Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Density Control for Decentralized Autonomous Agents with Conflict Avoidance

Nazlı Demir * **Behçet Açıkmeşe** * **Can Pehlivantürk** *

* *Department of Aerospace Engineering and Engineering Mechanics*
The University of Texas at Austin, TX 78712–1221, USA
ndemir@utexas.edu, behcet@austin.utexas.edu,
canpehlivanturk@gmail.com.

Abstract: This paper describes a method to control the density distribution of a large number of autonomous agents. Our approach leverages from the fact that there are a large number of agents in the system, and hence the time evolution of the probabilistic density distribution of agents can be described as a Markov chain. Once this description is obtained, a Markov chain matrix is synthesized to drive the multi-agent system density to a desired steady-state density distribution, in a probabilistic sense, while satisfying some motion and conflict avoidance constraints. Later, we introduce an adaptive density control method based on real time density feedback to synthesize a time-varying Markov matrix, which leads to better convergence to the desired density distribution. This paper also introduces a decentralized density computation method, which guarantees that all agents will have a best, and common, density estimate in a finite, with an explicit bound, number of communication updates.

Keywords: Multi-agent system control, convex optimization, linear matrix inequalities, Markov chain theory.

1. INTRODUCTION

This paper describes a method to control the density distribution of a large number of autonomous agents. The proposed approach breaks away from traditional multi-agent coordination methods (Lumelsky and Harinarayan, 1997; Richards et al., 2002; Tillerson et al., 2002; Scharf et al., 2003; Kim et al., 2004; Hadaegh et al., 2013; Açıkmeşe et al., 2007; Ramirez et al., 2010) that control individual agents motions directly and in a tightly coupled manner in order to achieve the desired collective behavior. Instead, we control the overall density distribution, i.e., all the agents collectively, motion directly by first synthesizing a Markov Chain that is used by locally to determine the agent motion commands and as a function of their state. Hence, the agent motion commands are generated locally and in a decentralized manner. Each agent makes independent motion plans to follow these commands, which result in the desired collective behavior/motion.

The proposed approach is based on formulating the multi-agent coordination problem as a density control problem. Here the overall density distribution of the multi-agent system is described as a discrete probability distribution over the operational region. The region is divided into subregions, i.e., bins, and the desired probability density is prescribed for each bin. Each agent is given a Markov chain matrix as a policy based on which the agents make independent decisions at each time step. These decisions and corresponding actions lead to a time evolution the overall probability density as the Markov chain defined by the Markov matrix. Since the density has a probabilistic interpretation, we need very large number of agents to have the density evolve exactly as the Markov chain defined by the Markov matrix. Hence, with finite number of agents, the time evolution is always an approximation. The density control is decomposed into generating motion commands between discrete time steps and motion plans are computed by each agent to follow these commands. Consequently, the multi-agent system coordination occurs due to higher level motion commands generated based on the Markov matrix policy, and it is done in a decentralized manner. Since the agent motion plans are generated by each agent, the whole multi-agent system control is achieved in a decentralized manner.

The basic idea of using a Markov chain for guiding large numbers of swarm agents to a desired spatial distribution is quite new. A different Markov chain based method appeared in (Chattopadhyay and Ray, 2009), using a probabilistic "disablement" approach found in (Lawford and Wonham, 1993). Other related stochastic approaches to swarm control have appeared in the literature (Mesquita et al., 2008; Grace and Baillieul, 2005; Pavone and Frazzoli, 2007; Pavone et al., 2011). The proposed approach is relatively new in that, it is the first to provide mathematical guarantees of satisfying all key requirements, ergodicity, conflict avoidance, and motion constraints in a decentralized manner by directly controlling the density distribution of the agents.

The key technical question that is addressed is: How does higher level coordination via Markov matrix policy incorporate agent mobility constraints? There are two constraints that are integrated into Markov matrix design, (i) the motion and (ii) the conflict avoidance constraints. The motion constraints are imposed by limiting the set of bins reachable from a each bin in the operational domain, i.e., an agent in a certain bin can only move to a subset of bins. The conflict avoidance constraints (e.g., collision avoidance), also referred to as safety constraints (Arapostathis et al., 2003), are imposed by limiting the density for each bin, that is, the density in each bin is not allowed to go beyond a prescribed level. The assumption is that, if the density is below the prescribed level, then the conflicts can be avoided locally by agents.

