2014 American Control Conference (ACC)
June 4-6, 2014. Portland, Oregon, USA

Probabilistic Swarm Guidance for Collaborative Autonomous Agents

Behçet Açıkmeşe and David S. Bayard

***Abstract*— This paper extends a probabilistic guidance approach for the coordination of swarms of autonomous agents, which contain sub-swarms with different mission objectives. The main idea of probabilistic guidance is to drive the swarm to a prescribed density distribution in the configuration space. Both non-collaborative and collaborative swarm guidance methods are considered. In the non-collaborative case, the probabilistic approach is decentralized and does not require communication between agents. Agents make statistically independent probabilistic decisions based solely on their own state, which ultimately guides the swarm to the desired density distribution. The probabilistic guidance idea is then extended to the collaborative case where there are sub-swarms of agents with different desired distributions. In this case, agents collaborate to decide on a common objective. This introduces collaboration at a higher level of decision making, and makes useful connections with consensus theory for networked agents. Our main result establishes that, under quite general conditions, the collaborative swarm behavior converges to the weighted average of the sub-swarm desired distributions. The formal development of this result now allows us to consider complex swarm behaviors in a mathematically rigorous framework.**

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. Both non-collaborative and collaborative swarm guidance methods are considered. The configuration variable can be a physical quantity such as the agent position, speed, temperature. It is chosen here to be the agent position without loss of generality. The main feature of our guidance method is its probabilistic nature, where the desired spatial probability density distribution for the swarm is specified rather than the exact desired positions of the individual agents. This represents a break with existing guidance methods for distributed systems where agent positions are assigned ahead of time, and there is one slot for each agent [1], [2], [3], [4], [5], [6].

The probabilistic guidance approach is based on designing a Markov chain such that its steady-state distribution corresponds to the desired swarm density. The ideas in this paper extend results that first appeared in [7]. A copy of the designed Markov chain transition probabilities is given to each agent. Each agent is assumed to have knowledge of its own position and the mobility to move to the location indicated by the Markov chain. As each agent propagates his position, the ensemble of statistically independent realizations converge to the desired asymptotic distribution. To an on-looker, this convergence appears as an ”emergent behavior” of the swarm. While many different methods for swarm guidance and control have appeared in the literature [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], the basic idea of using independent realizations of a Markov chain for guiding large numbers of agents to a desired spatial distribution is relatively new [7]. A different Markov chain based method appeared in [18], using a probabilistic “disablement” approach found in [19]. Aside from the similar idea of using Markov chains, the current paper does not use disablement in [19], and main results are not directly comparable. A different but related approach appeared in the literature in [20], which used partial differential equations to describe the evolution of the probability density function and to design convergent swarms. The non-collaborative swarm guidance method [7] is generalized to become a collaborative method by making connections with recent developments in consensus theory for networked agents [21], [22], [23], [24], [25]. Specifically we proved that establishing consensus on mission goals as each agent reacts to its current mission goal can produce a convergent behavior under assumption that the communication graph is connected. This extension gives the ability to form useful swarm behaviors by collaborating between sub-swarms having different desired distribution specifications. For example the objectives of an existing swarm can be altered by adding agents with alternative objectives such that the resulting consensus leads the whole swarm to converge to a different desired distribution. Hence swarm behavior can be modified by inserting new agents having different desired distribution specifications.

Partial List of Notation: $\mathbf{0}$ and $\mathbf{1}$ are the matrices of zeros and ones; e_i is a vector with its ith entry $+1$ and others zero; I is the identity matrix; $x[i]=\mathrm{e}_i^T x$ and $A[i,j]=\mathrm{e}_i^T A\mathrm{e}_j$; $Q=Q^T\succ(\succeq)0$ is a symmetric positive (semi-)definite matrix; $R>(\geq)H$ implies that $R[i,j]>(\geq)H[i,j]$ for all i,j; v is a *probability vector* if $v\geq\mathbf{0}$ and $\mathbf{1}^T v=1$; $\mathcal{P}$ denotes the probability of a random variable; $\|v\|$ is the 2-norm of v; For $P=P^T\succ 0$, $\|v\|_P=\|P^{1/2}v\|$; $(v_1,\ldots,v_n)$ is an augmentation of vectors as $[v_1^T\ldots v_n^T]^T$; $\lambda_{max}(P)$ and $\lambda_{min}(P)$ are max and min eigenvalues of $P=P^T$; $\sigma(A)$ is the spectrum and $\rho(A)$ is the spectral radius of A; $\odot$ denotes the Hadamard product; $\mathtt{i}(A)$ is the indicator matrix given by $\mathtt{i}(A)[i,j]=1$ if $A[i,j]\neq 0$ and $\mathtt{i}(A)[i,j]=0$ otherwise.

