NOVEL PROBABILISTIC AND DISTRIBUTED ALGORITHMS FOR GUIDANCE, CONTROL, AND NONLINEAR ESTIMATION OF LARGE-SCALE MULTI-AGENT SYSTEMS

BY

SAPTARSHI BANDYOPADHYAY

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Aerospace Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Doctoral Committee:

Associate Professor Soon-Jo Chung, Chair and Director of Research
Professor N. Sri Namachchivaya
Professor Petros G. Voulgaris
Assistant Professor Alexander Olshevsky
Doctor Fred Hadaegh, Jet Propulsion Laboratory, California Institute of Technology

Abstract

Multi-agent systems are widely used for constructing a desired formation shape, exploring an area, surveillance, coverage, and other cooperative tasks. This dissertation introduces novel algorithms in the three main areas of shape formation, distributed estimation, and attitude control of large-scale multi-agent systems.

In the first part of this dissertation, we address the problem of shape formation for thousands to millions of agents. Here, we present two novel algorithms for guiding a large-scale swarm of robotic systems into a desired formation shape in a distributed and scalable manner. These probabilistic swarm guidance algorithms adopt an Eulerian framework, where the physical space is partitioned into bins and the swarm's density distribution over each bin is controlled using tunable Markov chains. In the first algorithm - Probabilistic Swarm Guidance using Inhomogeneous Markov Chains (PSG–IMC) - each agent determines its bin transition probabilities using a time-inhomogeneous Markov chain that is constructed in real-time using feedback from the current swarm distribution. This PSG–IMC algorithm minimizes the expected cost of the transitions required to achieve and maintain the desired formation shape, even when agents are added to or removed from the swarm. The algorithm scales well with a large number of agents and complex formation shapes, and can also be adapted for area exploration applications. In the second algorithm - Probabilistic Swarm Guidance using Optimal Transport (PSG–OT) - each agent determines its bin transition probabilities by solving an optimal transport problem, which is recast as a linear program. In the presence of perfect feedback of the current swarm distribution, this algorithm minimizes the given cost function, guarantees faster convergence, reduces the number

of transitions for achieving the desired formation, and is robust to disturbances or damages to the formation. We demonstrate the effectiveness of these two proposed swarm guidance algorithms using results from numerical simulations and closed-loop hardware experiments on multiple quadrotors.

In the second part of this dissertation, we present two novel discrete-time algorithms for distributed estimation, which track a single target using a network of heterogeneous sensing agents. The Distributed Bayesian Filtering (DBF) algorithm, the sensing agents combine their normalized likelihood functions using the logarithmic opinion pool and the discrete-time dynamic average consensus algorithm. Each agent's estimated likelihood function converges to an error ball centered on the joint likelihood function of the centralized multi-sensor Bayesian filtering algorithm. Using a new proof technique, the convergence, stability, and robustness properties of the DBF algorithm are rigorously characterized. The explicit bounds on the time step of the robust DBF algorithm are shown to depend on the time-scale of the target dynamics. Furthermore, the DBF algorithm for linear-Gaussian models can be cast into a modified form of the Kalman information filter. In the Bayesian Consensus Filtering (BCF) algorithm, the agents combine their estimated posterior pdfs multiple times within each time step using the logarithmic opinion pool scheme. Thus, each agent's consensual pdf minimizes the sum of Kullback–Leibler divergences with the local posterior pdfs. The performance and robust properties of these algorithms are validated using numerical simulations.

In the third part of this dissertation, we present an attitude control strategy and a new nonlinear tracking controller for a spacecraft carrying a large object, such as an asteroid or a boulder. If the captured object is larger or comparable in size to the spacecraft and has significant modeling uncertainties, conventional nonlinear control laws that use exact feed-forward cancellation are not suitable because they exhibit a large resultant disturbance torque. The proposed nonlinear tracking control law guarantees global exponential convergence of tracking errors with finite-gain $\mathcal{L}_p$ stability in the presence of modeling uncertainties

and disturbances, and reduces the resultant disturbance torque. Further, this control law permits the use of any attitude representation and its integral control formulation eliminates any constant disturbance. Under small uncertainties, the best strategy for stabilizing the combined system is to track a fuel-optimal reference trajectory using this nonlinear control law, because it consumes the least amount of fuel. In the presence of large uncertainties, the most effective strategy is to track the derivative plus proportional-derivative based reference trajectory, because it reduces the resultant disturbance torque. The effectiveness of the proposed attitude control law is demonstrated by using results of numerical simulation based on an Asteroid Redirect Mission concept.

The new algorithms proposed in this dissertation will facilitate the development of versatile autonomous multi-agent systems that are capable of performing a variety of complex tasks in a robust and scalable manner.

To Haripriya, Subrata and Anuradha Bandyopadhyay.

Acknowledgments

This thesis would not have been possible without the support of many people. First, I would like to thank my adviser, Prof. Soon-Jo Chung, for giving me the opportunity to work in his lab. I will forever be indebted to him for his guidance and for helping me understand the quality and rigor necessary for research. Special thanks to Dr. Fred Hadaegh for his guidance and comments on my research as well as his support during my summer internships at the Jet Propulsion Laboratory. Also thanks to my committee members, Prof. Namachchivaya, Prof. Voulgaris, and Prof. Olshevsky, who offered valuable guidance and support.

I would like to thank my current and former lab mates Aditya Paranjape, Ashwin Dani, Alireza Ramezani, Daniel Morgan, Nitish Sanghi, Michael Dorothy, Junho Yang, Kevin Meier, Rebecca Foust, Shripad Gade, Xichen Shi, and Giri Prashant Subramanian for their insightful discussions and suggestions. Special thanks to Giri, Dan, and all of the other students who helped develop the formation flying testbed and the experimental results in this dissertation. I would also like to thank the administrative staff Staci McDannel, Kendra Lindsey, Heather Glanzer, and Angela Ellis for their help.

Finally, I would like to thank my friends Abhishek Gupta, Ashish Mishra, Mansi Agarwal, Ritwika Ghosh, Shripad, and Giri for their support, and especially for the tea sessions. I would like to thank my wife Haripriya for being my greatest support and inspiration, for being a great colleague and my best friend, for enduring this roller-coaster journey with me, and for enriching my life with her love, beauty, compassion, and humor. I also thank my parents Subrata and Anuradha Bandyopadhyay and my mother-in-law Vaidehi Narayanan for their constant support and encouragement throughout this long journey.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

PSG	Probabilistic Swarm Guidance
IMC	Inhomogeneous Markov Chain
HMC	Homogeneous Markov Chain
OT	Optimal Transport
OTP	Optimal Transport Problem
LP	Linear Programming
ARM	Asteroid Redirect Mission
NEO	Near Earth Orbit
DBF	Distributed Bayesian Filtering
BCF	Bayesian Consensus Filtering
FF	Formation Flying

List of Symbols

Chapter 2

$\boldsymbol{A}_k^j$	Motion constraints matrix of the j^{th} agent at the k^{th} time instant
$\mathcal{B}$	Compact convex Euclidean space over which the swarm is distributed
$B[i]$	Bins, where $1 \leq i \leq n_{\text{bin}}$
$\boldsymbol{M}_k^j$	Markov matrix of the j^{th} agent at the k^{th} time instant
$\boldsymbol{M}_{k,\text{sub}}^j$	Sub-matrix of the Markov matrix $\boldsymbol{M}_k^j$ that encapsulates the bin transition probabilities between recurrent bins
$\boldsymbol{S}_k^j$	Condition for escaping transient bins for the j^{th} agent at the k^{th} time instant
$\boldsymbol{C}_k$	Cost matrix at the k^{th} time instant
m_k	Number of agents in the swarm at the k^{th} time instant
n_{bin}	Number of bins
$\boldsymbol{p}_k^j$	Actual location of the j^{th} agent at the k^{th} time instant
$\boldsymbol{r}_k^j$	Actual bin position of the j^{th} agent at the k^{th} time instant
$\boldsymbol{x}_k^j$	Predicted position of the j^{th} agent at the k^{th} time instant
$\boldsymbol{\Theta}$	Desired formation shape (see Def. 2.2)
$\boldsymbol{\gamma}_k^j$	Optimum transference plan of the j^{th} agent at the k^{th} time instant
ϵ_{est}	Estimation error between $\boldsymbol{\mu}_k^\star$ and $\boldsymbol{\mu}_k^j$
$\boldsymbol{\mu}_k^\star$	Current swarm distribution at the k^{th} time instant
$\boldsymbol{\mu}_k^j$	Estimate of $\boldsymbol{\mu}_k^\star$ by the j^{th} agent at the k^{th} time instant
ξ_k^j	Hellinger distance based feedback gain of the j^{th} agent at the k^{th} time instant
$\xi_{\min}$	Smallest positive feedback gain

Chapter 3

$\mathcal{A}_k$	Adjacency matrix at the k^{th} time instant
$\mathfrak{b}$	Periodicity of communication network topology
$\boldsymbol{f}_k$	Target dynamics at the k^{th} time instant
$\boldsymbol{h}_k^i$	Measurement model of the i^{th} agent at the k^{th} time instant
$\mathcal{J}_k^i$	Inclusive neighbors of the i^{th} agent at k^{th} time instant
$\mathcal{L}_k^i$	Normalized likelihood function of the i^{th} agent at the k^{th} time instant
$\mathcal{L}_k^C$	Normalized joint likelihood function of all the agents at the k^{th} time instant
N	Number of agents in the network
$\mathcal{S}_k^i$	Prior pdf of the i^{th} agent at the k^{th} time instant
$\mathcal{T}_k^i$	Estimated likelihood function of the i^{th} agent at the k^{th} time instant
$\mathcal{U}_k^i$	Estimated KL-divergence-minimizing pdf of the i^{th} agent at the k^{th} time instant
$\mathcal{W}_k^i$	Posterior pdf estimate of the i^{th} agent at the k^{th} time instant
$\mathcal{W}_k^{C,i}$	Centralized posterior pdf of the i^{th} agent at k^{th} time instant
$\mathcal{X}$	State space of the target's states
$\boldsymbol{v}_k$	Process noise at the k^{th} time instant
$\boldsymbol{w}_k^i$	Measurement noise of the i^{th} agent at the k^{th} time instant
$\boldsymbol{x}$	General variable that spans $\mathcal{X}$ ($\boldsymbol{x} \in \mathcal{X}$)
$\boldsymbol{x}_k$	True states of the target at the k^{th} time instant
$\boldsymbol{x}_{k\|k-1}$	Predicted states of the target at the k^{th} time instant
$\boldsymbol{x}_{k\|k}$	Updated states of the target at the k^{th} time instant
$\boldsymbol{y}_k^i$	Measurement taken by the i^{th} agent at the k^{th} time instant
Δ	Time step between any two time instants
$\Phi(\mathcal{X})$	Set of all pdfs over $\mathcal{X}$
γ	Minimum positive value in $\mathcal{A}_k$

Chapter 4

$\boldsymbol{B}$	Control influence matrices ($\boldsymbol{B} \in \mathbb{R}^{3\times n_t}$), where n_t is number of actuators
B_{CM}	Center of mass of the combined system
I_{sp}	Specific impulse of the fuel
$\boldsymbol{J}_{\text{obj}}^{B_{CM}}$	Inertia tensors of the object at B_{CM} in $\mathcal{F}_B$
$\boldsymbol{J}_{\text{sc}}^{S_{CM}}$	Inertia tensor of the spacecraft at S_{CM} in $\mathcal{F}_S$
$\boldsymbol{J}_{\text{tot}}^{B_{CM}}$	Inertia tensors of the combined system at B_{CM} in $\mathcal{F}_B$
$\hat{\boldsymbol{M}}(\hat{\boldsymbol{q}})$	Inertia matrices in the Euler–Lagrangian formulation
$\mathcal{P}(\cdot)$	Height of the power spectral density of the white noise
P_i	Thruster pods ($i = \{1, 2, 3, 4\}$)
$\boldsymbol{R}$	Rotation Matrices ($\boldsymbol{R} \in \mathbb{R}^{3\times 3}$)
$\mathbb{R}$	Real numbers
S_{CM}	Center of mass of the spacecraft
S_O	Point of contact between the spacecraft and the object
$\mathbb{S}^2$	Two-sphere ($\{(x_1, x_2, x_3) \in \mathbb{R}^3 : \|(x_1, x_2, x_3)\| = 1\}$)
$\mathbb{S}^3$	Three-sphere ($\{(x_1, x_2, x_3, x_4) \in \mathbb{R}^4 : \|(x_1, x_2, x_3, x_4)\| = 1\}$)
SO(3)	Special orthogonal group ($\{\boldsymbol{R} \in \mathbb{R}^{3\times 3} : \boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \mathbf{I}, \det(\boldsymbol{R}) = 1\}$)
$\boldsymbol{d}_{\text{ext}}$	External disturbance torque
$\boldsymbol{d}_{\text{res}}$	Resultant disturbance torques
m_{obj}	Mass of object
m_{sc}	Mass of spacecraft
$\boldsymbol{q}$	Modified Rodrigues Parameters ($\boldsymbol{q} \in \mathbb{R}^3$)
$\boldsymbol{q}_d$, $\boldsymbol{R}_d$	Desired attitude trajectory
$\boldsymbol{q}_e$, $\boldsymbol{e}_{\hat{\boldsymbol{R}}}$	Attitude error vector
$\boldsymbol{r}^{B/A}$	Vector from A to B
$\hat{\boldsymbol{s}}$	Unit vectors in $\mathcal{F}_S$ ($\hat{\boldsymbol{s}} = [\hat{s}_x, \hat{s}_y, \hat{s}_z]$)

$\boldsymbol{e}$, Φ	Euler Axis of Rotation ($\boldsymbol{e} \in \mathbb{R}^3$) and Angle ($\Phi \in \mathbb{R}^1$, rad)
g_0	Nominal acceleration due to the gravity (9.8 m sec^{-2})
$\boldsymbol{u}$	Actuator output ($\boldsymbol{u} \in \mathbb{R}^{n_t}$), where n_t is number of actuators
$\boldsymbol{u}_c$	Control input ($\boldsymbol{u}_c = \hat{\boldsymbol{B}}\hat{\boldsymbol{u}}$)
$u_{\max}$	Maximum thrust output of each thruster (N)
$\Delta(\cdot)$	Modeling, measurement or actuator error
$\boldsymbol{\beta}$	Quaternions ($\boldsymbol{\beta} \in \mathbb{R}^4$)
$\varepsilon_{\text{trans}}$	Transient error bound
ε_{ss}	Steady-state error bound
$\boldsymbol{\omega}$	Angular velocities ($\boldsymbol{\omega} \in \mathbb{R}^3$)
$\boldsymbol{\omega}_e$, $\boldsymbol{e}_{\hat{\boldsymbol{\omega}}}$	Angular velocity error vector

General

$\mathbf{I}$	Identity matrix
$\mathbb{N}$	Natural numbers (positive integers)
$\mathbb{P}(\cdot)$	Probability of an event
$\mathbb{R}$	Real numbers
$\lambda(\cdot)$	Eigenvalue of matrix
$\sigma(\cdot)$	Singular value of matrix
$\mathbf{0}$	Zero vector
$\mathbf{1}$	Ones (column) vector ($[1, 1, \ldots, 1]^T$)
$(\cdot)^j$	Agent index
$(\cdot)_k$	Time index
$\|\cdot\|_p$	ℓ_p vector norm
$\emptyset$	Empty set
$\|\cdot\|$	Absolute value or the cardinality of a set
$\lceil\cdot\rceil$	Ceiling function

Chapter 1

Introduction

Multi-agent systems are widely studied because they can collectively exhibit useful emergent behaviors like pattern (shape) formation [8]–[19], area exploration [20]–[25], and cooperative control [26]–[34]. For example, swarms of hundreds to thousands of femtosatellites (100-gram-class satellites) are being developed for challenging formation flying missions like synthetic aperture radar, distributed sensing, etc [35]. A recent survey paper lists state-of-the-art formation flying and constellation missions using small satellites and their applications [36, 37, 38]. Such multi-agent systems are popular because they can collaboratively complete tasks that are very difficult for a single agent, with significantly enhanced flexibility, adaptability, and robustness [35]. In this dissertation, we present the development of novel algorithms for shape formation, distributed estimation, and attitude control of multi-agent systems.

In the future, a large-scale swarm of agents may be deployed for challenging missions like constructing a desired formation shape or exploring an unknown environment. The path planning algorithm for such a large-scale swarm (having 10^3–10^6 or more agents) should be versatile (perform multiple tasks like maintaining the formation shape or exploring the area), robust (adapt to loss or addition of agents and accommodate large estimation errors), and scalable (easily scale with the number of agents and the size of the area). In the first part of this dissertation, we lay the theoretical foundations of versatile, robust, and scalable path planning algorithms for large-scale swarms using probabilistic swarm guidance (PSG).