It is important to note that many classical results on the Markov chain synthesis uses the Perron-Frobenius theory of primitive matrices (Horn and Johnson, 1985, 1991; Fiedler, 1973; Berman and Plemmons, 1994), which focuses mainly on the ergodicity of the Markov chains. Perron-Frobenius theory is heavily leveraged here, but since it cannot handle safety constraints with ergodicity and motion constraints, we also made connections with Lyapunov theory (Boyd et al., 2004) and the duality theory with convex optimization. Furthermore Lyapunov theory and the resulting Linear Matrix Inequalities (LMIs) allow not only simultaneous incorporation of these constraints in the design, it allows to define performance metrics such as convergence rate or fuel use.

Our earlier work has developed the probabilistic interpretation and a Metropolis-Hastings algorithm to synthesize feasible Markov matrices with motion constraints (Açıkmeşe and Bayard, 2012, 2013b,a). The current paper extends this by: (i) Develop an *offline* LMI based Markov matrix synthesis methods with density upper bound, safety, constraints for conflict avoidance; (ii) Develop an *online* method to have time-varying Markov matrix based on real-time multi-agent system density feedback, i.e., adaptive density control.

Notation: The following is a partial list of notation used: $\mathbb{N}_+$ are nonnegative natural numbers; $\mathbb{R}^n$ is the n dimensional real vector space; $\mathbf{0}$ 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 $x \in \mathbb{R}^n$ and $A[i,j] = \mathrm{e}_i^T A\, \mathrm{e}_j$ for any $A \in \mathbb{R}^{n\times m}$; $Q = Q^T \succ (\succeq) 0$ 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; $v \in \mathbb{R}^n$ is said to be a *probability vector* if $v \geq \mathbf{0}$ and $\mathbf{1}^T v = 1$; prob denotes probability of a random variable; $\|v\|$ is the 2-norm of the vector v; For $P = P^T \succ 0$, $\|v\|_P = \|P^{1/2} v\|$; $P^{1/2} = U \Lambda^{1/2} U^T$ 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; $\mathrm{diag}(A) = (A[1,1], \ldots, A[n,n])$ 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 ($\max_{\lambda\in\sigma(A)} |\lambda|$); $\otimes$ denotes the Kronecker product; $\odot$ represents the Hadamard (Schur) product; A *directed graph* $\mathbf{G}_a(A) = (\mathbf{V}_a, \mathbf{E}_a)$ *of a matrix* A is defined $\mathtt{i}(A)$ 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. 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.

2. DENSITY CONTROL FORMULATION

This section introduces the probabilistic density control (PDC) problem formulation. The physical domain over which the 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 $t \in \mathbb{N}_+$ and $x(t)$ 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,

$$x[i](t) := \mathrm{prob}(r(t) \in R_i), \quad i = 1, ..., m. \tag{1}$$

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

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

where $r_k(t)$ denotes the position of the k'th agent at time index t, and the probabilities of these N events are statistically independent. We refer to $x(t)$ as the *probabilistic density 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 $x(0) \in \mathbb{R}^m$, $x(0) \geq 0$, $\mathbf{1}^T x_0 = 1$, it is desired to guide the agents toward a specified steady-state distribution v such that $v \in \mathbb{R}^m$, $v \geq 0$, $\mathbf{1}^T v = 1$,

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

subject to motion constraints given by an adjacency matrix A_a as follows:

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

The adjacency matrix A_a of the edges of a directed graph is used to specify the allowable transitions between bins.

The other important constraint to capture the physical limitations of the agents is imposed on the densities in each bin to capture the ability of agents for conflict avoidance. The density constraint is expressed as follows: $x[i](t) \leq d[i]$, $i = 1, \ldots, m$, or more compactly in a a vectorial form

$$x(t) \leq d \quad \forall\, t \in \mathbb{N}_+. \tag{5}$$

Clearly the assumption is that the given initial distribution does not violate the density constraint. The idea is that by controlling the maximum number of agents in any bin, we enable the agents to avoid any possible conflicts. For example, if we have unmanned aerial vehicles (UAVs), by keeping the number of UAVs in a given bin below a prescribed number, we can guarantee that any collisions locally detected can be avoided with a sense and avoid type algorithm locally. An implication of this assumption is that, if the number of agents go beyond the prescribed level, there are too many vehicles in a given subregion that the sense and avoid algorithm at hand looses its ability to ensure collision avoidance.