II. SWARM DISTRIBUTION GUIDANCE PROBLEM

This section summarizes the description of the swarm distribution guidance problem given in [7]. The physical

Aerospace Engineering and Engineering Mechanics, The University of Texas at Austin, TX 78712, USA. behcet@austin.utexas.edu

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109, USA. david.s.bayard@jpl.nasa.gov

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, $i = 1, \ldots, m$, (referred to as *bins*) such that $\mathcal{R} = \bigcup_{i=1}^{m} R_i$, and $R_i \cap R_j = \emptyset$ for $i \neq j$. Let $r(t)$ be the position of an agent at time index $t \in I\!N_+$. Let $x(t)$ be a probability vector such that $x[i](t)$ is the probability of the event that this agent will be in bin R_i at time t,

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

Consider a swarm comprised of N independently acting agents, such 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 t, and the probabilities of these N events are jointly statistically independent. Here $x(t)$ is referred as the *swarm distribution*.

The distribution guidance problem is defined as: Given any initial probability distribution $x(0)$, the swarm must be guided to a specified steady-state probability distribution v

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

subject to motion constraints (i.e., allowable transitions between bins), specified by an adjacency matrix A_a as:

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

The distribution x has the following interpretation: The element $x[i]$ is the probability of finding an agent in the i'th bin. If there are N agents, then $Nx[i]$ describes the expected number of agents in the i'th bin. Let $\mathbf{n} = [n[1], ..., n[m]]^T$ denote the actual number of agents in each bin. Then $n[i]$ is generally different from $Nx[i]$, although it follows from the independent and identically distributed agent realizations that $x = E[\mathbf{n}]/N$. 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 in general $n/N \neq x$, it will always be equal to x *on the average*, and can be made *arbitrarily close* to x with large number of agents.

III. Probabilistic Guidance Algorithm (PGA)

Suppose that the desired swarm distribution is given by the vector v. The key idea of the probabilistic guidance is to synthesize a column stochastic matrix [26], [27] $M \in \mathbb{R}^{m \times m}$, the *Markov matrix for PGA*, with v as its eigenvector corresponding to its largest eigenvalue 1 [26], [28],

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

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

$$M[i,j] = \mathcal{P}(r(t+1) \in R_i | r(t) \in R_j). \tag{6}$$

i.e., an agent in bin j transitions to bin i with probability $M[i, j]$. The matrix M determines the time evolution of the probability vector $x \in \mathbb{R}^m$ as

$$x(t+1) = Mx(t) \qquad t = 0, 1, 2, \ldots. \tag{7}$$

The probabilistic guidance problem becomes one of designing a specific Markov process (7) for x that converges to a desired distribution v. The constraint $Mv = v$ guarantees that v is a stationary distribution of M, which follows from the equation (7). In practice, it is desirable to impose additional constraints on matrix M to restrict allowable agent motion over a single time step. Let A_a be the corresponding adjacency matrix as in (4), then the motion constraints are imposed on M using the following constraint,

$$(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}. \tag{8}$$

This condition forces zeros of A_a^T to also be the zeros of M. The set of all admissible Markov matrices for a given probability vector v and adjacency matrix A_a is

$$\mathcal{M}(v, A_a) := \{\text{Markov matrices with } v \text{ satisfying (8)}\}. \tag{9}$$

The evolution of $x(t)$ is described as follows, see [7].

Lemma 1: Consider a swarm of N agents in $R = \cup_{i=1}^{m} R_i$ 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 t, and

$$M[i,j](t) := \mathcal{P}(r(t+1) \in R_i | r(t) \in R_j). \tag{10}$$

Then the probability density vector x over R evolves as

$$x(t+1) = M(t)x(t). \tag{11}$$

The probabilistic guidance algorithm is implemented by providing the matrix M to each agent, and then having each 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) Agent "i" moves to bin j, $r_k(t+1) \in R_j$, 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. The last two steps sample from the discrete distribution given by the column of M corresponding to the agent's bin. The following result [7] is useful and it shows that asymptotic convergence to v is ensured by imposing a the <u>spectral radius condition</u>,

$$\rho(M - v\mathbf{1}^T) < 1. \tag{12}$$

A proof of Theorem 1 can be found in [7].

Theorem 1: Consider the PGA above with column stochastic matrix M such that $Mv = v$. Then for any probability vector $x(0)$, it follows that $\lim_{t \to \infty} x(t) = v$ for the system (7) if and only if (12) holds.

The condition that $v > \mathbf{0}$, which is central to Perron-Frobenius theory, does not appear in Theorem 1, hence it presents weaker convergence conditions. The next result [7] is a useful extension on the Perron-Frobenious theory to characterize a Markov M matrix that ensures convergence when $v \geq \mathbf{0}$.

Theorem 2: Consider a Markov matrix M with $v \geq \mathbf{0}$ as in (5) where $v = (\ \mathbf{0}, \quad \hat{v}\ )$ and $\hat{v} \in \mathbb{R}^q$, $\hat{v} > \mathbf{0}$. Then $\rho(M - v\mathbf{1}^T) < 1$ if and only if M satisfies the following *generalized primitivity condition*

$$M = \begin{bmatrix} M_1 & \mathbf{0} \\ M_2 & M_3 \end{bmatrix} \qquad \text{where} \tag{13}$$

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

Theorems 1 and 2 imply the following result.