PSG is centered on the idea of controlling the swarm density distribution in a distributed manner so that each autonomous agent independently determines its own trajectory while the

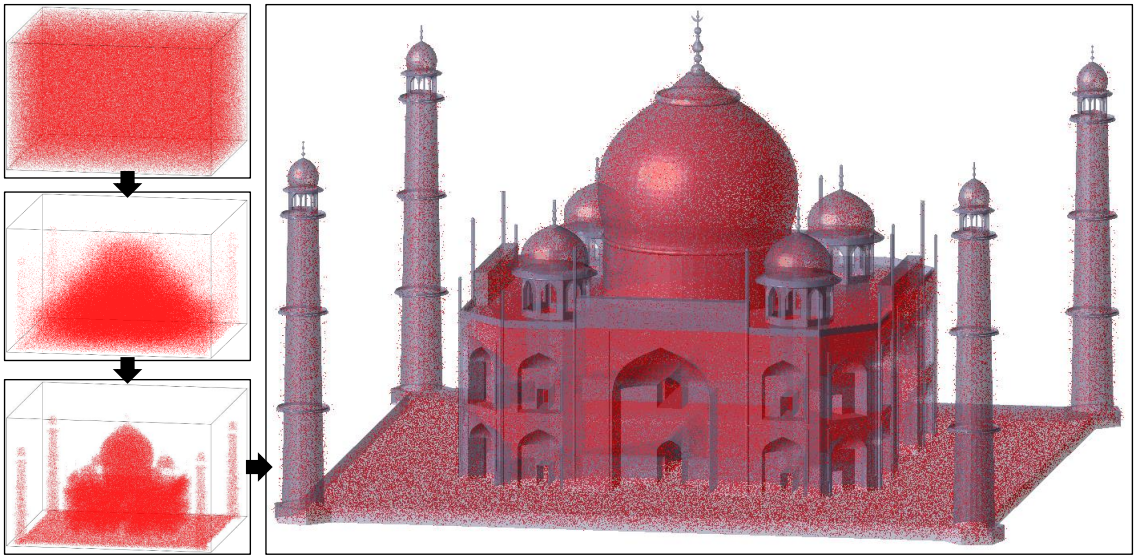


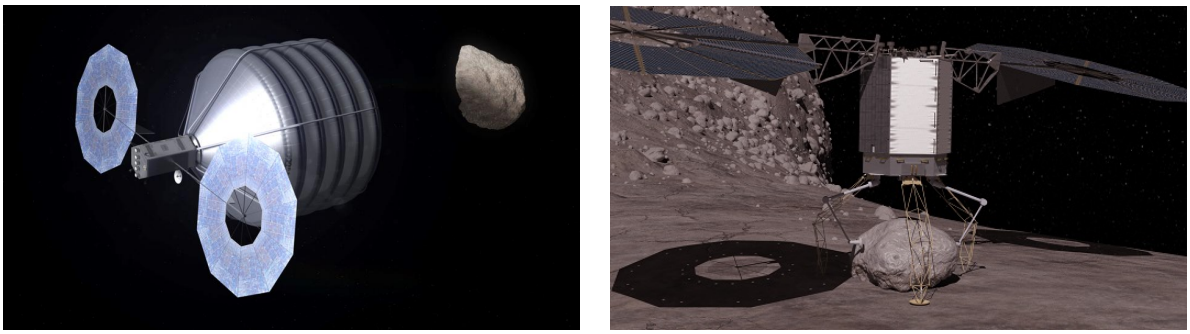
Figure 1.1: Using our PSG–IMC algorithm for shape formation, a million swarm agents (shown in red) attain the complex 3-D shape of the Taj Mahal (translucent silhouette shown in gray). The physical space is partitioned into $100 \times 100 \times 70$ bins.

entire swarm converges to the desired formation shape. PSG algorithms for shape formation using Inhomogeneous Markov Chains (IMC) and Optimal Transport (OT) are presented in Chapter 2. As an example, Fig. 1.1 demonstrates the capability of our proposed algorithm for attaining the complex 3-D shape of the Taj Mahal using one million agents.

Multi-agent systems are often used as a sensor network to track a target. Such a network of sensing agents uses a distributed estimation algorithm to estimate the states of a single target in a distributed manner. Potential applications of distributed estimation algorithms include environment and pollution monitoring, tracking mobile targets such as flying targets or space debris, and monitoring communication networks. In the second part of this dissertation, we present discrete-time distributed estimation algorithms for a heterogeneous sensor network that guarantee bounded convergence to the optimal probability distribution of the targets states. In these algorithms, the agents combine their local probability distributions using the logarithmic opinion pool and the consensus algorithm during each time instant. These distributed estimation algorithms are presented in Chapter 3.

Multi-agent systems can be used to cooperatively control the state of a large system.

A special case of this problem arises when a single agent (e.g., a spacecraft) is tasked with controlling the state of another captured object (e.g., an asteroid). As shown in Fig. 1.2, National Aeronautics and Space Administration's (NASA) proposed Asteroid Redirect Mission (ARM) aims to capture a Near Earth Orbit (NEO) asteroid or to pick up a boulder from some bigger asteroid and transport the captured object to the Earth–Moon system [1]. Other space agencies have also announced plans for future small body exploration and hazard mitigation missions [39, 40]. The third part of this dissertation is motivated by one of the main control challenges in the proposed ARM concept: despinning and three-axis attitude control of the combined spacecraft and asteroid system. The captured object could be significantly larger and heavier (10 to 100 times by mass) than the spacecraft and could have large uncertainties in its physical model.



(a) (b)

Figure 1.2: (a) Artist's rendering of the conceptual ARM spacecraft about to capture a NEO asteroid (image credit: NASA [3]). (b) Artist's rendering of the conceptual ARM spacecraft about to pickup a boulder from a large asteroid (image credit: NASA [4]).

This control problem also arises in other space applications. For example, a spacecraft tasked with removal of orbital debris has to stabilize the spacecraft–debris combination after capturing the debris [41]. Similarly, small satellites could be launched for the purpose of reviving obsolete satellites already in space or mining them for usable parts [42]. The main control problem in all these applications is that the spacecraft has to stabilize the attitude of the combined system after the spacecraft has captured a target object (e.g., asteroid, debris, and satellite) with large model uncertainties. Moreover, the captured object could be larger

or comparable in size to the spacecraft. In Chapter 4, we present novel control algorithms for stabilizing the attitude of a spacecraft that has captured a significantly larger object. The large uncertainties associated with the captured object gives rise to significant control challenges, like large resultant disturbance torques. Therefore, we present new attitude control algorithms and methods for designing desired attitude trajectories to address these challenges.

1.1 Literature Review

Analogous to fluid mechanics, path planning for swarms can be performed in two ways: the individual-agent-based *Lagrangian* framework and the continuum-based *Eulerian* framework. Most prior works [8]–[13], [23]–[28] use the Lagrangian framework, where each agent's trajectory is generated separately. In contrast, the Eulerian framework controls the collective properties of the swarm (e.g., its density distribution). In the Lagrangian framework, the computation cost for each agent's target assignment and trajectory generation increases very fast with a large number (10^3–10^6 or more) of agents [11, 12, 13]. The Eulerian framework decouples these two tasks by first solving the target assignment problem at appropriate spatial resolution. Moreover, the Lagrangian framework does not efficiently handle a loss or addition of agents, nor does it scale well with the size of the area and arbitrary formation shapes [8, 9, 10]. Therefore, we adopt the Eulerian framework to establish a highly scalable and robust guidance algorithm for formation reconfiguration and area exploration.

There exist prior results that employ the Eulerian framework for path planning [43]–[47]. For shape formation and reconfiguration applications, the physical space over which the swarm is distributed is first partitioned into discrete bins [48, 49]. The bin size is determined by the spatial resolution of the desired formation shape. Assuming that the number of agents is much larger than the number of non-empty bins, the density distribution of the swarm over these bins is controlled to achieve the desired spatial configuration.

Within the Eulerian framework, Homogeneous Markov Chain (HMC) based algorithms are a popular choice for shape formation [18, 19], area exploration [20, 21, 22], task allocation [30], and surveillance applications [34, 50]. In such algorithms, the agent's transition probability between bins is encoded in a constant Markov matrix that has the desired formation shape as its stationary distribution. Such an approach is probabilistic, as opposed to deterministic, because each agent determines its next bin location by inverse-transform sampling of the Markov matrix [51]. These HMC-based algorithms possess the aforementioned benefits of robustness and scalability, because addition or removal of agents from the swarm does not affect the convergence of the HMC to the stationary distribution. However, a major drawback of these HMC-based algorithms is that they are also inherently open-loop strategies and cannot incorporate feedback. Clearly, the efficiency of these algorithms can be greatly improved by refining the Markov matrix at each time step or at regular time intervals using the feedback of the current swarm distribution. Such refinement results in an Inhomogeneous Markov Chain (IMC), which is at the core of our algorithm. In Chapter 2, we derive the Probabilistic Swarm Guidance using Inhomogeneous Markov Chains (PSG–IMC) algorithm, which incorporates feedback from the current swarm density distribution at each time step. Hence, the PSG–IMC algorithm is a closed-loop guidance strategy that retains the original robustness and scalability properties associated with a Markovian approach. Another disadvantage of HMC-based algorithms is that they suffer undesirable transitions from bins that are deficient in agents to bins with surplus agents. Such undesirable transitions prevent the swarm from converging to the desired formation. The PSG–IMC algorithm avoids such undesirable transitions between bins, thereby reducing the control effort needed for achieving and maintaining the formation. This benefit also results in smaller convergence error to the desired formation shape than HMC-based algorithms.

If perfect feedback of the current swarm distribution is available, then the motion of swarm agents can be formulated as an optimal transport problem [52]. Optimal transport, which has applications in transportation, resource allocation, assignment, etc. [53, 54, 55];

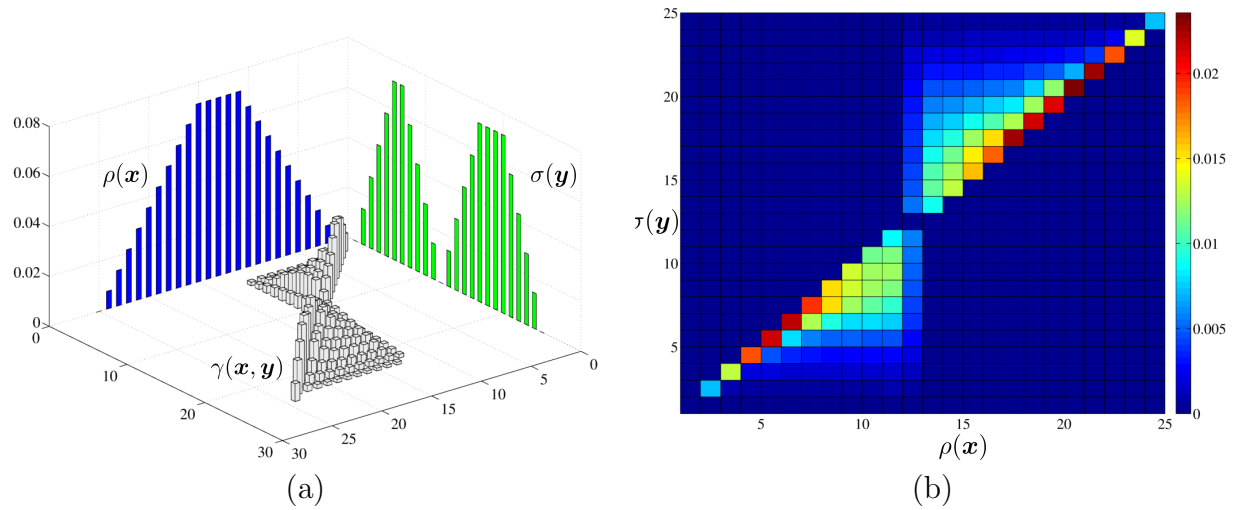


Figure 1.3: (a) The initial probability mass function (pmf) $\rho(\boldsymbol{x})$ (in blue) is transported to the desired pmf $\sigma(\boldsymbol{y})$ (in green) while minimizing the cost function $C(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|_2$. The optimum transference plan $\gamma(\boldsymbol{x},\boldsymbol{y})$ is shown in gray. (b) Another view of the optimum transference plan $\gamma(\boldsymbol{x},\boldsymbol{y})$ from $\rho(\boldsymbol{x})$ to $\sigma(\boldsymbol{y})$, where the colorbar represents the transported mass.

is used in Fig. 1.3 to find the optimum transference plan $\gamma(\boldsymbol{x},\boldsymbol{y})$ from an initial density distribution $\rho(\boldsymbol{x})$ to the desired swarm distribution $\sigma(\boldsymbol{y})$ which minimizes the cost function $C(\boldsymbol{x},\boldsymbol{y})$. The key concept of the Probabilistic Swarm Guidance using Optimal Transport (PSG–OT) algorithm proposed in Chapter 2 is to use optimal transport to find optimal trajectories for each agent in a distributed manner so that the given bin-based cost function is minimized while the swarm density distribution converges to the desired formation. Therefore, each agent independently solves the discrete optimal transport problem (OTP) using linear programming (LP) [56, 57] and then probabilistically selects the target bin from the OTP solution using inverse transform sampling [51]. LP and Mixed-Integer LP were previously used in a centralized manner for formation flying applications [58, 59] and path planning of robots [60]–[63]. Compared to existing distributed multi-agent assignment algorithms [64]–[67], each agent in our distributed bin assignment algorithm solves a LP with a significantly smaller number of variables and the resulting solution is optimal for the higher level bin-based cost function. However, there are two major disadvantages of such an approach. First, the performance of an optimal transport-based algorithm drops precipitously

with small errors in the feedback loop or small measurement errors of the current swarm distribution. Measurement and estimation errors are routinely encountered in practice and it is often impossible or impractical to generate perfect feedback of the current swarm distribution. Second, the computation time of the optimization problem increases very fast with an increasing number of bins, which is a notable drawback, because a large number of bins are necessary for capturing fine spatial details in the desired formation shape. The PSG–IMC algorithm can overcome both challenges, since it works efficiently in the presence of error-prone feedback and a large number of bins.

A different approach to swarm formation within the Eulerian framework is to model the continuum dynamics using partial differential equations (PDEs) [68]–[71]. Using this approach, agents can achieve a rich family of 2-D manifolds in 3-D space corresponding to the equilibrium states of the PDE [70]. Since the goal is to design guidance strategies for arbitrary formation shapes that are not limited to these equilibria, we do not consider a PDE-based approach in the present chapter.

Our approach is also different from the multi-agent Markov decision process approach used in multi-agent decision-making problems [72, 73], where agents keep track of the states and actions of other agents. In contrast, each agent transitions independently using the PSG–IMC algorithm.

Distributed Eulerian approaches for distributed estimation applications (like area exploration) using region-based shape controllers and attraction-repulsion forcing functions are discussed in [74, 75, 76]. We show that a slight modification of our PSG–IMC algorithm also results in an efficient area-exploration algorithm.

Each agent determines its higher-level bin-to-bin guidance trajectory using the PSG–IMC or PSG–OT algorithm so that the swarm achieves its objective. Each agent also needs a lower-level guidance and control algorithm, which depends on the agent's dynamics, to track this higher-level guidance trajectory in a collision-free manner. A recent survey paper [77] compares a number of existing collision avoidance algorithms based on model

predictive control, boundary following algorithms, artificial potential fields, etc. A lower-level algorithm based on model predictive control and sequential convex programming is presented in [78]. Voronoi partitions are widely used for collision avoidance by combining them with navigation functions [79] or potential field-based path planning algorithms [80]. In Chapter 2, we present a distributed collision-free trajectory generation algorithm for multi-agent path planning by modifying the Voronoi partition-based coverage control algorithm using the minimum collision avoidance distance [28].

Discrete-time distributed estimation algorithms are used when the target dynamics and measurement models are represented in discrete-time. Such algorithms can be broadly classified into three categories based on their representation of the target's states. Algorithms in the first category only estimate the mean and the covariance matrix of the target's states [81]–[85]. These algorithms usually deal with linearized target dynamics and measurement models. These algorithms also neglect the information in the higher-order moments of the estimated probability distribution of the target's states. The second category of algorithms aims to reach an agreement across the sensor network over a discrete set of hypotheses about the states of the target [86]–[88]. Although these algorithms use the entire information in the estimated probability distribution of the target's states, they are only applicable in cases where the target's states can be represented by a discrete (finite) set of hypotheses. Therefore, these algorithms are not suitable for estimation over continuous domains.

