constraints. These (possibly time-varying) motion constraints are specified by the matrix $\boldsymbol{A}_k^j \in \mathbb{R}^{n_{\text{bin}}\times n_{\text{bin}}}$ whose element is given by

$$\boldsymbol{A}_k^j[i,\ell] = \begin{cases} 1 & \text{if the transition from bin } B[i] \text{ to bin} \\ & B[\ell] \text{ is allowed at the } k^{\text{th}} \text{ time instant,} \\ 0 & \text{if this transition is not allowed.} \end{cases} \tag{2}$$

We assume that $\boldsymbol{A}_k^j$ satisfies the following properties:

1) The matrix $\boldsymbol{A}_k^j$ is symmetric and irreducible[1],
2) $\boldsymbol{A}_k^j[i,i] = 1$ for all agents, bins, and time instants,
3) Since the first n_{rec} bins are recurrent bins, the sub-matrix $\boldsymbol{A}_{k,\text{sub}}^j := \boldsymbol{A}_k^j[1:n_{\text{rec}}, 1:n_{\text{rec}}]$ encapsulates the motion constraints between the recurrent bins. The matrix $\boldsymbol{A}_{k,\text{sub}}^j$ is irreducible.

These properties are visualized in Fig. 3. □



(a) Allowed (b) Not Allowed

Figure 3. In this example, the bins 1 to 4 are recurrent bins. The allowed transitions (motion constraints) are shown as red arrows. Case (a) satisfies all the properties in Definition 4. Case (b) does not satisfy property (3) because the recurrent bins are not strongly connected.

As shown in Fig. 3, the recurrent bins need not be contiguous. Therefore, the desired distribution can have multiple disconnected components. Note that the matrix $\boldsymbol{A}_k^j$ is different from the Markov matrix introduced in Section III.

Definition 5 (*Cost matrix* $\boldsymbol{C}_k$). Consider a matrix $\boldsymbol{C}_k \in \mathbb{R}^{n_{\text{bin}}\times n_{\text{bin}}}$ whose element $\boldsymbol{C}_k[i,\ell]$ denotes the transition cost incurred by each agent moving from bin $B[i]$ to bin $B[\ell]$ at the k^{th} time instant. This cost represents the control effort, the fuel cost, or any other metric that the agents seek to minimize. The agents are assumed to incur no cost if they remain in their present bins. However, the agents incur some positive cost if they transition out of their present bin (i.e., $\boldsymbol{C}_k[i,i] = 0$ and $\boldsymbol{C}_k[i,\ell] > 0$ for all $i,\ell \in \{1,\ldots,n_{\text{bin}}\}$ and $i \neq \ell$). □

[1]A matrix is irreducible if and only if the graph conforming to that matrix is strongly connected.

Assumption 4 (*A priori information required*). The desired formation shape $\boldsymbol{\Theta}$ is given or communicated before the algorithm starts. The time-varying cost matrices $\boldsymbol{C}_k$ and the motion constraint matrices $\boldsymbol{A}_k^j$ are given a priori. Moreover, the four design variables (ε_M, ε_C, β^j, and τ^j), which are introduced later, are also given a priori. Depending on the motion planning inside the bins (e.g., Remark 15), additional information, such as the location of each agent in the current bin, might be required. □

A. Distributed Estimation of the Current Swarm Distribution

The algorithms in this paper use the feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$. In order to generate this estimate in a distributed manner, we need the following assumption.

Assumption 5. The time-varying communication network topology of the swarm is assumed to be strongly connected. Multiple distributed consensus algorithms exist in the literature for estimating $\boldsymbol{\mu}_k^\star$ on a strongly-connected graph [50], [51], [52] (see Remark 13 in Appendix). □

Any distributed estimation algorithm will have some residual estimation error between the current swarm distribution $\boldsymbol{\mu}_k^\star$ and the j^{th} agent's estimate of the current swarm distribution at the k^{th} time instant, which is denoted by the probability vector $\boldsymbol{\mu}_k^j \in \mathbb{R}^{n_{\text{bin}}}$. Let the positive parameter ϵ_{est} represent the maximum estimation error between $\boldsymbol{\mu}_k^\star$ and $\boldsymbol{\mu}_k^j$:

$$D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\mu}_k^j) = \sum_{i=1}^{n_{\text{bin}}} \left|\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\mu}_k^j[i]\right| \leq \epsilon_{\text{est}} \,, \forall k \in \mathbb{N} \,. \tag{3}$$

We later show that our algorithm works well in the presence of this estimation error bound ϵ_{est} in (3).

B. Problem Statement for Shape Formation

Under Assumptions 1–5, the objectives of PSG–IMC for shape formation are as follows:

- Each agent independently determines its bin-to-bin trajectory using a feedback-based inhomogeneous Markov chain, which obeys motion constraints $\boldsymbol{A}_k^j$, so that the overall swarm converges to a desired formation shape $\boldsymbol{\Theta}$.
- The algorithm minimizes the expected cost of transitions at every time instant (see Definition 8) for all the agents, where the cost matrix $\boldsymbol{C}_k$ is defined in Definition 5.
- The algorithm automatically detects and repairs damages to the formation.

C. Outline of the PSG–IMC Algorithm

The key steps in the proposed PSG–IMC algorithm for shape formation are shown in Fig. 4. The agent first determines its present bin and estimates the current swarm distribution (Section II-A). If the agent is in a transient bin, then it selects another bin using the condition for escaping transient bins (Section III-C). Otherwise, the agent computes the Markov matrix (Section III-A) and then modifies it to suppress undesirable transitions (Section IV). Finally, the agent uses inverse transform sampling to select the next bin (Remark 14 in Appendix). The agent uses a lower-level guidance and control

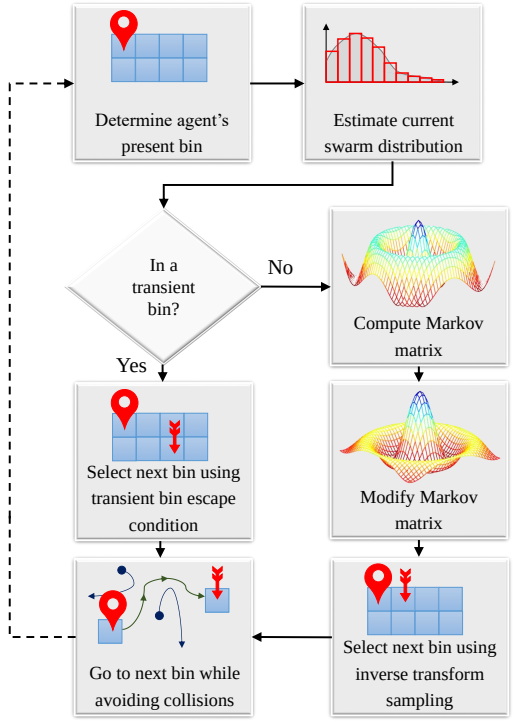


Figure 4. Flowchart of the PSG–IMC algorithm for shape formation.

algorithm to go from its present bin to the selected bin in a collision-free manner. Such lower-level algorithms based on real-time optimal control or Voronoi partitions are presented in [37], [53], [7] and also discussed briefly in Remark 15 in Appendix. The pseudo-code of the PSG–IMC algorithm for shape formation is given in Method 2 in Section IV.

III. Feedback-based Markov Matrix

Our method of constructing time-varying Markov matrices using swarm distribution errors is presented.

A. Construction of Minimum Cost Markov Matrix

In this subsection, we construct Markov matrices that minimize the expected cost of transitions at each time instant.

Definition 6 (*Feedback error ξ_k^j and desired error ξ_{des}*). The feedback error ξ_k^j is given by the Hellinger distance (HD) between the current swarm distribution $\boldsymbol{\mu}_k^j$ and the desired formation $\boldsymbol{\Theta}$:

$$\xi_k^j = D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j) := \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n_{\text{bin}}} \left(\sqrt{\boldsymbol{\Theta}[i]} - \sqrt{\boldsymbol{\mu}_k^j[i]} \right)^2}. \quad (4)$$

The HD is a symmetric measure of the difference between two probability distributions and bounded by 1 [54], [55].

Let ξ_{des} represent the desired convergence error threshold between the final swarm distribution and $\boldsymbol{\Theta}$. □

Remark 1 (*Advantages of Hellinger distance*). The HD between $\boldsymbol{\mu}_k^j$ and $\boldsymbol{\Theta}$ in (4) is bounded as follows [56]:

$$\frac{1}{2\sqrt{2}} D_{\mathcal{L}_1}(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j) \leq D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j) \leq \frac{1}{\sqrt{2}} D_{\mathcal{L}_1}(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j)^{\frac{1}{2}}. \quad (5)$$

We choose HD over other popular metrics like $\mathcal{L}_1$ and $\mathcal{L}_2$ distances, because of its properties illustrated in Fig. 5. The $\mathcal{L}_1$ distances for the cases ($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\mu}_3$) from $\boldsymbol{\Theta}$ are equal. But in Case 1, the wrong agent is in a bin where there should be no agent, hence HD heavily penalizes this case. If all the agents are only in those bins which have positive weights in $\boldsymbol{\Theta}$, then HD is significantly smaller. Finally, if an agent is missing from a bin that has fewer agents in $\boldsymbol{\Theta}$ (Case 2) compared to a bin that has more agents in $\boldsymbol{\Theta}$ (Case 3), then HD penalizes Case 2 slightly more than Case 3. These properties are useful for swarm guidance. □

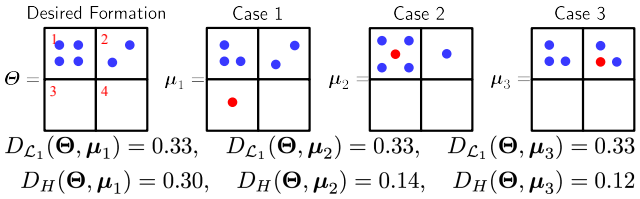


Figure 5. In this example, the desired distribution $\boldsymbol{\Theta}$ has 4 and 2 agents in bins 1 and 2, respectively. In the three cases, one agent (marked in red) is not in its correct bin. The $\mathcal{L}_1$ distances are equal, but the HDs are different.

Consider the Markov matrix $\boldsymbol{M}_k^j$ in $\mathbb{R}^{n_{\text{bin}} \times n_{\text{bin}}}$ whose element $\boldsymbol{M}_k^j[i, \ell]$ represents the transition probability that the j^{th} agent in bin $B[i]$ at the k^{th} time instant will transition to bin $B[\ell]$ at the $(k+1)^{\text{th}}$ time instant:

$$\boldsymbol{M}_k^j[i, \ell] := \mathbb{P}\left(\boldsymbol{r}_{k+1}^j[\ell] = 1 | \boldsymbol{r}_k^j[i] = 1 \right). \quad (6)$$

Therefore, the Markov matrix $\boldsymbol{M}_k^j$ is row stochastic (i.e., $\boldsymbol{M}_k^j \mathbf{1} = \mathbf{1}$). Its stationary distribution is defined as follows.

Definition 7 (*Stationary distribution*). The stationary distribution $\boldsymbol{e}_k^j$ of the Markov matrix $\boldsymbol{M}_k^j$ is given by the solution of $\boldsymbol{e}_k^j \boldsymbol{M}_k^j = \boldsymbol{e}_k^j$, where $\boldsymbol{e}_k^j$ is a probability (row) vector in $\mathbb{R}^{n_{\text{bin}}}$ (i.e., $\boldsymbol{e}_k^j \geq 0$, $\boldsymbol{e}_k^j \mathbf{1} = 1$). The stationary distribution is unique if the Markov matrix is irreducible [47, pp. 119]. □

Definition 8 (*Expected cost of transitions at each time instant*). The expected cost of transitions for the j^{th} agent at the k^{th} time instant is given by $\sum_{i=1}^{n_{\text{bin}}} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i, \ell] \boldsymbol{M}_k^j[i, \ell]$, where the cost matrix $\boldsymbol{C}_k$ is defined in Definition 5. □

Method 1, Theorem 1, and Corollary 1 present our construction of the optimal Markov matrix $\boldsymbol{M}_k^j$ that minimizes this expected cost of transitions at the each time instant. Our construction technique has no relation with the well-known Metropolis-Hastings (MH) algorithm, which is commonly used for constructing Markov matrices with a given stationary distribution [57], [58]. 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. There is no direct method for incorporating feedback into the MH algorithm. In contrast, the feedback of the current swarm distribution is directly incorporated within our construction process using the feedback error term.

Method 1: Computation of Optimal Markov Matrix

Under Assumptions 1–5, the optimal Markov matrix $\boldsymbol{M}_k^j$ that minimizes the expected cost of transitions at each time instant is constructed as follows:

(A) If $\xi_k^j < \xi_{\text{des}}$, then set $\boldsymbol{M}_k^j = \mathbf{I}$.

(B) Otherwise, $\boldsymbol{M}_k^j$ is computed as follows:

(CS1) If $\boldsymbol{A}_k^j[i,\ell] = 0$, then set $\boldsymbol{M}_k^j[i,\ell] = 0$ for all bins $i,\ell \in \{1,\dots,n_{\text{bin}}\}$.

(CS2) If $\boldsymbol{\Theta}[\ell] = 0$, then set $\boldsymbol{M}_k^j[i,\ell] = 0$ for all bins $i,\ell \in \{1,\dots,n_{\text{bin}}\}$ with $i \neq \ell$.

The remaining elements in $\boldsymbol{M}_k^j$ are computed using the following linear program (LP):

$$\text{minimize} \sum_{i=1}^{n_{\text{bin}}}\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell]\boldsymbol{M}_k^j[i,\ell]\,, \tag{7}$$

$$\text{subject to} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{M}_k^j[i,\ell] = 1,\ \forall i\,, \tag{LP1}$$

$$\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i]\boldsymbol{M}_k^j[i,\ell] = \boldsymbol{\Theta}[\ell],\ \forall \ell\,, \tag{LP2}$$

$$(1-\xi_k^j) \le \boldsymbol{M}_k^j[i,i] \le 1,\ \forall i\,, \tag{LP3}$$

$$\varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right) \le \boldsymbol{M}_k^j[i,\ell] \le \frac{\xi_k^j}{\varepsilon_M},\ \forall i \neq \ell\,, \tag{LP4}$$

where ε_M is a positive scalar constant in $(0,1]$, $C_{k,\max}$ is the maximum transition cost (i.e., $C_{k,\max} = \max_{i,\ell} \boldsymbol{C}_k[i,\ell]$), and ε_C is a positive scalar constant.

Remark 2. The Markov matrix $\boldsymbol{M}_k^j$ designed in Method 1 has the following desirable properties:

- If $\xi_k^j < \xi_{\text{des}}$ (see Definition 6), the swarm is deemed to have converged to the desired formation. Then, $\boldsymbol{M}_k^j$ is set to the identity matrix so that the agents do not transition anymore and that the swarm remains converged.
- If the swarm has not converged to the desired formation (i.e., $\xi_k^j \ge \xi_{\text{des}}$), then Step (B) is initiated.
- (CS1) prevents those transitions that are not allowed by the motion constraints.
- (CS2) prevents transitions into transient bins.
- The objective function in (7) is the expected cost of transitions at the current time instant (see Definition 8).
- (LP1) ensures that $\boldsymbol{M}_k^j$ is row stochastic.
- (LP2) ensures that $\boldsymbol{M}_k^j$ has $\boldsymbol{\Theta}$ as its stationary distribution (i.e., $\boldsymbol{\Theta}\boldsymbol{M}_k^j = \boldsymbol{\Theta}$).
- The lower bound in (LP3) ensures that there is a non-zero probability for each agent to remain in it present bin if $\xi_k^j < 1$. The upper bound in (LP3) is derived from (LP1).
- The lower bound in (LP4) ensures that the minimum transition probability to a target bin is non-zero and directly proportional to both the feedback error ξ_k^j and the target bin's desired distribution $\boldsymbol{\Theta}[\ell]$. But the minimum transition probability decreases with increasing cost of transition to the target bin.
- The upper bound in (LP4) ensures that the maximum transition probability is also directly proportional to the feedback error ξ_k^j.

A salient feature of the constraints (LP3,4) is that they depend on the feedback error ξ_k^j. Therefore, if the swarm distribution $\boldsymbol{\mu}_k^j$ converges to $\boldsymbol{\Theta}$ (i.e., $\boldsymbol{\mu}_k^j \to \boldsymbol{\Theta}$), then $\xi_k^j \to 0$ (because $\xi_k^j = D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j)$) and $\boldsymbol{M}_k^j \to \mathbf{I}$ based on these constraints. The identity matrix ensures that agents settle down after the desired formation is achieved, thereby reducing unnecessary transitions. In Section IV-A, we show that these constraints also help prove the convergence of the algorithm. □

Theorem 1. The feasible set of Markov matrices that satisfy the constraints (CS1,2) and the linear constraints in LP (7) in Method 1 is non-empty. The optimal Markov matrix $\boldsymbol{M}_k^j$ is row-stochastic, has $\boldsymbol{\Theta}$ as its stationary distribution, and only allows transitions into recurrent bins.