The idea behind density control is to mathematically control the propagation of probability vector x, rather than individual agent positions. While the actual distribution of 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, density control simplifies the underlying mathematics by assuming that there are a large number of agents in the system.

Here we will consider two control problems. In the first problem, each agent is guided based on knowledge of its own position $r_k(t) \in R_i$. In the second problem, agents are guided by feeding back from their own position, plus additional knowledge of the entire density distribution $x(t)$. We refer to the first problem as the *probabilistic density control problem*, and the second as the *probabilistic density control problem with feedback*.

3. DECENTRALIZED PROBABILISTIC DENSITY CONTROL ALGORITHM

In the probabilistic density control algorithm, each agent is given a Markov matrix and they make statistically independent decisions based on this policy. As a result of these decisions and corresponding actions, the overall density evolves according to the Markov chain policy.

The key idea of the probabilistic guidance law is to synthesize a column stochastic matrix (Horn and Johnson, 1985; Berman and Plemmons, 1994) $M \in \mathbb{R}^{m\times m}$, which will be referred as *Markov matrix*, that determines the time evolution of the probability distribution as the probabilistic control policy.

The entries of matrix M are defined as transition probabilities. Specifically, for any $t \in \mathbb{N}_+$

$$\begin{aligned} &M[i,j] = \mathrm{e}_i^T M \mathrm{e}_j = \mathrm{prob}\left(r(t+1) \in R_i | r(t) \in R_j\right) \\ &\forall i,j = 1,\dots,m. \end{aligned} \tag{6}$$

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 $x \in \mathbb{R}^m$ as

$$\begin{aligned} &x(t+1) = Mx(t), \quad t = 0,1,2,\dots, \\ &\text{with } x(0) \geq 0 \ \text{ and } \ \mathbf{1}^T x(0) = 1. \end{aligned} \tag{7}$$

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 $\mathbf{1}^T M^t x(0) = \mathbf{1}^T M^{(t-1)} x(0) = \dots = \mathbf{1}^T x(0) = 1$. Also note 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, that is

$$M \geq 0, \qquad \mathbf{1}^T M = \mathbf{1}^T. \tag{8}$$

We can characterize such matrices as follows.

Definition 1. Matrix $M \in \mathbb{R}^{m\times m}$ is a *Markov matrix*, $M \in \mathcal{M}^m$ if $M \geq 0$ and $\mathbf{1}^T M = \mathbf{1}^T$.

The evolution of the probability density is described for a time-varying M by the following theorem (first appeared in (Açıkmeşe and Bayard, 2012).

Lemma 1. Suppose we have 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 $x[i](t) = \mathrm{prob}(r(t) \in R_i)$ where $r(t)$ be the position vector of an agent at time instance t, and

$$M[i,j](t) := \mathrm{prob}(r(t+1) \in R_i | r(t) \in R_j). \tag{9}$$

Then the density vector x defined over R evolves as follows

$$x(t+1) = M(t)x(t). \tag{10}$$

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 (Papoulis, 1991) implies that,
$\mathrm{prob}(r(t+1) \in R_i) = \sum_{j=1}^m \mathrm{prob}(r(t+1) \in R_i | \mathrm{prob}(r(t) \in R_j))\mathrm{prob}(r(t) \in R_j)$.
Consequently, since
$M[i,j](t) = \mathrm{prob}(r(t+1) \in R_i | \mathrm{prob}(r(t) \in R_j))$, $x[i](t+1) = \mathrm{prob}(r(t+1) \in R_i)$,
and $x[j](t] = \mathrm{prob}(r(t) \in R_j)$, the equation (10) follows. ■

Throughout the paper, we will use M as the constant (offline) Markov chain policy and $M(t)$ as the time-varying (online) Markov chain policy based on the density feedback.

Assuming that each agent is able to determine its current bin and is given the updated Markov Policy $M(t)$ in each time step, the probabilistic density is controlled with the following algorithm:

PDC Algorithm with density feedback

1. Each agent determines its current bin, $r_k(t) \in R_i$.
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 $\sum_{l=1}^{j-1} M_k[l,i] \leq z \leq \sum_{l=1}^{j} M_k[l,i]$.

