INHOMOGENEOUS MARKOV CHAIN APPROACH TO PROBABILISTIC SWARM GUIDANCE ALGORITHMS

Saptarshi Bandyopadhyay(1), **Soon-Jo Chung**(2), **and Fred Y. Hadaegh**(3)

(1) *University of Illinois at Urbana-Champaign, 306 Talbot Lab, 104 S. Wright St., Urbana, IL 61801, (217) 979-8113, bandyop2@illinois.edu*
(2) *University of Illinois at Urbana-Champaign, 306 Talbot Lab, 104 S. Wright St., Urbana, IL 61801, (217) 244-2737, sjchung@illinois.edu*
(3) *Jet Propulsion Laboratory, M/S 198-326, 4800 Oak Grove Dr., Pasadena, CA 91109, (818) 354-8777, Fred.Y.Hadaegh@jpl.nasa.gov*

Abstract

Probabilistic swarm guidance involves designing a homogeneous Markov chain, such that each agent determines its own trajectory in a statistically independent manner. Not only does the overall swarm converge to the desired stationary distribution of the Markov chain but the agents also repair the formation even if sections of the swarm are damaged. In this paper, with the help of communication with neighboring agents, we discuss an inhomogeneous Markov chain approach to probabilistic swarm guidance algorithms for minimizing the number of transitions required for achieving desired swarm distribution and then maintaining it. We first develop a simple method for designing a family of Markov transition matrices for a given stationary distribution, where the tuning parameter dictates the number of transitions. Next, we show that the agents reach an agreement across the swarm on the current swarm distribution by communicating with neighboring agents using the Bayesian Consensus Filtering algorithm. Finally, we prove the convergence and stability guarantees of the proposed algorithm.

Keywords: *Markov chains, consensus, guidance, swarms, formation flying*

1 Introduction

Small satellites are well suited for formation flying missions, where multiple satellites operate together in a cluster or predefined geometry to accomplish the task of a single conventional large satellite. In comparison with traditional large satellites, small satellites are modular in nature and offer low development cost by enabling rapid manufacturing using commercial–off–the–shelf components. Flight of swarms of hundreds to thousands of femtosatellites (100-gram-class satellites) for Synthetic Aperture applications has been discussed in [1]. In this paper, we discuss an inhomogeneous Markov chain approach to probabilistic swarm guidance algorithms for achieving satellite formations and maintaining them.

Consensus algorithms have been extensively studied for formation flying applications [2, 3, 4, 5, 6]. Analogous to fluid mechanics, this traditional view of multi–agent systems is *Lagrangian*, as it deals with an indexed collection of agents [7]. In this paper we adopt an *Eulerian* view, as we study the distribution of index–free agents over the state space. One such probabilistic guidance approach is discussed in [8], where each agent determines its own trajectory without any communication such that the overall swarm converges to a desired distribution. Instead of allocating agent positions ahead of time, probabilistic guidance is based on designing a homogeneous Markov chain, such that the steady-state distribution corresponds to the desired swarm density. Acikmese and Bayard [8] show that, although each agent propagates its position in a statistically independent manner, the swarm asymptotically converges to the desired steady-state distribution associated with the homogeneous Markov chain and also automatically repairs any damage. Similar study on self-organization of swarms using homogeneous Markov chains has been done in [9]. The desired Markov matrices, to guide individual swarm agents in a completely decentralized fashion, are synthesized using the Metropolis-Hastings algorithm [10]. The main limitation of probabilistic guidance using homogeneous Markov chains is that the agents are not allowed to settle down even after the swarm has reached the desired steady-state distribution resulting in significant fuel loss. This paper develops probabilistic swarm guidance algorithms using inhomogeneous Markov chains, with the help of communication with neighboring agents, to address these limitations. In order to achieve these objectives, it is only necessary that each agent communicates with its neighboring agents. Note that inter–agent collisions have been ignored in this paper.

1.1 Notation

Let $\mathcal{R} \in \mathbb{R}^{n_x}$ be the n_x–dimensional compact physical domain over which the swarm is distributed. Let $m \in \mathbb{N}$ agents belong to this swarm. The *time index* is denoted by a right subscript and the *agent index* is denoted by a lower–case right superscript. For example, r_k^j represents the position of the jth agent at the kth time instant.

The communication network topology at the kth time instant is represented by the directed time–varying graph $\mathcal{G}_k$, where all the agents of the system form the set of vertices $\mathcal{V}_k$ and the set of directed edges is denoted by $\mathcal{E}_k$. The neighbors of the jth agent at the kth time instant is the set of agents from which the jth agent receives information at the kth time instant and is denoted by $\mathcal{N}_k^j$. Hence, if $\ell, j \in \mathcal{V}_k$, then $\ell \in \mathcal{N}_k^j$ if and only if $\overrightarrow{\ell j} \in \mathcal{E}_k$. The set of inclusive neighbors of the jth agent is denoted by $\mathcal{J}_k^j \triangleq \mathcal{N}_k^j \cup \{j\}$.

Let x_k represent the true probability mass function (pmf) of the swarm of agents over the space $\mathcal{R}$ at the kth time instant. $x_{k,\nu}^j$ represent the estimated pmf of the swarm of agents over the space $\mathcal{R}$, by the jth agent during the νth consensus loop at the kth time instant. The symbol $\mathcal{P}(\cdot)$ refers to the probability of an event.