The third category of distributed estimation algorithms estimates the posterior probability distribution of the states of the target [89]–[95]. This category forms the most general class of distributed estimation algorithms because these algorithms can incorporate nonlinear target dynamics, heterogeneous nonlinear measurement models, and non-Gaussian uncertainties, and they can be used for estimation over continuous domains. These algorithms also use the entire information (i.e., not just the mean and the covariance matrix) in the estimated probability distribution of the target's states. In light of these advantages, Chapter 3 focuses on the development of a distributed estimation algorithm that belongs to

this category.

In the discrete-time distributed estimation algorithms that belong to the third category, the agents exchange their local probability distributions with their neighboring agents and combine them using fusion rules to estimate the aggregate probability distribution. Combining probability distributions in a distributed manner was first studied in the statistics literature [96]–[100]. Schemes for combining probability distributions, like the Linear Opinion Pool (LinOP) and the Logarithmic Opinion Pool (LogOP), are discussed in [99, 100]. The LogOP scheme is ideal for distributed estimation algorithms because of its favorable properties (e.g., externally Bayesian) [99].

The earliest distributed estimation algorithm that combined local probability distributions using the LogOP scheme was proposed in [89]. In particular, [90] focused on generating conservative information-theoretically-optimal weights for the LogOP scheme. Similar algorithms for combining probability distributions within the exponential family (i.e., probability distributions that can be expressed as exponential functions) are discussed in [91, 92]. In the distributed estimation algorithm in [94], the agents recursively combine their local posterior probability distributions multiple times within each time step using the LogOP scheme and the consensus algorithm. Using such an algorithm, which was also independently derived in our prior work [95], each agent's estimated probability distribution of the target's states converges around the pdf that minimizes the sum of Kullback–Leibler (KL) divergences from all the posterior probability distributions of the target's states. Similar algorithms for combining local likelihood functions multiple times within each time step using the LogOP scheme and the consensus algorithm are proposed in [92, 93]. Since these algorithms rely on the convergence properties of the standard consensus algorithm [101]–[104], the required number of consensus loops within each time step grows very fast (at least linearly [105]) with the number of agents. If the time-scale of the target dynamics is comparatively fast, then it is not feasible to execute a large number of consensus loops within each time step. This connection between the time-scale of the target dynamics and the time step of the distributed

estimation algorithm has not been explored in the literature. In contrast, in Chapter 3, we provide an explicit upper bound on the time step of the distributed algorithm that depends on the convergence error bound and the time-scale of the target dynamics. Furthermore, [94] analyzed their algorithm using linear-Gaussian models while [91] focused on probability distributions within the exponential family. A convergence analysis for general probability distributions using the LogOP scheme is currently missing in the literature. In Chapter 3, we also address this gap using a novel proof technique that was first introduced in our prior work [95].

An open question is how to design a distributed estimation algorithm for a sensor network so that each agent's estimate converges to the posterior probability distribution of the target's states that would be generated by a centralized multi-sensor Bayesian filtering algorithm. In other words, if all the agents are hypothetically connected by a complete graph (i.e., each agent could communicate instantaneously with every other agent without any loss of information in the communication links), then the agents can exchange their local likelihood functions and each agent can use the multi-sensor Bayesian filtering algorithm to estimate the posterior probability distribution of the target's states. Our aim is to design a distributed estimation algorithm where each agent's estimated probability distribution of the target's states converges to this posterior probability distribution. We assume that the time-varying communication network topology of the agents is periodically strongly connected and each agent can only communicate once with its neighboring agents during each time step. In Chapter 3, we present the Distributed Bayesian Filtering (DBF) algorithm to solve this open question.

Attitude control of a spacecraft with large uncertainty is a topic of intense research. Nonlinear adaptive attitude control strategies are discussed in [106]–[109]. In [110, 111, 112], sliding mode control and robust $\mathcal{H}_\infty$ linear control are used for attitude control of spacecraft with uncertainties and disturbances. In Chapter 4, we show that common nonlinear attitude control tracking laws that use exact feed-forward cancellation, similar to feedback lineariza-

tion, exhibit a large resultant disturbance torque due to unprecedentedly large modeling uncertainties of the captured object. In contrast, attitude control laws that do not have a feed-forward term (e.g., see [113]–[116]) experience a much smaller resultant disturbance torque. For the purpose of achieving superior robustness and tracking performance, nonlinear attitude tracking control should be used in lieu of linear control. Therefore, the proposed robust nonlinear tracking control law in Chapter 4 is designed to exploit the benefit of no feed-forward cancellation while achieving superior tracking performance in the presence of large modeling uncertainties, measurement errors, and actuator saturations.

1.2 Main Contributions

As stated previously, this dissertation focuses on the development of novel algorithms for shape formation, distributed estimation, and attitude control of multi-agent systems. The main contributions of this dissertations are as follows:

- In Section 2.2, we present a novel technique for constructing feedback-based Markov matrices for a given stationary distribution, where the expected cost of transitions is minimized. Each Markov matrix satisfies the constraints on the agent's motion that might arise due to the dynamics or other physical constraints. The Markov matrix converges to the identity matrix when the current swarm distribution converges to the desired formation. The identity matrix ensures that the swarm agents settle down after the desired formation is achieved, thereby reducing unnecessary transitions.

- In Section 2.3, we describe the PSG–IMC algorithm for shape formation, with the key idea that agents preferentially transition out of bins with surplus agents. We rigorously derive the convergence proofs for the PSG–IMC algorithm based on the analysis of IMC, which is more involved than the convergence proof for HMC. We show that each agent's IMC is strongly ergodic and that the unique limit is indeed the

desired formation shape. Further, we also provide a time-varying probabilistic-bound on the convergence error between the swarm distribution and the desired formation as well as a lower bound on the number of agents necessary for ensuring that the final convergence error is below the given threshold. We present an extension of the PSG–IMC algorithm that uses multiple spatial resolutions of the desired shape. Furthermore, for area exploration applications we show the benefits of the PSG–IMC algorithms as compared to HMC-based algorithms in Section 2.4.

- In Section 2.5, we solve the bin assignment problem in a distributed optimal manner using a discrete OTP. A centralized node could solve an OTP using the exact location of each agent and allocate the final positions to each agent. Instead, in PSG–OT, each agent independently solves an equivalent OTP using its best estimate of the current swarm distribution and then determines its trajectory in a distributed manner. This is possible because the agents reach a consensus on the estimate of the current swarm distribution, which is used as the initial distribution in the equivalent OTP. Note that the equivalent OTP solved by each agent has n_{bin}^2 variables compared to the centralized OTP with m^2 variables; where n_{bin} is the number of bins, m is the number of agents, and $m \gg n_{\text{bin}}$. Although it is possible to solve the OTP with m^2 variables in a distributed manner using the simplex algorithm [64, 65] or the auction algorithm [66, 67], these algorithms are not robust to addition or removal of agents from the swarm, which is the main benefit of probabilistic swarm guidance.

- In Section 2.6, we present a distributed collision-free trajectory generation algorithm suitable for the bin-based architecture of this chapter, which is obtained by modifying the Voronoi partition based Lloyd's descent algorithm [28]. We modify the Voronoi sets within each bin, using the minimum collision avoidance distance, to generate guaranteed collision-free trajectories for each agent. In essence, we present a complete distributed collision-free guidance algorithm for large-scale swarms by integrating the

collision-free trajectory generation algorithm with the PSG–IMC or PSG–OT algorithm.

- In Section 2.7, we demonstrate using multiple quadrotors that the PSG–IMC algorithm can be executed in real-time along with a lower-level collision-free motion planner to achieve a desired formation shape. Using results of numerical simulations, we show that the PSG–IMC algorithm yields a smaller convergence error and more robust convergence result than the HMC-based and PSG–OT algorithms since the distance between the swarm density distribution and the desired formation is considerably less. We also show that the PSG–IMC algorithm significantly reduces the number of transitions for achieving and maintaining the desired formation in the presence of large estimation errors. Thus, the PSG–IMC algorithm is best suited if error-prone feedback of the current swarm distribution is available, the swarm contains a large number of agents, or the desired formation contains a large number of bins.

- In Section 3.2, we present a rigorous proof technique for the LogOP scheme that is applicable for general probability distributions and some convergence results.

- In Section 3.3, we present the development of the DBF algorithm that does not require separate consensus loops within each time step. During each time instant, the agents exchange their normalized likelihood functions with their neighboring agents and combine them using our fusion rule. Our fusion rule for combining arbitrary probability distributions (not just exponential or Gaussian families) relies on the LogOP scheme and the discrete-time dynamic average consensus algorithm. We show that after finite time instants, the estimated likelihood function of each agent converges to an error ball centered on the joint likelihood function of the centralized multi-sensor Bayesian filtering algorithm. In order to guarantee the convergence error bound, we provide an explicit bound on the time step of the DBF algorithm that depends on the time-scale of the target dynamics. We perform robustness analysis to determine the

effect of modeling errors on the estimated likelihood functions. If the target dynamics are linear-Gaussian models, we show that the DBF algorithm can be simplified to the modified (Kalman) information filter. Finally, we show that the distributed estimation algorithms in [92, 93] are special cases of the DBF algorithm.

- In Section 3.4, we present the development of the BCF algorithm, where multiple consensus loops are allowed within each time instant. We also show that the BCF algorithm [94, 95] and the likelihood consensus algorithm [92, 93] are special cases of the distributed estimation algorithms presented in Chapter 3.

- In Section 4.1.2, we compare the resultant disturbance torques for different types of attitude control laws. We show that attitude control laws that use feed-forward cancellation experience a large resultant disturbance torque that can lead to actuator saturation. Therefore, such attitude control laws are not suitable for the present control problem.

- In Section 4.2, we present the development of a new robust nonlinear tracking control law that guarantees global exponential convergence of the system's attitude trajectory to the desired attitude trajectory and permits the use of any attitude representation. In the presence of bounded disturbances, this control law is finite-gain $\mathcal{L}_p$ stable and input-to-state stable. We show that this control law is related to the well-known tracking control law for Euler-Lagrangian systems [115, 117], but the new attitude tracking control law directly prescribes the control torque input with less dependence on the kinematic relationship. Another advantage of this new control law is that it can be easily extended with an integral control term to eliminate constant disturbances while retaining the original global exponential convergence property. Moreover, this exponentially-stabilizing attitude control law can be extended to employ SO(3) for global attitude representation.

- In Section 4.3, we discuss techniques for generating fuel-optimal desired attitude trajectories. We also outline a framework for reducing the resultant disturbance torque for the new attitude tracking control law.

- In Section 4.4, we demonstrate the effectiveness of our control strategy using results of numerical simulation based on an ARM mission type. We present a comparative study of the fuel usage and time of convergence of multiple attitude control laws. This study indicates that the best control strategy under very small modeling uncertainties, which can be achieved using online system identification from both proximity and contact operations, is to track the fuel-optimal reference trajectory using the globally-exponentially-stable robust nonlinear tracking control law. On the other hand, in the presence of large modeling uncertainties, measurement errors, and actuator saturations, the best control strategy is to have the robust nonlinear tracking control law track a derivative plus proportional-derivative based desired attitude trajectory. We also present a detailed sensitivity analysis of the robust nonlinear tracking control law to show that the fuel consumed by the conceptual ARM spacecraft using this control strategy is upper bounded by 300 kg for the nominal range of NEO asteroid parameters.

1.3 Organization

The key algorithms presented in this dissertation are shown in Fig. 1.4. The organization of this dissertation is as follows. The probabilistic swarm guidance algorithms for shape formation and area explorations are presented in Chapter 2. The distributed estimation algorithms are presented in Chapter 3. The nonlinear attitude control of spacecraft with a large captured object is presented in Chapter 4. This dissertation is concluded in Chapter 5.

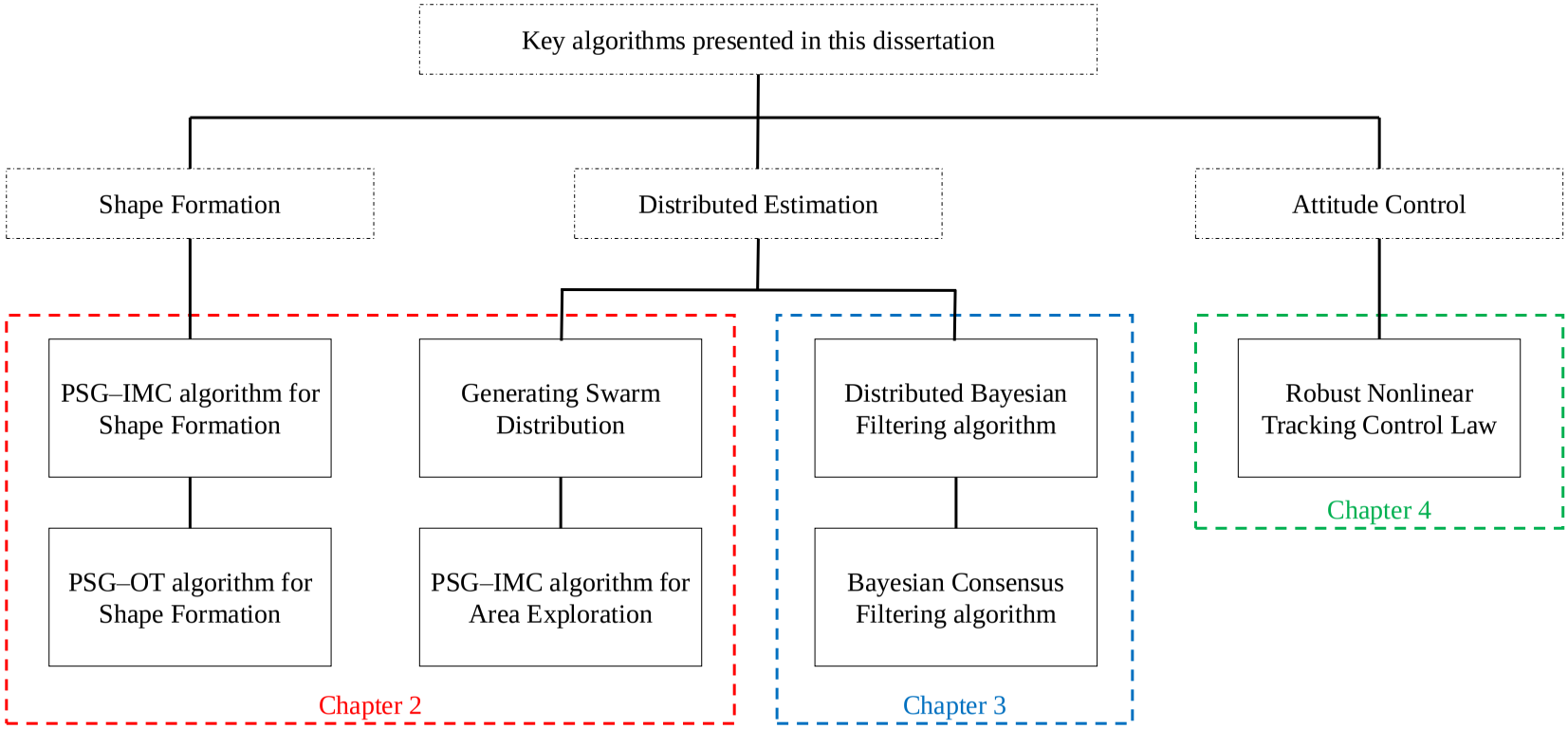


Figure 1.4: Key algorithms presented in this dissertation.

Chapter 2

Probabilistic Swarm Guidance

In this chapter, we present path planning algorithms for shape formation and area exploration. The algorithms presented in this chapter have been published in [118, 119, 120, 78, 121].

This chapter is organized as follows. First, the problem statement for shape formation is discussed in Section 2.1. In Section 2.2, we present our techniques for constructing feedback-based Markov matrices for a given stationary distribution. Subsequently, the PSG–IMC algorithm for shape formation and its convergence analysis are presented in Section 2.3. The extension of the PSG–IMC algorithm for area exploration applications is presented in Section 2.4. The PSG–OT algorithm for shape formation and its properties are presented in Section 2.5. The Voronoi-partition based collision-free trajectory generation algorithm is presented in Section 2.6. Lastly, in Section 2.7, Monte Carlo simulations and experimental results are used to compare these algorithms. This chapter is concluded in Section 2.8.