Note that each agent makes independent decisions.

4. CONVEX FORMULATIONS OF CONSTRAINTS FOR MARKOV MATRIX SYNTHESIS

4.1 Desired steady-state distribution

Suppose that it is desired to guide the agents to a specific steady-state probability distribution denoted by the vector v which can be imposed on the synthesis of the *Markov Matrix* with the following condition:

$$Mv = v \tag{11}$$

that is, v is the eigenvector of M corresponding to its largest eigenvalue 1 (Horn and Johnson, 1985; Fiedler, 2008). This constraint (11) guarantees that v is a stationary distribution of M, which follows from the equation (7).

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, that is, the probability distribution of the agents does not change with time for $t \geq T$.

4.2 Asymptotic Convergence

With given steady-state probability distribution v, it is desired to achieve the asymptotic convergence of the density distribution, $x(t)$ to v which can be guaranteed by imposing the *spectral radius condition* on the synthesis of the matrix M:

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

The necessary and sufficient conditions for asymptotic convergence to v is given in the following Lemma, whose proof can be found in (Açıkmeşe and Bayard, 2012).

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

From linear system theory (Corless and Frazho, 2003; Kalman and Bertram, 1960), if there exists a lyapunov matrix $P = P^T \succeq 0$, the spectral radius condition (12) is equivalent to following inequality for some $\lambda \in [0, 1)$:

$$\lambda^2 P - (M - v\mathbf{1}^T)^T P(M - v\mathbf{1}^T) \succeq \mathbf{0} \tag{13}$$

Note that this condition is a bilinear inequality with both M and P as solution variables. For a discrete time dynamics system, the Lyapunov inequality (13) is equivalent to the existence of matrices $P = P^T \succ 0$ and G such that the following matrix inequality holds for some $\lambda \in [0, 1)$ (de Oliveira et al., 1999; Açıkmeşe and Bayard, 2013a)

$$\begin{bmatrix} \lambda^2 P & (M - v\mathbf{1}^T)^T G^T \\ G(M - v\mathbf{1}^T) & G + G^T - P \end{bmatrix} \succeq 0. \tag{14}$$

Here, G is a prescribed quantity which is selected as $G = diag(v)^{-1}$ in our example and λ determines the convergence rate to the desired stationary distribution v.

We have an LMI characterization because it is desired to impose additional constraints like safety constraints together with ergodicity constaints.

4.3 Motion Constraints

Additional constraints are imposed 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 using the following constraint,

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

4.4 Conflict Avoidance Constraints

The conflict avoidance constraints are also known as density upper bound or safety constraints (Arapostathis et al., 2003). The idea is that we can avoid any possible conflicts between agents by setting an upper bound for the density of each bin, in other words, by limiting the maximum number of agents for each bin. The conflict avoidance constraints ensures that the probability of the event that an agent is in the i^{th} bin stays below a prescribed value during the time interval, that is,

$$x(t)[i] \leq d[i] \quad \forall\, i = 0, 1, \ldots, m \;\text{ and }\; \forall\, t = 0, 1, \ldots, \tag{16}$$

which can also be expressed as

$$x(t) \leq d \;\; \forall\, t = 0, 1, \ldots, \;\text{ where }\; \mathbf{0} < d < \mathbf{1},\; \mathbf{1}^T d > 1, \tag{17}$$

i.e., $d \in \mathbb{R}^m$ and $d[i]$ defines the upper bound on the probability of being at the i'th bin. Here the initial probabilistic density distribution is assumed to satisfy the conflict avoidance constraints. These constraints are imposed in the Markov matrix by equivalent linear inequality constraints that are presented in the Theorem 1 whose proof can be found in (Demir et al., 2014). The theorem is proved by using the duality theory of convex optimization, which also provides a useful geometrical insight. It presents an equivalent convex optimization formulation that does lend itself well to computationally tractable synthesis by using Interior Point Method (IPM) algorithms (Nesterov and Nemirovsky, 1994).

