Density Control of Multi-Agent Systems with Safety Constraints:
A Markov Chain Approach

Nazlı Demirer

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2017

Committee members:

Behçet Açıkmeşe, Chair

Mehran Mesbahi

Kristi Morgansen

Maryam Fazel

Dana Dabiri

Program Authorized to Offer Degree:
Aeronautics & Astronautics

University of Washington

Abstract

Density Control of Multi-Agent Systems with Safety Constraints:
A Markov Chain Approach

Nazlı Demirer

Chair of the Supervisory Committee:
Behçet Açıkmeşe
Aeronautics and Astronautics

The control of systems with autonomous mobile agents has been a point of interest recently, with many applications like surveillance, coverage, searching over an area with probabilistic target locations or exploring an area. In all of these applications, the main goal of the swarm is to distribute itself over an operational space to achieve mission objectives specified by the density of swarm. This research focuses on the problem of controlling the distribution of multi-agent systems considering a hierarchical control structure where the whole swarm coordination is achieved at the high-level and individual vehicle/agent control is managed at the low-level. High-level coordination algorithms uses macroscopic models that describes the collective behavior of the whole swarm and specify the agent motion commands, whose execution will lead to the desired swarm behavior. The low-level control laws execute the motion to follow these commands at the agent level. The main objective of this research is to develop high-level decision control policies and algorithms to achieve physically realizable commanding of the agents by imposing mission constraints on the distribution. We also make some connections with decentralized low-level

motion control. This dissertation proposes a Markov chain based method to control the density distribution of the whole system where the implementation can be achieved in a decentralized manner with no communication between agents since establishing communication with large number of agents is highly challenging. The ultimate goal is to guide the overall density distribution of the system to a prescribed steady-state desired distribution while satisfying desired transition and safety constraints. Here, the desired distribution is determined based on the mission requirements, for example in the application of area search, the desired distribution should match closely with the probabilistic target locations. The proposed method is applicable for both systems with a single agent and systems with large number of agents due to the probabilistic nature, where the probability distribution of each agent's state evolves according to a finite-state and discrete-time Markov chain (MC). Hence, designing proper decision control policies requires numerically tractable solution methods for the synthesis of Markov chains. The synthesis problem has the form of a Linear Matrix Inequality Problem (LMI), with LMI formulation of the constraints. To this end, we propose convex necessary and sufficient conditions for safety constraints in Markov chains, which is a novel result in the Markov chain literature. In addition to LMI-based, offline, Markov matrix synthesis method, we also propose a QP-based, online, method to compute a time-varying Markov matrix based on the real-time density feedback. Both problems are convex optimization problems that can be solved in a reliable and tractable way, utilizing existing tools in the literature. A Low Earth Orbit (LEO) swarm simulations are presented to validate the effectiveness of the proposed algorithms. Another problem tackled as a part of this research is the generalization of the density control problem to autonomous mobile agents with two control modes: ON and OFF. Here, each mode consists of a (possibly overlapping) finite set of actions, that is, there exist a set of actions for the ON mode and another set for the OFF mode. We give formulation for a new Markov chain synthesis problem, with additional measurements for the state transitions, where a policy is designed to ensure desired safety and convergence properties for the underlying Markov chain.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

Table Number Page

ACKNOWLEDGMENTS

I would like to acknowledge and express my gratitude to people who have supported me throughout this journey.

First, I would like to thank my advisor, Dr. Behçet Açıkmeşe for his continuous guidance and support, and for sharing his knowledge and experience with me. His motivation and enthusiasm has been an inspiration for me in the five years we spent working together. Thanks to him, I had the opportunity to attend two great universities during my PhD. I also would like to acknowledge DARPA for the financial support and valuable feedback they provided through the course of my research.

I would like to thank my dissertation committee: Dr. Mehran Mesbahi, Dr. Kristi Morgansen, Dr. Maryam Fazel and Dr. Dana Dabiri for attending my defense and taking the time to review my dissertation. I would also like to thank Dr. Efstathios Bakolas for giving one of the most useful and informative classes on controls.

I would like to thank my friends and colleagues in Autonomous Controls Lab for their support and company. Special thanks to Utku Eren for his valuable contributions in this project and more importantly, his friendship that helped me overcome the challenges of graduate student life. Also, thanks to my dear friends Fulya and Can Çelikbilek for making my time in Seattle more enjoyable than I ever thought possible.

I would like to thank my friends Ilker Karakaşoğlu, Mustafa Kangül, Şeyma

Edikli, Nermin Uğur and Tilbe Mutluay, my cousin Burcu Kandar for their constant love and support from the other side of the world. You were always with me, through the good and the bad, even though we were thousands of miles apart.

Greatest thanks to my parents, although words cannot possibly express how grateful I am to both of them. I couldn't achieve any of this without their endless love and support. Oğuz, thanks for being the best brother anyone could ever hope for. You encourage me and make me smile everyday. Last but not least, I'd like to thank my husband Onur, for putting up with me during my PhD. You have been a great friend and an amazing husband, I am so lucky to have you by my side.

Hayattaki en büyük hazinem

Annem, babam, kardeşim ve Onur'a...

Chapter 1

INTRODUCTION

Recent years have witnessed a growing interest in the cooperative and coordinated control of multi-agent systems. It has become an interdisciplinary field with various useful applications in many areas like engineering [14, 91, 28], physics [103], biology [56] and sociology [59]. From the engineering perspective, multi-agent systems have the advantage of ability to perform in complex missions more efficiently or complete tasks that are not achievable with a single agent. This growing interest naturally leads to many technical questions and unsolved problems with different aspects. This research focuses on the problem of controlling the distribution of multi-agent systems that can be used in many applications/problems like surveillance, coverage, searching over an area with probabilistic target locations or exploring an area. In all of these applications, the main goal of the swarm is to distribute itself over an operational space to achieve mission objectives specified by the density of swarm. For example, an Earth orbiting swarm of spacecrafts or small satellites may be tasked to form a desired shape such as an antenna pattern or a mirror pattern. The swarm may also be required to switch to different patterns/configurations based on evolving mission objectives. Another interesting application is the search and rescue tasks using aerial swarm of micro-aircrafts over a region with prior information on possible target locations and risk areas. In both cases, the swarm must distribute itself to match a target distribution which can be represented by a probability density distribution. In such scenarios, the control and coordination of large number of agents cannot be managed with a centralized remote unit that commands each agent what action to pursue. This research aims to find a systematic way to solve this distribution problem in a decentralized manner while providing mathematical guarantees for the desired system behavior; specifically focusing on the following area of needs:

Decentralization: For the systems with very large number of agents, having a centralized unit that gives each agent the motion commands to achieve mission objectives is not the most effective way as establishing communication with 100s-1000s agents might become cumbersome. Hence the optimal scenario is the one where vehicle motion commands are generated locally by agents in a decentralized manner. The fact that agents have no information on the global state and on the other agents in the system makes the distribution problem very challenging. This research approaches this problem by considering the swarm as a statistical ensemble and modeling the density evolution as a Markov chain based on which agents make statistically independent decisions.

Safety: In the density control problem of the swarm, while it is desired to guide the swarm to a final density, it is also essential to avoid crowding for mitigating the possibility of conflicts/collisions between vehicles. With high-level coordination that uses the macroscopic models, collision free trajectories are very hard to generate. This is generally captured by low-level vehicle-based control policies. The proposed research provides a synthesis of Markov chain such that the safety constraints on the density are satisfied. The idea is that if the density stays below a prescribed value and the flow of agents are also bounded, then collisions/conflicts will be diminished and can be avoided with a low-level policy.

Mathematical guarantees of convergence and safety: To ensure the implementability of the proposed method and algorithm, mathematical results must be constructed to guarantee the overall convergence and safety as discussed in the preceding paragraph. This research provides theoretical results for convex formulations of these constraints on the Markov chain using Lyapunov theory and duality theory of convex optimization. Formulation of the necessary and sufficient convex conditions for density safety constraints in Markov chains is one of the main results of the research and it is a novel contribution to Markov chain literature.

Improvement with density feedback: The proposed method does not assume any feedback on the overall density of the multi-agent system. The agents are assumed to know their states and make their decisions solely based on their states. A natural question comes with this: Could any feedback on the density be utilized for a better performance? The answer is yes and a method for adapting the Markov chain with density feedback for better convergence is proposed. Overall, we consider two methods: (i) Offline Markov chain synthesis when there is no feedback; (ii) Online adaptive Markov chain synthesis with feedback on the density. We also propose a decentralized density estimation for feedback which is based on the assumption that agents can communicate locally.

Control of ON/OFF agents: Another interesting problem in multi-agent framework stems from the limitations based on the vehicles such as ability to go in only certain directions at a given state. So, the actions that a vehicle can take may be limited due to limited actuation of the vehicle. In such cases, desired trajectory or vehicle motion commands might be infeasible, hence the capabilities of the vehicle must be taken into account. Also, in many of real-world decision making problems an action does not necessarily have a deterministic outcome. In that sense, this research introduces the concept of ON/OFF agents that captures these problems. ON/OFF agents have binary control modes, and these modes can include one or many actions selected from a discrete action space while each action has corresponding transitional probabilities.

1.1 Methodology

There are two main approaches for the modeling of the swarm systems: (i) Microscopic point of view considering each agent as a fundamental unit of the system and modeling the behavior of each agent and interaction between the agents; (ii) Macroscopic point of view considering a high-level abstraction of the system and describing the collective behavior of the whole swarm rather than modeling agents individually. As the number of agents increases in the system, the complexity of the microscopic approach grows, and stability and convergence characteristics of the system cannot be established in an analytically verifiable manner. In this regard, macroscopic models are relatively easier to construct for the systems with large number of agents; furthermore, stochastic approaches based on macroscopic models offers a way to achieve coordination in a decentralized manner with no communication between agents. These two different level of abstractions of the system models can be also utilized in a hierarchical control structure (see Figure 1.1). Here the high-level swarm coordination is distinguished from the low-level vehicle/agent control as follows: High-level coordination algorithms uses macroscopic models and specify the agent motion commands, whose execution will lead to the desired swarm behavior, and the low-level control laws execute the motion to follow these commands.

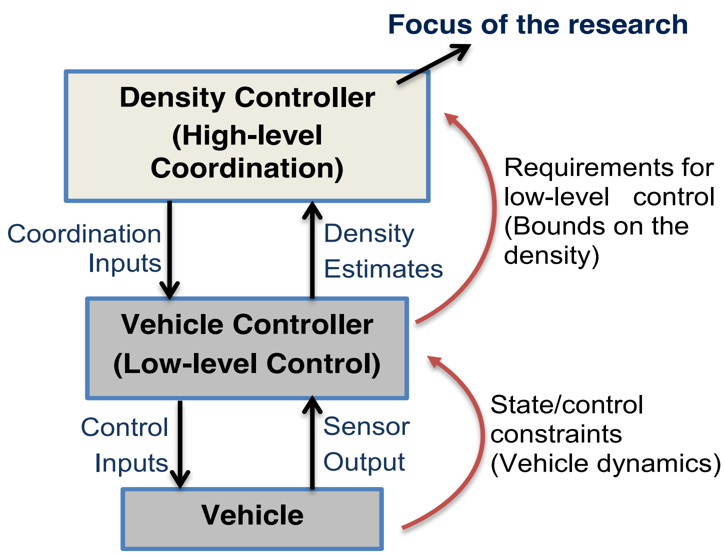

Figure 1.1: A hierarchical swarm density coordination and control architecture: High-level Coordination provides motion commands(coordination inputs) for the vehicle controller and low-level control generates motion plans (control inputs) to follow these commands.

The main objective of the proposed research is to design decision control policies for multi-agent systems that are implementable in a decentralized manner with no communication between agents, to control the density distribution of the whole system. As the mentioned above in the context of decentralization, establishing communication between each agent with large number agents is highly challenging, so we first consider the case when there is no communication. Then, extending on this case, we investigate how to enhance performance when the agents can communicate locally. The objective is to guide the overall density distribution of the system to a prescribed steady-state desired distribution while satisfying some transition and safety constraints. The proposed work adopts the viewpoint presented in [3] that considers the swarm as a statistical ensemble for which guidance can be performed from a probabilistic point of view. The density distribution of the multi-agent system is described as a discrete probability distribution over a discrete, finite state-space, and the time evolution of the probabilistic density distribution is specified using a Markov chain. A Markov chain can be represented with a column stochastic matrix, that will be referred to as a *Markov matrix*, with each entry of the matrix giving the probability of transition between each two states over the discrete state-space. Since the density distribution can be interpreted as the temporal probability distribution of the state of a single agent, or the state probability distribution over multiple agents, the proposed method is applicable for systems with single and multiple agents. Another aspect of the probabilistic interpretation is the need for a large number of agents to make the density evolve exactly as the Markov chain; with a finite number of agents in a swarm, the time evolution of the ensemble density is always an approximation. Hence, the actual distribution will generally be different from the probability distribution, but it will be equal to probability distribution on average, meaning that the desired distribution and the satisfaction of constraints can be violated by a small number of agents. The corresponding error can be made arbitrarily small by using sufficiently large number of agents. Also, in the numerical simulations presented here, the state is considered to be based on the position of an agent in the operational space as it makes the distribution problem more intuitive. In general, the agent state does not have to be the agent position, e.g., it can be the agent's velocity, temperature, etc. The results presented here are applicable for the general state definition over a discrete state-space.

As mentioned above, the probability distribution of each agent's state evolves according to a finite-state and discrete-time Markov chain (MC), hence, designing proper decision control policies requires numerically tractable solution methods for the synthesis of Markov chains that will guide the overall density distribution of the system. The proposed approach controls the overall density distribution directly by first synthesizing a Markov Chain that is used locally to determine the agent motion commands as a function of their state [34, 32, 31]. To this end, the decision control policy synthesis problem is formulated as a Markov chain synthesis problem, while all desired constraints are imposed on the Markov matrix. The synthesis problem has the form of a Linear Matrix Inequality Problem (LMI), with LMI formulation of the constraints. LMI problems, also referred to as Semidefinite Programming problems, are convex optimization problems that can be solved numerically to global optimality in polynomial time [19, 88]. Hence, formulating Markov chain design constraints as LMIs makes the synthesis problem numerically tractable. The main classes of constraints we consider for this problem are: (i) ergodicity constraints; (ii) motion constraints; and (iii) safety constraints. The ergodicity constraints are imposed to ensure that the distribution of agent states converges to a desired prescribed probability distribution. The motion constraints are imposed to prevent undesirable transitions between states by limiting the set of states reachable from each state, i.e., an agent in a certain state can only transition to a subset of states. The safety constraints are considered in a general linear form that encompasses a number of important constraints. For example, probability of being in a state can be bounded by a prescribed quantity. When the agent state is defined based on position, this constraint corresponds to limiting the expected number of agents in an area, hence mitigating possible collisions/conflicts. Among the three design constraints, the characterization of a general class of safety constraints is novel and the most challenging.

This research also considers the Markov chain synthesis problem when there is feedback on the density and proposes a Quadratic Programming formulation for this problem. The idea is that if there is feedback on the current density distribution of the system through measurement or estimation, then this information can be used to update the Markov chain matrix for better convergence and resource consumption. Adaptation of Markov chain synthesis with real-time information also yields robustness to changes of mission constraints. For implementation of adaptive density control, in addition to an online synthesis method for Markov chains, there is a need for a decentralized method to estimate the density distribution of the system. For this purpose, we propose a decentralized counting algorithm to estimate the density of the system.

In this research, we also consider the density control problem for the agents with limited mobility or actuation, in which case only the actions that are attainable by the agents must be considered. We model this problem by considering two control modes: ON and OFF. Here, each mode consists of a (possibly overlapping) finite set of actions, that is, there exist a set of actions for the ON mode and another set for the OFF mode, which may have a non-empty intersection. At each time step, an agent is allowed to measure (observe) the instantaneous outcome, i.e. transition, for a single action it chooses among the set of actions for the ON mode (the actions that can be taken while in ON mode). The objective is to design a decision making policy based on which the agents decide whether to be ON or OFF and which action to choose so that the density distribution of the swarm converges to a desired final density while satisfying transition and safety constraints described above. In our model, the decision policy can be constructed in such a way that more favorable actions have higher probability of being selected. Overall, this problem is inherently more challenging than the standard density control problem for agents with full mobility, as the actions that an agent can take are limited.

We consider a discrete action space and transitional probabilities on a discretized finite state-space corresponding to each action, hence there is an inherent connection to Markov Decision Processes (MDPs) [93]. MDPs are discrete time stochastic control processes with the set of available actions, the rewards, and the transition probabilities between states. Our model is fundamentally different from the standard MDP models, since the mission objectives here are embedded within the underlying Markov chain, rather than having a reward function for each action and transition.

In our decision making model, the discrete probability distribution for the agent's state evolves according to a discrete-time Markov chain that is a linear function of the stochastic environment (i.e., transition probabilities) and the ON/OFF decision policy. This linear relation indicates that, all the problem constraints which are formulated as convex constraints on the Markov matrix are also convex on the ON/OFF decision policy. Hence, the LMI formulation for Markov chain synthesis is directly applicable to ON/OFF decision policy synthesis problem. The main challenge here is the modeling of the system. As an initial approach to this problem, the case where ON mode has a single action and OFF mode has deterministic outcome, is considered [33]. In this case, the agent either conforms to the motion induced by the environment or it remains motionless. This problem is further extended to the case where the ON mode captures multiple actions and OFF mode has stochastic transitions

[35]. Similarly, overall evolution of the density is formulated as a Markov chain that is a function of stochastic environment and ON/OFF decision policy. It is noteworthy that the OFF mode captures multiple interesting scenarios: (i) There is only a single action for the OFF mode for which the outcome, i.e., resulting transition, is not observable; (ii) There are multiple actions for the OFF mode without outcome observations and a standard decision policy running over these actions, which results in an effective transition matrix for the underlying Markov chain; iii) Transitions can be observed for all available actions and there is a standard decision policy that can run over the actions without requiring transition observations, which is utilized as the default policy for the OFF mode when the observed action is rejected. Note that in the third interpretation, the set of actions for the ON mode and the OFF mode are identical. Hence, a rejected observed action at some instant may be still chosen, i.e., the rejected observed action for the ON mode can also be the resulting action for the OFF policy. When implementing this method, the decision policies are computed offline by solving an LMI problem and are given to the agent assuming that at each time step, the current state can be observed, one-step outcome of a single action for the ON mode can be measured, and the transition corresponding to the selected action can be accepted or rejected. In this sense ON and OFF modes can be seen as higher level actions, under which there are the lower level motion actions.

The concept of ON/OFF control is naturally inspired by the Metropolis-Hastings (M-H) Algorithm [85, 18], where the notion of accepting or rejecting a proposed transition is emphasized. Hence, the proposed method has some connections to M-H Algorithm. We show that in the absence of motion and safety constraints the decision policy synthesis problem for the single "ON" case is equivalent to designing the *acceptance matrix* in the M-H Algorithm, when the reversibility condition is imposed. This is done by novel usage of an algebraic operator on nonnegative matrices.

Assumptions:

Throughout the dissertation, it is assumed that the agents are able to determine their own states at any given time, as the decision policy depends solely on the state. In general, there is no assumption on the sensing and communication capabilities on the agents since the algorithms proposed here does not require communication between agents. One exception is the decentralized counting algorithm which runs based on the local density feedback. Additional assumptions on the communication capabilities of the agents for this case is provided in Section 3.3. For the implementation of the decision policy, it is also assumed that agents have the processing capability to generate a random number.

1.2 Related Work

1.2.1 Markov chains

Here, as the main objective is to obtain numerically tractable solution methods for the synthesis of Markov chains [73, 86], we formulate the problem constraints for Markov chain synthesis as equivalent convex constraints, specifically as equivalent linear inequality constraints. Markov chain synthesis has been of interest in many areas of science and engineering in the context of Monte Carlo Markov Chain (MCMC) methods [11, 37, 51, 53, 58, 75, 95]. Specifically, safety constraints in Markov chain are first considered in [6, 64] that proposes set of equivalent conditions for upper bound constraints on the states. The result presented here is a useful analysis result to check whether a given Markov chain satisfies the density upper bound constraints or not. Given this result, there's still a need for a *synthesis* condition that is numerically tractable. The dissertation proposes necessary and sufficient conditions for the satisfaction of a general class of linear safety constraints on the density that captures various types of constraints such as density upper bound, density flow, etc.

Markov chains are also widely studied in the framework of nonnegative matrices [13]. A well-known fundamental result which demonstrates necessary and sufficient conditions for Markov chains to have a unique steady-state distribution is shown via Perron-Frobenius Theory of nonnegative matrices [67]. The problems of guaranteeing the convergence and maximizing the convergence rate are studied through the second largest eigenvalue (SLE) of the Markov chain transition matrix [99]. Geometric bounds for the SLE and spectral gap of a reversible Markov chain are developed in [39]. The results are extended for the eigenvalue bounds of nonreversible chains in [50]. This research uses more recent studies [3, 19] which present equivalent Linear-Matrix Inequality (LMI) conditions for the convergence of the Markov matrix, i.e., convex conditions for the bound on the SLE of Markov matrix. Similar problems have also been extensively studied in the design of networked systems, where the connectivity of the underlying graph is measured via SLE of the system Laplacian. Reference [106] considers the distributed averaging protocol over a network and proposes on SDP formulation to optimize the communication weights for the fastest convergence. A complementary problem of determining the topology of the network has been studied in [29], where the authors categorized the topology design problem into three categories and formulated the corresponding optimization formulation for each case.

We also extend the density control problem to agents with limited mobility, which we refer to as ON/OFF agents. As discussed earlier, the method we proposed has some connections to Metropolis-Hastings Algorithm. Relevant work has appeared in the literature in the context of "Multiple try Metropolis-Hastings Algorithm" where multiple states are proposed at each time epoch. This generalization of standard M-H Algorithm is first presented in [76] with multiple proposal states from the same probability distribution. Then, this algorithm is further generalized in [23] for multiple proposal states from different distributions.

1.2.2 Multi-agent systems

There are not many examples in the literature on the use of Markov chains for guiding large numbers of agents to a desired distribution since it's a relatively new idea, first proposed by [3]. An inhomogeneous Markov chain method is proposed in [10] where the number of transitions is minimized to achieve and maintain formations. A similar problem in the task allocation framework is considered in [14], where probability of switching between tasks is designed to achieve maximum redistribution rate, without any additional constraints. A different Markov chain based method is proposed in [25] by using a probabilistic disablement approach [72] which enables or disables transitions with certain probabilities. They propose an algorithm that computes the necessary perturbations in the local agents behavior, which guarantees convergence to the desired observed state of the swarm. An algorithm for swarm of bots is proposed in [60] to find peaks in a search space using birth/death Markov chains without the individual bots knowing their position in the space and without requiring communication among bots. Their analysis shows probabilistically that the agents congregate or cluster near the peaks in the measurement function. Similar problem is considered in [84] which provides a method to find the maximum of a scalar function defined over a region of interest. It uses biased random walk inspired by bacterial chemotaxis, which leads vehicle positions to evolve toward a probability density that is a specified function of the spatial profile of the measured signal. Gibbs sampling-based algorithm is another stochastic method used for coordination of multi-agent systems, [101] provides analysis for the parallel Gibbs sampling-based coordination algorithm where the agents take their moves by sampling in parallel their locally perceived Gibbs distributions corresponding to a pairwise, nearest-neighbor potential. The research presented here differs from the listed references as it considers additional safety constraints imposed on the Markov chain.

A different approach to swarm formation is considering the swarm as a continuum and modeling the system using partial differential equations (PDEs). A method to control the probability density distribution of a swarm via velocity fields is proposed in [47] where each agent calculates the desired velocity locally by utilizing the number of agents within a prescribed communication distance. Here, the swarm probability density distribution is propagated with the heat equation. The work presented in this paper is a great fit for the low-level control scheme introduced in the Methodology section. We show some preliminary results on the velocity field generation here and also implement in the simulations. [52] presents a distributed optimal control problem formulation where optimality conditions are derived analytically for the case in which the macroscopic description is characterized by the agent distribution, and the macroscopic dynamics are modeled by the advection equation. A deployment problem for large scale system of agents is considered in [94] which models the agents' collective dynamics by complex-valued reaction-diffusion 2-D partial differential equations (PDEs) in polar coordinates. A diffusion-based approach to achieve a spatial distribution of swarm activity is proposed by [45] where the agents have only local sensing and take local measurements of an underlying scalar field. A distributed control approach is considered in [70] where the system is modeled using PDEs. Here, agents in the swarm do not have access the position information, the authors propose a Laplacian-based distributed algorithm which the agents implement to localize themselves in a new coordinate frame. These approaches offer smooth evolution of the distribution and the handling of agent interactions; however, it has difficulty of imposing higher level constraints on the density. PDE-based approaches to multi-agent coordination also uses Smoothed Particle Hydrodynamics (SPH) technique [87] which is a mesh-free numerical method used to simulate flow fields. In the multi-agent framework, SPH methods consider each agent as particle where global properties of the system is determined by looking at the contributions of each particle to a property based on their distance known as the smoothing length or interpolation length. SPH in swarm coordination is used in [92] to obtain decentralized controllers that force the robots to behave like fluid particles. An approach to design decentralized control policies is proposed in [15] for inhomogeneous robotic swarm where the swarm behavior is modeled as an advection-diffusion-reaction PDE, which is solved using SPH. The SPH method has various advantages. It is based on local information in the neighborhood of agents, hence it's decentralized and scalable. It can also handle moving boundaries and large deformations. However, the biggest challenge and ongoing research direction in SPH methods is the selection of interpolation length which determines the effect of each agent on the collective behavior. It is also challenging to impose higher level constraints on the global behavior of the system.

Another method for multi-agent coordination is to use nearest neighbor information to establish consensus [17, 65, 80, 96, 98]. [28] proposes a method based on locational optimization and centroidal Voronoi diagrams, while [91, 8] provide adaptive methods of multi-agent coordination. Other stochastic approaches to multi-agent control include gradient-based decentralized controllers [63] utilizing relative positions to neighbors; game theoretical formulations [7], where each vehicle is considered as a self-interested player; maze searching techniques [78] and cyclic pursuit strategies [90], where each agent pursues its leading neighbor resulting in convergence to a prescribed geometric pattern. In [62], a quorum based method is proposed where the multi-agent system is modeled as a hybrid system. Reference [81] presents a comprehensive survey of recently developed theoretical tools for modeling, analysis, and design of motion coordination algorithms.

The problem of guiding a multi-agent system has been also a recent subject of research in Markov Decision Processes (MDPs) [44, 100, 108]. The ON/OFF control problem presented here can also be considered as an MDP [93] with finite number of states and actions, and a new set of observations. However, rather than having a reward function for each action and transition [43], the mission objectives here are embedded within the underlying Markov chain.

With the utilization of MDPs in multi-agent system guidance, the problems regarding the safety constraints in MDPs have gained more interest. [108] considers a decision-making problem for heterogeneous multi-agent systems with safety density constraints where the system is modeled as a state constrained MDP. A risk-aware path planning problem is considered in [48] which proposes a hierarchical method to manage the computational complexity in constrained MDPs [9]. In contrast to these approaches, we utilize additional measurements on the action/transition pairs and impose hard safety constraints on the states directly on the underlying Markov chain. The proposed model and the MDP framework can be seen as two complimentary models. Our model builds on MDPs and expands MDP formulations further with new type of measurements. We also consider constraints that cannot be captured by the classical framework of MDPs as also shown in [24].

Another direction of research in the area of MDPs and multi-agent systems focuses on the decentralized control of Markov decision processes (Dec-MDP) and decentralized partially-observed Markov decision processes (Dec-POMDP), e.g., [5], where each agent has partial or incomplete observations of its state. These problems, generally, are very difficult to solve and become intractable for large scale problems [16]. These problems with incomplete observations are quite interesting, but not directly comparable to the problems considered in this research. An interesting application, which can also be explored in the proposed ON/OFF framework, is the control of balloons in stochastic wind fields for atmospheric science observations [46]. Earlier work converted this motion planning problem into a more standard Markov decision model [71, 105]. The main distinction of this work from the listed references above is the existence of the ON control mode and its observed actions. This allows us to devise new methods to control the density distribution of autonomous agents via a new Markov decision model with measurements on the state transitions. Measurements for the ON mode can be obtained by the deployment of additional sensors to extend the agents' sensing capabilities.

1.3 Contributions of the Dissertation

1.4 Thesis Outline

Following gives a brief summary of each chapter of this dissertation. The work presented here have been published in [2, 34, 35, 33, 31, 32]. More detailed information is provided in at the beginning of each chapter.

Chapter 2

In this chapter, we introduce the probabilistic density control problem and define the problem constraints. Then, we give an overview on the fundamental results on the Markov chains to constitute basis for establishing the main technical results of the research. We also give the basic definitions and notations that are used throughout the dissertation.After giving a brief background on Markov chains, we present the probabilistic density control algorithm to demonstrate how Markov chain based policy is implemented at the agent level to control the density distribution of the system.

Chapter 3

This chapter provides more detailed information on the problem constraints and gives their convex formulations that are imposed on the Markov matrix. One of the main results of the dissertation is presented in Theorem 1 that establishes convex necessary and sufficient conditions on density safety constraints on the Markov chains. Then, a Linear Matrix Inequality (LMI) Optimization problem, that is solved offline, is constructed for synthesis of Markov chain with all relevant constraints. Finally, we propose a new Quadratic Programming (QP) based method that computes the Markov matrix online, based on density feedback. For this purpose, a decentralized method to compute the density distribution is also proposed. Finally, we present 2D numerical simulations to demonstrate the technical results of the chapter.

Chapter 4

This chapter presents numerical simulations for probabilistic density control method applied to a Low-earth orbiting swarm in 2D. The probabilistic guidance algorithm is adapted to earth orbiting swarms using Hill's equations for the relative dynamics of each spacecraft to the circular orbit defined in local-vertical local horizontal (LVLH) coordinates. A brief background on the relative dynamics in a circular orbit is given and, the simulation parameters and their selection are explained. Finally, simulation results are given in the figures and they're explained in detail.

Chapter 5

This section extends the probabilistic density control problem defined in Chapter 3 to swarms of autonomous ON/OFF agents by proposing a new Markov decision model. First, the model is constructed for a simple case where ON mode has a single action and OFF mode is deterministic. Then the result is generalized to the case where ON mode has multiple actions and OFF mode has stochastic transitions. Numerical simulations are presented for both cases. Also, connection of the proposed model to Markov decision processes are discussed.

Chapter 6

This chapter gives the brief summary on the contributions of the dissertation and it discusses the open research questions.

Numerical Simulations

Throughout the dissertation, we provide numerical simulations of 2D multi-agent systems at the end of each main section to demonstrate the performance and effectiveness of the proposed method. In all numerical simulations, the agents' state is defined based on their position in a prescribed region that is partitioned to subregions. Hence, the density distribution corresponds to fraction of expected number of agents in each subregion. In Chapter 3, after proposing an offline LMI synthesis method and an online QP synthesis method with feedback, we provide multi-agent simulations for both cases with different types of constraints introduced in this chapter. In chapter 5, we provide numerical simulations for both models and demonstrate the effect and performance of ON/OFF decision control policy by comparing to uncontrolled cases. In addition to above mentioned examples, higher-fidelity simulations are also conducted to verify the algorithms' reliability and to reveal the resource requirements for a swarm, which is covered in Chapter 4. Here, the probabilistic density control method is applied to a Low Earth Orbit swarm in 2D. The idea is adopted from ([57]) where the probabilistic guidance algorithm is adapted to earth orbiting swarms using Hill's equations for the relative dynamics of each spacecraft to the circular orbit defined in local-vertical local horizontal (LVLH) coordinates.

NOTATION

The following is a partial list of notation used:

- $\mathbf{0}$ ($\mathbf{1}$) is the vector/matrix of zeros (ones) with appropriate dimensions;
- I is the identity matrix;
- e_i is a vector of appropriate dimension with its i^{th} entry $+1$ and its other entries zero;
- $x[i] = \mathrm{e}_i^T x$ 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;
- $\mathcal{P}(A)$ denotes probability of event A;
- $\mathbb{E}(X)$ denoted the expected value of the random variable X;
- $\mathbb{R}^n$ is the n dimensional real vector space;
- $\mathbb{N}$ is set of nonnegative integers, i.e., $\mathbb{N} = \{0, 1, 2, \ldots\}$;
- $\mathbb{N}^+$ is set of positive integers, i.e., $\mathbb{N}^+ = \{1, 2, \ldots\}$, $\mathbb{N}_n^+ = \{1, 2, \ldots n\}$;
- $\emptyset$ denotes the empty set;

- $\|v\|$ is the 2-norm of the vector v;
- For $P = P^T \succ 0$, $\|v\|_P = \|P^{1/2}v\|$ where $P = P^{1/2}P^{1/2}$ is a factorization of P with $P^{1/2} = {P^{1/2}}^T \succ 0$ (for concreteness, one choice is $P^{1/2} = U\Lambda^{1/2}U^T$ where $P = U\Lambda U^T$ is an eigenvector decomposition of P);
- $(v_1, v_2, ..., v_n)$ represents a vector obtained by concentrating vectors $v_1, \ldots, v_n$ such that $(v_1, v_2, ..., v_n) \equiv \begin{bmatrix} v_1^T & v_2^T & \ldots & v_n^T \end{bmatrix}^T$ where v_i can have arbitrary dimensions;
- $\text{diag}(A) = (A[1,1], \ldots, A[n,n])$ for matrix A; $\text{diag}(v)$ is a square diagonal matrix with the elements of vector v on the main diagonals;
- $\otimes$ denotes the Kronecker product;
- $\odot$ represents the Hadamard (Schur) product; $\texttt{i}(A)$ is the indicator matrix for any matrix A, whose entries are given by $\texttt{i}(A)[i,j] = 1$ if $A[i,j] \neq 0$ and $\texttt{i}(A)[i,j] = 0$ otherwise;
- $\eta \sim U(0,1)$ denotes a random variable sampled from the uniform distribution in the interval $[0,1]$.
- $\lambda_{max}(P)$ and $\lambda_{min}(P)$ are maximum and minimum eigenvalues of $P = P^T$;
- $\sigma(A)$ is the spectrum (set of eigenvalues) of A;
- $\rho(A)$ is the spectral radius of A
- $(\max_{\lambda \in \sigma(A)} |\lambda|)$; $||A||_1 = \sum_i \sum_j |A[i,j]|$ denotes the 1-norm of matrix A, and $|||A|||_1 = \max_j \sum_i |A[i,j]|$ denotes the induced 1-norm of matrix A;

Chapter 2

BACKGROUND

This section provides a brief background on stochastic processes and the Markov chains. The dissertation presents the background from a linear algebra perspective since the solution of the density control problem is achieved using Linear Matrix Inequalities (LMI). As mentioned in the introduction, to address the multi-agent distribution problem, we consider the multi-agent system as a statistical ensemble and model the evolution of this distribution as a discrete stochastic process, i.e., Markov chain. In this section, we explain the basic concepts and notations which are used throughout the dissertation. We also give some of the fundamental results on Markov chains that will help establish the main results of the research and give more insight on the Markov chains. Readers may also refer to references [73, 86] for more detailed information and further results on Markov chains.