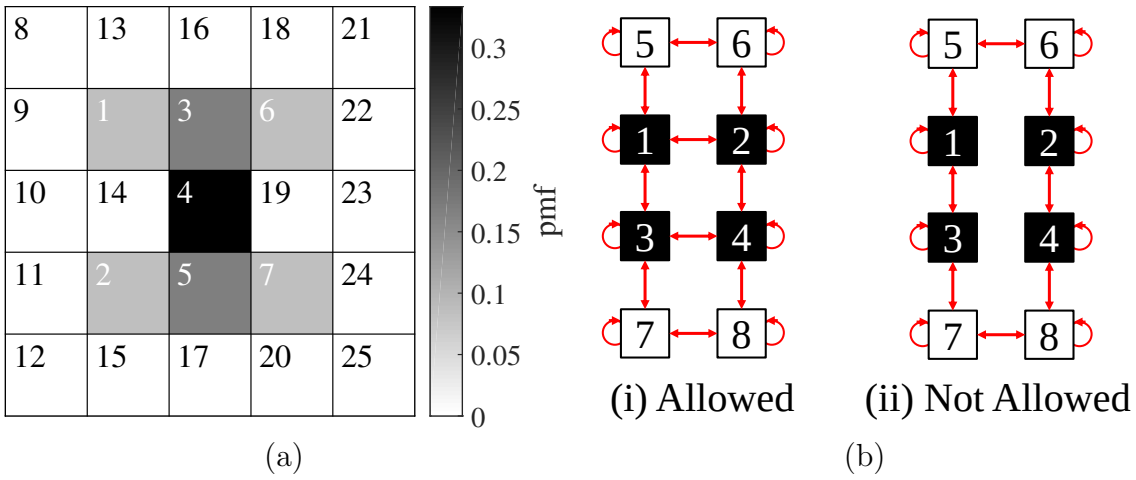
2.1 Preliminaries and Problem Statement

In this section, we first state some key terms and assumptions, then state the problem statement for shape formation in Section 2.1.1, and finally discuss methods for generating the feedback of the current swarm distribution in Section 2.1.2.

The n_x-dimensional compact convex Euclidean space over which the swarm is distributed is denoted by $\mathcal{B} \subset \mathbb{R}^{n_x}$.

Definition 2.1. *(Bins $B[i]$)* The space $\mathcal{B}$ is partitioned into n_{bin} mutually disjoint non-

empty convex bins. These bins are represented by $B[i]$ for all $i \in \{1, \ldots, n_{\text{bin}}\}$. If $i \neq \ell$ then $B[i] \cap B[\ell] = \emptyset$; and their union is $\mathcal{B}$, i.e., $\cup_{i=1}^{n_{\text{bin}}} B[i] = \mathcal{B}$ [122]. The convexity of the bins is used for formulating the OTP and generating collision-free trajectories. The size of the bins is determined by the spatial resolution of the desired formation shape. For example, the physical space is partitioned into 25 disjoint bins in Fig. 2.1(a). □



(a) (b)

Figure 2.1: (a) In this example, $n_{\text{bin}} = 25$. The desired formation $\boldsymbol{\Theta}$ is given by $[\frac{1}{12}, \frac{1}{12}, \frac{1}{6}, \frac{1}{3}, \frac{1}{6}, \frac{1}{12}, \frac{1}{12}, \mathbf{0}^{1\times 18}]$. The bins 1 to 7 are recurrent bins. (b) In this example, the bins 1 to 4 are recurrent bins. The allowed transitions (motion constraints) are shown in red. Case (i) satisfies all four properties of Definition 2.5. Case (ii) does not satisfy property (iv) because an agent in bin 1 cannot go to bins 2 or 4 without exiting the set of recurrent bins.

Definition 2.2. *(Desired Formation $\boldsymbol{\Theta}$ and Recurrent Bins)* The desired formation shape $\boldsymbol{\Theta}$ is a probability (row) vector in $\mathbb{R}^{n_{\text{bin}}}$ (i.e., $\boldsymbol{\Theta} \geq 0$, $\boldsymbol{\Theta}\mathbf{1} = 1$). Each element $\boldsymbol{\Theta}[i]$ represents the desired swarm density distribution in the corresponding bin $B[i]$. The bins that have nonzero elements in $\boldsymbol{\Theta}$ are called recurrent bins. Let n_{rec} denote the number of recurrent bins. The remaining bins, with zero elements in $\boldsymbol{\Theta}$, are called transient bins. Without loss of generality, we re-label the bins such that the first n_{rec} bins are recurrent bins (i.e., $\boldsymbol{\Theta}[i] > 0$ for all $i \in \{1, \ldots, n_{\text{rec}}\}$) and the remaining bins are transient bins (i.e., $\boldsymbol{\Theta}[i] = 0$ for all $i \in \{n_{\text{rec}} + 1, \ldots, n_{\text{bin}}\}$). For example, $\boldsymbol{\Theta}$ and recurrent bins are shown in Fig. 2.1(a). □

Definition 2.3. *(Cost Matrix $\boldsymbol{C}_k$)* Consider a matrix $\boldsymbol{C}_k \in \mathbb{R}^{n_{\text{bin}} \times n_{\text{bin}}}$, which captures the cost of transitioning between bins (e.g., control effort or fuel consumed by the agents). Each

element $\boldsymbol{C}_k[i, \ell]$ is the cost incurred by an agent while transitioning from bin $B[i]$ to bin $B[\ell]$ at the k^{th} time instant. We assume that the agents do not incur any cost if they remain in their present bin and the agents incur some positive cost if they transition out of their present bin, i.e., $\boldsymbol{C}_k[i, i] = 0$ and $\boldsymbol{C}_k[i, \ell] > 0$ for all bins $i, \ell \in \{1, \ldots, n_{\text{bin}}\}$ and $i \neq \ell$. □

Assumption 2.1. Let the scalar $m_k \in \mathbb{N}$ denote the number of agents in the swarm at the k^{th} time instant. The agents do not keep track of the number of agents in the swarm. We assume that $m_k \gg n_{\text{rec}}$, because we control the swarm density distribution over the bins.

Due to the quantization error $\frac{1}{m_k}$, we can only achieve the best quantized representation of the desired formation shape using m_k agents. For example, if $\boldsymbol{\Theta} = [\frac{1}{3}, \frac{2}{3}]$ and $m_k = 10$, then the best-quantized representation of this desired formation is [0.3, 0.7]. □

Assumption 2.2. We assume that the agents are *anonymous* and *identical*, i.e., the agents do not have any global identifiers and all agents execute the same algorithm [123]. If the agents are indexed (non-anonymous), then a spanning-tree-based algorithm can be executed [124], but this is not possible in our case. □

Assumption 2.3. We assume that each agent can determine its actual location in $\mathcal{B}$, which is denoted by $\boldsymbol{p}_k^j \in \mathbb{R}^{n_x}$. The indicator (row) vector $\boldsymbol{r}_k^j \in \mathbb{R}^{n_{\text{bin}}}$ represents the actual bin position the j^{th} agent at the k^{th} time instant. If the element $\boldsymbol{r}_k^j[i] = 1$, then the j^{th} agent is present inside the bin $B[i]$ at the k^{th} time instant; otherwise $\boldsymbol{r}_k^j[i] = 0$. □

Definition 2.4. *(Current Swarm Distribution $\boldsymbol{\mu}_k^\star$)* The current swarm distribution $\boldsymbol{\mu}_k^\star$ is a probability (row) vector in $\mathbb{R}^{n_{\text{bin}}}$. It is given by the ensemble mean of actual bin positions of the agents:

$$\boldsymbol{\mu}_k^\star := \frac{1}{m_k} \sum_{j=1}^{m_k} \boldsymbol{r}_k^j \,. \tag{2.1}$$

Each element $\boldsymbol{\mu}_k^\star[i]$ gives the swarm density distribution in the corresponding bin $B[i]$ at the k^{th} time instant. □

Definition 2.5. *(Matrix $\boldsymbol{A}_k^j$ of Motion Constraints)* An agent in a particular bin can only transition to some bins but cannot transition to other bins, because of the dynamics or

physical constraints. These (possibly time-varying) motion constraints are specified by the matrix $\boldsymbol{A}_k^j \in \mathbb{R}^{n_{\text{bin}} \times n_{\text{bin}}}$, where each element is given by:

$$\boldsymbol{A}_k^j[i,\ell] = \begin{cases} 1 & \text{if the transition from bin } B[i] \text{ to bin } B[\ell] \text{ is allowed at the } k^{\text{th}} \text{ time instant,} \\ 0 & \text{if this transition is not allowed.} \end{cases} \tag{2.2}$$

We assume that the matrix $\boldsymbol{A}_k^j$ satisfies the following properties: (i) the matrix $\boldsymbol{A}_k^j$ is symmetric, (ii) the graph conforming to the matrix $\boldsymbol{A}_k^j$ is strongly connected and balanced, (iii) an agent can always choose to remain in its present bin, i.e., $\boldsymbol{A}_k^j[i,i] = 1$ for all bins, and (iv) an agent can move (using multiple transitions) from any recurrent bin to any other recurrent bin, without exiting the set of recurrent bins. For example, property (iv) is visualized in Fig. 2.1(b). □

2.1.1 Problem Statement for Shape Formation

At the start, each agent has the knowledge of the bins $B[i]$, the desired formation shape $\boldsymbol{\Theta}$, the time-varying cost matrix $\boldsymbol{C}_k$ and its motion constraints matrix $\boldsymbol{A}_k^j$. The objectives of the PSG–IMC algorithm for shape formation are as follows:

(i) Each agent independently determines its bin-to-bin trajectory using a Markov chain, which obeys motion constraints $\boldsymbol{A}_k^j$, so that the overall swarm converges to a desired formation shape $\boldsymbol{\Theta}$.

(ii) The algorithm automatically detects and repairs damages to the formation.

(iii) The algorithm minimizes the expected cost of transitions of all the agents over all time instants.

2.1.2 Generating Feedback of the Current Swarm Distribution

The PSG–IMC algorithm uses the feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$. Several distributed estimation algorithms exist in the literature for estimating $\boldsymbol{\mu}_k^\star$ [82, 84, 95].

For example, the consensus algorithm [101, 102, 103] can be used to estimate the current swarm distribution $\boldsymbol{\mu}_k^\star$ in a distributed manner. Let the probability (row) vector $\hat{\boldsymbol{\mu}}_{k,\nu}^j \in \mathbb{R}^{n_{\text{bin}}}$ represent the j^{th} agent's estimate of the current swarm distribution during the ν^{th} consensus loop at the k^{th} time step. Each agent first locally estimates the swarm distribution, i.e., $\hat{\boldsymbol{\mu}}_{k,1}^j = \boldsymbol{r}_k^j$. Then the agents recursively combine their local estimates with their neighboring agents using the Linear Opinion Pool (LinOP) of probability measures [96, 99]:

$$\hat{\boldsymbol{\mu}}_{k,\nu+1}^j = \sum_{\ell \in \mathcal{J}_k^j} a_k^{\ell j} \hat{\boldsymbol{\mu}}_{k,\nu}^j, \; \forall j, \ell \in \{1, \ldots, m\}, \qquad \forall \nu \in \mathbb{N}, \tag{2.3}$$

where $\mathcal{J}_k^j$ is the set of inclusive neighbors of the j^{th} agent and $\sum_{\ell \in \mathcal{J}_k^j} a_k^{\ell j} = 1$. The pseudo-code for estimating the swarm distribution using the consensus algorithm is shown in Algorithm 1.

Consensus loop of the j^{th} agent during k^{th} time instant
Given $\boldsymbol{p}_k^j$, n_{loop}, and $a_k^{j\ell}$
for $\nu = 1$ to n_{loop}
    if $\nu = 1$ **then**
        Set $\hat{\boldsymbol{\mu}}_{k,1}^j = \boldsymbol{r}_k^j$
    end if
    Exchange pmfs $\hat{\boldsymbol{\mu}}_{k,\nu}^\ell, \forall \ell \in \mathcal{J}_k^j$
    Compute the new pmf $\hat{\boldsymbol{\mu}}_{k,\nu}^j$ using LinOP (2.3)
end for
Set $\boldsymbol{\mu}_k^j = \hat{\boldsymbol{\mu}}_{k,n_{\text{loop}}}^j$

Algorithm 1: Estimate Current Swarm Distribution $\boldsymbol{\mu}_k^j$

The matrix P_k, with entries $P_k[\ell, j] = a_k^{\ell j}$, conforms with the time-varying communication network topology of the multi-agent system $\mathcal{G}_k$. Since $m \gg n_{\text{bin}}$ and multiple agents are within the same bin, it is guaranteed almost surely using Erdös–Rényi random graphs or

random nearest–neighbor graphs that $\mathcal{G}_k$ is strongly connected [125, 126, 127]. Moreover, distributed algorithms exist for the agents to generate strongly connected balanced graphs [128, 129, 130].

If $\mathcal{G}_k$ is strongly connected and balanced, then each agents local estimate $\hat{\boldsymbol{\mu}}^j_{k,\nu}$ globally exponentially converges to the current swarm distribution $\boldsymbol{\mu}^\star_k$ pointwise with a rate faster or equal to the second largest singular value of P_k [131]. For some $\varepsilon_{\text{cons}} > 0$, if the number of consensus loops within each consensus stage $n_{\text{loop}} \geq \left\lceil \frac{\ln\left(\varepsilon_{\text{cons}}/(2\sqrt{m})\right)}{\ln \sigma_{m-1}(P_k)} \right\rceil$; then each agent has a good estimate of the current swarm distribution $\hat{\boldsymbol{\mu}}^j_{k,n_{\text{loop}}}$, i.e., $\|\boldsymbol{\theta}_{k,n_{\text{loop}}}\|_2 \leq \varepsilon_{\text{cons}}$, where $\boldsymbol{\theta}_{k,n_{\text{loop}}} = \left[\theta^1_{k,n_{\text{loop}}}, \ldots, \theta^m_{k,n_{\text{loop}}}\right]$ and $\theta^j_{k,n_{\text{loop}}} = \sum_{i=1}^{n_{\text{bin}}} |\hat{\boldsymbol{\mu}}^j_{k,n_{\text{loop}}}[i] - \boldsymbol{\mu}^\star_k[i]|$.

Let the j^{th} agent's estimate of the current swarm distribution at the k^{th} time instant be represented by the probability (row) vector $\boldsymbol{\mu}^j_k \in \mathbb{R}^{n_{\text{bin}}}$, where $\boldsymbol{\mu}^j_k = \hat{\boldsymbol{\mu}}^j_{k,n_{\text{loop}}}$ if the consensus algorithm is used. Let the positive parameter ϵ_{est} represent the estimation error between $\boldsymbol{\mu}^\star_k$ and $\boldsymbol{\mu}^j_k$, i.e., for all agents:

$$D_{\mathcal{L}_1}(\boldsymbol{\mu}^\star_k, \boldsymbol{\mu}^j_k) = \sum_{i=1}^{n_{\text{bin}}} \left|\boldsymbol{\mu}^\star_k[i] - \boldsymbol{\mu}^j_k[i]\right| \leq \epsilon_{\text{est}}\,, \; \forall k \in \mathbb{N}\,. \tag{2.4}$$

In Section 2.3.1, we show that our proposed PSG–IMC algorithm works remarkably well in the presence of this estimation error.

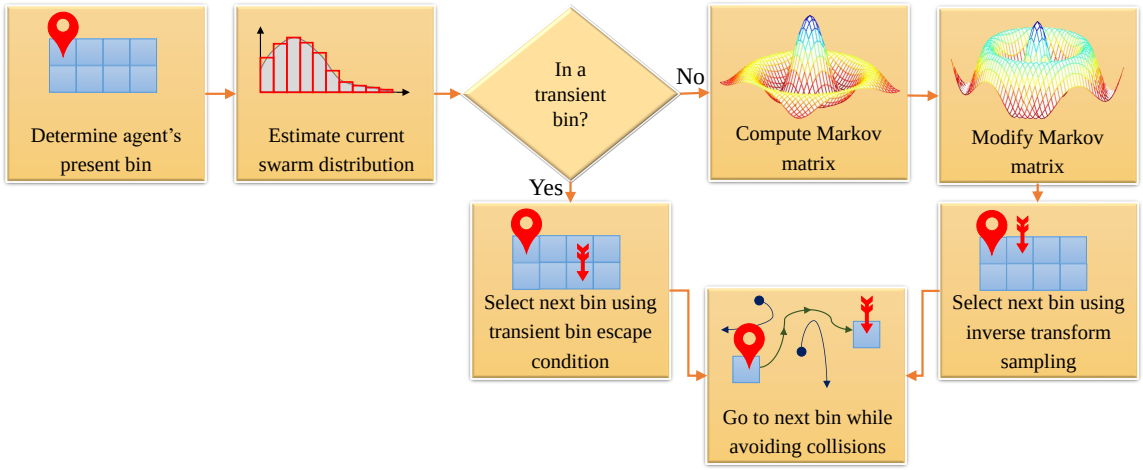


Figure 2.2: Flowchart of the PSG–IMC algorithm for shape formation.

2.2 Construction of Feedback-based Markov Matrix

The key steps in the proposed PSG–IMC algorithm for shape formation are shown in Fig. 2.2. The agent first determines its present bin and the current swarm distribution. If the agent's present bin is a transient bin, then it transitions to another bin using the condition for escaping transient bins. Otherwise, the agent computes the Markov matrix and then modifies it to avoid undesirable transitions. Finally, the agent uses inverse transform sampling to select the next bin. The pseudo-code for the PSG–IMC algorithm for shape formation is given in Algorithm 2 in Section 2.3.