Corollary 1: Suppose that the PGA is used with Markov matrix M with v, $v = (\mathbf{0}, \hat{v})$ where $\hat{v} > \mathbf{0}$ and $x(t+1) = Mx(t)$ with $x(t)$ defined by (1). Then $\lim_{t\to\infty} x(t) = v$, $\forall x(0)$, if and only if M *satisfies the generalized primitivity condition.*

A. PGA with Metropolis-Hastings (M-H) Algorithm

This section presents a synthesis method for the matrix M used in PGA. The Metropolis-Hastings (M-H) algorithm [29], [30] is a Markov Chain Monte Carlo (MCMC) method for obtaining a sequence of random samples defined by propagating a special Markov chain. The following result appeared in [7], which is a generalization of the M-H algorithm.

Theorem 3: Consider the M-H algorithm, for $v \geq 0$, given by (14) where $\mathtt{i}(K) = A_a^T$ for a strongly connected adjacency matrix $A_a = A_a^T$, and matrix F as in (16) between the bins of the recurrent states

$$M[i,j] = \begin{cases} \text{as in (15)} & \text{if} \quad i \in I_r,\ j \in I_r \\ 0 & \text{if} \quad i \leq m_t,\ j \in I_r \\ 1/(\sum_{k\in I_r} A_a[j,k]) & \text{if} \quad i \in I_r, j \leq m_t, A_a[j,i] = 1 \\ 1/(\sum_{k\in I_{k+1}} A_a[j,k]) & \text{if} \quad i \in I_{k+1}, j \in I_k, A_a[j,i] = 1 \\ 0 & \text{elsewhere} \end{cases} \tag{14}$$

where, for $i \in I_r,\ j \in I_r$,

$$M[i,j] = \begin{cases} K[i,j]F[i,j] & \text{if} \quad i \neq j \\ K[j,j] + \sum_{k\neq j}(1 - F[k,j])K[k,j] & \text{if} \quad i = j \end{cases} \tag{15}$$

with K (*proposal matrix*) satisfies $K \geq 0$ and $\mathbf{1}^T K = \mathbf{1}^T$; $v > 0$; and F (*acceptance matrix*) satisfies the following two conditions for $i \neq j$, and

$$F[i,j] = \quad \alpha \min(1, R[i,j]) \quad \text{where} \quad \alpha \in (0,1]. \tag{16}$$

$$\text{where} \qquad R[i,j] = \frac{v[i]K[j,i]}{v[j]K[i,j]} \qquad i,j = 1,\ldots,m. \tag{17}$$

Then $M \in \mathcal{M}(v, A_a)$, and $\rho(M - v\mathbf{1}^T) < 1$.

IV. Swarm Behavior Analysis and Connections with Consensus Theory of Networked Agents

In the previous section, the desired behavior was parametrized by the target distribution vector v, which was prescribed to be the same among all agents. This section considers a more general case where v is not identically specified among the agents. This can happen when they are not initialized with the same objectives or when some of the agents change their objectives to react to the changes in their environment. Here, agents can act collaboratively to decide on a common objective with some rules of collaboration. This leads to a consensus among networked agents [15], [22], [23], [24], [31]. The possibility of having different desired swarm behavior specifications among agents leads to many complex and practically useful behaviors in the swarms.

Theorem 4: Consider a swarm of N agents in $R = \cup_{i=1}^m R_i$ where $R_i \cap R_j = \emptyset$ for $i \neq j$. Let $x[i](t) = \mathcal{P}(r(t) \in R_i)$ and

$$M_l[i,j](t) := \mathcal{P}(r_l(t+1) \in R_i | r_l(t) \in R_j) \tag{18}$$

where position vectors $r(t)$ is of an agent and $r_l(t)$ is of the l'th agent. Then the probability density x evolves as

$$\xi(t+1) = M(t)\xi(t) \ \text{ and } \ x(t) = \frac{1}{N}(\mathbf{1}^T \otimes I_m)\xi(t), \tag{19}$$

$$\begin{aligned} &\text{where } M(t) = \mathrm{diag}(M_1(t), \ldots, M_N(t)), \\ &\xi = \begin{pmatrix} x_1(t), & \ldots, & x_N(t) \end{pmatrix}, \quad x_l[i](t) = \mathcal{P}(r_l(t) \in R_i). \end{aligned} \tag{20}$$

Proof: Given M_l in (18), Theorem 1 implies that $x_l(t+1) = M_l(t)x_l(t),\ \ l = 1,\ldots,N$. This directly implies the first equation in (19) by simply augmenting the vectors $x_l(t), l = 1,...,N$. Next, letting A_l be the event that an agent is lth agent, note that $\mathcal{P}(r(t) \in R_i) = \sum_{l=1}^N \mathcal{P}(r(t) \in R_i | A_l)\mathcal{P}(A_l)$ where $\mathcal{P}(r(t) \in R_i | A_l) = \mathcal{P}(r_l(t) \in R_i) = x_l[i](t)$ and $\mathcal{P}(A_l) = 1/N$. Hence Total Probability theorem implies that $x[i](t) = \mathcal{P}(r(t) \in R_i) = (1/N)\sum_{l=1}^N x_l[i](t),\ \ i = 1,...,m$, which implies the second equation in (19). ∎

In the remainder of the paper, it is assumed that the probability of agent positions is given by (18) and, from Theorem 4, the swarm probability density evolves as in (19).