The idea of modeling the density evolution of multi-agent system as a Markov chain is adopted from [3] where the authors also review useful results from Perron-Frobenius theory, and present a generalization of primitivity condition for asymptotic convergence of Markov chains.

Next, we introduce the probabilistic density control problem and define the problem constraints briefly. Then, after giving an overview on Markov chains, we present the density control algorithm to show how Markov chain based policy is implemented at the agent level to control the density distribution of the system.

2.1 Density Control Problem

The density control problem can be defined as follows: Given any initial distribution $x(0) \in \mathbb{P}^n$, it is desired to guide the agents toward a specified steady-state distribution $v \in \mathbb{P}^n$, while satisfying desired transition and safety constraints. Here, transition constraints guarantee that undesirable transitions between states are prohibited and safety constraints capture a general class of linear state constraints that bounds the density for each state, rate of change of density, etc. More detailed discussion on these constraints are given in the next chapter along with their convex formulations.

For the problem defined above, the density, as defined in (2.4), should satisfy the following constraints for $t \in \mathbb{N}$:

$$\mathcal{P}\{s(t+1)=s_i|s(t)=s_i\}=0 \text{ if } A_a[j,i]=0, \quad \text{(Transition)} \qquad (2.1)$$

$$Lx(t) \leq p, \quad \forall x(t) \leq q, \qquad \text{(Safety)} \qquad (2.2)$$

$$Mv = v, \lim_{t\to\infty} x(t) = v, \forall x(0) \in \mathbb{P}^n, \qquad \text{(Convergence)} \qquad (2.3)$$

where v is a desired, discrete, probability distribution, and A_a is the adjacency matrix which is a square matrix used to represent the graph structure. The elements of the matrix indicate whether pairs of states (vertices) are connected (adjacent) or not in the graph. Here, it defines the allowable transitions between states between two consecutive time steps. L, q, and p are given matrices and arrays specifying safety constraints. In order to formulate this problem as a Markov chain synthesis problem, we will formulate all the constraints as convex constraints on M and construct a convex optimization problem with solution variable M in the next chapter.

2.2 Discrete-time Markov Chains

We consider a finite set of states $\mathcal{S} = \{s_1, \ldots, s_n\}$, that is, there is a finite-dimensional state space with cardinality n and s_j is referred to as "j^{th} state". Here, $\mathcal{S}$ is the set of possible values of the state. The elements of $\mathcal{S}$ can also be defined as integers, i.e., $\mathcal{S} = \{1, \ldots, n\}$, where each integer corresponds to a state in the state-space. In the scope of this research, the state is considered as a function of the position of an agent in the operational space as it makes the distribution problem more intuitive. In general, the agent state does not have to be the agent position, e.g., it can be the agent's velocity, temperature, etc. The results presented here are applicable for the general state definition over a discrete state-space. Figure 2.1 shows an example discretization over a region where the state is defined based on the postion. Here, each small subregion is a state and defined with an integer, i.e., $\mathcal{S} = \{1, \ldots, 9\}$.

Figure 2.1: Example discretization of a state-space

Given the set of states $\mathcal{S}$, a stochastic process is defined as a sequence of random variables $\{X_n\}$ that takes values in $\mathcal{S}$. In general, describing the evolution of a stochastic process, i.e. computing the joint distribution of $(X_0, X_1, ..., X_n)$, $\forall n \in \mathbb{N}$ is very difficult since the complexity grows exponentially [86]. In that sense, Markov chains, a subclass of stochastic processes, offer a substantial reduction in complexity by introducing a *memoryless property*. Following gives a more formal definition of a Markov chain.

Definition 1 *A Markov chain is a discrete-time stochastic process such that each random variable takes values in a discrete set and for all states the conditional probability distribution of the future states of the process (conditional on both past and present states) depends only on the present state (i.e. Markovian property).*

Hence, controlling the overall density distribution of the agents by using a Markov chain allows to determine agent motion commands based on solely their states at a given time. This enables the multi-agent system to behave in a completely decentralized manner. Each agent makes independent decisions to go to the next state based on a Markov chain, which result in the desired collective behavior/motion. Here, any known dynamics of the agents and limitations are incorporated by imposing constraints on the allowable transitions between states as we will discuss in more detail in the next chapter.

The overall density distribution of the agent states is described as a discrete probability distribution over a discretized state-space. $s(t) \in \mathcal{S}$ is the state of the agent at time t and '$s(t) = s_i$' is the event that the state is the i^{th} state at time t. Then, the probability density distribution $x(t) \in \mathbb{P}^n$ is defined as

$$x[i](t) = \mathcal{P}\{s(t) = s_i\}, \tag{2.4}$$

where t is the discrete time index and $x[i](t)$ is the probability of an agent to be in i^{th} state. Here, $x(t) \in \mathbb{P}^n$ is also referred to as probability vector and satisfies $\mathbf{1}^T x(t) = 1$ as the sum of probabilities of being in each state is 1.

Now, we will give a more mathematical definition of the Markov chain and the transition matrix. A Markov chain can be represented with three parameters $\mathcal{S}$, M and $x(0)$ [6]. Here, M is the the state transition matrix, which will be referred to as *Markov matrix*, whose entries are defined as the transition probabilities; i.e.,

$$M[i,j] = \mathcal{P}\{s(t+1) = s_i | s(t) = s_j\}, \tag{2.5}$$

and $x(0)$ is a probability vector giving the initial state probability distribution, where $x(0)[i]$ gives the probability of the i^{th} initial state. Since $M[i,j]$ gives the probability of transition from state j to state i, Markov matrix M determines the time evolution of the probability distribution as

$$x(t+1) = Mx(t), \quad t \in \mathbb{N}. \tag{2.6}$$

If the transitional probabilities do not depend on time k, the Markov chain is called *stationary* or *homogeneous*. In the general Markov chain literature, the probability distribution is generally defined as a row vector and the transition matrix is defined as the transpose of M, i.e, standard evolution of density is shown as $\tilde{x}(t+1) = \tilde{x}(t)P$ where $\tilde{x} = x^T$ and $P = M^T$. At this point, we diverge from the standard notation and define the system (2.6) in the form of a linear system to readily use standard tools of linear algebra and optimization.

The well-established conditions that a Markov matrix should satisfy can be given as follows:

$$M \geq 0, \quad \mathbf{1}^T M = \mathbf{1}^T \tag{2.7}$$

Here the first property states that each entry of the Markov matrix must be nonnegative as it corresponds to a probabilistic quantity. The second property means that M is a column stochastic matrix and ensures that the sum of probabilities of transitions from state j to all the states in $\mathcal{S}$ is 1. Such matrices can be characterized as follows.

Definition 2 *Matrix* $M \in \mathbb{R}^{n\times n}$ *is a* Markov matrix, $M \in \mathbb{P}^{n\times n}$, *if* $M \geq 0$ *and* $\mathbf{1}^T M = \mathbf{1}^T$.

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

$$\mathbf{1}^T x(0) = 1 \quad \textit{and} \quad \mathbf{1}^T M = \mathbf{1}^T,$$

which implies that

$$\mathbf{1}^T M^t x(0) = \mathbf{1}^T M^{(t-1)} x(0) = \ldots = \mathbf{1}^T x(0) = 1.$$

where t is the discrete time index and M^t is the t^{th} power of M.

Definition 3 *Consider the Markov chain* M, *as given in (2.6). A distribution* $v \in \mathbb{P}^n$ *is called a* stationary distribution *of* M *if it satisfies the equation*

$$Mv = v. \tag{2.8}$$

Clearly if v is the stationary distribution of M, then

$$x(t) = v, \quad \forall t = t_o, t_o+1, \ldots, \text{ if } x(t_o) = v,\ t_o \in \mathbb{N}^+.$$

which follows from (2.6) and (2.8). A matrix $M \in \mathbb{P}^{n\times n}$ that satisfies the constraint (2.8) is a column stochastic matrix where v is the eigenvector corresponding to eigenvalue 1 [61, 49].

Definition 4 *[73] A Markov chain M is called irreducible if for any two states $i, j \in \mathbb{N}_n^+$ there exists an integer t such that $M^t(i,j) > 0$.*

The above definition implies that a Markov chain is irreducible if it is possible to get from any state to any other state using only transitions of positive probability. The notion of irreducibility given here directly applies to general class of nonnegative matrices as well [61]. Following two lemmas construct result for the convergence of Markov chain to desired steady-state distribution. Lemma 2 presents the *spectral radius condition* that is also equivalent to *primitivity condition* of nonnegative matrices. Brief review of primitive matrices is provided in the appendix.

Lemma 1 *[73] Let M be the transition matrix of an irreducible Markov chain. There exists a unique probability distribution v satisfying $Mv = v$.*

Lemma 2 *[3] Consider the Markov chain M, with a stationary distibution v. Then for any probability vector $x(0) \in \mathbb{P}^n$, it follows that $\lim_{t\to\infty} x(t) = v$ for the system (2.6) if and only if*

$$\rho(M - v\mathbf{1}^T) < 1. \tag{2.9}$$

Proof: First show that

$$(M - v\mathbf{1}^T)^t = M^t - v\mathbf{1}^T. \tag{2.10}$$

By inspection (2.10) is true for $t = 1$. Suppose that

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

then

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

Let $e(t) := x(t) - v$ be the error relative to the desired distribution v. Then, by using the above observations, the error dynamics can be expressed as

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

This proves that e evolves as $e(t+1) = (M - v\mathbf{1}^T)e(t)$, $t \in \mathbb{N}$. If $\rho(M - v\mathbf{1}^T) < 1$ then the error dynamics will be asymptotically stable, i.e., $\lim_{t\to\infty} e(t) = \mathbf{0}$. This implies that $\lim_{t\to\infty} x(t) = v$. Next, we want to show that if $\lim_{t\to\infty} e(t) = \mathbf{0}$ for all $x(0)$ then $\rho(M - v\mathbf{1}^T) < 1$. Using (2.10), we have

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

The fact that $\lim_{t\to\infty} e(t) = \mathbf{0}$ for any $x(0)$ implies that $\lim_{t\to\infty} e(t) = \lim_{t\to\infty}(M - v\mathbf{1}^T)^t x(0) = 0$. The vector, $x(0)$, is constrained as $x(0) \geq \mathbf{0}$ and $\mathbf{1}^T x(0) = 1$, and can be chosen, for example, any of the basis vectors $\{e_1, ..., e_m\}$. The fact that this basis set spans $\mathbb{R}^m$ ensures that, $\lim_{t\to\infty}(M - v\mathbf{1}^T)^t = \mathbf{0}$. which implies that $\rho(M - v\mathbf{1}^T) < 1$. ∎

Another important property of Markov chains is the *reversibility condition* that is given in the following definition. The assumption of reversibility condition is very useful when constructing convergence constraints for Markov chain, as discussed in the next chapter.

Definition 5 *A Markov process is* reversible *if the transitional probabilities remain the same when the direction of time is reversed. This condition can be expressed as*

$$M\text{diag}(v) = \text{diag}(v)M^T, \quad M \in \mathbb{P}^{n\times n}, \tag{2.11}$$

for the stationary distribution v and the Markov matrix M.

The reversibility condition is also known as a *detailed balance* condition [69] and having reversibility implies the stationarity of v for the Markov chain, which can be shown as follows:

$$\mathbf{1}^T M\text{diag}(v) = \mathbf{1}^T\text{diag}(v)M^T \text{ where } \mathbf{1}^T\text{diag}(v) = v^T,$$

$$\Rightarrow \mathbf{1}^T\text{diag}(v) = v^T M^T$$

$$\Rightarrow v^T = v^T M^T \Rightarrow Mv = v.$$

Note that having v as the stationary distribution of M does not necessarily imply its reversibility.

A reversible Markov process with a stationary distribution v satisfies the following condition

$$M[i,j]v[j] = M[j,i]v[i] \quad \forall i,j \in \mathbb{N}_n^+$$

which implies

$$M\mathrm{diag}(v) = \mathrm{diag}(v)M^T. \tag{2.12}$$

2.3 Probabilistic Density Control Algorithm

This section gives the probabilistic density algorithm to demonstrate the implementation of a Markov chain based policy to control the density of a multi-agent system. Consider a swarm comprised of N agents. Let the kth agent state be $s_k(t)$ at time index $t \in \mathbb{N}$ and $x_k(t) \in \mathbb{P}^n$ such that the i'th entry $x_k[i](t)$ is the probability of the event that $s_k(t)$ is s_t at time t,

$$x_k[i](t) := \mathcal{P}(s_k(t) = s_i), \quad i \in \mathbb{N}_n^+, \quad k \in \mathbb{N}_N^+. \tag{2.13}$$

where the probabilities of these N events are statistically independent, and the time evolution of the probability distribution of each agent's state is given as:

$$M_k[i,j] = \mathcal{P}\left(s_k(t+1) = s_i | s_k(t) = s_j\right), \quad i,j \in \mathbb{N}_n^+,\ k \in \mathbb{N}_N^+,\ t \in \mathbb{N}, \tag{2.14}$$

where $M_k \in \mathbb{P}^{n \times n}$ is the *Markov matrix* for k^{th} agent and

$$x_k(t+1) = M_k x_k(t), \quad k \in \mathbb{N}_N^+, \quad \text{with} \quad x_k(0) \in \mathbb{P}^n. \tag{2.15}$$

Assuming that each agent can determine its current state and has an algorithm to sample a random number from a uniform distribution, the PDC can be implemented by using the Algorithm 1 with given M_k.

Algorithm 1: Probabilistic Density Control (PDC)

Each Agent:

Determines its current state, $s_k(t) = s_i$;

Generates a random number $z \sim U[0,1]$;

Transitions to state j, $s_k(t+1) = s_j$, if $j = 1: \ z \leq M_k[1,i]$

$j > 1: \ \sum_{l=1}^{j-1} M_k[l,i] \leq z \leq \sum_{l=1}^{j} M_k[l,i]$.

In the scope of this research, each agent has the same Markov matrix at any given time and same initial density distribution, that is, $M_1 = \ldots = M_N = M$ and $x_k(0) = x(0) = x_0, \forall k$. When all agents evolve independently by using an identical Markov matrix, the swarm density evolves as:

$$x(t+1) = Mx(t) \quad \text{with} \quad x(0) = x_0 \tag{2.16}$$

The idea behind density control is to mathematically control the propagation of probability vector x, rather than individual agent states. While the actual distribution of agent states 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.

Lemma 3 *Consider N agents where $x_1(0) = \ldots = x_N(0) = x_0 \in \mathbb{P}^n$ with x_k, $k \in \mathbb{N}_N^+$ defined as in (2.13). Further, suppose that each agent uses the*

PDC algorithm with the same Markov matrix M, that is, $M_1 = \ldots = M_N = M$. Then, $x_1(t) = \ldots = x_N(t) = x(t)$ for $t \in \mathbb{N}$ where (2.16) holds and

$$x[i](t) = \mathcal{P}(s(t) = s_i) = \mathbb{E}\left(\frac{\mathbf{n}[i](t)}{N}\right), \quad i \in \mathbb{N}_n^+, \tag{2.17}$$

$\mathbf{n}(t)$ is the vector of the number of agents in each state at time t. Furthermore,

$$\mathcal{P}\left(\lim_{N\to\infty} \frac{\mathbf{n}(t)}{N} = x(t)\right) = 1, \quad t \in \mathbb{N}. \tag{2.18}$$

Proof: It is straight forward to show that (2.15) implies (2.6) by using the the Total Probability theorem [26] and noting that

$$\mathcal{P}(r(t+1) \in R_i) = \sum_{j=1}^{m} \mathcal{P}(r(t+1) \in R_i | \mathcal{P}(r(t) \in R_j))\mathcal{P}(r(t) \in R_j).$$

Since the PDC algorithm uses the same Markov matrix for each agent at each time, $M_1(t) = \ldots = M_N(t) = M(t)$ with $x_1(0) = \ldots = x_M(0) = x(0)$, the probability distribution of all agents evolves according to (2.6). Clearly $\mathcal{P}(r(t) \in R_i)$ is the probability of finding any of the agents in ith state. Since

$$\mathbb{E}(\mathbf{n}[i](t)) = \sum_{k=1}^{N} x_k[i](t) = Nx[i](t),$$

we have

$$x(t) = \mathbb{E}(\mathbf{n}(t)/N).$$

Next, we prove (2.18) by using standard arguments as in Theorem 5.4.2 of [26]. Consider $x[i](t)$ which is the probability of finding any agent in state i at time t. Consider a new Random Variable (RV), $Z_k[i](t)$, such that $Z_k[i](t) = 1$ if agent k is in state i at time t, and zero otherwise. Clearly, $\mathbb{E}(Z_k[i](t)) =$

$x_k[i](t) = x[i](t)$ and $Z_k[i](t),\ k = 1, ..., N$, form Independently Identically Distributed (iid) RVs. Then it follows that $\mathbf{n}[i](t) = Z_1[i](t) + \ldots + Z_N[i](t)$. Since $Z_k[i](t),\ k = 1, ..., N$ are iid RVs and $\mathrm{E}[|Z_k[i](t)|] < \infty$, we can use the strong law of large numbers theorem, Theorem 5.4.2 [26], to conclude,

$$\mathcal{P}\left(\lim_{N\to\infty} \frac{\mathbf{n}[i](t)}{N} = x[i](t)\right) = 1,$$

which implies (2.18). ■

Chapter 3

MARKOV CHAIN SYNTHESIS PROBLEM WITH CONSTRAINTS

The main objective of the research is: *To synthesize decision-making policies for ON/OFF agents to make statistically independent decisions, which result in a desired behavior of the overall agent density distribution while satisfying safety constraints on the density.* In this regard, the decision control policy synthesis problem is formulated as a Markov chain synthesis problem, while all desired constraints are imposed on the Markov matrix. The desired constraints for the density control problem are: (i) Ergodicity constraint that ensures asymptotic convergence to prescribed steady-state final distribution; (ii) Transition constraints that prohibits undesirable transitions between states; and (iii) Safety constraints that is used to bound a prescribed linear function of the density. The safety constraints are formulated in a general linear form and captures several types of constraints with different selection of the parameters. For example, it can be used to put an upper bound constraint on the density or to bound the rate of change of density for each state. The ergodicity constraints have been studied widely in Markov chain domain [73] and also recent studies [3, 19, 20] presented equivalent Linear-Matrix Inequality (LMI) conditions on the Markov matrix. Transition constraints are easier to capture and can be imposed by using a linear equality condition on Markov matrix. The most challenging constraints in terms of formulation are the safety constraints and constitutes a basis for this research. Density upper bound constraints are also referred to as safety upper bound constraints and another set of equivalent conditions for Markov chains were presented in [6]. This previous result is very useful to check whether a given Markov chain satisfies the density upper bound constraints or not, i.e., it is a useful analysis result. This research provides a *synthesis* condition that is numerically tractable; that is, a condition for the design of a Markov matrix that satisfies all relevant problem constraints. Chronologically, first the proof is done for safety upper bound and safety rate constraints [2]. Then, it is generalized to a linear form that captures these constraints and additional constraints [34]. This dissertation includes the proof for the generalized case and explains how to obtain the form for a specific type constraint. The proof is based on duality theory of convex optimization and gives equivalent Linear-Matrix Inequality (LMI) conditions on the Markov matrix.

3.1 Convexification of the Problem Constraints

3.1.1 Transition Constraints

For some cases, the transition from j^{th} state to i^{th} state in a single time step may not be desirable or physically possible in practice. Such transitions can be mathematically disallowed by setting the associated element of M to zero, i.e., $M[i,j] = 0$. These transition constraints are imposed on M by the following equation

$$(\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0} \tag{3.1}$$

where A_a is the adjacency matrix, that is, $A_a[i,j] = 1$ if the transition from i^{th} state to j^{th} state is allowable, and is zero otherwise. The adjacency matrix here defines the connectivity of the graph $\mathcal{G}_a$ associated with the Markov matrix

M. Here, $\mathcal{G}_a = (V_a, A_a)$ is a directed graph with the set of vertices V_a is chosen to be the states of the Markov chain. The transpose of matrix A_a is needed because the element $A_a[i,j]$ of A_a is defined for the transition from i^{th} state to j^{th} state while the element $M[i,j]$ of M is defined for the transition from $j^{th} state$ to i^{th} state.

Note that, in order to synthesize an ergodic Markov chain, which is valid for any initial distribution, jth state must be reachable from ith state for every (i,j). This can be ensured by selecting A_a as the edges of a strongly connected graph, i.e. for some integer k, A_a^k is a positive matrix. The transition constraint given in (3.1) is linear in M, hence it can be directly put in the optimization problem in the existing form.

3.1.2 Convergence to Desired Density – Ergodicity Constraint

For the problem defined, it is desired to achieve a specified steady-state distribution $v \in \mathbb{P}^n$ by synthesizing the matrix M which will guide the state probability distribution from any given initial distribution $x(0) \in \mathbb{P}^n$ toward the desired steady-state final distribution v, i.e.,

$$\lim_{t\to\infty} x(t) = v, \quad \text{and} \quad Mv = v$$

Notice that $Mv = v$ are simple linear equalities, and hence they are convex constraints but are not sufficient for convergence. It is desired for x to asymptotically converge to v, i.e., for v to be a globally attractive stationary distribution for M. Lemma 2 established a necessary and sufficient condition for asymptotic convergence on the design of matrix M, denoted as the <u>*spectral radius condition*</u> 2.9 [3].

We will present 3 different approaches to formulate ergodicity constraints as convex constraints on the Markov matrix, M.

1. From linear system theory [27, 66], if there exists a Lyapunov matrix $P = P^T \succ 0$, the spectral radius condition (2.9) is equivalent to the following inequality for some $\lambda \in [0, 1)$:

$$\lambda^2 P - (M - v\mathbf{1}^T)^T P(M - v\mathbf{1}^T) \succeq \mathbf{0}. \tag{3.2}$$

   Note that this inequality is a bilinear matrix inequality with both M and P as solution variables. [30] presents a very useful LMI condition which is an expansion of the discrete Lyapunov condition for stability analysis by introducing a new matrix variable. For a given compact matrix set, this novel stability condition determines whether each matrix element has its eigenvalues with magnitude less than one, or not. This condition then is generalized for the spectral radius condition in [3]. Using the results presented in [3] and [30] inequality (3.2) can be shown to be equivalent to the following matrix inequality with some matrix G:

$$\begin{bmatrix} \lambda^2 P & (M - v\mathbf{1}^T)^T G^T \\ G(M - v\mathbf{1}^T) & G + G^T - P \end{bmatrix} \succeq 0. \tag{3.3}$$

   Here, the equivalence can be shown by applying Schur complement with respect to the block (2,2) and multiplying block (1,1) with $[I \quad -(M - v\mathbf{1}^T)^T]$ from the left and with its transpose from the right.

Remark 1 *In (3.3) G is a prescribed matrix, which is selected as* $G = \mathrm{diag}(v)^{-1}$ *in our examples. To determine the λ, a line search can be conducted to find minimum feasible λ, hence maximum convergence rate [2].*

Prescribing matrix G is a source of conservatism. Incorporating G into the optimization can be achieved by solving Bilinear Matrix Inequalities (BMIs), whose solution complexity grows exponentially with the problem size. BMI formulations can be pursued to remove any conservatism for small size problems.

2. As a second approach, *reversibility* can be imposed on the Markov matrix as given in (2.12) [19].

   In order to formulate the LMI condition for a reversible Markov chain, consider the spectral radius condition (2.9) with $v > \mathbf{0}$ and let $r := v^{1/2}$ (element-wise square root) and $H = \text{diag}(r)$,

$$\rho(M - v\mathbf{1}^T) = \rho(M - Hr\mathbf{1}^T) = \rho\left(H^{-1}(M - Hr\mathbf{1}^T)H\right),$$

   which follows from the fact that the similarity transformations do not change the eigenvalues. This implies that $\rho(M - v\mathbf{1}^T) = \rho(H^{-1}MH - rr^T)$. Next since M is constrained to be reversible, we must have $M\text{diag}(v) = \text{diag}(v)M^T$, which implies that $MH^2 = H^2M^T \Rightarrow H^{-1}MH = HM^TH^{-1}$, that is, $H^{-1}MH$ is a symmetric matrix. Then, it is noted that the spectral radius condition on the symmetric matrix $\rho\left(H^{-1}MH - rr^T\right) < 1$ can be equivalently expressed as [19]

$$\lambda I \succeq H^{-1}MH - rr^T \succeq -\lambda I \quad \text{where} \quad \lambda \in [0, 1). \tag{3.4}$$

   Note that λ, the convergence rate, can be minimized for fastest convergence within a convex problem formulation.

3. Finally, the third approach is to impose strong connectivity on the corre-

sponding adjacency matrix of the Markov matrix, i.e. $\texttt{i}(M)$ is strongly connected. This can be obtained using the following linear equations:

$$M[i,j] \geq \epsilon \quad \text{if} \quad A_a[i,j] = 1, \tag{3.5}$$

where $\epsilon > 0$ is a small positive scalar (e.g., it can be set as the machine precision). This provides a sufficient condition for ergodicity of the Markov chain. Thus if $Mv = v$ and the adjacency matrix A_a corresponds to a connected graph, then v is unique stationary distribution and $\lim_{t\to\infty} x(t) = v, \ \ \forall\, x(0) \in \mathbb{P}^n$. Note that, though convergence can be ensured with simpler inequalities, convergence rate cannot be imposed or minimized directly in this approach.

Remark 2 *Conditions 1 & 2 given above are LMI constraints, which allow us to specify the convergence rate λ for the Markov chain; however, for large dimensional problems LMI's can be computationally expensive. The condition 3 is a sufficient condition for ergodicity and does not impose any condition on convergence rate, but it is a simple constraint which is preferable for large dimensional problems.*

3.1.3 Density Safety Constraints:

The *safety* constraints are the hardest to capture as convex constraints, which are expressed in a generic form as follows

$$L(t)\, x(t) \leq q(t), \ \ \forall\, x(t) \leq p(t), \ t \in \mathbb{N}. \tag{3.6}$$

Here, L is a linear function of M. Since x(t) evolves with (2.16), this constraint is not convex when both M and $x(t)$ are solution variables. The above constraint captures several types of constraints with different selection of L, q and p. First example of these constraints is the density upper bound constraint [6, 2, 31], which can be imposed with $L(t)=M$, $q(t)=d$, and $p(t)=d$, that is,

$$x(t) \leq d, \quad t = \mathbb{N}^+ \tag{3.7}$$

where $\mathbf{0} < d \leq \mathbf{1}$ is a vector defining the prescribed density upper bounds for each state and it is assumed that $x(0) \leq d$. This constraint is also known as *safety constraint* in Markov chain literature [6].

In addition to the density upper bound constraints, the general form (3.6) can also capture density rate constraint that is used to limit the rate of change of density for each state of Markov chain, i.e.,

$$-f \leq x(t+1) - x(t) \leq f, \quad \forall x(t) \leq d, \; t \in \mathbb{N}, \tag{3.8}$$

where $f \geq 0$ bounds the density rate. Note that this is equivalent to case where

$$L(t) = \begin{bmatrix} M - I \\ I - M \end{bmatrix}, \quad q = \begin{bmatrix} f \\ f \end{bmatrix},$$

and $p(t) = d$ in (3.6).

The safety upper bound constraints are imposed in the Markov matrix by equivalent linear inequality constraints that are presented in Theorem 1. 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 [88].

Theorem 1 *[34, 2] Consider the Markov chain given by (2.16). Then,*

$$L(t)x(t) \le q(t), \quad \forall x(t) \le p(t), \tag{3.9}$$

if and only if *there exist* $S(t) \in \mathbb{R}^{n\times n}$ *and* $y(t) \in \mathbb{R}^n$ *such that*

$$\begin{aligned} S(t) \ge \mathbf{0}, \quad \left[L(t)+S(t)+y(t)\mathbf{1}^T\right] \ge 0, \\ y(t)+q(t) \ge \left[L(t)+S(t)+y(t)\mathbf{1}^T\right] p(t). \end{aligned} \tag{3.10}$$

Proof: The safety condition holds if and only if the following conditions hold for $i \in \mathbb{N}_n^+$:

$$\max_{\hat{x}\in\mathcal{P}^m,\, \hat{x}\le p(t)} \mathrm{e}_i^T(L(t))\hat{x} - q[i](t) \le 0. \tag{3.11}$$

This maximization problem can be equivalently written as, for $i \in \mathbb{N}_n^+$,

$$\min_{\hat{x}\in\mathcal{P}^m, \hat{x}\le p(t)} -\mathrm{e}_i^T(L(t))\hat{x} + q[i](t) \ge 0. \tag{3.12}$$

By introducing slack variables, this problem can be written as follows:

$$\begin{aligned} &\min_{\hat{x},\xi} -\mathrm{e}_i^T L(t)\hat{x}, \qquad i \in \mathbb{N}_n^+, \quad \text{s.t.} \\ &\hat{x} + \xi = p(t), \quad \mathbf{1}^T\hat{x} = 1 \\ &\hat{x} \ge \mathbf{0}, \quad \xi \ge \mathbf{0}. \end{aligned}$$

Note that the above problems are linear programming problems written in a standard form [22, 12],

$$\min_{\eta\ge\mathbf{0},\, A\eta=b} c^T\eta$$

where $\eta = (\hat{x}, \xi)$ and

$$A = \begin{bmatrix} I & I \\ \mathbf{1}^T & \mathbf{0} \end{bmatrix}, \; b = \begin{bmatrix} p(t) \\ \mathbf{1} \end{bmatrix}, \; c = \begin{bmatrix} -L(t)^T \mathrm{e}_i \\ \mathbf{0} \end{bmatrix}.$$

The duals of these optimization problems,

$$\max_{g(t) \geq \mathbf{0},\, A^T f(t) + g(t) = c} b^T f(t),$$

which are given explicitly as follows

$$\begin{aligned} \max_{z(t), y(t), s(t), \rho(t)} p^T(t) z(t) + y(t), \qquad & i \in \mathbb{N}_n^+, \quad \text{s.t.} \\ z(t) + \mathbf{1} y(t) + s(t) = -L(t)^T \mathrm{e}_i & \\ z(t) + \rho(t) = 0, \quad s(t) \geq \mathbf{0}, \quad \rho(t) \geq \mathbf{0}, & \end{aligned} \tag{3.13}$$

where $g(t) = (s(t), \rho(t))$ and $f(t) = (z(t), y(t))$. These imply that:

$$\begin{aligned} & \rho(t) = -z(t) \Rightarrow \\ & z(t) = -L(t)^T \mathrm{e}_i - \mathbf{1} y(t) - s(t). \end{aligned}$$

Using this relation, the dual problem can be written as follows: For $i \in \mathbb{N}_n^+$,

$$\begin{aligned} \max\nolimits_{y(t), s(t)} \; y(t) - p^T(t)[(L(t))^T \mathrm{e}_i + \mathbf{1} y(t) + s(t)] \;\; \text{s.t.} \\ L(t)^T \mathrm{e}_i + \mathbf{1} y(t) + s(t) \geq \mathbf{0}, \quad s(t) \geq \mathbf{0}. \end{aligned} \tag{3.14}$$

Since both the primal and dual are LP problems, the existence of a primal or dual feasible solution implies that both the primal and the dual are feasible with zero duality gap [12], that is, $-\mathrm{e}_i^T L(t) \hat{x} - p(t)^T z(t) - y(t) = \hat{x}^T s(t) + \xi^T \rho(t) = 0$.

Now, the optimal solution to the i^{th} problem will be denoted with a subscript i, i.e., $y(t)_i$ and $s(t)_i$. Since the duality gap is zero, a necessary and sufficient condition for the safety is, by using (3.12), that the optimal solution of the dual problem must satisfy, for $i \in \mathbb{N}_n^+$,

$$\begin{aligned} & y_i(t) - p^T(t)[L(t)^T \mathrm{e}_i + \mathbf{1} y_i(t) + s_i(t)] \geq -q[i](t) \\ & s_i(t) \geq \mathbf{0}, \quad L(t)^T \mathrm{e}_i + \mathbf{1} y_i(t) + s_i(t) \geq \mathbf{0}. \end{aligned}$$

When the above equations are written in matrix form for all i, with

$$S(t) = \begin{bmatrix} \leftarrow & s_1(t)^T & \rightarrow \\ \vdots & \vdots & \vdots \\ \leftarrow & s_m(t)^T & \rightarrow \end{bmatrix}, \quad y(t) = \begin{bmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{bmatrix},$$

the conditions given by (3.10) are obtained, which concludes the proof. ∎

Note that $p(t) = \mathbf{1}$ can be a choice in general, but other values of $p(t)$ will also be useful as demonstrated below. The safety bounds are generally taken as time-invariant but the proof is constructed for the time-varying case to provide most general linear result. It is also critical to note that the conditions given in (3.10) define linear inequality constraints on the solution variables, S, y, M, if L is a linear function of M, which is the main utility of the theorem.

Density upper bound constraint: In this case we want to ensure that the density vector components stay below the components of a prescribed vector, d as in (3.7), which is equivalent to (3.9) with

$$L(t) = M, \; p(t) = d, \; q(t) = d. \tag{3.15}$$

The following corollary shows the equivalent linear matrix inequalities for the density upper bound constraints.