Proof: The optimization problem in (7) is an LP because the constraints are all linear inequalities or equalities and the objective function is linear. An optimal solution for the LP exists if the feasible set of Markov matrices is non-empty. We now show that the following family of Markov matrices $\boldsymbol{Q}_k^j$ are within the set of feasible solutions:

$$\boldsymbol{Q}_k^j[i,\ell] = \begin{cases} \frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]\right) & \text{if } \boldsymbol{A}_k^j[i,\ell] = 1 \\ 0 & \text{if } \boldsymbol{A}_k^j[i,\ell] = 0 \end{cases}, \quad \forall i,\ell \in \{1,\dots,n_{\text{bin}}\} \text{ and } i \neq \ell\,, \tag{8}$$

$$\boldsymbol{Q}_k^j[i,i] = \frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[i]\boldsymbol{\Theta}[i]\right) + \left(1-\xi_k^j\boldsymbol{\alpha}_k^j[i]\right) + \sum_{s\in\{1,\dots,n_{\text{bin}}:\boldsymbol{A}_k^j[i,s]=0\}} \frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[s]\boldsymbol{\Theta}[s]\right)\,, \tag{9}$$

where $\varepsilon_\alpha = \sqrt{\varepsilon_M}$ and $\boldsymbol{\alpha}_k^j$ is a positive column vector in $\mathbb{R}^{n_{\text{bin}}}$, with $\varepsilon_\alpha \le \boldsymbol{\alpha}_k^j[i] \le 1$ for all bins.

The matrix $\boldsymbol{Q}_k^j$ satisfies (CS1) due to (8). If $\boldsymbol{\Theta}[\ell] = 0$, then the off-diagonal element satisfies $\boldsymbol{Q}_k^j[i,\ell] = 0$ and the matrix $\boldsymbol{Q}_k^j$ satisfies (CS2).

We now show that the matrix $\boldsymbol{Q}_k^j$ satisfies (LP1):

$$\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{Q}_k^j[i,\ell] = \tfrac{\xi_k^j\boldsymbol{\alpha}_k^j[i]}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\sum_{\ell=1}^{n_{\text{bin}}}\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell] + 1 - \xi_k^j\boldsymbol{\alpha}_k^j[i] = 1\,,$$

where $\sum_{\ell=1}^{n_{\text{bin}}}\left(\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]\right) = \boldsymbol{\Theta}\boldsymbol{\alpha}_k^j$. We now prove that the matrix $\boldsymbol{Q}_k^j$ satisfies (LP2):

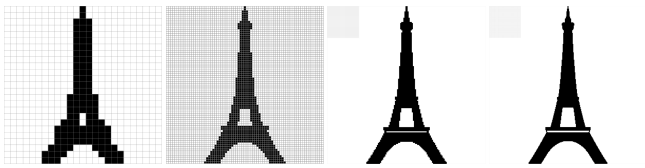
$$\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i]\boldsymbol{Q}_k^j[i,\ell] = \frac{\xi_k^j\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\sum_{i=1}^{n_{\text{bin}}}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\Theta}[i]\right) + \left(\boldsymbol{\Theta}[\ell] - \xi_k^j\boldsymbol{\Theta}[\ell]\boldsymbol{\alpha}_k^j[\ell]\right) = \boldsymbol{\Theta}[\ell]\,.$$

The matrix $\boldsymbol{Q}_k^j$ satisfies (LP3) because each diagonal element $\boldsymbol{Q}_k^j[i,i]$ is lower bounded by $\left(1-\xi_k^j\boldsymbol{\alpha}_k^j[i]\right)$, which is one of the positive terms in (9). The term $\left(1-\xi_k^j\boldsymbol{\alpha}_k^j[i]\right)$ is lower bounded by $(1-\xi_k^j)$ because $\boldsymbol{\alpha}_k^j[i] \le 1$.

We now prove that the matrix $\boldsymbol{Q}_k^j$ satisfies (LP4). Since the element $\boldsymbol{\Theta}[\ell] > 0$, the off-diagonal element $\boldsymbol{Q}_k^j[i,\ell]$ is upper bounded by $\frac{\xi_k^j}{\varepsilon_M}$ because $\boldsymbol{\alpha}_k^j[i] \le 1$, $\boldsymbol{\Theta}[\ell] \le 1$,

and $\left(\frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\right) \leq \frac{\xi_k^j}{\varepsilon_\alpha} \leq \frac{\xi_k^j}{\varepsilon_M}$. On the other hand, the off-diagonal element $\boldsymbol{Q}_k^j[i,\ell]$ is lower bounded by $\varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]$ because $\boldsymbol{\alpha}_k^j[i] \geq \varepsilon_\alpha$ and $\left(\frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\right) \geq \xi_k^j$ as $0 < \boldsymbol{\Theta}\boldsymbol{\alpha}_k^j \leq 1$. Therefore, $\varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right) \leq \boldsymbol{Q}_k^j[i,\ell]$ because $\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right) < 1$. Thus the matrix $\boldsymbol{Q}_k^j$ satisfies (LP4).

As a result, the feasible set is non-empty and the optimal Markov matrix $\boldsymbol{M}_k^j$ has the desirable properties discussed in Remark 2. ■



(a) $\boldsymbol{\nu}_{25}$ (25^2 bins) (b) $\boldsymbol{\nu}_{75}$ (75^2 bins) (c) $\boldsymbol{\nu}_{150}$ (150^2 bins) (d) $\boldsymbol{\nu}_{300}$ (300^2 bins)

Figure 6. Multiresolution images of the Eiffel Tower are shown, where the spatial resolution increases from (a) to (d). All the bins are shown in (a) and (b), whereas only a few bins are shown in the left-top corner in (c) and (d).

Remark 3 (*Computation time*). Although each agent only needs the row of the Markov matrix $\boldsymbol{M}_k^j$ corresponding to its present bin, it has to solve the entire LP (7). The computation time for an LP increases with the increasing number of bins because the number of variables in $\boldsymbol{M}_k^j$ is approximately equal to n_{bin}^2. For example, if the desired formation is given by $\boldsymbol{\nu}_{25}$ or $\boldsymbol{\nu}_{75}$ in Fig. 6, then the computation time is a few minutes on a standard desktop computer. If the desired formation is given by $\boldsymbol{\nu}_{150}$ (with 5×10^8 variables) or $\boldsymbol{\nu}_{300}$ (with 8×10^9 variables), then the LP is impractical for real-time computation. □

Therefore, we need a faster method for computing the Markov matrices. Corollary 1 gives the closed-form optimal Markov matrix, if the cost matrix is symmetric.

Corollary 1. The optimal Markov matrix of the LP (7) in Method 1 is given by

$$\boldsymbol{M}_k^j[i,\ell] = \begin{cases} 0 & \text{if } \boldsymbol{A}_k^j[i,\ell] = 0 \\ \varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right) & \text{otherwise} \end{cases}, \quad \forall i,\ell \in \{1,\ldots,n_{\text{bin}}\} \text{ and } i \neq \ell\,, \tag{10}$$

$$\boldsymbol{M}_k^j[i,i] = 1 - \sum_{\ell \in \{1,\ldots,n_{\text{bin}}\}\setminus\{i\}} \boldsymbol{M}_k^j[i,\ell]\,. \tag{11}$$

if the cost matrix $\boldsymbol{C}_k$ is symmetric (i.e., $\boldsymbol{C}_k = \boldsymbol{C}_k^T$).

Proof: The original LP (7) can be simplified by neglecting the constraints (LP1,2) and using the following substitutions for all positive elements $\boldsymbol{R}_k^j[i,i] = \boldsymbol{M}_k^j[i,i] - (1-\xi_k^j)$ and $\boldsymbol{R}_k^j[i,\ell] = \boldsymbol{M}_k^j[i,\ell] - \varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$:

$$\text{minimize} \sum_{i=1}^{n_{\text{bin}}}\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell]\boldsymbol{R}_k^j[i,\ell] + \sum_{i=1}^{n_{\text{bin}}} \sum_{\ell \in \{\boldsymbol{A}_k^j[i,\ell]=1, i\neq\ell\}} \boldsymbol{C}_k[i,\ell]\varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right), \tag{12}$$

subject to:

$$0 \leq \boldsymbol{R}_k^j[i,i] \leq \xi_k^j, \forall i\,, \tag{$\widetilde{\text{LP3}}$}$$

$$0 \leq \boldsymbol{R}_k^j[i,\ell] \leq \frac{\xi_k^j}{\varepsilon_M} - \varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right). \tag{$\widetilde{\text{LP4}}$}$$

According to Definition 5, $\boldsymbol{C}_k[i,i] = 0$ and $\boldsymbol{C}_k[i,\ell] > 0$ for all $i \neq \ell$. The minimum cost of this simpler LP (12) is obtained when $\sum_{i=1}^{n_{\text{bin}}}\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell]\boldsymbol{R}_k^j[i,\ell] = 0$, because the second term in the objective function is a constant. Therefore, all the positive off-diagonal elements $\boldsymbol{M}_k^j[i,\ell]$ are equal to their respective lower bounds $\varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ in the optimal solution of the simpler LP (12). This optimal solution $\boldsymbol{M}_k^j$ of the simpler LP (12) is given by (10)-(11).

If the optimal solution of the simpler LP (12) also satisfies the constraints (LP1,2) that we neglected previously, then it is the optimal solution of the original LP (7). It follows from the construction of the diagonal elements in $\boldsymbol{M}_k^j$ (11) that it satisfies (LP1). The diagonal elements of $\boldsymbol{M}_k^j$ are given by

$$\boldsymbol{M}_k^j[i,i] = 1 - \sum_{\ell \in \{\boldsymbol{A}_k^j[i,\ell]=1, i\neq\ell\}} \varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right). \tag{13}$$

Note that the matrix $\boldsymbol{M}_k^j$ is a reversible Markov matrix because of the symmetric cost matrix, i.e., $\boldsymbol{\Theta}[i]\boldsymbol{M}_k^j[i,\ell] = \boldsymbol{\Theta}[\ell]\boldsymbol{M}_k^j[\ell,i] = \varepsilon_M \xi_k^j \boldsymbol{\Theta}[i]\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ for all i,ℓ. Hence, the matrix $\boldsymbol{M}_k^j$ also satisfies (LP2) because

$$\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i]\boldsymbol{M}_k^j[i,\ell] = \sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[\ell]\boldsymbol{M}_k^j[\ell,i] = \boldsymbol{\Theta}[\ell]\left(\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{M}_k^j[\ell,i]\right) = \boldsymbol{\Theta}[\ell]\,.$$

Therefore, the matrix $\boldsymbol{M}_k^j$ is the optimal solution of the original LP (7). ■

If the cost matrix $\boldsymbol{C}_k$ is symmetric, using (10)-(11) saves significant computational time because each agent can directly compute its row of the optimal Markov matrix $\boldsymbol{M}_k^j$. For example, if the desired formation is given by $\boldsymbol{\nu}_{300}$ (in Fig. 6, with 300×300 bins), then the computation time for a single row is less than a second and that of the entire Markov matrix is less than 2 minutes on a standard desktop computer.

Remark 4 (*Alternative constraints*). Note that our construction technique holds even if the term $\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ in the constraint (LP4) in Method 1 and (10) in Corollary 1 is replaced by any monotonic function in $(0,1]$ that decreases with an increasing $\boldsymbol{C}_k[i,\ell]$. Similarly, the term ξ_k^j in the constraints (LP3,4) can be replaced by any monotonic function in $(0,1]$ that decreases with a decreasing ξ_k^j. For example, see Fig. 12(b) in Section VI-B. □

B. Construction of Fastest Mixing IMC

In this subsection, we construct the fastest mixing IMC whose convergence rate to the rank one matrix $\mathbf{1}\boldsymbol{\Theta}$ is optimized. The convergence rate of HMC, with time-invariant Markov matrix $\boldsymbol{M}$, is determined by the second largest

eigenvalue modulus (i.e., $\max_{r\in\{2,\dots,n_{\text{bin}}\}}|\lambda_r(\boldsymbol{M})|$) [59], [60]. On the other hand, the convergence rate of IMC is determined by the coefficient of ergodicity [47, pp. 137]. Since the first n_{rec} bins are recurrent bins, the Markov matrix $\boldsymbol{M}_k^j$ can be decomposed as follows:

$$\boldsymbol{M}_k^j = \begin{bmatrix} \boldsymbol{M}_{k,\text{sub}}^j & \mathbf{0}^{n_{\text{rec}}\times(n_{\text{bin}}-n_{\text{rec}})} \\ \boldsymbol{M}_k^j[n_{\text{rec}}+1:n_{\text{bin}},1:n_{\text{rec}}] & \boldsymbol{M}_k^j[n_{\text{rec}}+1:n_{\text{bin}},n_{\text{rec}}+1:n_{\text{bin}}] \end{bmatrix}, \quad (14)$$

where $\boldsymbol{M}_{k,\text{sub}}^j := \boldsymbol{M}_k^j[1:n_{\text{rec}},1:n_{\text{rec}}]$ encapsulates the bin transition probabilities between the recurrent bins.

Definition 9 (*Coefficient of ergodicity* [47, pp. 137–139]). For the stochastic matrix $\boldsymbol{M}_{k,\text{sub}}^j$, the coefficient of ergodicity $\tau_1(\boldsymbol{M}_{k,\text{sub}}^j)$ is defined as

$$\begin{aligned}\tau_1(\boldsymbol{M}_{k,\text{sub}}^j) &= \sup_{\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_1\neq\boldsymbol{v}_2} \frac{D_{\mathcal{L}_1}(\boldsymbol{v}_1\boldsymbol{M}_{k,\text{sub}}^j,\boldsymbol{v}_2\boldsymbol{M}_{k,\text{sub}}^j)}{D_{\mathcal{L}_1}(\boldsymbol{v}_1,\boldsymbol{v}_2)}, \\ &= 1-\min_{i,\ell}\sum_{s=1}^{n_{\text{rec}}}\min\left(\boldsymbol{M}_{k,\text{sub}}^j[i,s],\boldsymbol{M}_{k,\text{sub}}^j[\ell,s]\right), \quad (15)\end{aligned}$$

where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are probability row vectors in $\mathbb{R}^{n_{\text{rec}}}$ and $i,\ell,s\in\{1,\dots,n_{\text{rec}}\}$. □

We define $n_{k,\text{dia}}^j$ as the graph diameter in the graph conforming to the matrix $\boldsymbol{A}_{k,\text{sub}}^j := \boldsymbol{A}_k^j[1:n_{\text{rec}},1:n_{\text{rec}}]$; i.e., it is the greatest number of edges in the shortest path between any pair of recurrent bins [61]. If $n_{k,\text{dia}}^j>2$, then there exist recurrent bins $B[i]$ and $B[\ell]$ such that either $\boldsymbol{M}_{k,\text{sub}}^j[i,s]=0$ or $\boldsymbol{M}_{k,\text{sub}}^j[\ell,s]=0$ for all $s\in\{1,\dots,n_{\text{rec}}\}$. Substituting these bins into (15) shows that $\tau_1(\boldsymbol{M}_{k,\text{sub}}^j)=1$ when $n_{k,\text{dia}}^j>2$. In order to avoid this trivial case, we choose to minimize the coefficient of ergodicity of the positive matrix $(\boldsymbol{M}_{k,\text{sub}}^j)^{n_{k,\text{dia}}^j}$ [62, Theorem 8.5.2, pp. 516].

Corollary 2 (*Construction of fastest-mixing Markov matrix*). The following convex optimization problem is used instead of the LP (7) in Method 1:

$$\min \tau_1\left((\boldsymbol{M}_{k,\text{sub}}^j)^{n_{k,\text{dia}}^j}\right), \quad (16)$$

subject to $(\text{LP}1-4)$ in (7), where τ_1 is defined in Definition 9.

