

Copyright  
by  
Nazlı Demir  
2014

The Report committee for Nazli Demir Certifies that this is the  
approved version of the following report:

**Decentralized Probabilistic Density Control of Swarm  
of Autonomous Agents with Conflict Avoidance  
Constraints**

APPROVED BY  
SUPERVISING COMMITTEE:

---

Behçet Açıkmeşe, Supervisor

---

Efstathios Bakolas

**Decentralized Probabilistic Density Control of Swarm  
of Autonomous Agents with Conflict Avoidance  
Constraints**

by

**Nazlı Demir, B.S.As.E.**

**REPORT**

Presented to the Faculty of the Graduate School of  
The University of Texas at Austin  
in Partial Fulfillment  
of the Requirements  
for the Degree of

**MASTER OF SCIENCE IN ENGINEERING**

THE UNIVERSITY OF TEXAS AT AUSTIN

August 2014

Dedicated to my family.

## Acknowledgments

I would like to express my gratitude to my advisor Dr. Behçet Açıkmeşe for the continuous support to my research, also for his motivation and enthusiasm which encouraged me all the time of research. I also would like to thank Can Pehlivan Türk for his valuable effort in this study and his friendship. Finally, I would like to acknowledge Dr. Efstathios Bakolas for taking time to read this report and for his suggestions. This work will appear in *IFAC World Congress, South Africa, 2014*.

# Decentralized Probabilistic Density Control of Swarm of Autonomous Agents with Conflict Avoidance Constraints

Nazlı Demir, M.S.E.

The University of Texas at Austin, 2014

Supervisor: Behçet Açıkmeşe

This report describes a method to control the density distribution of a large number of autonomous agents. The approach is based on 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. The main contribution of this paper is the synthesis of a Markov matrix which will guide the multi-agent system density to a desired steady-state density distribution, in a probabilistic sense, while satisfying some motion and safety constraints. Also, an adaptive density control method based on real time density feedback is introduced to synthesize a time-varying Markov matrix, which leads to better convergence to the desired density distribution. Finally, a decentralized density computation method is described. This method guarantees that all agents will have a best, and common, density estimate in a finite, with an explicit bound, number of communication updates.

# Table of Contents

<b>Acknowledgments</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>List of Figures</b>	<b>viii</b>
<b>Chapter 1. Introduction</b>	<b>1</b>
1.1 Notation . . . . .	4
<b>Chapter 2. Formulation of the Density Control Problem</b>	<b>6</b>
<b>Chapter 3. Decentralized Probabilistic Density Control Algorithm</b>	<b>9</b>
<b>Chapter 4. Convex Formulations of Constraints for Markov Matrix Synthesis</b>	<b>12</b>
4.1 Desired steady-state distribution . . . . .	12
4.2 Asymptotic Convergence . . . . .	12
4.3 Motion Constraints . . . . .	14
4.4 Conflict Avoidance Constraints . . . . .	15
<b>Chapter 5. Synthesis of Markov Matrix</b>	<b>17</b>
5.1 LMI Synthesis Without Density Feedback . . . . .	17
5.2 QP Synthesis With Density Feedback . . . . .	18
5.3 A Decentralized Density Estimation Algorithm . . . . .	21
<b>Chapter 6. Numerical Example</b>	<b>23</b>
<b>Chapter 7. Conclusions</b>	<b>28</b>
<b>Bibliography</b>	<b>33</b>

## List of Figures

5.1	An illustration of decentralized density computation algorithm	22
6.1	Evolution of density distribution with time. . . . .	25
6.2	Evolution of error with time for four different cases . . . . .	26
6.3	Time history of the density of each bin for four different cases.	27

# Chapter 1

## Introduction

This report describes a method to control the density distribution of a large number of autonomous agents. Existing guidance methods for multi-agent systems [25, 32, 34, 33, 22, 18, 4, 31] controls individual agents motions directly and in a tightly coupled manner in order to achieve the desired collective behavior. The proposed approach [11, 13, 12] controls the overall density distribution 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. Recent work in this area [1, 3, 2] has developed the probabilistic interpretation and a Metropolis-Hastings algorithm to synthesize feasible Markov matrices with motion constraints. A different Markov chain based method appeared in [8], using a probabilistic “disablement” approach found in [23]. Other related stochastic approaches to swarm control have appeared in the literature [26, 17, 29, 30]. 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 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 [5], 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 [19, 20, 15, 6], 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 [7] 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.