Let $\mathbb{N}, \mathbb{Z}^+, \mathbb{C}, \mathbb{R}$ are the sets of natural numbers, non–negative integers, complex numbers and real numbers respectively. Let $\mathbf{1} = [1, 1, \ldots, 1]^T$, $\mathbf{I}$, and $\mathbf{0}$ be the ones vector, the identity matrix, and the zero matrix of appropriate sizes respectively. Finally, $\|\cdot\|_{\ell_2}$, $|\cdot|$ and $\lceil\cdot\rceil$ represent the ℓ_2 norm, absolute value, and ceiling functions respectively.

2 Problem Statement

Similar to the notations adopted in [8], let $\mathcal{R}$ be the physical domain over which the swarm is distributed. It is assumed that region $\mathcal{R}$ is partitioned as the union of n_{cell} disjoint subregions. These subregions or bins are represented by $R_i, i = 1, \ldots, n_{\text{cell}}$, such that $\bigcup_{i=1}^{n_{\text{cell}}} R_i = \mathcal{R}$ and $R_i \cap R_j = \phi$, if $i \neq j$.

Let $m \in \mathbb{N}$ agents belong to this swarm. Let the jth agent have position r_k^j at the kth time instant. Let x_k^j be a column vector of probabilities for the jth agent ($\mathbf{1}^T x_k = 1$), such that the ith element $x_k^j[i]$ is the probability of the event that the jth agent will be in bin R_i at the kth time instant:

$$x_k^j[i] = \mathcal{P}(r_k^j \in R_i), \quad \forall j \in \{1, \ldots, m\}, \forall i \in \{1, \ldots, n_{\text{cell}}\} \tag{1}$$

Let $\boldsymbol{x}_k$ be the swarm distribution, which is a concatenation of all the individual probability vectors of all the agents:

$$\boldsymbol{x}_k = \frac{1}{m} \sum_{j=1}^{m} x_k^j \tag{2}$$

Note that, the ensemble of agent positions $\{r_k^j\}_{k=1}^m$ has a distribution that approaches $\boldsymbol{x}_k$ as the number of agents is increased.

The objective of probabilistic guidance algorithms (PGA) running onboard each agent is to determine its trajectory such that the overall swarm converges to a desired formation. The desired formation is represented as a column vector $\boldsymbol{\pi} \in \mathbb{R}^{n_{\text{cell}}}$ ($\mathbf{1}^T \boldsymbol{\pi} = 1$) over the region $\mathcal{R}$. Given an initial swarm distribution $\boldsymbol{x}_0 \in \mathbb{R}^{n_{\text{cell}}}$, the algorithm guides the agents such that the swarm distribution converges the desired steady–state distribution [8]:

$$\lim_{k \to \infty} \boldsymbol{x}_k[i] = \boldsymbol{\pi}[i], \quad \forall i \in \{1, \ldots, n_{\text{cell}}\} \tag{3}$$

The key idea of PGA using inhomogeneous Markov chains is to synthesize a family of column stochastic Markov transition matrices for each agent $M_k^j \in \mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}, \forall j \in \{1, \ldots, m\}$, called *Markov matrix*, with $\boldsymbol{\pi}$ as their stationary distribution ($M_k^j \geq 0$, $\mathbf{1}^T M_k^j = \mathbf{1}^T$, $M_k^j \boldsymbol{\pi} = \boldsymbol{\pi}$). The entries of matrix M_k^j are defined as transition probabilities at the kth time instant.

$$M_k^j[i, \ell] = \mathcal{P}(r_{k+1}^j \in R_i | r_k^j \in R_\ell), \quad \forall j \in \{1, \ldots, m\} \tag{4}$$

i.e., the jth agent in bin ℓ transitions to bin i during the kth time instant with probability $M_k^j[i, \ell]$. In this paper, we seek to minimize this transition from bin–to–bin during each time step while achieving and maintaining formation, with the help of communication with neighboring agents. The matrix M_k^j determines the time evolution of the probability vector x_k^j as:

$$x_{k+1}^j = M_k^j x_k^j \tag{5}$$

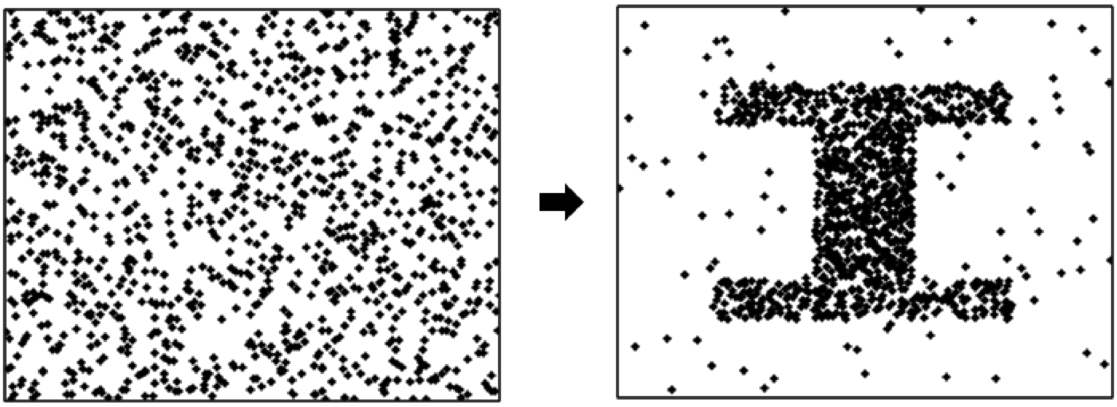

Figure 1: The objective of PGA, running onboard each agent, is to independently determine its trajectory such that the overall swarm converges to a desired final distribution (here letter “I”), starting from any initial distribution.