Next we consider a scenario where each agent starts with a distinct desired behavior specification, $v_l(0)$, $l = 1,...,N$. At each time t, the agent determines:

- $r_l(t+1)$ based on $v_l(t)$,
- $v_l(t+1)$ based on the agents that it communicates with at time t; $\mathcal{C}_l(t)$ is the set of agents that l'th agent communicates with at time t with a cardinality $N_l(t)$.

Now the question is: How does v_l evolve in time among agents? We propose a consensus approach that is widely used in the literature [23], for $l = 1,...,N$ with some $\delta > 0$

$$v_l(t+1) = (1 - \delta N_l(t))\, v_l(t) + \delta \sum_{k \in \mathcal{C}_l(t)} v_k(t). \tag{21}$$

By letting $\eta := (v_1, \ldots, v_N)$, the above recursive relationship can also be expressed as,

$$\begin{aligned} &\eta(t+1) = [(I - \delta\mathcal{L}_c(t)) \otimes I_m]\,\eta(t) \quad \text{where} \\ &\mathcal{L}_c(t) := \mathcal{L}(G_c(t)) = D_c(t) - A_c(t) \end{aligned} \tag{22}$$

and $D_c(t) \in \mathbb{R}^{N\times N}$ is a diagonal matrix with $D_c[i,i](t)$ denoting the number of agents connected to the ith agent at time t and $A_c(t)$ is the adjacency matrix such that $A_c[l,k](t) = 1$ when there is a communication link between agents k and l and it is 0 otherwise, and hence $\mathcal{L}_c(t)$ is the Laplacian of the communication graph $G_c(t)$ at time t.The following result on convergence of consensus appeared in the literature [22], [22], [23].

Lemma 2: Consider N agents applying a consensus algorithm in (22) over a communication network with an undirected connected graph $G_c(t)$ (without any self-loops or multiple edges) for all t. Then there exists $\delta > 0$ such that $\sigma(I - \delta\mathcal{L}_c(t)) = \{1\} \cup [0, \lambda)$ where $\lambda < 1$ and

$$\lim_{t\to\infty} v_l(t) = v, \forall l, \quad \text{where} \quad v = \frac{1}{N}\sum_{l=1}^N v_l(0). \tag{23}$$

The following is a new result on swarm convergence with respect to time-varying transition matrices and it will be useful to prove our main result of this section.

Lemma 3: Consider a swarm of N agents evolving as in Theorem 4 with a Markov matrix $M_l(t)$, and suppose that there exist unique probability vectors $v_l(t)$ such that $M_l(t)v_l(t) = v_l(t)$ for each l and t. If $\lim_{t\to\infty} M_l(t) = \hat{M}, \forall l$, such that $\hat{M}$ is a Markov matrix with v and $\rho(\hat{M} - v\mathbf{1}^T) < 1$ for some probability vector v, then $\lim_{t\to\infty} x(t) = v$.

Proof: First it is shown that $\lim_{t\to\infty} v_l(t) = v$ for all l. Since $\lim_{t\to\infty} \|M_l(t) - \hat{M}\| = 0$ and $\|v_l(t)\| \leq 1$ for all t and l, for any $\varepsilon > 0$ there exists $T > 1$ such that $\|(M_l(t)-\hat{M})v_l(t)\| \leq \|(M_l(t)-\hat{M})\|\|v_l(t)\| < \varepsilon$ for all $t \geq T$. This implies that $\lim_{t\to\infty}(M_l(t)-\hat{M})v_l(t) = 0$, which then implies that $\lim_{t\to\infty}(I-\hat{M})v_l(t) = 0$. Since v is the unique eigenvector of $\hat{M}$ for the eigenvalue one, $\lim_{t\to\infty} v_l(t) = v$. To show this, first observe that $I - \hat{M}$ has one dimensional null-space spanned by v. Therefore there exists a full column rank matrix $R \in \mathbb{R}^{m\times(m-1)}$ such that $B := (I - \hat{M})R$ is full column rank and $R^T v = 0$. Now $v_l(t) = \alpha_l(t)v + Rz_l(t)$ where $z_l(t) \in \mathbb{R}^{m-1}$. Suppose that $z_l(t)$ does not converge to zero, that is, there exists a subsequence $\{z_l(t_k)\}_{k=1,2,\ldots}$ such that $\|z_l(t_k)\| \geq \delta > 0$. This implies that: $\lim_{k\to\infty}(I-\hat{M})v_l(t_k) = \lim_{k\to\infty}(I-\hat{M})(\alpha_l(t_k)\hat{v} + Rz_l(t_k)) = \lim_{k\to\infty} Bz_l(t_k) = 0$. However $\|z_l(t_k\| \geq \delta > 0$ implies that $\|Bz_l(t_k\|^2 = z_l(t_k)^T B^T B z_l(t_k) \geq a\|z_l(t_k)\|^2 \geq a\delta^2 > 0$ where $a > 0$ is the smallest eigenvalue of $B^T B$, which a positive definite matrix since B is full column rank. This implies that $\lim_{k\to\infty} Bz_l(t_k) \neq 0$, which is a contradiction that resulted from the assumption that the subsequence $z_l(t)$ does not converge to zero. Hence $\lim_{t\to\infty} \|z_l(t)\| = 0$ and hence $\lim_{t\to\infty} v_l(t) - \alpha_l(t)v = 0$. This implies that $\lim_{t\to\infty} \mathbf{1}^T v_l(t) - \alpha_l(t)\mathbf{1}^T v = \lim_{t\to\infty} 1 - \alpha_l(t) = 1 - \lim_{t\to\infty} \alpha_l(t) = 0$ which implies that $\lim_{t\to\infty} \alpha_l(t) = 1$, and therefore $\lim_{t\to\infty} v_l(t) - \alpha_l(t)v = \lim_{t\to\infty} v_l(t) - v = 0 \;\Rightarrow\; \lim_{t\to\infty} v_l(t) = v$.

