2012 American Control Conference
Fairmont Queen Elizabeth, Montréal, Canada
June 27-June 29, 2012

A Markov Chain Approach to Probabilistic Swarm Guidance

Behçet Açıkmeşe and David S. Bayard

Abstract— **This paper introduces a probabilistic guidance approach for the coordination of swarms of autonomous agents. The main idea is to drive the swarm to a prescribed density distribution in a prescribed region of the configuration space. In its simplest form, the probabilistic approach is completely decentralized and does not require communication or collaboration between agents. Agents make statistically independent probabilistic decisions based solely on their own state, that ultimately guides the swarm to the desired density distribution in the configuration space. In addition to being completely decentralized, the probabilistic guidance approach has a novel autonomous self-repair property: Once the desired swarm density distribution is attained, the agents automatically repair any damage to the distribution without collaborating and without any knowledge about the damage.**

I. Introduction

This paper introduces a probabilistic guidance approach applicable to a swarm of autonomous agents. The probabilistic guidance approach provides a method for each agent to determine its own trajectory in the configuration variable such that the overall swarm converges to a desired distribution in the configuration space. The main novel feature is that the guidance law is probabilistic in nature, and specifies the desired spatial probability density distribution for the swarm rather than the exact desired paths of the individual agents.

Existing guidance methods for distributed systems allocate agent positions ahead of time [24], [31], [33], [32], [19]. Probabilistic guidance represents a break with this approach, and is instead based on designing a Markov chain, using a suitable parametrization, such that the steady-state distribution corresponds to the desired swarm density.In real time, each agent propagates its position as a statistically independent realization of the Markov chain. The swarm converges to the desired steady-state distribution associated with the Markov chain. Various different schemes applicable to swarm guidance have appeared in the literature [2], [17], [29], [20], [27], [26], [7]. In preparing this manuscript, a literature survey has turned up a recent paper [6] that uses Markov chains for "swarm self-organization", which appears similar to how the current paper uses Markov chains for swarm guidance. However, [6] controls the swarm using a probabilistic "disablement" approach based on ideas taken from [21]. The current paper does not use disablement or any ideas from [21], and the main results are not directly comparable. Nevertheless, we acknowledge that the basic idea of using Markov chains for controlling swarms is also contained in this recent paper.

This paper presents a new approach to swarm guidance based on Metropolis-Hastings algorithm [10], [3], [22] to synthesize the desired Markov matrices and to guide individual swarm agents in a completely decentralized fashion. The probabilistic guidance development relies heavily on the theory of Markov processes, Monte-Carlo-Markov-Chain sampling methods [28], [9], [14], [23], [13], graph theory [8], and Lyapunov stability analysis [25]. The development is aided by recent research in designing fast mixing Markov chains that converge to desired distributions and incorporate constraints on transition probabilities [11], [5], [4], and many classical results on convergence of Markov chains. One example is the use of the Metropolis-Hastings algorithm [10], [3], [22] to generate a Markov guidance policy that is both convergent and incorporates motion constraints. A new useful connection is made between classical Perron-Frobenius theory [15], [1], and modern Lyapunov stability theory [18]. This connection allows several generalizations to be made to classical results that are particularly relevant to designing practical guidance laws. This method allows strict enforcement of ”keep-out” regions where agents are not allowed, and lead to synthesis of probabilistic guidance laws that are fast converging, fuel minimizing, and incorporate physical motion constraints.

The paper is organized as follows: Section II and III presents the probabilistic guidance problem formulation and review of basic results on the Markov chain based swarm guidance; Section IV presents useful connections with Perron-Frobenius theory; Section V has the main results of the paper, Theorems 4 and 5, on the Metropolis-Hastings to synthesize probabilistic guidance laws; and Section VI presents an illustrative example.

Notation

The following is a partial list of notation used: $\mathbf{0}$ is the zero matrix of appropriate dimensions; e_i is a vector of appropriate dimension with its ith entry $+1$ and its other entries zeros; $x[i] = \mathrm{e}_i^T x$ for any $x \in I\!R^n$ and $A[i,j] = \mathrm{e}_i^T A \mathrm{e}_j$ for any $A \in I\!R^{n\times m}$; $Q = Q^T \succ (\succeq) 0$ implies that Q is a symmetric positive (semi-)definite matrix; $R > (\geq) H$ implies that $R[i,j] > (\geq) H[i,j]$ for all i,j; $R > (\geq) \mathbf{0}$ implies that R a positive (non-negative) matrix; $v \in I\!R^n$ is said to be a *probability vector* if $v \geq \mathbf{0}$ and $\mathbf{1}^T v = 1$; $B = |A|$ implies that $B[i,j] = |A[i,j]|$ for all i,j; $\mathcal{P}$ denotes

Authors are with Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California, USA. `behcet@jpl.nasa.gov`, `david.s.bayard@jpl.nasa.gov`.

Approved for Public Release, Distribution Unlimited. The views, opinions, and/or findings contained in this article/presentation are those of the author/presenter and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

978-1-4577-1094-0/12/$26.00 ©2012 AACC

probability of a random variable; $\emptyset$ denotes the empty set; $\|v\|$ is the 2-norm of the vector v; For $P = P^T \succ 0$, $\|v\|_P = \|P^{1/2}v\|$ where $P = P^{1/2}P^{1/2}$ is a factorization of P with $P^{1/2} = P^{1/2^T} \succ 0$ (for concreteness, one choice is $P^{1/2} = U\Lambda^{1/2}U^T$ where $P = U\Lambda U^T$ is an eigenvector decomposition of P); I is the identity matrix; $\mathbf{1}$ is the matrix of ones with appropriate dimensions; $(v_1, v_2, ..., v_n)$ represents a vector obtained by augmenting vectors $v_1, \ldots, v_n$ such that $(v_1, v_2, ..., v_n) \equiv \begin{bmatrix} v_1^T & v_2^T & \ldots & v_n^T \end{bmatrix}^T$ where v_i have arbitrary dimensions; $\text{diag}(A) = (A[1,1], \ldots, A[n,n])$ for matrix A; $\lambda_{max}(P)$ and $\lambda_{min}(P)$ are maximum and minimum eigenvalues of $P = P^T$; $\sigma(A)$ is the spectrum (set of eigenvalues) of A; $\rho(A)$ is the spectral radius of A $(\max_{\lambda\in\sigma(A)} |\lambda|)$; $||A||_1 = \sum_i \sum_j |A[i,j]|$ denotes the 1-norm of matrix A, and $|||A|||_1 = \max_j \sum_i |A[i,j]|$ denotes the induced 1-norm of matrix A; $\otimes$ denotes the Kronecker product; $\odot$ represents the Hadamard (Schur) product; $\mathtt{i}(A)$ is the indicator matrix for any matrix A, whose entries are given by $\mathtt{i}(A)[i,j] = 1$ if $A[i,j] \neq 0$ and $\mathtt{i}(A)[i,j] = 0$ otherwise. A *directed graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is defined by a finite set of *vertices* $\mathbf{V}$ and *edges* $\mathbf{E}$ such that the edges $\mathbf{E} \in \mathbf{V} \times \mathbf{V}$ contain specified ordered pairs of elements of $\mathbf{V}$. A *directed graph* $\mathbf{G}_a(A) = (\mathbf{V}_a, \mathbf{E}_a)$ *of a matrix* A is defined by letting $\mathbf{V}_a$ be the set of integers $1, 2, ..., n$ and letting $\mathbf{E}$ be the set of such pairs $(i,j),\ i \in \mathbf{V}_a, j \in \mathbf{V}_a$ for which $A[i,j] \neq 0$. The *adjacency matrix* A_a of a graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is defined such that $A_a[i,j] = 1$ if $(i,j) \in \mathbf{E}$ and $A_a[i,j] = 0$ otherwise. In particular, if the graph $\mathbf{G}_a(A)$ is associated with a matrix A, then $A_a = \mathtt{i}(A)$.