The report is organized as follows: Chapter 2 formulates the density control problem; Chapter 3 introduces the decentralized probabilistic density

control algorithm; Chapter 4 introduces the desired constraints and the convex formulations of these constraints; Chapter 5 presents the synthesis of the Markov matrix; Chapter 6 presents a numerical example to illustrate the theoretical results of the paper and finally conclusions are stated in Chapter 7.

## 1.1 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;  $\mathbf{e}_i$  is a vector of appropriate dimension with its  $i$ th entry +1 and its other entries zeros;  $x[i] = \mathbf{e}_i^T x$  for any  $x \in \mathbb{R}^n$  and  $A[i, j] = \mathbf{e}_i^T A \mathbf{e}_j$  for any  $A \in \mathbb{R}^{n \times m}$ ;  $Q = Q^T \succ (\succeq) \mathbf{0}$  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  $\mathbf{1}^T v = 1$ ;  $\text{prob}$  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;  $\text{diag}(A) = (A[1, 1], \dots, A[n, n])$  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  $\mathbf{i}(A)$  is the indicator matrix for any matrix  $A$ , whose entries are given by

$\mathbf{i}(A)[i, j] = 1$  if  $A[i, j] \neq 0$  and  $\mathbf{i}(A)[i, j] = 0$  otherwise. by letting  $\mathbf{V}_a$  be the set of integers  $1, 2, \dots, n$  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.

## Chapter 2

### Formulation of the Density Control Problem

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  $R_i$ ,  $i = 1, \dots, m$ , 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  $t$ ,

$$x[i](t) := \text{prob}(r(t) \in R_i), \quad i = 1, \dots, m. \quad (2.1)$$

The time index  $t$  will also be referred to as the “stage”. Consider a group comprised of  $N$  agents. Each agent is assumed to act independently of the other agents, so that (2.1) holds for  $N$  separate events,

$$x[i](t) := \text{prob}(r_k(t) \in R_i), \quad k = 1, \dots, N \quad (2.2)$$

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  $v \in \mathbb{R}^m$ ,  $v \geq 0$ ,  $\mathbf{1}^T v = 1$ ,

$$\lim_{t \rightarrow \infty} x[i](t) = v[i] \quad \text{for } i = 1, \dots, m, \quad (2.3)$$

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

$$A_a[i, j] = 0 \Rightarrow r(t+1) \notin R_j \text{ when } r(t) \in R_i, \forall t \in \mathbb{N}_+ \quad (2.4)$$

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:  $x[i](t) \leq d[i]$ ,  $i = 1, \dots, m$ , or more compactly in a vectorial form

$$x(t) \leq d \quad \forall t \in \mathbb{N}_+. \quad (2.5)$$

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 loses 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*.

# Chapter 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 [19, 6]  $M \in \mathbb{R}^{m \times m}$ , 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  $t \in \mathbb{N}_+$

$$M[i, j] = e_i^T M e_j = \text{prob}(r(t+1) \in R_i | r(t) \in R_j) \quad (3.1)$$

$$\forall i, j = 1, \dots, m.$$

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

$$x(t+1) = Mx(t), \quad t = 0, 1, 2, \dots, \quad (3.2)$$

with  $x(0) \geq 0$  and  $\mathbf{1}^T x(0) = 1$ .

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

$$M \geq 0, \quad \mathbf{1}^T M = \mathbf{1}^T. \quad (3.3)$$

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 [1]).

**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  $x[i](t) = \text{prob}(r(t) \in R_i)$  where  $r(t)$  be the position vector of an agent at time instance  $t$ , and*

$$M[i, j](t) := \text{prob}(r(t+1) \in R_i | r(t) \in R_j). \quad (3.4)$$

*Then the density vector  $x$  defined over  $R$  evolves as follows*

$$x(t+1) = M(t)x(t). \quad (3.5)$$

*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 [28] implies that,  $\text{prob}(r(t+1) \in R_i) = \sum_{j=1}^m \text{prob}(r(t+1) \in R_i | \text{prob}(r(t) \in R_j)) \text{prob}(r(t) \in R_j)$ .

Consequently, since

$$M[i, j](t) = \text{prob}(r(t+1) \in R_i | \text{prob}(r(t) \in R_j)), x[i](t+1) = \text{prob}(r(t+1) \in R_i),$$

and  $x[j](t) = \text{prob}(r(t) \in R_j)$ , the equation (3.5) follows.  $\square$

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.