Proof: The objective function $\tau_1\left((\boldsymbol{M}_{k,\text{sub}}^j)^{n_{k,\text{dia}}^j}\right)$ is a convex function of the stochastic matrix $\boldsymbol{M}_{k,\text{sub}}^j$ because it can be expressed as follows [47, Lemma 4.3, pp. 139]:

$$\tau_1\left((\boldsymbol{M}_{k,\text{sub}}^j)^{n_{k,\text{dia}}^j}\right) = \sup_{\|\boldsymbol{\delta}\|_2=1,\,\boldsymbol{\delta}\mathbf{1}=0}\left\|\boldsymbol{\delta}\cdot(\boldsymbol{M}_{k,\text{sub}}^j)^{n_{k,\text{dia}}^j}\right\|_1,$$

where $\boldsymbol{\delta}=\frac{(\boldsymbol{v}_1-\boldsymbol{v}_2)}{\|\boldsymbol{v}_1-\boldsymbol{v}_2\|_2}$ is a row vector in $\mathbb{R}^{n_{\text{rec}}}$. Hence, (16) is a convex optimization problem and the family of Markov matrices $\boldsymbol{Q}_k^j$ (8)-(9) is a subset of its feasible set. ■

C. Condition for Escaping Transient Bins

Here the condition for escaping transient bins is presented.

Definition 10 (*Trapping bins*). If an agent is inside a transient bin $B[i]$ (with $\boldsymbol{\Theta}[i]=0$) and its motion constraint matrix $\boldsymbol{A}_k^j$ only allows transitions to other transient bins at the k^{th} time instant, then this transient bin $B[i]$ is called a trapping bin. This agent is trapped in the bin $B[i]$ because the Markov matrix $\boldsymbol{M}_k^j$ does not allow transitions out of this bin. Let $\boldsymbol{\mathcal{T}}_k^j$ represent the set of trapping bins for the j^{th} agent at the k^{th} time instant. For example, see Fig. 7. □

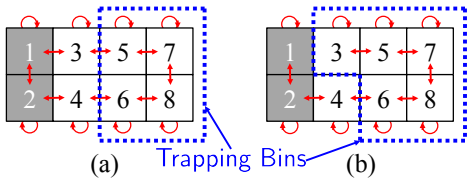


Figure 7. In this example, the bins 1 and 2 are recurrent bins. The allowed transitions (motion constraints) are shown in red. The trapping bins for the two cases are enclosed in blue.

Since the irreducible motion constraint matrices $\boldsymbol{A}_k^j$ are known a priori (Assumption 4 and Definition 4), the deterministic path for exiting the set of trapping bins is stored on board each agent. For each trapping bin $B[i]\in\boldsymbol{\mathcal{T}}_k^j$, the j^{th} agent transitions to a transient bin $\Psi_k^j[i]$, chosen a priori, such that the transition from bin $B[i]$ to bin $\Psi_k^j[i]$ is allowed by motion constraints. This deterministic path ensures that the agent exits the set of trapping bins, using multiple transitions, as soon as possible. This bin $\Psi_k^j[i]$ has to be chosen on a case-by-case basis depending on the motion constraints matrix $\boldsymbol{A}_k^j$. For example, in Fig. 7, for the trapping bin 5, the best option is bin 3 in case (a) and bin 7 in case (b). Therefore, the agent can follow this path to deterministically exit the set of trapping bins in finite time instants.

If an agent is in a transient bin, but not in a trapping bin, then its motion constraint matrix allows transitions to some recurrent bins. We can speed up the process of exiting this transient bin by forcing the agent to transition to any reachable recurrent bin, with equal probability, during the current time instant. Thus the agent transitions from its current transient bin to a recurrent bin in one time instant.

The matrix $\boldsymbol{S}_k^j\in\mathbb{R}^{n_{\text{bin}}\times n_{\text{bin}}}$ encapsulates the condition for escaping transient bins. If $B[i]$ is a transient bin (i.e., $\boldsymbol{\Theta}[i]=0$), then each element in the corresponding row $\boldsymbol{S}_k^j[i,1:n_{\text{bin}}]$ is given by

$$\boldsymbol{S}_k^j[i,\ell]=\begin{cases}1 & \textbf{if } B[i]\in\boldsymbol{\mathcal{T}}_k^j \text{ and } B[\ell]=\Psi_k^j[i]\\ \frac{1}{n_{k,i}^j} & \textbf{if } B[i]\notin\boldsymbol{\mathcal{T}}_k^j \text{ and } \boldsymbol{A}_k^j[i,\ell]=1 \\ & \text{and } \boldsymbol{\Theta}[\ell]>0\\ 0 & \textbf{otherwise}\end{cases}, \quad (17)$$

where $n_{k,i}^j$ is the number of recurrent bins that the j^{th} agent can transition to, from bin $B[i]$ at the k^{th} time instant. This condition is used only if the agent is in a transient bin, as shown in Method 1. In Section IV-A, we show that the agent exits the set of transient bins within finite time instants due to this condition.

IV. PSG–IMC for Shape Formation

The PSG–IMC algorithm for shape formation is precisely defined by using the results of the previous sections and its properties of convergence and robustness are elucidated.

The pseudo-code of the PSG–IMC algorithm for shape formation is given in Method 2, whose key steps are shown in

Method 2: PSG–IMC for Shape Formation

1: One iteration during k^{th} time instant for j^{th} agent located in bin $B[i]$
2: Given $\boldsymbol{\Theta}$, $\boldsymbol{C}_k$, $\boldsymbol{A}_k^j$, $\boldsymbol{\mu}_k^j$, ε_M, ε_C, τ^j, and β^j
3: **if** $\boldsymbol{\Theta}[i] = 0$, **then**
4: Compute $\boldsymbol{S}_k^j[i, 1\!:\!n_{\text{bin}}]$ using (17)
5: Sample a random number $z \in \text{unif}[0, 1]$
6: Select bin $B[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{S}_k^j[i,\ell] \le z < \sum_{\ell=1}^{q} \boldsymbol{S}_k^j[i,\ell]$
7: **else**
8: Compute the feedback error ξ_k^j using (4)
9: Compute $\boldsymbol{M}_k^j[i, 1\!:\!n_{\text{bin}}]$ using Corollary 1 or compute $\boldsymbol{M}_k^j$ using Method 1
10: Compute the term $\eta_{k,i}^j$ using (20)
11: Compute $\boldsymbol{P}_k^j[i, 1\!:\!n_{\text{bin}}]$ using (18) and (19)
12: Sample a random number $z \in \text{unif}[0; 1]$
13: Select bin $B[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{P}_k^j[i,\ell] \le z < \sum_{\ell=1}^{q} \boldsymbol{P}_k^j[i,\ell]$
14: **end if**
15: Go to bin $B[q]$ while avoiding collision

Fig. 4. At the start, the j^{th} agent knows the desired formation shape $\boldsymbol{\Theta}$, the time-varying cost matrix $\boldsymbol{C}_k$, and its time-varying motion constraint matrix $\boldsymbol{A}_k^j$ (Assumption 4). During each iteration, the agent determines the bin it belongs to (Assumption 3) and the current swarm distribution $\boldsymbol{\mu}_k^j$ from Section II-A (lines 1–2).

If the agent is in a transient bin (line 3), then it uses inverse transform sampling (Remark 14 in Appendix) to select the next bin $B[q]$ from the corresponding row of the matrix $\boldsymbol{S}_k^j[i, 1\!:\!n_{\text{bin}}]$ (lines 4–6). Otherwise, the agent first computes the HD-based feedback error ξ_k^j (line 8). If the cost matrix $\boldsymbol{C}_k$ is symmetric, then the agent can directly compute the row $\boldsymbol{M}_k^j[i, 1\!:\!n_{\text{bin}}]$ using Corollary 1 (line 9). Otherwise, the agent computes the Markov matrix $\boldsymbol{M}_k^j$ using Method 1 (line 9).

In order to avoid undesirable transitions from bins that are deficient in agents (i.e., where $\boldsymbol{\Theta}[i] > \boldsymbol{\mu}_k^j[i]$), the agent modifies its Markov matrix row $\boldsymbol{M}_k^j[i, 1\!:\!n_{\text{bin}}]$ as follows:

$$\boldsymbol{P}_k^j[i,\ell] = \left(1 - \eta_{k,i}^j\right) \boldsymbol{M}_k^j[i,\ell]\,, \qquad \forall i \neq \ell \tag{18}$$

$$\boldsymbol{P}_k^j[i,i] = \left(1 - \eta_{k,i}^j\right) \boldsymbol{M}_k^j[i,i] + \eta_{k,i}^j\,, \tag{19}$$

$$\text{where } \eta_{k,i}^j = \exp(-\tau^j k) \frac{\exp\left(\beta^j(\boldsymbol{\Theta}[i] - \boldsymbol{\mu}_k^j[i])\right)}{\exp\left(\beta^j|\boldsymbol{\Theta}[i] - \boldsymbol{\mu}_k^j[i]|\right)}\,, \tag{20}$$

and τ^j and β^j are time-invariant positive constants (lines 10–11). Then, the agent uses inverse transform sampling (Remark 14 in Appendix) to select the next bin $B[q]$ from the bin transition probabilities $\boldsymbol{P}_k^j[i, 1\!:\!n_{\text{bin}}]$ (lines 12–13). Finally, the agent goes to the selected bin $B[q]$ from the current bin $B[i]$ in a collision-free manner using any lower-level guidance and control algorithm (see Remarks 9 and 15) (line 15).

Remark 5 (*Effects of* $\eta_{k,i}^j$). The term $\eta_{k,i}^j$ (20) greatly lowers the transition probability of a bin that is deficient in agents; i.e., if $\boldsymbol{\Theta}[i] > \boldsymbol{\mu}_k^j[i]$, then $\eta_{k,i}^j = \exp(-\tau^j k)$. Its effect decreases with increasing time instants. The design variables β^j and τ^j dictate the amplitude and time constant of this suppression. It is shown later in Section VI-A that the undesirable transitions, suppressed using this term, greatly reduce the total number of transitions, which in turn significantly improves the convergence error. If $\boldsymbol{\Theta}[i] \le \boldsymbol{\mu}_k^j[i]$, then the term $\eta_{k,i}^j$ becomes very small and its effect is negligible. □

Remark 6. Under Assumptions 3 and 5, the agent determines to which bin it belongs and estimates the current swarm distribution in a distributed manner. The remaining terms in lines 1–2 are known a priori. Lines 3–14 are executed individually by each agent. Finally, in line 15, the agent only needs to communicate with its neighboring agents as shown in Remark 15 in Appendix. Thus, all the steps in Method 2 can be accomplished in a distributed manner. □

A. Main Result: Convergence Analysis

In this subsection, we prove that the swarm distribution $\boldsymbol{\mu}_k^\star$ converges to the desired formation shape $\boldsymbol{\Theta}$ with prescribed convergence errors using the PSG–IMC algorithm for shape formation given in Method 2. Unlike the convergence proof for HMC, which is a direct application of the Perron–Frobenius theorem, the convergence proof for IMC is rather involved (e.g., see [47], [63]). We first state an assumption on ξ_k^j.

Assumption 6 (*Minimum value of* ξ_k^j). If $\xi_k^j < \xi_{\text{des}}$, then the current swarm distribution is sufficiently close to the desired formation (see Definition 6). Moreover, the Markov matrix in Method 1 becomes the identity matrix; hence the agents do not transition any more. The swarm has converged to the desired formation and no further convergence is necessary. Therefore, in this subsection, $\xi_k^j \ge \xi_{\text{des}}$ is assumed, indicating that the swarm has not converged to the desired shape. □

We first show that agents in recurrent bins transition using the following modified Markov matrix $\boldsymbol{P}_k^j$.

Theorem 2. According to Method 2, if an agent is in a recurrent bin, then it transitions using the following modified Markov matrix $\boldsymbol{P}_k^j$ from (18)–(19):

$$\boldsymbol{P}_k^j = \left(\mathbf{I} - \boldsymbol{D}_k^j\right) \boldsymbol{M}_k^j + \boldsymbol{D}_k^j\,, \tag{21}$$

where $\boldsymbol{D}_k^j = \text{diag}\left(\eta_{k,1}^j, \ldots, \eta_{k,n_{\text{bin}}}^j\right)$. The Markov matrix $\boldsymbol{P}_k^j$ is row stochastic (i.e., $\boldsymbol{P}_k^j \mathbf{1} = \mathbf{1}$), asymptotically homogeneous with respect to $\boldsymbol{\Theta}$ (i.e., $\lim_{k\to\infty} \boldsymbol{\Theta}\boldsymbol{P}_k^j = \boldsymbol{\Theta}$), and only allows transitions into recurrent bins.

Proof: The modified Markov matrix $\boldsymbol{P}_k^j$ (21) is derived from (18)–(20). It follows from lines 3 and 6 of Method 2 that the agent uses the Markov matrix $\boldsymbol{P}_k^j$ to transition if and only if it is in a recurrent bin (i.e., $\boldsymbol{\Theta}[i] > 0$).

The matrix $\boldsymbol{P}_k^j$ is row stochastic because $\boldsymbol{M}_k^j \mathbf{1} = \mathbf{1}$. The matrix $\boldsymbol{M}_k^j$ has $\boldsymbol{\Theta}$ as its stationary distribution for all $k \in \mathbb{N}$. It follows from the definition of the term $\eta_{k,i}^j$ (20) that $\lim_{k\to\infty} \boldsymbol{D}_k^j = \mathbf{0}^{n_{\text{bin}} \times n_{\text{bin}}}$, because $\lim_{k\to\infty} \exp(-\tau^j k) = 0$. Therefore $\lim_{k\to\infty} \boldsymbol{P}_k^j = \lim_{k\to\infty} \boldsymbol{M}_k^j$. Hence, the sequence of matrices $\boldsymbol{P}_k^j$ is asymptotically homogeneous with respect to $\boldsymbol{\Theta}$ because $\lim_{k\to\infty} \boldsymbol{\Theta}\boldsymbol{P}_k^j = \lim_{k\to\infty} \boldsymbol{\Theta}\boldsymbol{M}_k^j = \boldsymbol{\Theta}$ (see Definition 13 in Appendix).

Note that the element $\boldsymbol{P}^j_k[i,\ell] > 0$ if and only if the corresponding element $\boldsymbol{M}^j_k[i,\ell] > 0$ for all $i,\ell \in \{1,\dots,n_{\text{bin}}\}$ and $k \in \mathbb{N}$. Therefore, like the matrix $\boldsymbol{M}^j_k$, the matrix $\boldsymbol{P}^j_k$ only allows transitions into recurrent bins. ■

We now show that all the agents leave the transient bins and enter the recurrent bins in finite time instants.

Theorem 3. According to Method 2, each agent is in a recurrent bin by the T^{th} time instant, where $T \leq (n_{\text{bin}} - n_{\text{rec}} + 1)$. Once an agent is inside a recurrent bin, it always remains within the set of recurrent bins.

Proof: If an agent is in a recurrent bin, then it follows from Theorem 2 that it cannot transition to any transient bin.

If the agent is in a trapping bin, then the matrix $\boldsymbol{S}^j_k$ (17) ensures that the agent exits the set of trapping bins as soon as possible in a deterministic manner. Therefore, the maximum number of steps inside the set of trapping bins is upper bounded by the number of transient bins $(n_{\text{bin}} - n_{\text{rec}})$.

If an agent is in a transient bin, but not in a trapping bin, then the matrix $\boldsymbol{S}^j_k$ (17) ensures that the agent transitions to a recurrent bin in one time instant. Hence each agent enters a recurrent bin in at most $(n_{\text{bin}} - n_{\text{rec}} + 1)$ time instants. ■

Consider a probability (row) vector $\boldsymbol{x}^j_k \in \mathbb{R}^{n_{\text{bin}}}$, which denotes the probability mass function (PMF) of the predicted bin position of the j^{th} agent at the k^{th} time instant. Each element $\boldsymbol{x}^j_k[i]$ gives the probability that the j^{th} agent is in bin $B[i]$ at the k^{th} time instant:

$$\boldsymbol{x}^j_k[i] = \mathbb{P}(\boldsymbol{r}^j_k[i] = 1), \quad \forall i \in \{1,\dots,n_{\text{bin}}\}\,. \tag{22}$$

We now discuss convergence of each agent's predicted bin position $\boldsymbol{x}^j_k$ to the desired formation $\boldsymbol{\Theta}$.

Theorem 4. The vector $\boldsymbol{x}^j_k$ of each agent executing Method 2 converges pointwise to the desired stationary distribution $\boldsymbol{\Theta}$ irrespective of the initial condition, i.e., $\lim_{k\to\infty} \boldsymbol{x}^j_k = \boldsymbol{\Theta}$ pointwise for all agents.