II. Swarm Distribution Guidance Problem

This section describes the swarm distribution guidance problem. The physical domain over which the swarm agents are distributed is denoted as $\mathcal{R}$. It is assumed that region $\mathcal{R}$ is partitioned as the union of m disjoint subregions $R_i,\ i = 1, \ldots, m$, such that $\mathcal{R} = \bigcup_{i=1}^{m} R_i$, and $R_i \cap R_j = \emptyset$ for $i \neq j$. The subregions R_i are referred to as *bins*.

Let an agent have position $r(t)$ at time index $t \in IN_+$. Let $x(t)$ be a vector of probabilities, $\mathbf{1}^T x(t) = 1$, such that the i'th element $x[i](t)$ is the probability of the event that this agent will be in bin R_i at time t,

$$x[i](t) := \mathcal{P}(r(t) \in R_i). \tag{1}$$

The time index t will also be referred to as the "stage" in the remainder of the paper. Consider a swarm comprised of N agents. Each agent is assumed to act independently of the other agents, so that (1) holds for N separate events,

$$x[i](t) := \mathcal{P}(r_k(t) \in R_i), \quad k = 1, ..., N \tag{2}$$

where $r_k(t)$ denotes the position of the k'th agent at time index t, and the probabilities of these N events are jointly statistically independent. We refer to $x(t)$ as the *swarm distribution*. This is to be distinguished from the ensemble of agent positions $\{r_k(t)\}_{k=1}^N$ which, by the law of large numbers, has a distribution that approaches $x(t)$ as the number of agents N is increased.

The distribution guidance problem is defined as follows: Given any initial distribution $x(0)$ such that $x(0) \in IR^m$, $x(0) \geq \mathbf{0}$, $\mathbf{1}^T x_0 = 1$, it is desired to guide the agents toward a specified steady-state distribution described by a probability vector $v \in IR^m$

$$\lim_{t\to\infty} x[i](t) = v[i] \quad \text{for} \quad i = 1, \ldots, m, \tag{3}$$

subject to motion constraints specified in terms of an adjacency matrix A_a as follows:

$$A_a^T[i,j] = 0 \ \Rightarrow\ r(t+1) \notin R_i \text{ when } r(t) \in R_j,\ \forall t \in IN_+. \tag{4}$$

Here, the adjacency matrix A_a of the edges of a directed graph specifies the allowable transitions between bins.

The desired distribution v has the following interpretation: We have m bins in the physical space corresponding to where agents can be located, and the element $v[i]$ is the desired probability of finding an agent in the i'th bin. If there are N agents total, then $Nv[i]$ describes the expected number of agents to be found in the i'th bin. Let $\mathbf{n} = [n[1], ..., n[m]]^T$ denote the actual number of agents in each bin. Then the number of agents $n[i]$ found in the i'th bin will generally be different from $Nv[i]$, although it follows from the independent and identically distributed (iid) agent realizations that $v = E[\mathbf{n}]/N$, and from the law of large numbers that $\mathbf{n}/N \to v$ as N becomes large. Hence the vector v is a discrete probability distribution specifying the desired average fraction of agents in each bin of the physical domain, that, in practice, will only be approximated by the histogram n/N of agents. However, the nature of the approximation is that v is equal to the mean $E[n/N]$ of the agent histogram, and by the law of large numbers, $n/N \to v$ as N becomes large.

The idea behind probabilistic guidance is to control the propagation of probability vector x, rather than individual agent positions $\{r_k(t)\}_{k=1}^N$. While the actual distribution of swarm agent positions n/N will generally be different from x, it will always be equal to x *on the average*, and can be made *arbitrarily close* to x by using a sufficiently large number of agents. In this sense, probabilistic guidance exploits the law of large numbers to simplify the coordination of swarms comprised of a statistical ensemble (i.e., a significantly large number), of agents.

III. Decentralized Probabilistic Swarm Guidance

A. Probabilistic Guidance Algorithm (PGA)

Suppose that it is desired for the swarm to assume a particular agent distribution described by the vector v. The key idea of the probabilistic guidance law is to synthesize a column stochastic matrix [15], [1] $M \in IR^{m\times m}$, which we call *Markov matrix for PGA*, with v as its eigenvector corresponding to its largest eigenvalue 1 [15], [12], that is, M must satisfy

$$M \geq \mathbf{0}, \quad \mathbf{1}^T M = \mathbf{1}^T, \quad Mv = v. \tag{5}$$

The entries of matrix M are defined as transition probabilities. Specifically, for any $t \in IN_+$ and $i, j = 1, \dots, m$

$$M[i,j] = \mathrm{e}_i^T M \mathrm{e}_j = \mathcal{P}\left(r(t+1) \in R_i | r(t) \in R_j\right). \qquad (6)$$

i.e., an agent in bin j transitions to bin i between two consecutive stages with probability $M[i,j]$. The matrix M determines the time evolution of the probability vector x as

$$x(t+1) = Mx(t) \qquad t = 0, 1, 2, \dots. \qquad (7)$$

Note that the probability vector $x(t)$ stays normalized as $\mathbf{1}^T x(t) = 1$ for all $t \geq 0$. This follows from the fact that $\mathbf{1}^T x(0) = 1$ and $\mathbf{1}^T M = \mathbf{1}^T$, which implies that $\mathbf{1}^T M^t x(0) = \mathbf{1}^T M^{(t-1)} x(0) = \dots = \mathbf{1}^T x(0) = 1$. The probabilistic guidance problem becomes one of designing a specific Markov process (7) for x that converges to a desired distribution v. The constraints $M \geq \mathbf{0}$ and $\mathbf{1}^T M = \mathbf{1}^T$ simply state that the probability of moving from one bin to another is nonnegative and the sum of probabilities of motion from a given bin is one. The constraint $Mv = v$ guarantees that v is a stationary distribution of M, which follows from the equation (7). Here having $Mv = v$ implies that: If $x(T) = v$ for some $T \geq 0$ then $x(t) = v$ for all $t \geq T$. This implies that v is a stationary distribution of M, i.e., the probability distribution of the agents does not change with time. The evolution of the probability density is described for a time-varying M by the following theorem.