## Chapter 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:

$$Mv = v \tag{4.1}$$

that is,  $v$  is the eigenvector of  $M$  corresponding to its largest eigenvalue 1 [19, 16]. This constraint (4.1) guarantees that  $v$  is a stationary distribution of  $M$ , which follows from the equation (3.2). 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  $M$ :

$$\rho(M - v\mathbf{1}^T) < 1. \quad (4.2)$$

The necessary and sufficient conditions for asymptotic convergence to  $v$  is given in the following Lemma, whose proof can be found in [1].

**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  $\lim_{t \rightarrow \infty} x(t) = v$  for the system (3.2) if and only if  $\rho(M - v\mathbf{1}^T) < 1$ .*

From linear system theory [9, 21], if there exists a Lyapunov matrix  $P = P^T \succeq 0$ , the spectral radius condition (4.2) is equivalent to following inequality for some  $\lambda \in [0, 1)$ :

$$\lambda^2 P - (M - v\mathbf{1}^T)^T P (M - v\mathbf{1}^T) \succeq \mathbf{0} \quad (4.3)$$

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 (4.3) is equivalent to the existence of matrices  $P = P^T \succ 0$  and  $G$  such that the following matrix inequality holds for some  $\lambda \in [0, 1)$  [10, 2]

$$\begin{bmatrix} \lambda^2 P & (M - v\mathbf{1}^T)^T G^T \\ G(M - v\mathbf{1}^T) & G + G^T - P \end{bmatrix} \succeq 0. \quad (4.4)$$

Here,  $G$  is a prescribed quantity which is selected as  $G = \text{diag}(v)^{-1}$  in our example and  $\lambda$  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 constraints.

### 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,

$$(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}. \quad (4.5)$$

## 4.4 Conflict Avoidance Constraints

The conflict avoidance constraints are also known as density upper bound or safety constraints [5]. 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,

$$x(t)[i] \leq d[i] \quad \forall i = 0, 1, \dots, m \quad \text{and} \quad \forall t = 0, 1, \dots, \quad (4.6)$$

which can also be expressed as

$$x(t) \leq d \quad \forall t = 0, 1, \dots, \quad \text{where} \quad \mathbf{0} < d < \mathbf{1}, \quad \mathbf{1}^T d > 1, \quad (4.7)$$

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 [12]. 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 [27].

**Theorem 1.** *Consider the Markov chain given by (3.2) with  $M \in \mathcal{M}^m$ . For every  $x(0) \leq d$ ,  $x(t) \leq d \quad \forall t = 1, 2, \dots$ , i.e. the density upper bound constraint*

holds, if and only if the following equivalent condition hold:

There exist  $S \in \mathbb{R}^{m \times m}$  and  $y \in \mathbb{R}^m$  such that

$$\begin{aligned} S &\geq \mathbf{0}, \quad M + S + y\mathbf{1}^T \geq \mathbf{0}, \\ y + d &\geq (M + S + y\mathbf{1}^T) d. \end{aligned} \tag{4.8}$$

## Chapter 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 achieved 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 [2]:

$$\mathbf{1}^T(\mathbf{1} - \text{diag}(M)) \tag{5.1}$$

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

$$\begin{aligned}
& \min_{M,P,S,y} \quad \mathbf{1}^T(\mathbf{1} - \text{diag}(M)) \quad \text{s.t} \\
& \quad \mathbf{1}^T M = \mathbf{1}^T, \quad M \geq 0, \quad Mv = v \\
& \quad (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0} \\
& \quad S \geq \mathbf{0}, \quad M + S + y\mathbf{1}\mathbf{1}^T \geq 0, \\
& \quad y + d \geq (M + S + y\mathbf{1}\mathbf{1}^T)d \\
& \quad \begin{bmatrix} \lambda^2 P & (M - v\mathbf{1}\mathbf{1}^T)^T G^T \\ G(M - v\mathbf{1}\mathbf{1}^T) & G + G^T - P \end{bmatrix} \succeq 0 \\
& \quad P = P^T \succ 0, \quad G = \text{diag}(v)^{-1}
\end{aligned} \tag{5.2}$$

## 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 (5.2). 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 (4.3) for a given  $\hat{M}$ . Now given  $x(t)$ , we first solve (online) the following optimization problem:

$$\begin{aligned} \min_M \quad & \|Mx(t) - v\|_P \quad \text{subject to} \\ & M \geq 0, \quad \mathbf{1}^T M = \mathbf{1}^T, \quad (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0} \\ & x(t+1) = Mx(t), \quad x(t+1) \leq d \end{aligned} \quad (5.3)$$

Note that the above optimization problem is a quadratic-programming (QP). Consequently, solving the optimization problem described by (5.3) 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[13].