Corollary 1 *Consider the Markov chain given by (2.16). For every* $x(0) \leq d(0)$*,* $x(t) \leq d(t), \;\; \forall t \in \mathbb{N}^+$, if and only if *there exists* $S \in \mathbb{R}^{n \times n}$ *and* $y \in \mathbb{R}^n$ *such that,*

$$\begin{aligned} S \geq \mathbf{0}, \quad M + S + y\mathbf{1}^T \geq \mathbf{0}, \\ y + d \geq \left(M + S + y\mathbf{1}^T\right) d. \end{aligned} \tag{3.16}$$

Density rate constraint: This constraint is used to limit the rate of change of density for each state of Markov chain as given in (3.8). That is equivalent to (3.9) with

$$L(t) = \begin{bmatrix} M - I \\ I - M \end{bmatrix}, \quad q = \begin{bmatrix} f \\ f \end{bmatrix}, \quad p(t) = d. \tag{3.17}$$

In addition to the density upper bound and density rate constraints, for the case when the probability distribution is defined based on the positions of the agent as in Section 3.6.1, two other useful constraints are captured by the general form:

Density flow constraint: In this case, the objective is to bound the density flow in and out of the bins (states) by setting a limit on the maximum density of each bin (state) at any time instance as in (3.21), which can be expressed by (3.9). Assuming $x(t)$ satisfies the density upper bound constraint (enforced via Corollary 1), the following constraints bound the flow:

$$\begin{aligned} L(t) = I + M(t) - \mathrm{diag}(M(t)) + \sum_{i=1}^{m} \mathrm{e}_i \mathbf{1}^T (T_i \odot M(t)), \\ p(t) = d(t),\ q(t) = \alpha(t). \end{aligned} \tag{3.18}$$

Corollary 2 *Consider the Markov chain given by (2.16). For* $x(t) \le d(t),\ t \in \mathbb{N}$*, the constraint (3.9) with (3.18) holds* if and only if *there exist* $S(t) \in \mathbb{R}^{m\times m}$ *and* $y(t) \in \mathbb{R}^m$*, such that, for* $t \in \mathbb{N}$.

$$\begin{aligned} S(t) \ge \mathbf{0}, \quad \Gamma(t) \ge 0, \\ y(t) + \alpha(t) \ge \Gamma(t) d(t), \end{aligned} \tag{3.19}$$

where $\Gamma(t) = I + M - \mathrm{diag}(M(t)) + \sum_{i=1}^{m} \mathrm{e}_i \mathbf{1}^T (T_i \odot M(t)) + S(t) + y(t)\mathbf{1}^T$.

Since the density is defined at discrete time instances, the density constraint can be violated during transitions between stages $t \to t+1$. This can be achieved via an additional constraint on the flow through the bins ensuring that the expected number of agents in each bin is bounded by a prescribed quantity for all time - for a given upper bound on the total number of agents. Let $\bar{x}[i](t)$ be the maximum possible density for the bin i during the time interval $\mathbf{I}(t)$ in between the temporal stages t and $t+1$, and $T_r \in \mathbb{R}^{n\times n}$ be a non-negative *trajectory* matrix for each bin that describes the feasible paths containing the r'th bin, $r = \in \mathbb{N}_n^+$: For $i, j \in \mathbb{N}_n^+$,

$$T_r[i,j] = \begin{cases} p_r[i,j] & \text{if path } j \to i \text{ passes through bin } r \\ 0 & \text{otherwise} \end{cases}, \tag{3.20}$$

where $p_r[i,j] \in [0,1]$ is the probability that the path from $j \to i$ passes through bin r. Note that $p_r[i,j]=1$ when there is only a single feasible path between bins i and j that must go through the r'th bin. Equipped with the trajectory matrix, the maximum density in R_i is bounded by the sum of the probability that an agent choses to stay in R_i, the probability that an agent from another bin transitions to R_i, and the probability that an agent passes through R_i,

$$\bar{x}[i](t) \leq \underbrace{x[i](t)}_{\text{agents in } R_i \text{ at } t} + \underbrace{\sum_{j\neq i} M[i,j](t)\, x[j](t)}_{\text{agents moving into } R_i} + \underbrace{\mathbf{1}^T (T_i \odot M(t)) x(t).}_{\text{agents passing through } R_i} \tag{3.21}$$

The above constraint bounds the expected number of agents in R_i for a PDC problem. Inequality (3.21) can be written in a matrix form for all bins as follows,

$$\begin{aligned} \bar{x}(t) &\leq x(t) + M(t)x(t) - \mathrm{diag}(M(t))x(t) + \sum_{i=1}^{m} \mathrm{e}_i \mathbf{1}^T (T_i \odot M(t)) x(t) \\ &= \left(I + M(t) - \mathrm{diag}(M(t)) + \sum_{i=1}^{m} \mathrm{e}_i \mathbf{1}^T (T_i \odot M(t)) \right) x(t). \end{aligned}$$

Now, we can put a bound on maximum density of the bins as $\bar{x}(t) \leq \alpha$ for all $t \in \mathbb{N}$. This is equivalent to, assuming that (3.7) holds,

$$\left(I + M(t) - \mathrm{diag}(M(t)) + \sum_{i=1}^{m} \mathrm{e}_i \mathbf{1}^T (T_i \odot M(t)) \right) x(t) \leq \alpha, \quad \forall\, x(t) \leq d, \quad t \in \mathbb{N} \tag{3.22}$$

The above constraint is also captured by the general form of the safety constraint given in (3.6), with $L(t) = I + M(t) - \mathrm{diag}(M(t)) + \sum_{i=1}^{m} \mathrm{e}_i \mathbf{1}^T (T_i \odot M(t)$, $q(t) = \alpha$, and $p(t) = d$.

The density diffusivity constraint: It limits the spatial gradients of the density with a prescribed vector, $w(t) \geq \mathbf{0}$,

$$-w(t) \leq D_1 x(t) \leq w(t), \quad \forall x(t) \leq d, \quad t \in \mathbb{N}^+ \tag{3.23}$$

where D_1 is a discrete approximation to the first order spatial derivative. The density diffusivity constraint ensures that the density evolves smoothly without steep jumps in the density between neighboring bins. This inequality can be obtain by letting

$$L(t) = \begin{bmatrix} D_1 \\ -D_1 \end{bmatrix}, \quad q(t) = \begin{bmatrix} w(t) \\ w(t) \end{bmatrix},$$

and $p(t) = d(t)$ in (3.6).

Remark 3 *For any Markov matrix M, $d = \mathbf{1}$ is feasible for (3.10) as expected, since the probability cannot become larger than one for any state. To see this, note that for $d = \mathbf{1}$, $y = -\mathbf{1}$ and $S = \mathbf{1}\mathbf{1}^T - M \geq 0$ satisfies the inequalities in (3.10).*

The above remark leads to the following corollary, which gives a sufficient condition for the density upper bound constraints.

Corollary 3 *A Markov chain with the matrix M satisfies the density upper bound constraints defined by $d > 0$ and $\mathbf{1}^T d > 1$, if $M \leq d\mathbf{1}^T$.*

Proof: The proof follows easily by letting $S = d\mathbf{1}^T - M \geq 0$ and $y = -d$, which define a feasible solution of (3.10). ∎

This corollary indicates that, without any state transition constraints (to be described later), $M = v\mathbf{1}^T$ for a $v \in \mathcal{P}^m$ such that $v \leq d$ leads to Markov chain that satisfies the density upper bound constraints and the ergodicity constraints (to be defined later).

We will present another condition for the satisfaction of the safety constraints, which was first given in [6]. This condition presents a way to check whether the synthesized Markov matrix satisfies the constraint or not. However, there is still a need computationally tractable condition to Markov chain synthesis with all relevant constraints which is achieved with the result presented in Theorem 1.

Theorem 2 *Consider the Markov chain given by (2.6) with* $M \in \mathcal{M}^m$*. For every* $x(0) \leq d$*,* $x(t) \leq d, \quad t = 1, 2, ...,$ *i.e. the density upper bound constraint holds,* if and only if *the following condition holds:*
For each $i = 1, ..., m$*, let* $r_i := M^T \mathrm{e}_i$ *(transpose of* i*'th row of* M*). Consider a permutation matrix* P_i *such that* $\eta_i := P_i r_i$ *is a vector in descending order and let* $\phi_i := P_i d$*. Let the index* k_i *be such that* $\mathbf{1}^T \phi_{i,k_i} < 1$ *and* $\mathbf{1}^T \phi_{i,k_i+1} \geq 1$*, where* $\phi_{i,k} := (\phi_i[1], ..., \phi_i[k-1])$ *for any index* k*. Similarly let* $\eta_{i,k} := (\eta_i[1], ..., \eta_i[k-1])$*. Then the following hold*

$$d[i] \geq \phi_{i,k_i}^T \eta_{i,k_i} + \left(1 - \mathbf{1}^T \phi_{i,k_i}\right) \eta_i[k_i], \quad i = 1, ..., m. \tag{3.24}$$

Proof: To prove the theorem, the dual optimal cost is investigated as a function of y_i, which is denoted as $F(y_i)$. Suppose $y_i \geq -\min(r_i)$, where $r_i = M^T \mathrm{e}_i$, which implies that $\mathbf{1} y_i + r_i \geq \mathbf{0}$. So we can choose $s_i = \mathbf{0}$ for the maximal dual cost, which implies that $F(y_i) = y_i - d^T (r_i + y\mathbf{1}) = (1 - d^T \mathbf{1}) y_i - d^T r_i$. Since $(1 - d^T \mathbf{1}) < 0$,

$$\mathrm{argmax}_{y_i \geq -\min(r_i)} F(y_i) = -\min(r_i),$$

$$\Rightarrow \max_{y_i \geq -\min(r_i)} F(y_i) = (d^T \mathbf{1} - 1) \min(r_i) - d^T r_i.$$

Next suppose that $y_i \leq -\max(r_i)$, that is, $\mathbf{1}y_i + r_i \leq \mathbf{0}$. In this case, to maximize the cost we choose $s_i = -(y\mathbf{1} + r_i) \geq \mathbf{0}$, which implies that $F(y_i) = y_i$.

Lastly we consider $-\max(r_i) \leq y_i \leq -\min(r_i)$. This means that there exists a k such that $-\eta_i[k-1] \leq y_i \leq -\eta[k]$. Further, define $\eta_i = (\eta_{i,k}, \bar{\eta}_{i,k})$ where $\bar{\eta}_{i,k} = (\eta_i[k], ..., \eta_i[m])$. Note that $\eta_i[k-1] = \min(\eta_{i,k})$ and $\eta_i[k] = \max(\bar{\eta}_{i,k})$. We can now express the dual problem (3.14) as

$$\begin{aligned} &\max_{y, s_{i,1}, s_{i,2}} \; y_i - \phi_{i,k}^T h_1 - \bar{\phi}_{i,k}^T h_2 \\ &h_1 = \eta_{i,k} + s_{i,1} + y_i \mathbf{1} \geq \mathbf{0}, \quad s_{i,1} \geq \mathbf{0} \\ &h_2 = \bar{\eta}_{i,k} + s_{i,2} + y_i \mathbf{1} \geq \mathbf{0}, \quad s_{i,2} \geq \mathbf{0} \end{aligned}$$

where $(s_1, s_2) = P_i s_i$. The cost of this problem is

$$(1 - \mathbf{1}^T \phi_{i,k}) y_i - \phi_{i,k}^T \eta_{i,k} - \phi_{i,k}^T s_{i,1} - \bar{\phi}_{i,k}^T h_2.$$

Noting h_2 and $\bar{\phi}_{i,k}$ are nonnegative vectors, $\bar{\phi}_{i,k}^T h_2 = 0$ by having $h_2 = \mathbf{0}$. $h_2 = \mathbf{0}$ is feasible since $s_2 = -\bar{\eta}_{i,k} - y_i \mathbf{1} \geq \mathbf{0}$, which follows from $y_i + \eta_i[k] = y_i + \max(\bar{\eta}_{i,k}) \leq 0$, that is, $y_i \mathbf{1} + \bar{\eta}_{i,k} \leq \mathbf{0}$. Next, since $\eta_i[k-1] = \min(\eta_{i,k})$ and $y_i + \min(\eta_{i,k}) \geq 0$, we have $y_i \mathbf{1} + \eta_{i,k} \geq \mathbf{0}$. Hence $s_{i,1} = \mathbf{0}$ is a feasible solution. Since $s_{i,1}$ appears as $-\phi_{i,k}^T s_{i,1}$ where $\phi_{i,k} > \mathbf{0}$, we must have $s_{i,1} = \mathbf{0}$ for maximal cost. Consequently $F(y_i) = (1 - \mathbf{1}^T \phi_{i,k}) y_i - \phi_{i,k}^T \eta_{i,k}$. In summary, we have

$$F(y_i) = \begin{cases} y_i & y_i \leq -\max(r_i) \\ (1 - \mathbf{1}^T \phi_{i,k}) y_i - \phi_{i,k}^T \eta_{i,k}, & -\eta_i[k-1] \leq y_i \leq -\eta_i[k] \\ (d^T \mathbf{1} - 1) \min(r_i) - d^T r_i & y_i \geq -\min(r_i) \end{cases}$$

By simple substitution, one can show that F is a continuous piecewise linear function, with its last piece being a constant function. It is a strictly increasing function for all $y_i < -\eta_i[k_i]$, and strictly decreasing for $-\eta_i[k_i] < y_i < -\min(r_i)$, and constant after $y_i \geq -\min(r_i)$. This implies that $\text{argmax} F(y_i) = -\eta_i[k_i]$, which then implies that $\max F(y_i) = (1 - \mathbf{1}^T \phi_{i,k_i})\eta_i[k_i] - \phi_{i,k_i}^T \eta_{i,k_i}$. Using the duality conditions (3.15) in the first part of the proof, one can deduce the safety condition (3.24). This concludes the proof. ∎

3.2 LMI Synthesis of Markov Chain without Feedback

All design constraints have been formulated as equivalent linear equality and linear inequality constraints as in (3.3), (3.1), and (3.10). Hence, an LMI optimization problem with the given constraints to minimize a metric on the overall action can be formulated. This can be achieved by making $M \simeq I$; note that $M = I$ is a limiting case since it stops the swarm evolution. For example, the following cost can be minimized for minimal overall action [3]:

$$\mathbf{1}^T(\mathbf{1} - diag(M)). \tag{3.25}$$

Consequently, with a prescribed matrix G and a convergence rate $\lambda \in [0, 1)$, the following LMI optimization problem can be solved to synthesize a desirable Markov matrix when L, q, p in (3.9) are all constant quantities in time:

$$\begin{aligned}
&\min_{M,P,S,y} \quad \mathbf{1}^T(\mathbf{1} - \text{diag}(M)) \qquad \text{s.t} \\
&\mathbf{1}^T M = \mathbf{1}^T, \quad M \geq 0, \quad Mv = v, \quad (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0} \\
&S \geq \mathbf{0}, \qquad [L + S + y\mathbf{1}^T] \geq 0, \quad y + q \geq [L + S + y\mathbf{1}^T]p \\
&\begin{bmatrix} \lambda^2 P & (M - v\mathbf{1}^T)^T G^T \\ G(M - v\mathbf{1}^T) & G + G^T - P \end{bmatrix} \succeq 0, \quad P = P^T \succ 0
\end{aligned} \tag{3.26}$$

Remark 4 *The LMI problem (3.26) can be infeasible for a given* $\lambda \in [0, 1)$*. Then it is also infeasible for any smaller value of* λ*. This observation is used to minimize* λ*, and hence to maximize the convergence rate, by using a standard line search. Hence our approach to the determination of* λ *does not introduce any conservatism. For each feasible* λ *in (3.26),* M *is chosen as close to* I *as possible to keep the number of transitions between the states to a minimum.*

3.3 QP Synthesis of Markov Chain with Density Feedback

This section presents the Markov chain synthesis problem when there is a real-time density feedback. When the measurements or estimates of the probabilistic density distribution are available, they can be utilized to update the Markov matrix, which leads to a faster convergence. The density of agent states can be estimated by any means, whether centrally or in some decentralized manner, which is assumed to be given at this point. The synthesis problem is formulated as a Quadratic Programming problem.

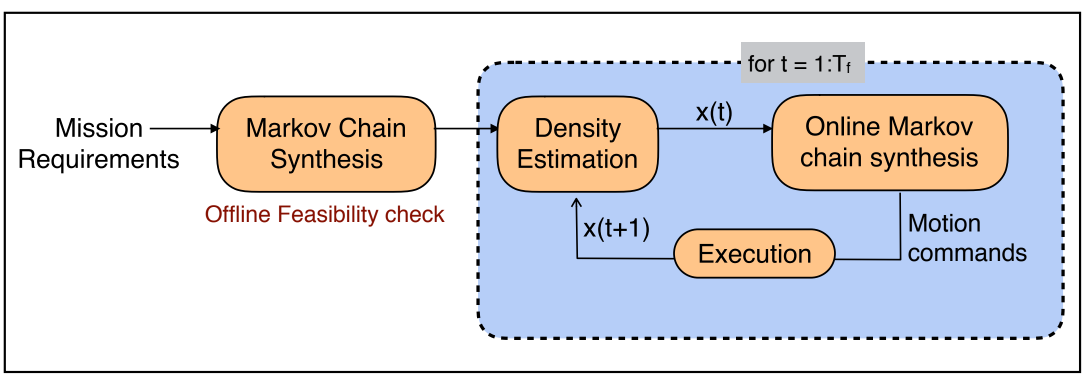

Figure 3.1: Control structure with feedback on the density

A receding horizon approach is proposed to synthesizing the $M(t)$ matrix for given $x(t)$. It is assumed that there exists a Markov matrix $\hat{M}$, a stochastic matrix satisfying the constraints of the optimization problem (3.26) with a corresponding positive definite matrix P.

Now given $x(t)$ estimate, the following QP problem is solved to update $M(t)$:

$$\begin{aligned} M(t) &= \arg\min_{M} \ \|Mx(t) - v\|_P \quad \text{subject to} \\ & M \geq 0, \ Mv = v, \ \mathbf{1}^T M = \mathbf{1}^T, \\ & (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0} \\ & x(t+1) = Mx(t), \quad Lx(t) \leq q. \end{aligned} \tag{3.27}$$

Problems (3.26) and (3.27) allow imposing the constraint $x(t) \leq d$, as in (3.15), together with $\hat{L}x(t) \leq \alpha$ with $\hat{L}$ as L in (3.18) with constant M, d, and α via

$$L = \begin{bmatrix} M \\ \hat{L} \end{bmatrix}, \quad p = d, \quad \text{and} \quad q = \begin{bmatrix} d \\ \alpha \end{bmatrix}.$$

Note that the above optimization problem is a quadratic programming (QP). Consequently, solving this optimization problem 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.

The optimal solution of (3.27) provides the *Markov matrix* $M(t)$ that leads to the closest density distribution to v, with respect to a norm defined by P.

Theorem 3 *Suppose that there exists a Markov matrix and* $P = P^T \succ \mathbf{0}$ *satisfying the constraints of the LMI synthesis problem (3.26). Then the PDC algorithm with the* Markov Matrix $M(t)$ *obtained by solving the QP given by (3.27) for each* $x(t)$ *results in* $\lim_{t\to\infty} x(t) = v$ *for any* $x(0) \leq p$ *while satisfying the motion and general safety constraints with constant values of* L, q, p *in (3.9). Furthermore a feasible solution of the QP (3.27) will exist for all time.*

Proof: Note that since the LMI design with constant Markov matrix, $\hat{M}$, is feasible, the QP problem has $\hat{M}$ as a solution for all time, which proves the last statement. All ergodicity, motion, and safety constraints are explicitly imposed in the QP problem, hence the resulting Markov chain will generate density histories obeying all of these constraints. The only constraint that is not explicit is the convergence to v for all initial conditions. To prove this, 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 (3.27). Since $\hat{M}$ is a feasible solution of this optimization problem, $\hat{M}v = v$ and $\mathbf{1}^T x(t) = 1$, and

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

The inequality (3.2) 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 $\lambda \in [0,1)$, we have $\lim_{t\to\infty}\|e(t)\|_P = 0 \Rightarrow \lim_{t\to\infty}\|e(t)\| = 0$. ■

Hence, if there exists a Markov matrix $\hat{M}$, satisfying the constraints of the optimization problem, then the convergence of the system is guaranteed with the given QP formulation. The control structure with density feedback is given in Figure 3.1. The result of this section will be illustrated via simulations and compared to the case without feedback.

QPs can be solved in real-time, which can be done via custom Interior Point Methods. Note that QPs are simpler to solve than SDPs and the reason we do not have to solve SDPs is that we do not have to impose the spectral radius condition for this case.

Decentralized Counting Algorithm for Density Estimation

This section introduces a decentralized counting algorithm (DCA) for the density estimation, which is needed for the online computations of the Markov matrix via the QP-based method (3.27). In many applications, the region of operation $\mathcal{R}$ describes a set of positions, i.e., the location of an agent. However in general, the position of a mobile agent can be different from the state associated with the set $\mathcal{R}$. For example the *agent state* $r(t)$ may represent the velocity of a vehicle, hence having two vehicles in the same bin R_i does not necessarily mean a physical proximity. Since the communication between two agents is typically associated with their relative distance, we divide the position space into *communication bins*, $\mathcal{C}_i,\ i = 1, ..., m_c$, for the following discussion. Hence the agents have two relevant domains $\mathcal{R}$ and $\mathcal{C}$ with the state bins R_i and the communication bins C_i.

In the special case when the agent state, $r(t)$, is the position vector, it is assumed that $\mathcal{R} = \mathcal{C}$, with the same binning. It is also assumed that the communication bins are selected such that each agent can communicate with all other agents in its communication bin and all agents from all neighboring communication bins can receive the broadcast from (at least) an agent in the bin. It is possible for the communication radius of an agent to span multiple communication bins and therefore multiple bins can be connected to each other. We define a communication adjacency matrix that shows which communication bins are able to communicate to each other, that is, Ac where $A_c[i,j] = 1$ if the communication from bin i to bin j is feasible, and is zero otherwise. Before stating the decentralized counting algorithm, we define a matrix for each communication bin, i.e., $\mathbf{N}_l(t) \in \mathbb{N}^{m \times m_c}$, $l = 1, \ldots, m_c$. $\mathbf{N}_l[i,j](t)$ is the estimate of the number of agents in ith bin that are also in jth communication bin which is available for the agents in lth communication bin. Hence, each agent in lth communication bin has the estimate $\mathbf{n}_{l,j}(t)$ for the number of agents in each bin, where $\mathbf{n}_{l,j}(t)$ is the jth column of $\mathbf{N}_l(t)$. Then the decentralized counting process consists of the following steps,

If we assume that the communication graph defined by A_c is connected, this process leads to identical $\mathbf{N}_1(t) = \ldots = \mathbf{N}_{m_c}(t) = \mathbf{N}(t)$ after at most $n_c - 1$ communication updates (after the first step) [36]. Hence, including the communication needed for the first step, there are at most n_c communication updates needed for every agent to reach to an identical $\mathbf{N}(t)$. Next, the best estimate of the density is obtained by

$$x(t) = \frac{\mathbf{N}(t)\mathbf{1}_{m_c}}{\mathbf{1}_m^T \mathbf{N}(t)\mathbf{1}_{m_c}}.$$

This counting method simplifies significantly to the following when the agent

Algorithm 2: Decentralized Counting Algorithm (DCA)

1 At time t, each agent broadcasts its ID, current state and communication bin numbers;

2 Agents construct $\mathbf{N}_l(t) \in \mathbb{N}^{m \times m_c}$: Each agent in lth communication bin initializes $\mathbf{n}_{l,l}(t)$ by using the information from the first step above, and initializes $\mathbf{n}_{l,j}(t) = \mathbf{0}$ for $j \neq l,\ j = 1, ..., m_c$. Hence all agents in the same communication bin have the identical initial value for $\mathbf{N}_l(t)$;

3 Each agent broadcasts $\mathbf{N}_l(t)$. Consequently, each agent in the jth communication bin receives $\mathbf{N}_l(t)$ from the neighboring lth bin, and adds to its own for all its neighbors:
$\mathbf{N}_j[:, i](t) = \mathbf{N}_j[:, i](t) + \sum_{l \in \mathcal{N}_j} \mathbf{N}_l[:, i](t),\ i \in Z$, where $\mathcal{N}_j$ is the index set of neighboring bins for the jth bin, Z is the index set of zero columns in $\mathbf{N}_j(t)$, and $\mathbf{N}_j[:, i](t)$ represents its i'th column. This addition is continued.

state relevant to $\mathcal{R}$ is the position state, i.e., $\mathcal{R} = \mathcal{C}$ and $m = m_c$, as follows. When a density update is requested, all agents broadcast their IDs and their current states' 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 information across bins is illustrated in Figure 3.2.

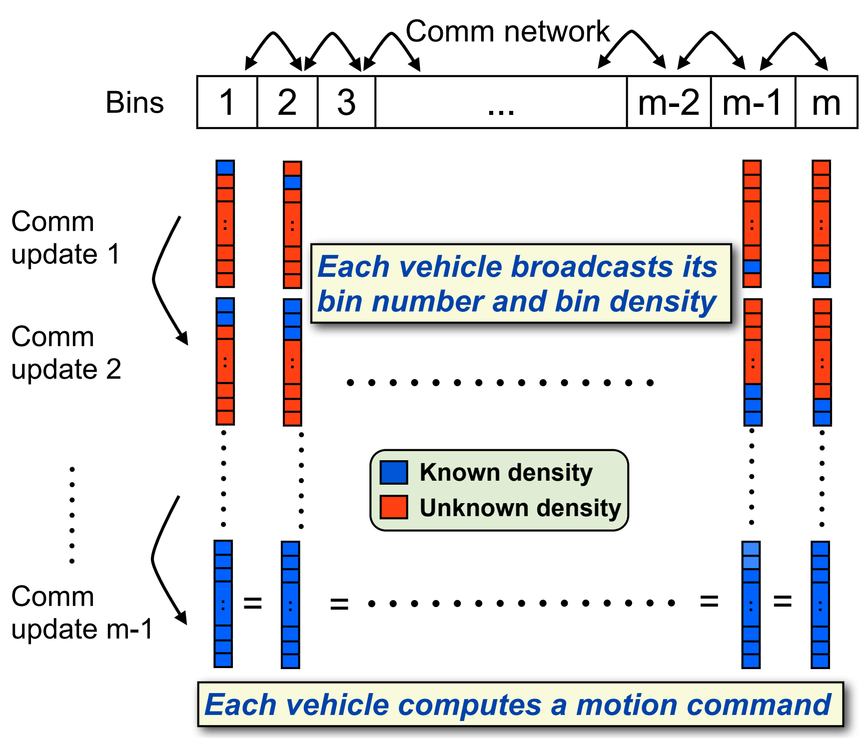

Figure 3.2: An illustration of decentralized density computation via decentralized counting to obtain $\mathbf{n}(t)$ when $\mathcal{R} = \mathcal{C}$.

3.4 End-to-End Density Control Algorithm

This section presents the complete density control algorithm both for the cases with and without density feedback. For these two cases, two different stopping conditions are employed: when there is density feedback, the algorithm stops when the norm of the difference between current and desired density is less than a prescribed threshold; when there is no feedback, the algorithm stops when $t = t_{stop}$, which is determined by convergence rate.

Algorithm 3: End-to-End Density Control Algorithm

LMI (3.26) is solved to compute matrix M, which is sent to all agents;
if *NO density feedback* **then**
    for $t = 0, \ldots, t_{stop}$ **do**
        run PDC
    end
else
    while $||x(t) - v|| > \epsilon$ **do**
        run PDC;
        run DCA;
        each agent computes $M(t)$ via QP (3.27) using its density estimate ;
        check $||x(t) - v||$
    end
end

Note that for the case without density feedback, LMI is solved once offline and all agents are given the resulting Markov matrix, via IPM algorithms [21, 55, 77]. For the case with density feedback, there has been recent research on custom IPMs for online solution of the QP problems [82, 41, 40], which can be used for online QP-based Markov matrix synthesis. Thus, both methods are applicable to large number of states. As far as the computation speed of the online algorithm is concerned, the speed of Decentralized Counting Algorithm mainly depends on the communication hardware and architecture between agents. QPs on the other hand can be solved in real-time by using recent developments on real-time convex optimization. Example 3.6.2 in the next section provides such an example, where each QP can be solved in $\sim 430\mu$s by using CVXGEN[83] on Intel Core i7 with clock speed of 3.5 GHz.

3.5 Discussion of Connections with Low-Level Motion Control

As explained earlier, the focus of this research is to develop decentralized coordination algorithms based on solving the probabilistic density control problem. Our goal is to achieve physically realizable commanding of the agents by imposing constraints on ergodicity, motion, and safety. Then the low-level algorithms are capable of executing the commands generated by the PDC algorithm, which is illustrated in Figure 3.3. This section aims to make some connections with the low-level motion control, whose detailed analysis is the focus of a future research. Specifically, we identify the key control problems to be tackled (construction of PDC problem and low-level motion control) and identify candidate solution approaches (reachability analysis and decentralized control with collision/conflict avoidance).

The imposition of the ergodicity and motion constraints to facilitate the satisfaction of lower level motion constraints is quite intuitive. Particularly the motion constraints given by matrix A_a define a reachability graph among bins in a single time step. For complicated dynamical systems, the discretization of the state-space and time requires an extensive trade-off. For example, when the set of reachable bins from a given bin is prohibitively limited in terms of mission requirements, the time step length can be increased to enlarge the set of reachable bins. However, large time steps can damage the accuracy of the high-level control policy and can introduce excessive time delays. Another solution is reducing the bin sizes to increase resolution, which may provide better spatial accuracy, but requires more computational resources. There are existing methods that can rigorously certify the validity of these reachability relationships by using finite horizon control problems with agent dynamics, and control and state constraints, e.g., [54, 79, 42], which describe the motions between temporal stages t and $t+1$. These methods can be used in a systematic construction of the PDC problem description with bins, motion constraints, and desired final density distribution based on the decomposition of the region.

A promising method for low-level control utilizing safe high-level coordination is the generation of velocity fields consistent with the density evolution. The idea is that, since high-level coordination controls the density and flow, it can be converted to a velocity field based on the density propagation. Conservation of mass principle can be used to determine the velocity field. An advantage of this approach is that the agents in the same neighborhood will have similar velocities. Also, a consensus policy which averages the velocities can be employed for the agents closer to each other than a prescribed radius, to avoid collisions. A simple one-dimensional example is provided in Section 3.6.4 as a preliminary demonstration of this idea.

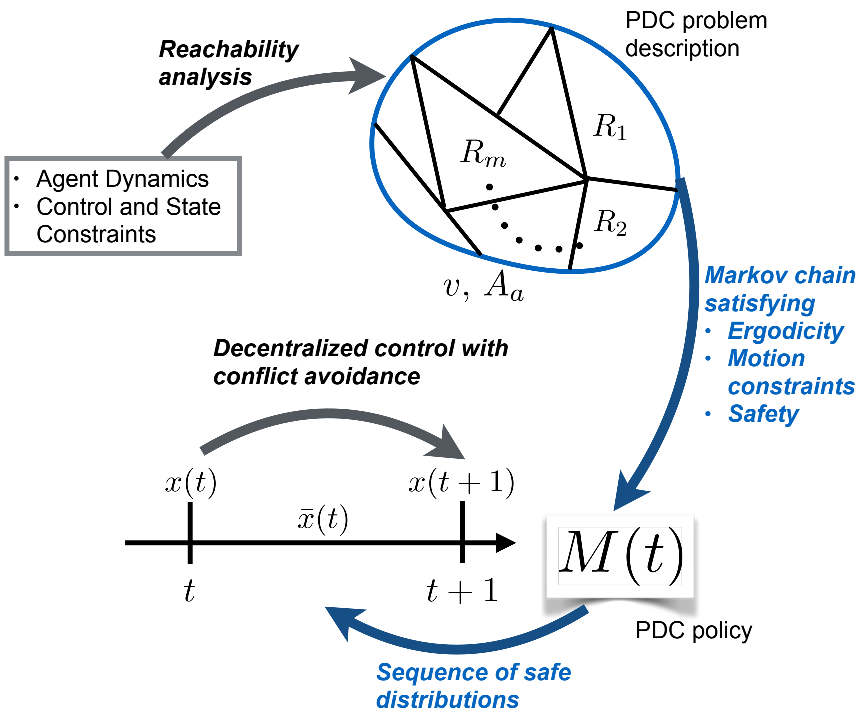

Figure 3.3: Interplay between high-level swarm coordination and low-level vehicle control

The high-level coordination policy can also facilitate existing results in decentralized motion control [89] to ensure collision/conflict free reconfigurations given that the initial and final distributions are "safe". Reference [89] introduces a decentralized problem with Reserved Regions (RRs), where each RR is reserved for a single agent. They propose a decentralized control method with a protocol to avoid collisions for planar motions, which requires safe desired initial and final distributions, e.g., the desired and initial distributions require agent densities with disjoint RRs. In this context, maximum allowable density in a bin is the maximum number of disjoint RRs that can fit into that bin. Our high-level PDC based decentralized coordination ensures a variety of such safety conditions, which can facilitate these earlier results to be combined with the PDC to accomplish the decentralization of both coordination and control with collision/conflict avoidance, which will be the subject of a future study.

3.6 Numerical Examples

3.6.1 Problem setup

We consider a swarm of mobile agents that are distributed over the configuration space $\mathcal{R}$ (see Figure 3.4) that is is assumed to be partitioned as the union of m disjoint subregions $B_i,\ i \in \mathbb{N}_n^+$, such that $\mathcal{R} = \bigcup_{i=1}^{n} B_i$, and $B_i \cap B_j = \emptyset$ for $i \neq j$. This decomposition represents a discretization of the continuous state-space into a finite one. The subregions B_i, i.e., *bins*, represent a discrete state in the state-space and the density distribution over the states evolves according to a prescribed Markov chain. In this configuration, probabilistic density distribution is given as $x[i](t) := \mathcal{P}\{r(t) \in R_i\}$ where $r(t)$ is the position vector of an agent at time step t.