Proof: It follows from Theorem 3 that all agents are always in the set of recurrent bins from the T^{th} time instant onwards. Since the first n_{rec} bins are recurrent bins, we decompose the vector $\boldsymbol{x}^j_k = [\bar{\boldsymbol{x}}^j_k,\, 0,\dots,0]$ for all $k \geq T$, where the probability row vector $\bar{\boldsymbol{x}}^j_k := [\boldsymbol{x}^j_k[1],\dots,\boldsymbol{x}^j_k[n_{\text{rec}}]] \in \mathbb{R}^{n_{\text{rec}}}$ denotes the agent's PMF over the set of recurrent bins. Similarly, we decompose $\boldsymbol{\Theta} = [\bar{\boldsymbol{\Theta}},\, 0,\dots,0]$, where $\bar{\boldsymbol{\Theta}} := [\boldsymbol{\Theta}[1],\dots,\boldsymbol{\Theta}[n_{\text{rec}}]]$. Note that convergence of $\bar{\boldsymbol{x}}^j_k$ to $\bar{\boldsymbol{\Theta}}$, implies the convergence of $\boldsymbol{x}^j_k$ to $\boldsymbol{\Theta}$.

According to Theorem 2, the time evolution of the PMF vector $\bar{\boldsymbol{x}}^j_k$ is given by

$$\bar{\boldsymbol{x}}^j_{k+1} = \bar{\boldsymbol{x}}^j_k \boldsymbol{P}^j_{k,\text{sub}}\,, \qquad \forall k \geq T\,, \tag{23}$$

where the row-stochastic sub-matrix $\boldsymbol{P}^j_{k,\text{sub}} := \boldsymbol{P}^j_k[1:n_{\text{rec}}, 1:n_{\text{rec}}]$ encapsulates the bin transition probabilities between the recurrent bins. The matrix $\boldsymbol{P}^j_{k,\text{sub}}$, like the matrix $\boldsymbol{M}^j_{k,\text{sub}}$ in (14), is irreducible because the matrix $\boldsymbol{A}^j_{k,\text{sub}}$ is irreducible (Definition 4).

It follows from (4) that $D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}^j_k) = 1$ if and only if $\boldsymbol{\mu}^j_k[i] = 0$ for all recurrent bins $i \in \{1,\dots,n_{\text{rec}}\}$, because $\boldsymbol{\Theta}[i] > 0$ only in recurrent bins (Definition 2). Therefore, the feedback error $\xi^j_k < 1$ for all time instant $k \geq T$ because all agents are in recurrent bins. Hence, the diagonal elements $\boldsymbol{M}^j_{k,\text{sub}}[i,i]$ and $\boldsymbol{P}^j_{k,\text{sub}}[i,i]$ are positive for all $i \in \{1,\dots,n_{\text{rec}}\}$ and $k \geq T$ due to constraint (LP3) in Method 1.

The overall time evolution of the agent's PMF vector for all $r > T$ is given by the IMC

$$\bar{\boldsymbol{x}}^j_r = \bar{\boldsymbol{x}}^j_T \boldsymbol{P}^j_{T,\text{sub}} \boldsymbol{P}^j_{T+1,\text{sub}} \cdots \boldsymbol{P}^j_{r-1,\text{sub}} = \bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r}\,. \tag{24}$$

We now show that this forward matrix product $\boldsymbol{U}^j_{T,r}$ is strongly ergodic (see Definition 14 in Appendix) and $\bar{\boldsymbol{\Theta}}$ is its unique limit vector (i.e., $\lim_{r\to\infty} \boldsymbol{U}_{T,r} = \mathbf{1}\bar{\boldsymbol{\Theta}}$).

The matrix $\boldsymbol{U}^j_{T,r}$ is a product of nonnegative matrices, hence it is a nonnegative matrix. If $\boldsymbol{P}^j_{k,\text{sub}}[i,\ell] > 0$ for some $k \in \{T,\dots,r-1\}$ and $i,\ell \in \{1,\dots,n_{\text{rec}}\}$, then the corresponding element $\boldsymbol{U}^j_{T,r}[i,\ell] > 0$ because, as shown below, the value of $\boldsymbol{U}^j_{T,r}[i,\ell]$ is lower bounded by the product of positive diagonal elements and $\boldsymbol{P}^j_{k,\text{sub}}[i,\ell]$:

$$\begin{aligned}
\boldsymbol{U}^j_{T,r}[i,\ell] &\geq \boldsymbol{P}^j_{T,\text{sub}}[i,\ell] \Big(\prod_{q=T+1}^{r-1} \boldsymbol{P}^j_{q,\text{sub}}[\ell,\ell] \Big) \\
&+ \sum_{s=T+1}^{r-2} \left(\Big(\prod_{q=T}^{s-1} \boldsymbol{P}^j_{q,\text{sub}}[i,i] \Big) \boldsymbol{P}^j_{s,\text{sub}}[i,\ell] \Big(\prod_{q=s+1}^{r-1} \boldsymbol{P}^j_{q,\text{sub}}[\ell,\ell] \Big) \right) \\
&+ \Big(\prod_{q=T}^{r-2} \boldsymbol{P}^j_{q,\text{sub}}[i,i] \Big) \boldsymbol{P}^j_{r-1,\text{sub}}[i,\ell]\,, \qquad \text{if } i \neq \ell\,, \qquad (25)
\end{aligned}$$

$$\boldsymbol{U}^j_{T,r}[i,i] \geq \Big(\prod_{q=T}^{r-1} \boldsymbol{P}^j_{q,\text{sub}}[i,i] \Big)\,, \qquad \text{if } i = \ell\,. \tag{26}$$

Therefore, the matrix $\boldsymbol{U}^j_{T,r}$, like the matrix $\boldsymbol{P}^j_{k,\text{sub}}$, is irreducible because $\boldsymbol{U}^j_{T,r}[i,\ell] > 0$ if $\boldsymbol{P}^j_{k,\text{sub}}[i,\ell] > 0$ for all $i,\ell \in \{1,\dots,n_{\text{rec}}\}$. Since the irreducible matrix $\boldsymbol{U}^j_{T,r}$ has positive diagonal elements (26), it is a primitive matrix [62, Lemma 8.5.4, pp. 516].

Some of the off-diagonal elements in $\boldsymbol{M}^j_{k,\text{sub}}$ and $\boldsymbol{P}^j_{k,\text{sub}}$ are zero due to the constraints (CS1,2) in Method 1. The lower bound γ^j, which is independent of k, for the remaining positive elements in $\boldsymbol{P}^j_{k,\text{sub}}$ is given by the constraint (LP4) in Method 1, the lower bound of ξ^j_k in Assumption 6, and the upper bound of the term $\eta^j_{k,i}$ (20):

$$\begin{aligned}
\gamma^j &= \big(1 - \exp(-\tau^j T)\big)\, \xi_{\text{des}} \varepsilon_M (\min{}^+ \boldsymbol{\Theta}) \left(1 - \tfrac{C_{\max}}{C_{\max} + \varepsilon_C}\right) \\
&\leq \min_{i,\ell}{}^+ \boldsymbol{P}^j_{k,\text{sub}}[i,\ell]\,, \qquad k \geq T\,, \qquad (27)
\end{aligned}$$

where $\min^+$ refers to the minimum of the positive elements and $C_{\max} = \max_{k\in\mathbb{N}} C_{k,\max}$. It follows from Theorem 2 that the sequence of matrices $\boldsymbol{P}^j_{k,\text{sub}}$, $k \geq T$ is asymptotically homogeneous with respect to $\bar{\boldsymbol{\Theta}}$. Since (i) the forward matrix product $\boldsymbol{U}_{T,r}$ is primitive and (ii) there exists γ^j (independent of k), it follows from Theorem 8 in Appendix that the forward matrix product $\boldsymbol{U}^j_{T,r}$ is strongly ergodic. Since (i) the matrices $\boldsymbol{P}^j_{k,\text{sub}}$, $k \geq T$ are irreducible and (ii) there exists γ^j (independent of k), it follows from Theorem 9 in Appendix that the limit vector $\boldsymbol{e} = \bar{\boldsymbol{\Theta}}$. Since (iii) $\boldsymbol{U}^j_{T,r}$ is strongly

ergodic, it follows from Corollary 3 in Appendix that the unique limit vector is given by $\bar{\boldsymbol{\Theta}}$ (i.e., $\lim_{r\to\infty} \boldsymbol{U}^j_{T,r} = \mathbf{1}\bar{\boldsymbol{\Theta}}$). Hence, each agent's PMF vector converges to

$$\lim_{r\to\infty} \bar{\boldsymbol{x}}^j_r = \lim_{r\to\infty} \bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r} = \bar{\boldsymbol{x}}^j_T \mathbf{1}\bar{\boldsymbol{\Theta}} = \bar{\boldsymbol{\Theta}}\,.$$

Therefore, $\lim_{k\to\infty} \boldsymbol{x}^j_k = \boldsymbol{\Theta}$ pointwise for all agents. ∎

Theorem 5. Since $\lim_{r\to\infty} \boldsymbol{U}^j_{T,r} = \mathbf{1}\bar{\boldsymbol{\Theta}}$, for all $\varepsilon_{\rm lim} > 0$, there exists a $k^j_{\epsilon,\rm lim} \in \mathbb{N}$ such that $D_{\mathcal{L}_1}(\bar{\boldsymbol{\Theta}}\boldsymbol{U}^j_{T,r}, \bar{\boldsymbol{\Theta}}) \le \varepsilon_{\rm lim}$ for all $r \ge k^j_{\epsilon,\rm lim}$. The convergence error between the $j^{\rm th}$ agent's PMF vector $\boldsymbol{x}^j_r$ and the desired formation $\boldsymbol{\Theta}$ is bounded by $D_{\mathcal{L}_1}(\boldsymbol{x}^j_r, \boldsymbol{\Theta}) \le \upsilon^j_k$ for all $r \ge k^j_{\epsilon,\rm lim}$, where

$$\upsilon^j_k = \varepsilon_{\rm lim} + D_{\mathcal{L}_1}(\boldsymbol{x}^j_T, \boldsymbol{\Theta}) \prod_{s=0}^{\rho-1} \left(1 - n_{\rm rec} \left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta^j_q\right)\right),$$

$\rho = \left\lfloor \frac{r-T}{n_{\rm rec}-1} \right\rfloor$, $\lfloor\cdot\rfloor$ is the floor function, and $\delta^j_q = \min_{i,\,\ell} {}^{+}\boldsymbol{P}^j_{q,\rm sub}[i,\ell]$.

Proof: It follows from the definition of $\tau_1(\boldsymbol{U}^j_{T,r})$ in (15) that

$$D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r}, \bar{\boldsymbol{\Theta}}\boldsymbol{U}^j_{T,r}) \le \tau_1(\boldsymbol{U}^j_{T,r})\, D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}^j_T, \bar{\boldsymbol{\Theta}})\,.$$

Since $\bar{\boldsymbol{x}}^j_r = \bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r}$ (24), we get from the triangle inequality:

$$\begin{aligned} D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}^j_r, \bar{\boldsymbol{\Theta}}) &\le D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r}, \bar{\boldsymbol{\Theta}}\boldsymbol{U}^j_{T,r}) + D_{\mathcal{L}_1}(\bar{\boldsymbol{\Theta}}\boldsymbol{U}^j_{T,r}, \bar{\boldsymbol{\Theta}}) \\ &\le \tau_1(\boldsymbol{U}^j_{T,r}) D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}^j_T, \bar{\boldsymbol{\Theta}}) + \varepsilon_{\rm lim}\,. \end{aligned}$$

The sub-multiplicative property of $\tau_1(\boldsymbol{U}^j_{T,r})$ [47, Lemma 4.3, pp. 139] gives

$$\tau_1(\boldsymbol{U}^j_{T,r}) \le \prod_{s=0}^{\rho-1} \tau_1\left(\boldsymbol{U}^j_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)}\right).$$

Here, if $r > T + \rho(n_{\rm rec}-1)$, then we neglect the contribution of the residual term by assuming $\tau_1\left(\boldsymbol{U}^j_{T+\rho(n_{\rm rec}-1),r}\right) = 1$.

The matrix $\boldsymbol{U}^j_{k,k+n_{\rm rec}-1}$, for any $k \ge T$, is a positive matrix because there exists a path of length smaller than $(n_{\rm rec}-1)$ between every two recurrent bins (see Theorem 10 in Appendix). A conservative lower bound on the elements in the positive matrix $\boldsymbol{U}^j_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)}$ is given by the product of the smallest positive elements in all the matrices, i.e., $\boldsymbol{U}^j_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)}[i,\ell] \ge \left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta^j_q\right)$ for all $i,\ell \in \{1,\ldots,n_{\rm rec}\}$. Therefore, it follows from (15) that $\tau_1\left(\boldsymbol{U}^j_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)}\right) \le 1 - n_{\rm rec}\left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta^j_q\right) < 1$. ∎

We now focus on the convergence of the 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.

Theorem 6. Let $\varepsilon_{\rm lim} > 0$, $\varepsilon_{\rm bin} > 0$, and $\varepsilon_{\rm conv} > 0$ represent convergence error thresholds. Let κ denote the latest time instant when an agent is added to or removed from the swarm, i.e., the number of agents $m_k = m_\kappa$ for all $k \ge \kappa$. Since $\lim_{k\to\infty} \boldsymbol{U}^j_{\kappa+T,k} = \mathbf{1}\bar{\boldsymbol{\Theta}}$ for all agents, there exists $k_{\epsilon,\rm lim} \in \mathbb{N}$ such that $D_{\mathcal{L}_1}(\bar{\boldsymbol{\Theta}}\boldsymbol{U}^j_{\kappa+T,k}, \bar{\boldsymbol{\Theta}}) \le \varepsilon_{\rm lim}$ for all $k \ge k_{\epsilon,\rm lim}$ and $j \in \{1,\ldots,m_\kappa\}$.

The convergence error between the swarm distribution $\boldsymbol{\mu}^\star_k$ and the desired formation $\boldsymbol{\Theta}$ is probabilistically bounded for all $k \ge k_{\epsilon,\rm lim}$ by

$$\mathbb{P}\left(D_{\mathcal{L}_1}(\boldsymbol{\mu}^\star_k, \boldsymbol{\Theta}) \ge \varepsilon_{\rm bin} + \upsilon_k\right) \le \frac{n^2_{\rm rec}}{4m_\kappa \varepsilon^2_{\rm bin}}\,, \tag{28}$$

$$\mathbb{P}\left(D_H(\boldsymbol{\mu}^\star_k, \boldsymbol{\Theta}) \ge \frac{1}{\sqrt{2}}\sqrt{\varepsilon_{\rm bin} + \upsilon_k}\right) \le \frac{n^2_{\rm rec}}{4m_\kappa \varepsilon^2_{\rm bin}}\,, \tag{29}$$

where $\upsilon_k \ge \max_{j\in\{1,\ldots,m_\kappa\}} \upsilon^j_k$, $\delta_q = \min_{j\in\{1,\ldots,m_\kappa\}} \delta^j_q$, and

$$\upsilon_k = \varepsilon_{\rm lim} + 2 \prod_{s=0}^{\left\lfloor \frac{k-\kappa-T}{n_{\rm rec}-1} \right\rfloor - 1} \left(1 - n_{\rm rec}\left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta_q\right)\right).$$

Also, δ^j_q and υ^j_k are defined in Theorem 5.

If the number of agents satisfies the inequality

$$m_\kappa \ge \frac{n^2_{\rm rec}}{16 \xi^4_{\rm des} \varepsilon_{\rm conv}}\,, \tag{30}$$

then the HD between the final swarm distribution and the desired formation is probabilistically bounded by $\varepsilon_{\rm conv}$, i.e.,

$$\mathbb{P}\left(D_H\left(\lim_{k\to\infty} \boldsymbol{\mu}^\star_k, \boldsymbol{\Theta}\right) \ge \xi_{\rm des}\right) \le \varepsilon_{\rm conv}\,, \tag{31}$$

where $\xi_{\rm des}$ is the desired convergence error defined in Definition 6. Similarly, if the number of agents satisfies the inequality

$$m_\kappa \ge \frac{n^2_{\rm rec}}{4 \xi^2_{\rm des} \varepsilon_{\rm conv}}\,, \tag{32}$$

$$\text{then} \quad \mathbb{P}\left(D_{\mathcal{L}_1}\left(\lim_{k\to\infty} \boldsymbol{\mu}^\star_k, \boldsymbol{\Theta}\right) \ge \xi_{\rm des}\right) \le \varepsilon_{\rm conv}\,. \tag{33}$$

Proof: Let $X^j_{k,i}$ denote the Bernoulli random variable, where $X^j_{k,i} = 1$ represents the event that the $j^{\rm th}$ agent is actually located in bin $B[i]$ at the $k^{\rm th}$ time instant (i.e., $\boldsymbol{r}^j_k[i] = 1$) and $X^j_{k,i} = 0$ otherwise (i.e., $\boldsymbol{r}^j_k[i] = 0$). We get from (22) that $\mathbb{P}(X^j_{k,i} = 1) = \boldsymbol{x}^j_k[i]$. Therefore $\mathbb{E}[X^j_{k,i}] = \boldsymbol{x}^j_k[i]$ and $\mathrm{Var}(X^j_{k,i}) = \mathbb{E}[X^j_{k,i}](1 - \mathbb{E}[X^j_{k,i}]) \le \frac{1}{4}$, where $\mathbb{E}[\cdot]$ and $\mathrm{Var}(\cdot)$ respectively denote the expected value and the variance of the random variable. It follows from Theorem 5 that $D_{\mathcal{L}_1}(\boldsymbol{x}^j_k, \boldsymbol{\Theta}) \le \upsilon_k$ for all $k \ge k_{\epsilon,\rm lim}$. Therefore $\sum_{i=1}^{n_{\rm rec}} |\mathbb{E}[X^j_{k,i}] - \boldsymbol{\Theta}[i]| \le \upsilon_k$ for all $k \ge k_{\epsilon,\rm lim}$.