Theorem 1: Suppose we have a swarm of N agents in a partitioned region $R = \cup_{i=1}^m R_i$ where $R_i \cap R_j = \emptyset$ for $i \neq j$. Let $x[i](t) = \mathcal{P}(r(t) \in R_i)$ where $r(t)$ is the position vector of an agent at time instance t, and

$$M[i,j](t) := \mathcal{P}(r(t+1) \in R_i | r(t) \in R_j). \qquad (8)$$

Then the probability density vector x over R evolves as

$$x(t+1) = M(t)x(t). \qquad (9)$$

Proof: Since the event of an agent being in bin i at time t is mutually exclusive from it being in another bin j and these events are exhaustive, i.e., they cover all possibilities. In this case, the Total Probability theorem [30] implies that, $\mathcal{P}(r(t+1) \in R_i) = \sum_{i=1}^m \mathcal{P}(r(t+1) \in R_i | \mathcal{P}(r(t) \in R_j))\mathcal{P}(r(t) \in R_j)$. Hence, since $M[i,j](t) = \mathcal{P}(r(t+1) \in R_i | \mathcal{P}(r(t) \in R_j))$, $x[i](t+1) = \mathcal{P}(r(t+1) \in R_i)$, and $x[j](t) = \mathcal{P}(r(t) \in R_j)$, the equation (9) follows. ■

B. Independent Agent Realizations

The probabilistic guidance 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

Probabilistic Guidance Algorithm (PGA)

1) Each agent determines its current bin, e.g., $r_k(t) \in R_i$.
2) Each agent generates a random number z that is uniformly distributed in $[0,1]$.
3) Each agent goes to bin j, i.e., $r_k(t+1) \in R_j$, if $\sum_{l=1}^{j-1} M[l,i] \leq z \leq \sum_{l=1}^{j} M[l,i]$.

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.

C. Asymptotic Convergence

A matrix M that satisfies constraints (5) is a column stochastic matrix where v is the eigenvector corresponding to eigenvalue 1. It is desired for x to asymptotically converge to v, i.e., for v to be a globally attractive stationary distribution for M. The main result of this section shows that asymptotic convergence to v is ensured by imposing an additional constraint on the design of matrix M, denoted as the *spectral radius condition*,

$$\rho(M - v\mathbf{1}^T) < 1. \qquad (10)$$

Following theorem gives a necessary and sufficient conditions for asymptotic convergence to v.

Theorem 2: Consider the Markov chain with column stochastic matrix M such that $Mv = v$. Then for any a probability vector $x(0) \in IR^m$, it follows that $\lim_{t\to\infty} x(t) = v$ for the system (7) if and only if $\rho(M - v\mathbf{1}^T) < 1$.

Proof: First we show that

$$(M - v\mathbf{1}^T)^t = M^t - v\mathbf{1}^T. \qquad (11)$$

By inspection (11) is true for $t = 1$. Suppose that $(M - v\mathbf{1}^T)^{t-1} = M^{t-1} - v\mathbf{1}^T$, then $(M - v\mathbf{1}^T)^t = (M - v\mathbf{1}^T)^{t-1}(M - v\mathbf{1}^T) = (M^{t-1} - v\mathbf{1}^T)(M - v\mathbf{1}^T) = M^t - M^{t-1}v\mathbf{1}^T - v(\mathbf{1}^T M) + v(\mathbf{1}^T v)\mathbf{1}^T = M^t - v\mathbf{1}^T - v\mathbf{1}^T + v\mathbf{1}^T = M^t - v\mathbf{1}^T$. Since $x(0) \geq \mathbf{0}$ and $\mathbf{1}^T x(0) = 1$,

$$x(t) \geq 0 \qquad \text{and} \qquad \mathbf{1}^T x(t) = 1, \qquad t = 0, 1, \dots$$

Let $e(t) := x(t) - v$ be the error relative to the desired distribution v. Then, by using the above observations, we can express the error dynamics as $e(t+1) = x(t+1) - v = Mx(t) - v = Mx(t) - v\mathbf{1}^T x(t) = (M - v\mathbf{1}^T)x(t) = (M - v\mathbf{1}^T)(e(t) + v) = (M - v\mathbf{1}^T)e(t) + (M - v\mathbf{1}^T)v$
$= (M - v\mathbf{1}^T)e(t)$. This proves that e evolves as

$$e(t+1) = (M - v\mathbf{1}^T)e(t), \qquad t = 0, 1, \dots \qquad (12)$$

If $\rho(M - v\mathbf{1}^T) < 1$ then the error dynamics will be asymptotically stable, i.e., $\lim_{t\to\infty} e(t) = \mathbf{0}$. This implies that $\lim_{t\to\infty} x(t) = v$. Next we want the show that if $\lim_{t\to\infty} e(t) = \mathbf{0}$ for all $x(0)$ then $\rho(M - v\mathbf{1}^T) < 1$. Using (11), we have

$$\begin{aligned} e(t) &= (M - v\mathbf{1}^T)^t e(0) = (M - v\mathbf{1}^T)^t (x(0) - v) \\ &= (M - v\mathbf{1}^T)^t x(0) - (M^t - v\mathbf{1}^T)v \\ &= (M - v\mathbf{1}^T)^t x(0) - v + v = (M - v\mathbf{1}^T)^t x(0). \end{aligned}$$

The fact that $\lim_{t\to\infty} e(t) = \mathbf{0}$ for any $x(0)$ implies that

$$\lim_{t\to\infty} e(t) = \lim_{t\to\infty} (M - v\mathbf{1}^T)^t x(0) = 0 \qquad (13)$$

The vector $x(0)$ in (13) is constrained as $x(0) \geq \mathbf{0}$ and $\mathbf{1}^T x(0) = 1$, and can be chosen, for example, as any of the basis vectors $\{e_1, ..., e_m\}$. The fact that this basis set spans $I\!R^m$ ensures that,

$$\lim_{t\to\infty}(M - v\mathbf{1}^T)^t = \mathbf{0} \tag{14}$$

which implies that $\rho(M - v\mathbf{1}^T) < 1$. ■

D. Motion Constraints

In practice, it is convenient to impose additional constraints on matrix M to restrict allowable agent motion. For example, it may not be desirable or even physically possible for an agent in bin j to move to some other bin i in a single time step. This transition is mathematically disallowed by setting the associated element of M to zero, i.e., $M[i, j] = 0$. More generally, connectivity constraints are imposed on the adjacency matrix for a graph associated with M. A directed graph $\mathbf{G}_a = (\mathbf{V}_a, \mathbf{E}_a)$ is defined where $\mathbf{V}_a$ is a set of m vertices chosen to correspond to the m bins of $\mathcal{R}$, and $\mathbf{E}_a$ are the edges of the graph defined such that the edge (i, j) exists if and only if there is an allowable transition from bin i to bin j. The graph is directed in the sense that edge (i, j) which denotes an allowable transition from i to j, is distinguished from edge (j, i) which denotes the transition back from j to i. Let A_a be the corresponding adjacency matrix for this graph, that is, $A_a[i, j] = 1$ if the transition from bin i to bin j is allowable, and is zero otherwise. The motion constraints are imposed on M as follows,

$$(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}. \tag{15}$$

The transpose of matrix A_a is needed because, by definition, elements $A_a[i, j]$ of A_a describe transitions from i to j while elements $M[i, j]$ of M describe transitions from j to i.