3.6.2 Density Upper Bound and Flow Constraints

In this section we present an example demonstrating the PDC algorithm on a swarm of $N = 5000$ autonomous agents that are distributed on a region R. The region is partitioned to 10 bins and initial distribution of the agents among 10 bins is chosen as,

$$x(0) = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T.$$

The agents are required to converge to the following final distribution:

$$v = [0.01\ \ 0.01\ \ 0.01\ \ 0.01\ \ 0.01\ \ 0.05\ \ 0.05\ \ 0.05\ \ 0.4\ \ 0.4]^T.$$

For this example, the region R is constructed as illustrated in Figure 3.4, and its adjacency matrix A_a is,

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

As it is captured by the adjacency matrix, an agent can travel at most two bins in a single time step and it cannot perform a diagonal transition, e.g. transition from 3rd bin to 5th bin (see Fig. 3.4) is not allowed. For comparison purposes, six different cases are simulated with different parameters which are shown in Table 3.1. Density upper bound constraint described by vector d and flow constraint described by vector α are imposed by using the following parameters,

$$d = [1 \;\; 0.30 \;\; 0.18 \;\; 0.12 \;\; 0.11 \;\; 0.10 \;\; 0.07 \;\; 0.10 \;\; 0.50 \;\; 0.50]^T,$$

$$\alpha = [1 \;\; 0.45 \;\; 0.28 \;\; 0.22 \;\; 0.15 \;\; 0.15 \;\; 0.09 \;\; 0.15 \;\; 0.55 \;\; 0.55]^T.$$

With these parameters, the convex optimization problem given in (3.26) is solved to obtain a constant Markov matrix by using YALMIP and SDPT3 [77, 102]. The optimal Markov matrices are obtained for the cases 1, 2, 3 separately with different set of constraints as described in Table 3.1. Using the result of

Table 3.1: Parameters used in simulations

Case	λ	**Time Step**	**d**	α	**Model**
1	0.955	100	off	off	constant M
2	0.965	100	on	off	constant M
3	0.985	300	on	on	constant M
4	–	100	off	off	time-varying M
5	–	100	on	off	time-varying M
6	–	200	on	on	time-varying M

Proposition 3, since there exist constant Markov chain matrices satisfying the constraints of the problem (3.26) for cases 1, 2, 3, we are able to compute feasible time-varying Markov matrices, based on the density estimates, by solving the QP given by (3.27).

As an example, we provide the Markov matrix M (3.28) used in Case 1, which is obtained by solving the LMI problem (3.26).

$$M = \frac{1}{10^4}\begin{bmatrix} 3730 & 3135 & 3135 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3135 & 2455 & 2455 & 1955 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3135 & 2455 & 2455 & 1955 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1955 & 1955 & 1129 & 2369 & 518 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1798 & 3130 & 1015 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3163 & 4501 & 3285 & 2591 & 2591 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2591 & 1837 & 1837 & 467 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2591 & 1837 & 1837 & 467 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3735 & 3735 & 4241 & 4825 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4825 & 5175 \end{bmatrix}. \tag{3.28}$$

As an illustration, Figure 3.4 shows the density evolution for Case 2 over the prescribed region by using the snapshots of the swarm distribution at selected time instances.

As shown in Table 3.1, three cases with constant Markov chains are first considered. Case 1 has only ergodicity and motion constraints, Case 2 adds on the density upper bound constraint via vector d, and Case 3 adds on both the density upper bound and flow constraints via vectors d and α. These cases are then repeated by using the QP-based synthesis method that uses the density feedback, Cases 4,5, and 6. To compare the cases, the density of each bin and the time history of the overall error, $e_t(t) := \|x(t) - v\|$, are presented. Figure 3.5 shows the time evaluation of the total error for all cases. The full density profile of bins in each case can be seen from the figures that follow next. Note that the time-varying cases do not have a prescribed convergence rate since we

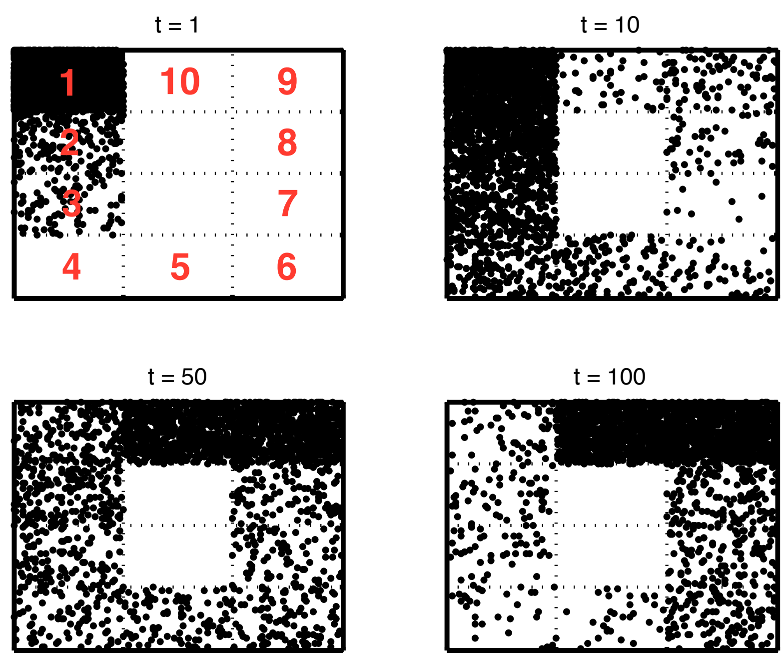

Figure 3.4: Evolution of density distribution in Case 2: Time-varying M with density upper bound constraints.

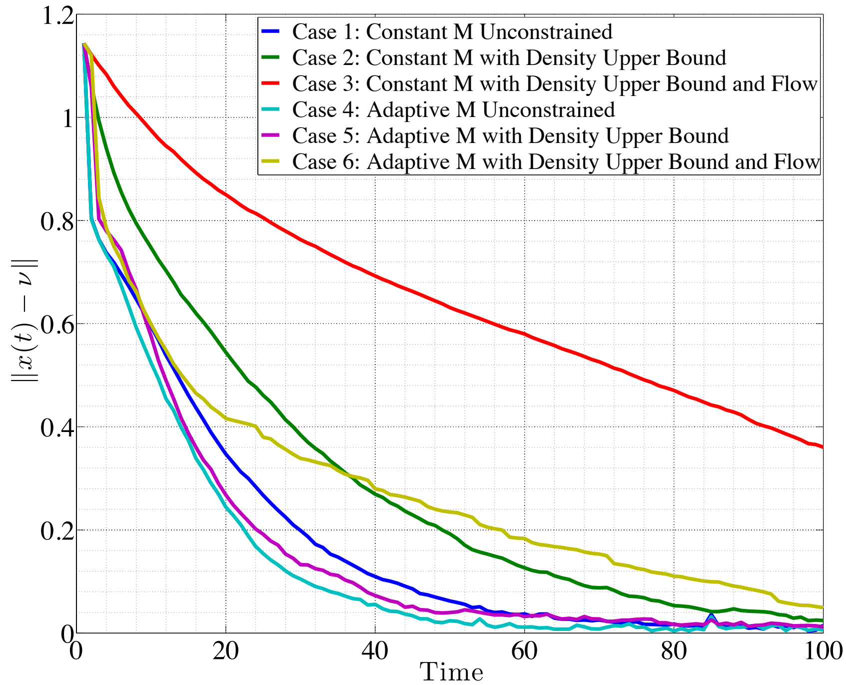

Figure 3.5: Time-history of the total error, e_t, for all cases.

maximize the convergence at each time instance.

As a first observation, Figure 5 shows that the convergence rates for the cases without safety constraints are faster than the corresponding cases with the safety constraints, e.g., Case 1 versus Cases 2-3 and Case 4 versus Cases 5-6. Secondly, all the cases with time-varying Markov matrices, and hence with feedback, have faster convergence to the desired distribution than the corresponding cases with constant Markov matrices, e.g., Case 1 vs. 4, Case 2 vs. 5, and Case 3 vs. 6.

The detailed time histories of the density of each bin are shown in figures 3.6-3.11 for all cases. In each plot, bars with four different colors represent the density of: (i) agents that stay in a bin; (ii) agents that transition into

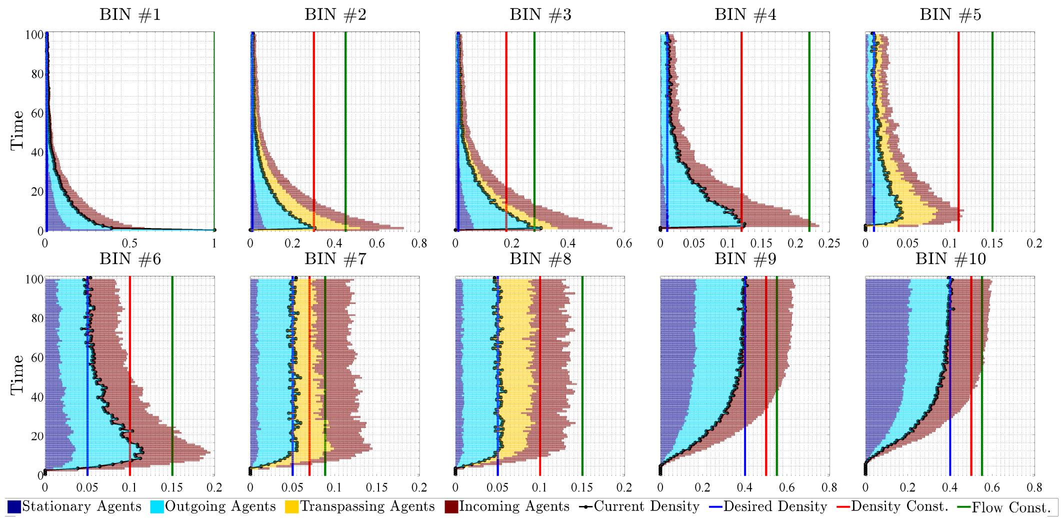

Figure 3.6: Time history of the density of each bin for Case 1

other bins; (iii) agents that pass through the bin; (iv) agents that transition into the bin from other bins. Additionally, the current population density of the specified bin which is a sum of (i) and (ii) are also provided along with the desired final density $v[i]$ for the bin, and the density upper bound and flow constraints $d[i]$ and $\alpha[i]$, $i = 1, ..., m$. Next a detailed discussion of simulation results is provided.

Comparison of Cases 1-4 and 2-5: As it can be seen from the corresponding density plots Figure 3.6 and 3.9, the density bounds are violated when they are not imposed explicitly, in cases 1 and 4. Imposing the density upper bound constraint d in Case 2 and 5, Figure 3.7 and 3.10, leads to the satisfaction of these constraints at the expanse of slower convergence rates as seen from Figure 3.5. As expected, Cases 4 and 5 show faster convergence

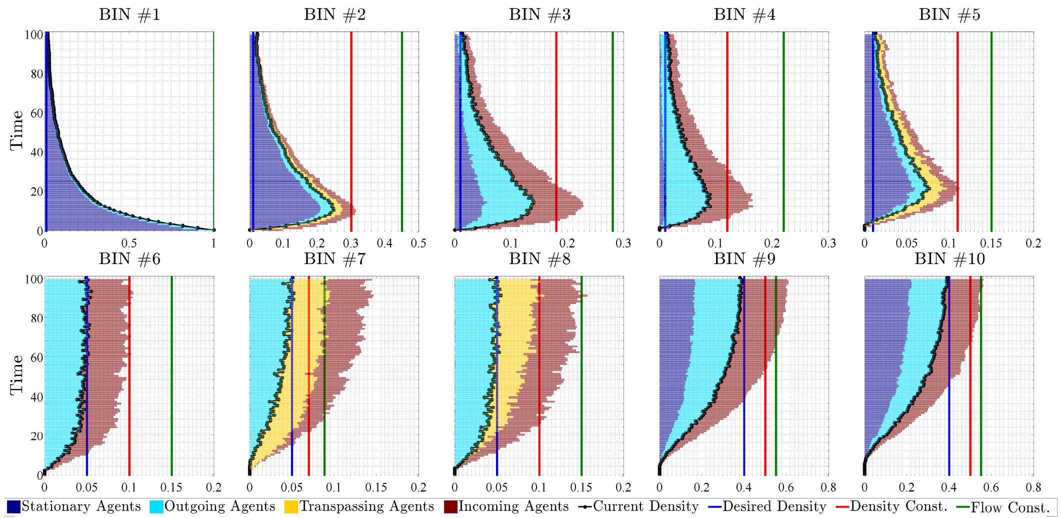

Figure 3.7: Time history of the density of each bin for Case 2

rates than the corresponding constant policy cases, Cases 1 and 2 as observed in Figure 3.5.

Comparison of Cases 2-5 and 3-6: Although the current density in each time step satisfies the density upper bound in the cases 2 and 5, the maximum density $\bar{x}(t)$ violates the constraint α between two consecutive time-steps, which can be seen by accounting for the incoming and trans-passing agents as given in equation (3.21). When the density flow constraints are imposed via α, the maximum density does not violate the prescribed bound α. Similar to the cases with the density upper bound, the time-varying policy results in faster convergence as seen in Case 6, Figure 3.11.

Comparison of Cases 1-2-3 and 4-5-6: The main difference between the cases is that Cases 1,2, and 3 converge slower than Cases 4, 5, and 6. Cases

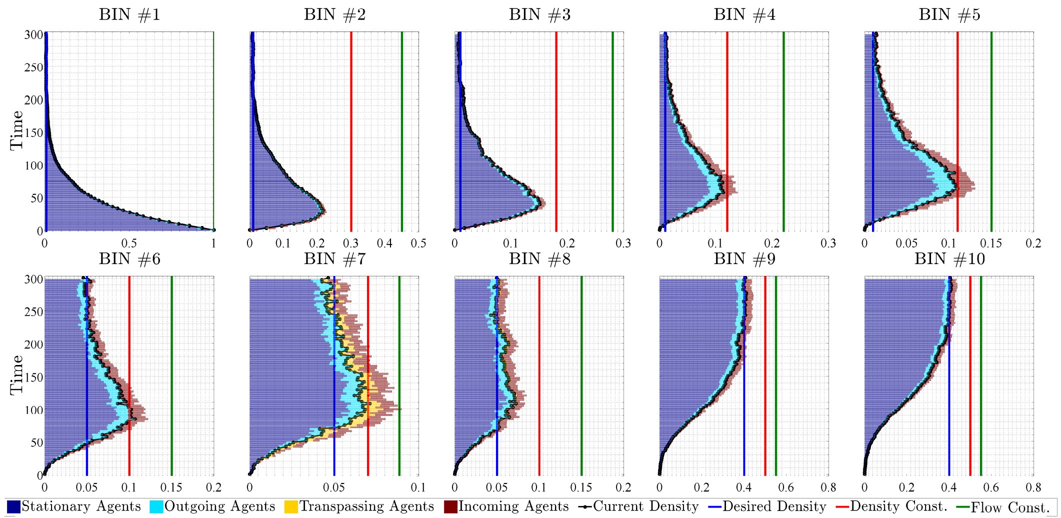

Figure 3.8: Time history of the density of each bin for Case 3

5 and 6 show that updating the Markov matrix in each time step pushes the density to the imposed upper bounds to optimize the convergence rates, while constant policies stay away from these bounds, i.e., they are more conservative.

3.6.3 Density Rate Constraints

In this section we demonstrate the use of density rate constraint, which bounds the change in density of each bin over a time step. This constraint can be utilized especially in swarm deployment process. In this example we use the same parameters with Example 3.6.2 except:

$$v = \frac{1}{1000}[1 \;\; 1 \;\; 1 \;\; 991 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1]^T.$$

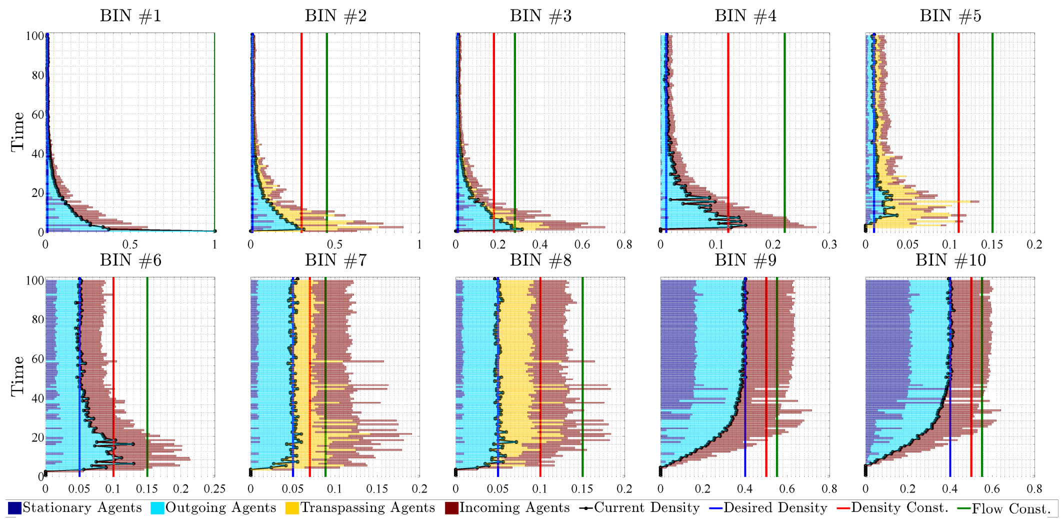

Figure 3.9: Time history of the density of each bin for Case 4

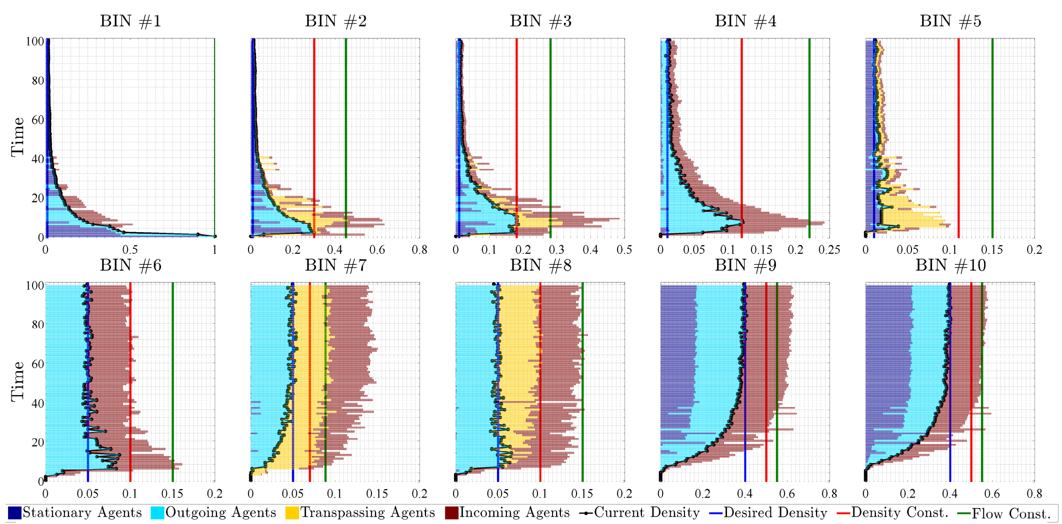

Figure 3.10: Time history of the density of each bin for Case 5

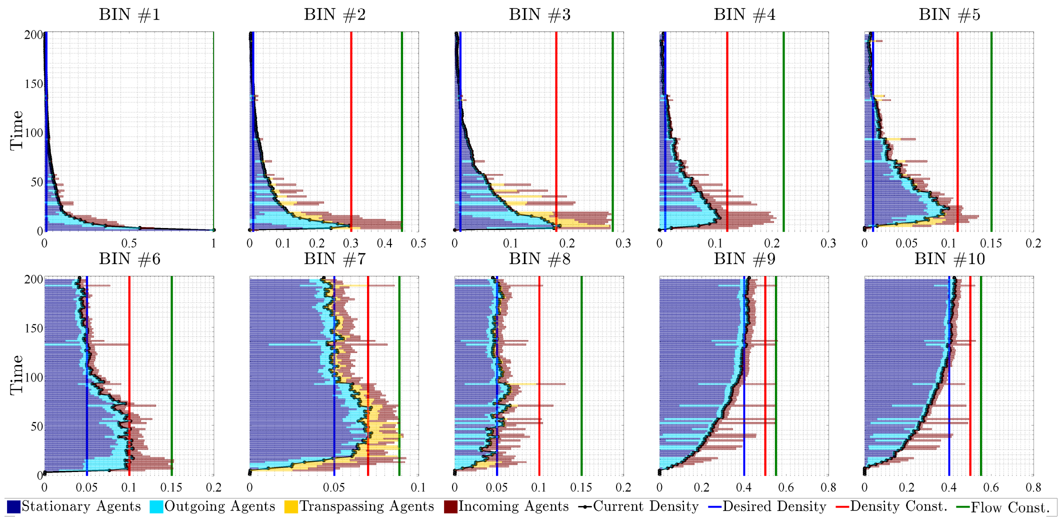

Figure 3.11: Time history of the density of each bin for Case 6

This final distribution can be interpreted as bin 1 is the deployment bin whereas bin 4 is the concentration bin. The following density rate constraint is imposed:

$$f = [0.05 \;\; 0.9 \;\; 0.9 \;\; 0.01 \;\; 0.1 \;\; 0.1 \;\; 0.1 \;\; 0.1 \;\; 0.1 \;\; 0.1]^T.$$

Results given in Figure 3.12 are used to compare cases with and without density rate constraints. Without density rate constraint, bins 1 and 4 has aggressive density rates initially. This situation is eliminated when density rate constraints are imposed.

3.6.4 Simple Example for Constructing Velocity Fields

This section presents a simple example on the construction of a velocity field based on the Markov chain propagation in a 1-D case. For this example, we use the region represented by 5 bins given in Figure 3.13.

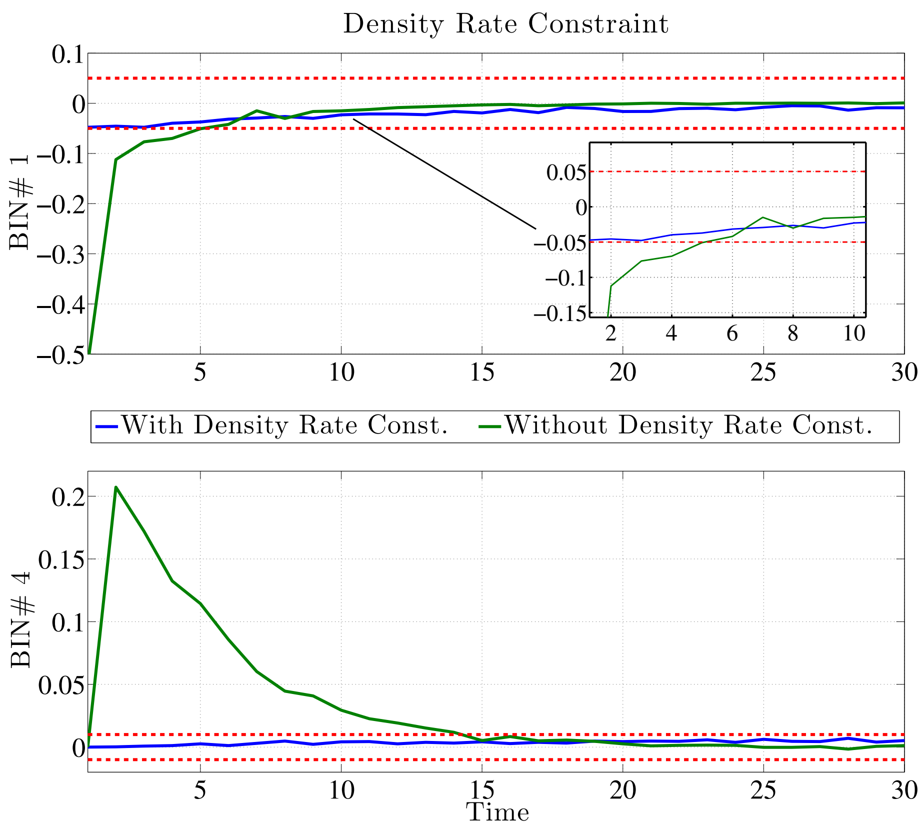

Figure 3.12: Density rate constraint for Bins (1,4).

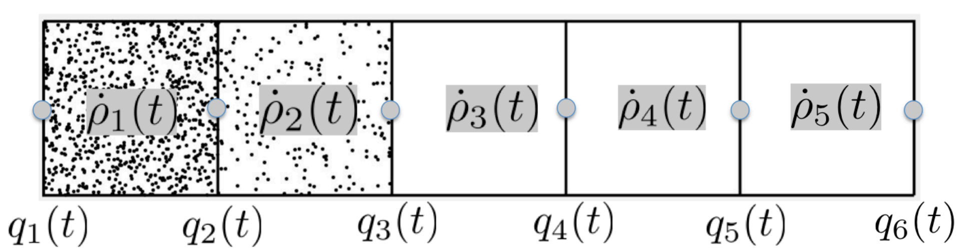

Figure 3.13: Density rate and flux in each bin in 1D.

Based on the idea mentioned in Section 3.5, spatially discretized integral form of the mass conservation equation is written for each bin as:

$$Bin\ \#i \quad \rightarrow \quad \dot{\rho}_i + (q_{i+1} - q_i) = 0, \quad \text{where} \quad q_i = \rho_i v_i.$$

Setting boundary conditions $(q_1(t), q_6(t) = 0)$ yields set of following equations:

$$q_i(t) = -\sum_{1}^{i-1} \dot{\rho}_i(t), \quad i = 2, \ldots, 5.$$

$\dot{\rho}$ values are obtained from Markov matrix as $\dot{\rho}(t) = (M - I)\rho(t)/\Delta t$. Consequently, q_i values are solved from the set of equations and the pair $(q(t), \rho(t))$ is interpolated over the region to calculate the velocity values of each agent based on agent's position with $v(t) = q(t)/\rho(t)$. This method results with smooth motion that reduces the probability of collisions (simply because two agents at the same location must have the same velocity). Additionally, for better collision avoidance, a consensus methodology is employed: A safety radius is defined such that if any agents are within this proximity then they compute their average velocities and tracks this average velocity. This eliminates the possibility of collisions, possibly causing imperfections in reaching the desired steady-state. The results for collision velocity and number of agents involved with/without consensus are given in Figure 3.14. Results presented in the first plot of Figure 3.14 give the normalized velocities of agents that are closer to each other than radius of collision, which is taken as 0.15 units (where square bin side length is 1 unit). The velocities are normalized with the speed of collision radius per time step. Results indicate that even without consensus policy, collision velocities are bounded by 0.1 units per time step. The results also show that consensus policy brings down, as expected, collision velocities

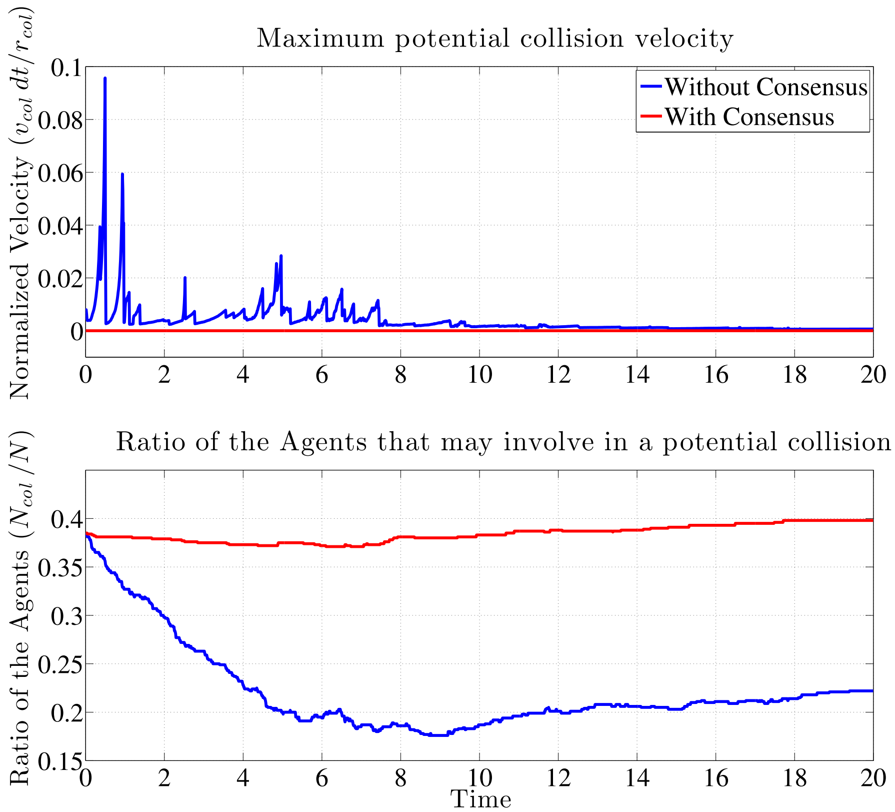

Figure 3.14: Normalized maximum collision velocities and ratio of agents involved with time.

to zero. Since this policy averages the velocities, if agents get too close, they stay close to each other with the same velocity which can be seen from the second plot in Figure 3.14.

Averaged velocities in consensus policy can be considered as small disturbances in the entire flow pattern. The preliminary simulation results given in Figure 3.15 show that, even though it is slightly slower, algorithm converges to the final desired distribution. Therefore, these disturbances are negligible compared the entire flow field.

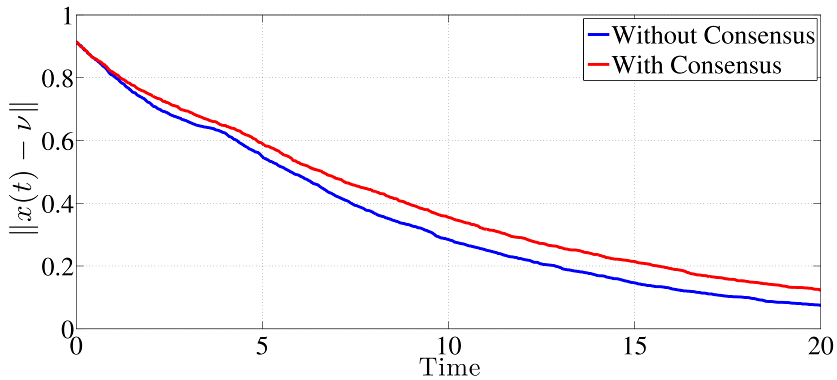

Figure 3.15: Time-history of the density vector error, for with/without consensus cases.

3.6.5 Single Agent Example

In this example we present the results for single agent case ($N = 1$), with M as computed for Case 1 in Example 3.6.2, and Figure 3.16 presents the results of this case. The average time that the agent spends in each bin is computed, with the final bar chart denotes the desired final density distribution. As it is further discussed in Appendix, the probabilistic interpretation of density in a bin is the probability of finding an agent in that bin. The simulation results clearly demonstrate this interpretation: after 1000 time step, the average time that the agent spends in each bin converges to the desired distribution.

3.7 Summary

This section we presented a Markov chain based method for controlling the probabilistic density distribution to coordinate a swarm of mobile agents. The main contribution is to formulate a general form of the safety constraints as the convex optimization constraints for the Markov chain synthesis, which can be used to impose various types of density constraints such as density upper

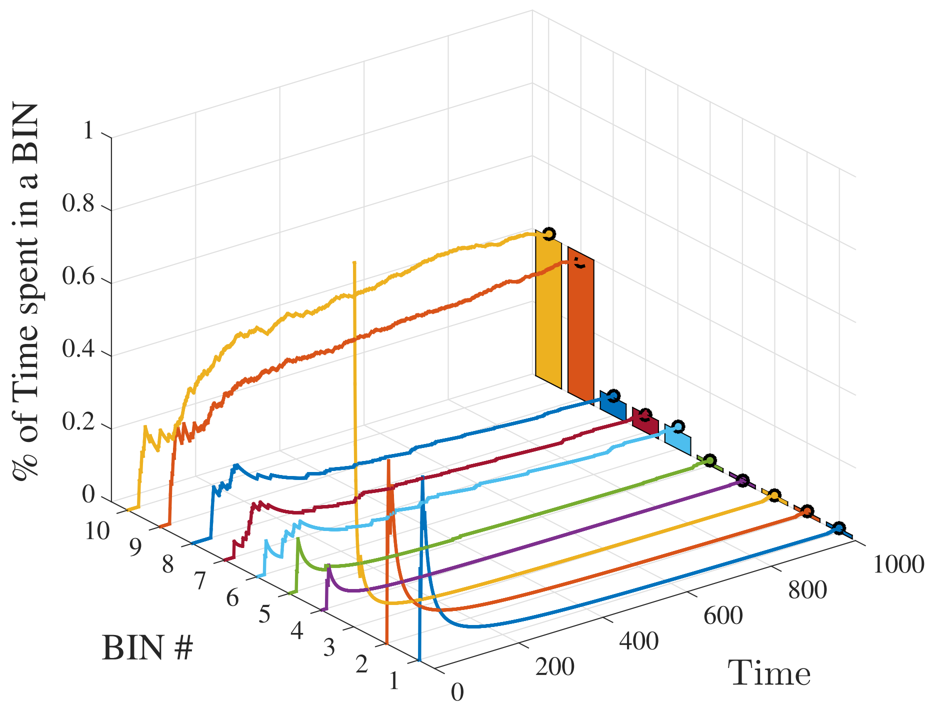

Figure 3.16: Percentage of time spent in each bin for single agent case

bound constraints and density flow constraints. A new synthesis method is introduced for time-varying Markov matrices via a quadratic programming based online synthesis method that utilizes the real-time swarm density estimates as feedback, and constantly updates the Markov matrix. This approach produces better convergence than the case with no density feedback, which is also illustrated in a numerical example. We also introduced a decentralized counting algorithm for density estimation, which converges to identical estimates for all agents in a predetermined number of communication updates.

Chapter 4

LEO SWARM SIMULATIONS WITH DENSITY CONTROL

This section presents the Low Earth Orbit (LEO) swarm simulations with density control. The objective is to deploy large number of agents into space to form a shape where they satisfy prescribed safety constraints throughout the mission. The simulations are also conducted to verify the density control algorithms' reliability and to reveal the resource requirements for a swarm.