**Proposition 1.** *Suppose that there exists  $\hat{M} \in \mathcal{M}^m$  and  $P = P^T \succ \mathbf{0}$  satisfying the inequality (4.3). Then the PDC approach that utilizes the Markov Matrix  $M$  obtained by solving QP given by (5.3) results in  $\lim_{t \rightarrow \infty} x(t) = v$  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 (5.3). 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 (4.3) 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  $\lambda \in [0, 1)$ :  $\lim_{t \rightarrow \infty} \|e(t)\|_P = 0$ , which implies that  $\lim_{t \rightarrow \infty} \|e(t)\| = 0$ .  $\square$

Based on the Proposition 1, if there exist a Markov Matrix satisfying the constraints defined in (5.2), the following problem can be solved to update  $M(t)$  in each time step:

$$\begin{aligned}
&\min_M \|M(t)x(t) - v\|_P \quad \text{s.t} \\
&\mathbf{1}^T M(t) = \mathbf{1}^T, \quad M(t) \geq 0, \quad M(t)v = v \\
&(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M(t) = \mathbf{0} \\
&x(t+1) = M(t)x(t), \quad x(0) = x_0 \leq d \\
&x(t) \leq d, \quad t = 1, 2, \dots
\end{aligned} \tag{5.4}$$

### 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 5.1. 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  $n \leq m - 1$  [14].

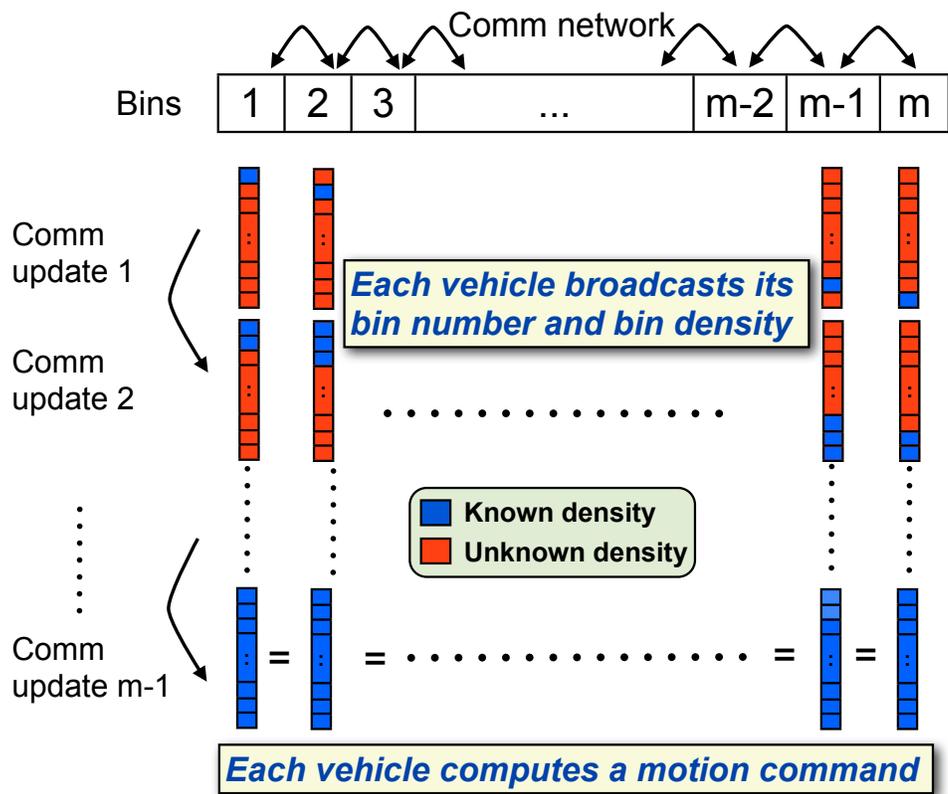


Figure 5.1: An illustration of decentralized density computation algorithm

## Chapter 6

### Numerical Example

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

$$\lambda = 0.975, \quad x(0) = [0.5 \ 0.5 \ 0 \ 0 \ 0 \ 0 \ 0]^T,$$

$$v = [0.05 \ 0.05 \ 0.05 \ 0.07 \ 0.07 \ 0.35 \ 0.36]^T \quad \text{and} \quad d = [0.5 \ 0.5 \ 0.15 \ 0.14 \ 0.13 \ 0.5 \ 0.5]^T$$