The first objective of this paper is to design a family of column stochastic Markov matrices with π as their stationary distribution. Note that if each agent recursively determines its own trajectory using these Markov transition matrices, without any communication with other agents, then it has been shown that the overall swarm converges to the desired distribution [8]. The solution to this objective along with a discussion on the tuning parameter, that dictates the number of transitions in each time step, is presented in Section 3.

The motion of agents in the swarm can be viewed as analogous to the random motion of molecules in fluids. Just as the temperature dictates the motion of molecules in fluids; this paper uses the Kullback–Leibler (KL) divergence between the current swarm distribution and the desired stationary distribution, to dictate the motion of agents in the swarm. The KL divergence is a non-symmetric measure of the difference between two probability distributions [11]. Each agent communicates with the agents in its surroundings and makes a localized guess of the current swarm distribution. Using the Bayesian Consensus Filtering algorithm, the agents reach a consensus regarding the current global swarm distribution [12]. This is discussed in detail in Section 4.

The final objective is synthesizing a series of inhomogeneous Markov chains, such that as the KL divergence decreases, the Markov matrices tend towards an identity matrix. The solution to this objective and their stability and convergence guarantees are presented in Section 5. In essence, when the KL divergence between the current swarm distribution and the desired stationary distribution is large, each agent propagates its position in a statistically–independent manner, and the swarm tends toward the desired distribution. When this KL divergence is small, the Markov matrices tend towards an identity matrix and each agent holds its own position.

3 Designing Markov Matrices

In this section, we will design the family of Markov matrices for a given stationary distribution. Note that this solution is much simpler than the recursive Metropolis–

Hastings algorithm discussed in the literature [10]. The following theorem is used by each agent to find the Markov matrix M_k^j during each time instant.

Theorem 3.1. *Let $\boldsymbol{\pi}$ be the desired stationary distribution and let $M^\star = \boldsymbol{\pi}\mathbf{1}^T$. Then a family of column stochastic Markov matrices M_k^j, for the jth agent at the kth time instant, with $\boldsymbol{\pi}$ as their stationary distribution is given by:*

$$M_k^j = \lambda_k^j M^\star + (1 - \lambda_k^j)\mathbf{I}, \qquad 0 \leq \lambda_k^j \leq \lambda_{\max} = \frac{1}{1 - \min_{i \in \{1,2,\dots,n_{cell}\}} \pi[i]} \tag{6}$$

Proof If M_k^j is a Markov matrix, then $\boldsymbol{\pi}$ is its eigenvector corresponding to its largest eigenvalue 1, i.e. $M_k^j\boldsymbol{\pi} = \boldsymbol{\pi}$. Hence from Eq. (6), we get:

$$\begin{aligned} M_k^j\boldsymbol{\pi} &= \lambda_k^j M^\star\boldsymbol{\pi} + (1 - \lambda_k^j)\mathbf{I}\boldsymbol{\pi} \\ &= \lambda_k^j \boldsymbol{\pi}\mathbf{1}^T\boldsymbol{\pi} + (1 - \lambda_k^j)\mathbf{I}\boldsymbol{\pi} \\ &= \lambda_k^j\boldsymbol{\pi} + (1 - \lambda_k^j)\boldsymbol{\pi} = \boldsymbol{\pi} \end{aligned}$$

Hence the matrix M_k^j given by Eq. (6) indeed is a Markov matrix. Moreover, in order to ensure that all the elements in the matrix M_k^j is non–negative, it is obvious that $\lambda_{\max}$ is the unique solution to the equation $\lambda_k^j \left(\min_{i \in \{1,2,\dots,n_{\text{cell}}\}} \pi[i]\right) + (1 - \lambda_k^j) = 0$. ■

Finding the appropriate tuning parameter λ_k^j, introduced in Eq. (6), is discussed in Section 5. If $\lambda_k^j = 0$ then $M_k^j = \mathbf{I}$ and no transitions occur, i.e., the jth agent remains in its original position. On the other hand, we expect that as $\lambda_k^j \to \lambda_{\max}$, the jth agent vigorously moves from one bin to another to achieve the stationary distribution. In the results shown in Figure 2 for a particular test case, we notice that for higher values of λ_k^j, the rate of decrease of KL divergence between the current global distribution and the desired stationary distribution is high but the number of transitions in each time step is also greater. An interesting problem, not discussed in this paper, would be to find the optimum λ_k^j such that the rate of decrease of KL divergence is maximized but the number of transitions in each time step is minimized. In this paper, we are interested in designing inhomogeneous Markov chain such that when the KL divergence is large, we deploy a large tuning parameter λ_k^j to achieve the formation. When the formation is achieved and the resulting KL divergence is small, we reduce the tuning parameter λ_k^j so that the formation is maintained but the number of transitions is less.

In contrast, the homogeneous Markov chain approach to PGA [8] uses common Markov matrix M across all agents for all time instants. The Markov matrix, which also satisfies the motion constraints, is found using the Metropolis–Hastings algorithm. The algorithm is implemented by providing a copy of the matrix M to each of the agents, and then having each agent propagate its position as an independent realization of the Markov chain as shown in Algorithm 1. The first step determines the agent's current bin number. The last two steps sample from the discrete distribution defined by the column of M corresponding to the agent's current bin number [8].

In this paper, each agent executes a different Markov matrix M_k^j during every time step. The tuning parameter λ_k^j depends on the agent's perception of the current swarm distribution. We show that the inhomogeneous Markov chain thus obtained, still guides the agents such that the swarm distribution converges the desired steady–state distribution.