Theorem 1. Consider the Markov chain given by (7) with $M \in \mathcal{M}^m$. For every $x(0) \leq d$, we have $x(t) \leq d \; \forall t = 1, 2, ...$, i.e. the density upper bound constraint holds, if and only if the following condition holds:
There exist $S \in \mathbb{R}^{m\times m}$ and $y \in \mathbb{R}^m$ such that

$$\begin{gathered} S \geq \mathbf{0}, \quad M + S + y\mathbf{1}^T \geq 0, \\ y + d \geq (M + S + y\mathbf{1}^T)d. \end{gathered} \tag{18}$$

5. SYNTHESIS OF MARKOV MATRIX

5.1 LMI Synthesis Without Density Feedback

In this section, we provide the convex optimization problem for the synthesis of the Markov matrix which satisfies the collision avoidance, ergodicity, and transition constraints. All these constraints have been formulated as equivalent linear equality and linear inequality constraints in section IV. Hence, we can construct the LMI optimization problem with the given constraints which minimizes the fuel or energy use. This can be achived by making $M \simeq I$; note that $M = I$ is a limiting case since it stops the movement of all agents. Setting the following function as the cost we can make M as close as to I and hence achieve the minimum fuel or energy (Açıkmeşe and Bayard, 2013a):

$$\mathbf{1}^T(\mathbf{1} - diag(M)) \tag{19}$$

With prescribed G and the convergence rate λ, the following LMI optimization problem can be solved to find the desired Markov Matrix:

$$\begin{aligned}
&\min_{M,P,S,y} \quad \mathbf{1}^T(\mathbf{1}-\mathrm{diag}(M)) \qquad \text{s.t}\\
&\mathbf{1}^T M = \mathbf{1}^T, \quad M \geq 0, \quad Mv = v\\
&(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}\\
&S \geq \mathbf{0}, \qquad M + S + y\mathbf{1}^T \geq 0,\\
&y + d \geq (M + S + y\mathbf{1}^T)d\\
&\begin{bmatrix} \lambda^2 P & (M - v\mathbf{1}^T)^T G^T \\ G(M - v\mathbf{1}^T) & G + G^T - P \end{bmatrix} \succeq 0\\
&P = P^T \succ 0, \quad G = \mathrm{diag}(v)^{-1}
\end{aligned} \tag{20}$$

5.2 QP Synthesis With Density Feedback

In this section, we provide a Quadratic Programming (QP) optimization problem for the synthesis of time-varying Markov Matrix. When the measurements or estimation of the probabilistic density distribution is available, they can be utilized to update the Markov Matrix which leads to a better convergence to desired density distribution. Here the assumption is that there is a Central Processing Agent (CPA) that determines the probabilistic density. Then the CPA computes a Markov Matrix and this information is communicated to all agents. Later we will propose a decentralized density estimation method to remove the need for a CPA.

We propose a receding horizon approach to synthesizing the M matrix for given $x(t)$. In order to do that we assume that there exists a Markov Chain Policy $\hat{M}$, a stochastic matrix satisfying the constraints of the optimization problem (20). Suppose that P is a Lyapunov matrix such that $P = P^T \succ \mathbf{0}$ and, for some $\lambda \in [0, 1)$, note that this matrix can be obtained offline by simply solving a Lyapunov inequality (13) for a given $\hat{M}$. Now given $x(t)$, we first solve (online) the following optimization problem:

$$\begin{aligned}
&\min_{M} \ \|Mx(t) - v\|_P \quad \text{subject to}\\
&M \geq 0, \quad \mathbf{1}^T M = \mathbf{1}^T, \quad (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}\\
&\quad x(t+1) = M(t)x(t), \quad x(t+1) \leq d
\end{aligned} \tag{21}$$

Note that the above optimization problem is a quadratic-programming (QP). Consequently, solving the optimization problem described by (21) synthesizes the *Markov matrix* M that leads to the closest density distribution to v, with respect to a norm defined by P, with the least effort. Note that $M = I$ is a solution of the second optimization problem when $x(t) = v$. The next question is whether this algorithm will lead to convergence to the desired density distribution or not, which is answered by the following lemma.

Proposition 1. Suppose that there exists $\hat{M} \in \mathcal{M}^m$ and $P = P^T \succ \mathbf{0}$ satisfying the inequality (13). Then the PDC approach that utilizes the *Markov Matrix M* obtained by solving QP given by (21) results in $\lim_{t\to\infty} x(t) = v$ for any $x(0) \geq 0$ with $\mathbf{1}^T x(0) = 1$.