E. Decentralization and Self-Repair Properties

Under the conditions on policy matrix M indicated in Theorem 2, the swarm will converge asymptotically to a statistical equilibrium condition achieving the desired distribution v, starting from any arbitrary initial distribution $x(0)$. This has two important implications:

(i) The probabilistic guidance law is completely decentralized and converges the swarm to any desired distribution starting from any initial distribution

(ii) The probabilistic guidance law provides an autonomous capability for decentralized swarm repair

Implication (i) is a direct result of Theorem 2 and the fact that agents in the swarm do not require any knowledge from, or communication with, other agents of the swarm. Implication (ii) is an important property that follows from the first property but requires some explanation. Suppose the swarm achieves a desired distribution and is in a statistical steady-state condition. If there is any damage to the swarm, in the sense of agents becoming destroyed, non-functional or dropping out, the statistical steady-state distribution is perturbed. Under the probabilistic guidance law, the swarm will automatically re-converge from its perturbed state to its desired steady-state distribution, thereby repairing the damage. It is emphasized that knowledge is not needed of the damage or of the specific agents involved. The guidance law simply continues as if nothing has happened and the swarm is repaired. The self-repair response is completely autonomous and decentralized.

IV. Connection with Perron-Frobenius Theory

The notion of a *primitive* matrix (see [1], [15]) is of central importance in Perron-Frobenius theory of positive matrices. The next result establishes a connection with Perron-Frobenius theory by showing the relationship between the spectral radius condition (10) and primitive matrices.

Lemma 1: Consider a matrix $M \in I\!R^{m\times m}$ such that $M \geq \mathbf{0}$, $\mathbf{1}^T M = \mathbf{1}^T$, and $Mv = v$ for some $v > \mathbf{0}$. Then: $\rho(M - v\mathbf{1}^T) < 1$ if and only if M is a primitive matrix.

Proof: Suppose that M is primitive. By using Theorem Thm. 8.5.1 in [15]

$$\lim_{t\to\infty} M^t = v\mathbf{1}^T.$$

Now using Equation (11)

$$\lim_{t\to\infty}(M - v\mathbf{1}^T)^t = \lim_{t\to\infty} M^t - v\mathbf{1}^T = v\mathbf{1}^T - v\mathbf{1}^T = \mathbf{0},$$

which implies that $\rho(M - v\mathbf{1}^T) < 1$.

Next suppose that $\rho(M - v\mathbf{1}^T) < 1$. This implies that

$$\lim_{t\to\infty}(M - v\mathbf{1}^T)^t = \mathbf{0} \;\Rightarrow$$
$$\lim_{t\to\infty} \|(M - v\mathbf{1}^T)^t\|_\infty = \lim_{t\to\infty} \|M^t - v\mathbf{1}^T\|_\infty = 0.$$

Consequently, for $\epsilon := \frac{1}{2}\min_i v[i] > 0$, there exists some $q \in I\!N_+$ such that, for all $k \geq q$,

$$\left|M^k[i, j] - v[i]\right| < \epsilon \qquad \forall i, j = 1, \ldots, m.$$

This implies that $M^k[i, j] > v[i] - \epsilon > \epsilon/2 > 0$ for all $k \geq q$ and $i, j = 1, \ldots, m$. Hence $M^q > \mathbf{0}$, which implies that M is primitive by using Theorem 8.5.2 in [15]. ■

The primitivity condition $M^k > 0$ of Theorem 8.5.2 in [15] ensures asymptotic convergence of $x(t)$ to $v > 0$ in Markov Chains for which M is not strictly positive, i.e., $M \geq 0$ [15]. However, the results of Lemma 1 and Theorem 2 taken together, indicate that the primitivity condition can be replaced by the new spectral radius condition $\rho(M - v\mathbf{1}^T) < 1$ (cf., (10)), for proving asymptotic convergence to $v > 0$. It must be observed that the condition that $v > 0$, which is central to Perron-Frobenius theory, does not appear in the asymptotic convergence result of Theorem 2. The relaxation of condition $v > \mathbf{0}$ to $v \geq \mathbf{0}$ allows specific elements of v to have zero probability. In this case, M will be reducible with the corresponding zeroed states forming a subset of transient states, and the non-zeroed states forming a subset of recurrent states.

Theorem 3: Consider a matrix $M \in I\!R^{m\times m}$ such that $M \geq \mathbf{0}$, $\mathbf{1}^T M = \mathbf{1}^T$, and $Mv = v$ for some $v \geq \mathbf{0}$ where, without loss of generality, $v = (\mathbf{0}, \hat{v})$ and $\hat{v} \in I\!R^q$, $\hat{v} > \mathbf{0}$. Then it follows that $\rho(M - v\mathbf{1}^T) < 1$ if and only if M has the following structure

$$M = \begin{bmatrix} M_1 & \mathbf{0} \\ M_2 & M_3 \end{bmatrix} \qquad \text{where} \tag{16}$$

$M_1 \in I\!R^{(m-q)\times(m-q)}$ and $M_3 \in I\!R^{q\times q}$ are nonnegative matrices such that M_3 is primitive and $\rho(M_1) < 1$.

Proof: Since $v = (\mathbf{0}, \hat{v})$, $Mv = v$ implies that

$$\underbrace{\left[\begin{array}{cc} M_1 & M_4 \\ M_2 & M_3 \end{array}\right]}_{=M} \left[\begin{array}{c} \mathbf{0} \\ \hat{v} \end{array}\right] = \left[\begin{array}{c} \mathbf{0} \\ \hat{v} \end{array}\right] \;\Rightarrow\; M_4\hat{v} = \mathbf{0}.$$

Since $M_4 \geq \mathbf{0}$ and $v > \mathbf{0}$, we have $M_4 = \mathbf{0}$. This proves the form of M given by (16).