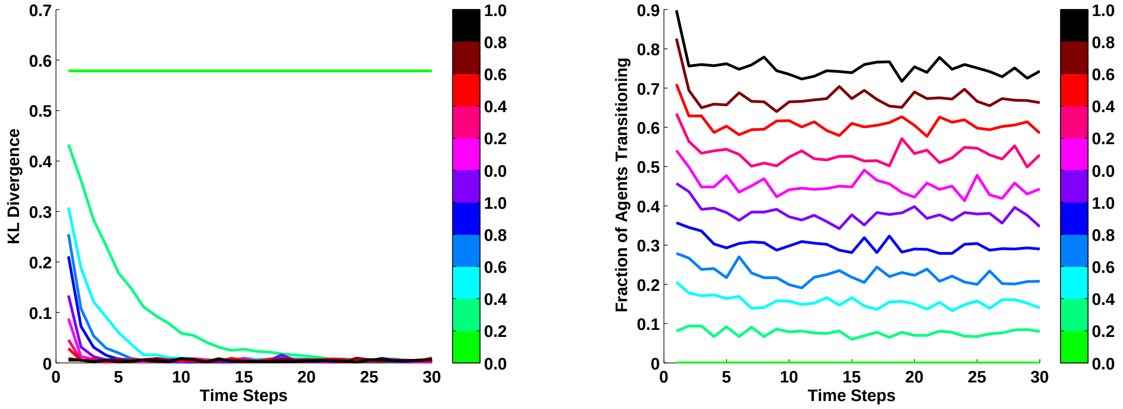

Figure 2: Behavior of the tuning parameter (λ_k^j): (a) the KL divergence falls faster with increasing λ_k^j, and (b) the number of transitions per time step increases with largerλ_k^j .

Algorithm 1 Probabilistic Guidance Algorithm [8]

```
(one cycle of $j$th agent during $k$th time instant)
Agent determines its current bin, e.g., $r_k^j \in R_i$.
Agent generates a random number $z$ that is uniformly distributed in $[0;1]$.
Agent goes to bin $q$ , i.e., $r_{k+1}^j \in R_q$ if $\sum_{\ell=1}^{q-1} M[\ell,j] \leq z \leq \sum_{\ell=1}^{q} M[\ell,j]$.
```

4 Bayesian Consensus Filtering

In the homogeneous Markov chain approach to PGA, no inter–agent communication is needed for the swarm to reach the desired formation but the agents would not realize that the desired formation has been achieved hence they would continue transitioning. On the other hand, in order to minimize the number of transitions to achieve and maintain formation, the agents need to communicate to understand the current swarm distribution and transition accordingly. In this section, we use the decentralized Bayesian Consensus Filtering (BCF) algorithm, proposed in [12], for the agents to reach an agreement on the current swarm distribution.

The objective of BCF is to estimate the current swarm distribution and maintain consensus across the network during each time step. This objective is achieved in two steps: (i) each agent locally estimates the pmf of the swarm, and (ii) the local estimates convergence to the global estimate of the current swarm distribution as each agent recursively transmits its estimated pmf to other agents, receives estimates from its neighboring agents and updates its estimated pmf of the swarm distribution.

In this section, we discuss the BCF to estimate the current swarm distribution which is illustrated in Algorithm 2. Each agent generates a local estimate of the swarm distribution by only determining its current bin location. The local estimate of the swarm's pmf by the jth agent at the start of the consensus stage during the kth time instant is given by $x_{k,0}^j$, where $x_{k,0}^j[i] = 1$ if $r_k^j \in R_i$ else $x_{k,0}^j[i] = 0$.

$$x_{k,0}^j[i] = \begin{cases} 1 & \text{if } r_k^j \in R_i \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

In essence, the local pmf is a discrete representation of the position of the jth agent, within the partitioned space $\mathcal{R}$. Hence the current swarm distribution is given by $\boldsymbol{x}_k = \sum_{i=1}^{m} \frac{1}{m} x^j_{k,0}$.

During the consensus stage, the agents recursively combine and update their local pmfs using the linear opinion pool to reach an agreement across the network. The number of consensus loops ($n_{\text{loop}} \in \mathbb{N}$) depends on the second largest eigenvalue of the matrix representing the balanced communication network topology. Note that $x^j_{k,\nu}$ represents the estimated swarm pmf by the jth agent during the νth consensus loop at the kth time instant. The Linear Opinion Pool (LinOP) of probability measures [13], which has been used for combining subjective probability distributions [14, 15], is described by the following equation [16]:

$$x^j_{k,\nu} = \sum_{\ell \in \mathcal{J}^j_k} a^{j\ell}_{k,\nu-1} x^j_{k,\nu-1}, \; \forall j \in \{1,\dots,m\}, \forall \ell \in \mathcal{J}^j_k, \forall \nu \in \mathbb{N} \tag{8}$$

where $\sum_{\ell \in \mathcal{J}^j_k} a^{j\ell}_{k,\nu-1} = 1$ and the updated pmf $x^j_{k,\nu}$ after the νth consensus loop is a weighted average of the pmfs of the inclusive neighbors $x^j_{k,\nu-1}, \forall \ell \in \mathcal{J}^j_k$ at kth time instant. Let $\mathcal{W}_{k,\nu} = \left(x^1_{k,\nu}, \dots, x^m_{k,\nu}\right)^T$ be an array of pmf estimates of the agents after the νth consensus loop, then the LinOP Eq. (8) can be expressed concisely as:

$$\mathcal{W}_{k,\nu} = P_{k,\nu-1} \mathcal{W}_{k,\nu-1}, \; \forall \nu \in \mathbb{N} \tag{9}$$

where $P_{k,\nu-1}$ is a matrix with entries $[P_{k,\nu-1}]_{j\ell} = a^{j\ell}_{k,\nu-1}$.