In this section, we present techniques for constructing feedback-based Markov matrices and then discuss a condition to quickly exit transient bins.

2.2.1 Construction of Minimum Cost Markov Matrix

In this section, we construct Markov matrices that minimize the expected cost of transitions. We first discuss our choice of the feedback gain ξ_k^j, which is directly proportional to the transition probabilities between bins.

Remark 2.1. *(Hellinger Distance based ξ_k^j, $\xi_{\min}$, and ξ_{des})* The feedback gain ξ_k^j is based on the Hellinger distance (HD) between the current swarm distribution $\boldsymbol{\mu}_k^j$ and the desired formation $\boldsymbol{\Theta}$:

$$\xi_k^j = D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j) := \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^{n_{\text{bin}}} \left(\sqrt{\boldsymbol{\Theta}[i]} - \sqrt{\boldsymbol{\mu}_k^j[i]} \right)^2}. \tag{2.5}$$

The HD is a symmetric measure of the difference between two probability distributions and it is upper bounded by 1 [132, 133].

We choose HD, over other popular metrics like $\mathcal{L}_1$ and $\mathcal{L}_2$ distances, because of its important properties shown in Fig. 2.3. The $\mathcal{L}_1$ distances for the cases ($\boldsymbol{\mu}_1$, $\boldsymbol{\mu}_2$, $\boldsymbol{\mu}_3$) from $\boldsymbol{\Theta}$ are equal. But in Case 1, the wrong agent is in a bin where there should be no agent,

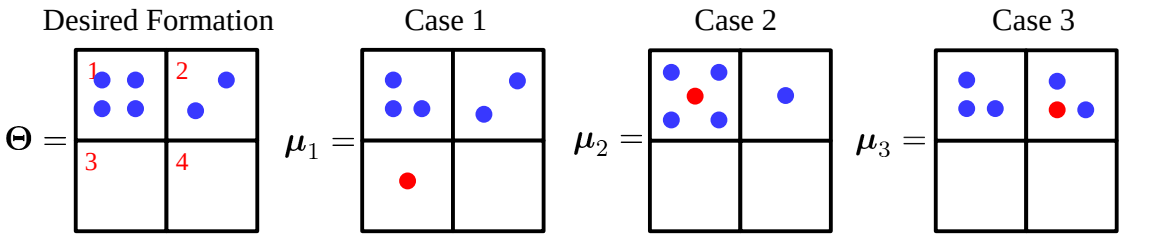


Figure 2.3: In this example, the desired distribution $\boldsymbol{\Theta}$ has 4 and 2 agents in bins 1 and 2 respectively. In the three cases, one agent (marked in red) is not in its correct bin. The $\mathcal{L}_1$ distances are equal, but the HD are different.

hence HD heavily penalizes this case. If all the agents are only in those bins which have non-zero weights in the desired distribution, then HD is significantly less. Finally, if an agent is missing from a bin that has a fewer number of agents in the desired distribution (Case 2) compared to a bin that has a higher number of agents in the desired distribution (Case 3), then HD penalizes Case 2 slightly more than Case 3. These important properties, which are encapsulated in HD, are useful for swarm guidance.

Let $\xi_{\min}$ represent the smallest positive feedback gain. If at least one agent is not in its correct bin, then $D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) = \sum_{i=1}^{n_{\text{bin}}} |\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| \geq \frac{1}{m_k}$. The HD between $\boldsymbol{\mu}_k^\star$ and $\boldsymbol{\Theta}$ is bounded by [134]:

$$\frac{1}{2\sqrt{2}} D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \leq D_H(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \leq \frac{1}{\sqrt{2}} D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta})^{\frac{1}{2}} . \tag{2.6}$$

Therefore, the smallest positive feedback gain is given by:

$$\xi_{\min} = \frac{1}{2\sqrt{2} \max\limits_{k \in \mathbb{N} \cup \{0\}} m_k} . \tag{2.7}$$

Note that $\xi_{\min}$ is very small because $m_k \gg n_{\text{rec}}$ in Assumption 2.1 (e.g., $\xi_{\min} = 3.5 \times 10^{-7}$ for $m_k = 10^6$).

Let ξ_{des} represent the acceptable convergence error between the final swarm distribution

and the desired formation. □

Consider the Markov matrix $\boldsymbol{M}_k^j$ in $\mathbb{R}^{n_{\text{bin}} \times n_{\text{bin}}}$ that encapsulates the transition probabilities between bins. Each element $\boldsymbol{M}_k^j[i, \ell]$ represents the probability that the j^{th} agent in bin $B[i]$ at the k^{th} time instant will transition to bin $B[\ell]$ at the $(k+1)^{\text{th}}$ time instant:

$$\boldsymbol{M}_k^j[i, \ell] := \mathbb{P}\left(\boldsymbol{r}_{k+1}^j[\ell] = 1 | \boldsymbol{r}_k^j[i] = 1\right) . \tag{2.8}$$

Therefore, the Markov matrix $\boldsymbol{M}_k^j$ is row stochastic (i.e., $\boldsymbol{M}_k^j \mathbf{1} = \mathbf{1}$). The stationary distribution of this Markov matrix is defined as follows.

Definition 2.6. [135, pp. 119] *(Stationary Distribution)* The stationary distribution $\boldsymbol{e}_k^j$ of the Markov matrix $\boldsymbol{M}_k^j$ is given by the solution of $\boldsymbol{e}_k^j \boldsymbol{M}_k^j = \boldsymbol{e}_k^j$, where $\boldsymbol{e}_k^j$ is a probability (row) vector in $\mathbb{R}^{n_{\text{bin}}}$ (i.e., $\boldsymbol{e}_k^j \geq 0$, $\boldsymbol{e}_k^j \mathbf{1} = 1$). This stationary distribution is unique if the Markov matrix is irreducible. □

The following theorem presents the construction of the Markov matrix $\boldsymbol{M}_k^j$ that minimizes the expected cost of transitions (i.e., $\sum_{i=1}^{n_{\text{bin}}} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i, \ell] \boldsymbol{M}_k^j[i, \ell]$). Our construction technique has no relation with the well-known Metropolis-Hastings (MH) algorithm, which is commonly used for constructing Markov matrices with a given stationary distribution [136, 137]. In the MH algorithm, the proposal distribution is used to iteratively generate the next sample, which is accepted or rejected based on the desired stationary distribution. There is no direct method for incorporating feedback into the MH algorithm. In contrast, the feedback of the current swarm distribution is directly incorporated within our construction process using the feedback gain.

Theorem 2.1. *The Markov matrix $\boldsymbol{M}_k^j$ is constructed as follows:*

(1) If $\xi_k^j < \xi_{\text{des}}$, where ξ_{des} is defined in Remark 2.1, then set $\boldsymbol{M}_k^j = \mathbf{I}$.

(2) Otherwise, the row stochastic Markov matrix $\boldsymbol{M}_k^j$ is computed as follows to minimize the expected cost of transitions. The resultant matrix $\boldsymbol{M}_k^j$ has $\boldsymbol{\Theta}$ as its stationary distribution (i.e., $\boldsymbol{\Theta} \boldsymbol{M}_k^j = \boldsymbol{\Theta}$) and only allows transitions into recurrent bins:

(CS1) *If* $\boldsymbol{A}_k^j[i,\ell] = 0$*, then set* $\boldsymbol{M}_k^j[i,\ell] = 0$ *for all bins* $i, \ell \in \{1, \ldots, n_{\text{bin}}\}$.

(CS2) *If* $\boldsymbol{\Theta}[\ell] = 0$*, then set* $\boldsymbol{M}_k^j[i,\ell] = 0$ *for all bins* $i, \ell \in \{1, \ldots, n_{\text{bin}}\}$ *with* $i \neq \ell$.

The remaining elements in the Markov matrix are determined using the following linear program (LP):

$$\min \sum_{i=1}^{n_{\text{bin}}} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell] \boldsymbol{M}_k^j[i,\ell]\,, \tag{2.9}$$

$$\textit{subject to} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{M}_k^j[i,\ell] = 1,\ \forall i\,, \tag{LP1}$$

$$\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i] \boldsymbol{M}_k^j[i,\ell] = \boldsymbol{\Theta}[\ell],\ \forall \ell\,, \tag{LP2}$$

$$(1 - \xi_k^j) \leq \boldsymbol{M}_k^j[i,i] \leq 1,\ \forall i\,, \tag{LP3}$$

$$\varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell] \left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max} + \varepsilon_C}\right) \leq \boldsymbol{M}_k^j[i,\ell] \leq \frac{\xi_k^j}{\varepsilon_M},\ \forall i \neq \ell\,, \tag{LP4}$$

where ε_M *is a positive scalar constant in* $(0, 1]$, $C_{k,\max}$ *is the maximum transition cost (i.e.,* $C_{k,\max} = \max_{i,\ell} \boldsymbol{C}_k[i,\ell]$*), and* ε_C *is a positive scalar constant.*

Proof: Assuming that the feedback gain $\xi_k^j \geq \xi_{\text{des}}$, we first show that the feasible set of Markov matrices constructed from the constraints (CS1,2) and the LP (2.9) is non-empty (Step 1). We then show that all the Markov matrices within the feasible set satisfy the given properties (Step 2).

Step 1: Let $\varepsilon_\alpha = \sqrt{\varepsilon_M}$. Let $\boldsymbol{\alpha}_k^j$ be a positive bounded column vector in $\mathbb{R}^{n_{\text{bin}}}$, with $\varepsilon_\alpha \leq \boldsymbol{\alpha}_k^j[i] \leq 1$ for all bins. We show that the following family of row stochastic Markov

matrices $\boldsymbol{Q}_k^j$ satisfy the constraints (CS1,2) and (LP1–4):

$$\boldsymbol{Q}_k^j[i,\ell] = \begin{cases} 0 & \text{if } \boldsymbol{A}_k^j[i,\ell] = 0 \\ \frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]\right) & \text{otherwise} \end{cases}, \qquad \forall i,\ell \in \{1,\ldots,n_{\text{bin}}\} \text{ and } i \neq \ell\,, \tag{2.10}$$

$$\boldsymbol{Q}_k^j[i,i] = \frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[i]\boldsymbol{\Theta}[\ell]\right) + \left(1 - \xi_k^j\boldsymbol{\alpha}_k^j[i]\right) + \sum_{\ell \in \left\{\boldsymbol{A}_k^j[i,\ell]=0\right\}} \frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]\right). \tag{2.11}$$

Clearly, the matrix $\boldsymbol{Q}_k^j$ satisfies (CS1) due to (2.10). The off-diagonal element $\boldsymbol{Q}_k^j[i,\ell] = 0$ if the element $\boldsymbol{\Theta}[\ell] = 0$, hence the matrix $\boldsymbol{Q}_k^j$ satisfies (CS2). If the element $\boldsymbol{\Theta}[\ell] > 0$, then the element $\boldsymbol{Q}_k^j[i,\ell]$ is upper bounded by $\frac{\xi_k^j}{\varepsilon_M}$ because $\boldsymbol{\alpha}_k^j[i] \leq 1$, $\boldsymbol{\Theta}[\ell] \leq 1$, and $\left(\frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\right) \leq \frac{\xi_k^j}{\varepsilon_\alpha} \leq \frac{\xi_k^j}{\varepsilon_M}$. The element $\boldsymbol{Q}_k^j[i,\ell]$ is lower bounded by $\varepsilon_M\xi_k^j\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ because $\boldsymbol{\alpha}_k^j[i] \geq \varepsilon_\alpha$, $\left(\frac{\xi_k^j}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\right) \geq \xi_k^j$ as $\boldsymbol{\Theta}\mathbf{1} = 1$, and $\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right) < 1$. This implies that the matrix $\boldsymbol{Q}_k^j$ also satisfies (LP4). The diagonal element $\boldsymbol{Q}_k^j[i,i]$ is lower bounded by $\left(1 - \xi_k^j\boldsymbol{\alpha}_k^j[i]\right) \geq (1 - \xi_k^j)$, which implies that the matrix $\boldsymbol{Q}_k^j$ satisfies (LP3).

We now show that matrix $\boldsymbol{Q}_k^j$ satisfies (LP1):

$$\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{Q}_k^j[i,\ell] = \frac{\xi_k^j\boldsymbol{\alpha}_k^j[i]}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell] + 1 - \xi_k^j\boldsymbol{\alpha}_k^j[i] = 1\,,$$

because $\sum_{\ell=1}^{n_{\text{bin}}}\left(\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]\right) = \boldsymbol{\Theta}\boldsymbol{\alpha}_k^j$. We now show that matrix $\boldsymbol{Q}_k^j$ satisfies (LP2):

$$\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i]\boldsymbol{Q}_k^j[i,\ell] = \frac{\xi_k^j\boldsymbol{\alpha}_k^j[\ell]\boldsymbol{\Theta}[\ell]}{\boldsymbol{\Theta}\boldsymbol{\alpha}_k^j}\sum_{i=1}^{n_{\text{bin}}}\left(\boldsymbol{\alpha}_k^j[i]\boldsymbol{\Theta}[i]\right) + \left(\boldsymbol{\Theta}[\ell] - \xi_k^j\boldsymbol{\Theta}[\ell]\boldsymbol{\alpha}_k^j[\ell]\right) = \boldsymbol{\Theta}[\ell]\,.$$

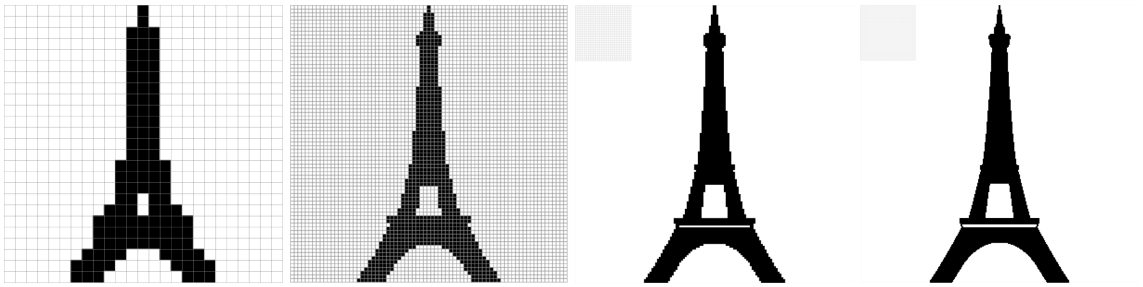
Thus, the family of Markov matrices $\boldsymbol{Q}_k^j$ are a subset of the feasible set, hence the feasible set is non-empty.

Step 2: The optimization problem in (2.9) is an LP because the constraints are all

linear inequalities or equalities and the objective function is linear. The resulting Markov matrix $\boldsymbol{M}_k^j$ is row stochastic (i.e., $\boldsymbol{M}_k^j \mathbf{1} = \mathbf{1}$) and has $\boldsymbol{\Theta}$ as its stationary distribution (i.e., $\boldsymbol{\Theta}\boldsymbol{M}_k^j = \boldsymbol{\Theta}$) because of (LP1) and (LP2) respectively. Moreover, (CS1,2) and (LP4) ensure that this Markov matrix only allows transitions into recurrent bins. ■

Remark 2.2. *(Properties of the Markov matrix $\boldsymbol{M}_k^j$ constructed using Theorem 2.1)* If $\xi_k^j < \xi_{\text{des}}$ in Remark 2.1, then the Markov matrix $\boldsymbol{M}_k^j$ is given by the identity matrix because the current swarm distribution is sufficiently close to the desired formation and the no further convergence is necessary.

Otherwise, as the swarm converges to the desired formation shape, the feedback gain ξ_k^j converges to 0 (i.e., $\xi_k^j \to 0$), and the Markov matrix $\boldsymbol{M}_k^j$ also converges to an identity matrix (i.e., $\boldsymbol{M}_k^j \to \mathbf{I}$) due to the constraints (LP3) and (LP4). The identity matrix ensures that agents settle down after the desired formation is achieved, thereby reducing unnecessary transitions. □



(a) $\boldsymbol{\nu}_{25}$ (25^2 bins) (b) $\boldsymbol{\nu}_{75}$ (75^2 bins) (c) $\boldsymbol{\nu}_{150}$ (150^2 bins) (d) $\boldsymbol{\nu}_{300}$ (300^2 bins)

Figure 2.4: Multiresolution images of the Eiffel Tower are shown, where (d) has the finest spatial resolution while (a) is coarsest. All the bins are shown in (a) and (b), whereas only a few bins are shown in the left-top corner in (c) and (d).