The swarm distribution in bin $B[i]$ at the $k^{\rm th}$ time instant is given by $\boldsymbol{\mu}^\star_k[i] = \frac{1}{m_\kappa}\sum_{j=1}^{m_\kappa} X^j_{k,i}$. Therefore, $\mathbb{E}[\boldsymbol{\mu}^\star_k[i]] = \frac{1}{m_\kappa}\sum_{j=1}^{m_\kappa} \mathbb{E}[X^j_{k,i}]$. Moreover,

$$\sum_{i=1}^{n_{\rm rec}} |\mathbb{E}[\boldsymbol{\mu}^\star_k[i]] - \boldsymbol{\Theta}[i]| \le \sum_{i=1}^{n_{\rm rec}} \frac{1}{m_\kappa} \sum_{j=1}^{m_\kappa} \left|\mathbb{E}[X^j_{k,i}] - \boldsymbol{\Theta}[i]\right| \le \upsilon_k.$$

The random variables $X^j_{k,i}$, $j \in \{1,\ldots,m_\kappa\}$ are negatively correlated because

$$\begin{aligned} \mathrm{Cov}(X^{j1}_{k,i}, X^{j2}_{k,i}) &= \mathbb{E}[X^{j1}_{k,i} X^{j2}_{k,i}] - \mathbb{E}[X^{j1}_{k,i}]\mathbb{E}[X^{j2}_{k,i}] \\ &= \binom{m_k-2}{n_{k,i}-2} \Big/ \binom{m_k}{n_{k,i}} - \binom{m_k-1}{n_{k,i}-1}^2 \Big/ \binom{m_k}{n_{k,i}}^2 \le 0\,, \end{aligned}$$

where $n_{k,i}$ is the number of agents in bin $B[i]$ at the k^{th} time instant and $\binom{\cdot}{\cdot}$ represents the Binomial coefficient. Therefore, we obtain

$$\begin{aligned}\operatorname{Var}(\boldsymbol{\mu}_k^\star[i]) &= \frac{1}{m_\kappa^2}\left(\sum_{j=1}^{m_\kappa}\operatorname{Var}(X_{k,i}^j) + 2\sum_{1\le j1<j2\le m_\kappa}\operatorname{Cov}(X_{k,i}^{j1}, X_{k,i}^{j2})\right)\\ &\le \frac{1}{m_\kappa^2}\sum_{j=1}^{m_\kappa}\operatorname{Var}(X_{k,i}^j) \le \frac{1}{4m_\kappa}\,.\end{aligned}$$

It follows from the vector version of the Chebyshev's inequality (cf. [64, Theorem 1.6.4, pp. 25]) that for any ε_{bin}, the $\mathcal{L}_2$ distance is probabilistically bounded by

$$\mathbb{P}\left(\sqrt{\sum_{i=1}^{n_{\text{rec}}}|\boldsymbol{\mu}_k^\star[i]-\mathbb{E}[\boldsymbol{\mu}_k^\star[i]]|^2} \ge \frac{\varepsilon_{\text{bin}}}{\sqrt{n_{\text{rec}}}}\right) \le \frac{n_{\text{rec}}^2}{4m_\kappa\varepsilon_{\text{bin}}^2}\,.$$

Since the $\mathcal{L}_2$ distance $\ge (\frac{1}{\sqrt{n_{\text{rec}}}})$ $\mathcal{L}_1$ distance, therefore

$$\mathbb{P}\left(\sum_{i=1}^{n_{\text{rec}}}|\boldsymbol{\mu}_k^\star[i]-\mathbb{E}[\boldsymbol{\mu}_k^\star[i]]| \ge \varepsilon_{\text{bin}}\right) \le \frac{n_{\text{rec}}^2}{4m_\kappa\varepsilon_{\text{bin}}^2}\,.$$

It follows from the triangle inequality that

$$\mathbb{P}\left(D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star,\boldsymbol{\Theta}) \ge \varepsilon_{\text{bin}}+\upsilon_k\right) \le \frac{n_{\text{rec}}^2}{4m_\kappa\varepsilon_{\text{bin}}^2}\,. \tag{34}$$

The bound on HD follows from (5).

It follows from Theorem 4 that $\lim_{k\to\infty}\boldsymbol{x}_k^j = \boldsymbol{\Theta}$, therefore $D_{\mathcal{L}_1}(\lim_{k\to\infty}\boldsymbol{x}_k^j,\boldsymbol{\Theta}) = 0$ and $\lim_{k\to\infty}\upsilon_k = 0$. By setting $\varepsilon_{\text{bin}} = 2\xi_{\text{des}}^2$, we obtain

$$\mathbb{P}\left(D_H\left(\lim_{k\to\infty}\boldsymbol{\mu}_k^\star,\boldsymbol{\Theta}\right) \ge \xi_{\text{des}}\right) \le \frac{n_{\text{rec}}^2}{16m_\kappa\xi_{\text{des}}^4}\,.$$

The lower bound on the number of agents is given by $\frac{n_{\text{rec}}^2}{16m_\kappa\xi_{\text{des}}^4} \le \varepsilon_{\text{conv}}$. Similarly, setting $\varepsilon_{\text{bin}} = \xi_{\text{des}}$ in (34) and $\lim_{k\to\infty}\upsilon_k = 0$ gives the bound on $\mathcal{L}_1$ distance in (32–33). ■

Remark 7. It follows from Theorem 6 and the weak law of large numbers [65, pp. 86] that the final swarm distribution $\lim_{k\to\infty}\boldsymbol{\mu}_k^\star$ converges in probability to the desired formation $\boldsymbol{\Theta}$ as the number of agents m_κ tends to infinity. □

Thus, we have proved the convergence of the PSG–IMC algorithm for shape formation. We now discuss its property of robustness and some extensions.

Remark 8 (*Robustness of PSG–IMC*). The PSG–IMC algorithm satisfies the Markov (memoryless) property because the action of each agent depends only on its present bin location and the current swarm distribution. This property ensures that all the agents re-start their guidance trajectory from their present bin location during every time instant. Thus, the swarm continues to converge to the desired shape even if agents are added to or removed from the swarm, or if some agents have not reached their target bin during the previous time instant.

Moreover, PSG–IMC can tolerate estimation errors ϵ_{est} (3) in the feedback of the current swarm distribution $\boldsymbol{\mu}_k^j$. The distance between the feedback error terms $\xi_k^j = D_H(\boldsymbol{\Theta},\boldsymbol{\mu}_k^j)$ and $\xi_k^\star = D_H(\boldsymbol{\Theta},\boldsymbol{\mu}_k^\star)$ is bounded by [56]

$$\left|\xi_k^j - \xi_k^\star\right| \le D_H(\boldsymbol{\mu}_k^\star,\boldsymbol{\mu}_k^j) \le \frac{1}{\sqrt{2}}\epsilon_{\text{est}}^{\frac{1}{2}}\,. \tag{35}$$

Even though ξ_k^j might differ from $\xi_k^\star$ substantially, the resulting Markov matrix $\boldsymbol{M}_k^j$ still has $\boldsymbol{\Theta}$ as its stationary distribution. Therefore the agent's PMF vector $\boldsymbol{x}_k^j$ still converges to $\boldsymbol{\Theta}$, and consequently the swarm distribution also converges to $\boldsymbol{\Theta}$. □

Remark 9 (*Collision avoidance in PSG–IMC*). The PSG–IMC algorithm can implement inter-agent collision avoidance using line 15 in Method 2. Collision avoidance with stationary obstacles can also be easily handled by the current method. If the stationary obstacles are comparable or larger than the bin size, then the bins are designed such that they do not overlap with these obstacles. Furthermore, the motion constraint matrices are designed to prevent transitions that are not allowed due to these obstacles. If the stationary obstacles are significantly smaller than the bins, then they can be handled by a lower-level collision avoidance algorithm (see Remark 15 in Appendix). □

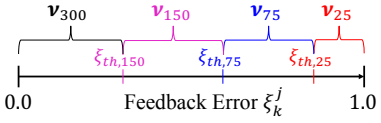


Figure 8. This image shows the different thresholds and the corresponding resolution of the desired formation that should be used when ξ_k^j is within those thresholds.

Remark 10 (*Multiresolution PSG–IMC for shape formation*). We can take advantage of multiresolution representation of the desired formation in our guidance strategy (see Fig. 6). The key idea is that the agents use an appropriate resolution of the desired formation depending on the feedback error. For example, as shown in Fig. 8, we select thresholds $\xi_{\text{th},150}$, $\xi_{\text{th},75}$, and $\xi_{\text{th},25}$ so that the agents use the appropriate resolution of the desired formation ($\boldsymbol{\nu}_{300}$, $\boldsymbol{\nu}_{150}$, $\boldsymbol{\nu}_{75}$ or $\boldsymbol{\nu}_{25}$ in Fig. 6) if the feedback error ξ_k^j is within these thresholds. The main advantage of this approach is its computational efficiency. □

Remark 11 (*Time-varying physical space of the swarm*). The compact physical space over which the swarm is distributed need not be time-invariant in the global reference frame. The local reference frame of the swarm can follow a predefined trajectory in the global reference frame (e.g., an orbit in space or a trajectory in the sea) and the time-varying position of each bin can be computed from this known trajectory. Consequently, all the algorithms discussed in this paper are also applicable in this scenario. □

V. PSG–IMC for Area Exploration

In this section, we present an extension of the PSG–IMC algorithm for area exploration in which a swarm of distributed agents are driven to match the unknown target distribution of some physical or artificial phenomena (e.g., oil spill). This problem is commonly called *goal searching* [29].

Definition 11 (*Unknown target distribution* $\mathbf{\Omega}$). The unknown target distribution $\mathbf{\Omega}$ is a probability (row) vector in $\mathbb{R}^{n_{\text{bin}}}$, where each element $\mathbf{\Omega}[i]$ represents the target distribution in the corresponding bin $B[i]$. Each agent can measure the target distribution in its present bin (i.e., agent in bin $B[i]$ can measure $\mathbf{\Omega}[i]$). □

Each agent independently determines its bin-to-bin trajectory using the PSG–IMC algorithm for area exploration so that the overall swarm converges to this unknown target distribution $\mathbf{\Omega}$. The key idea of this algorithm is that the waiting time in a bin is directly proportional to the target distribution in that bin.

Method 3: PSG–IMC for Area Exploration

1: Lines 1–2 in Method 2 and given τ_c, ξ^j
2: Measure target distribution in present bin $\mathbf{\Omega}[i]$
3: **if** $k - k_0 < \tau_c \mathbf{\Omega}[i]$, **then**
4: Wait in bin $B[i]$
5: **else** Set $\mathbf{\Theta} = \frac{\mathbf{1}^T}{n_{\text{bin}}}$ and $\xi_k^j = \xi^j$
6: Compute the term $\eta_{k,i}^j$ using (36)
7: Lines 9, 11–13, 15 in Method 2, Set $k_0 = k$
8: **end if**

The pseudo-code of this PSG–IMC algorithm for area exploration is given in Method 3. The j^{th} agent first measures the target distribution in its present bin $\mathbf{\Omega}[i]$ (line 2). The waiting time in bin $B[i]$ is greater than or equal to $\tau_c \mathbf{\Omega}[i]$, where τ_c is the constant of proportionality.

Assumption 7 (*Waiting time constant* τ_c). Let Δ denote the time step of the algorithm. Each agent has a-priori information about the common waiting time constant τ_c, which is selected such that $\tau_c > \frac{\Delta}{\min^+ \mathbf{\Omega}}$. The actual value of τ_c plays a crucial role in the convergence analysis. □

The agent checks if it has spent enough time instants in bin $B[i]$ (line 3), where k_0 is set in line 7. When the algorithm starts ($k = 1$), we set $k_0 = 1$. If the agent has not spent enough time instants in bin $B[i]$, then it continues to wait in bin $B[i]$ (line 4).

If the agent has spent enough time instants in bin $B[i]$, then it sets $\mathbf{\Theta} = \frac{\mathbf{1}^T}{n_{\text{bin}}}$ because it wants to uniformly explore all the bins (line 5). The agent sets the feedback error ξ_k^j to some positive known constant $\xi^j \in (0, 1)$ because it does not know the target distribution $\mathbf{\Omega}$ (line 5). In order to suppress undesirable transitions from deficient bins, the agent computes the feedback-based term $\eta_{k,i}^j$ as follows (line 6):

$$\eta_{k,i}^j = \exp(-\tau^j k) \frac{\exp\left(\beta^j (\mathbf{\Omega}[i] - \boldsymbol{\mu}_k^j[i])\right)}{\exp\left(\beta^j |\mathbf{\Omega}[i] - \boldsymbol{\mu}_k^j[i]|\right)}. \tag{36}$$

Then, the agent computes the transition probabilities $\boldsymbol{P}_k^j[i, 1{:}\, n_{\text{bin}}]$ using lines 9 and 11 from Method 2, selects the next bin using lines 12–13 from Method 2, and goes to the selected bin using line 15 from Method 2 (line 7). Finally, the agent sets k_0 equal to the current time instant k (line 7). We now discuss the convergence analysis of this algorithm.

Theorem 7. Let $\hat{\mathbf{\Omega}} \in \mathbb{R}^{n_{\text{bin}}}$ represent the following probability (row) vector

$$\hat{\mathbf{\Omega}}[i] = \begin{cases} \Xi\Delta \left\lceil \frac{\tau_c \mathbf{\Omega}[i]}{\Delta} \right\rceil & \text{if } i \in \mathcal{V} \\ \Xi\Delta & \text{if } i \in \bar{\mathcal{V}} \end{cases},$$

where $\mathcal{V}$ represents the set of all bins $B[i]$ with $\mathbf{\Omega}[i] > 0$, $\bar{\mathcal{V}} = \{1, \ldots, n_{\text{bin}}\} \setminus \mathcal{V}$, $\lceil \cdot \rceil$ is the ceiling function, and the normalizing constant $\Xi = \frac{1}{\sum_{\ell \in \bar{\mathcal{V}}} \Delta + \sum_{\ell \in \mathcal{V}} \Delta \lceil \frac{\tau_c \mathbf{\Omega}[\ell]}{\Delta} \rceil}$. According to Method 3, the PMF vector $\boldsymbol{x}_k^j$ converges pointwise to the distribution $\hat{\mathbf{\Omega}}$ irrespective of the initial condition, where

$$D_{\mathcal{L}_1}(\hat{\mathbf{\Omega}}, \mathbf{\Omega}) \leq \frac{n_\Omega \frac{\Delta}{\tau_c}}{(n_{\text{bin}} - n_\Omega)\frac{\Delta}{\tau_c} + 1}.$$

and n_Ω denotes the number of bins with nonzero elements in $\mathbf{\Omega}$. If $\tau_c \gg \Delta$, then $\lim_{k \to \infty} \boldsymbol{x}_k^j = \mathbf{\Omega}$ pointwise for all agents.