Next the evolution swarm probability density (11) is considered $x_l(t+1) = M_l(t)x_l(t)$. Then the evolution for the error vector $e_l(t) = x_l(t) - v_l(t)$ can be expressed as

$$\begin{aligned}
e_l(t+1) &= M_l(t)x_l(t) - v_l(t+1) \\
&= M_l(t)x_l(t) - v_l(t)\mathbf{1}^T x_l(t) - (v_l(t+1) - v_l(t)) \\
&= (M_l(t) - v_l(t)\mathbf{1}^T)x(t) - (v_l(t+1) - v_l(t)) \\
&= (M_l(t) - v_l(t)\mathbf{1}^T)e_l(t) \\
&\quad + M_l(t) - v_l(t)\mathbf{1}^T)v_l(t) - (v_l(t+1) - v_l(t)) \\
&= (M_l(t) - v_l(t)\mathbf{1}^T)e_l(t) + (v_l(t) - v_l(t+1)) \\
&= (\hat{M} - v\mathbf{1}^T)e_l(t) + \big[(M_l(t) - \hat{M}) \\
&\quad + (v - v_l(t))\mathbf{1}^T\big]\, e_l(t) + (v_l(t) - v_l(t+1)).
\end{aligned}$$

Since $e_l(t) = x_l(t) - v_l(t)$ and $x_l(t) \geq 0$ and $v_l(t)$ are probability vectors, $e_l(t)$ is a bounded signal. Hence $g_l(t)$ is a bounded signal such that $\lim_{t\to\infty} g_l(t) = 0$ where

$$g_l(t) := \big[(M_l(t) - \hat{M}) + (v - v_l(t))\mathbf{1}^T\big]e_l(t) + (v_l(t) - v_l(t+1)),$$

which follows from $\lim_{t\to\infty} M_l(t) = \hat{M}$ and $\lim_{t\to\infty} v_l(t) = v$. Then

$$e_l(t+1) = (\hat{M} - v\mathbf{1}^T)e_l(t) + g_l(t).$$

Since $\rho(\hat{M} - v\mathbf{1}^T) < 1$, the following inequality holds [32] for some $P = P^T \succ \mathbf{0}$, $P - \big(M - v\mathbf{1}^T\big)^T P \big(M - v\mathbf{1}^T\big) \succ \mathbf{0}$. Letting $V_l(t) = e_l(t)^T P e_l(t)$, this implies that

$$V_l(t+1) - V_l(t) \leq -e_l(t)^T Q e_l(t) + \underbrace{2|e_l(t)^T P g_l(t)| + g_l(t)^T P g_l(t)}_{:= h_l(t)},$$

for some $Q = Q^T \succ \mathbf{0}$, where $h_l(t)$ is a bounded nonnegative sequence with $\lim_{t\to\infty} h_l(t) = 0$ (since $e_l(t)$ is bounded and $\lim_{t\to\infty} g_l(t) = 0$). Since $Q = Q^T \succ \mathbf{0}$ is positive definite matrix, we have the following recursive inequality for some $\gamma \in (0,1)$, $V_l(t+1) \leq \gamma V_l(t) + h_l(t)$. Since $V_l(t)$ and $h_l(t)$ form bounded nonnegative sequences and $\lim_{t\to\infty} h_l(t) = 0$, for any $\varepsilon > 0$, there exists some positive integer $q > 1$ such that $h_l(t) < \varepsilon$ for all $t \geq q$ and

$$\begin{aligned}
V_l(q+k) &\leq \gamma^k V_l(q) + \varepsilon \sum_{i=0}^{k} \gamma^i \\
&\leq \gamma^k V_l(q) + \frac{1-\gamma^k}{1-\gamma}\varepsilon \leq \gamma^k V_l(q) + \frac{\varepsilon}{1-\gamma}.
\end{aligned}$$

Note that there exists some positive number $\bar{V}$ such that $V_l(t) \leq \bar{V}$ for all t and there exists a large enough k such that $\gamma^k < \varepsilon$. Consequently, for any $\varepsilon > 0$, there exists some large enough positive integer n such that

$$V_l(t) \leq \left(\bar{V} + \frac{1}{1-\gamma}\right)\varepsilon \quad \text{for all } t \geq n.$$