Proof: Let $e(t) := x(t) - v$, which implies that $e_{t+1} = M(t)x(t) - v$, where $M(t)$ is the optimal solution of the synthesis via solving the optimization problem given by (21). Since $\hat{M}$ is also a feasible solution of this optimization problem, $\hat{M}v = v$, and $\mathbf{1}^T x(t) = 1$, we have

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

The inequality (13) implies that, for all $e(t)$,

$$\lambda^2\|e(t)\|_P^2 \geq \|(\hat{M} - v\mathbf{1}^T)e(t)\|_P^2.$$

Now, by using $\|e(t+1)\|_P \leq \|(\hat{M} - v\mathbf{1}^T)e(t)\|_P$,this implies that $\|e(t+1)\|_P \leq \lambda\|e(t)\|_P$, which then implies that $\|e(t)\|_P \leq \lambda^t\|e(0)\|_P$. Since $\lambda \in [0, 1)$: $\lim_{t\to\infty} \|e(t)\|_P = 0$, which implies that $\lim_{t\to\infty} \|e(t)\| = 0$. ■

Based on the Proposition 1, if there exist a Markov Matrix satisfying the constraints defined in (20), the following problem can be solved to update $M(t)$ in each time step:

$$\begin{aligned}
&\min_{M} \|M(t)x(t) - v\|_P \qquad \text{s.t}\\
&\mathbf{1}^T M(t) = \mathbf{1}^T, \quad M(t) \geq 0, \quad M(t)v = v\\
&(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M(t) = \mathbf{0}\\
&x(t+1) = M(t)x(t), \quad x(0) = x_0 \leq d\\
&x(t) \leq d, \ t = 1, 2, \ldots
\end{aligned} \tag{22}$$

5.3 A Decentralized Density Estimation Algorithm

This section introduces a decentralized counting algorithm for decentralized density estimation used in updating the Markov matrix. It is assumed that the bins are selected in a way that each agent can communicate with all other agents in its bin. It is also assumed that all agents from all neighboring bins can receive the broadcast from an agent in the bin. It is possible for the communication radius of an agent to span multiple bins and therefore multiple bins can be connected to each other. We define a communication adjacency matrix that shows which bins are able to communicate to each other. Let A_c be the corresponding communication adjacency matrix, that is, $A_c[i, j] = 1$ if the communication from bin i to bin j is feasible, and is zero otherwise.

The counting process consists of the following steps. When a density update is requested, all agents broadcast their ID and their current bin numbers. This enables each agent to determine the number of agents located in its own bin. After the first step, the agents broadcast the number of agents in the bins that they know of and update this data with the new information from other agents. This step is repeated until the information is uniformly shared among all bins. The propagation of density information across the bins is illustrated in Figure 1. With this method, it is guaranteed that each agent will be able to compute the correct density after a finite number of communication updates. $A_c^n > 0$ where n is the number of steps required to achieve uniformly shared information. Note that it is guaranteed that $n \leq m - 1$ (Deo, 1974).

6. NUMERICAL EXAMPLE

This example demonstrates the probabilistic density control algorithm on a multi-agent system of $N = 1500$