The density control with Markov chains is so-called *high level method* that considers a high-level abstraction of the system. Hence, there is a need for a low-level method to describe the individual agent commands to achieve the desired collective behavior. To this end, we consider the generation of velocity fields consistent with the density evolution imposed by the Markov chain. The idea is that, since high-level coordination controls the density and flow, it can be converted to a velocity field based on the density propagation. An advantage of this approach is that the agents in the same neighborhood will have similar velocities. The preliminary formulation is introduced in section 3.6.4 and generation of velocity fields is an ongoing work done by Eren [47]. Here, we use this method to make comparisons with the case where positions are determined in a randomized fashion.

4.1 Low Earth Orbit (LEO) Swarm Simulations with Density Control

This section presents simulation results for probabilistic density control method applied to a Low-earth orbiting swarm in 2D. The idea is adopted from ([57]) where the probabilistic guidance algorithm is adapted to earth orbiting swarms using Hill's equations for the relative dynamics of each spacecraft to the circular orbit defined in local-vertical local horizontal (LVLH) coordinates. The idea behind this approach uses a mapping from LVLH coordinates $(x(t), y(t), z(t))$ to the coordinates in a 2D configuration space (Scaled-Rotated (SR) coordinates), $(r(t) = \bar{x}(t), \bar{y}(t))$ where the motion of the spacecraft is determined via Markov chain. We use the same problem setup given in Section 3.6.1 where the density distribution is defined with position of each agent.

With this setting, we consider a low earth orbiting (LEO) swarm where the relative dynamics of each spacecraft to the circular orbit are given by the following nonlinear equations with gravitational effects: [107]

$$\begin{aligned} \ddot{x}_j &= 2\omega\dot{y}_j - x_j(\eta_j^2 - \omega^2) - r(\eta_j^2 - \omega^2) + f_x \\ \ddot{y}_j &= -2\omega\dot{x}_j - y_j(\eta_j^2 - \omega^2) + f_y \end{aligned}$$

where $\omega^2 = \mu/r^3$, $\eta_j^2 = \mu/r_j^3$, μ is Earth's gravitational constant, r is the geocentric distance of orbit reference point and

$$r_j = \sqrt{(r + x_j)^2 + y_j^2 + z_j^2}$$

geocentric distance of agent j.

For control analysis, we also consider Hill's equations [97] that describes a simplified model of orbital relative motion with respect to a reference point in the circular orbit and are also referred to as Clohessy-Wiltshire equations:

$$\begin{aligned} \ddot{x}_j &= 2\omega\dot{y}_j + 3\omega^2 x_j + f_x \\ \ddot{y}_j &= -2\omega\dot{x}_j + f_y \end{aligned}$$

We refer to this model as linearized dynamics. The analytical solution of the Hill's equations may result in trajectories in the form of a stationary points, lines, ellipses, spirals, etc.,depending on the initial conditions. If the initial conditions satisfy:

$$y_j(0) = -2\omega x_j(0)$$

then the trajectory will be periodic with a period as the same as period of nominal orbit. When this condition is satisfied, the spacecrafts are *period matched* with parameter ω [57]. If, in addition, the initial conditions for each spacecraft satisfy:

$$y_j(0) = \frac{2}{\omega}\dot{x}_j(0)$$

then the trajectory is elliptical that is centered at the origin and the spacecrafts are *centroid matched*.

The time dependent coordinate transformation to the new set of position coordinates in the Scaled Rotated (SR) coordinates are defined as follows:

$$\begin{bmatrix} \bar{x}(0) \\ \bar{y}(0) \end{bmatrix} = R^T(\omega t) S^{-1} \begin{bmatrix} x(t) \\ y(y) - y_0 \end{bmatrix}$$

where

$$R(\omega t) = \begin{bmatrix} cos(\omega t) & sin(\omega t) \\ -sin(\omega t) & cos(\omega t) \end{bmatrix}, \; S = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

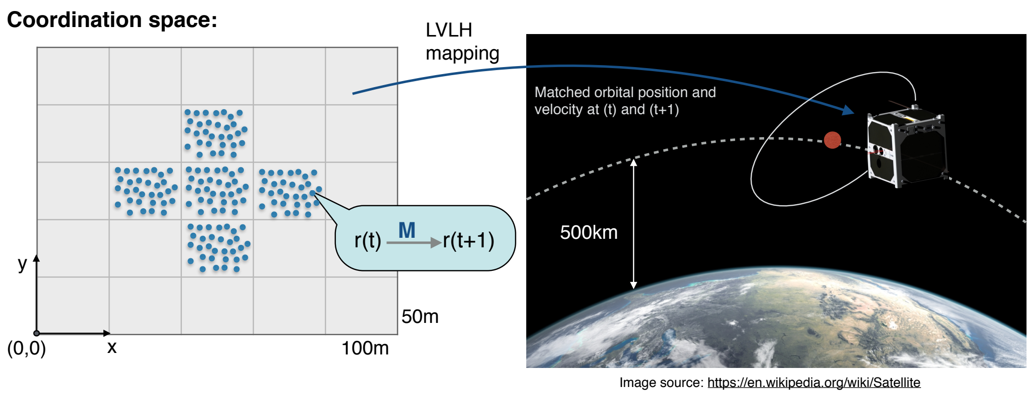

Figure 4.1: LVLH mapping between the coordination space and SR coordinates

The idea is to solve the distribution problem in the configuration space via synthesizing a Markov chain with the configuration space parameters. Then the swarm distribution in the LVLH coordinates will be rotated and stretched version of the distribution in the configuration space based on the time-varying mapping between two coordinate systems. At each Markov time step, the initial and final states of the agents are determined along with an optimal trajectory; all these parameters are mapped to LVLH coordinates and control inputs are computed for agents to follow the desired trajectory. Computation of control inputs and implementation are discussed in more detail in the next section.

4.2 Simulation Results

For the simulation we considered two different cases: (i) Simple configuration space with rectangular bins and (ii) More complex configuration space with triangular bins. In both cases, a low-earth circular orbit with $500km$ distance to earth is considered. Initially agents are distributed randomly according to given initial density distribution. Then each agent find its next bin based on Markov chain and choose its location either randomly within that bin or follow the trajectory based on the velocity field. Corresponding orbital positions and velocities are found using LVLH mapping. Then with the given dynamics, corresponding input forces are computed for each agent to follow a trajectory from given initial state to final state and this is done for each Markov step.

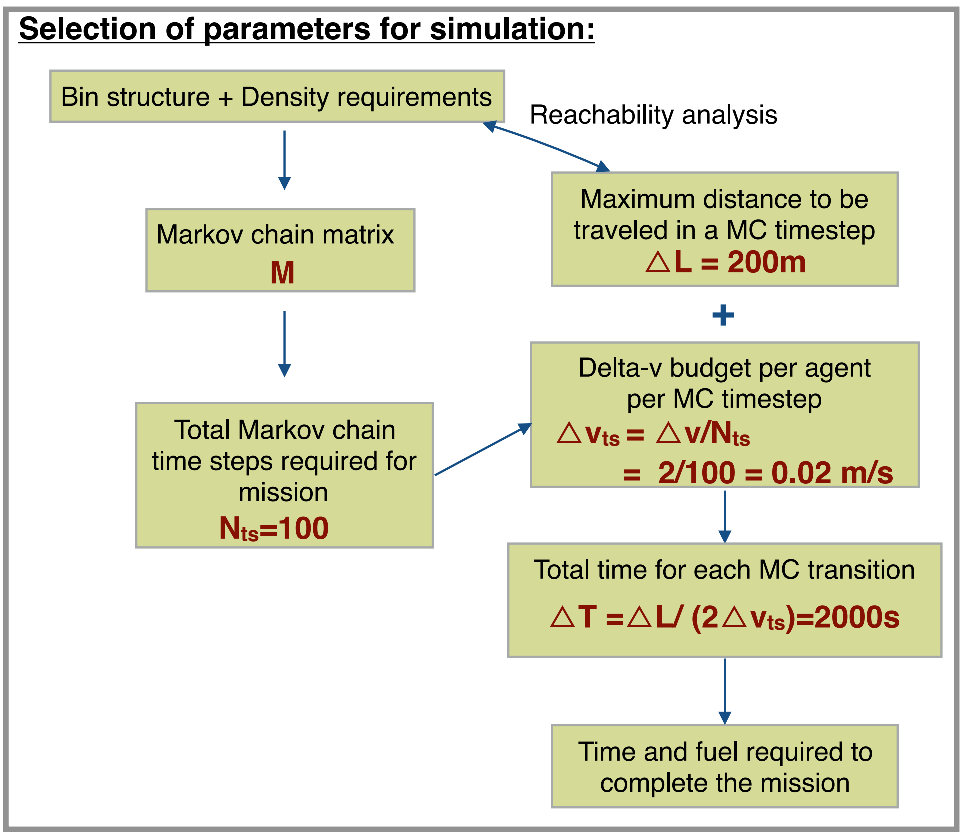

Figure 4.2: Steps to determine simulation time for a given structure and requirements

4.2.1 Configuration space with rectangular bins

In this section we consider the configuration space ($\mathcal{R}$) with rectangular bins that is shown in Figure 4.4. The simulation parameters (adjacency structure, upper bounds, desired densities) are shown in the figure. $N = 1000$ number of agents are used for the simulations. The simulation time spent in a single Markov step is chosen to be 2000 seconds based on the delta-v budget (2m/s) and total Markov chain time steps required for convergence (100) given the Markov chain parameters (see Figure 4.4). The steps to determine suitable simulation time is shown in Figure 4.2. We also ran a sample simulation with smaller simulation time to see the effect on total delta-v and fuel consumption. Overall, we ran the simulation for the following cases:

- Open-loop density control
  - Impulse control with delta-v
    - Linearized vs. Nonlinear dynamics
    - Velocity field vs. Randomized positions
  - Continuous control and optimization based guidance
    - Nonlinear dynamics and velocity field
- Density Control with feedback (Update MC with density feedback for above applications)

In the case impulsive control, at the beginning and at the end of in each simulation segment, an instantaneous delta-v is applied to each agent. The required delta-v is determined based on the linearized system. Then, we generalized this case to continuous control where the acceleration input is applied to system for a given time interval again at the beginning and at the end of each

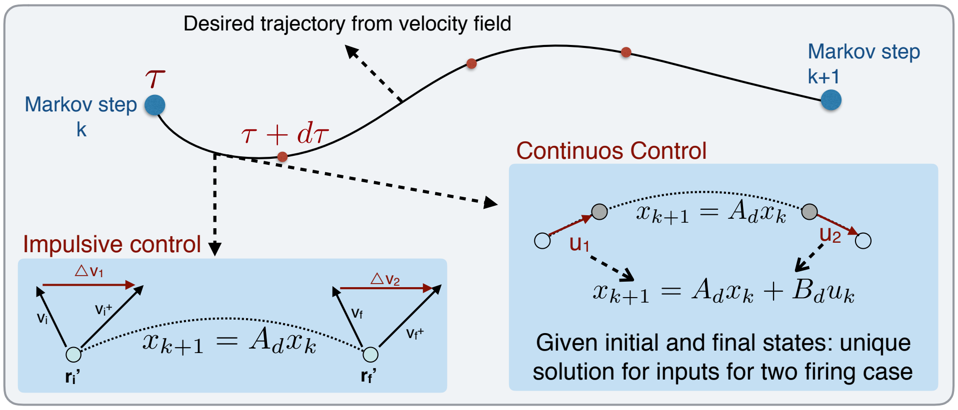

Figure 4.3: Impulsive vs. continuous control to follow desired trajectory

simulation segment (see Figure 4.3). The continuous control case is simulated both with linear and nonlinear equations. As expected, with the linear dynamics the agents followed the trajectory perfectly while with nonlinear dynamics, the agents had some deviation from the reference trajectory. This deviation is reduced by applying additional input based on feedback. For this case we also compared the performance of velocity fields to randomized positions where at each Markov step, the position of an agent is determined in a randomized manner within the objective bin. The results of all simulations are summarized in Table 4.1.

For a sample case, we showed the density history for each bin in Figure 4.6 for bounded bins which also shows the evolution of density when upper bounds are not imposed. The snapshot of the simulation taken at different time instances are shown in Figure 4.5.

Based on the simulation results presented in Table 1, it can be seen that total delta-v and fuel consumption is higher for nonlinear dynamics as expected. Also, when we decrease the simulation time for the linear dynamics, the fuel

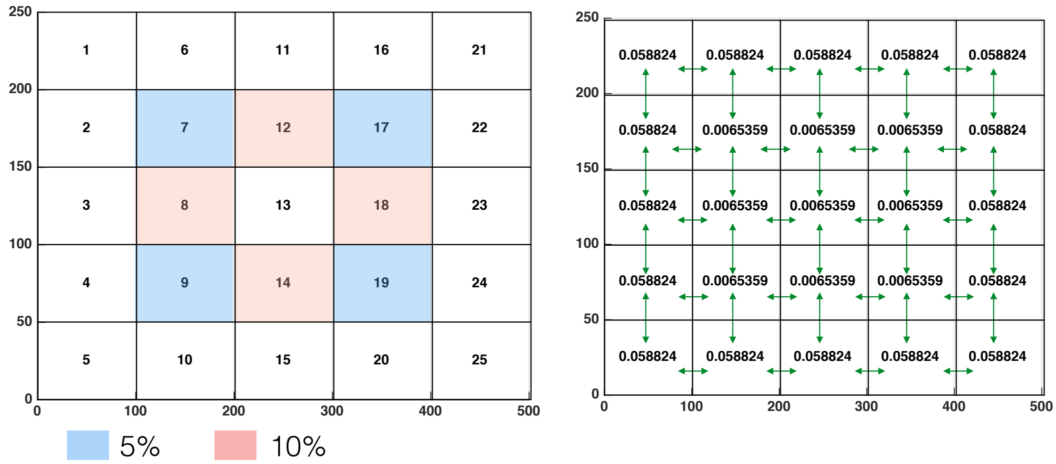

Figure 4.4: Configuration space $\mathcal{R}$ with rectangular bin structure. Left figure shows the bin numbers and the upper bounds for bounded bins (labeled with colors). Right figure shows the allowable transitions and desired densities for each bin.

consumption increases and this amount seems to be directly proportional to decrease in time. But this may not be the case for nonlinear dynamics since increase in simulation time requires more input to keep the agent in the desired trajectory. Another thing to note is that, using randomized positions requires less fuel since it only impose initial and final states and does not force agent to follow a specific trajectory. For the space swarm, using randomized positions does not cause a high collision count as the configuration space is big, but for the cases when the configuration space is small this method is expected to perform worse compared to velocity field method.

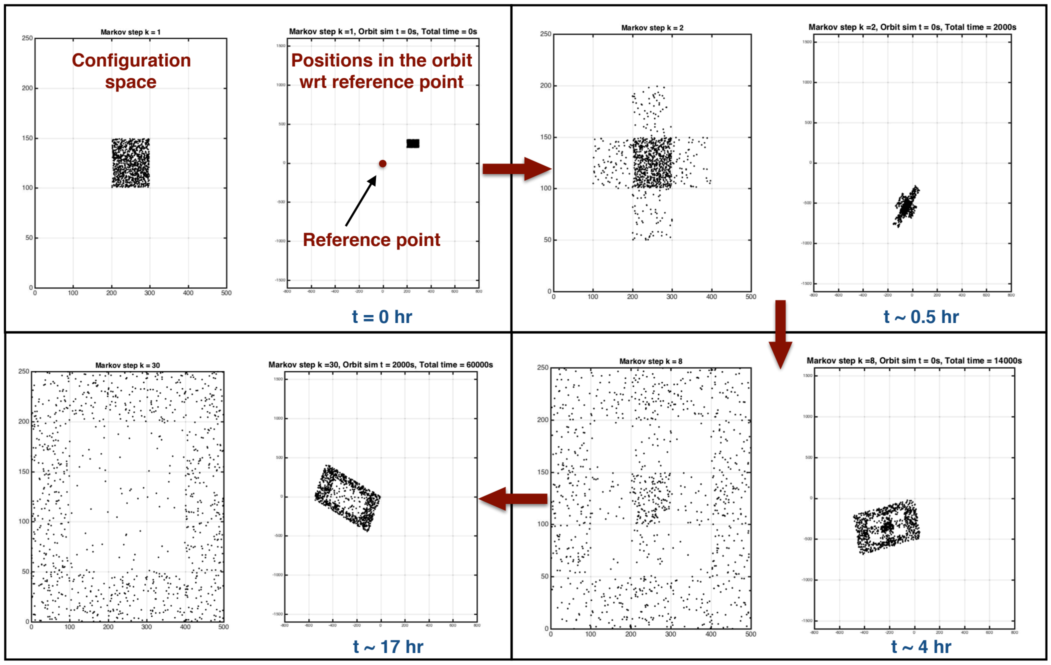

Figure 4.5: Snapshots of the simulation taken at different time instances

4.2.2 Configuration space with triangular bins

In this section, we consider a more complex configuration space with triangular bins. Similar to previous example, the simulation parameters (adjacency structure, upper bounds, desired densities) are shown in the figure. $N = 1000$ number of agents are used for the simulations. For this case we considered velocity fields with continuous control and ran the simulations with linear and nonlinear dynamics.

Figure 4.8 shows the snapshots of a sample simulation. The agents start in the middle 6 bins (hexagon) and try to distribute themselves on the boundary bins to form a mirror shape. As we imposed upper bound constraints on the neighboring bins to center bins, the number of agents are kept small in

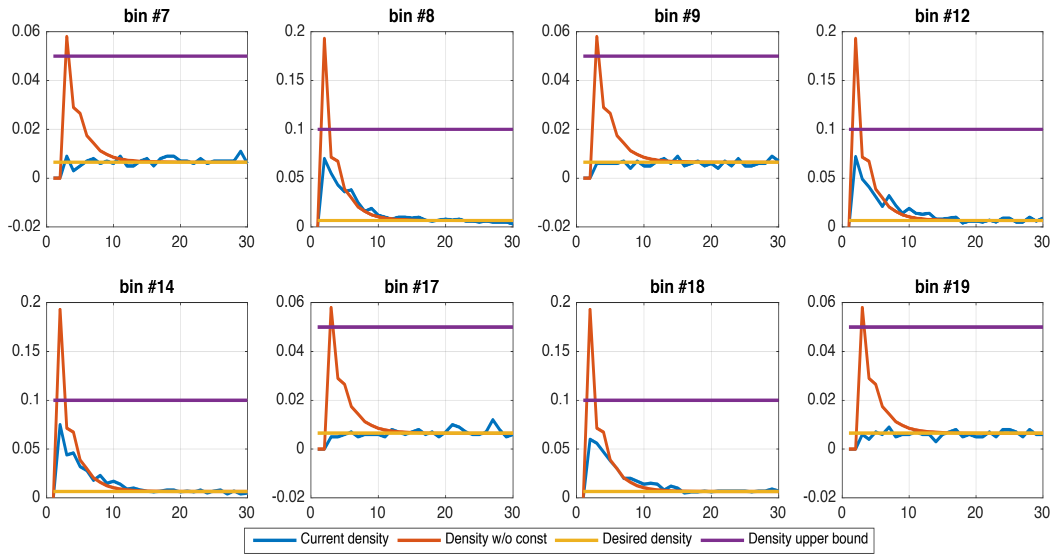

Figure 4.6: Density history for the bins for which there is an upper bound. Figure also shows the density when upper bound is not imposed.

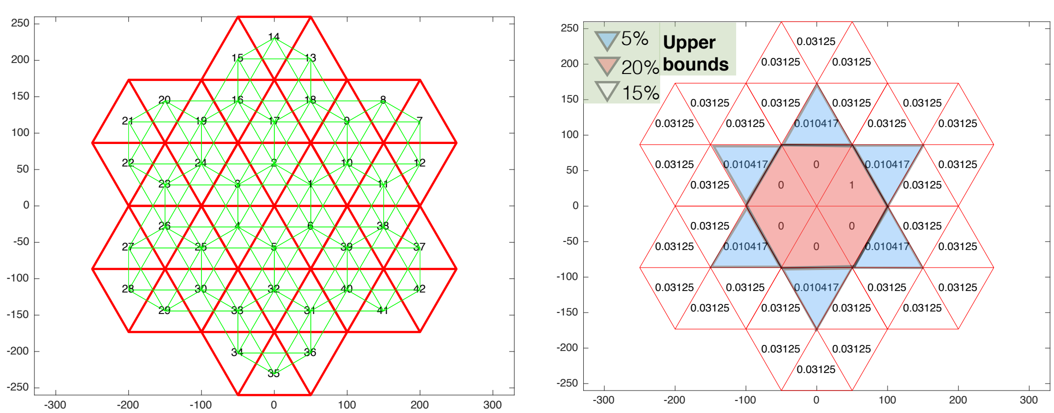

Figure 4.7: Configuration space $\mathcal{R}$ with rectangular bin structure. Left figure shows the bin numbers and the upper bounds for bounded bins (labeled with colors). Right figure shows the allowable transitions and desired densities for each bin.

Table 4.1: Summary of simulation results for the case with rectangular bins

Case	Total Mission Time	Total delta-v usage	Total fuel usage when converged	Avg. Collisions after 5th step
Randomized-Impulsive-Linear	∼11hrs	0.4 m/s (same w/ feedback)	2.2 kg (same w/ feedback)	∼8
Randomized-Continuous-Linear	∼11hrs	0.4 m/s (same w/ feedback)	2.2 kg (same w/ feedback)	∼10
Randomized-Continuous-Nonlinear	∼11hrs	2 m/s	10 kg	∼4
Velocity field-Continuous-Linear	∼11hrs	1.4 m/s	7 kg	0
Velocity field-Continuous-Nonlinear	∼4.5hrs	2.1 m/s	11 kg	0

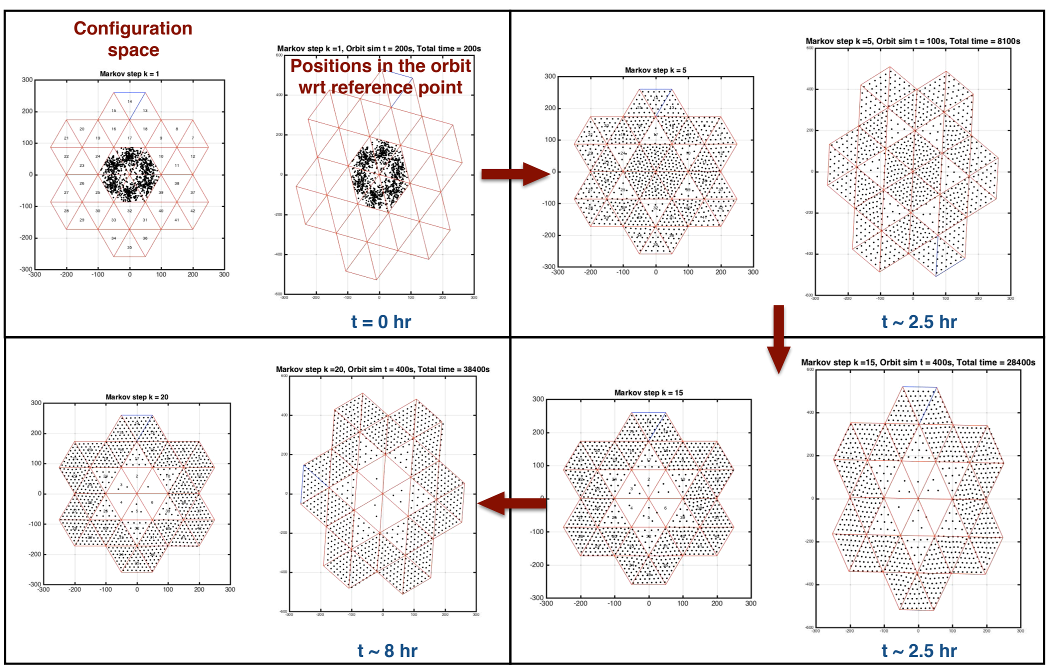

Figure 4.8: Snapshots of the simulation taken at different time instances

those bins throughout the simulation preventing agents to disperse rapidly. Figure 4.9 show the comparison between the results of linear and nonlinear dynamics. Similar to the rectangular bin example, the case with nonlinear dynamics has a higher fuel consumption. In both cases, there is no collisions after 2 Markov chain time steps. The high collision count at the beginning is caused by randomized initialization.

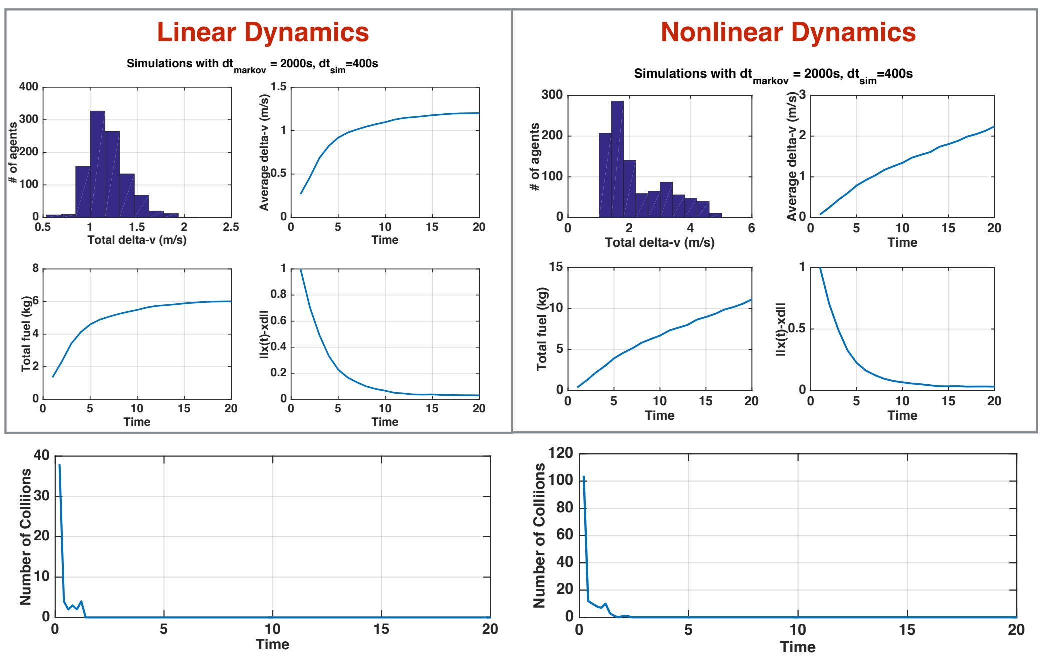

Figure 4.9: Simulation results: Comparison of linear and nonlinear dynamics

For this configuration space we also consider the problem of deployment and agent loss. For the deployment simulation, we deployed 50 agents at the beginning of each Markov step during first 20 steps until the desired number of agents is reached. The simulation instances are given in Figure 4.10. The convergence is achieved in 30 steps. Resulting delta-v and fuel consumption are shown in Figure 4.11. In order to test the robustness to agent failure, we ran a simulation where the agents in 6 different bins are removed at Markov step 10 as seen in Figure 4.12. The system successfully recovered and converged to desired density without any collisions.

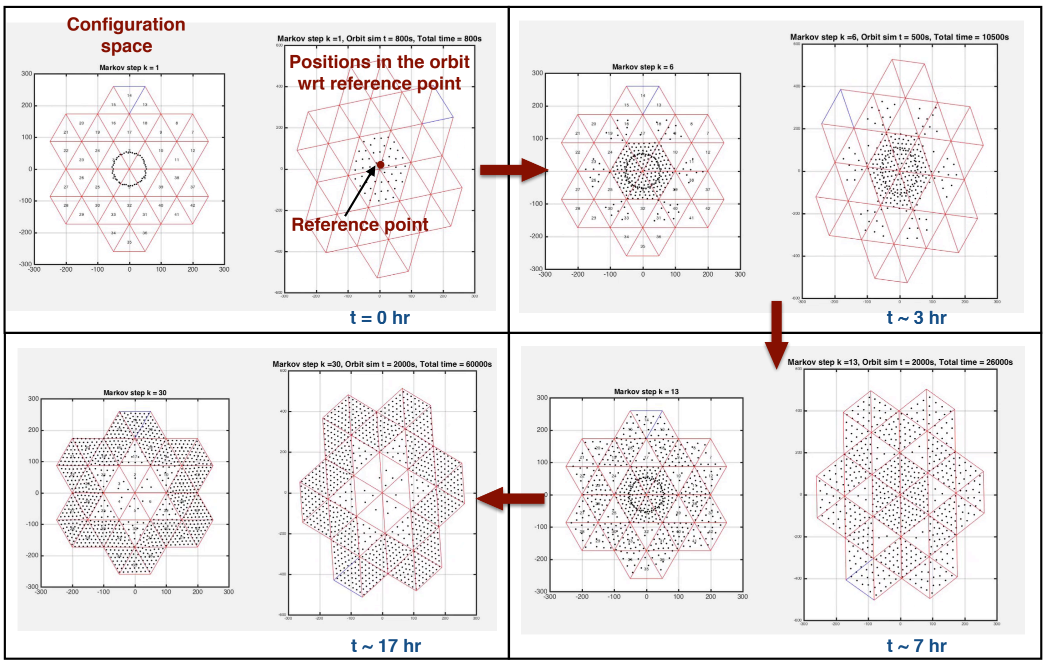

Figure 4.10: Deployment simulation: Snapshots taken at different time instances

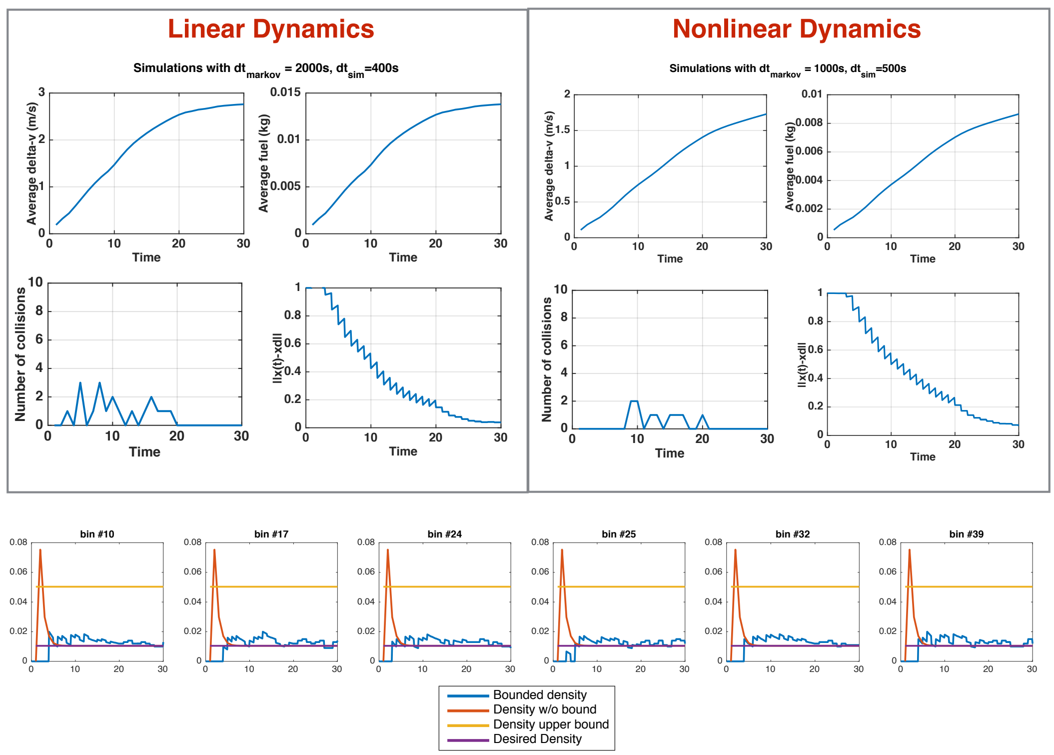

Figure 4.11: Simulation results for the deployment problem

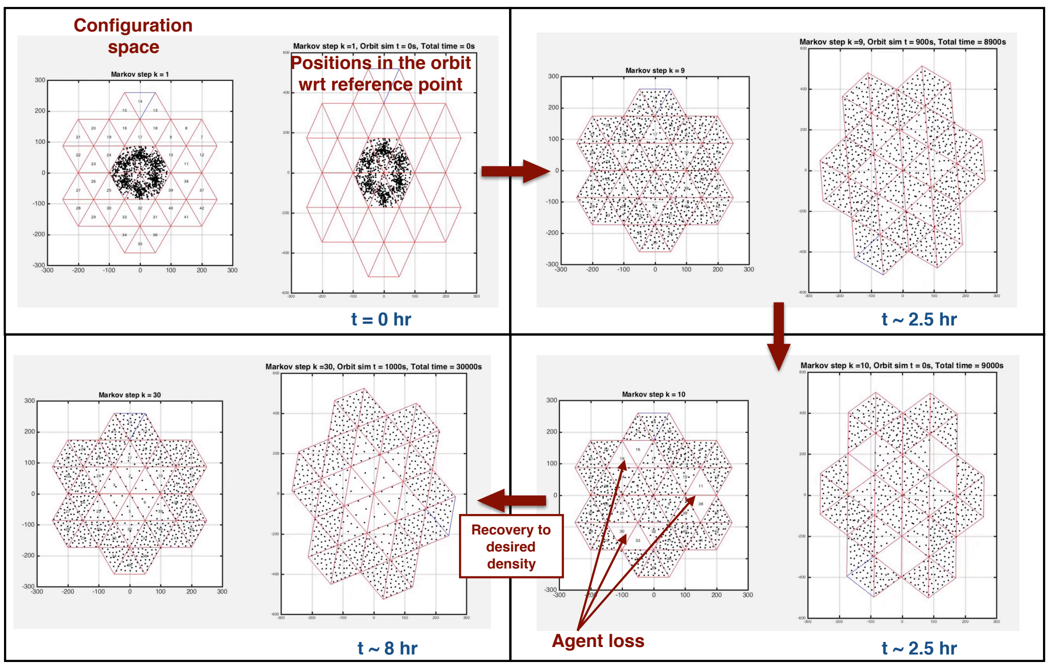

Figure 4.12: Snapshots of the simulation with agent loss taken at different time instances

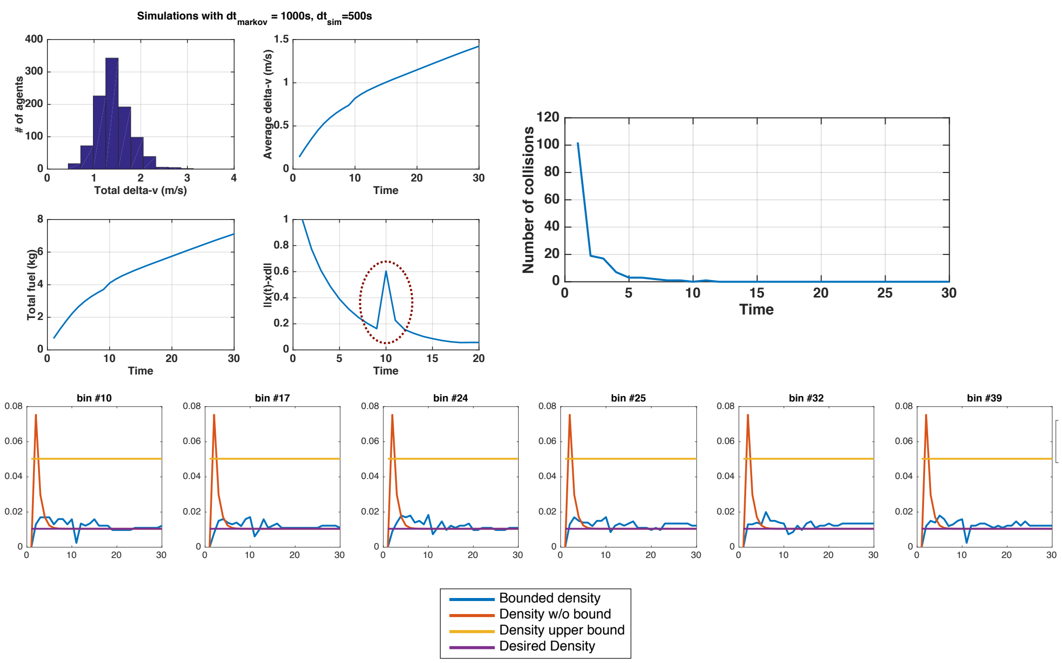

Figure 4.13: Simulation results with agent loss

Chapter 5

DENSITY CONTROL OF ON/OFF AGENTS

This section extends the probabilistic density control problem defined in Chapter 3 to swarm of autonomous ON/OFF agents. In this configuration, the mobile agents switch between two modes: ON and OFF. Each mode consists of a (possibly overlapping) finite set of actions, that is, there exist a set of actions for the ON mode and another set for the OFF mode, which may have a non-empty intersection. When the agents are in the ON mode, they move with the environmental forces such as, winds, sea currents, gravity, which assumably can be modeled as a stochastic environment with transitional probabilities on a discretized finite state-space. The density distribution of the agents is determined by the state-probability of an underlying discrete-time Markov chain. As the main objective is to control the density of the swarm, we formulate the density control problem as synthesis of a Markov chain which, as we show in Theorem 4 and Theorem 6, is a linear function of stochastic environment with transitional probabilities and the ON/OFF decision policy.