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<sup>nd</sup> bin to 5<sup>th</sup> bin). The adjacency (motion constraint) matrix

which satisfies such constraints 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 (5.2) is solved by using YALMIP [24].

Using the result of the Proposition 1, since there exists a Markov Chain Matrix satisfying the constraints of the problem (5.2), 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 6.1. The performance of the density control is 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 6.2 and the time histories of the density for the bins are shown in Figure 6.3. For the constant M case, the densities in the bins located at the bottom in Figure 6.1 ( $3^{rd}, 4^{th}$  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

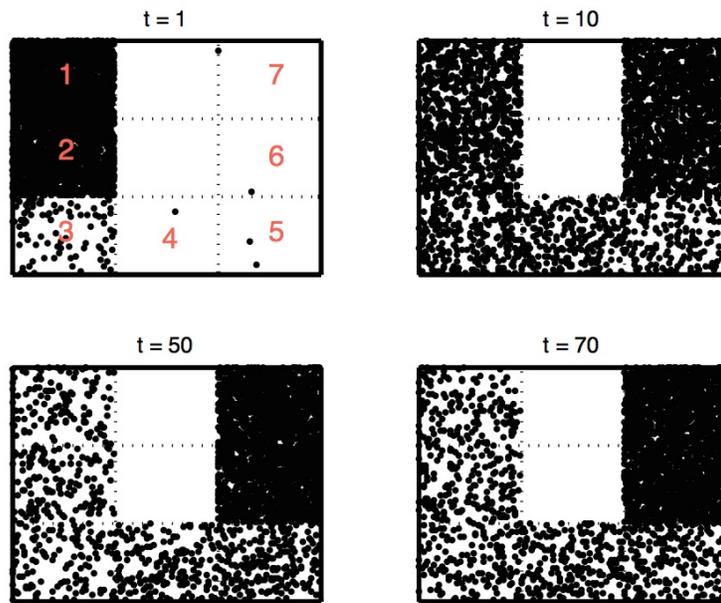


Figure 6.1: Evolution of density distribution with time.

the convergence rate drastically in both cases i.e. with and without the density constraints.