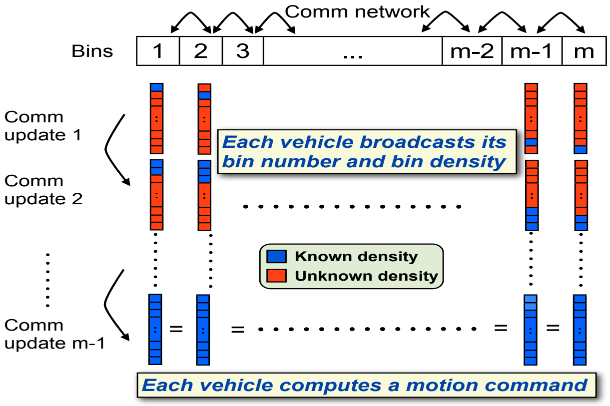

Fig. 1. An illustration of decentralized density computation algorithm

autonomous agents that are distributed on a U-shaped region R which is partitioned to 7 equally sized rectangular bins. Initially, the agents are assumed to be uniformly distributed in first two bins, i.e. $x(0) = [0.5\ 0.5\ 0\ 0\ 0\ 0\ 0]^T$. The convergence rate, desired density distribution and the density upper bound are set as $\lambda = 0.975$, $x(0) = [0.5\ 0.5\ 0\ 0\ 0\ 0\ 0]^T$, $v = [0.05\ 0.05\ 0.05\ 0.07\ 0.07\ 0.35\ 0.36]^T$ and $d = [0.5\ 0.5\ 0.15\ 0.14\ 0.13\ 0.5\ 0.5]^T$ respectively. For this example, it is desired that an agent can travel at most two bins in a single time step and it cannot perform a transverse transition(e.g. transition from 2^{nd} bin to 5^{th} bin). The adjacency (motion constraint) matrix which satisfies such contraints is given as follows:

$$A_a = \begin{bmatrix} 1&1&1&0&0&0&0 \\ 1&1&1&0&0&0&0 \\ 1&1&1&1&0&0&0 \\ 0&0&1&1&1&0&0 \\ 0&0&0&1&1&1&1 \\ 0&0&0&0&1&1&1 \\ 0&0&0&0&1&1&1 \end{bmatrix}.$$

The convex optimization problem given in (20) is solved by using YALMIP (Löfberg, 2004).

Using the result of the the Proposition 1, since there exists a Markov Chain Matrix satisfying the constraints of the problem (20), we were able to compute a time-varying Markov matrix to control the multi-agent system. The density evolution in the prescribed region is shown in Figure 2. The performance of the density control is

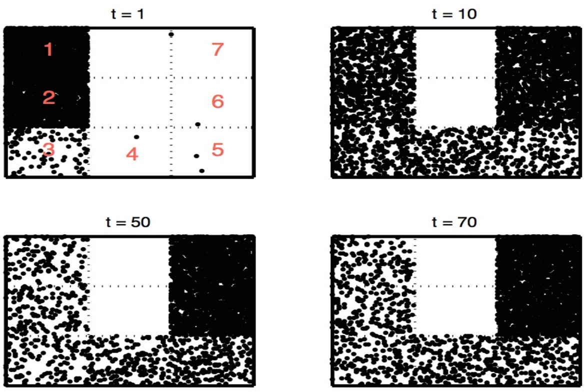

Fig. 2. Evolution of density distribution with time.

evaluated by recording the evolution of the error $x(t) - v$ in each bin. For comparison purposes simulation is performed for four different cases: (i) Constant M without density constraint; (ii) Time-varying M without density constraint; (iii) Constant M with density constraint; (iv) Time-varying M with density constraint. Error growth for all cases are shown in Figure 3 and the time histories of the density for the bins are shown in Figure 4. For the constant M case, the densities in the bins located at the bottom in Figure 2 (3^{rd},4^{th} and $5^{th}bin$) go beyond the desired upper bounds for a faster convergence rate when the density constraints are not imposed. Then imposing the safety constrains reduced the convergence rates but provided safety. Updating the *Markov Matrix* in each time step enhances the convergence rate drastically in both cases i.e. with and without the density constraints.

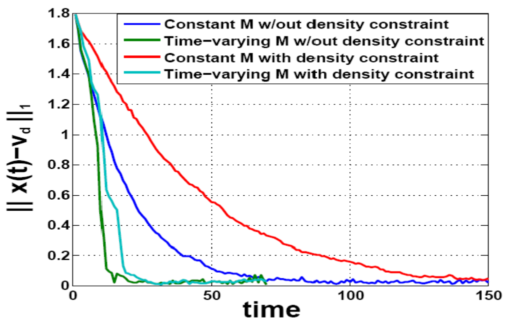

Fig. 3. Evolution of error with time for four different cases

7. CONCLUSIONS

This paper presented a Markov chain based method for controlling the probabilistic density distribution , which is decentralized in the sense that each agent makes statistically independent decisions to achieve the prescribed final distribution. The main contribution of this paper is time-varying Markov matrices with adaptive density control via real time density feedback. This approach produces better convergence than the no density feedback case and this is illustrated in a numerical example. We also introduced a decentralized counting algorithm for density estimation which guarantees finite time convergence.

REFERENCES

Açıkmeşe, B. and Bayard, D.S. (2012). A markov chain approach to probabilistic swarm guidance. *American Control Conference, Montreal, Canada*, 6300–6307.