Note that

$$M - v\mathbf{1}^T = \left[\begin{array}{cc} M_1 & \mathbf{0} \\ M_2 - \hat{v}\mathbf{1}^T & M_3 - \hat{v}\mathbf{1}^T \end{array}\right],$$

which is block lower triangular. This implies that $\rho(M - v\mathbf{1}^T) < 1$ if and only if $\rho(M_1) < 1$ and $\rho(M_3 - \hat{v}\mathbf{1}^T) < 1$. By Lemma 1, $\rho(M_3 - \hat{v}\mathbf{1}^T) < 1$ is equivalent to M_3 being primitive. ■

Remark 1: For any matrix $X \in I\!R^{m\times m}$ such that $X \geq \mathbf{0}$, the condition $\rho(X) < 1$ is known to be equivalent to the condition that $I - X$ is an *M-matrix* (cf., Theorem 2.5.3 [16]). M-matrices arise naturally in many diverse fields such as large scale systems, networks, and interconnected systems. In the context of Theorem 3, the condition that $\rho(M_1) < 1$ can be equivalently interpreted as $I - M_1$ being an M-matrix. Now we can state a direct corollary to Theorem 2.

Corollary 1: Suppose that the PGA is used for swarm guidance with a column stochastic matrix M such that $Mv = v$ where $v = (\mathbf{0}, \hat{v})$ with $\hat{v} > \mathbf{0}$ and $x(t+1) = Mx(t)$ with $x(t)$ defined by (1). Then for any initial swarm distribution $x(0) \in I\!R^m$, it follows that $\lim_{t\to\infty} x(t) = v$ if and only if M has the structure described by (16), where M_3 is a primitive matrix and $\rho(M_1) < 1$.

The ability to specify $v \geq \mathbf{0}$ rather than $v > \mathbf{0}$ is important in guidance problems where agents are constrained to be located outside of certain specified regions.

Corollary 2: Suppose that the PGA is used for swarm guidance with a column stochastic matrix M such that $Mv = v$ where $v = (\mathbf{0}, \hat{v})$ with $\hat{v} > \mathbf{0}$ and $x(t+1) = Mx(t)$ with $x(t)$ defined by (1). Then for any initial swarm distribution $x(0) \in I\!R^m$: $\lim_{t\to\infty} x(t) = v$ only if:

(i) $\mathtt{i}(M_3)$ is an adjacency matrix of a strongly connected graph,

(ii) $M_2 \neq \mathbf{0}$,

where M is of the form described by (16). Furthermore, if condition (i) is satisfied and $\mathrm{trace}(M_3) > 0$, then M_3 is primitive.

Proof: By using Corollary 1, M_3 being primitive is a necessary condition for convergence to v. Since this implies that M_3 is irreducible, hence $\mathtt{i}(M_3)$ must represent a strongly connected graph, which follows from [1], Thm.1.3 and 2.1. Asymptotic convergence to v also necessitates that $\rho(M_1) < 1$, which is from Corollary 1. However $M_2 = \mathbf{0}$ implies that M_1 is a column stochastic matrix with an eigenvalue of 1, which is a contradiction. This proves (ii). Now suppose that $\mathtt{i}(M_3)$ is an adjacency matrix of a strongly connected graph. If $\mathrm{trace}(M_3) > 0$, which implies that M_3 is irreducible, M_3 is primitive by using Lemma 2.28 in p.34 of [1]. ■

The above corollary implies that the adjacency matrix A_a must satisfy the following. Consider the partition of v as done in Corollary 1, $v = (\mathbf{0}, \hat{v})$. Note that this partitioning can always be done via renumbering the bins, hence there is no loss of generality. Then suppose A_a is decomposed as:

$$A_a = \left[\begin{array}{cc} A_1 & A_4 \\ A_2 & A_3 \end{array}\right], \quad \text{where} \quad A_3 \in I\!R^{q\times q}. \qquad (17)$$

Then A_3 must be the adjacency graph of a strongly connected graph and $A_2 \neq \mathbf{0}$ in order to have a convergent M.

V. Metropolis-Hastings Algorithm for Synthesis of M

In this section we present several approaches to constructing M matrices that satisfy the desired conditions for probabilistic guidance.

Definition 1: Given a vector v such that $v \geq \mathbf{0}$ and $\mathbf{1}^T v = 1$, and a specified adjacency matrix A_a, the set of *admissible* matrices $\mathcal{M}$ is defined as,

$$\begin{aligned} &\mathcal{M}(v, A_a) := \{M \in I\!R^{m\times m} : M \geq \mathbf{0}, \\ &\mathbf{1}^T M = \mathbf{1}^T,\ Mv = v,\ (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}\}. \end{aligned} \qquad (18)$$

Note that the set $\mathcal{M}(v, A_a)$ is completely characterized via linear constraints on the matrix M. Methods will be presented for synthesizing matrices M that are in the admissible set $\mathcal{M}(v, A_a)$ in (18), and that satisfy the spectral radius condition (10) needed for asymptotic convergence.

A. Overview of the Metropolis-Hastings Algorithm

The Metropolis-Hastings (M-H) algorithm [28], [3] is a Markov Chain Monte Carlo (MCMC) method for obtaining a sequence of random samples defined by propagating a special Markov chain. The Markov chain is defined by starting with an arbitrary stochastic matrix K, denoted as a *proposal matrix*, and transforming it into a Markov matrix M having a specified stationary distribution $v > 0$, by making use of an intermediary matrix F, denoted as an *acceptance matrix*.

Definition 2: [Metropolis-Hastings Algorithm] The M-H algorithm is defined by the matrix M constructed as follows,

$$M[i,j] = \left\{ \begin{array}{cl} K[i,j]F[i,j] & \text{if} \quad i \neq j \\ K[j,j] + \displaystyle\sum_{k\neq j}(1 - F[k,j])K[k,j] & \text{if} \quad i = j \end{array} \right. \qquad (19)$$

where the matrix $K[i,j]$ satisfies $K \geq 0$ and $\mathbf{1}^T K = \mathbf{1}^T$; the desired stationary distribution satisfies $v > 0$; and the matrix $F[i,j]$ satisfies the following two conditions for $i \neq j$,

$$\begin{aligned} 0 &\leq F[i,j] \leq \min(1, R[i,j]) && (20) \\ F[j,i] &= \frac{1}{R[i,j]} F[i,j] && (21) \end{aligned}$$

$$\text{where } R[i,j] = \frac{v[i]K[j,i]}{v[j]K[i,j]} \quad i,j = 1,\ldots,m. \qquad (22)$$

Note that swapping i and j in condition (20) yields,

$$0 \leq F[j,i] \leq \min(1, R[j,i]) \qquad (23)$$

By construction, condition (20) is satisfied if and only if condition (23) is satisfied. Hence, the function $F[j,i]$ derived by swapping indices in $F[i,j]$ must simultaneously satisfy the two conditions (21) and (23).