Since $\varepsilon > 0$ can be chosen arbitrarily small, this implies that $\lim_{t\to\infty} V_l(t) = 0$, and hence $\lim_{t\to\infty} e_l(t) = 0$. Therefore $\lim_{t\to\infty} v_l(t) = v$ for all l and, hence, $\lim_{t\to\infty} \xi(t) = \mathbf{1} \otimes v$. Since $x(t) = (1/N)(\mathbf{1}^T \otimes I_m)\xi(t)$,

$$\begin{aligned}
\lim_{t\to\infty} x(t) &= \lim_{t\to\infty} (1/N)(\mathbf{1}^T \otimes I_m)\xi(t) \\
&= (1/N)(\mathbf{1}^T \otimes I_m)(\mathbf{1} \otimes v) = (1/N)(\mathbf{1}^T\mathbf{1} \otimes v)
\end{aligned}$$

$\Rightarrow \lim_{t\to\infty} x(t) = v$. ∎

We next propose a method of collaborative swarm coordination by using averaging for consensus and the M-H algorithm, which is useful to build up complex swarm behaviors via collaborations between sub-swarms.

Theorem 5: Consider the swarm of N agents evolving as in Theorem 4, where matrix $M_l(t)$ is determined by modified M-H algorithm (14) with F determined by (16) and a proposal matrix K that is the same for each agent with $\mathtt{i}(K) = A_a$, where A_a is a strongly connected adjacency matrix. The M-H algorithm uses $v_l(t)$, $l = 1,\ldots,N$, that evolve as in (22) where the communication network satisfies the hypothesis of Lemma 2 with $\delta = 1/(2N-2)$. Then $\lim_{t\to\infty} x(t) = v$ where $v = \frac{1}{N}\sum_{l=1}^{N} v_l(0)$.

Proof: Since the communication graph satisfies the hypothesis of Lemma 2, it follows that $\lim_{t\to\infty} v_l(t) = v$. The modified M-H algorithm generates $M_l(t)$ such that $M_l(t)v_l(t) = v_l(t)$. Hence to arrive at the desired conclusion of the theorem by using Lemma 3, it must be shown that $\lim_{t\to\infty} M_l(t) = \hat{M}$ for some column stochastic matrix $\hat{M}$ that satisfies $\hat{M}v = v$ and $\rho(\hat{M} - v\mathbf{1}^T) < 1$.

First it is shown that $\lim_{t\to\infty} M_l(t) = \hat{M}$ where $\hat{M}$ corresponds to the matrix generated by the modified M-H algorithm for v. To show that it is first established that M generated by the modified M-H algorithm is a continuous function of v on $\mathbf{V}_k = \{v\colon v[i]=0, i=1,\ldots,k, v[i]>0, i>k, \mathbf{1}^T v = 1\}$. First observe that, since A_a is the specified adjacency matrix used by all agents, the "(1,1)" and "(2,1)" blocks of $M_l(t)$, $M_{l,1}(t)$ and $M_{l,2}$, are time invariant and common for all agents. The partitioning of any Markov matrix M is given as follows:

$$\mathcal{M}(v) = \begin{bmatrix} M_1 & \mathbf{0} \\ M_2 & M_3 \end{bmatrix},$$

where $v = (\mathbf{0}, \hat{v})$ with $\hat{v} > 0$ and $\mathcal{M}(v)$ represents the matrix generated by M-H algorithm with v, and hence $M_3 = \mathcal{M}(\hat{v})$. Since R in (17) is a continuous function of $\hat{v} > 0$ and F in (16) is a continuous function of R, F is a continuous function of $\hat{v} > 0$. This observation together with equation (15) directly imply that $\mathcal{M}$ is a continuous function of $\hat{v} > 0$. Hence $\mathcal{M}$ is a continuous function on $\mathbf{V}_k$ for any $0 \le k < N$. Now it is shown that there exist some integers $0 \le k < N$ and $T > 0$ such that $v_l(t) \in \mathbf{V}_k$ for all $t \ge T$. Suppose k is determined by v, that is, $v \in \mathbf{V}_k$. This implies that $v_l[i](0) = 0$ for all $i \le k$. Otherwise, since $v = (1/N)\sum_l v_l(0)$, there would be a nonzero entry of v at $i \le k$. Next let $\varepsilon := \min_{i>k} v[i]$. Since $\lim_{t\to\infty} v_l(t) = v$, there exists some T such that, for all l, $v_l(t)[i] > (1/2)\varepsilon$, $i > k$. Hence, since $v_l(t)[i] = 0$ for all t, $i \le k$, and l, $v_l(t) \in \mathbf{V}_k$ for all $t \ge T$. This then implies that, since $\mathcal{M}$ is continuous on $\mathbf{V}_k$, $\lim_{t\to\infty} M_l(t) = \lim_{t\to\infty} \mathcal{M}(v_l(t)) = \mathcal{M}(v) = \hat{M}$, which is a column stochastic matrix generated by the M-H algorithm (14), that is, $\hat{M}v = v$. Since the M-H algorithm generates $\hat{M}$ such that $\rho(\hat{M} - v\mathbf{1}^T) < 1$, the proof follows from Lemma 3. ■