Açıkmeşe, B. and Bayard, D.S. (2013a). Markov chain approach to probabilistic guidance for swarms of autonomous agents. *Invited paper, accepted with a minor revision, Asian Journal of Control.*

Açıkmeşe, B. and Bayard, D. (2013b). Probabilistic swarm guidance for collaborative autonomous agents. *Revised version in review for IEEE Trans. on Automatic Control.*

Açıkmeşe, B., Hadaegh, F.Y., Scharf, D.P., and Ploen, S.R. (2007). Formulation and analysis of stability for spacecraft formations. *IET Control Theory & Applications*, 1(2), 461–474.

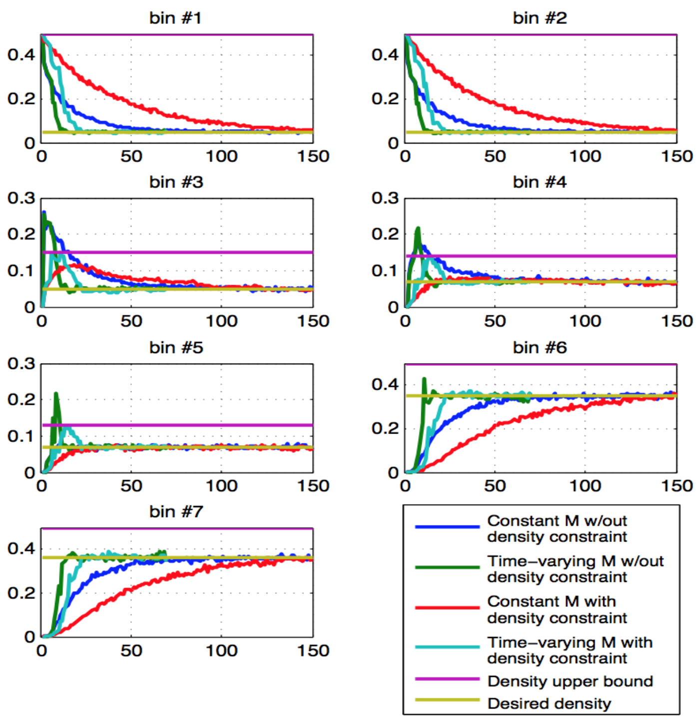

Fig. 4. Time history of the density of each bin for four different cases.

Arapostathis, A., Kumar, R., and Tangirala, S. (2003). Controlled markov chains with safety upper bound. *IEEE Transactions on Automatic Control*, 48(7), 1230–1234.

Berman, A. and Plemmons, R.J. (1994). *Nonnegative Matrices in the Mathematical Sciences*. SIAM.

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

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

Corless, M. and Frazho, A.E. (2003). *Linear Systems and Control: An Operator Perspective*. Marcel Dekker.

de Oliveira, M., Bernussou, J., and Geromel, J. (1999). Anew discrete-time robust stability condition. *Systems and Control Letters*, 261–265.

Demir, N., Açıkmeşe, B., and Harris, M. (2014). Convex optimization formulation of density upper bound constraints in Markov chain synthesis. *Submitted to American Control Conference*.

Deo, N. (1974). *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2), 298–305.

Fiedler, M. (2008). *Special Matrices and Their Applications in Numerical Mathematics*. Dover.

Grace, J. and Baillieul, J. (2005). Stochastic strategies for autonomous robotic serveillance. *Proc. IEEE Conf. on Decision and Control*.

Hadaegh, F., Açıkmeşe, B., Bayard, D., Singh, G., Mandic, M., Chung, S., and Morgan, D. (2013). Guidance and control of formation flying spacecraft: From two to thousands. *Festschrift honoring John L. Junkins, Tech Science Press*, 327–371.

Horn, R.A. and Johnson, C.R. (1985). *Matrix Analysis*. Cambridge University Press.

Horn, R.A. and Johnson, C.R. (1991). *Topics in Matrix Analysis*. Cambridge University Press.

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

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

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

Löfberg, J. (2004). YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*. Taipei, Taiwan. URL `http://control.ee.ethz.ch/ joloef/yalmip.php`.

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

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

Nesterov, Y. and Nemirovsky, A. (1994). *Interior-point Polynomial Methods in Convex Programming*. SIAM.

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill.

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

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

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

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

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

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