Remark 2.3. *(Computation Time)* If the Markov matrix is computed using the LP (2.9), then each agent has to solve the entire Markov matrix $\boldsymbol{M}_k^j$, even though it only needs the row of the Markov matrix corresponding to its present bin. The computation time for an LP increases with an increasing number of bins because the number of variables in $\boldsymbol{M}_k^j$ is approximately equal to n_{bin}^2. For example, Fig. 2.4 shows multiple images of the Eiffel Tower

with different spatial resolutions. If the desired formation shape is given by $\boldsymbol{\nu}_{25}$ or $\boldsymbol{\nu}_{75}$, then the computation time is a few minutes on a standard desktop computer. If the desired formation shape is given by $\boldsymbol{\nu}_{150}$ (with 5×10^8 variables) or $\boldsymbol{\nu}_{300}$ (with 8×10^9 variables), then the LP is impractical for real-time computation. Within the Eulerian framework, this escalating computation time with an increasing number of bins is also an issue for optimal transport based algorithms. □

Therefore, we need a faster method for generating the chain of inhomogeneous Markov matrices. The following corollary gives the analytical formula of the optimal Markov matrix when the cost matrix is symmetric.

Corollary 2.2. *If the cost matrix $\boldsymbol{C}_k$ is symmetric (i.e., $\boldsymbol{C}_k = \boldsymbol{C}_k^T$), then the Markov matrix $\boldsymbol{M}_k^j$ is constructed as follows:*

(1) If $\xi_k^j < \xi_{\text{des}}$, then set $\boldsymbol{M}_k^j = \mathbf{I}$.

(2) Otherwise, the optimal solution of the LP (2.9) that satisfies the constraints (CS1, 2) *from Theorem 2.1 is given by:*

$$\boldsymbol{M}_k^j[i,\ell] = \begin{cases} 0 & \text{if } \boldsymbol{A}_k^j[i,\ell] = 0 \\ \varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell] \left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max} + \varepsilon_C}\right) & \text{otherwise} \end{cases}, \qquad \forall i, \ell \in \{1, \ldots, n_{\text{bin}}\} \text{ and } i \neq \ell\,, \tag{2.12}$$

$$\boldsymbol{M}_k^j[i,i] = 1 - \sum_{\ell \in \{1,\ldots,n_{\text{bin}}\} \setminus \{i\}} \boldsymbol{M}_k^j[i,\ell]\,. \tag{2.13}$$

The transition probability $\boldsymbol{M}_k^j[i,\ell]$ from bin $B[i]$ to target bin $B[\ell]$ is directly proportional to both the HD (ξ_k^j) and $\boldsymbol{\Theta}[\ell]$. Moreover, target bins with lower cost of transitions have higher transition probabilities. In addition, $\boldsymbol{M}_k^j$ satisfies the properties of the Markov matrix in Theorem 2.1 and Remark 2.2.

Proof: According to Definition 2.3, $\boldsymbol{C}_k[i,i] = 0$ and $\boldsymbol{C}_k[i,\ell] > 0$ for all $i \neq \ell$. We first transform the LP (2.9) from Theorem 2.1 into a simpler form using the following substi-

tutions for all positive elements $\boldsymbol{Q}_k^j[i,i] = \boldsymbol{M}_k^j[i,i] - (1-\xi_k^j)$ and $\boldsymbol{Q}_k^j[i,\ell] = \boldsymbol{M}_k^j[i,\ell] - \varepsilon_M \xi_k^j \boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ and neglecting the constraints (LP1,2):

$$\min \sum_{i=1}^{n_{\text{bin}}}\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell]\boldsymbol{Q}_k^j[i,\ell] + \sum_{i=1}^{n_{\text{bin}}} \sum_{\ell\in\{\boldsymbol{A}_k^j[i,\ell]=1, i\neq\ell\}}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell]\varepsilon_M\xi_k^j\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right),$$

$$\text{subject to } 0 \le \boldsymbol{Q}_k^j[i,i] \le \xi_k^j,\ \forall i\,, \tag{$\widetilde{\text{LP3}}$}$$

$$0 \le \boldsymbol{Q}_k^j[i,\ell] \le \frac{\xi_k^j}{\varepsilon_M} - \varepsilon_M\xi_k^j\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right). \tag{$\widetilde{\text{LP4}}$}$$

The minimum cost of this transformed LP is obtained when $\sum_{i=1}^{n_{\text{bin}}}\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{C}_k[i,\ell]\boldsymbol{Q}_k^j[i,\ell] = 0$. Therefore, the minimum possible cost of the LP (2.9) is obtained when all positive off-diagonal elements $\boldsymbol{M}_k^j[i,\ell]$ are equal to their respective lower bounds $\varepsilon_M\xi_k^j\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$.

We now show that the Markov matrix $\boldsymbol{M}_k^j$ (2.12)-(2.13) satisfies all the constraints from Theorem 2.1. It follows from the construction of $\boldsymbol{M}_k^j$ that it satisfies (CS1,2) and (LP1,3,4). The diagonal elements of $\boldsymbol{M}_k^j$ are given by:

$$\boldsymbol{M}_k^j[i,i] = 1 - \sum_{\ell\in\{\boldsymbol{A}_k^j[i,\ell]=1, i\neq\ell\}} \varepsilon_M\xi_k^j\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right). \tag{2.14}$$

Note that $\boldsymbol{M}_k^j$ (2.12)-(2.13) is a reversible Markov matrix because of the symmetric cost matrix $\boldsymbol{c}_k = \boldsymbol{c}_k^T$, i.e., $\boldsymbol{\Theta}[\ell]\boldsymbol{M}_k^j[\ell,i] = \boldsymbol{\Theta}[i]\boldsymbol{M}_k^j[i,\ell] = \varepsilon_M\xi_k^j\boldsymbol{\Theta}[i]\boldsymbol{\Theta}[\ell]\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ for all $i \neq \ell$. This reversible property results in (LP2).

$$\begin{aligned}\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i]\boldsymbol{M}_k^j[i,\ell] &= \boldsymbol{\Theta}[\ell]\boldsymbol{M}_k^j[\ell,\ell] + \sum_{i\in\left\{\boldsymbol{A}_k^j[i,\ell]=1,\ell\neq i\right\}} \boldsymbol{\Theta}[i]\boldsymbol{M}_k^j[i,\ell]\,,\\ &= \boldsymbol{\Theta}[\ell]\boldsymbol{M}_k^j[\ell,\ell] + \sum_{i\in\left\{\boldsymbol{A}_k^j[i,\ell]=1,\ell\neq i\right\}} \boldsymbol{\Theta}[\ell]\boldsymbol{M}_k^j[\ell,i] = \boldsymbol{\Theta}[\ell]\,.\end{aligned}$$

Therefore, $\boldsymbol{M}_k^j$ (2.12)-(2.13) is the optimal solution of the LP (2.9). ■

If the cost matrix $\boldsymbol{C}_k$ is symmetric, then (2.12)-(2.13) give significant savings in compu-

tation time because each agent can directly compute the row of the Markov matrix that it needs. Moreover, the computation times for all four cases in Fig. 2.4 are less than 2 minutes on a standard desktop computer. In fact, if the desired formation is given by $\boldsymbol{\nu}_{300}$, then no LP-based approach is practical. On the other hand, we show later that the PSG–IMC algorithm using (2.12)-(2.13) achieves the desired formation $\boldsymbol{\nu}_{300}$.

Remark 2.4. *(Alternative functions for constraints)* Note that our construction technique holds even if the term $\left(1 - \frac{\boldsymbol{C}_k[i,\ell]}{C_{k,\max}+\varepsilon_C}\right)$ in the constraint (LP4) in Theorem 2.1 is replaced by any monotonic function in $(0, 1]$ that decreases with an increasing $\boldsymbol{C}_k[i, \ell]$. Similarly, the term ξ_k^j in the constraints (LP3,4) can be replaced by any monotonic function in $(0, 1]$ that decreases with a decreasing ξ_k^j (see Fig. 2.15 in Section 2.7.2). □

2.2.2 Construction of Fastest Mixing Inhomogeneous Markov Chain

In this section, we present an extension of our construction technique for constructing the fastest mixing IMC, where the IMC's convergence rate to the rank one matrix $\mathbf{1}\boldsymbol{\Theta}$ is optimized. The convergence rate of HMC, with time-invariant Markov matrix $\boldsymbol{M}$, is determined by the second largest eigenvalue modulus (i.e., $\max_{r\in\{2,\ldots,n_{\text{bin}}\}} |\lambda_r(\boldsymbol{M})|$) [138, 139]. On the other hand, the convergence rate of IMC is determined by the coefficient of ergodicity [135, pp. 137]. Since the first n_{rec} bins are recurrent bins, the Markov matrix $\boldsymbol{M}_k^j$ can be decomposed as follows:

$$\boldsymbol{M}_k^j = \begin{bmatrix} \boldsymbol{M}_{k,\text{sub}}^j & \mathbf{0}^{n_{\text{rec}}\times(n_{\text{bin}}-n_{\text{rec}})} \\ \boldsymbol{M}_k^j[n_{\text{rec}}+1:n_{\text{bin}}, 1:n_{\text{rec}}] & \boldsymbol{M}_k^j[n_{\text{rec}}+1:n_{\text{bin}}, n_{\text{rec}}+1:n_{\text{bin}}] \end{bmatrix}, \tag{2.15}$$

where the sub-matrix $\boldsymbol{M}_{k,\text{sub}}^j := \boldsymbol{M}_k^j[1:n_{\text{rec}}, 1:n_{\text{rec}}]$ encapsulates the bin transition probabilities between the recurrent bins.

Definition 2.7. [135, pp. 137–139] *(Coefficient of Ergodicity)* For the stochastic matrix

$\boldsymbol{M}^j_{k,\text{sub}}$, the coefficient of ergodicity $\tau_1(\boldsymbol{M}^j_{k,\text{sub}})$ is defined as:

$$\begin{aligned}\tau_1(\boldsymbol{M}^j_{k,\text{sub}}) &= \sup_{\boldsymbol{v}_1,\boldsymbol{v}_2,\boldsymbol{v}_1\neq\boldsymbol{v}_2} \frac{D_{\mathcal{L}_1}(\boldsymbol{v}_1\boldsymbol{M}^j_{k,\text{sub}}, \boldsymbol{v}_2\boldsymbol{M}^j_{k,\text{sub}})}{D_{\mathcal{L}_1}(\boldsymbol{v}_1,\boldsymbol{v}_2)}\,, \\ &= 1 - \min_{i,\ell}\sum_{s=1}^{n_{\text{rec}}} \min\left(\boldsymbol{M}^j_{k,\text{sub}}[i,s], \boldsymbol{M}^j_{k,\text{sub}}[\ell,s]\right)\,, \end{aligned} \tag{2.16}$$

where $\boldsymbol{v}_1$, $\boldsymbol{v}_2$ are probability row vectors in $\mathbb{R}^{n_{\text{rec}}}$ and $i,\ell,s \in \{1,\ldots,n_{\text{rec}}\}$. □

Let the sub-matrix $\boldsymbol{A}^j_{k,\text{sub}} := \boldsymbol{A}^j_k[1:n_{\text{rec}}, 1:n_{\text{rec}}]$ encapsulate the motion constraints between the recurrent bins. We define $n^j_{k,\text{dia}}$ as the graph diameter in the graph conforming to the matrix $\boldsymbol{A}^j_{k,\text{sub}}$, i.e., it is the greatest number of edges in a shortest path between any pair of recurrent bins [140]. If $n^j_{k,\text{dia}} > 2$, then there exists recurrent bins $B[i]$ and $B[\ell]$ such that either $\boldsymbol{M}^j_{k,\text{sub}}[i,s] = 0$ or $\boldsymbol{M}^j_{k,\text{sub}}[\ell,s] = 0$ for all $s \in \{1,\ldots,n_{\text{rec}}\}$, which implies that $\tau_1(\boldsymbol{M}^j_{k,\text{sub}}) = 1$. In order to avoid this trivial case, we minimize the coefficient of ergodicity of the positive matrix $(\boldsymbol{M}^j_{k,\text{sub}})^{n^j_{k,\text{dia}}}$ [141, Theorem 8.5.2, pp. 516].

The following corollary presents the construction of the Markov matrix that minimizes the coefficient of ergodicity of the IMC while retaining the original structure.

Corollary 2.3. *In order to construct the Markov matrix $\boldsymbol{M}^j_k$ that generates the fastest mixing IMC, the following convex optimization problem is used instead of the LP (2.9) in Theorem 2.1:*

$$\min \tau_1\left((\boldsymbol{M}^j_{k,\text{sub}})^{n^j_{k,\text{dia}}}\right)\,, \tag{2.17}$$

subject to (LP1 − 4) *in (2.9).*

Proof: It follows from (2.16) that the cost function $\tau_1\left((\boldsymbol{M}^j_{k,\text{sub}})^{n^j_{k,\text{dia}}}\right)$ is a convex function of the stochastic matrix $\boldsymbol{M}^j_{k,\text{sub}}$ because it can be expressed as [135, Lemma 4.3, pp. 139]:

$$\tau_1\left((\boldsymbol{M}^j_{k,\text{sub}})^{n^j_{k,\text{dia}}}\right) = \sup_{\|\boldsymbol{\delta}\|_2=1,\,\boldsymbol{\delta}\mathbf{1}=0} \left\|\boldsymbol{\delta}\cdot(\boldsymbol{M}^j_{k,\text{sub}})^{n^j_{k,\text{dia}}}\right\|_1\,,$$

where $\boldsymbol{\delta} = \text{const}(\boldsymbol{v}_1 - \boldsymbol{v}_2)$ is a row vector in $\mathbb{R}^{n_{\text{rec}}}$. Hence the nonlinear optimization problem (2.17) is a convex optimization problem. The family of Markov matrices $\boldsymbol{Q}_k^j$ (2.10)-(2.11) introduced in the proof of Theorem 2.1 is also within the feasible set for this problem. ■

Note that Corollary 2.3 also suffers the computation time issue discussed in Remark 2.3.

2.2.3 Condition for Escaping Transient Bins

In this section, we discuss some additional complexities that arise due to motion constraints and then state the condition for escaping transient bins.

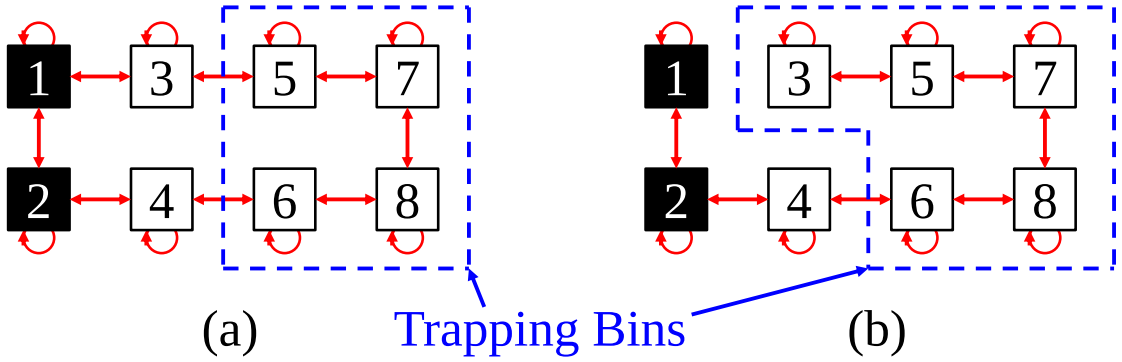


Figure 2.5: In this example, the bins 1 and 2 are recurrent bins. The allowed transitions (motion constraints) are shown in red. The trapping bins for the two cases are enclosed in blue.

Definition 2.8. *(Trapping Bins)* If an agent is inside a transient bin ($\boldsymbol{\Theta}[i] = 0$) and its motion constraints matrix $\boldsymbol{A}_k^j$ only allows transitions to other transient bins, then the agent's present bin is called a trapping bin. This agent is trapped in the trapping bin because the Markov matrix does not allow transitions out of this bin. For example, the trapping bins are shown in Fig. 2.5. Let $\boldsymbol{\mathcal{T}}_k^j$ represent the set of trapping bins for the j^{th} agent at the k^{th} time instant. □

In order to exit the set of trapping bins, for each bin $B[i] \in \boldsymbol{\mathcal{T}}_k^j$, we choose another transient bin $\Psi_k^j[i]$ such that the transition is allowed by motion constraints. The bin $\Psi_k^j[i]$ is either outside or close to other bins that are outside the set of trapping bins in the next

time instant. The j^{th} agent transitions from bin $B[i]$ to bin $\Psi_k^j[i]$ during the k^{th} time instant. This bin $\Psi_k^j[i]$ has to be chosen on a case-by-case basis depending on the motion constraints matrix $\boldsymbol{A}_k^j$. For example, in Fig. 2.5, for the trapping bin 5, the best option is bin 3 in case (a) and bin 7 in case (b). Using this condition, the agent exits the set of trapping bins because the graph conforming to the motion constraints matrix is strongly connected.