Proof: Here, all bins are recurrent bins because $\mathbf{\Theta} = \frac{\mathbf{1}^T}{n_{\text{bin}}}$. It follows from Theorem 4 that as $k \to \infty$, an agent is equally likely to transition to any bin $B[i]$. But the waiting time in each bin, under Assumption 7, is given by

$$\begin{matrix} \text{Waiting time} \\ \text{in bin } B[i] \end{matrix} = \begin{cases} \Delta \left\lceil \frac{\tau_c \mathbf{\Omega}[i]}{\Delta} \right\rceil & \text{if } i \in \mathcal{V} \\ \Delta & \text{if } i \in \bar{\mathcal{V}} \end{cases}.$$

Therefore, $\lim_{k \to \infty} \boldsymbol{x}_k^j = \hat{\mathbf{\Omega}}$ pointwise for all agents. Note that the set $\mathcal{V}$ has a cardinality of n_Ω. The $\mathcal{L}_1$ distance between $\hat{\mathbf{\Omega}}$ and $\mathbf{\Omega}$ is given by

$$\begin{aligned} D_{\mathcal{L}_1}(\hat{\mathbf{\Omega}}, \mathbf{\Omega}) &= \sum_{i \in \bar{\mathcal{V}}} \Xi\Delta + \sum_{i \in \mathcal{V}} \left| \Xi\Delta \left\lceil \frac{\tau_c \mathbf{\Omega}[i]}{\Delta} \right\rceil - \mathbf{\Omega}[i] \right|, \\ &\leq \frac{\sum_{i \in \bar{\mathcal{V}}} \Delta + \sum_{i \in \mathcal{V}} (\Delta - \mathbf{\Omega}[i](n_{\text{bin}} - n_\Omega)\Delta)}{(n_{\text{bin}} - n_\Omega)\Delta + \tau_c}, \\ &\leq \frac{n_\Omega \Delta}{(n_{\text{bin}} - n_\Omega)\Delta + \tau_c}. \end{aligned}$$

If $\tau_c \gg \Delta$, then $D_{\mathcal{L}_1}(\hat{\mathbf{\Omega}}, \mathbf{\Omega}) = 0$ and $\lim_{k \to \infty} \boldsymbol{x}_k^j = \mathbf{\Omega}$ pointwise for all agents. ■

The remaining convergence analysis straightforwardly follows that of the previous algorithm given in Section IV-A.

VI. Numerical Simulation and Experimentation

Results of simulation and experimentation for shape formation are discussed in Sections VI-A–VI-C while results of numerical simulation for area exploration are discussed in Section VI-D.

A. Numerical Simulation for Shape Formation

In this subsection, we show that PSG–IMC for shape formation can be used to achieve multiple complex formation shapes with fine spatial resolutions. At the start of each simulation, a swarm of 10^5 agents are uniformly distributed in the physical space. During each time instant, each agent incorporates error-free feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$. The cost of transition is equal to the ℓ_1 distance between bins, therefore it is symmetric. We use the following constants $\varepsilon_M = 1$, $\varepsilon_C = 0.1$, $\tau^j = 10^{-3}$, and $\beta^j = 1.8 \times 10^5$.

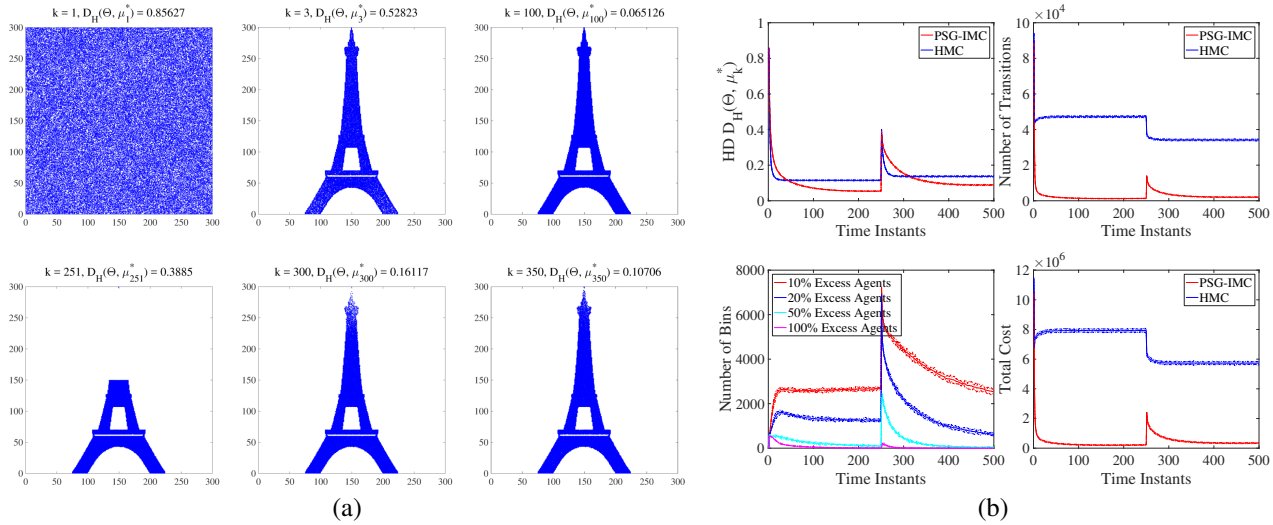


Figure 9. **(a)** This plot shows the swarm distribution at different time instants, in a sample run of the Monte Carlo simulation. Starting from a uniform distribution, the swarm converges to the desired formation of the Eiffel Tower. At the 251^{st} time instant, the agents in the top half of the formation are removed and the remaining agents reconfigure into the desired formation. See the supplementary video (SV2). **(b)** The cumulative results of 10 Monte Carlo simulations are shown. The sudden jump at the 251^{st} time instant is because of the removal of agents from the top half of the formation.

In the first example, the desired formation $\boldsymbol{\Theta}$ is based on the Eiffel Tower ($\boldsymbol{\nu}_{300}$ in Fig. 6, with 300×300 bins). Each agent is allowed to transition to only those bins that are at most 50 steps away. Starting from a uniform distribution, the agents attain the desired formation in 100 time instants (see Fig. 9(a)). At the 251^{st} time instant, approximately 3×10^4 agents are removed from the top half of the formation and the remaining agents reconfigure into the desired formation.

Here, the HMC algorithm uses the homogeneous Markov matrix constructed by using (10)-(11) and setting $\xi_k^j = 1$. The cumulative results of 10 Monte Carlo simulations of the PSG–IMC and HMC algorithms are shown in Fig. 9(b). Compared to the HMC-based algorithm, PSG–IMC provides approximately 2 times improvement in HD, 16 times reduction in the cumulative number of transitions in 500 time instants, and 16 times reduction in the total cost incurred by all the agents in 500 time instants. The key reasons behind the superior performance of PSG–IMC are as follows:
(i) In Fig. 9(b), the HD of the HMC algorithm reaches an equilibrium at 0.115 after approximately 40 time instants. The HMC algorithm allows undesirable transitions (i.e., transitions from bins with fewer agents to bins with surplus agents) which increases the HD. Since these undesirable transitions reach an equilibrium with the other favorable transitions, the HD for the HMC algorithm also reaches an equilibrium. Such undesirable transitions are largely avoided in PSG–IMC (due to lines 10–11 in Method 2), hence the resulting HD after 250 time instants is 0.055 (i.e., ≈ 2 times improvement compared to HMC). The final HD can be further reduced by tuning τ^j and β^j. But undesirable transitions prevent both these Markovian approaches from achieving desired convergence.
(ii) In the HMC algorithm, there are 1.9×10^6 transitions in the first 40 time instants. This is significantly more than that of PSG–IMC (i.e., 5.6×10^5 transitions in 250 time instant). In PSG–IMC, the number of transitions at each time instant 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. Note that the total number of transitions in the HMC algorithm in 250 time instant is extremely large (i.e., 1.2×10^7 transitions).
(iii) There are 7–9 agents in each recurrent bin. For PSG–IMC, the number of bins with 1–2 excess agents (i.e., 10–20%) is shown in Fig. 9(b). The number of bins with a large number of excess agents (i.e., 50–100%) is a small fraction of the total number of bins. Hence this algorithm also avoids traffic jams or bottlenecks.
Consequently, the PSG–IMC algorithm achieves a smaller convergence error than the HMC-based algorithm and significantly reduces the number of transitions for achieving and maintaining the desired formation. Moreover, these three key reasons depend on the feedback and, therefore, do not hold true for HMC-based algorithms.

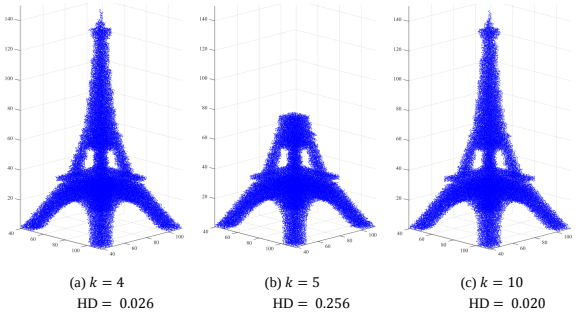


Figure 10. The swarm attains the 3-D shape of the Eiffel Tower. When agents are removed from the top half of the formation, the remaining agents reconfigure to the desired formation.

In the next example, the desired formation $\boldsymbol{\Theta}$ is based on the 3-D Eiffel Tower (see Fig. 10, with $150 \times 150 \times 150$ bins). Starting from a uniform distribution and no motion constraints, a swarm of 10^5 agents achieve the desired formation in a few

time instants. When 1.25×10^4 agents are removed from the top half of the formation, the remaining agents reconfigure to the desired formation in a few more time instants.

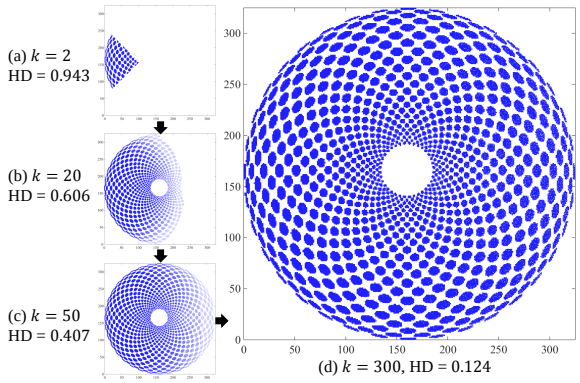


Figure 11. This plot shows the swarm distribution at different time instants, where the swarm attains the desired formation shape with multiple disconnected parts. See the supplementary video (SV3).

In the next example, the desired formation $\boldsymbol{\Theta}$ in Fig. 11(d), with 325×325 bins, has multiple disconnected parts. Each agent is allowed to transition to only those bins that are at most 50 steps away. In this case, the recurrent bins are not contiguous, but they satisfy property (3) in Definition 4. A swarm of 10^6 agents starts from the left-most bin (located at $(1, 163)$) and attains the desired formation in 300 time instants (as shown in Fig. 11(a)–(d)).

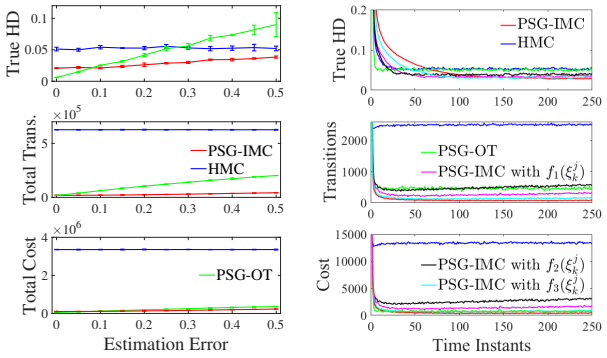
Table II
COMPUTATION TIMES FOR THE PSG–IMC AND HMC-BASED ALGORITHMS

Desired Distribution	Number of Agents	PSG–IMC	HMC
$\boldsymbol{\nu}_{25}$ (625 bins)	5000	15 seconds	23 seconds
	10^4	32 seconds	43 seconds
	10^5	5 minutes	6 minutes
	10^6	45 minutes	54 minutes
$\boldsymbol{\nu}_{75}$ (5625 bins)	10^4	3 minutes	3.5 minutes
	10^5	30 minutes	36 minutes
	10^6	5 hours	5.9 hours
$\boldsymbol{\nu}_{150}$ (2.25×10^4 bins)	10^5	2.0 hours	2.4 hours
	10^6	23 hours	1 day
$\boldsymbol{\nu}_{300}$ (9×10^4 bins)	10^5	5.3 hours	8 hours
	10^6	2.5 days	3.3 days

In Table II, the computation times using the PSG–IMC and HMC-based algorithms on a desktop computer are shown. The simulation setup for all these runs is exactly the same as shown in Fig. 9. Although both the algorithms scale well with the spatial resolution of the desired distribution and the number of agents in the swarm, PSG–IMC performs better because of the smaller number of transitions. Thus, the robustness and scalability properties of PSG–IMC for shape formation are evident in these simulations.

B. Numerical Simulation for Shape Formation with Coarse Spatial Resolution and Estimation Errors

The objective of this subsection is to study the effect of estimation errors on the three Eulerian algorithms, namely PSG–IMC, HMC, and the Probabilistic Swarm Guidance using Optimal Transport (PSG–OT) algorithm [37] (see Remark 12 in Appendix). The desired formation $\boldsymbol{\Theta}$ is given by the coarse Eiffel Tower ($\boldsymbol{\nu}_{25}$ in Fig. 6, with 25×25 bins) because the computation time for PSG–OT's LP (38) becomes impractical for finer resolutions. The simulation setup is similar to that in Section VI-A. A swarm of 5000 agents is used and each agent is allowed to transition to only those bins which are at most 9 steps away.



(a) (b)

Figure 12. **(a)** The estimation error is varied from 0.0 to 0.5. The performance of the three algorithms, along with 1σ error-bars, are shown for the true HD $D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}^\star_{250})$ between the actual swarm distribution after 250 time instants and the desired formation, the cumulative number of transitions in 250 time instants, and the total cost incurred by all the agents in 250 time instants. **(b)** The cumulative results of the three algorithms and alternative functions for ξ_k^j are shown, with an estimation error $\epsilon_{\text{est}} = 0.25$.

During each time instant, each agent incorporates feedback of the current swarm distribution $\boldsymbol{\mu}_k^j$ with an estimation error ϵ_{est}. The cumulative results of Monte Carlo simulations are shown in Fig. 12(a). The PSG–OT algorithm performs slightly better than PSG–IMC in the absence of an estimation error ($\epsilon_{\text{est}} = 0.0$), but such a situation does not arise in practical scenarios. Since the estimated swarm distribution directly appears as a constraint in the optimization problem of the PSG–OT algorithm, its convergence error increases precipitously with the estimation error and it performs worse than the open-loop HMC-based algorithm if $\epsilon_{\text{est}} \geq 0.25$. On the other hand, PSG–IMC works reliably well for all estimation errors and much outperforms the other two algorithms. Thus, PSG–IMC can tolerate large estimation errors in the current swarm distribution.

The cumulative results for the three algorithms are shown in Fig. 12(b), where the estimation error ϵ_{est} is equal to 0.25. Compared to the HMC and PSG–OT algorithms, PSG–IMC achieves a smaller convergence error with fewer transitions. The results of a few alternative functions for ξ_k^j are also shown in Fig. 12(b) (see Remark 4). The two functions $f_1(\xi_k^j) = \tanh(\pi \xi_k^j)$ and $f_2(\xi_k^j) = \sin\left(\cos^{-1}(1 - \xi_k^j)\right)$ are always larger than ξ_k^j. The sigmoid function $f_3(\xi_k^j) = \left(\xi_k^j + 0.1 \sin(2\pi \xi_k^j)\right)$ is larger than ξ_k^j when $\xi_k^j < 0.5$. Fig. 12(b) shows that the rate of convergence increases with these functions, but there is also a corresponding increase in the number of transitions. The collision-free motion of the agents, where the agents use the Voronoi-based algorithm in

Remark 15, is shown in the supplementary video (SV4).

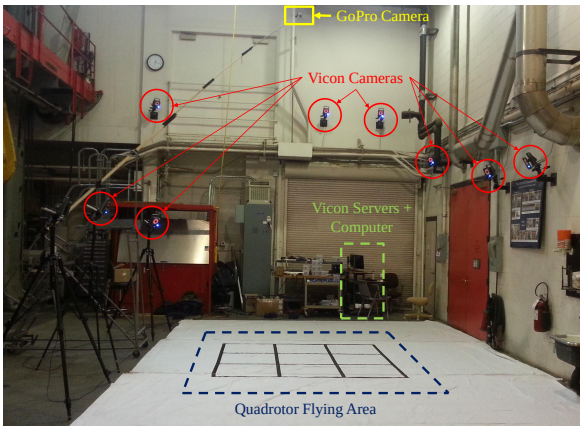
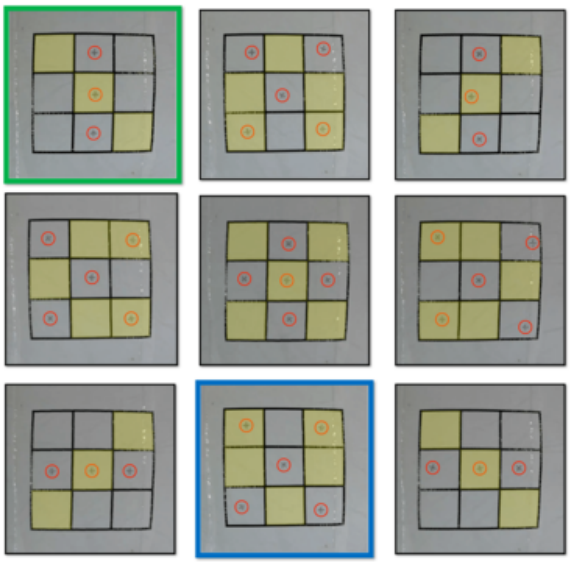


Figure 13. The experimental setup is shown. Notice the 3×3 grid in the quadrotor flying area.