The proposed model has potential applications for autonomous systems operating in complex dynamic environments like flow fields where both historical and observation based data can be utilized. These applications include density control of blimp type UAV's [68], wind-energy based UAV's like gliders [4], underwater vehicles [104], etc. An interesting example to illustrate the concept of ON/OFF agents is controlling air balloons in uncertain wind fields [105] for scientific measurements in Earth and other planets [71, 46]. Proba-

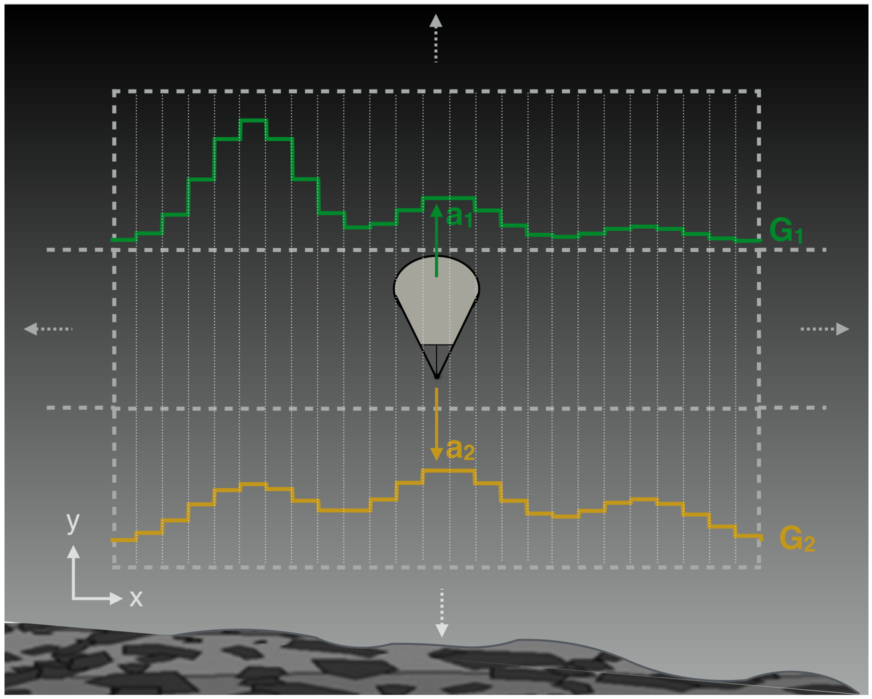

Figure 5.1: A simple 2D air balloon illustration with actions, a_i. G_i is the discrete probability density distribution for the x-position of the balloon resulting from taking action a_i from its current position. In this example, the balloon observes outcome of an action by changing its altitude based on the action, and the OFF mode is an MDP [93] running over all actions, hence G_{off} is a function of G_i's, i.e., $G_{\text{off}} = f(G_1, G_2, ...)$.

bilistic wind velocity field information, i.e., the probability density functions for the wind speed and direction, changes as a function of the balloon's altitude. The hot air balloons can change their altitudes to choose the velocity field that they ride with, i.e., horizontal motion induced by the wind. Previous research proposed controlled Markov process models for this example [105]. In the ON/OFF approach, the action in this Markov process model is the choice of altitude (see Figure 5.1). Then a default Markov Decision Process (MDP) policy can be synthesized to distribute the balloons based on this prior wind field knowledge (transition probabilities as a function of the altitude), which defines the OFF mode. Also an ON policy can be designed, for which the instantaneous velocity observed at different altitudes can be measured. The selected altitude (action) for the ON mode can be accepted or rejected based on the instantaneous velocity observed at this altitude. If it is rejected based on the observed outcome, we can go to the altitude suggested by the OFF mode policy. Note that, the sets of actions for the ON and OFF modes are the same in this example.

In this section, first we introduce the model for a simpler case with single action in the ON mode and a deterministic action in the OFF mode. Then we generalize this model to case with multiple actions in the ON mode and stochastic action OFF mode. Both models are illustrated with a numerical example. We also make connections to Markov Decision Processes.

5.1 Single Action in the ON Mode and Deterministic OFF Action

This section presents the simple model of mode-switching ON/OFF agents to demonstrate the concept and formulation [33]. For this simple model, the ON mode has a single action and the OFF mode has identity as the transition matrix (deterministic transitions). Then, we generalize this model in the next section. In both general and simplified models, agents are assumed to know their current states. For the simplified model, we consider the case where the ON mode contains only one action for which the action outcome is observable, i.e., agents can measure their next state at a given instant if they take the action (i.e., they decide to be in the ON mode). However agents do not have this transitional observation for the single action in the OFF mode. For ease of demonstration of the idea, we assume that the stochastic transition matrix for the action in OFF mode is an identity matrix. Note that having an identity matrix means that the outcomes of the action for the OFF mode are deterministic (they do not require measurement of the transitions), which is a trivial case and it will be generalized much further in the next section. This assumption makes the initial formulation more transparent. Furthermore, this simpler model also captures an interesting interpretation of the density control problem as presented in [33], that is, ON/OFF agents are moving with the dynamics induced by the stochastic environment when they are ON and they are staying stationary when they are OFF. This model also captures an interesting interpretation of the density control problem as presented in [33], that is, ON/OFF agents are moving with the dynamics induced by the stochastic environment when they are ON and they are staying stationary when they are OFF. This problem is inherently more challenging since the agents have very limited mobility. More precisely, they have control over a binary decision variable, which determines whether they are ON or OFF.

The density control problem for this simplified model can be formulated as a Markov chain synthesis problem. For that, the same definitions for the state-space, probabilistic density distribution and the Markov matrix as given in (2.4) and (2.6) are used.

Differently from the density control problem described in Chapter 3, here the transition matrix M is a function of the stochastic environment and the ON/OFF policy, as will be explained next. We define the following events to properly define the stochastic environment and the decision policy, for $t \in \mathbb{N}$:

$$y(t+1) = s_l : \text{ Observing a transition to state } s_l$$

$$\sigma(t) = \sigma_{on} : \text{ Accepting to execute the action for the ON mode.}$$

Note that there are two *modes* of operation $\sigma(t) \in \{\sigma_{on}, \sigma_{off}\}$. Even though $y(t+1)$ has a time index $t+1$, this observation occurs at time t. In particular, observation of a transition is different from the actual transition taking place. For example, $y(t+1) = s_l$ is the event that the stochastic environment would have caused a transition to l^{th} state at time $t+1$ if the action for the ON mode were to be accepted at time t (i.e., observing one-step ahead in the future), whereas $s(t+1) = s_l$ is the event that the transition to l^{th} state has actually occurred.

The stochastic environment is defined with the transition matrix $G \in \mathbb{P}^{n\times n}$, where $G[i,j]$ is the probability of observing a transition from j^{th} state to i^{th} state when the action for the ON mode is taken, i.e.,

$$G[i,j] := \mathcal{P}\{y(t+1) = s_i | s(t) = s_j\}. \tag{5.1}$$

Here, the environment transition matrices and the decision policy are assumed to be time-invariant (i.e., the processes are stationary), hence the corresponding Markov chain transition matrix given in (2.6) is also time-invariant. The objective is to synthesize a decision policy for an agent to accept or reject the corresponding transition observed for the action in the ON mode at each time epoch, i.e. to decide whether it should be ON or OFF, such that the resulting Markov chain will satisfy the desired transition and safety constraints while guiding the density distribution to a desired final distribution. In this section, the state of the agent is assumed not to change if it is OFF, i.e., $s(t+1) = s(t)$ when $\sigma(t) = \sigma_{off}$. Then the probabilistic ON/OFF decision policy is defined by a matrix $K \in \mathbb{R}^{n\times n}$ (to be designed) that satisfies:

$$\mathbf{0} \leq K \leq \mathbf{1}\mathbf{1}^T, \quad \mathrm{diag}(K) = \mathbf{1}, \tag{5.2}$$

where $K[i,j]$ is the probability of being ON, given that the transition from j^{th} state to i^{th} state is observed, i.e., the acceptance probability of the environmentally induced motion determined by G:

$$K[i,j] = \mathcal{P}\{\sigma(t) = \sigma_{on} | s(t) = s_j, y(t+1) = s_i\}. \tag{5.3}$$

Note that for the transition $i \to i$, $\forall i$, accepting or rejecting the transition corresponds to the same outcome in this case (in this section). Hence, the diagonal elements of acceptance matrix K are set to 1. We can now give the resulting ON/OFF decision-making policy for an agent in Algorithm 4.

Suppose that M is the effective Markov matrix for the system when the ON/OFF policy in Algorithm 4 is active. The next theorem shows that M is a linear function of the design matrix K. This linearity property will be used for a convex synthesis of the algorithm design matrix K so that the matrix M satisfies some favorable properties as convergence and safety.

Theorem 4 *Consider a system of single or multiple ON/OFF agents moving in a stochastic environment defined by a finite number of states $\mathcal{S}$ with the transition probabilities given by G as in (5.1) for the ON mode and with no transitions for the OFF mode. Suppose that each agent executes the ON/OFF decision-making Algorithm 4 with the matrix K defined in (5.3). Then the density distribution $x(t)$ evolves based on a Markov chain as in (2.6), where the Markov matrix $M \in \mathcal{P}^{n\times n}$ is given as follows: For $i,j \in \mathbb{N}_n^+$,*

$$M[i,j] = \begin{cases} G[i,j]\,K[i,j] & \text{if} \quad i \neq j, \\ 1 - \sum\limits_{\substack{k=1 \\ k\neq j}}^{n} G[k,j]K[k,j] & \text{if} \quad i = j. \end{cases} \tag{5.4}$$

Algorithm 4: ON/OFF Decision-Making Policy – Single action case

Inputs : K (matrix designed offline), $\mathcal{S}, t_{\max}$

for $t \leftarrow 1$ **to** $t_{\max}$ **do**

  Observe the current state, $s(t) \in \mathcal{S}$ (assume $s(t) = s_j$) ;

  Generate a random number $\eta(t) \sim U(0, 1)$;

  Observe the transition outcome if the agent is in the ON mode, i.e., $y(t+1)$ (assume $y(t+1) = s_i$);

  if $\eta(t) \in [0, K[i, j]]$ **then**

    The agent switches to the mode ON $\sigma(t) = \sigma_{on}$, and $s(t+1) = y(t+1)$;

  else

    The agent switches to the mode OFF $\sigma(t) = \sigma_{off}$, and $s(t+1) = s(t)$;

  end

end

In matrix form, the above relationship is equivalent to

$$M = G \odot K + \mathrm{diag}(\mathbf{1}^T - \mathbf{1}^T(G \odot K)). \tag{5.5}$$

Proof: Transition from the j^{th} state to the i^{th} state ($i \neq j$) can only take place when that transition is observed (line 4 in Algorithm 4) and accepted (line 6 in Algorithm 4):

$$\begin{aligned} M[i,j] &= \mathcal{P}\{s(t+1) = s_i | s(t) = s_j\} \\ &= \mathcal{P}\{\sigma(t) = \sigma_{on}, y(t+1) = s_i | s(t) = s_j\} \end{aligned} \tag{5.6}$$

Using Bayes' Rule, we have:

$$\begin{aligned} M[i,j] &= \mathcal{P}\{y(t{+}1) = s_i | s(t) = s_j\} \\ &\qquad \times \mathcal{P}\{\sigma(t) = \sigma_{on} | s(t) = s_j, y(t+1) = s_i\} \\ &= G[i,j]K[i,j] \end{aligned} \tag{5.7}$$

The system stays at the j^{th} state if either (a) the observed state at line 4 in Algorithm 4 is rejected (line 8 in Algorithm 4) or (b) the j^{th} state is observed at line 4 in Algorithm 4 and is accepted (line 6 in Algorithm 4), then

$$\begin{aligned} M[j,j] &= \mathcal{P}\{s(t{+}1) = s_j | s(t) = s_j\} \\ &= \mathcal{P}\{\sigma(t) = \sigma_{\mathit{off}} | s(t) = s_j\} + \mathcal{P}\{\sigma(t) = \sigma_{on}, y(t+1) = s_j | s(t) = s_j\} \\ &= 1 - \sum_{k=1}^{n} \mathcal{P}\{\sigma(t){=}\sigma_{on}, y(t+1){=}s_k | s(t){=}s_j\} \\ &\qquad + \mathcal{P}\{\sigma(t){=}\sigma_{on}, y(t+1){=}s_j | s(t){=}s_j\} \\ &= 1 - \sum_{\substack{k=1 \\ k \neq j}}^{n} G[k,j]K[k,j] \end{aligned}$$

Note that we can verify that $M \in \mathbb{P}^{n\times n}$ by showing that $M \geq 0$ and $\mathbf{1}^T M = \mathbf{1}^T$.
$M \geq 0$: Letting $S := G \odot K$, since $0 \leq G,\ K \leq \mathbf{1}\mathbf{1}^T$, we have $\mathbf{1}^T S = \mathbf{1}^T (G \odot K) \leq \mathbf{1}^T G = \mathbf{1}^T$. Hence $\mathbf{1}^T - \mathbf{1}^T S \geq \mathbf{0}$, which implies that $M = S + \operatorname{diag}(\mathbf{1}^T - \mathbf{1}^T S) \geq \mathbf{0}$.
$\mathbf{1}^T M = \mathbf{1}^T$: Let $\xi^T := \mathbf{1}^T G \odot K$. Then $\mathbf{1}^T M = \xi^T + \mathbf{1}^T \operatorname{diag}(\mathbf{1}^T - \xi^T) = \xi^T + \mathbf{1}^T (I - \operatorname{diag}(\xi)) = \xi^T + \mathbf{1}^T - \mathbf{1}^T \operatorname{diag}(\xi) = \xi^T + \mathbf{1}^T - \xi^T = \mathbf{1}^T$. ■

Remark: The formulation for Markov matrix M given in (5.4) is quite intuitive. The probability of making transition $i \to j$ is simply the environment induced probability of this transition times the probability of accepting this transition. The diagonal entries of M are set so that the resulting Markov matrix satisfies the column stochasticity property. ■

Using the new formulation of Markov matrix for ON/OFF agents, we can give a set for feasible Markov matrices which satisfy safety, transition and ON/OFF constraints:

$$
\boxed{
\begin{aligned}
& M \in \mathcal{M}_F^S \quad \text{if and only if} \\
& \qquad \mathbf{1}^T M = \mathbf{1}^T, \quad M \geq 0, \quad Mv = v, \quad (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0} \\
& \qquad M = G \odot K + \operatorname{diag}(\mathbf{1}^T - \mathbf{1}^T (G \odot K)), \\
& \qquad S \geq \mathbf{0}, \qquad [L + S + y\mathbf{1}^T] \geq 0, \quad y + q \geq [L + S + y\mathbf{1}^T] p
\end{aligned}
} \tag{5.8}
$$

Formulation of synthesis as an optimization problem:

We can formulate the Markov matrix synthesis as a minimization problem with the desired constraints. Using the cost function given in (3.25) which aims to minimize overall action, i.e., $M \simeq I$:

$$\begin{aligned} \min_{M} \quad & \mathbf{1}^T(\mathbf{1} - \text{diag}(M)) \quad \textit{such that} \\ & M \in \mathcal{M}_F^S \\ & (3.3) \text{ or } (3.4) \text{ or } (3.5). \end{aligned} \tag{5.9}$$

An analytical solution for K without safety constraints:

Note that we cannot determine the transitional probabilities when an agent is ON, we can only accept or reject the motion in this model. In this sense, the proposed approach has connections with the celebrated Metropolis-Hastings algorithm [38, 18, 74]. More precisely, under some assumptions, the design of ON/OFF policy becomes equivalent to designing the *acceptance matrix* in the M-H algorithm. Indeed we prove that, in the absence of the safety constraints and when reversibility of the chain is enforced, the ON/OFF policy must have the form of the general acceptance matrix in M-H algorithm. When we have safety constraints, the resulting ON/OFF policy can be seen as a new M-H algorithm with the safety constraints. It is a new M-H algorithm because the classical algorithm cannot ensure the safety constraints. The following theorem introduces an analytical synthesis method for the matrix K to make connections with the celebrated Metropolis-Hastings (M-H) algorithm that is used extensively in Monte Carlo Markov Chain (MCMC) field. To this end, we impose the *reversibility condition* on the stationary distribution v.

Theorem 5 *Consider the Markov matrix, M, given by (5.5) where $\boldsymbol{i}(G\,\text{diag}(v)) = \boldsymbol{i}(\text{diag}(v)\,G^T)$, with $v \in \mathcal{P}^m$, $v > \mathbf{0}$, and $\text{tr}(G) > 0$. A control policy matrix K such that $\boldsymbol{i}(K) = \boldsymbol{i}(K^T)$ satisfies the ergodicity constraint (2.3) with reversibility, $M\text{diag}(v) = \text{diag}(v)M^T$,* if and only if *$\boldsymbol{i}(G \odot K)$ corresponds to the adjacency matrix of a strongly connected graph and there exists a matrix*

$L = L^T$ that satisfies the following conditions:

$$\begin{aligned} &\mathbf{0} \le L \le \mathbf{1}\mathbf{1}^T, \quad K = L \odot \min(\mathbf{1}\mathbf{1}^T, R) \quad \textit{where} \\ &R := (\mathrm{diag}(v)\, G^T) \oslash (G\, \mathrm{diag}(v)). \end{aligned} \tag{5.10}$$

The following proof utilizes an element-wise division on positive vectors, which presents is not standard. We believe that this notation (and approach) presents an alternative to the standard notation Markov chain literature and it can be insightful in constructing Markov matrices analytically. Here, $\odot$ represents the Hadamard (Schur) product and $\oslash$ denotes element-wise matrix division defined for non-negative matrices by $A = B \oslash C$ means that $A[i,j] = B[i,j]/C[i,j]$ if $C[i,j] > 0$ and $A[i,j] = 0$ if $B[i,j] = C[i,j] = 0$. So $\oslash$ is well-defined if $B[i,j] = 0$ when $C[i,j] = 0$. Further $A \oslash A = \mathtt{i}(A)$, $A = B \oslash C$ implies that $C = B \oslash A$ and $B = A \odot C$.

Proof: • We first establish the equivalence of (5.10) to the reversibility. Observe that $M\mathrm{diag}(v) = \mathrm{diag}(v)M^T$ is equivalent to having

$$(G \odot K)\mathrm{diag}(v) = \mathrm{diag}(v)(G^T \odot K^T). \tag{5.11}$$

This can be established by observing that $D\mathrm{diag}(v) = \mathrm{diag}(v)D$, for $D = \mathrm{diag}(\mathbf{1}^T - \mathbf{1}^T(G \odot K))$, since D is diagonal.

Let $U := K \oslash K^T$, which is well-defined since $\mathtt{i}(K) = \mathtt{i}(K^T)$. The equality (5.11) implies that

$$\begin{aligned} &(G \odot K^T \odot U)\mathrm{diag}(v) - \mathrm{diag}(v)(G^T \odot K^T) = \mathbf{0} \\ &\Rightarrow\ (G\mathrm{diag}(v)) \odot U \odot K^T - (\mathrm{diag}(v)G^T) \odot K^T = \mathbf{0} \\ &\Rightarrow\ \left[(G\mathrm{diag}(v)) \odot U - \mathrm{diag}(v)G^T\right] \odot K^T = \mathbf{0}, \end{aligned} \tag{5.12}$$

which implies that the reversibility can equivalently be expressed by the equation (5.12). Noting that $K \geq \mathbf{0}$ and $\mathtt{i}(K) = \mathtt{i}(K^T)$, the equation (5.12) can be equivalently expressed as

$$\underbrace{\left[(G\mathrm{diag}(v)) \odot U - \mathrm{diag}(v)G^T\right]}_{:=\Phi} \odot \mathtt{i}(K) = \mathbf{0}. \tag{5.13}$$

The above equality is equivalent to having $\Phi[i, j] = 0$ for $K[i, j] > 0$. Dividing (in the matrix sense) both sides by $G\mathrm{diag}(v)$, which is a well-defined operation since it does not lead to a strictly positive number being divided by zero,

$$\big(U - (\mathrm{diag}(v)\, G^T) \oslash (G\, \mathrm{diag}(v))\big) \odot \mathtt{i}(K) = \mathbf{0}.$$

This equation implies the following condition, which is an equivalent characterization of reversibility in this case,

$$(U - R) \odot \mathtt{i}(K) = \mathbf{0} \quad \text{where} \quad U = K \oslash K^T. \tag{5.14}$$

The equation (5.14) can be expanded to have $U \odot \mathtt{i}(K) = R \odot \mathtt{i}(K)$. Since $U = K \oslash K^T$ with $\mathtt{i}(K) = \mathtt{i}(K^T)$, $U \odot \mathtt{i}(K) = U$, which implies that the equation (5.14) is equivalent to

$$K \oslash K^T = \mathtt{i}(K) \odot R. \tag{5.15}$$

Now, under the assumptions of the theorem, there is a new equivalent condition to reversibility given by the equation (5.15). It implies that, since $\mathtt{i}(K)$ is symmetric and $\mathbf{0} \leq K \leq \mathbf{1}\mathbf{1}^T$,

$$K = R \odot \mathtt{i}(K) \odot K^T = R \odot \mathtt{i}(K^T) \odot K^T = R \odot K^T$$
$$\Rightarrow \; K \leq \min(\mathbf{1}\mathbf{1}^T, R),$$

which then implies that there exists some $\mathbf{0} \leq L \leq \mathbf{1}\mathbf{1}^T$ such that $\mathtt{i}(L) = \mathtt{i}(K)$

$$K = L \odot \min(\mathbf{1}\mathbf{1}^T, R) \;\Rightarrow\; K^T = L^T \odot \min(\mathbf{1}\mathbf{1}^T, R^T).$$

Then we have

$$\begin{aligned} K \oslash K^T &= \big(L \odot \min(\mathbf{1}\mathbf{1}^T, R)\big) \oslash \big(L^T \odot \min(\mathbf{1}\mathbf{1}^T, R^T)\big) \\ &= \big(L \odot \min(\mathbf{1}\mathbf{1}^T, R)\big) \oslash \big(L^T \odot \min(\mathbf{1}\mathbf{1}^T, \mathtt{i}(R) \oslash R)\big), \end{aligned}$$

where $R^T = \mathtt{i}(R) \oslash R$ can be derived as follows:

$$\begin{aligned} & R^T = (G\mathrm{diag}(v)) \oslash (\mathrm{diag}(v)G^T) \Rightarrow \; R^T \odot R \\ &= (G\mathrm{diag}(v)) \oslash (\mathrm{diag}(v)G^T) \odot (\mathrm{diag}(v)G^T) \oslash (G\mathrm{diag}(v)) \\ &= (G\mathrm{diag}(v)) \oslash (G\mathrm{diag}(v)) \odot (\mathrm{diag}(v)G^T) \oslash (\mathrm{diag}(v)G^T) \\ &= \mathtt{i}(\mathrm{diag}(v)G^T) \odot \mathtt{i}(G\mathrm{diag}(v)) = \mathtt{i}(R), \end{aligned}$$

which then implies $R^T = \mathtt{i}(R) \oslash R$. Now, since $\mathtt{i}(L) = \mathtt{i}(K)$ (which is symmetric)

$$K \oslash K^T = (L \oslash L^T) \odot \underbrace{\big(\min(\mathbf{1}\mathbf{1}^T, R) \oslash \min(\mathbf{1}\mathbf{1}^T, \mathtt{i}(R) \oslash R)\big)}_{:=\psi}.$$

Clearly ψ is well-defined since $\mathtt{i}(\min(\mathbf{1}\mathbf{1}^T, R)) = \mathtt{i}(\min(\mathbf{1}\mathbf{1}^T, \mathtt{i}(R) \oslash R))$, further

$$\psi[i,j] = \begin{cases} 1/(1/R[i,j]) & \text{if } \; R[i,j] \geq 1 \\ R[i,j]/1 & \text{if } \; R[i,j] \in (0,1) \\ 0 & \text{if } \; R[i,j] = 0 \end{cases} \quad i,j = 1,\ldots,m,$$

which implies that $\psi = R$. This then implies that

$$K \oslash K^T = (L \oslash L^T) \odot R.$$

Using the equation (5.15), the above implies that

$$(L \oslash L^T) \odot R = \mathtt{i}(K) \odot R \;\Rightarrow\; L \oslash L^T = \mathtt{i}(K) \odot R \oslash R = \mathtt{i}(K) \odot \mathtt{i}(R)$$

where $\mathtt{i}(K) = \mathtt{i}(L)$ and $\mathtt{i}(R)$ are symmetric. Since $K \leq \min(\mathbf{1}\mathbf{1}^T, R)$, $R[i,j] = 0$ implies that $K[i,j] = 0$ and hence $\mathtt{i}(K) \odot \mathtt{i}(R) = \mathtt{i}(K) = \mathtt{i}(L)$, which is symmetric. Consequently,

$$L \oslash L^T = \mathtt{i}(L) \;\Rightarrow\; L \oslash \mathtt{i}(L) = L^T \;\Rightarrow\; L = L^T.$$

This concludes the necessity of having symmetric L satisfying (5.10).

Next we show the sufficiency. Consider a symmetric L as in (5.10). To show sufficiency, it is enough to show that (5.15) will be satisfied. Then

$$K \oslash K^T = \left(L \odot \min(\mathbf{1}\mathbf{1}^T, R)\right) \oslash \left(L^T \odot \min(\mathbf{1}\mathbf{1}^T, R^T)\right).$$

Then, as done above, we can show that $K \oslash K^T = (L \oslash L^T) \odot R$ and since $L = L^T$, $L \oslash L^T = \mathtt{i}(L)$, which concludes the sufficiency proof. This then concludes the proof.

- Next we present the equivalence of the ergodicity with the strong connectivity of the graph of $\mathtt{i}(G \odot K)$. First observe that the ergodicity constraint is equivalent to the primitivity of the matrix M when $v > \mathbf{0}$ (see [1, 3]).

Now we will show that, when $\mathrm{tr}(G) > 0$, the primitivity of M is equivalent to the strong connectivity of $G \odot K$. Since $\mathrm{tr}(G) > 0$, there exists $G[j,j] > 0$ for some index j. Note that $\sum_{i=1}^{m} M[i,j] = M[j,j] + \sum_{i \neq j} M[i,j]$ where $\sum_{i \neq j} M[i,j] \leq \sum_{i \neq j} G[i,j] < 1$ since $G[j,j] > 1$ and $G \in \mathcal{P}^m$. This implies that $\sum_{i \neq j} M[i,j] < 1$, and hence $M[j,j] > 0$, which then implies that $\mathrm{tr}(M) > 0$. Since $\mathrm{tr}(M) > 0$, and since the strong connectivity of the graph of M is same as the graph of $G \odot K$, the strong connectivity of $G \odot K$ is equivalent to the primitivity of M [61]. This concludes the proof. ■

5.1.1 *Numerical Example*

This section presents preliminary results for simulations which demonstrates the PDC algorithm for ON/OFF agents. In this problem, $N = 3000$ agents are assumed to be distributed on a region which is partitioned to 8 equally sized rectangular bins (see Fig 5.2) and position based definitions for $x(t)$ and M given in section 3.6.1 are used. When all agents are in the ON mode, the swarm density evolves according to the following G matrix:

$$G = \begin{bmatrix} \frac{24}{125} & \frac{57}{2000} & \frac{187}{5000} & 0 & 0 & 0 & 0 & 0 \\ \frac{943}{2500} & \frac{137}{5000} & \frac{973}{2500} & \frac{2763}{10000} & \frac{2763}{10000} & 0 & 0 & 0 \\ \frac{1077}{2500} & \frac{1957}{5000} & \frac{2867}{5000} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2763}{10000} & 0 & \frac{387}{5000} & \frac{387}{5000} & 0 & \frac{697}{5000} & 0 \\ 0 & \frac{691}{2500} & 0 & \frac{387}{5000} & \frac{387}{5000} & 0 & \frac{697}{5000} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{308}{625} & 0 & \frac{317}{625} \\ 0 & 0 & 0 & \frac{5689}{10000} & \frac{5689}{10000} & \frac{93}{1000} & \frac{57}{625} & \frac{129}{2000} \\ 0 & 0 & 0 & 0 & 0 & \frac{2071}{5000} & \frac{63}{100} & \frac{4283}{10000} \end{bmatrix}$$

Uncontrolled swarm density that evolves according to given G has the following final steady-state distribution.

$$v_n = [0.05 \;\; 0.1 \;\; 0.05 \;\; 0.1 \;\; 0.1 \;\; 0.275 \;\; 0.05 \;\; 0.275]^T.$$

The aim is to design an ON/OFF decision policy, K, such that the resulting Markov chain, M, will lead the swarm to converge to a particular final distribution while satisfying the desired safety upper bound constraints. Simulation parameters are set as follows:

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

$$v_d = [0.005 \;\; 0.02 \;\; 0.005 \;\; 0.04 \;\; 0.05 \;\; 0.34 \;\; 0.2 \;\; 0.34]^T,$$

$$d = [1 \;\; 0.15 \;\; 1 \;\; 0.12 \;\; 0.12 \;\; 1 \;\; 0.4 \;\; 1]^T.$$

Since the bins at the corners behave like accumulation points in the initial and final conditions, the safety upper bound constraints are not imposed for these bins, i.e. the corresponding entries of the d vector are set as 1. With the given parameters, the optimization problem given in (5.8) generates the following K matrix, with YALMIP and SDPT3 [77, 102]:

$$\begin{bmatrix} 1.0000 & 0.4037 & 0.9283 & 0 & 0 & 0 & 0 & 0 \\ 0.0778 & 1.0000 & 0.0776 & 0.2276 & 0.1544 & 0 & 0 & 0 \\ 0.1193 & 0.0086 & 1.0000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.6699 & 0 & 1.0000 & 0.6298 & 0 & 0.7507 & 0 \\ 0 & 0.1711 & 0 & 0.7467 & 1.0000 & 0 & 0.9575 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000 & 0 & 0.8349 \\ 0 & 0 & 0 & 0.9773 & 0.8924 & 0.9400 & 1.0000 & 0.9324 \\ 0 & 0 & 0 & 0 & 0 & 0.8112 & 0.3982 & 1.0000 \end{bmatrix}$$

Simulations are done for the cases i) No ON/OFF policy, all the agents are in the ON mode; ii) with ON/OFF decision policy; which helps to observe effect of ON/OFF decision policy. Figure 5.2 shows the time histories of the overall density distribution for these two cases. The density goes above the desired upper bound for the bins $2, 4$ and 5 when all the agents are ON. By using binary control actions, we are able to modify the final distribution and ensure that the density does not go beyond the prescribed upper limit at the expense of convergence rate.

The snapshots of the overall distribution taken at selected time instances for the case with ON/OFF decision policy are shown in Figure 5.2 which also shows the bin numbers.