Assumption 1. *The communication network topology of the multi–agent system $\mathcal{G}(k)$ is strongly connected (SC). The weighting factors $a^{j\ell}_{k,\nu-1}, \forall j, \ell \in \{1,\dots,m\}$ and the matrix $P_{k,\nu-1}$ have the following properties: (i) the weighting factors are the same for all consensus loops within each time instants, i.e., $a^{j\ell}_{k,\nu-1} = a^{j\ell}_k$ and $P_{k,\nu-1} = P_k, \forall \nu \in \mathbb{N}$; (ii) the matrix P_k conforms with the graph $\mathcal{G}(k)$, i.e., $a^{j\ell}_k > 0$ if and only if $\ell \in \mathcal{J}^j_k$, else $a^{j\ell}_k = 0$; (iii) the matrix P_k is row stochastic, i.e., $\sum_{\ell=1}^{m} a^{j\ell}_k = 1$; and (iv) the weighting factors $a^{j\ell}_k$ are balanced, i.e., the in–degree equals the out–degree $\sum_{\ell \in \mathcal{J}^j_k} a^{j\ell}_k = \sum_{r, s.t. j \in \mathcal{J}^r_k} a^{rj}_k$, where $j, \ell, r \in \{1,\dots,m\}$.*

Theorem 4.1. *[12]* (BCF-LinOP on SC Balanced Digraphs) *Under Assumption 1, using the linear opinion pool Eq. (8), each $x^j_{k,\nu}$ exponentially converges in distribution to the pmf $\boldsymbol{x}_k = \sum_{i=1}^{m} \frac{1}{m} x^j_{k,0}$ with a rate faster or equal to $\lambda_{m-1}(P^2_k)$, i.e., $\lim_{\nu\to\infty} x^j_{k,\nu} \xrightarrow{dist.} \boldsymbol{x}_k$ exponentially, $\forall j \in \{1,\dots,m\}$. The disagreement vector is defined as $\boldsymbol{\theta}_{k,\nu} = \left(\theta^1_{k,\nu},\dots,\theta^m_{k,\nu}\right)^T$, where $\theta^j_{k,\nu} = \sum_{i=1}^{n_{cell}} \left|x^j_{k,\nu}[i] - x_k[i]\right|$. For the ℓ_2 norm of the disagreement vector to be less than some quantity $\epsilon > 0$, i.e., $\left\|\boldsymbol{\theta}_{k,n_{loop}}\right\| \le \epsilon$; the number of consensus loops within each consensus stage should be at least $n_{loop} \ge \left\lceil \frac{\log(\epsilon/2\sqrt{m})}{\log \lambda_{m-1}(P_k)} \right\rceil$.*

Proof Under Assumption 1, P_k is a nonnegative, row stochastic and irreducible matrix which implies that that P_k is a primitive matrix. The Perron–Frobenius Theorem (cf. [17, pp. 3]) states that there exists a left eigenvector $\mathbf{1}$ of P_k corresponding to the eigenvalue 1 which is unique to constant multiples, i.e., $P^T_k \mathbf{1} = 1\mathbf{1}$. The Ergodic Theorem for primitive Markov Chains (cf. [17, pp. 119]) states that $\lim_{\nu\to\infty} P^\nu_k = \frac{1}{m}\mathbf{1}\mathbf{1}^T$. Hence, $\lim_{\nu\to\infty} x^j_{k,\nu} = \boldsymbol{x}_k = \frac{1}{m}\sum_{j=1}^{m} x^j_{k,0}$ is the consensual pmf. By Lemma 1 of [12], the measure induced by $x^j_{k,\nu}$ on $\mathscr{R}$ converges weakly to the measure induced by x_k on $\mathscr{R}$ and $\lim_{\nu\to\infty} x^j_{k,\nu} \xrightarrow{\text{dist.}} x_k$, , where $\mathscr{R}$ is the Borel σ–algebra of $\mathcal{R}$.

Let $\theta^j_{k,\nu} = \sum_{i=1}^{n_{\text{cell}}} \left| x^j_{k,\nu}[i] - x_k[i] \right|$ be the $\mathcal{L}_1$ distance between the local estimated pmf during each consensus loop ($x^j_{k,\nu}$) and the final consensual pmf (x_k). The disagreement vector is defined as $\boldsymbol{\theta}_{k,\nu} = \left(\theta^1_{k,\nu}, \ldots, \theta^m_{k,\nu}\right)^T$, and its dynamics due to the LinOP Eq. (8) is given by $\boldsymbol{\theta}_{k,\nu+1} = P_k \boldsymbol{\theta}_{k,\nu}$. A special case of the Courant–Fisher Theorem (cf. [18, pp. 179]) gives:

$$\max_{[\mathbf{1}]^T \boldsymbol{\theta}_{k,\nu}} \frac{\boldsymbol{\theta}^T_{k,\nu} P_k^2 \boldsymbol{\theta}_{k,\nu}}{\boldsymbol{\theta}^T_{k,\nu} \boldsymbol{\theta}_{k,\nu}} \leq \lambda_{m-1}(P_k^2) \|\boldsymbol{\theta}_{k,\nu}\|^2_{\ell_2} \tag{10}$$

Then the Lyapunov function $\Phi_{k,\nu} = \boldsymbol{\theta}^T_{k,\nu} \boldsymbol{\theta}_{k,\nu} = \|\boldsymbol{\theta}_{k,\nu}\|^2_{\ell_2}$ vanishes exponentially as $\Phi_{k,\nu+1} \leq \lambda_{m-1}(P_k^2)\Phi_{k,\nu}$. Since $\|\boldsymbol{\theta}_{k,0}\|_{\ell_2} \leq 2\sqrt{m}$, the number of loops in the consensus stage are $n_{\text{loop}} \geq \left\lceil \frac{\log(\epsilon/2\sqrt{m})}{\log \lambda_{m-1}(P_k)} \right\rceil$. ■

It follows from the above Theorem, that the consensual pmf $x^j_{k,n_{\text{loop}}}$ is the average of the individual pmfs at the start of the consensus stage $x^j_{k,0}, \forall j \in \{1, \ldots, m\}$. Hence the agents indeed exponentially convergence to the global estimate of the current swarm distribution during the consensus stage.