V. Illustrative Numerical Example

The example is comprised of two cases which compare a noncollaborative swarm (Case 1), with a collaborative swarm (Case 2). In each case, the swarm is initialized with 5200 agents having a uniform distribution over the entire space, but comprised of two sub-swarms. Swarms comprised of two or more sub-swarms are denoted as *composite swarms*. The first sub-swarm consists of 3200 agents having a Markov matrix targeting the letter "E". The second sub-swarm consists of 2000 agents have a Markov matrix targeting the letter "I" (see Fig. 1). To aid visualization, agents of the sub-swarm targeting the letter "E" are colored blue, and agents of the sub-swarm targeting the letter "I" are colored red.

Case 1 results for a noncollaborative swarm are shown in Fig. 2. Case 2 results for a collaborative swarm are shown in Fig. 3. In each case, the swarm converges to a distribution for the number "8". Here, the numbers of agents in each sub-swarm have been specifically chosen so that the final distribution is uniform over the shape of the "8". The noncollaborative swarm converges by time $t = 19$ while the collaborative swarm is faster and converges by time $t = 9$. This shows that swarm convergence is achieved with or without collaboration. We also study the effect of inflicting a damage to the swarms. Specifically, at time $t = 20$, damage is inflicted on both the converged swarms. By $t = 45$, the swarms in both cases have re-converged to a steady-state distribution. However, the final converged distribution for the non-collaborative swarm is not uniform over the "8" (the recovery is not accomplished), while the collaborative swarm distribution is. This convergence to the desired distribution, even after damage, demonstrates how collaboration endows composite swarms with a self-repair capability that is not possible without collaboration. Snapshot images are provided

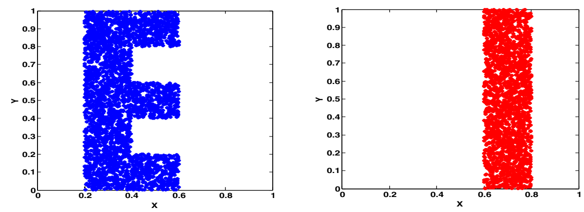

Fig. 1. Two sub-swarms corresponding to blue agents targeting the letter "E", and red agents targeting the letter "I". The swarm is designed to asymptotically approach a steady-state uniform distribution for the number "8": "E"+"I"="8" using both noncollaborative and collaborative guidance.

in Fig 2 and Fig 3 that show the evolution of the swarm distributions. To visualize the convergence, the time histories of $\|x(t) - v\|_1$, where $v = \frac{1}{N}\sum_{l=1}^{N} v_l(0)$ (average of the desired distributions), and $||\delta(t)||_1$ are given: $\delta(t)$ is a vector of the standard deviation of the swarm specifications, $i = 1,\ldots,m$

$$\delta[i](t) = \sqrt{\frac{1}{N-1}\sum_{l=1}^{N}\left(v_l[i](t) - \bar{v}[i](t)\right)^2} \quad \text{where} \quad \bar{v}(t) = \frac{1}{N}\sum_{l}^{N} v_l(t).$$

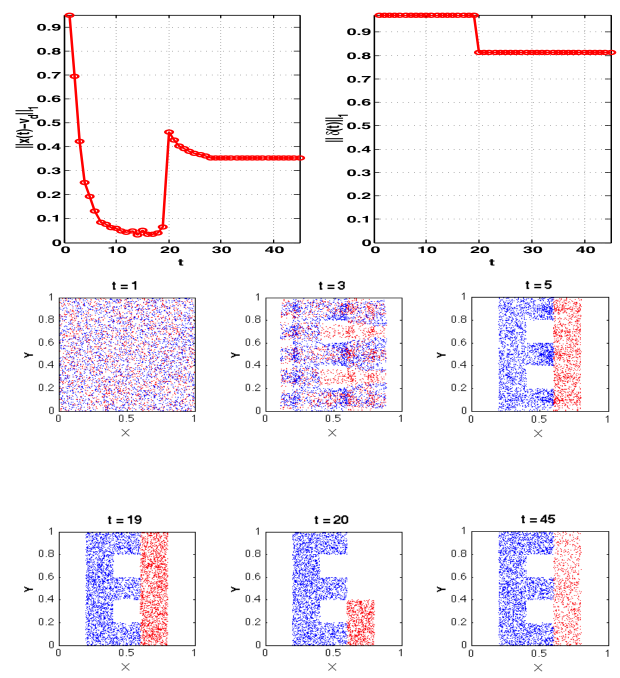

Fig. 2. **Case 1: Noncollaborative Swarm**.