If an agent is in a transient bin, but not in a trapping bin, then it will eventually transition to a recurrent bin. We can also speed up this process. The matrix $\boldsymbol{S}_k^j \in \mathbb{R}^{n_{\text{bin}} \times n_{\text{bin}}}$ encapsulates the condition for escaping transient bins. If $\boldsymbol{\Theta}[i] = 0$ (i.e., $B[i]$ is a transient bin), then each element in the corresponding row $\boldsymbol{S}_k^j[i, 1 : n_{\text{bin}}]$ is given by:

$$\boldsymbol{S}_k^j[i,\ell] = \begin{cases} 1 & \textbf{if } B[i] \in \boldsymbol{\mathcal{T}}_k^j \text{ and } B[\ell] = \Psi_k^j[i] \\ \frac{1}{n_{k,i}^j} & \textbf{if } B[i] \notin \boldsymbol{\mathcal{T}}_k^j \text{ and } \boldsymbol{A}_k^j[i,\ell] = 1 \text{ and } \boldsymbol{\Theta}[\ell] > 0\,, \\ 0 & \textbf{otherwise} \end{cases} \tag{2.18}$$

where $n_{k,i}^j$ is the number of recurrent bins that the j^{th} agent can transition to, from bin $B[i]$ at the k^{th} time instant. This condition is used only when the agent is in a transient bin. In Section 2.3.1, we show that the agent exits the transient bins within finite time instants due to this condition.

2.3 Probabilistic Swarm Guidance using Inhomogeneous Markov Chains Algorithm for Shape Formation

In this section, we first state the PSG–IMC algorithm for shape formation and then present its convergence analysis and its property of robustness.

The pseudo-code for the PSG–IMC algorithm for shape formation is given in **Method 1**,

1: One iteration of j^{th} agent during k^{th} time instant,
where the j^{th} agent is in bin $B[i]$
2: Given $\boldsymbol{\Theta}$, $\boldsymbol{C}_k$, $\boldsymbol{A}_k^j$, and $\boldsymbol{\mu}_k^j$
3: **if** $\boldsymbol{\Theta}[i] = 0$, **then**
4: Compute $\boldsymbol{S}_k^j[i, 1 : n_{\text{bin}}]$ using (2.18)
5: Generate a random number $z \in \text{unif}[0; 1]$
6: Go to bin $B[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{S}_k^j[i, \ell] \leq z < \sum_{\ell=1}^{q} \boldsymbol{S}_k^j[i, \ell]$
7: **else**
8: Compute the feedback gain ξ_k^j using (2.5)
9: Compute $\boldsymbol{M}_k^j[i, 1 : n_{\text{bin}}]$ using Corollary 2.2 or
compute $\boldsymbol{M}_k^j$ using Theorem 2.1
10: Compute the term $\eta_{k,i}^j$ using (2.21)
11: Compute $\boldsymbol{P}_k^j[i, 1 : n_{\text{bin}}]$ using (2.19) and (2.20)
12: Generate a random number $z \in \text{unif}[0; 1]$
13: Go to bin $B[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{P}_k^j[i, \ell] \leq z < \sum_{\ell=1}^{q} \boldsymbol{P}_k^j[i, \ell]$
14: **end if**

Algorithm 2: PSG–IMC Algorithm for Shape Formation

whose key steps are shown in Fig. 2.2. At the start, the j^{th} agent knows the desired formation shape $\boldsymbol{\Theta}$, the time-varying cost matrix $\boldsymbol{C}_k$, and its time-varying motion constraints matrix $\boldsymbol{A}_k^j$. During each iteration, the agent determines the bin it belongs to (we assume that the j^{th} agent is in bin $B[i]$) and the current swarm distribution $\boldsymbol{\mu}_k^j$ (lines 1–2).

If the agent is in a transient bin (line 3), then it uses inverse transform sampling [51] to select the next bin from the corresponding row of the matrix $\boldsymbol{S}_k^j[i, 1 : n_{\text{bin}}]$ (lines 4–6).

Otherwise, the agent first computes the HD-based feedback gain ξ_k^j (line 8). If the cost matrix $\boldsymbol{C}_k$ is symmetric, then the agent can directly compute the row $\boldsymbol{M}_k^j[i, 1 : n_{\text{bin}}]$ using Corollary 2.2 (line 9). Otherwise, the agent can compute the entire Markov matrix $\boldsymbol{M}_k^j$ using Theorem 2.1 (line 9). In order to avoid undesirable transitions from bins that are deficient in agents (i.e., where $\boldsymbol{\Theta}[i] > \boldsymbol{\mu}_k^j[i]$), the agent modifies its Markov matrix row $\boldsymbol{M}_k^j[i, 1 : n_{\text{bin}}]$

as follows:

$$\boldsymbol{P}_k^j[i,\ell] = \left(1-\eta_{k,i}^j\right)\boldsymbol{M}_k^j[i,\ell]\,, \qquad \forall i \neq \ell \tag{2.19}$$

$$\boldsymbol{P}_k^j[i,i] = \left(1-\eta_{k,i}^j\right)\boldsymbol{M}_k^j[i,i] + \eta_{k,i}^j\,, \tag{2.20}$$

$$\text{where } \eta_{k,i}^j = \exp(-\tau^j k)\frac{\exp\left(\beta^j(\boldsymbol{\Theta}[i]-\boldsymbol{\mu}_k^j[i])\right)}{\exp\left(\beta^j|\boldsymbol{\Theta}[i]-\boldsymbol{\mu}_k^j[i]|\right)}\,, \tag{2.21}$$

and τ^j and β^j are time-invariant positive constants (lines 10–11). The term $\eta_{k,i}^j$ (2.21) is designed to reduce the transition probabilities from a bin that is deficient in agents and it decreases with increasing time instants. Finally, the agent uses inverse transform sampling [51] to select the next bin from the bin transition probabilities $\boldsymbol{P}_k^j[i,1:n_{\text{bin}}]$ (lines 12–13).

2.3.1 Main Result: Convergence Analysis

In this section, we discuss the convergence analysis of the PSG–IMC algorithm for shape formation given in Algorithm 2. The objective is to show that the swam distribution $\boldsymbol{\mu}_k^\star$ converges to the desired formation shape $\boldsymbol{\Theta}$ with acceptable convergence errors. Unlike the convergence proof for HMC, which is a direct application of the Perron–Frobenius theorem, the convergence proof for IMC is rather involved (e.g., see [135, 142]).

Assuming $\xi_{\text{des}} = \xi_{\min}$ from (2.7), if $\xi_k^j < \xi_{\min}$, then the current swarm distribution is very close to the desired formation and no further convergence is necessary. In the remainder of this section, we assume that the swarm has not converged and $\xi_k^j \geq \xi_{\min}$.

We first show that agents in recurrent bins transition using to the modified Markov matrix $\boldsymbol{P}_k^j$ derived from (2.19)-(2.20).

Theorem 2.4. *According to Algorithm 2, if an agent is in a recurrent bin, then it transitions using the following modified Markov matrix* $\boldsymbol{P}_k^j$*:*

$$\boldsymbol{P}_k^j = \left(\mathbf{I}-\boldsymbol{D}_k^j\right)\boldsymbol{M}_k^j + \boldsymbol{D}_k^j\,, \tag{2.22}$$

where $\boldsymbol{D}_k^j = diag\left(\eta_{k,1}^j, \ldots, \eta_{k,n_{\text{bin}}}^j\right)$. *The Markov matrix* $\boldsymbol{P}_k^j$ *is row stochastic (i.e.,* $\boldsymbol{P}_k^j \mathbf{1} = \mathbf{1}$*), asymptotically homogeneous with respect to* $\boldsymbol{\Theta}$ *(i.e.,* $\lim_{k\to\infty} \boldsymbol{\Theta}\boldsymbol{P}_k^j = \boldsymbol{\Theta}$*), and only allows transitions into recurrent bins.*

Proof: The modified Markov matrix $\boldsymbol{P}_k^j$ (2.22) is derived from (2.19)–(2.21). It follows from lines 3 and 6 in **Method 1** that the agent uses the Markov matrix $\boldsymbol{P}_k^j$ to transition if and only if it is in a recurrent bin (i.e., $\boldsymbol{\Theta}[i] > 0$).

The matrix $\boldsymbol{P}_k^j$ is row stochastic because $\boldsymbol{M}_k^j \mathbf{1} = \mathbf{1}$. The matrix $\boldsymbol{M}_k^j$ has $\boldsymbol{\Theta}$ as its stationary distribution for all $k \in \mathbb{N}$. It follows from the definition of the term $\eta_{k,i}^j$ (2.21) that $\lim_{k\to\infty} \boldsymbol{D}_k^j = \mathbf{0}^{n_{\text{bin}}\times n_{\text{bin}}}$. Therefore $\lim_{k\to\infty} \boldsymbol{P}_k^j = \lim_{k\to\infty} \boldsymbol{M}_k^j$. Hence, the sequence of matrices $\boldsymbol{P}_k^j$ is asymptotically homogeneous with respect to $\boldsymbol{\Theta}$ because $\lim_{k\to\infty} \boldsymbol{\Theta}\boldsymbol{P}_k^j = \boldsymbol{\Theta}$ (see Definition A.1 in Appendix A). Note that the Markov matrix $\boldsymbol{P}_k^j$ need not have $\boldsymbol{\Theta}$ as its stationary distribution for all time instants.

Finally, each element $\boldsymbol{P}_k^j[i,\ell] > 0$ if and only if the corresponding element $\boldsymbol{M}_k^j[i,\ell] > 0$ for all $i, \ell \in \{1, \ldots, n_{\text{bin}}\}$ and $k \in \mathbb{N}$. Therefore the matrix $\boldsymbol{P}_k^j$ only allows transitions into recurrent bins because the matrix $\boldsymbol{M}_k^j$ only allows transitions into recurrent bins. ■

We now show that all the agents leave the transient bins and enter the recurrent bins in finite time instants.

Theorem 2.5. *According to Algorithm 2, there exists a finite time instant* $T \leq (n_{\text{bin}} - n_{\text{rec}} + 1)$ *such that each agent is in a recurrent bin by the* T^{th} *time instant. If an agent is inside a recurrent bin, then it always remains within the set of recurrent bins.*

Proof: If an agent is in a recurrent bin, then it follows from Theorem 2.4 that it cannot transition to any transient bin.

If an agent is in a transient bin, but not a trapping bin, then the matrix $\boldsymbol{S}_k^j$ (2.18) ensures that the agent transitions to a recurrent bin in the next time instant.

If the agent is in a trapping bin, then the matrix $\boldsymbol{S}_k^j$ (2.18) ensures that the agent exits the set of trapping bins as soon as possible. Therefore, the maximum number of steps inside

the set of trapping bins is upper bounded by $(n_{\text{bin}} - n_{\text{rec}})$. Hence each agent enters a recurrent bin in at most $(n_{\text{bin}} - n_{\text{rec}} + 1)$ time instants. ■

Consider a probability (row) vector $\boldsymbol{x}_k^j \in \mathbb{R}^{n_{\text{bin}}}$, which denotes the probability mass function (pmf) of the predicted position of the j^{th} agent at the k^{th} time instant. Each element $\boldsymbol{x}_k^j[i]$ gives the probability of the event that the j^{th} agent is in bin $B[i]$ at the k^{th} time instant:

$$\boldsymbol{x}_k^j[i] = \mathbb{P}(\boldsymbol{r}_k^j[i] = 1), \quad \forall i \in \{1, \ldots, n_{\text{bin}}\}\,. \tag{2.23}$$

We now discuss convergence of each agent's pmf vector $\boldsymbol{x}_k^j$ to the desired formation $\boldsymbol{\Theta}$.

Theorem 2.6. *(Convergence of IMC) If each agent executes the PSG–IMC algorithm given in Algorithm 2, then each agent's time evolution of the pmf vector $\boldsymbol{x}_k^j$ converges pointwise to the desired stationary distribution $\boldsymbol{\Theta}$ irrespective of the initial condition, i.e., $\lim_{k\to\infty} \boldsymbol{x}_k^j = \boldsymbol{\Theta}$ pointwise for all agents.*

Proof: It follows from Theorem 2.5 that all agents are always in the set of recurrent bins from the T^{th} time instant onwards. Since the first n_{rec} bins are recurrent bins, we decompose the pmf vector $\boldsymbol{x}_k^j = [\bar{\boldsymbol{x}}_k^j, 0, \ldots, 0]$ for all $k \geq T$, where the probability row vector $\bar{\boldsymbol{x}}_k^j := [\boldsymbol{x}_k^j[1], \ldots, \boldsymbol{x}_k^j[n_{\text{rec}}]] \in \mathbb{R}^{n_{\text{rec}}}$ denotes the agent's pmf vector over the set of recurrent bins. Similarly, we decompose the desired formation as $\boldsymbol{\Theta} = [\bar{\boldsymbol{\Theta}}, 0, \ldots, 0]$, where $\bar{\boldsymbol{\Theta}} := [\boldsymbol{\Theta}[1], \ldots, \boldsymbol{\Theta}[n_{\text{rec}}]]$. Note that convergence of $\bar{\boldsymbol{x}}_k^j$ to $\bar{\boldsymbol{\Theta}}$, implies the convergences of $\boldsymbol{x}_k^j$ to $\boldsymbol{\Theta}$. The flowchart for this proof is given in Fig. 2.6.

According to **Method 1** and Theorem 2.4, the time evolution of the pmf vector $\bar{\boldsymbol{x}}_k^j$ is given by:

$$\bar{\boldsymbol{x}}_{k+1}^j = \bar{\boldsymbol{x}}_k^j \boldsymbol{P}_{k,\text{sub}}^j\,, \qquad \forall k \geq T\,, \tag{2.24}$$

where the sub-matrix $\boldsymbol{P}_{k,\text{sub}}^j := \boldsymbol{P}_k^j[1 : n_{\text{rec}}, 1 : n_{\text{rec}}]$ encapsulates the bin transition probabilities between the recurrent bins (similar to $\boldsymbol{M}_{k,\text{sub}}^j$ in (2.15)). It follows from Theorem 2.4

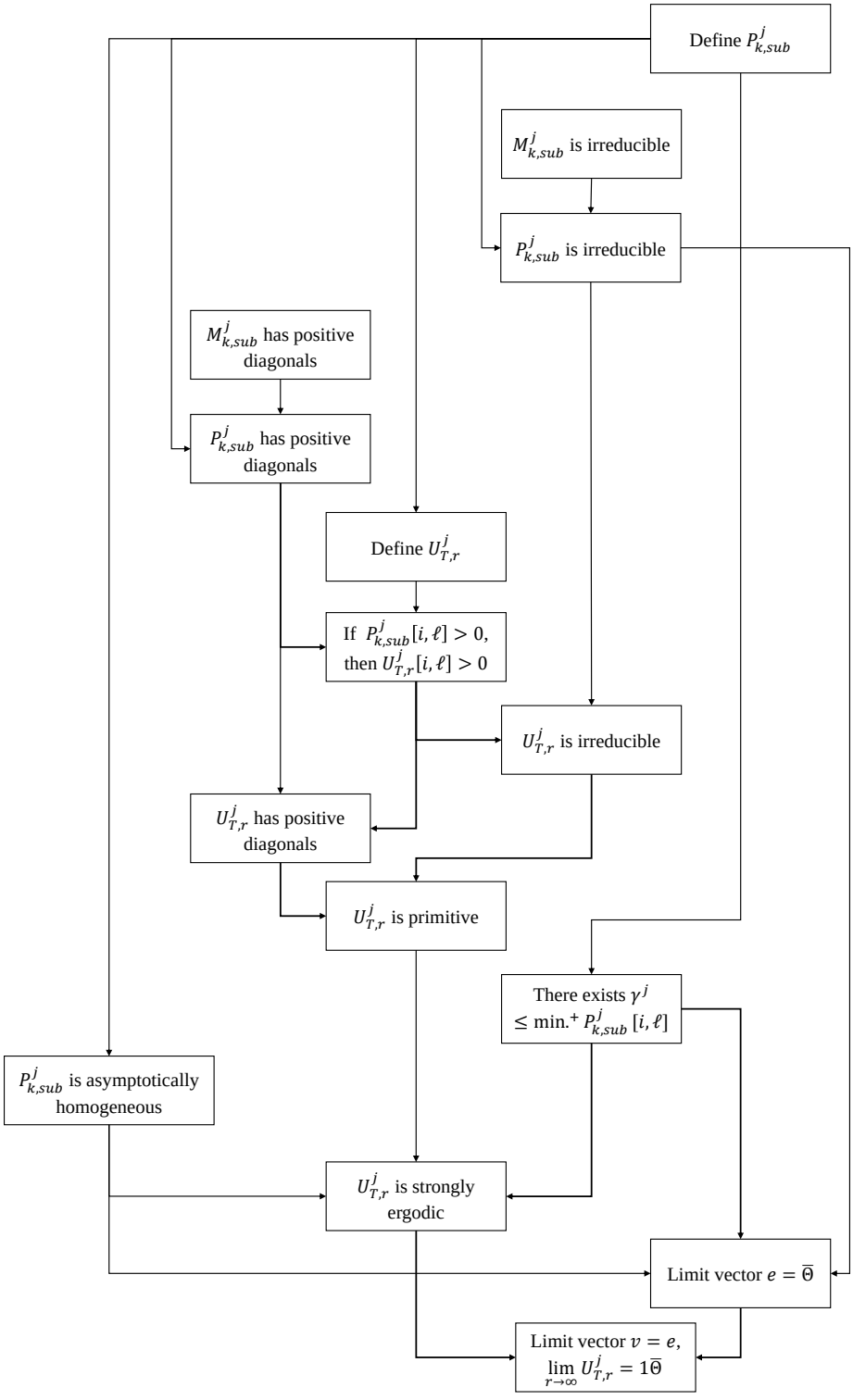


Figure 2.6: Flowchart for the proof of Theorem 2.6.

that the Markov matrix $\boldsymbol{P}^j_{k,\text{sub}}$ is row stochastic.