5 Designing Inhomogeneous Markov Chains

In this section, we shall study the stability and convergence characteristics of the inhomogeneous Markov chains which converge to the given stationary distribution. Let $\boldsymbol{\pi}$ represent the desired pmf of the swarm of satellites over the space $\mathcal{R}$. Each agent chooses the tuning parameter based on the KL divergence of the current swarm distribution from the desired distribution, using the following equation:

$$\lambda^j_k = \min\left(\lambda_{\max}, D_{KL}(\boldsymbol{\pi} || x^j_{k,n_{\text{loop}}})\right) \quad \text{where } D_{KL}(\boldsymbol{\pi} || x^j_{k,n_{\text{loop}}}) = \sum_{i=1}^{n_{\text{cell}}} \boldsymbol{\pi}[i] \ln \left(\frac{\boldsymbol{\pi}[i]}{x^j_{k,n_{\text{loop}}}[i]} \right) \tag{11}$$

Each agent shall then choose the appropriate Markov matrix using Theorem 3.1 with the tuning parameter given by Eq. (11). Conditions for such series of inhomogeneous Markov matrices that tend to the same stationary distribution, called 'hardening-position scheme', are discussed in [19].

From Eq. (5), the overall time evolution of the probability vector x^j_k is given by:

$$x^j_k = M^j_{k-1} M^j_{k-2} \ldots M^j_1 M^j_0 x^j_0 \tag{12}$$

where each $M^j_\ell, \ell = 0, \ldots, k-1$ are $\mathbb{R}^{n_{\text{cell}} \times n_{\text{cell}}}$ are column stochastic matrices obtained using Theorem 3.1. This is similar to inhomogeneous Markov chains, except that the matrix product goes backward, hence we shall use only those results which are direction free. Let us denote this backward matrix product $M^j_{k-1} M^j_{k-2} \ldots M^j_1 M^j_0$ by $\prod_{\ell=0}^{k-1} M^j_\ell$. The main convergence result of the sequence $\prod_{\ell=0}^{k-1} M^j_\ell$ from Eq. (12) is given by the following theorem, which has been partially adapted from the homogeneous case discussed in [8].

Algorithm 2 PGA using inhomogeneous Markov chains

```
(one cycle of $j$th agent during $k$th time instant)
Agent determines its current bin, e.g., $r_k^j \in R_i$.
Set $n_{\text{loop}}$, the weighting factors $a_k^{j\ell}$          }Theorem $4.1$
for $\nu = 1$ to $n_{\text{loop}}$                                   ⎫
    if $\nu = 1$ then                                                ⎪
        Set $x_{k,0}^j$ from $r_k^j$                                 ⎪
    end if                                                           ⎬ Consensus Stage
    Transmit the pmf $x_{k,\nu-1}^j$ to other agents                 ⎪
    Obtain the pmfs $x_{k,\nu-1}^j, \forall \ell \in \mathcal{J}_k^j$ from neighboring agents
    Compute the new pmf $x_{k,\nu}^j$ using LinOP Eq. (8)            ⎪
end for                                                              ⎭
Compute the tuning parameter $\lambda_k^j$ using Eq. $(11)$
Compute the Markov matrix $M_k^j$                                    }Theorem $3.1$
Agent generates a random number $z \in \mathrm{unif}[0;1]$
Agent goes to bin $q$ , i.e., $r_{k+1}^j \in R_q$
if $\sum_{\ell=1}^{q-1} M_k^j[\ell,j] \leq z \leq \sum_{\ell=1}^{q} M_k^j[\ell,j]$.
```

Theorem 5.1. *(PGA using inhomogeneous Markov chains) Consider the Markov chain in Eq. (5) with column stochastic matrices M_k^j and $\boldsymbol{\pi}$ as their stationary distribution. Then for any initial probability vector $\boldsymbol{x}_0 \in \mathbb{R}^{n_{cell}}$, the global swarm distribution $\boldsymbol{x}_k$ asymptotically converges in distribution to the desired pmf $\boldsymbol{\pi}$, i.e., $\lim_{k\to\infty} \boldsymbol{x}_k \xrightarrow{dist.} \boldsymbol{\pi}$.*

Proof Theorem 1 of [19] states that for the matrix product in Eq. (12), weak and strong ergodicity is equivalent. Moreover, the sequence $\prod_{\ell=0}^{k-1} M_\ell^j$ is ergodic as there is only one absolute probability vector $\boldsymbol{\pi}$ common to all matrices [20]. Hence, the sequence is strongly ergodic (in backward direction) and $\lim_{k\to\infty} \prod_{\ell=0}^{k-1} M_\ell^j = \boldsymbol{\pi}\mathbf{1}^T$. Hence, not only do the agent's distribution converge with respect to each other, but also the swarm distribution tends to stabilize in time to a fixed distribution [19].