VI. Conclusions

A probabilistic coordination method is introduced for swarm guidance. This method guides the swarm to conform to a prescribed probability distribution. The main idea is to have each agent follow an independent realization of a Markov chain. The desired distribution emerges as the

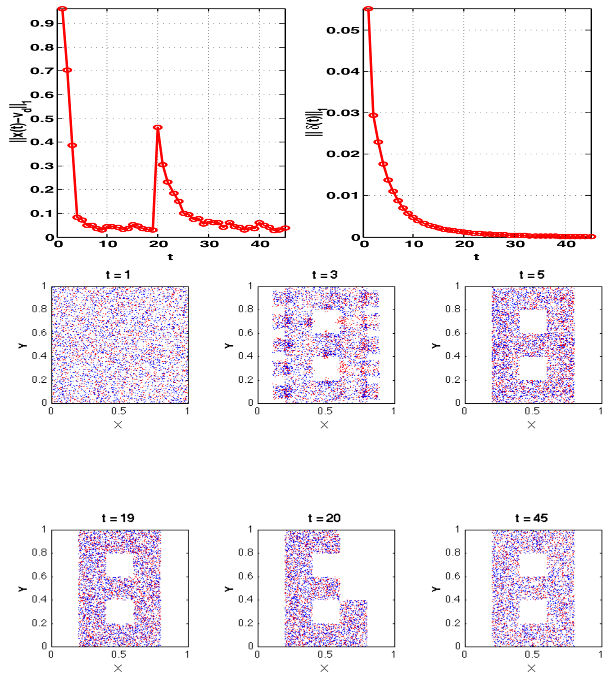

Fig. 3. **Case 2: Collaborative Swarm**.

ensemble of agents in the swarm maneuver about, asymptotically achieving a desired statistical steady-state condition, and eliciting a clear emergent behavior from the swarm. The current approach is different from earlier approaches in that it focuses on the guidance problem, and extends previous results by 1) explicitly allowing for motion constraints; 2) by allowing for strictly enforced "keep-out" regions; and 3) by allowing agents to act collaboratively through consensus.

ACKNOWLEDGMENTS

We acknowledge F. Hadaegh, M. Mandic, M. Quadrelli, E. Mettler of JPL, L. Blackmore of Space Exploration Technologies Corp., and S. Chung of University of Illinois for their valuable comments.

REFERENCES

[1] 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.

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

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

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

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

[6] J. Ramirez, M. Pavone, E. Frazzoli, and D. W. Miller. Distributed control of spacecraft formations via cyclic pursuit: Theory and experiments. *AIAA Jrnl. of Guidance, Control, and Dynamics*, 33(5):1655–1669, 2010.

[7] B. Açıkmeşe and D. S. Bayard. A Markov chain approach to probabilistic swarm guidance. *American Control Conference, Montreal, Canada*, pages 6300–6307, 2012.

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

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

[10] M. Pavone, E. Frazzoli, and F. Bullo. Adaptive and distributed algorithms for vehicle routing in a stochastic and dynamic environment. *IEEE Trans. on Automatic Control*, 56(6):1259–1274, 2011.

[11] M. Pavone and E. Frazzoli. Decentralized policies for geometric pattern formation and path coverage. *ASME Jrnl. of Dynamic Systems, Measurement, and Control*, 129(5):633–643, 2007.

[12] M. A. Hsieh, A. Halasz, S. Berman, and V. Kumar. Biologically inspired redistribution of a swarm of robots among multiple sites. *Swarm Intelligence*, 2(2-4):121–141, 2008.

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

[14] G. Arslan, J. R. Marden, and J. S. Shamma. Autonomous vehicle-target assignment: A game-theoretical formulation. *ASME Jrnl. of Dynamic Systems, Measurement, and Control*, 129(5):584–197, 2007.

[15] M. Mesbahi and M. Egerstedt. *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[16] B. Acikmese, F. Y. Hadaegh, D. P. Scharf, and S. R. Ploen. Formulation and analysis of stability for spacecraft formations. *IET Control Theory and Applications*, 1:461–474, 2007.

[17] D. Morgan, S.-J Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh. Swarm keeping strategies for spacecraft under j2 and atmospheric drag perturbations. *Journal of Guidance, Control, and Dynamics*, 35(5):1492–1506, 2012.

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

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

[20] A. R. Mesquita, J. P. Hespanha, and K. Astrom. Optimotaxis: A stochastic multi-agent on site optimization procedure. *Hybrid Systems: Computation and Control, Lecture Notes in Computional Science*, (4981):358–371, 2008.

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

[22] D. P. Bertsekas and J. N. Tsitsiklis. Comments on coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Trans. on Automatic Control*, 52(5):968–969, 2007.

[23] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, 49(9):1520–1533, 2004.

[24] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Trans. on Automatic Control*, 50(11):1867–1872, 2005.

[25] B. Açıkmeşe, M. Mandic, and J. L. Spyer. Decentralized observers with consensus filters for distributed discrete-time linear systems. *Automatica*, provisionally accepted, 2013.

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

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

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

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

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

[31] L. Moreau. Stability of multiagent systemswith time-dependent communication links. *IEEE Trans. on Automatic Control*, 50(2):169–182, 2005.

[32] M. Corless and A. E. Frazho. *Linear Systems and Control: An Operator Perspective*. Marcel Dekker, 2003.