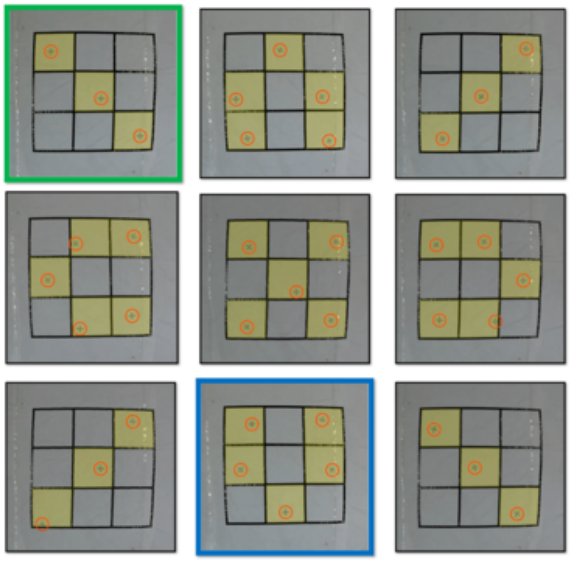
C. Experimental Results for Shape Formation in Real Time

In this subsection, we show that PSG–IMC along with lower-level guidance in Remark 15 can be executed in real time to control quadrotors. The experimental setup is shown in Fig. 13 and described in [7], [66]. A 3×3 grid is placed on the ground where the quadrotor experiments are performed. The quadrotors are tracked using a motion capture system. A desktop computer executes the PSG–IMC algorithm for each agent in a virtually distributed manner; i.e., each quadrotor's computations are performed by an independent thread on the computer. The trajectories computed by each quadrotor's thread are then communicated to that quadrotor. Finally, each quadrotor follows its desired trajectory using the nonlinear tracking control law [7], [66].

We first present nine different experiments using three or five quadrotors. The desired formation shape for these experiments are shown in Fig. 14. In these experiments, the time step of PSG–IMC is 9 seconds and the time step for the lower-level guidance algorithm in Remark 15 is 3 seconds. As shown in the supplementary video (SV5), the quadrotors first take off from the ground and climb to 1 m altitude. Thereafter, PSG–IMC is switched on, and the quadrotors achieve the desired formation shape within a few time instants. The quadrotors then land inside their selected bins. Note that there exists some parallax error in the video (SV5) because the grid is marked on the ground, the quadrotors are flying at 1 m altitude, and the camera is located directly above the central square at 5 m height. This parallax error vanishes when the quadrotors land and the desired formation shape is clearly visible in the end of video (SV5). The quadrotors experience measurement errors, actuator errors, and inter-quadrotor aerodynamic coupling due to downwash. In addition, the quadrotors experience environmental disturbance and intermittent communication loss. These experiments show that PSG–IMC can be implemented in real time to achieve a variety of desired formation shapes while dealing with various real-world disturbance sources.



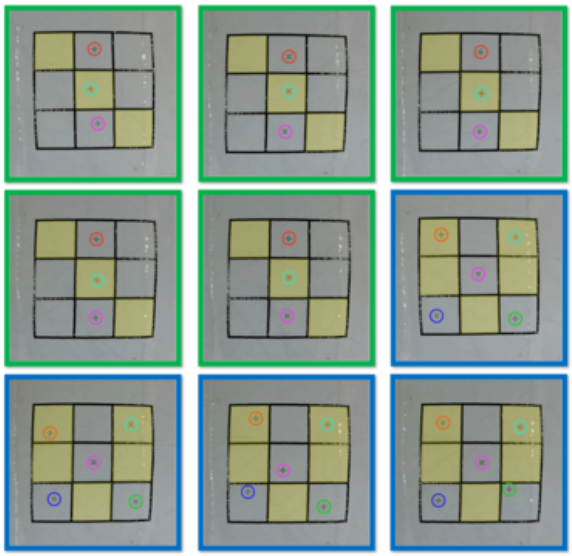
(a) Initial position



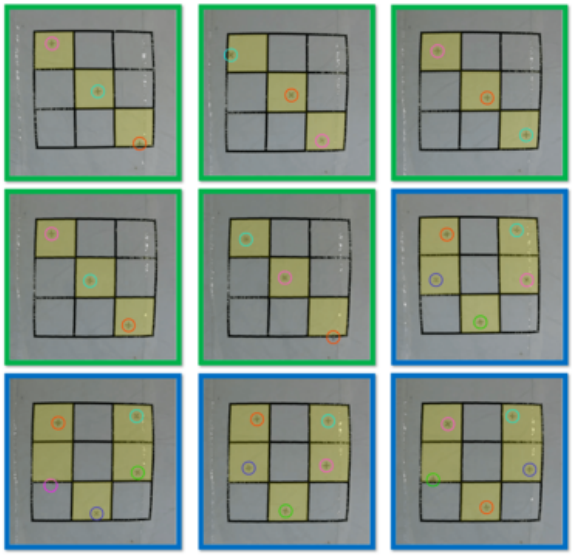
(b) Final position

Figure 14. Nine different experiments are shows where three or five quadrotors execute PSG–IMC in real time to achieve the desired formations shown in yellow. The quadrotors are encircled in red. See the supplementary video (SV5).

A key feature of PSG–IMC is that each agent probabilistically selects the bin that it transitions to. We demonstrate this property using two sets of experiments in Fig. 15. In the first experiment, five quadrotors start from the same initial condition and reach the same desired formation highlighted in blue in Fig. 14. But in each of the four experimental runs, the actual trajectory of each quadrotor is significantly different as shown in Fig. 15. Similarly, in the second experiment, five experimental runs are shown where three quadrotors reach the desired formation highlighted in green in Fig. 14. These repeated experiments show that the quadrotors select different bins during different runs due to the probabilistic nature of PSG–IMC.



(a) Initial position



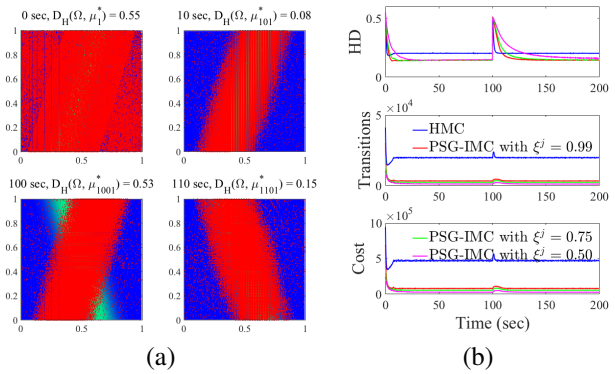
(b) Final position

Figure 15. Repetitions each of the same two experiments are shown. The desired formations are shown in yellow and the quadrotors are encircled in different colors. The actual trajectories of the quadrotors in the these repetitions are different. See the supplementary video (SV6).

D. Numerical Simulation for Area Exploration

In this numerical example, a swarm of 10^5 agents use the PSG–IMC algorithm for area exploration to attain the unknown target distribution. The physical space $[0,1] \times [0,1]$ is partitioned into 100×100 bins and the time step $\Delta = 0.1$ sec. The unknown target distribution $\boldsymbol{\Omega}_1$ for the first 100 sec is given by the PMF representation of the multivariate normal distribution $\boldsymbol{\mathcal{N}}\left(\left[\begin{smallmatrix}0.5 & 0.5\end{smallmatrix}\right], \left[\begin{smallmatrix}0.1 & 0.3\\ 0.3 & 1.0\end{smallmatrix}\right]\right)$, as shown in the background contour plots in Fig. 16(a). Similarly, the unknown target distribution $\boldsymbol{\Omega}_2$ for the next 100 sec is given by $\boldsymbol{\mathcal{N}}\left(\left[\begin{smallmatrix}0.5 & 0.5\end{smallmatrix}\right], \left[\begin{smallmatrix}0.1 & -0.3\\ -0.3 & 1.0\end{smallmatrix}\right]\right)$. Here we use the constants $\tau^j = 2.5 \times 10^{-3}$, and $\beta^j = 200$ in (36).

The cumulative results of 10 Monte Carlo simulations for different values of ξ^j are shown in Fig. 16(b). A results for the HMC-based area exploration algorithm are also shown in



(a) (b)

Figure 16. **(a)** 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. Starting from a uniform distribution, the swarm converges to the unknown target distribution. After 100 sec, the unknown target distribution is suddenly changed and the agents reconfigure to this new target distribution. See the supplementary video (SV6). **(b)** The cumulative results of 10 Monte Carlo simulations are shown. The jump after 100 sec is because of the sudden change in the unknown target distribution.

Fig. 16(b). Compared to the HMC-based algorithm, PSG–IMC provides approximately 1.5 times improvement in HD, 6 times reduction in the cumulative number of transitions and the total cost incurred by the agents in 200 sec.

VII. Conclusion

In this paper, we have presented the new distributed control algorithm for large-scale swarms to achieve the desired formation shape or unknown target distribution from any initial condition by systematically combining swarm-density control with a probabilistic approach. The resulting PSG–IMC algorithm constructs time-inhomogeneous Markov chains in real time using the Hellinger-distance-based feedback error between the current swarm distribution and the desired distribution. The Markov matrices satisfy suitable motion constraints, minimize the expected cost of transitions at each time instant, and circumvent transitions from bins that are deficient in the number of agents. Then, PSG–IMC essentially solves both the stochastic target assignment problem and generation of optimal bin-to-bin transition trajectories, which can be hierarchically combined with a lower-level guidance and control inside each partition. We have also presented the rigorous convergence analysis of PSG–IMC as well as the probabilistic bounds relating the size of the swarm to the convergence error.

Results of numerical simulation show that PSG–IMC achieves 6-16 times reduction in total cost of transitions and 1.5-2 times reduction in HD, as compared to the existing HMC-based algorithms for shape formation and area exploration applications. This is because PSG–IMC avoids undesirable transitions, and the number of transitions at each time instant is proportional to the HD. In the presence of estimation errors, PSG–IMC also outperforms the optimal transport-based algorithm, because the PMF of the predicted position of each agent converges to the desired formation regardless of estimation errors. We have demonstrated the robustness and computational benefits of PSG–IMC using hardware experiments with multiple quadrotors, where a Voronoi-based lower-level guidance and control algorithm has been used. This also

provides an avenue for future research in tightly integrating a lower-level guidance and control algorithm with PSG–IMC.

PSG–IMC can also solve other cooperative control tasks, such as surveillance, task allocation, and coverage, since such problems can also be cast as shape formation or area exploration problems. We envisage that the proposed algorithm will facilitate the development of autonomous swarm robotic systems that are capable of performing a variety of complex tasks, by providing a versatile, robust, and scalable path planning strategy.

ACKNOWLEDGMENT

The authors would like to thank D. Morgan, G. Subramanian, J. Yu, I. Jang, and S. Rahili for their valuable inputs. 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.

APPENDIX

We first state some definitions and results used in the proofs of Theorems 4 and 5, and then present the PSG–OT algorithm.

Definition 12 (*Primitive matrix* [47, pp. 3]). A square non-negative matrix $\boldsymbol{T}$ is said to be primitive if there exists a positive integer k such that $\boldsymbol{T}^k > 0$. □

Definition 13 (*Asymptotic homogeneity* [47, pp. 92, 149]). A sequence of stochastic matrices $\boldsymbol{P}_k$, $k \geq 1$ is said to be asymptotically homogeneous (with respect to $\boldsymbol{d}$) if there exists a probability (row) vector $\boldsymbol{d}$ such that $\lim_{k\to\infty} \boldsymbol{d}\boldsymbol{P}_k = \boldsymbol{d}$. □

Definition 14 (*Strong ergodicity* [47, pp. 92, 149]). The forward matrix product $\boldsymbol{U}_{T,r} := \boldsymbol{P}_T\boldsymbol{P}_{T+1}\cdots\boldsymbol{P}_{r-1}$, formed from a sequence of stochastic matrices $\boldsymbol{P}_k$, $k \geq 1$, is said to be strongly ergodic if for each i, ℓ, T, we get $\lim_{r\to\infty} \boldsymbol{U}_{T,r}[i,\ell] = \boldsymbol{v}[\ell]$, where $\boldsymbol{v}$ is a probability vector and the element $\boldsymbol{v}[\ell]$ is independent of i. Therefore, $\boldsymbol{v}$ is the unique limit vector and $\lim_{r\to\infty} \boldsymbol{U}_{T,r} = \boldsymbol{1}\boldsymbol{v}$. □

Theorem 8 ([47, pp. 150]). If (i) the forward matrix product $\boldsymbol{U}_{T,r}$ is primitive and (ii) there exists γ (independent of k) such that

$$0 < \gamma \leq \min_{i,\ell}{}^{+} \boldsymbol{P}_k[i,\ell]\,, \tag{37}$$

where $\min^+$ refers to the minimum of the positive elements, then the asymptotic homogeneity of $\boldsymbol{P}_k$ is necessary and sufficient for strong ergodicity of $\boldsymbol{U}_{T,r}$.

Theorem 9 ([47, pp. 149]). Let $\boldsymbol{e}_k$ be the unique stationary distribution vector of the matrix $\boldsymbol{P}_k$ (i.e., $\boldsymbol{e}_k\boldsymbol{P}_k = \boldsymbol{e}_k$). If (i) all $\boldsymbol{P}_k$, $k \geq 1$ are irreducible and (ii) there exists γ (independent of k) such that (37) is satisfied, then asymptotic homogeneity of $\boldsymbol{P}_k$ (with respect to $\boldsymbol{d}$) is equivalent to $\lim_{k\to\infty} \boldsymbol{e}_k = \boldsymbol{e}$, where $\boldsymbol{e}$ is a limit vector. Moreover, $\boldsymbol{d} = \boldsymbol{e}$.

Corollary 3 ([47, pp. 150]). Under the prior conditions (i) and (ii) of Theorem 9 and if (iii) $\boldsymbol{U}_{T,r}$ is strongly ergodic with unique limit vector $\boldsymbol{v}$, then $\boldsymbol{v} = \boldsymbol{e}$.

Theorem 10. The matrix $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}$ is a positive matrix.