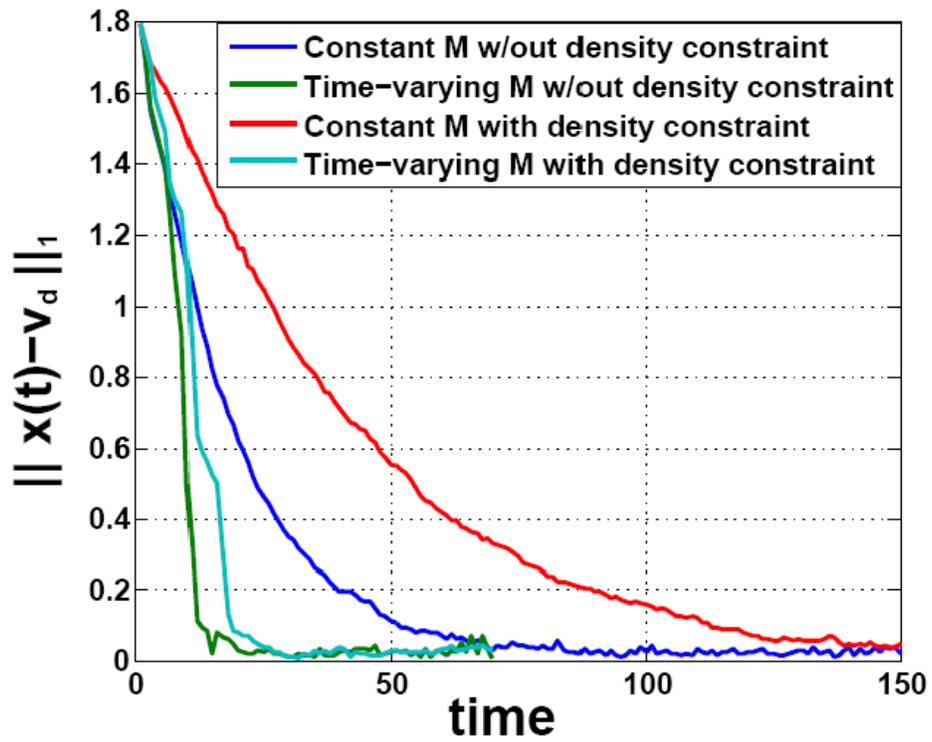


Figure 6.2: Evolution of error with time for four different cases

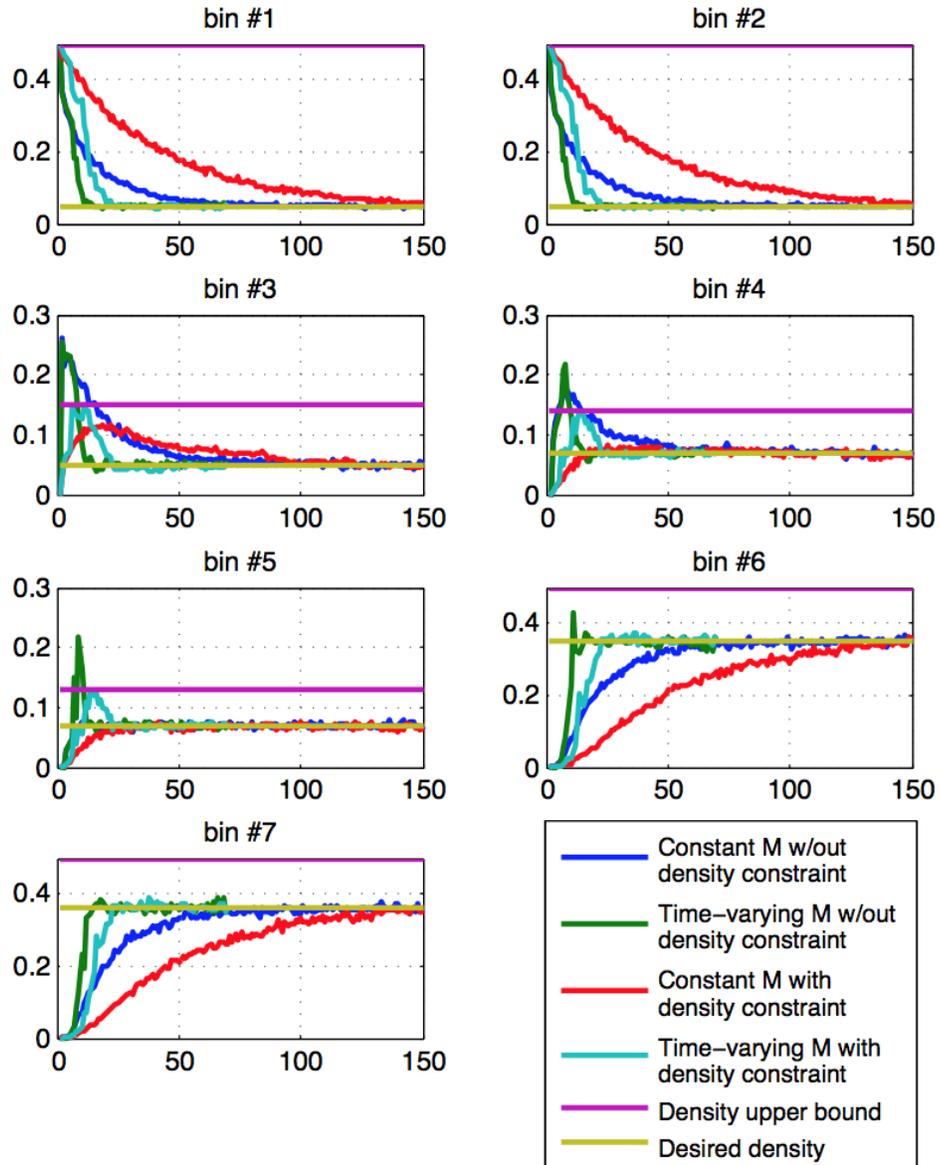


Figure 6.3: Time history of the density of each bin for four different cases.