Definition 3: The reversibility condition for a Markov matrix M with respect to a probability vector v is,

$$M[i,j]v[j] = M[j,i]v[i], \quad i=1,...,m; \quad j=1,...,m. \quad (24)$$

Note that the reversibility condition (24) only needs to be checked for $i \neq j$ since it holds trivially for $i = j$.

Lemma 2: Let the reversibility condition (24) hold for a Markov matrix M with respect to a probability vector v. Then v is a stationary solution to the Markov chain, i.e., $Mv = v$.

Proof: Let $w = Mv$. Then, $w[i] = \sum_{j=1}^{m} M[i,j]v[j] =$

$$\underbrace{\left(\sum_{j=1}^{m} M[j,i]\right)}_{1} v[i] = v[i].$$ ■

Lemma 3: The matrix M defined by the M-H algorithm (19)-(22) is a Markov matrix with v as a stationary solution, i.e., $Mv = v$.

Proof: (Proof of $M \geq 0, \mathbf{1}^T M = \mathbf{1}^T$). Clearly since $K \geq \mathbf{0}$ and $v > 0$, it follows that $M \geq \mathbf{0}$. Now consider $\sum_{i=1}^{m} M[i,j]$ for any j, $\sum_{i=1}^{m} M[i,j]$

$$\begin{aligned}
&= \sum_{i\neq j} K[i,j]F[i,j] + K[j,j] + \sum_{k\neq j}(1 - F[k,j])K[k,j] \\
&= \sum_{i\neq j} K[i,j]F[i,j] + K[j,j] + \sum_{k\neq j} K[k,j] - \sum_{k\neq j} F[k,j]K[k,j] \\
&= K[j,j] + \sum_{k\neq j} K[k,j] = 1,
\end{aligned}$$

which follows from $\mathbf{1}^T K = \mathbf{1}^T$. This shows that $\mathbf{1}^T M = \mathbf{1}^T$.

(Proof of $Mv = v$). For the M-H algorithm, $M[i,j] = K[i,j]F[i,j]$ when $i \neq j$ with $F[j,i] = F[i,j]/R[i,j]$. This implies that, for $i \neq j$,

$$\begin{aligned}
&M[i,j]v[j] \\
&= K[i,j]F[i,j]v[j] = K[i,j]R[i,j]F[j,i]v[j] \\
&= K[i,j]\frac{v[i]K[j,i]}{v[j]K[i,j]}F[j,i]v[j] = K[j,i]F[j,i]v[i] \\
&= M[j,i]v[i].
\end{aligned}$$

Hence, the reversibility condition (24) holds for Markov matrix M with respect to a probability vector v. By Lemma 2, it follows that $Mv = v$. ■

B. Alpha-Min Acceptance Matrix

One particular choice for the acceptance matrix F is of the *alpha-min* form,

$$F[i,j] = \alpha \min(1, R[i,j]) \quad \text{where} \quad \alpha \in (0,1]. \qquad (25)$$

In this case, we can similarly choose $F[j,i] = \alpha \min(1, R[j,i])$ while satisfying conditions (20), (23), and (21). To see that, we only need to consider the case when $R[i,j] \geq 1$ (switching indices will imply the case when $R[i,j] < 1$). Then $F[i,j] = \alpha$ and $F[j,i] = \alpha R[j,i]$, since $R[j,i] = 1/R[i,j] < 1$. This implies that $F[j,i] = F[i,j]R[j,i] = F[i,j]/R[i,j]$ Note that if $K[i,j] = 0$ then $M[i,j] = M[j,i] = 0$. Similarly having $K[i,j] > 0$ and $K[j,i] > 0$ imply that $M[i,j] > 0$ and $M[j,i] > 0$. Consequently, when $v > 0$, $A_a = A_a^T$ and matrix F is chosen as given in (25), we can impose the desired motion constraints on the proposal matrix K in order to guarantee their satisfaction by the matrix M and $\mathtt{i}(M) = A_a^T$. Based on these insights, the M-H algorithm is applied to the probabilistic guidance problem.

C. M-H for Probabilistic Guidance

Theorem 4: Consider the M-H algorithm, for some $v > 0$, given by (19) where the proposal matrix satisfy $\mathtt{i}(K) = A_a^T$ for some strongly connected symmetric adjacency matrix $A_a = A_a^T$, and matrix F is constructed using alpha-min form (25). Then for the resulting matrix M, it follows that $M \in \mathcal{M}(v, A_a)$ and $\rho(M - v\mathbf{1}^T) < 1$.

Proof: The previous discussion of the alpha-min acceptance matrix established that equation (25) results in a matrix F satisfying conditions (20), (23), and (21). It also shows that $\mathbf{1}^T M = \mathbf{1}^T$ and $Mv = v$. As observed earlier, $A_a[j,i] = K[i,j] = 0$ implies that $M[i,j] = 0$, and $A[j,i] = \text{sign}(K[i,j]) = 1$ implies that $M[i,j] > 0$, that is, $\mathtt{i}(M) = A_a^T$. Hence $(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}$, and therefore $M \in \mathcal{M}(v, A_a)$. Next since A_a is strongly connected, there exists some k such that $(A_a^k)^T = \mathtt{i}(M)^k > \mathbf{0}$, which implies that M is primitive via Thm. 8.5.1 in [15]. Consequently $\rho(M - v\mathbf{1}^T) < 1$ by using Theorem 3. ■

D. M-H with Gaussian Proposal Matrix

Next we will describe an algorithm that uses the alpha-min acceptance matrix (25), and constructs a proposal matrix K by making use of a Gaussian distribution. Given A_a be as above, the connectivity matrix for the graph with all possible links between the bins that satisfies $A_a^{m-1} > 0$ and a Gaussian distribution with a mean μ and a standard deviation $\delta > 0$ given by $g : IR \to IR$

$$g(z; \eta, \delta) = \frac{1}{\delta\sqrt{2\pi}} e^{-(x-\mu)^2/2\delta^2}. \qquad (26)$$

Let c_i, $i = 1, \ldots, m$, be the vectors describing the center of each bin. Then we form K as follows: Given $\delta_i > 0$ $i = 1, \ldots, m$ and $j = 1, ..., m$

$$K[j,i] = \begin{cases} g(\|c_j - c_i\|; 0, \delta_i) & \text{if} \quad A_a[i,j] = 1,\ i \neq j \\ 0 & \text{if} \quad A_a[i,j] = 0 \\ 1 - \sum_{k\neq i} K[k,i] & \text{if} \quad j = i \end{cases} \qquad (27)$$

The proposal matrix K is chosen such that $\mathtt{i}(K)$ is an adjacency matrix of a connected graph, that is, $\mathtt{i}(K) = A_a^T$, which follows from the fact that $K[i,j] > 0$ for $A_a[j,i] = 1$. Also note that, when $A_a = A_a^T$, we have $K[i,j] = K[j,i]$, i.e., $K = K^T$, which implies that $R[i,j]$ is independent of K, i.e., $R[i,j] = \frac{v[i]}{v[j]}$.