Proof: Proof by Contradiction. Assume that $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$ for some $i \neq \ell$. Then $\boldsymbol{U}^j_{k,r}[i,\ell] = 0$ for all $r \leq k + n_{\text{rec}} - 2$ because $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] \geq \boldsymbol{U}^j_{k,r}[i,\ell]\Big(\prod_{q=r}^{k+n_{\text{rec}}-2} \boldsymbol{P}^j_{q,\text{sub}}[\ell,\ell]\Big)$. Since the matrix $\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}$ is irreducible, there exists $s_1 \in \{1,\ldots,n_{\text{rec}}\}\backslash\{i,\ell\}$ such that $\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}[s_1,\ell] > 0$. Since $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] \geq \boldsymbol{U}^j_{k,k+n_{\text{rec}}-2}[i,s_1]\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}[s_1,\ell]$, consequently $\boldsymbol{U}^j_{k,r}[i,s_1] = 0$ for all $r \leq k + n_{\text{rec}} - 2$. Similarly, there exists $s_2 \in \{1,\ldots,n_{\text{rec}}\}\backslash\{i,\ell,s_1\}$ such that either $\boldsymbol{P}^j_{k+n_{\text{rec}}-3,\text{sub}}[s_2,\ell] > 0$ or $\boldsymbol{P}^j_{k+n_{\text{rec}}-3,\text{sub}}[s_2,s_1] > 0$. Therefore $\boldsymbol{U}^j_{k,r}[i,s_2] = 0$ for all $r \leq k + n_{\text{rec}} - 3$. Continuing this argument till the k^{th} time instant, we see that if $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$, then $\boldsymbol{P}^j_{k,\text{sub}}[i,s] = 0$ for all $s \in \{1,\ldots,n_{\text{rec}}\}\backslash\{i\}$. But this is a contradiction since $\boldsymbol{P}^j_{k,\text{sub}}$ is irreducible. ■

Remark 12 (*PSG Optimal Transport (OT)* [37]). The cost matrix $\boldsymbol{C}_k$ from Definition 5 is first modified to capture motion constraints, i.e., $\tilde{\boldsymbol{C}}_k[i,\ell] = \boldsymbol{C}_k[i,\ell]$ if $\boldsymbol{A}^j_k[i,\ell] = 1$ and $\tilde{\boldsymbol{C}}_k[i,\ell] = C_{\max}$ otherwise, where $C_{\max} \gg \boldsymbol{C}_k[i,\ell]$ for all i,ℓ. The optimal transport map $\boldsymbol{\Gamma}^j_k \in \mathbb{R}^{n_{\text{bin}}\times n_{\text{bin}}}$ is found using the following LP:

$$\min_{\boldsymbol{\Gamma}^j_k[i,\ell]\geq 0,\forall i,\ell} \sum_{i=1}^{n_{\text{bin}}}\sum_{\ell=1}^{n_{\text{bin}}} \tilde{\boldsymbol{C}}_k[i,\ell]\boldsymbol{\Gamma}^j_k[i,\ell]\,, \tag{38}$$

$$\text{sub. to (i)} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{\Gamma}^j_k[i,\ell] = \boldsymbol{\mu}^j_k[i], \forall i, \text{ (ii)} \sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Gamma}^j_k[i,\ell] = \boldsymbol{\Theta}[\ell], \forall \ell.$$

Note that the estimated current swarm distribution $\boldsymbol{\mu}^j_k$ directly appears a marginal constraint on $\boldsymbol{\Gamma}^j_k$ in (38)(i), hence $\boldsymbol{\Gamma}^j_k$ is sensitive to estimation errors. The matrix $\boldsymbol{\Gamma}^j_k$ is not a Markov matrix because it is not row stochastic. □

Remark 13 (*Distributed estimation of* $\boldsymbol{\mu}^\star_k$ [52]). Let the probability vector $\hat{\boldsymbol{\mu}}^j_{k,\nu} \in \mathbb{R}^{n_{\text{bin}}}$ represent the j^{th} agent's estimate of the current swarm distribution during the ν^{th} consensus loop at the k^{th} time step. During each consensus loop, the agents recursively combine their local estimates with their neighboring agents as follows:

$$\hat{\boldsymbol{\mu}}^j_{k,\nu+1} = \sum_{\ell\in\mathcal{J}^j_k} G_k[\ell,j]\,\hat{\boldsymbol{\mu}}^j_{k,\nu}, \qquad \forall \nu \in \mathbb{N}, \tag{39}$$

where $\mathcal{J}^j_k$ is the set of inclusive neighbors of the j^{th} agent, $\hat{\boldsymbol{\mu}}^j_{k,1} = \boldsymbol{r}^j_k$ is the initial local estimate of each agent, and the matrix G_k represents the weights with $\sum_{\ell\in\mathcal{J}^j_k} G_k[\ell,j] = 1$. Under Assumption 5, the matrix G_k is irreducible. Distributed algorithms in [67], [68] are used to ensure that the matrix G_k is balanced.

If the matrix G_k is irreducible and balanced, then each agent's local estimate $\hat{\boldsymbol{\mu}}^j_{k,\nu}$ globally exponentially converges to $\boldsymbol{\mu}^\star_k$ pointwise with a rate faster or equal to the second-largest singular value of G_k (i.e., $\sigma_2(G_k)$). For some $\varepsilon_{\text{cons}} > 0$, if the number of consensus loops within each consensus stage $n_{\text{loop}} \geq \left\lceil \frac{\ln(\varepsilon_{\text{cons}}/2m)}{\ln \sigma_2(G_k)} \right\rceil$; then the convergence error is bounded by $\sum_{j=1}^{m_k} D_{\mathcal{L}_1}(\boldsymbol{\mu}^\star_k, \hat{\boldsymbol{\mu}}^j_{k,n_{\text{loop}}}) \leq \varepsilon_{\text{cons}}$. The j^{th} agent's estimate

of the current swarm distribution at the k^{th} time instant is given by $\boldsymbol{\mu}_k^j = \hat{\boldsymbol{\mu}}_{k,n_{\text{loop}}}^j$. □

Remark 14 (*Inverse transform sampling* [35]). This is a standard sampling technique for generating samples at random from a given PMF over the set of bins. The key steps are: (1) Sample a random number z from the uniform distribution in the interval $[0,\ 1]$, denoted by unif$[0,\ 1]$. (2) Represent the PMF as a cumulative distribution function (CDF). (3) Find bin $B[i]$ such that CDF of bins up to (but not including) bin $B[i]$ is less than z; and the CDF of bin $B[i]$ is greater than or equal to z. Then, the bin $B[i]$ is selected as the sample. □

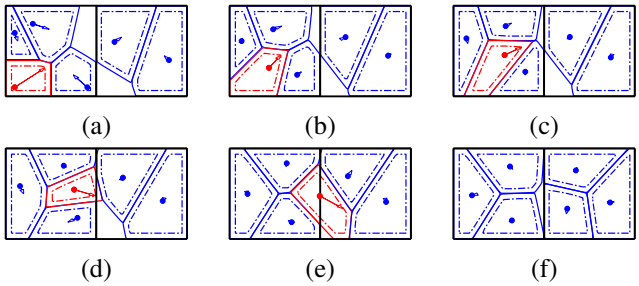


Figure 17. Six agents in two bins (left and right bin) using the Voronoi-based collision-free trajectory generation algorithm are shown. The red agent goes from the left bin to the right bin. The remaining blue agents stay in their present bin. The Voronoi sets of all the agents along with their trajectories (denoted using arrows) are also shown.

Remark 15 (*Voronoi-based collision-free motion to target bin* [37]). The agents generate and update their Voronoi partitions by communicating with their neighboring agents and considering nearby stationary obstacles (see Fig. 17). The agents that need to transition to another bin move to the location in their Voronoi partition that is closest to their target bin, while maintaining a buffer distance for collision avoidance. The agents that remain in their present bin move to stay in the centroid of each Voronoi partition. This results in a collision-free trajectory for each agent. □

REFERENCES

[1] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, "On development of 100-gram-class spacecraft for swarm applications," *IEEE Syst. J.*, vol. 10, no. 2, pp. 673–684, June 2016.

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

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

[4] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1115–1130, Dec. 2006.

[5] P. Yang, R. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, 2008.

[6] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, "Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs," *Automatica*, vol. 49, no. 5, pp. 1148–1161, May 2013.

[7] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *Int. J. Robotics Research*, vol. 35, pp. 1261–1285, 2016.

[8] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *J. Guid. Control Dyn.*, vol. 37, no. 6, pp. 1725–1740, 2014.

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

[10] N. Correll and A. Martinoli, "Robust distributed coverage using a swarm of miniature robots," in *IEEE Int. Conf. Robotics Automation*, Rome, Italy, Apr. 2007.

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

[12] D. J. Bruemmer, D. D. Dudenhoeffer, M. D. McKay, and M. O. Anderson, "A robotic swarm for spill finding and perimeter formation," in *Spectrum*, Reno, NV, 2002.

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

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

[15] D. Grünbaum and A. Okubo, "Modelling social animal aggregations," in *Frontiers in Mathematical Biology*. Springer, 1994, pp. 296–325.

[16] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.,*, vol. 34, no. 1, pp. 539–557, 2004.

[17] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 327–341, Feb. 2015.

[18] E. Şahin and W. M. Spears, *Swarm Robotics*. Berlin: Springer Berlin Heidelberg, 2005, vol. 3342.

[19] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, March 2013.

[20] C. Canuto, F. Fagnani, and P. Tilli, "A Eulerian approach to the analysis of rendezvous algorithms," in *Proc. IFAC World Congr.*, 2008.

[21] P. K. Menon, G. D. Sweriduk, and K. D. Bilimoria, "New approach for modeling, analysis, and control of air traffic flow," *J. Guid. Control Dyn.*, vol. 27, no. 5, pp. 737–744, 2004.

[22] P. Kingston and M. Egerstedt, "Index-free multiagent systems: An Eulerian approach," in *Proc. 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, Annecy, France, 2010, pp. 215–220.

[23] R. A. Brooks and T. Lozano-Perez, "A subdivision algorithm in configuration space for findpath with rotation," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 2, pp. 224–233, 1985.

[24] R. V. Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decompositions for path planning using wavelet transforms," in *Proc. IEEE Conf. Decision and Control*, New Orleans, LA, Dec. 2007, pp. 1392–1397.

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

[26] B. Açikmeşe and D. S. Bayard, "A Markov chain approach to probabilistic swarm guidance," in *Proc. Amer. Control Conf.*, June 2012, pp. 6300–6307.

[27] B. Açıkmeşe and D. S. Bayard, "Markov chain approach to probabilistic guidance for swarms of autonomous agents," *Asian Journal of Control*, vol. 17, no. 4, pp. 1–20, 2014.

[28] N. Demir, U. Eren, and B. Açıkmeşe, "Decentralized probabilistic density control of autonomous swarms with safety constraints," *Autonomous Robots*, vol. 39, no. 4, pp. 537–554, 2015.

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

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

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

[32] T. W. Mather and M. A. Hsieh, "Ensemble synthesis of distributed control and communication strategies," in *IEEE Int. Conf. Robotics Automation*, St Paul, MN, May 2012, pp. 4248–4253.

[33] J. Grace and J. Baillieul, "Stochastic strategies for autonomous robotic surveillance," in *Proc. IEEE Conf. Decision and Control*, Seville, Spain, Dec. 2005, pp. 2200–2205.

[34] P. Agharkar, R. Patel, and F. Bullo, "Robotic surveillance and Markov chains with minimal first passage time," in *Proc. IEEE Conf. Decision and Control*, Los Angeles, CA, Dec. 2014.

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

[36] D. P. Bertsekas, *Linear network optimization: algorithms and codes*. MIT Press, 1991.

[37] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using optimal transport," in *Proc. IEEE Conf. Control Applicat.*, Antibes, France, Oct. 2014, pp. 498–505.

[38] D. Milutinović and P. Lima, "Modeling and optimal centralized control of a large-size robotic population," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1280–1285, 2006.

[39] G. Foderaro, S. Ferrari, and T. A. Wettergren, "Distributed optimal control for multi-agent trajectory optimization," *Automatica*, vol. 50, no. 1, pp. 149–154, 2014.

[40] J. Qi, R. Vazquez, and M. Krstic, "Multi-agent deployment in 3-D via PDE control," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 891–906, 2015.

[41] D. A. Dolgov and E. H. Durfee, "The effects of locality and asymmetry in large-scale multiagent MDPs," in *Coordination of Large-Scale Multiagent Systems*, P. Scerri, R. Vincent, and R. Mailler, Eds. Springer, 2005, pp. 3–25.

[42] Y. F. Chen, N. K. Ure, G. Chowdhary, J. P. How, and J. Vian, "Planning for large-scale multiagent problems via hierarchical decomposition with applications to UAV health management," in *Proc. Amer. Control Conf.*, 2014, pp. 1279–1285.

[43] A. Mogilner and L. Edelstein-Keshet, "A non-local model for a swarm," *J. Math. Biology*, vol. 38, no. 6, pp. 534–570, 1999.

[44] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.

[45] S. Zhao, S. Ramakrishnan, and M. Kumar, "Density-based control of multiple robots," in *Proc. Amer. Control Conf.*, San Francisco, CA, 2011.

[46] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "A probabilistic Eulerian approach for motion planning of a large-scale swarm of robots," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Daejeon, Korea, Oct. 2016.

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

[48] A. Olshevsky, "Efficient information aggregation strategies for distributed control and signal processing," Ph.D. dissertation, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2010, p. 99.

[49] G. Habibi and J. McLurkin, "Maximum-leaf spanning trees for efficient multi-robot recovery with connectivity guarantees," in *Distributed Autonomous Robotic Systems*. Springer, 2014, pp. 275–289.

[50] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: covariance intersection revisited," *IEEE Trans. Autom. Control*, vol. 47, no. 11, pp. 1879–1882, 2002.

[51] R. Olfati-Saber, "Kalman-consensus filter : Optimality, stability, and performance," in *IEEE Conf. Decision Control*, Shanghai, China, December 2009, pp. 7036–7042.

[52] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using Bayesian consensus filtering," in *Proc. Amer. Control Conf.*, Portland, OR, June 2014, pp. 634–641.

[53] D. Morgan, G. P. Subramanian, S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic guidance of distributed systems using sequential convex programming," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Chicago, IL, Sept. 2014, pp. 3850–3857.

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

[55] S. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models and Methods in Appl. Sci.*, vol. 1, no. 4, pp. 300–307, 2007.

[56] D. Pollard, "Asymptopia," 2000, manuscript in progress, available at http://www.stat.yale.edu/~pollard.

[57] S. Chib and E. Greenberg, "Understanding the Metropolis–Hastings algorithm," *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

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

[59] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[60] O. Cihan and M. Akar, "Fastest mixing reversible Markov chains on graphs with degree proportional stationary distributions," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 227–232, 2015.

[61] D. B. West, *Introduction to graph theory*. Prentice Hall, 2001, vol. 2.

[62] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, England: Cambridge University Press, 1985.

[63] B. Touri and A. Nedić, "On ergodicity, infinite flow, and consensus in random models," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1593–1605, 2011.

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

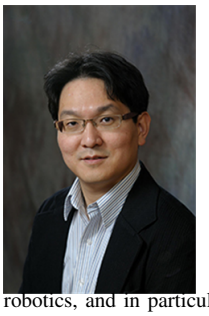
[65] P. Billingsley, *Probability and Measure*. New York: J. Wiley & Sons, 1995.

[66] G. P. Subramanian, "Nonlinear control strategies for quadrotors and CubeSats," Master's thesis, University of Illinois at Urbana-Champaign, 2015.

[67] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65 – 78, 2004.

[68] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European J. Control*, vol. 18, no. 6, pp. 539–557, 2012.

Saptarshi Bandyopadhyay received his Ph.D. in Aerospace Engineering in January 2016 from the University of Illinois at Urbana-Champaign (UIUC), USA. He received his Dual Degree (B.Tech and M.Tech) in Aerospace Engineering in 2010 from Indian Institute of Technology (IIT) Bombay, India. He is currently a Postdoctoral researcher at the Jet Propulsion Laboratory (JPL), California Institute of Technology, USA. His research interests include aerospace systems, robotics, multi-agent systems and swarms, dynamics and controls, estimation theory, probability theory, and systems engineering.



Soon-Jo Chung (M'06–SM'12) received the B.S. degree summa cum laude from Korea Advanced Institute of Science and Technology, in 1998, the S.M. degree in aeronautics and astronautics, and the Sc.D. degree in estimation and control from Massachusetts Institute of Technology, in 2002 and 2007, respectively. He is Associate Professor of Aerospace and Bren Scholar and Jet Propulsion Laboratory Research Scientist in the California Institute of Technology. His research areas include distributed spacecraft systems, aerial swarm robotics, and aerospace robotics, and in particular, on the theory and application of complex nonlinear dynamics, control, estimation, guidance, and navigation of autonomous aerospace systems. Dr. Chung is a recipient of the UIUC Engineering Dean's Award for Excellence in Research, the Beckman Faculty Fellowship of the UIUC Center for Advanced Study, the U.S. Air Force Office of Scientific Research Young Investigator Award, the National Science Foundation Faculty Early Career Development Award, and three Best Conference Paper Awards from the IEEE and the American Institute of Aeronautics and Astronautics.

Fred Y. Hadaegh (S'75–M'80–SM'89–F'07) received the B.S.E.E. (with honors) and M.S.E.E. degrees from the University of Texas at Austin, Austin, TX, USA, in 1975 and the Ph.D. degree in electrical engineering from the University of Southern California, Los Angeles, CA, USA, in 1984. In 1984, he joined the Jet Propulsion Laboratory (JPL), California Institute of Technology, Pasadena, CA, where he is currently JPL's Chief Technologist. For two decades, he supervised the Guidance and Control (G&C) Analysis Group, which is responsible for leading research in G&C theory and for developing algorithms and software for planetary and astrophysics missions, including the Mars Pathfinder, the Mars Exploration Rover, the Cassini Mission to Saturn, the space telescope Spitzer, and the Mars Science Laboratory for the delivery of the Curiosity rover among others. His research interests include optimal estimation and control as applied to distributed spacecraft. Dr. Hadaegh is a JPL Technical Fellow and Senior Research Scientist and a Fellow of The American Institute of Aeronautics and Astronautics. He has been a recipient of numerous awards, including NASA's Exceptional Service and Exceptional Achievement medals and JPL's Award of Excellence for "Flight Validation of Autonomous Rendezvous in Earth Orbit."