Next, we need to show that the final swarm distribution is indeed the desired distribution given by $\boldsymbol{\pi}$. We first prove by induction that:

$$\prod_{\ell=0}^{k-1} (M_\ell^j - \boldsymbol{\pi}\mathbf{1}^T) = \prod_{\ell=0}^{k-1} M_\ell^j - \boldsymbol{\pi}\mathbf{1}^T \tag{13}$$

Since Eq. (13) is true for $k = 1$, we assume it is true for k. Then for $k+1$ we get:

$$\begin{aligned}
\prod_{\ell=0}^{k} (M_\ell^j - \boldsymbol{\pi}\mathbf{1}^T) &= \left(\prod_{\ell=0}^{k-1} M_\ell^j - \boldsymbol{\pi}\mathbf{1}^T \right) (M_k^j - \boldsymbol{\pi}\mathbf{1}^T) \\
&= \left(\prod_{\ell=0}^{k} M_\ell^j - \boldsymbol{\pi}\mathbf{1}^T M_k^j \right) + \left(\boldsymbol{\pi}\mathbf{1}^T \boldsymbol{\pi}\mathbf{1}^T - \prod_{\ell=0}^{k-1} M_\ell^j \boldsymbol{\pi}\mathbf{1}^T \right) \\
&= \prod_{\ell=0}^{k-1} M_\ell^j - \boldsymbol{\pi}\mathbf{1}^T
\end{aligned}$$

Due to strong ergodicity, Eq. (13) vanishes which implies that $\rho(M_k^j - \boldsymbol{\pi}\mathbf{1}^T) < 1, \forall k \in \mathbb{N}$. It follows from Theorem 2 of [8] that for any a probability vector $x_0^j \in \mathbb{R}^{n_{\text{cell}}}$, $\lim_{k\to\infty} x_k^j = \boldsymbol{\pi}$ if and only if $\rho(M_k^j - \boldsymbol{\pi}\mathbf{1}^T) < 1$. This result follows from the error dynamics of $e_k^j (= x_k^j - \boldsymbol{\pi})$ which evolves as $e_{k+1}^j = (M_k^j - \boldsymbol{\pi}\mathbf{1}^T)e_k^j$.

Since the individual probability distribution of agents converges to $\boldsymbol{\pi}$, from Eq. (2) we get $\lim_{k\to\infty} \boldsymbol{x}_k = \boldsymbol{\pi}$. Finally, by Lemma 1 of [12] the measure induced by $\boldsymbol{x}_k$ on $\mathscr{R}$ converges weakly to the measure induced by $\boldsymbol{\pi}$ on $\mathscr{R}$ and $\lim_{k\to\infty} \boldsymbol{x}_k \xrightarrow{\text{dist.}} \boldsymbol{\pi}$, where $\mathscr{R}$ is the Borel σ–algebra of $\mathcal{R}$. ■

Thus we have shown that the swarm executing the PGA using inhomogeneous Markov chain asymptotically converges to the desired stationary distribution.

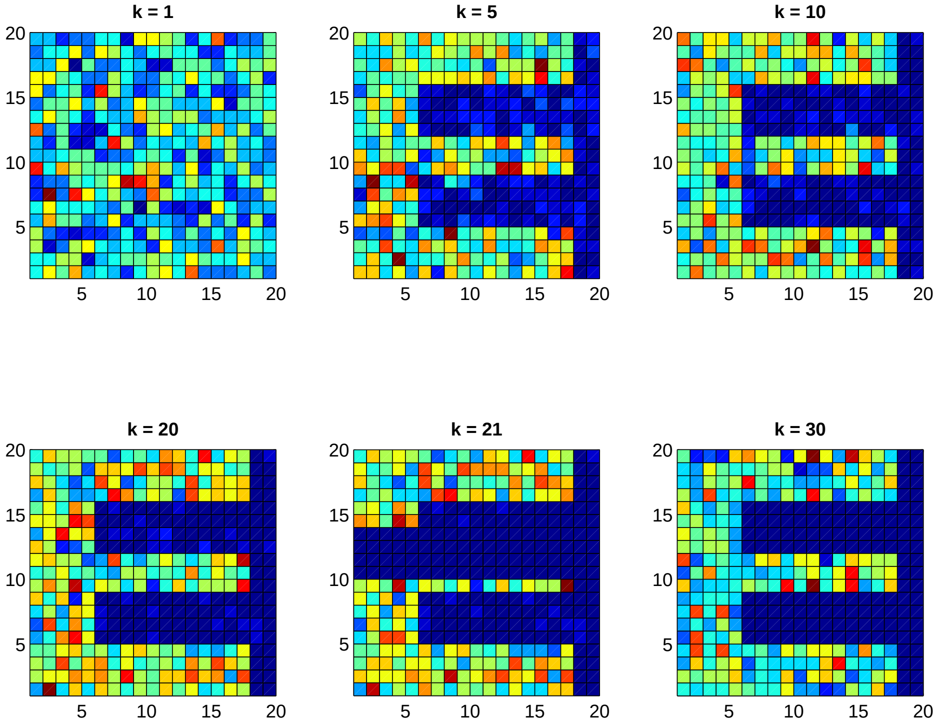

Figure 3: Histogram plots of the swarm distribution at different time instants. Starting form an uniform distribution, the swarm converges to a desired distributions within 20 time steps. The middle section of the swarm is damaged during the 21st time step, but the swarm autonomously recovers within 10 time steps.