E. Metropolis-Hastings Algorithm with Transient States

Use of the Metropolis-Hastings (M-H) algorithm to construct the matrix M in (19) is not well defined in the special case where the desired distribution v has transient states, that is, $v = (0, \hat{v})$ where $\hat{v} > 0$. A modification of the M-H algorithm is introduced to deal with this important special case. Suppose that there are m_t transient states in the desired distribution and $m_r = m - m_t$ recurrent states. Further, suppose that the bins numbered from $m_t - m_1 + 1$ to m_t (for some m_1), are all bins corresponding to transient states that are directly connected to the bins corresponding to the recurrent states, which are numbered from $m_t + 1$ to m, as indicated by matrix A_a. Then let the bins numbered from $m_t - m_2 + 1$ to $m_t - m_1$ (for some m_2), be all the bins corresponding to the transient states that are directly connected to the bins numbered from $m_t - m_1 + 1$ to m_t, and so on. If A_a corresponds to a connected undirected graph, then we continue this process to group all the bins corresponding the transient states. This construction leads to a desired density vector v of the following form

$$v = (v_1, \dots, v_s, \hat{v}) \text{ where } v_1 = 0, ..., v_s = 0,\ \hat{v} > 0, \tag{28}$$

and the bins corresponding to v_1 are directly connected to the bins of v_2, the bins of v_2 are directly connected to the bins of v_3, ad so on. We also define index sets $\mathcal{I}_r = \{m_t + 1, \dots, m\}$, $\mathcal{I}_s = \{m_t - m_1 + 1, \dots, m_t\}$,$\mathcal{I}_{s-1} = \{m_t - m_2 + 1, \dots, m_t - m_1\}$ …. Hence $\mathcal{I}_r$ is the index set of recurrent states and $\mathcal{I}_s$ is the index set of transient states directly connected to recurrent states, and so on.

Theorem 3 implies that the matrix resulting from the modified M-H algorithm must generate a matrix M of the form given by (16), where M_3 is a primitive matrix and $\rho(M_1) < 1$. The following modified M-H algorithms achieves this objective:

$$M[i,j] = \begin{cases} \text{as in (19)} & \text{if} \quad i \in \mathcal{I}_r,\ j \in \mathcal{I}_r \\ 0 & \text{if} \quad i \leq m_t,\ j \in \mathcal{I}_r \\ 1/(\sum_{k \in \mathcal{I}_r} A_a[j,k]) & \text{if} \quad i \in \mathcal{I}_r, j \leq m_t, A_a[j,i] = 1 \\ 1/(\sum_{k \in \mathcal{I}_{k+1}} A_a[j,k]) & \text{if} \quad i \in \mathcal{I}_{k+1}, j \in \mathcal{I}_k, A_a[j,i] = 1 \\ 0 & \quad\quad \text{elsewhere} \end{cases} \tag{29}$$

The modified construction of matrix M results in the following form

$$M = \left[\begin{array}{c|c} \begin{bmatrix} 0 & 0 & \dots & 0 \\ M_{1,2} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & M_{1,s} & 0 \end{bmatrix} & 0 \\ \hline \begin{bmatrix} 0 & \dots & 0 & M_{2,1} \\ 0 & \dots & \dots & 0 \end{bmatrix} & M_3 \end{array}\right] \tag{30}$$

where $M_{1,k}$, $k = 1, ..., s$, and $M_{2,1}$ are columns stochastic matrices. Assuming that A_a represents a connected graph, M_3 is primitive by the above construction. Furthermore, M is clearly a column stochastic matrix such that $Mv = v$. The question of whether $\rho(M_a) < 1$ is addressed next. Consider a vector $\eta > 0$ as follows (vector $\mathbf{1}$ in each entry has the size corresponding to that of the decomposed v)

$$\eta = (\alpha_2 \mathbf{1}, \dots, \alpha_s \mathbf{1}, \mathbf{1}) \quad \text{where}$$

$$\alpha_s = \frac{1}{|||M_{1,s}^T|||_1 + 1}$$

$$\alpha_k = \frac{\alpha_{k+1}}{|||M_{1,k}^T|||_1 + 1}, \quad k = 2, \dots, s-1.$$

Then $\eta > 0$ and

$$(I - M_1)\eta = \begin{pmatrix} \alpha_2 \mathbf{1}, & \alpha_3 \mathbf{1} - \alpha_2 M_{1,2} \mathbf{1}, & \dots & \mathbf{1} - \alpha_s M_{1,s} \mathbf{1} \end{pmatrix} > 0,$$

which shows, by using Theorem 2.5.3 (parts 2.5.3.2 and 2.5.3.12) in [16], that $\rho(M_1) < 1$. This concludes the proof of the fact that M provides an asymptotically convergent guidance policy. The following theorem, which is our main result, follows from the above discussion.

Theorem 5: Consider the M-H algorithm, for some $v \geq 0$, given by (29) where the proposal matrix satisfies $\mathtt{i}(K) = A_a^T$ for some strongly connected symmetric adjacency matrix $A_a = A_a^T$, and matrix F is constructed using equation (25) between the bins for the recurrent states. Then the resulting matrix satisfies $M \in \mathcal{M}(v, A_a)$, and $\rho(M - v\mathbf{1}^T) < 1$.

VI. Numerical Example

This example demonstrates decentralized swarm guidance using PGA. The swarm contains $N = 5000$ autonomous agents that are guided to form the sequence of probability distributions (v_A, v_E), where v_A is a distribution associated with the letter "A" and v_E is associated with the letter "E". Convergence is monitored using the *total variation*,

$$T(t) = \sum_{j=1}^{N} \Big| x(t)[j] - v_d[j] \Big| \tag{31}$$

The scenario starts at $t = 0$ with the swarm uniformly distributed across $\mathcal{R}$ (cf., Fig. 2). PGA turns on and begins with each agent flying an independent realization of a Markov chain having stationary distribution v_A. Here, PGA is implemented using the M-H algorithm. The total variation in Fig. 1 indicates convergence is achieved at roughly $t = 250$ steps, with the resulting distribution for the letter A shown in Fig. 2. During this period (cf., $t = 40$), an onlooker would witness "emergent swarm behavior" in the sense of seeing the letter A emerge from the starting uniform distribution of agents. At time $t = 251$, PGA begins each agent flying a new Markov chain having stationary distribution v_E. Fig. 1 indicates convergence is achieved at time $t = 500$, with the resulting distribution for the letter E shown in Fig. 2. During this period, an onlooker would witness the swarm morphing from the letter A into the letter E. At time $t = 501$, damage is inflicted on the swarm, wiping out most agents in the middle arm of the letter E. As time progresses, the agents repair this damage without collaborating or even knowing about the existence of the damage. Damage is repaired and the swarm is re-converged back to the letter E by time $t = 950$.