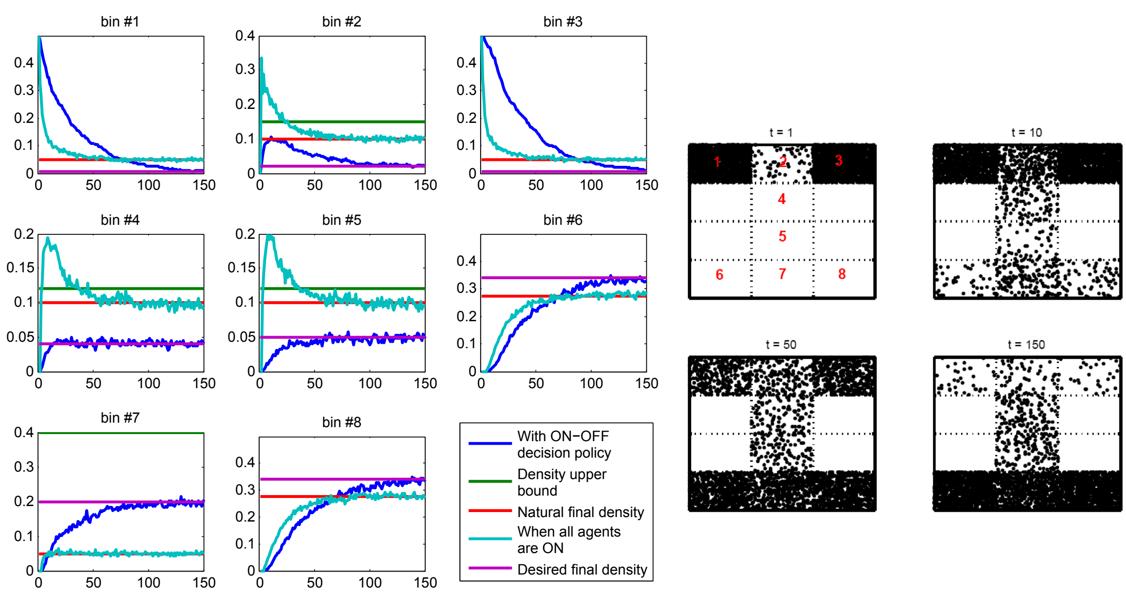

Figure 5.2: Left: Time history of the density of each bin for the simulations with ON/OFF agents. Right: Evolution of the density distribution with ON-OFF decision policy also showing bin numbers

5.2 *Generalization for Multiple ON Actions*

In this section, we present the generalization of the ON/OFF decision control policy problem for the case where the ON mode encapsulates multiple actions and the OFF mode features a single action that does not necessarily correspond to "no motion". As explained earlier: in the ON mode, the "next step" outcomes of the actions are observable, while a Markov chain, G_{off}, is propagated when OFF mode is chosen. Now, we can have multiple actions in the ON mode whose transitions can be observed, i.e.,

$$\sigma(t) = \sigma_{on} \implies a(t) \in \mathcal{A}_{on} = \{a_1, \ldots, a_m\}, \tag{5.16}$$

where $a(t)$ is the action taken in the ON mode. In the simple ON/OFF case of the previous section, we had $\mathcal{A}_{on} = \{a_1\}$, hence we did not need to explicitly define actions. Since we have multiple actions in the ON mode now, we have to explicitly identify them.

Using the same definitions for $x(t)$ and M given by (2.4) and (2.5), we will formulate the Markov matrix M as a function of the stochastic environment and the actions determined by a predetermined ON/OFF policy. For this general case, we expand the definitions of the probabilistic events for $t \in \mathbb{N}$:

$$\begin{aligned} y(t+1) = s_l &: \text{ Observing a transition to state } s_l, \\ v(t) = a_k &: \text{ Observing the outcome of taking action} \\ &\quad\; a_k \text{ from } \mathcal{A}_{on}, \\ a(t) = a_k &: \text{ Accepting to execute action } a_k. \end{aligned}$$

Comparing to the previous model with single action, the addition here is the event "$v(t) = a_k$", which is used to define the probability of choosing an action whose outcome will be observed. The stochastic environment is defined with the transition matrices $G_k, G_{\mathit{off}} \in \mathbb{P}^{n\times n},\ k \in \mathbb{N}_m^+$ where $G_k[i,j]$ gives the probability of observing a transition from j^{th} state to i^{th} state, given that the k^{th} action is selected to be observed, i.e.,

$$G_k[i,j] = \mathcal{P}\{y(t+1) = s_i | s(t) = s_j,\, v(t) = a_k\}, \tag{5.17}$$

where $i, j \in \mathbb{N}_n^+,\ k \in \mathbb{N}_m^+$ and similarly $G_{\mathit{off}}[i,j]$ defines the corresponding transition probabilities for the action in the OFF mode (e.g., $G_{\mathit{off}} = I$ when being OFF means no motion as in the previous section). The model for ON/OFF decision-making has the following assumptions (see Fig. 5.3):

- Each agent measures its own state at time instant t.
- Agent chooses a single action for the ON mode, say a_k, whose outcome will be observed, i.e., $v(t) = a_k$.
- Agent accepts or rejects to take the observed action.
- If action is accepted then it is taken, i.e., $a(t) = a_k$, and transitions occurs according to G_k.
- If action is rejected, the agent chooses the OFF mode, $\sigma(t) = \sigma_{\mathit{off}}$ and the transition occurs according to G_{off}.

Remark: In the above setting, G_{off} can have two interpretations: (i) the environment transition matrix when there is only one action for which the outcome is not observable; (ii) the effective transition matrix when there are

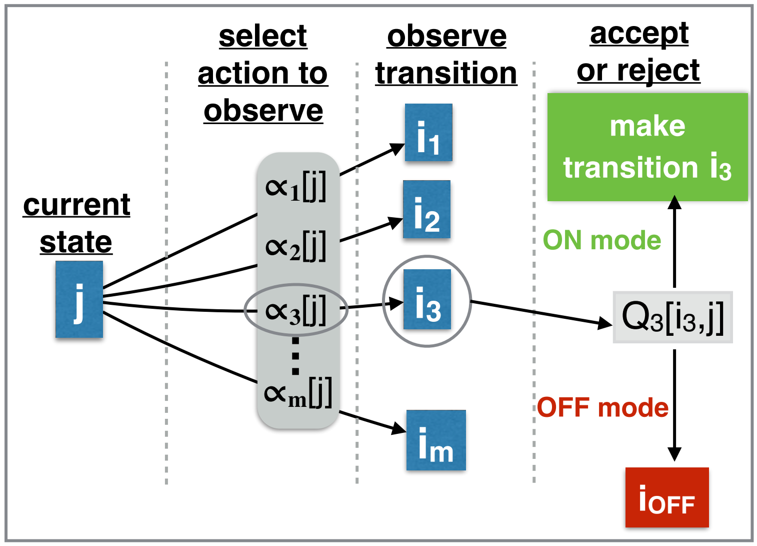

Figure 5.3: Implementation of the decision policy.

multiple actions with a prescribed decision policy. For the latter case, G_{off} is a Markov matrix defining the underlying Markov chain of an MDP with an existing policy running over the actions, which is discussed in more detail in the next subsection. ■

In the generalized model, we consider two sets of decision variables (to be designed offline):

$$\alpha_k[j] = \mathcal{P}\{v(t) = a_k | s(t) = s_j\} \quad (5.18)$$

$$Q_k[i,j] = \mathcal{P}\{a(t) = a_k | s(t) = s_j, v(t) = a_k, y(t+1) = s_i\} \quad (5.19)$$

with $k \in \mathbb{N}_m^+$, $i, j \in \mathbb{N}_n^+$. Namely, $\alpha_k[j]$ is the probability of choosing action $a_k \in \mathcal{A}_{on}$ at state s_j and $Q_k[i,j]$ is the probability of accepting an achievable transition $j \to i$ observed as an outcome of taking an action a_k.

Clearly Q_k matrices must be non-negative, $Q_k \in [0,1]^{n\times n},\ k \in \mathbb{N}_m^+$. Also, non-negative action variables α_k should satisfy the inequality $\sum_{k=1}^{m} \alpha_k[j] \leq 1,\ j \in \mathbb{N}_n^+$. It turns out in our model (which will become more clear later) that we can combine these two variables by considering the change of variables

$$P_k := Q_k \mathrm{diag}(\alpha_k) = Q_k \odot (\mathbf{1}\alpha_k^T), \quad k \in \mathbb{N}_m^+. \tag{5.20}$$

The following property holds for P_k matrices,

$$P_k \leq \mathbf{1}\,\alpha_k^T, \quad k \in \mathbb{N}_m^+. \tag{5.21}$$

This inequality can simply be proven by contradiction. Suppose there exist i and j such that $P_k[i,j] > \alpha_k[j]$, then $Q_k[i,j] = P_k[i,j]/\alpha_k[j] > 1$ which is a contradiction because $Q_k[i,j] \in [0,1]$.

We can now give by Algorithm 5 the ON/OFF decision-making policy for the general case. For this algorithm, we define variable $\phi_j \in \mathbb{R}^{m+2}$ for each state j, where

$$\phi_j[1] = 0,\ \phi_j[r] = \sum_{l=1}^{r} \alpha_l[j],\ \phi_j[m+2] = 1$$

where $r = 2, \ldots, m+1$.

The following theorem presents the key result in converting the ON/OFF decision policy design problem into a Markov chain synthesis problem.

Theorem 6 *Consider a system of single or multiple mode switching ON/OFF agents moving in a stochastic environment defined by a finite number of states $\mathcal{S}$ with the transition probabilities given by G_k as in (5.17) for the k^{th} action $a_k \in \mathcal{A}_{on}$ in the ON mode and G_{off} for the OFF mode. Suppose that each*

Algorithm 5: ON/OFF Decision-Making Policy – General case

Inputs : $\{\alpha_k, Q_k : k \in \mathbb{N}_m^+\}$ (designed offline), $\mathcal{S}, t_{\max}$

for $t \leftarrow 1$ **to** t_{max} **do**
  Determine current state $s(t) \in \mathcal{S}$ (assume $s(t) = s_j$);
  Generate a random numbers $\mu(t) \sim U(0,1)$ and $\eta(t) \sim U(0,1)$;
  if $\mu(t) \in [\phi_j[k],\ \phi_j[k+1])$ **then**
    $v(t) = a_k$;
    Observe the next achievable transition for a_k: $y(t+1)$ (suppose$y(t+1) = s_i$);
    if $\eta(t) \in [0, Q_k[i,j]]$ **then**
      The agent switches to the mode ON, $a(t) = a_k$ and $s(t+1) = s_i$;
    end
  else
    The agent switches to the mode OFF $a(t) = a_{\mathit{off}}$, and $s(t+1)$ transitions according to G_{off};
  end
end

agent executes the ON/OFF decision-making Algorithm 5 with the matrices Q_k as in (5.19) and the vectors α_k as in (5.18). Then the state p.d.f. $x(t)$, defined in (2.4), evolves based on the Markov chain (2.6) with the Markov matrix $M \in \mathcal{P}^{n\times n}$ given by

$$\begin{aligned} M =& \sum_{l=1}^{n}\sum_{k=1}^{m} e_l \mathbf{1}^T \left({}^l R_k \odot P_k \odot G_k\right) \\ &+ G_{\mathit{off}} \odot \left(\mathbf{1}\left(\mathbf{1}^T - \mathbf{1}^T \sum_{l=1}^{n}\sum_{k=1}^{m} e_l \mathbf{1}^T \left({}^l R_k \odot P_k \odot G_k\right)\right)\right) \end{aligned} \tag{5.22}$$

where P_k, $k \in \mathbb{N}_m^+$ are given by (5.20) and they satisfy

$$\sum_{k=1}^{m} \max_{i\in\mathbb{N}_n^+} P_k[i,j] \leq 1, \quad j \in \mathbb{N}_n^+. \tag{5.23}$$

Proof: The probability of making transition from j^{th} state to i^{th} state can be written as the following sum, since being ON and being OFF are mutually exclusive events:

$$\begin{aligned} M[i,j] =& \mathcal{P}\{s(t+1) = s_i | s(t) = s_j\} \\ =& \underbrace{\mathcal{P}\{\sigma(t) = \sigma_{on},\, s(t+1) = s_i | s(t) = s_j\}}_{:=T_1[i,j]} \\ &+ \underbrace{\mathcal{P}\{\sigma(t) = \sigma_{\mathit{off}},\, s(t+1) = s_i | s(t) = s_j\}}_{:=T_2[i,j]}. \end{aligned} \tag{5.24}$$

In Algorithm 5, since executing the ON mode implies that an action from $\mathcal{A}_{on}$ must be taken, the first term can be written as the summation over all actions for the ON mode:

$$
\begin{aligned}
T_1[i,j] &= \sum_{k=1}^{m} \mathcal{P}\{s(t+1)=s_i, a(t)=a_k | s(t)=s_j\} \\
&= \sum_{k=1}^{m} \mathcal{P}\{s(t+1)=s_i, a(t)=a_k, v(t)=a_k | s(t)=s_j\}
\end{aligned}
$$

The second equation above follows from the fact that $v(t)$ should always precedes $a(t)$, i.e., $\mathcal{P}(v(t) = a_k | a(t) = a_k) = 1$. By applying Bayes' rule [26] to the term inside the sum, we obtain:

$$
T_1[i,j] = \sum_{k=1}^{m} \mathcal{P}\{s(t+1)=s_i, a(t)=a_k | v(t)=a_k, s(t)=s_j\} \underbrace{\mathcal{P}\{v(t)=a_k | s(t)=s_j\}}_{\alpha_k[j]}.
$$

Since observing transitions to distinct states are mutually exclusive events,

$$
T_1[i,j] = \sum_{k=1}^{m} \left[\sum_{l=1}^{n} \mathcal{P}\{s(t+1)=s_i, a(t)=a_k, y(t+1)=s_l | v(t)=a_k, s(t)=s_j\} \right] \alpha_k[j]
$$

Applying Bayes' rule, we obtain:

$$= \sum_{k=1}^{m}\left[\sum_{l=1}^{n}\mathcal{P}\{s(t+1)=s_i, a(t)=a_k|y(t+1)=s_l, v(t)=a_k, s(t)=s_j\}\right.$$
$$\left.\times\underbrace{\mathcal{P}\{y(t+1)=s_l|v(t)=a_k, s(t)=s_j\}}_{G_k[l,j]}\right]\alpha_k[j]$$
$$= \sum_{k=1}^{m}\left[\sum_{l=1}^{n}\mathcal{P}\{s(t+1)=s_i|y(t+1)=s_l, a(t)=a_k, v(t)=a_k, s(t)=s_j\}\right.$$
$$\times\underbrace{\mathcal{P}\{a(t)=a_k|y(t+1)=s_l, v(t)=a_k, s(t)=s_j\}}_{Q_k[l,j]}$$
$$\left.\times G_k[l,j]\right]\alpha_k[j]. \tag{5.25}$$

Let

$$\mathcal{P}\{s(t+1)=s_i|y(t+1)=s_l, a(t)=a_k, v(t)=a_k, s(t)=s_j\} = {}^iR_k[l,j] \tag{5.26}$$

Then,

$$T_1[i,j] = \sum_{k=1}^{m}\left[\sum_{l=1}^{n}{}^iR_k[l,j]Q_k[l,j]G[l,j]\right]\alpha_k[j]$$
$$= \sum_{k=1}^{m}\left[\mathbf{1}^T\left({}^iR_k\odot Q_k\odot G_k\right)e_j\right]\alpha_k[j]$$
$$e_i^T T_1 = \sum_{k=1}^{m}\left[\mathbf{1}^T\left({}^iR_k\odot Q_k\odot G_k\right)\right]\odot\alpha_k$$
$$= \sum_{k=1}^{m}\mathbf{1}^T\left({}^iR_k\odot(Q_k\odot\mathbf{1}\alpha_k^T)\odot G_k\right) = \sum_{k=1}^{m}\mathbf{1}^T\left({}^iR_k\odot P_k\odot G_k\right)$$

Then T_1 can be expressed in matrix form as follows:

$$T_1 = \sum_{i=1}^{n}\sum_{k=1}^{m}e_i\mathbf{1}^T\left({}^iR_k\odot P_k\odot G_k\right) \tag{5.27}$$

Now, consider the second term in (5.24):

$$\begin{aligned} T_2[i,j] &= \mathcal{P}\{\sigma(t)=\sigma_{\mathit{off}}, s(t+1)=s_i|s(t)=s_j\} \\ &= \underbrace{\mathcal{P}\{s(t+1)=i|s(t)=s_j, \sigma(t)=\sigma_{\mathit{off}}\}}_{G_{\mathit{off}}[i,j]} \\ &\qquad \times \underbrace{\mathcal{P}\{\sigma(t)=\sigma_{\mathit{off}}|(s(t)=s_j\}}_{1-\mathcal{P}\{\sigma(t)=\sigma_{on}|s(t)=s_j\}}. \end{aligned}$$

Here, given the current state, the probability of being ON is sum of the probabilities of taking each action for the ON mode, i.e.,

$$\begin{aligned} &\mathcal{P}\{\sigma(t)=\sigma_{on}|s(t)=s_j\} \\ &\qquad = \sum_{l=1}^{n} \mathcal{P}\{\sigma(t)=\sigma_{on}, s(t+1)=s_l|s(t)=s_j\} \\ &\qquad = \sum_{l=1}^{n} T_1[l,j] \end{aligned}$$

Hence, T_2 can be written in matrix form as follows:

$$\begin{aligned} T_2 &= G_{\mathit{off}} \odot \left(\mathbf{1}\left(\mathbf{1}^T - \mathbf{1}^T T_1\right)\right) \\ &= G_{\mathit{off}} \odot \left(\mathbf{1}\left(\mathbf{1}^T - \mathbf{1}^T \sum_{i=1}^{n}\sum_{k=1}^{m} e_i \mathbf{1}^T \left({}^iR_k \odot P_k \odot G_k\right)\right)\right) \end{aligned} \tag{5.28}$$

Now, combining the expressions we get for T_1 and T_2 as $M = T_1 + T_2$ yields (5.22).

Finally, we will show that $M \in \mathbb{P}^{n\times n}$. For nonnegativity, we will consider M in two terms. As both terms corresponds to probabilistic quantities, they both must be nonnegative. The nonnegativity of the first term is clear since $G_k \geq 0$, $P_k \geq 0$ and ${}^iR_k > 0$, $k \in \mathbb{N}_m^+$, $i \in \mathbb{N}_n^+$. For the nonnegativity of the second term, we need to show

$$\mathbf{1}^T \sum_{i=1}^{n} \sum_{k=1}^{m} e_i \mathbf{1}^T \left({}^i R_k \odot P_k \odot G_k \right) \leq \mathbf{1}^T, \tag{5.29}$$

which is equivalent to:

$$\mathbf{1}^T \sum_{i=1}^{n} \sum_{k=1}^{m} \left({}^i R_k \odot P_k \odot G_k \right) \leq \mathbf{1}^T. \tag{5.30}$$

Here, $\mathbf{1}^T \sum_{i=1}^{n} \sum_{k=1}^{m} ({}^i R_k \odot P_k \odot G_k)$ can be written in index notation as follows: For $j \in \mathbb{N}_m^+$,

$$\begin{aligned} &\sum_{i=1}^{n} \sum_{k=1}^{m} \mathbf{1}^T \left({}^i R_k \odot P_k \odot G_k \right) e_j \\ &\qquad = \sum_{l=1}^{n} \sum_{i=1}^{n} \sum_{k=1}^{m} {}^i R_k[l,j] P_k[l,j] G_k[l,j] \end{aligned}$$

Here, $\sum_{k=1}^{m} P_k[l,j] \leq 1$ for any l and j since

$$\sum_{k=1}^{m} P_k \leq \sum_{k=1}^{m} \mathbf{1} \alpha_k^T \leq \mathbf{1}\mathbf{1}^T,$$

$\sum_{l=1}^{n} G_k[l,j] = 1$ for any j and k since G_k is column stochastic and $\sum_{i=1}^{n} {}^i R_k[l,j] = 1$.

Hence,

$$\begin{aligned}
\sum_{l=1}^{n}\sum_{i=1}^{n}\sum_{k=1}^{m} {}^{i}R_k[l,j]P_k[l,j]G_k[l,j] \\
&= \sum_{l=1}^{n}\sum_{k=1}^{m}\sum_{i=1}^{n} {}^{i}R_k[l,j]P_k[l,j]G_k[l,j] \\
&= \sum_{l=1}^{n}\sum_{k=1}^{m} P_k[l,j]G_k[l,j] \\
&= \sum_{k=1}^{m} \underset{l\in\mathbb{N}_n^+}{\mathrm{conv}} P_k[l,j] \le \sum_{k=1}^{m} \max_{l\in\mathbb{N}_n^+} P_k[l,j] \;\le 1.
\end{aligned}$$

The last inequality follows from (5.23), which then implies that

$$\sum_{i=1}^{n}\sum_{k=1}^{m} \mathbf{1}^T \left({}^{i}R_k \odot P_k \odot G_k\right) e_j \le 1,\ j \in \mathbb{N}_n^+,$$

and hence (5.29) holds. This concludes that $M \ge 0$.

Next let $H := \sum_{i=1}^{n}\sum_{k=1}^{m}({}^{i}R_k \odot P_k \odot G_k)$, then

$$\begin{aligned}
\mathbf{1}^T M &= \mathbf{1}^T H + \mathbf{1}^T G_{\mathit{off}} \odot (\mathbf{1}(\mathbf{1}^T - \mathbf{1}^T H)) \\
&= \mathbf{1}^T H + \mathbf{1}^T \left(G_{\mathit{off}} - G_{\mathit{off}} \odot (\mathbf{1}\mathbf{1}^T H)\right) \\
&= \mathbf{1}^T H + \mathbf{1}^T - \mathbf{1}^T \left(G_{\mathit{off}} \odot (\mathbf{1}\mathbf{1}^T H)\right).
\end{aligned}$$

Let $\phi := H^T\mathbf{1}$, then

$$\begin{aligned}
\mathbf{1}^T M &= \phi^T + \mathbf{1}^T - \mathbf{1}^T \left(G_{\mathit{off}} \odot (\mathbf{1}\phi^T)\right) \\
&= \phi^T + \mathbf{1}^T - (\mathbf{1}^T G_{\mathit{off}}) \odot \phi^T \\
&= \phi^T + \mathbf{1}^T - \mathbf{1}^T \odot \phi^T = \phi^T + \mathbf{1}^T - \phi^T = \mathbf{1}^T,
\end{aligned}$$

hence $M \in \mathbb{P}^{n\times n}$. ∎

The model given in (5.22) captures a specific case when the transition probability after accepting an observed action/transition pair is 1, i.e.,

$$\begin{aligned} {}^{i}R_k[l,j] = \mathcal{P}\{&s(t{+}1){=}s_i|y(t{+}1){=}s_l, a(t){=}a_k, \\ &v(t){=}a_k, s(t){=}s_j\} = \delta_{ij} \end{aligned} \tag{5.31}$$

which is equivalent to ${}^{i}R_k = e_i\mathbf{1}^T$. The result is given in Corollary 4

Corollary 4 *When ${}^{i}R_k = e_i\mathbf{1}^T$, the model given in (5.22) is equivalent to:*

$$M = \sum_{k=1}^{m} G_k \odot P_k \; + \; G_{\mathit{off}} \odot \left(\mathbf{1}\left(\mathbf{1}^T - \mathbf{1}^T \sum_{k=1}^{m} G_k \odot P_k\right)\right). \tag{5.32}$$

Proof:

$$\begin{aligned} &\sum_{i=1}^{n}\sum_{k=1}^{m} e_i\mathbf{1}^T\left(e_i\mathbf{1}^T \odot P_k \odot G_k\right) \\ =&\sum_{i=1}^{n}\sum_{k=1}^{m} e_i e_i^T\left(P_k \odot G_k\right) = \sum_{k=1}^{m} G_k \odot P_k \end{aligned}$$

∎

Theorem 6 shows that M is a linear function of $\{P_k : k = 1, \ldots, m\}$. This linearity property will be used for a convex synthesis of P_k so that the matrix M satisfies some favorable properties as convergence and safety. The algorithm design parameters α_k and Q_k can then be extracted from P_k as we will show next. Thus the design of $\{P_k : k = 1, \ldots, m\}$ is an intermediary step to set the parameters of Algorithm 5.

Extraction of Decision Variables: Once P_k matrices are computed (which will explained in later sections), the selection of α_k and Q_k can be done in multiple ways, that is, the same P_k matrices can be parameterized in multiple ways via α_k and Q_k. The choice must preserve the following conditions on α_k and Q_k (since they contain probabilities of events as entries)

$$0 \leq Q_k \leq \mathbf{1}\mathbf{1}^T,\ 0 \leq \alpha_k \leq \mathbf{1},\ k \in \mathbb{N}_m^+, \quad \sum_{k=1}^{m} \alpha_k \leq \mathbf{1}. \tag{5.33}$$

Our default parameterization is:

$$\alpha_k[j] = \max_{i \in \mathbb{N}_n^+} P_k[i,j], \quad k \in \mathbb{N}_m^+, \quad j \in \mathbb{N}_n^+. \tag{5.34}$$

Hence the last inequality in (5.33) is satisfied (due to (5.23)), and we can choose Q_k as

$$Q_k = P_k \operatorname{diag}(\alpha_k)^{-1}, \quad k \in \mathbb{N}_m^+. \tag{5.35}$$

Here, Q_k is obtained by dividing each entry in a column of P_k by the maximum element in that column, hence $0 \leq Q_k \leq \mathbf{1}\mathbf{1}^T$. For the same reason, note that any choice of α_k greater than or equal to the choices given in (5.34) would have resulted in feasible Q_k. Also observe that this particular choice α_k allows "no action observed" cases, since it leads to $\sum_{k=1}^m \alpha_k \leq \mathbf{1}$ (the sum does not have to be one). We can normalize α_k's such that an action is always observed, without changing the resulting M, as follows: Form a matrix Γ with α_k's computed via (5.34) as its columns. Then compute a new set of α_k's by using the following expression and Q_k's by using (5.35),

$$\Lambda := [\alpha_1 \ldots \alpha_m] = \operatorname{diag}(\Gamma\, \mathbf{1})^{-1}\, \Gamma. \tag{5.36}$$

Note that $\sum_{k=1}^{m} \alpha_k = \Lambda \mathbf{1} = \text{diag}(\Gamma \mathbf{1})^{-1} \Gamma \mathbf{1} = \mathbf{1}$. Since the second choice of α_k always produces values that are at least as large as the first choice in (5.34), we ensure that $0 \leq Q_k \leq \mathbf{1}\mathbf{1}^T$.

5.2.1 Convex Synthesis of Safe Markov Chain for ON/OFF Agents

So far, we have obtained the linear equivalent conditions on Markov matrix for the transition, safety and ON/OFF constraints. Hence, for the convex optimization problem, we define a set for feasible Markov matrices which satisfy safety, transition and ON/OFF constraints:

$$
\begin{aligned}
& M \in \mathcal{M}_F \quad \textit{if and only if:} \\
& M \geq \mathbf{0},\ S \geq \mathbf{0},\ P_k \geq \mathbf{0} \text{ for } k = 1 \ldots m, \\
& \mathbf{1}^T M = \mathbf{1}^T, \\
& (\mathbf{1}\mathbf{1}^T - A_a^T) \odot M = \mathbf{0}, \\
& [L + S + y\mathbf{1}^T] \geq 0, \\
& y + q \geq [L + S + y\mathbf{1}^T] p, \\
& M = \sum_{k=1}^{m} G_k \odot P_k + G_{\mathit{off}} \odot \left(\mathbf{1} \left(\mathbf{1}^T - \mathbf{1}^T \sum_{k=1}^{m} G_k \odot P_k \right) \right), \\
& \sum_{k=1}^{m} \max_{i \in \mathbb{N}_n^+} P_k[i,j] \leq 1, \quad j \in \mathbb{N}_n^+
\end{aligned}
\tag{5.37}
$$

Note that the last inequality above in (5.37) can be written by using linear inequalities, hence it is a convex constraint. To see that define

$$
Z_j := [P_1(:,j)\ P_2(:,j)\ \ldots\ P_m(:,j)],\ j = 1, ..., n,
$$

where $P_k(:,j)$ is the j'th column of P_k. Then we can replace the last inequality by the following inequalities

$$Z_j \leq \mathbf{1}\beta_j^T, \quad \mathbf{1}^T\beta_j \leq 1, \quad j = 1, ..., n,$$

where β_j's are $m \times 1$ slack variables.

Formulation of synthesis as an optimization problem:

Similar to the previous case We can formulate the Markov matrix synthesis as a minimization problem as follows:

$$\begin{aligned} \min_{M} \quad & \mathbf{1}^T(\mathbf{1} - \text{diag}(M)) \quad \textit{such that} \\ & M \in \mathcal{M}_F \\ & (3.3) \text{ or } (3.4) \text{ or } (3.5). \end{aligned} \tag{5.38}$$

5.2.2 Numerical Example

This section presents an illustrative numerical example for the density control problem for autonomous agents with ON/OFF control modes. For this setting, safety upper bound constraint is used to limit the expected number of agents in each bin. Two sets of simulations are performed by using the ON/OFF policy synthesized by solving the LMI problem in (5.38) with (3.3), both with the density upper bound constraints and $m = 5$ actions for the ON mode: (i) Total $N_{s1} = 3000$ simulations with same safe initial condition, i.e., same $x(0)$ with different realizations; (ii) Total $N_{s2} = 3000$ simulations with randomly generated 3000 safe initial conditions. For all cases, OFF case corresponds to "no action", i.e., $G_{\mathit{off}} = I$. For the actions in the ON mode, column stochastic G_k matrices are selected such that they have different steady-state final distributions and do not satisfy safety and transition constraints.

Other parameters for the simulations are set as follows:

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

$$N_a = 3000$$

$$x(0) = [0.5 \;\; 0 \;\; 0.5 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^T$$

$$v = [0.005 \;\; 0.02 \;\; 0.005 \;\; 0.04 \;\; 0.05 \;\; 0.34 \;\; 0.2 \;\; 0.34]^T$$

$$d = [1 \;\; 0.15 \;\; 1 \;\; 0.12 \;\; 0.12 \;\; 1 \;\; 0.4 \;\; 1]^T$$

$$\lambda = 0.975$$

where A_a is the adjacency matrix of the bin connections, N_a is the total number of agents, $x(0)$ is the initial distribution of agents, v is the desired final distribution as in equation (2.3), d is a safety upper bound constraint (as in equation (2.2) with $L = I$ and $p = q = d$), and λ is the convergence rate of the system.

Since the bins at the corners behave like accumulation points in the initial and final distributions, the safety upper bound constraints are not imposed for these bins, i.e., the corresponding entries of the d vector are set as 1. With the given parameters, the optimization problem given in (5.37) with (3.3) is solved using YALMIP and SDPT3 [77, 102]. G_k matrices used in the simulations and

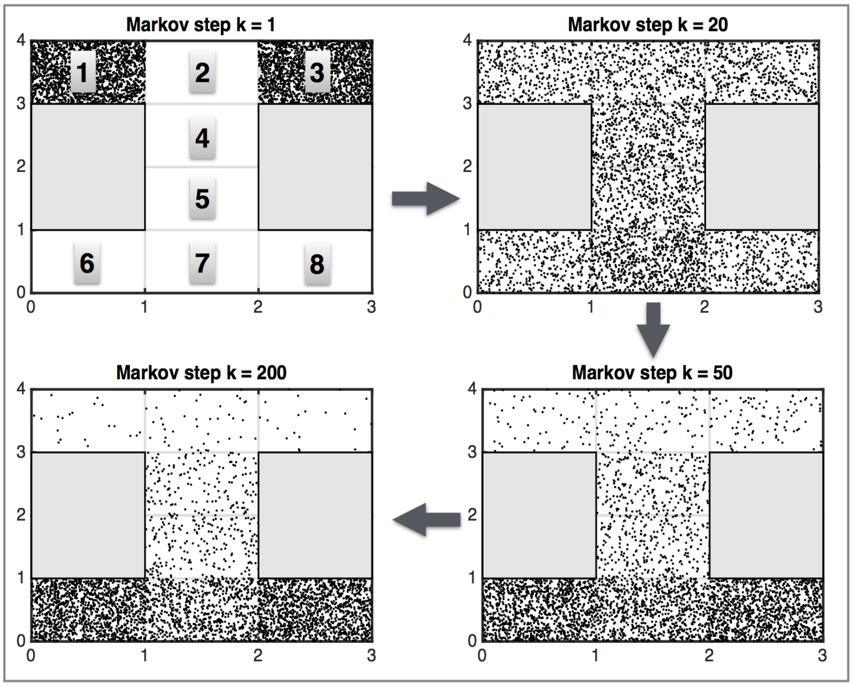

Figure 5.4: Snapshots of simulation: configuration space with bin numbers

the final distributions, v_k, corresponding to each G_k are given in the appendix along with the resulting solution variables Q_k and α_k. In many applications, environmental transition matrices, G_k's, may satisfy transition constraints in most examples, i.e., $G_k[i, j] = 0$ when $A_a[j, i] = 0$. However this example considers some G_k matrices that do not have this property, that is, we do not allow some motions even when they can be induced by the environment. For example, as given in the appendix, $G_1[4, 1] > 0$ even though $A_a[1, 4] = 0$ and hence $Q_1[4, 1] = 0$. This makes the problem more challenging since motion constraints are not automatically satisfied by the environment and they must be ensured by the policy. Such scenarios can arise in the balloon motion control example given in the introduction [46, 71, 105] where environmental transition matrices may not satisfy the desired constraints. Though some altitudes may induce high velocities, we may not choose to ride with such fast winds, for example, not to damage the structural integrity of the balloon (there may be a maximum speed limit for structural safety purposes).

Simulation results are presented in Figure 5.5. The mean density $\mathbf{x}$ and 3σ confidence bounds are shown for the case with density upper bound d. The average density for the case without constraints is obtained by evolving the density according to equation (2.6). Density goes above the desired upper bound for the bins $2, 4$ and 5 when the constraint is not imposed. By using ON/OFF control policy, we are able to ensure that the density does not go beyond the prescribed upper limit at a reasonable cost of reduced convergence rate.