6 Numerical Example

This example demonstrates decentralized swarm guidance using PGA while minimizing transitions. The swarm contains $m = 2000$ agents that are guided to form the probability distributions $\boldsymbol{\pi}$ associated with the letter "E". As shown in Figure 3, the scenario starts at $k = 1$ with the swarm uniformly distributed across $\mathcal{R}$. Each agent independently executes the PGA using inhomogeneous Markov chains illustrated in Algorithm 2. As shown in the KL divergence graph in Figure 4, the desired distribution is almost achieved within the first 10 time steps.

After the 20th time step, the swarm is damaged by removing approximately 340 agents from the middle section of the formation. This can been seen by comparing the images for the 20th and 21st time step in Figure 3. Note that the swarm quickly recovers from this damage and the remaining agents attain the desired stationary distribution within another 10 time steps. From the great match between the plots of KL divergence and number of transitions in Figure 4, we can infer that the agents in the swarm actively move only when it is required of them to do so.

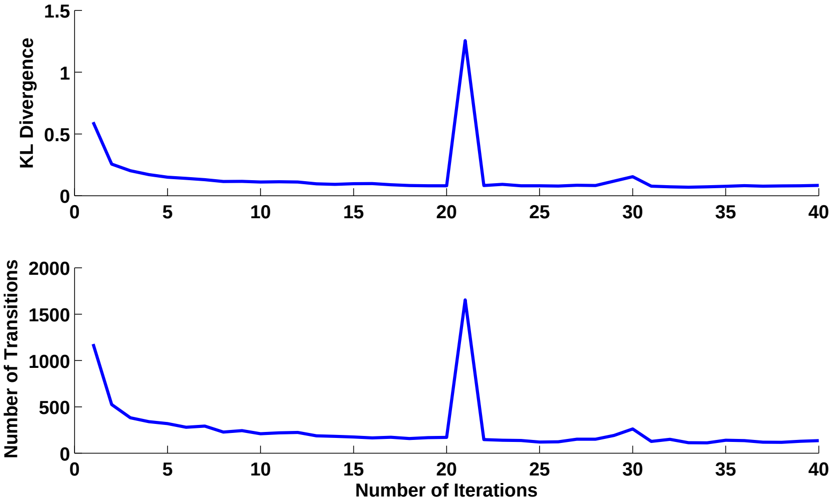

Figure 4: Convergence of the swarm to the desired distribution in terms of KL divergence and number of transitions per time step. Note the spike during the 21st time step due to the damage in the middle section of the swarm.

7 Conclusions

We have extended the scope of probabilistic guidance algorithms by augmenting it with inhomogeneous Markov chains for minimizing the number of transitions for achieving and maintaining formations. Note that the algorithm requires the agents to communicate with their neighboring agents. The swarm converges asymptotically to the desired distribution and the algorithm is robust to external disturbances or damages to sections of the swarm. A simulation result demonstrates the properties of convergence and self-repair. Future directions of research should include motion constraints and avoid inter–agent collision.

Acknowledgment

The authors would like to thank Dr. David Bayard from JPL, NASA; Prof. Behcet Acikmese from UT Austin and Abhishek Gupta, Dr. Ashwin Dani, Dr. Aditya Paranjape, and Ashish Mishra from UIUC for stimulating discussions about this paper. This research was supported by AFOSR grant (FA95501210193). This research was carried out in part at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. © 2013 California Institute of Technology.

References

[1] S.-J. Chung and F. Y. Hadaegh, "Swarms of femtosats for synthetic aperture applications," in *4th International Conference on Spacecraft Formation Flying Missions and Technologies*, (St.-Hubert, Quebec), May 2011.

[2] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520 – 1533, 2012.

[3] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, pp. 988 – 1001, 2003.

[4] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, pp. 2508 – 2530, June 2006.

[5] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, pp. 655 – 661, May 2005.

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

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

[8] B. Acikmese and D. S. Bayard, "A markov chain approach to probabilistic swarm guidance," in *American Control Conference*, pp. 6300–6307, June 2012.

[9] I. Chattopadhyay and A. Ray, "Supervised self-organization of homogeneous swarms using ergodic projections of markov chains," *IEEE Transactions on Systems, Man., and Cybernetics*, vol. 39, no. 6, pp. 1505–1515, 2009.

[10] L. J. Billera and P. Diaconis, "A geometric interpretation of the metropolis-hastings algorithm," *Statistical Science*, vol. 16, no. 4, pp. 335–339, 2001.

[11] T. M. Cover and J. A. Thomas, *Elements of information theory*. New York, NY: Wiley, 1991.

[12] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using bayesian consensus filtering," *IEEE Transaction in Automatic Control*. To be submitted.

[13] C. Genest and J. V. Zidek, "Combining probability distributions: A critique and an annotated bibliography," *Statistical Science*, vol. 1, no. 1, pp. 114 – 135, 1986.

[14] M. H. DeGroot, “Reaching a consensus,” *Journal of the American Statistical Association*, vol. 69, no. 345, pp. 688 – 704, 1960.

[15] K. Lehrer, “When rational disagreement is impossible,” *No$\hat{u}$s*, vol. 10, no. 3, pp. 327 – 332, 1976.

[16] M. Bacharach, “Normal bayesian dialogues,” *Journal of the American Statistical Association*, vol. 74, no. 368, pp. 837 – 846, 1979.

[17] E. Seneta, *Non-negative matrices and Markov chains*. New York, NY: Springer-Verlag, 2006.

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

[19] S. Chatterjee and E. Seneta, “Towards consensus: Some convergence theorems on repeated averaging,” *Journal of Applied Probability*, vol. 14, no. 1, pp. 89 – 97, 1977.

[20] D. Blackwell, “Finite non-homogeneous chains,” *Annals of Mathematics*, vol. 46, pp. 594–599, Oct. 1945.