VII. Conclusions

A new probabilistic method is introduced for performing swarm guidance that guides the shape of the swarm to conform to a prescribed probability distribution. The probabilistic guidance approach is completely decentralized in the

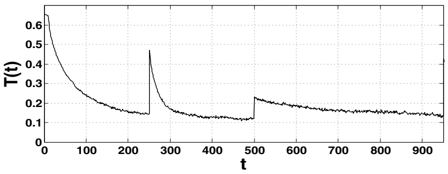

Fig. 1. Convergence for a 5000-agent swarm as a function of time in terms of the total variation $T(t)$.

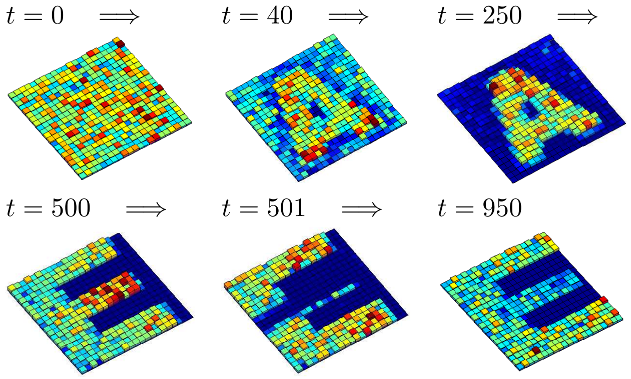

Fig. 2. Histogram of agents at different time instances: Guided by PGA, a swarm of $5,000$ agents evolves through a sequence of desired distributions, and autonomously recovers from inflicted damage.

sense that there is no communication between agents, yet the swarm asymptotocally achieves its desired distribution. In addition, the swarm has a novel self-repair property that fixes damaged portions of its distribution, and handles transient states to allow strict enforcement of desired keep-out (i.e., zero-probability) regions. A simulation of the guided swarm demonstrates the properties of convergence and self-repair.

ACKNOWLEDGMENTS

The authors gratefully acknowledge Fred Hadaegh, Lars Blackmore, Milan Mandic of Jet Propulsion Laboratory and Soon-Jo Chung of Univ. of Illinois for their comments. This research was funded by the Defense Advanced Research Projects Agency, and performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

REFERENCES

[1] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, 1994.

[2] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar. Optimized stochastic policies for task allocation in swarms of robots. *IEEE Transactions on Robotics*, 25(4):927–937, 2009.

[3] L. J. Billera and P. Diaconis. A geometric interpretation of the Metropolis-Hastings algorithm. *Statistical Science*, 16(4):335–339, 2001.

[4] S. Boyd, P. Diaconis, P. Parillo, and L. Xiao. Fastest mixing Markov Chain on graphs with symmetries. *SIAM Journal of Optimization*, 20(2):792–819, 2009.

[5] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov Chain on a graph. *SIAM Review*, 46:667–689, 2004.

[6] I. Chattopadhyay and A. Ray. Supervised self-organization of homogeneous swarms using ergodic projections of markov chains. *IEEE Transactions on Systems, Man., and Cybernetics*, 39(6):1505–1515, 2009.

[7] J. Cortés, S. Martínez, and F. Bullo. Robust rendezvous for mobile autonomous agents via proximity graphs in arbitrary dimensions. *IEEE Transactions on Automatic Control*, 51(8):1289–1298, 2006.

[8] N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, 1974.

[9] P. Diaconis. The Markov chain Monte Carlo revolution. *Bulletin of the American Mathematical Society*, 46:179–205, 2009.

[10] P. Diaconis and L. Saloff-Coste. What do we know about the Metropolis algorithm? *Journal of Computer and System Sciences*, 57:20–36, 1998.

[11] P. Diaconis and D. Stroock. Geometric bounds for eigenvalues of Markov Chains. *The Annals of Applied Probability*, 1(1):36–61, 1991.

[12] M. Fiedler. *Special Matrices and Their Applications in Numerical Mathematics*. Dover, 2008.

[13] G. Fishman. *Monte Carlo, Concepts, algorithms and applications*. Springer, New York, 1996.

[14] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

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

[16] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[17] A. Jadbabaie, G. J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[18] R. E. Kalman and J. E. Bertram. Control system analysis and design via the second method of Lyapunov, ii: Discrete-time systems. *Journal of Basic Engineering*, 82:394–400, 1960.

[19] Y. Kim, M. Mesbahi, and F. Hadaegh. Multiple-spacecraft reconfiguration through collision avoidance, bouncing, and stalemate. *Journal of Optimization Theory and Applications*, 122(2):323–343, 2004.

[20] E. Klavins, R. Ghrist, and D. Lipsky. A grammatical approach to self organizing robotic systems. *IEEE Transactions on Automatic Control*, 51(6):949–962, 2006.

[21] M. Lawford and W. Wonham. Supervisory control of probabilistic discreteevent systems. *Proc. 36th Midwest Symp. Circuits Syst.*, pages 327–331, 1993.

[22] J. S. Liu. Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Statistics and Computing*, 6:113–119, 1996.

[23] J. S. Liu. *Monte Carlo strategies in scientific computing*. Springer-Verlag, 2001.

[24] V. J. Lumelsky and K. R. Harinarayan. Decentralized motion planning for multiple mobile robots: The cocktail party model. *Autonomous Robots*, 4(1):121–135, 1997.

[25] A. M. Lyapunov. Probleme general de la stabilite du movement (in french). *Ann. Fac. Sci. Toulousse. Reprinted in Ann. Math. Study No. 17, 1949, Princeton University Press*, pages 203–474, 1907.

[26] S. Martínez, F. Bullo, J. Cortés, and E. Frazzoli. On synchronous robotic networks-part ii: Time complexity of rendezvous and deployment algorithms. *IEEE Transactions on Automatic Control*, 52(12):2214–2226, 2007.

[27] S. Martínez, J. Cortés, and F. Bullo. Motion coordination with distributed information. *IEEE Control System Magazine*, pages 75–88, 2007.

[28] N. Metropolis and S. Ulam. The Monte-Carlo method. *American Statistical Association*, 44:335–341, 1949.

[29] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):410–420, 2006.

[30] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.

[31] A. Richards, T. Schouwenaars, J. How, , and E. Feron. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *AIAA Journal of Guidance, Control, and Dynamics*, 25(4):755–764, 2002.

[32] D. P. Scharf, F. Hadaegh, and S. R. Ploen. A survey of spacecraft formation flying guidance and control (part i): guidance. *In Proceedings of American Control Conference*, 2003.

[33] M. Tillerson, G. Inalhan, and J. How. Co-ordination and control of distributed spacecraft systems using convex optimization techniques. *International Journal of Robust Nonlinear Control*, 12(1):207–242, 2002.