## Chapter 7

### Conclusions

This report 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. Two approaches have been considered for the synthesis of Markov chain: i) LMI based (offline) method without density feedback; ii) Time-varying (online) method with density feedback. For both approaches, the Markov chain is synthesized as a solution of a convex optimization problem. To impose the desired constraints such as ergodicity, motion and safety constraints on the Markov chain, each constraint is expressed as a linear equality or inequality constraint on the Markov matrix. Also a decentralized counting algorithm for density estimation is introduced which guarantees finite time convergence. The results are illustrated in a numerical example for both cases. The online method with density feedback produces better convergence than the no density feedback case while putting safety constraints slows down the convergence to desired distribution.

## Bibliography

- [1] B. Açıkmeşe and D. S. Bayard. A markov chain approach to probabilistic swarm guidance. *American Control Conference, Montreal, Canada*, pages 6300–6307, 2012.
- [2] B. Açıkmeşe and D. S. Bayard. Markov chain approach to probabilistic guidance for swarms of autonomous agents. *Invited paper, accepted with a minor revision, Asian Journal of Control*, 2013.
- [3] B. Açıkmeşe and D.S. Bayard. Probabilistic swarm guidance for collaborative autonomous agents. *Revised version in review for IEEE Trans. on Automatic Control*, 2013.
- [4] B Açıkmeşe, Fred Y Hadaegh, Daniel P Scharf, and Scott R Ploen. Formulation and analysis of stability for spacecraft formations. *IET Control Theory & Applications*, 1(2):461–474, 2007.
- [5] A. Arapostathis, R. Kumar, and S. Tangirala. Controlled markov chains with safety upper bound. *IEEE Transactions on Automatic Control*, 48(7):1230–1234, 2003.
- [6] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. SIAM, 1994.

- [7] S. Boyd, P. Diaconis, and L. Xiao. Fastest mixing Markov Chain on a graph. *SIAM Review*, 46:667–689, 2004.
- [8] 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.
- [9] M. Corless and A. E. Frazho. *Linear Systems and Control: An Operator Perspective*. Marcel Dekker, 2003.
- [10] M.C. de Oliveira, J. Bernussou, and J.C. Geromel. A new discrete-time robust stability condition. *Systems and Control Letters*, pages 261–265, 1999.
- [11] N. Demir and B. Açıkmeşe. Decentralized probabilistic density control of mobile agent swarms with spatial and temporal safety constraints. *Automatica*, 2014. In review.
- [12] N. Demir, B. Açıkmeşe, and M. Harris. Convex optimization formulation of density upper bound constraints in Markov chain synthesis. *Submitted to American Control Conference*, 2014.
- [13] N. Demir, B. Açıkmeşe, and C. Pehlivanurk. Density control for decentralized autonomous agents with conflict avoidance. *To appear in IFAC World Congress, South Africa*, 2014.
- [14] N. Deo. *Graph Theory with Applications to Engineering and Computer Science*. Prentice-Hall, 1974.

- [15] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305, 1973.
- [16] M. Fiedler. *Special Matrices and Their Applications in Numerical Mathematics*. Dover, 2008.
- [17] J. Grace and J. Baillieul. Stochastic strategies for autonomous robotic surveillance. *Proc. IEEE Conf. on Decision and Control*, 2005.
- [18] F.Y. Hadaegh, B. Açıkmeşe, D.S. Bayard, G. Singh, M. Mandic, S. Chung, and D. Morgan. Guidance and control of formation flying spacecraft: From two to thousands. *Festschrift honoring John L. Junkins*, Tech Science Press, pages 327–371, 2013.
- [19] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- [20] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.
- [21] R. E. Kalman and J. E. Bertram. Control system analysis and design via the second method of Lyapunov, ii: Discrete-time systems. *Jrnl. of Basic Engineering*, 82:394–400, 1960.
- [22] 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.

- [23] M. Lawford and W. Wonham. Supervisory control of probabilistic discrete event systems. *Proc. 36th Midwest Symp. Circuits Syst.*, pages 327–331, 1993.
- [24] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.
- [25] 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.
- [26] 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 Computational Science*, (4981):358–371, 2008.
- [27] Y. Nesterov and A. Nemirovsky. *Interior-point Polynomial Methods in Convex Programming*. SIAM, 1994.
- [28] A. Papoulis. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1991.
- [29] 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.

- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] 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.