For the results of the second set of simulations, point-wise maximum values of the density at each time step over all 3000 simulations are plotted. This is a good demonstration of our claim in Lemma 1: For all safe initial conditions, i.e., $x(0) \leq d$, the density is guaranteed to satisfy safety constraints for all time, i.e., $x(t) \leq d,\ t \in \mathbb{N}^{+}$.

The snapshots of the overall distribution taken at the beginning and at the end of a simulation from the first set are shown in Figure 5.4 which also shows the bin numbers.

5.3 Connections to Markov Decision Processes

The model presented here can be considered as a *controlled Markov chain* model, since for any designed decision variables the whole system behaves as a Markov chain that is given by Theorem 6. We consider a discrete action space

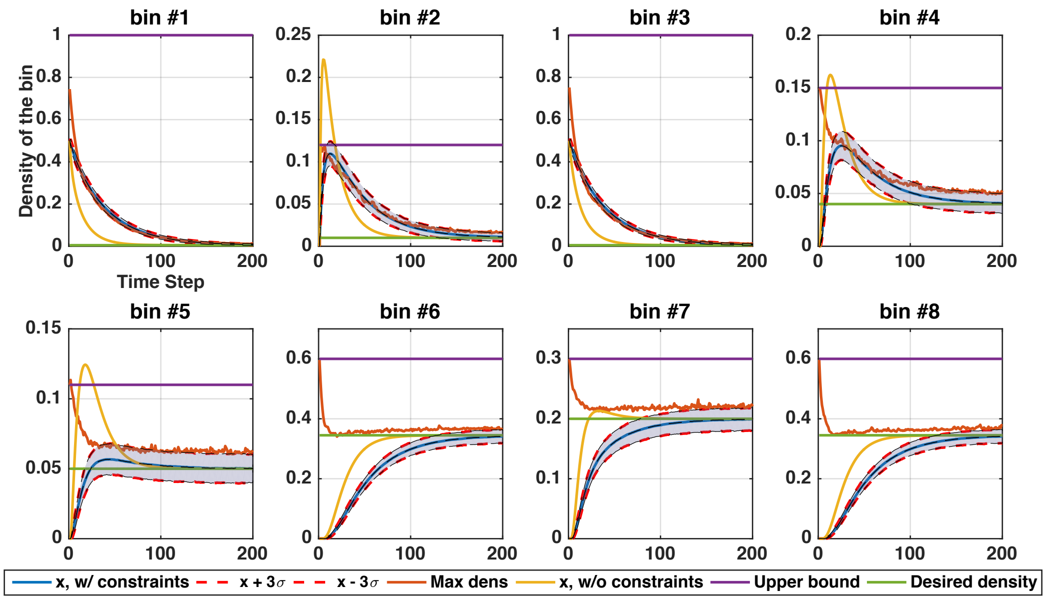

Figure 5.5: Time history of the density of each bin. With the density upper bound constraints the density is guaranteed to stay in the tube between red dashed lines with 99.7% confidence.

and transitional probabilities on a discretized finite state-space corresponding to each action, hence there is an inherent connection to Markov Decision Processes (MDPs). Our main objective is to design the ON/OFF decision variables, hence design the transition matrix M, to achieve a density distribution with desired constraints such as *safety* or a desired stationary distribution. In MDPs, differently from our model, the main objective is to select policies that optimize a reward-based function. Here, we do not explicitly define a reward function, instead we can implicitly include an optimized decision-making policy to our system thorough the OFF mode. In our model, we utilize additional observations in the ON mode where, the case where observations are not available is captured by the OFF mode. MDP models, in general, do not observe the outcome of actions before transition, hence the OFF mode can correspond to a Markov chain resulting from an MDP with a standard decision policy, i.e., the transition matrix for the OFF mode would correspond to underlying Markov chain of an MDP with a reward-optimized policy π: $G_{\mathit{off}} = M^{\pi}_{\text{MDP}}$. Here, the default MDP policy is not guaranteed to satisfy desired steady-state distribution or the safety constraints. In this case, we utilize the additional observations in the ON mode via the synthesized decision policies to achieve these additional constraints. We provide a simple motion planning example

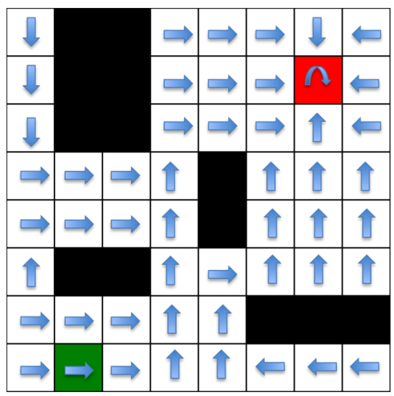

Figure 5.6: An MDP optimal policy for deterministic motion planning where the objective is to go from the green bin to the red bin. Black squares are obstacles.

in Figure 5.6, which illustrates that the mode-switching agent model can implicitly incorporate a reward-optimizing MDP policy. In this example, the objective is to send an agent from a source state, the green bin, to a destination state, the red bin, using the least number of transitions, which corresponds to moving along the shortest path. The arrows in the figure show an optimal MDP policy to go from any state to the destination state using the shortest path. This optimal policy can be derived using the well-established backward induction algorithm in dynamic programming. It should be noted that the environment is assumed to be non-stochastic in this example, so transitions due to the actions of the optimal policy are assumed to be deterministic. This means that if many agents start from the same green bin, they will all follow the same path. Thus, if bins are subject to varying capacity constraints (i.e., each bin has a maximum capacity for the number of agents that can be present in that bin at any instant of time), then the optimal MDP policy violates the constraints.

The mode-switching agent model explained here could be used to handle the following case, where the OFF mode is the transition matrix corresponding to the optimal MDP policy given in Figure 5.6. Since the environmental transitions are deterministic, then so are the observations. Thus, $\{\alpha_k, k \in \mathbb{N}_m^+\}$ boils down to the probability of choosing an action that deviates from the optimal policy to satisfy the capacity constraints. Such capacity constraints can be imposed directly on the transition matrix of the system that is obtained by substituting the relevant quantities into equation (2.6). By doing so, we obtain a 'randomized' policy that deviates from the optimal policy to satisfy the imposed constraints. In this example, the set of actions for ON and OFF modes are the same, where OFF mode executes a standard MDP policy (e.g., choosing actions for the shortest path) and ON mode executes an action in order to deviate from the policy in the OFF mode.

5.4 Summary

In this section, we develop a probabilistic density control policy for autonomous mobile agents with two modes: ON and OFF. When the agent is in the ON mode, it can observe the one-step outcome of a single action chosen from actions for the ON mode and can decide whether to take this action or not. If it does not take the action, it switches to the OFF mode. The density distribution of agents in the system evolves according to a Markov chain that, as we proved, is a linear function of the stochastic environment and the decision policy. We formulate a convex optimization problem, that can be solved reliably via interior-point methods, to synthesize the decision policy which ensures desired safety, transition and convergence properties for the underlying Markov chain. The given constraints on the density are equivalently expressed as constraints on the Markov chain. The resulting density control model is illustrated with a numerical example on autonomous mobile agents.

Chapter 6

CONCLUSIONS AND FINAL REMARKS

6.1 Summary of the contributions

This dissertation presented a Markov chain based method for controlling the probabilistic density distribution to coordinate a swarm of mobile agents. The proposed method allows each agent to make statistically independent decisions based on a Markov chain, hence it's implementable in a decentralized manner. We presented an LMI formulation of the Markov chain synthesis problem with the density safety, ergodicity and transition constraints.

One of the main contribution of the research is new necessary and sufficient conditions for the density safety constraints. The result presented here is novel in the Markov chain literature and fills an important void in the previous work with its convex optimization formulation of the density safety constraints in Markov chain synthesis. We obtained useful results by applying these new synthesis results on the Markov chain based decentralized vehicle swarm coordination problems. We also introduced a new quadratic programming based online synthesis method for time-varying Markov matrices. This method allows updating the Markov matrix based on the real-time density feedback which results in better convergence. For this purpose, we also introduced a decentralized counting algorithm for density estimation, which converges to identical estimates for all agents in a predetermined number of communication updates. For both methods, we presented numerical simulations that shows the effect of different type of safety constraints.

We presented simulation results for probabilistic density control method applied to a Low-earth orbiting swarm in 2D. The probabilistic guidance algorithm is adapted to earth orbiting swarms using Hill's equations for the relative dynamics of each spacecraft to the circular orbit defined in local-vertical local horizontal (LVLH) coordinates.

In the last part of the dissertation we considered autonomous mobile agents with two modes: ON and OFF. The main distinction of this research from the existing work is the existence of the ON control mode and its observed actions. This allows us to devise new methods to control the density distribution of autonomous agents via a new Markov decision model with measurements on the state transitions. Measurements for the ON mode can be obtained by the deployment of additional sensors to extend the agents' sensing capabilities. Hence, the key contributions of the research are: i) Formulation of a new Markov chain synthesis problem through a new Markov decision model, with additional measurements for the state transitions, where a policy is designed to ensure that the desired safety and convergence properties for the underlying Markov chain; ii) Convexification of the synthesis problem; iii) Application of the model to density control of swarm of autonomous mobile agents.

6.2 Future work

The methodology for the probabilistic density control of multi-agent system may build a foundation for various research directions. The dissertation presented novel convex necessary and sufficient conditions for density safety constraints in Markov chains using the duality theory of optimization. These conditions construct invariant safe sets for the Markov chains. Further analysis can be pursued to find the largest possible invariant sets and tightest possible bounds for a given initial condition. These results may also be further generalized to positive systems to find conditions to bound the system states.

Another research direction is to construct low-level guidance policies that will follow the density commands and generate collision free trajectories for each agent in a decentralized manner. [47] presents preliminary results on a velocity field approach to achieve safe, desired density distributions. The method synthesizes smooth velocity fields, which specify a desired velocity as a function of time and position in a decentralized manner and performs really well in terms of generating optimal and collision-free trajectories.

Further research can also be pursued on the discretization of the time and state-space which requires an extensive trade-for accuracy and computational efficiency. There are existing methods that can rigorously certify the validity of these reachability relationships by using finite horizon control problems with agent dynamics, and control and state constraints, e.g., [54, 79, 42], which describe the motions between temporal stages. These methods can be used in a systematic construction of the PDC problem description with bins, motion constraints, and desired final density distribution based on the decomposition of the region.

In this dissertation, we also introduced a new Markov decision model where the mission objectives are embedded within the underlying Markov chain rather than having a reward function for each action and transition. One may also consider a cost for having an observation causing the deviation from an optimal policy. Having costs for observations and comparing the effect of deviation on the overall performance of the resulting MDP may also be another research direction.

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, N. Demir, B. Açıkmeşe, and M. Harris. Convex necessary and sufficient conditions for density safety constraints in markov chain synthesis. *IEEE Trans. on Automatic Control*, 60(10):2813–2818, 2015.

[3] B. Açıkmeşe and D. Bayard. Markov chain approach to probabilistic guidance for swarms of autonomous agents. *Asian Journal of Control*, 17(4):1105–1124, 2015.

[4] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith. Wind-energy based path planning for unmanned aerial vehicles using markov decision processes. In *2013 IEEE International Conference on Robotics and Automation*, pages 784–789, 2013.

[5] C. Amato, G. Chowdhary, A. Geramifard, N. K. Üre, and M. J. Kochenderfer. Decentralized control of partially observable markov decision processes. In *52nd IEEE Conference on Decision and Control*, pages 2398–2405. IEEE, 2013.

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

[7] 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.

[8] H. Bai, M. Arcak, and J. T. Wen. Adaptive design for reference velocity recovery in motion coordination. *Systems & control letters*, 57(8):602–610, 2008.

[9] S. Balachandran and E. Atkins. Markov decision process framework for flight safety assessment and management. *Journal of Guidance, Control, and Dynamics*, 40(4), 2016.

[10] Saptarshi Bandyopadhyay, Soon-Jo Chung, and Fred Y Hadaegh. Inhomogeneous markov chain approach to probabilistic swarm guidance algorithm. In *5th Int. Conf. Spacecraft Formation Flying Missions and Technologies*, 2013.

[11] D.S. Bayard and R.W. Jelliffe. A Bayesian approach to tracking patients having changing pharmacokinetic parameters. *Journal of pharmacokinetics and pharmacodynamics*, 31(1):75–107, 2004.

[12] L. D. Berkovitz. *Convexity and Optimization in R^n*. John Wiley & Sons, Inc., 2002.

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

[14] 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.

[15] S. Berman, V. Kumar, and R. Nagpal.

[16] Daniel S. Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4):819–840, 2002.

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

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

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

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

[21] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, 1994.

[22] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.

[23] Roberto Casarin, Radu Craiu, and Fabrizio Leisen. Interacting multiple try algorithms with different proposal distributions. *Statistics and Computing*, 23(2):185–200, 2013.

[24] M. El Chamie, Y. Yu, and B. Açıkmeşe. Convex synthesis of randomized policies for controlled markov chains with density safety upper bound constraints. *to appear in American Control Conference (ACC), 2016*, 2016.

[25] 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.

[26] K. L. Chung. *A Course in Probability Theory.* Academic Press, 2001.

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

[28] J. Cortés, S. Martínez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[29] R. Dai and M. Mesbahi. Optimal topology design for dynamic networks. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 1280–1285, Dec 2011.

[30] M.C. de Oliveira, J. Bernussou, and J.C. Geromel. Anew discrete-time robust stability condition. *Systems and Control Letters*, pages 261–265, 1999.

[31] N. Demir, B. Açıkmeşe, and M. Harris. Convex optimization formulation of density upper bound constraints in Markov chain synthesis. *American Control Conference*, pages 483– 488, 2014.

[32] N. Demir, B. Açıkmeşe, and C. Pehlivanturk. Density control for decentralized autonomous agents with conflict avoidance. *IFAC World Congress, South Africa*, pages 11715–11721, 2014.

[33] N. Demir and B. Açıkmeşe. Probabilistic density control for swarm of decentralized on-off agents with safety constraints. *American Control Conference (ACC), 2015*, pages 5238–5244, 2015.

[34] N. Demir, U. Eren, and B. Açıkmeşe. Decentralized probabilistic density control of autonomous swarms with safety constraints. *Autonomous Robots*, 39(4):537–554, 2015.

[35] N. Demirer, M. El Chamie, and B. Acikmese. Safe Markov Chains for Density Control of ON/OFF Agents with Observed Transitions. *Accepted to IEEE Transactions on Automatic Control*, 2017.

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

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

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

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

[40] A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

[41] D. Dueri, , J. Zhang, and B. Açıkmeşe. Automated custom code generation for embedded, real-time second order cone programming. *To appear in IFAC World Congress*, 2014.

[42] D. Dueri, B. Açıkmeşe, M. Baldwin, and R.S. Erwin. Finite-horizon controllability and reachability for deterministic and stochastic linear control systems with convex constraints. In *American Control Conference (ACC), 2014*, pages 5016–5023. IEEE, 2014.

[43] M. El Chamie and B. Açıkmeşe. Convex synthesis of optimal policies for markov decision processes with sequentially-observed transitions. In *2016 American Control Conference (ACC)*, July 2016.

[44] M. El Chamie, Y. Yu, and B. Açıkmeşe. Convex synthesis of randomized policies for controlled Markov processes with hard safety constraints. In *2016 American Control Conference (ACC)*, July 2016.

[45] K. Elamvazhuthi, C. Adams, and S. Berman. Coverage and field estimation on bounded domains by diffusive swarms. In *2016 IEEE 55th Conference on Decision and Control*. IEEE, 2016.

[46] A. Elfes, K. Reh, P. Beauchamp, N. Fathpour, L. Blackmore, C. Newman, Y. Kuwata, M. Wolf, and C. Assad. Implications of wind-assisted aerial navigation for titan mission planning and science exploration. In *Aerospace Conference, 2010 IEEE*, pages 1–7. IEEE, 2010.

[47] U. Eren and B. Acikmese. Velocity field generation for density control of swarms using heat equation and smoothing kernels. *IFAC World Congress*, 2017.

[48] S. Feyzabadi and S. Carpin. Risk-aware path planning using hirerachical constrained markov decision processes. In *2014 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, 2014.

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

[50] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible markov chains, with an application to the exclusion process. *The Annals of Applied Probability*, 1(1):62–87, 1991.

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

[52] G. Foderaro, S. Ferrari, and T. A. Wettergren. Distributed optimal control for multi-agent trajectory optimization. *Automatica*, 50(1):149 – 154, 2014.

[53] W.R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*, volume 2. CRC press, 1996.

[54] A. Girard and C. Le Guernic. Zonotope/hyperplane intersection for hybrid systems reachability analysis. In *Hybrid Systems: Computation and Control*, volume 4981 of *Lecture Notes in Computer Science*, pages 215–228. 2008.

[55] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta, 2013.

[56] D. Grünbaum and A. Okubo. *Modelling Social Animal Aggregations*. Springer Berlin Heidelberg, 1994.

[57] F. Y. Hadaegh, A. Johnson, D. S. Bayard, B. Acikmese, S.J. Chung, and R. Mehra. *New Guidance, Navigation, and Control Technologies for Formation Flying Spacecraft and Planetary Landing*. Advances in Control System Technology for Aerospace Applications, Lecture Notes in Control and Information Sciences (LNCIS), pp. 49-80, Springer-Verlag.

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

[59] C. Hemelrijk. *Self-Organisation and Evolution of Biological and Social Systems*. Cambridge University Press, 2005.

[60] J. M. Hereford. Analysis of a new swarm search algorithm based on trophallaxis. In *IEEE Congress on Evolutionary Computation*, pages 1–8, July 2010.

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

[62] 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.

[63] M. A. Hsieh, V. Kumar, and L. Chaimowicz. Decentralized controllers for shape generation with robotic swarms. *Robotica*, 26(5):691–701, 2008.

[64] S. Hsu, A. Arapostathis, and R. Kumar. On optimal control of Markov chains with safety constraint. *In Proc. of American Control Conference*, pages 4516–4521, 2006.

[65] 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.

[66] 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.

[67] S. Karlin and H.M. Taylor. *A First Course in Stochastic Processes*. Number v. 1. Academic Press, 1975.

[68] H. Kawano. Three dimensional obstacle avoidance of autonomous blimp flying in unknown disturbance. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 123–130, 2006.

[69] F. P. Kelly. *Reversibility and Stochastic Networks*. John Wiley & Sons, Inc., 1979.

[70] V. Krishnan and S. Martínez. Distributed Control for Spatial Self-Organization of Multi-Agent Swarms. *ArXiv e-prints*, May 2017.

[71] Y. Kuwata, L. Blackmore, M. Wolf, N. Fathpour, C. Newman, and A. Elfes. Decomposition algorithm for global reachability analysis on a time-varying graph with an application to planetary exploration. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3955–3960. IEEE, 2009.

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

[73] D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov chains and mixing times*. American Mathematical Soc.

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

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

[76] Jun S. Liu, Faming Liang, and Wing Hung Wong. The multiple-try method and local optimization in metropolis sampling. *Journal of the American Statistical Association*, 95(449):121–134, 2000.

[77] J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[78] 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.

[79] J. Maidens, S. Kaynama, I. Mitchell andM. Oishi, and G. Dumont. Lagrangian methods for approximating the viability kernel in high-dimensional systems. *Automatica*, 49(7):2017–2029, 2013.

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

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

[82] J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, 2010.

[83] Jacob Mattingley and Stephen Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

[84] 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.

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

[86] G. Modica and L. Poggiolini. *A First Course in Probability and Markov Chains*. Wiley, 2012.

[87] J. J. Monaghan. Smoothed particle hydrodynamics. *Reports on Progress in Physics*, 68(8):1703, 2005.

[88] Y. Nesterov and A. Nemirovsky. *Interior-point Polynomial Methods in Convex Programming*. SIAM, 1994.

[89] L. Pallottino, V. Scordio, A. Bicchi, and E. Frazzoli. Decentralized cooperative policy for conflict resolution in multivehicle systems. *IEEE Trans. on Robotics*, 23(6):1170–1183, 2007.

[90] 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.

[91] 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.

[92] L. C. A. Pimenta, G. A. S. Pereira, N. Michael, R. C. Mesquita, M. M. Bosque, L. Chaimowicz, and V. Kumar. Swarm coordination based on smoothed particle hydrodynamics technique. *IEEE Transactions on Robotics*, 29(2):383–399, 2013.

[93] M.L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[94] J. Qi, R. Vazquez, and M. Krstic. Multi-agent deployment in 3-d via pde control. *IEEE Transactions on Automatic Control*, 60(4):891–906, April 2015.

[95] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[96] A. Rahmani, M. Ji, M. Mesbahi, and M. Egerstedt. Controllability of multi-agent systems from a graph-theoretic perspective. *SIAM Journal on Control and Optimization*, 48(1):162–186, 2009.

[97] I. M. Ross. Linearized dynamic equations for spacecraft subject to j perturbations. *Journal of Guidance, Control, and Dynamics*, 26(4):657–659, 2003.

[98] S. Roy. Scaled consensus. *Automatica*, 51:259–262, 2015.

[99] E. Seneta. *Non-negative matrices and Markov chains; rev. version.* Springer series in statistics. Springer, New York, NY, 2006.

[100] O. Sigaud and O. Buffet. *Markov Decision Processes in Artificial Intelligence.* Wiley-IEEE Press, 2010.

[101] X. Tan, W. Xi, and J. S. Baras. Decentralized coordination of autonomous swarms using parallel gibbs sampling. *Automatica*, 46(12):2068 – 2076, 2010.

[102] R. H. Tutuncu, K. C. Toh, and M. J. Todd. Solving semidefinite-quadratic-linear programs using sdpt3. *Mathematical Programming*, 95(2):189–217, 2003.

[103] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.

[104] R. N. Smith W. H. Al-Sabban, L. F. Gonzalez. Extending persistent monitoring by combining ocean models and markov decision processes. In *2012 Oceans*, pages 1–10, 2012.

[105] M. T. Wolf, L. Blackmore, Y. Kuwata, N. Fathpour, A. Elfes, and C. Newman. Probabilistic motion planning of balloons in strong, uncertain wind fields. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1123–1129. IEEE, 2010.

[106] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems and Control Letters*, 53(1):65 – 78, 2004.

[107] G. Xu and D. Wang. Nonlinear dynamic equations of satellite relative motion around an oblate earth. *Journal of Guidance, Control, and Dynamics*, 31(5):1521–1524, 2008.

[108] R. Zhang, Y. Yu, M. El Chamie, B. Açıkmese, and D. Ballard. Decision-making policies for heterogeneous autonomous multi-agent systems with safety constraints. pages 546–552, 2016.

APPENDIX

Review of Primitive Matrices

This section gives brief definition of primitive matrices and useful facts.

Definition 6 *A non-negative matrix $M \in \mathbb{R}^{n\times n}$ is said to be primitive if it is irreducible and has only one eigenvalue of maximum modulus.*

Lemma 4 *A nonnegative matrix $M \in \mathbb{R}^{n\times n}$ is primitive if and only if $M^k > 0$ for some $k \in \mathbb{N}_+$ (see Thm.8.5.2 in [61]).*

Lemma 5 *If a matrix $M \in \mathbb{R}^{n\times n}$ is nonnegative and primitive then (see Thm. 8.5.1 in [61]):*

$$\lim_{t\to\infty}(\rho(M)^{-1}M)^t = xy^T > 0$$

where

$$Mx = \rho(M)x,\ \ M^T y = \rho(M)y,\ \ x > \mathbf{0},\ \ y > \mathbf{0}, \text{ and } x^T y = 1$$

The next result is proposed in [3] and it establishes an important connection with Perron-Frobenius theory by showing the relationship between the spectral radius condition and primitive matrices.

Lemma 6 *[3] Consider a column stochastic matrix $M \in \mathbb{R}^{n\times n}$ such that $Mv = v$ for some $v > \mathbf{0}$. Then: $\rho(M - v1^T) < 1$ if and only if M is a primitive matrix.*

Proof: Suppose M is primitive. By using Lemma 5:

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

Now using the identity

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

we get:

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

which implies $\rho(M - v1^T) < 1$. Next suppose $\rho(M - v1^T) < 1$. This implies that

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

Hence,

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

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

$$|M^k[i,j] - v[i]| < \epsilon, \quad \forall i,j \in \mathbb{N}_m^+$$

This implies that $M^k[i,j] > v[i] - \epsilon > \epsilon/2 > 0$ for all $k \geq q$ and $i,j \in \mathbb{N}_m^+$. Hence $M^q > 0$, which implies that M is primitive by using Lemma 4 ■

Numerical Data for Section 5.2.2

The problem parameters and the resulting solution variables for the numerical example are:

$$G_1 = 10^{-4}\begin{bmatrix} 6735 & 384 & 284 & 146 & 179 & 27 & 34 & 22 \\ 773 & 5252 & 290 & 339 & 337 & 24 & 42 & 22 \\ 680 & 1080 & 7611 & 245 & 141 & 116 & 135 & 83 \\ 505 & 1196 & 226 & 6428 & 278 & 95 & 436 & 122 \\ 202 & 891 & 389 & 613 & 6425 & 79 & 421 & 88 \\ 391 & 308 & 585 & 450 & 546 & 8012 & 506 & 1264 \\ 349 & 540 & 157 & 1339 & 1696 & 666 & 7437 & 379 \\ 365 & 349 & 458 & 440 & 398 & 981 & 989 & 8020 \end{bmatrix}$$

$$G_2 = 10^{-4}\begin{bmatrix} 1871 & 2359 & 1563 & 1590 & 1349 & 2650 & 2471 & 235 \\ 1616 & 82 & 1651 & 2019 & 1008 & 56 & 1159 & 1820 \\ 1269 & 194 & 283 & 1387 & 1117 & 924 & 2903 & 1992 \\ 1314 & 2350 & 2205 & 149 & 1586 & 522 & 402 & 1211 \\ 1209 & 695 & 1657 & 1766 & 1792 & 1928 & 101 & 1641 \\ 824 & 614 & 1911 & 1827 & 511 & 2070 & 1478 & 1032 \\ 1463 & 1806 & 155 & 778 & 1038 & 111 & 992 & 453 \\ 434 & 1900 & 575 & 484 & 1599 & 1739 & 494 & 1616 \end{bmatrix}$$

$$G_3 = 10^{-4}\begin{bmatrix} 4579 & 389 & 324 & 291 & 360 & 53 & 69 & 46 \\ 975 & 3954 & 359 & 293 & 320 & 47 & 84 & 43 \\ 822 & 1304 & 5688 & 490 & 282 & 233 & 270 & 167 \\ 1009 & 1171 & 451 & 5333 & 269 & 190 & 336 & 244 \\ 404 & 787 & 777 & 750 & 5460 & 158 & 301 & 175 \\ 783 & 617 & 1171 & 901 & 1092 & 7854 & 1012 & 697 \\ 699 & 1081 & 314 & 1062 & 1422 & 878 & 7083 & 432 \\ 729 & 697 & 916 & 880 & 795 & 587 & 845 & 8196 \end{bmatrix}$$

$$G_4 = 10^{-4} \begin{bmatrix} 5864 & 878 & 526 & 686 & 679 & 764 & 70 & 534 \\ 780 & 4938 & 783 & 671 & 516 & 768 & 799 & 744 \\ 571 & 906 & 4983 & 689 & 638 & 1027 & 761 & 424 \\ 795 & 1231 & 693 & 5033 & 461 & 124 & 849 & 814 \\ 687 & 771 & 593 & 925 & 4788 & 751 & 873 & 612 \\ 744 & 560 & 836 & 665 & 598 & 4581 & 951 & 1066 \\ 388 & 276 & 572 & 804 & 1530 & 807 & 4885 & 738 \\ 171 & 440 & 1014 & 527 & 790 & 1178 & 812 & 5068 \end{bmatrix}$$

$$G_5 = 10^{-4} \begin{bmatrix} 8890 & 377 & 244 & 0 & 0 & 0 & 0 & 0 \\ 572 & 6550 & 221 & 385 & 354 & 0 & 0 & 0 \\ 538 & 856 & 9535 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1221 & 0 & 7522 & 287 & 0 & 535 & 0 \\ 0 & 996 & 0 & 477 & 7390 & 0 & 540 & 0 \\ 0 & 0 & 0 & 0 & 0 & 8170 & 0 & 1830 \\ 0 & 0 & 0 & 1616 & 1969 & 455 & 7792 & 326 \\ 0 & 0 & 0 & 0 & 0 & 1375 & 1133 & 7844 \end{bmatrix}$$

$$v_1 = \begin{bmatrix} 0.0200 \\ 0.0200 \\ 0.0600 \\ 0.0600 \\ 0.0600 \\ 0.2900 \\ 0.2000 \\ 0.2900 \end{bmatrix}, v_2 = \begin{bmatrix} 0.1756 \\ 0.1191 \\ 0.1197 \\ 0.1246 \\ 0.1391 \\ 0.1246 \\ 0.0883 \\ 0.1090 \end{bmatrix}, v_3 = \begin{bmatrix} 0.0200 \\ 0.0200 \\ 0.0600 \\ 0.0600 \\ 0.0600 \\ 0.2900 \\ 0.2000 \\ 0.2900 \end{bmatrix}, v_4 = \begin{bmatrix} 0.1250 \\ 0.1250 \\ 0.1250 \\ 0.1250 \\ 0.1250 \\ 0.1250 \\ 0.1250 \\ 0.1250 \end{bmatrix}, v_5 = \begin{bmatrix} 0.0200 \\ 0.0200 \\ 0.0600 \\ 0.0600 \\ 0.0600 \\ 0.2900 \\ 0.2000 \\ 0.2900 \end{bmatrix}.$$

$$\alpha_1 = \begin{bmatrix} 0.0940 \\ 0.1139 \\ 0.0568 \\ 0.2006 \\ 0.1975 \\ 0.1656 \\ 0.1573 \\ 0.1477 \end{bmatrix}, \alpha_2 = \begin{bmatrix} 0.5594 \\ 0.5165 \\ 0.6630 \\ 0.1332 \\ 0.1125 \\ 0.1229 \\ 0.2167 \\ 0.1553 \end{bmatrix}, \alpha_3 = \begin{bmatrix} 0.1201 \\ 0.1036 \\ 0.0616 \\ 0.1457 \\ 0.1492 \\ 0.1846 \\ 0.2036 \\ 0.1488 \end{bmatrix}, \alpha_4 = \begin{bmatrix} 0.0941 \\ 0.0987 \\ 0.0969 \\ 0.1135 \\ 0.1676 \\ 0.1845 \\ 0.2150 \\ 0.1955 \end{bmatrix}, \alpha_5 = \begin{bmatrix} 0.0976 \\ 0.1464 \\ 0.0679 \\ 0.3711 \\ 0.3404 \\ 0.1896 \\ 0.1537 \\ 0.1903 \end{bmatrix}$$

$$Q_1 = 10^{-4} \begin{bmatrix} 6184 & 6205 & 7025 & 0 & 0 & 0 & 0 & 0 \\ 10000 & 5983 & 10000 & 4886 & 7151 & 0 & 0 & 0 \\ 7165 & 9178 & 7084 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 5888 & 4599 & 0 & 6142 & 0 \\ 0 & 9932 & 0 & 8553 & 5895 & 0 & 7780 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6645 & 9431 & 8970 \\ 0 & 0 & 0 & 10000 & 10000 & 10000 & 6141 & 10000 \\ 0 & 0 & 0 & 0 & 0 & 8186 & 10000 & 7044 \end{bmatrix}$$

$$Q_2 = 10^{-4} \begin{bmatrix} 5788 & 4625 & 5634 & 0 & 0 & 0 & 0 & 0 \\ 10000 & 5780 & 10000 & 2621 & 8247 & 0 & 0 & 0 \\ 9130 & 8424 & 5778 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 6133 & 2381 & 0 & 5928 & 0 \\ 0 & 9913 & 0 & 9595 & 6170 & 0 & 6557 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7643 & 10000 & 8495 \\ 0 & 0 & 0 & 10000 & 10000 & 8729 & 5970 & 10000 \\ 0 & 0 & 0 & 0 & 0 & 10000 & 9441 & 6867 \end{bmatrix}$$

$$Q_3 = 10^{-4}\begin{bmatrix} 6037 & 6198 & 6769 & 0 & 0 & 0 & 0 & 0 \\ 10000 & 6009 & 10000 & 5376 & 7016 & 0 & 0 & 0 \\ 7375 & 9295 & 6838 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 6006 & 4978 & 0 & 6045 & 0 \\ 0 & 9875 & 0 & 8594 & 5989 & 0 & 7564 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6406 & 10000 & 8110 \\ 0 & 0 & 0 & 10000 & 10000 & 10000 & 6069 & 10000 \\ 0 & 0 & 0 & 0 & 0 & 7443 & 9937 & 6929 \end{bmatrix}$$

$$Q_4 = 10^{-4}\begin{bmatrix} 6181 & 6542 & 6026 & 0 & 0 & 0 & 0 & 0 \\ 10000 & 6010 & 10000 & 4975 & 7535 & 0 & 0 & 0 \\ 7022 & 8884 & 6154 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 6170 & 4073 & 0 & 5961 & 0 \\ 0 & 9840 & 0 & 8807 & 5947 & 0 & 8956 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6455 & 10000 & 8130 \\ 0 & 0 & 0 & 10000 & 10000 & 10000 & 6073 & 10000 \\ 0 & 0 & 0 & 0 & 0 & 8266 & 9948 & 6375 \end{bmatrix}$$

$$Q_5 = 10^{-4}\begin{bmatrix} 6326 & 6227 & 7107 & 0 & 0 & 0 & 0 & 0 \\ 10000 & 5957 & 10000 & 3764 & 7591 & 0 & 0 & 0 \\ 7150 & 9133 & 7195 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10000 & 0 & 5857 & 3727 & 0 & 6123 & 0 \\ 0 & 9974 & 0 & 8794 & 5868 & 0 & 8017 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6913 & 6109 & 9742 \\ 0 & 0 & 0 & 10000 & 10000 & 10000 & 6134 & 10000 \\ 0 & 0 & 0 & 0 & 0 & 9090 & 10000 & 7037 \end{bmatrix}$$