Each matrix $\boldsymbol{M}^j_{k,\text{sub}}$ is irreducible because the graph conforming to the corresponding motion constraints sub-matrix $\boldsymbol{A}^j_{k,\text{sub}}$ is strongly connected (due to property (iv) in Definition 2.5). Therefore, each matrix $\boldsymbol{P}^j_{k,\text{sub}}$ is irreducible because the term $\eta^j_{k,i} \leq \exp(-\tau^j T)$ for all $k \geq T$.

It follows from (2.5) that $D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}^j_k) = 1$ if and only if $\boldsymbol{\mu}^j_k[i] = 0$ and $\boldsymbol{\Theta}[i] > 0$ for all $i \in \{1, \ldots, n_{\text{rec}}\}$. Therefore, the feedback gain $\xi^j_k < 1$ for all time instant $k \geq T$ because all agents are in recurrent bins. Hence the diagonal element $\boldsymbol{M}^j_{k,\text{sub}}[i,i]$ for all $i \in \{1, \ldots, n_{\text{rec}}\}$ and $k \geq T$ is bounded by $0 < \boldsymbol{M}^j_{k,\text{sub}}[i,i] \leq 1$ (due to the constraint (LP3) in Theorem 2.1). Therefore, the diagonal element $\boldsymbol{P}^j_{k,\text{sub}}[i,i] > 0$ for all $i \in \{1, \ldots, n_{\text{rec}}\}$ and $k \geq T$.

The overall time evolution of the agent's pmf vector is given by the IMC for all $r > T$:

$$\bar{\boldsymbol{x}}^j_r = \bar{\boldsymbol{x}}^j_T \boldsymbol{P}^j_{T,\text{sub}} \boldsymbol{P}^j_{T+1,\text{sub}} \cdots \boldsymbol{P}^j_{r-1,\text{sub}} = \bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r}. \tag{2.25}$$

We now show that this forward matrix product $\boldsymbol{U}^j_{T,r}$ is strongly ergodic (see Definition A.2 in Appendix A) and $\bar{\boldsymbol{\Theta}}$ is its unique limit vector (i.e., $\lim_{r\to\infty} \boldsymbol{U}_{T,r} = \mathbf{1}\bar{\boldsymbol{\Theta}}$).

The matrix $\boldsymbol{U}^j_{T,r}$ is a product of nonnegative matrices, hence it is also a nonnegative matrix. If $\boldsymbol{P}^j_{k,\text{sub}}[i,\ell] > 0$ for some $k \in \{T, \ldots, r-1\}$ and $i, \ell \in \{1, \ldots, n_{\text{rec}}\}$, then the corresponding element $\boldsymbol{U}^j_{T,r}[i,\ell] > 0$ because:

$$\begin{aligned}
\boldsymbol{U}^j_{T,r}[i,\ell] \geq & \boldsymbol{P}^j_{T,\text{sub}}[i,\ell] \Big(\prod_{q=T+1}^{r-1} \boldsymbol{P}^j_{q,\text{sub}}[\ell,\ell] \Big) + \sum_{s=T+1}^{r-2} \left(\Big(\prod_{q=T}^{s-1} \boldsymbol{P}^j_{q,\text{sub}}[i,i] \Big) \boldsymbol{P}^j_{s,\text{sub}}[i,\ell] \Big(\prod_{q=s+1}^{r-1} \boldsymbol{P}^j_{q,\text{sub}}[\ell,\ell] \Big) \right) \\
& + \Big(\prod_{q=T}^{r-2} \boldsymbol{P}^j_{q,\text{sub}}[i,i] \Big) \boldsymbol{P}^j_{r-1,\text{sub}}[i,\ell] \,, \qquad \text{if } i \neq \ell \,, \qquad (2.26) \\
\boldsymbol{U}^j_{T,r}[i,i] \geq & \Big(\prod_{q=T}^{r-1} \boldsymbol{P}^j_{q,\text{sub}}[i,i] \Big) \,, \qquad \text{if } i = \ell \,. \qquad (2.27)
\end{aligned}$$

Since the matrix $\boldsymbol{P}^j_{k,\text{sub}}$ is irreducible and it follows from (2.26) that $\boldsymbol{U}^j_{T,r}[i,\ell] > 0$ if $\boldsymbol{P}^j_{k,\text{sub}}[i,\ell] > 0$, therefore the matrix $\boldsymbol{U}^j_{T,r}$ is also irreducible. Since the irreducible ma-

trix $\boldsymbol{U}^j_{T,r}$ has positive diagonal elements (2.27), it is a primitive matrix [141, Lemma 8.5.4, pp. 516].

A number of off-diagonal elements in $\boldsymbol{M}^j_{k,\mathrm{sub}}$ and $\boldsymbol{P}^j_{k,\mathrm{sub}}$ are zero because those transitions are not allowed (due to the constraints (CS1,2) in Theorem 2.1). The lower bound γ^j (independent of k) for the remaining positive elements in $\boldsymbol{P}^j_{k,\mathrm{sub}}$ is given by the constraint (LP4) in Theorem 2.1 and the upper bound of the term $\eta^j_{k,i}$ (2.21):

$$\gamma^j = \left(1 - \exp(-\tau^j T)\right) \xi_{\min} \varepsilon_M (\min{}^+ \boldsymbol{\Theta}) \left(1 - \tfrac{C_{\max}}{C_{\max}+\varepsilon_C}\right) \leq \min_{i,\,\ell}{}^+ \boldsymbol{P}^j_{k,\mathrm{sub}}[i,\ell]\,, \qquad k \geq T\,, \tag{2.28}$$

where $\min^+$ refers to the minimum of the positive elements and $C_{\max} = \max_{k\in\mathbb{N}} C_{k,\max}$. It follows from Theorem 2.4 that the sequence of matrices $\boldsymbol{P}^j_{k,\mathrm{sub}}$, $k \geq T$ is asymptotically homogeneous with respect to $\bar{\boldsymbol{\Theta}}$. Since (i) the forward matrix product $\boldsymbol{U}_{T,r}$ is primitive and (ii) there exists γ^j (independent of k), it follows from Theorem A.1 in Appendix A that the forward matrix product $\boldsymbol{U}^j_{T,r}$ is strongly ergodic. Since (i) the matrices $\boldsymbol{P}^j_{k,\mathrm{sub}}$, $k \geq T$ are irreducible and (ii) there exists γ^j (independent of k), it follows from Theorem A.2 in Appendix A that the limit vector $\boldsymbol{e} = \bar{\boldsymbol{\Theta}}$. Since (iii) $\boldsymbol{U}^j_{T,r}$ is strongly ergodic, it follows from Corollary A.3 in Appendix A that the unique limit vector is given by $\bar{\boldsymbol{\Theta}}$ (i.e., $\lim_{r\to\infty} \boldsymbol{U}^j_{T,r} = \mathbf{1}\bar{\boldsymbol{\Theta}}$). Hence, each agent's pmf vector converges to:

$$\lim_{r\to\infty} \bar{\boldsymbol{x}}^j_r = \lim_{r\to\infty} \bar{\boldsymbol{x}}^j_T \boldsymbol{U}^j_{T,r} = \bar{\boldsymbol{x}}^j_T \mathbf{1}\bar{\boldsymbol{\Theta}} = \bar{\boldsymbol{\Theta}}\,,$$

which implies that $\lim_{k\to\infty} \boldsymbol{x}^j_k = \boldsymbol{\Theta}$ pointwise for all agents. ■

Theorem 2.7. *For all $\varepsilon_{\lim} > 0$, there exists a $k^j_{\epsilon,\lim} \in \mathbb{N}$ such that the convergence error between the agent's pmf vector $\boldsymbol{x}^j_r$ and the desired formation $\boldsymbol{\Theta}$ is bounded for all $r \geq k^j_{\epsilon,\lim}$*

by:

$$D_{\mathcal{L}_1}(\boldsymbol{x}_r^j, \boldsymbol{\Theta}) \leq \varepsilon_{\lim} + D_{\mathcal{L}_1}(\boldsymbol{x}_T^j, \boldsymbol{\Theta}) \prod_{s=0}^{\left\lfloor \frac{r-T}{n_{\rm rec}-1} \right\rfloor - 1} \left(1 - n_{\rm rec} \left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta_q^j \right) \right),$$

where $\lfloor \cdot \rfloor$ *is the floor function and* $\delta_q^j = \min_{i,\,\ell} {}^{+}\boldsymbol{P}_{q,\rm sub}^j[i,\ell]$.

Proof: Since $\lim_{r\to\infty} \boldsymbol{U}_{T,r}^j = \mathbf{1}\bar{\boldsymbol{\Theta}}$, there exists a $k_{\epsilon,\lim}^j \in \mathbb{N}$ such that $D_{\mathcal{L}_1}(\bar{\boldsymbol{\Theta}}\boldsymbol{U}_{T,r}^j, \bar{\boldsymbol{\Theta}}) \leq \varepsilon_{\lim}$ for all $r \geq k_{\epsilon,\lim}^j$. It follows from the definition of $\tau_1(\boldsymbol{U}_{T,r}^j)$ in (2.16) that:

$$D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}_T^j \boldsymbol{U}_{T,r}^j, \bar{\boldsymbol{\Theta}}\boldsymbol{U}_{T,r}^j) \leq \tau_1(\boldsymbol{U}_{T,r}^j) D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}_T^j, \bar{\boldsymbol{\Theta}})\,.$$

Since $\bar{\boldsymbol{x}}_r^j = \bar{\boldsymbol{x}}_T^j \boldsymbol{U}_{T,r}^j$ (2.25), we get from the triangle inequality:

$$D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}_r^j, \bar{\boldsymbol{\Theta}}) \leq D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}_T^j \boldsymbol{U}_{T,r}^j, \bar{\boldsymbol{\Theta}}\boldsymbol{U}_{T,r}^j) + D_{\mathcal{L}_1}(\bar{\boldsymbol{\Theta}}\boldsymbol{U}_{T,r}^j, \bar{\boldsymbol{\Theta}}) \leq \tau_1(\boldsymbol{U}_{T,r}^j) D_{\mathcal{L}_1}(\bar{\boldsymbol{x}}_T^j, \bar{\boldsymbol{\Theta}}) + \varepsilon_{\lim}\,.$$

The sub-multiplicative property of $\tau_1(\boldsymbol{U}_{T,r}^j)$ gives:

$$\tau_1(\boldsymbol{U}_{T,r}^j) \leq \prod_{s=0}^{\left\lfloor \frac{r-T}{n_{\rm rec}-1} \right\rfloor - 1} \tau_1 \left(\boldsymbol{U}_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)}^j \right).$$

Here, if $r > T + \left\lfloor \frac{r-T}{n_{\rm rec}-1} \right\rfloor (n_{\rm rec}-1)$, then we neglect the contribution of $\tau_1 \left(\boldsymbol{U}^j_{T+\left\lfloor \frac{r-T}{n_{\rm rec}-1} \right\rfloor (n_{\rm rec}-1),r} \right)$.

The matrix $\boldsymbol{U}_{k,k+n_{\rm rec}-1}^j$, for any $k \geq T$, is a positive matrix because there exists a path of length $\leq (n_{\rm rec}-1)$ between every two recurrent bins (see Lemma A.4 in Appendix A). A conservative lower-bound on the elements in the positive matrix $\boldsymbol{U}^j_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)}$ is $\left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta_q^j \right)$. Therefore, it follows from (2.16) that $\tau_1 \left(\boldsymbol{U}^j_{T+s(n_{\rm rec}-1),T+(s+1)(n_{\rm rec}-1)} \right) \leq 1 - n_{\rm rec} \left(\prod_{q=T+s(n_{\rm rec}-1)}^{T+(s+1)(n_{\rm rec}-1)} \delta_q^j \right) < 1$. ■

We now focus on the convergence of the swarm distribution to the desired formation. In practical scenarios, the number of agents is finite, hence the following theorem gives a lower

bound on the number of agents.

Theorem 2.8. *Let $\varepsilon_{\text{lim}} > 0$, $\varepsilon_{\text{bin}} > 0$, and $\varepsilon_{\text{conv}} > 0$ represent convergence error thresholds. Let κ represent the latest time instant when an agent is added to or removed from the swarm. The convergence error between the swarm distribution $\boldsymbol{\mu}_k^\star$ and the desired formation $\boldsymbol{\Theta}$ is probabilistically bounded for all $k \geq k_{\epsilon,\text{lim}}$ by:*

$$\mathbb{P}\left(D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \geq \varepsilon_{\text{bin}} + \sigma_k\right) \leq \frac{n_{\text{rec}}}{4m_\kappa \varepsilon_{\text{bin}}^2}\,, \tag{2.29}$$

$$\mathbb{P}\left(D_H(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \geq \frac{1}{\sqrt{2}}\sqrt{\varepsilon_{\text{bin}} + \sigma_k}\right) \leq \frac{n_{\text{rec}}}{4m_\kappa \varepsilon_{\text{bin}}^2}\,, \tag{2.30}$$

where $m_k = m_\kappa$ for all $k \geq \kappa$, $\delta_q = \min_{j\in\{1,\ldots,m_\kappa\}} \delta_q^j$, and $\sigma_k = \varepsilon_{\text{lim}} + 2\prod_{s=0}^{\left\lfloor \frac{k-\kappa-T}{n_{\text{rec}}-1}\right\rfloor - 1}\left(1 - n_{\text{rec}}\left(\prod_{q=T+s(n_{\text{rec}}-1)}^{T+(s+1)(n_{\text{rec}}-1)} \delta_q\right)\right)$.

If the number of agents satisfies the inequality:

$$m_\kappa \geq \frac{n_{\text{rec}}}{16\xi_{\text{des}}^4 \varepsilon_{\text{conv}}}\,, \tag{2.31}$$

then the HD between the final swarm distribution and the desired formation is probabilistically bounded by $\varepsilon_{\text{conv}}$, i.e.,

$$\mathbb{P}\left(D_H\left(\lim_{k\to\infty} \boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}\right) \geq \xi_{\text{des}}\right) \leq \varepsilon_{\text{conv}}\,, \tag{2.32}$$

where ξ_{des} is defined in Remark 2.1.

Proof: Let $X_{k,i}^j$ denote the Bernoulli random variable, where $X_{k,i}^j = 1$ represents the event that the j^{th} agent is actually located in bin $B[i]$ at the k^{th} time instant (i.e., $\boldsymbol{r}_k^j[i] = 1$) and $X_{k,i}^j = 0$ otherwise (i.e., $\boldsymbol{r}_k^j[i] = 0$). We get from (2.23) that $\mathbb{P}(X_{k,i}^j = 1) = \boldsymbol{x}_k^j[i]$. Therefore $\mathbb{E}[X_{k,i}^j] = \boldsymbol{x}_k^j[i]$ and $\text{Var}(X_{k,i}^j) = \mathbb{E}[X_{k,i}^j](1-\mathbb{E}[X_{k,i}^j])$, where $\mathbb{E}[\cdot]$ and $\text{Var}(\cdot)$ respectively denote the expected value and the variance of the random variable.

Since $\lim_{k\to\infty} \boldsymbol{U}_{\kappa+T,k}^j = \mathbf{1}\bar{\boldsymbol{\Theta}}$ for all agents, there exists $k_{\epsilon,\text{lim}} \in \mathbb{N}$ such that $D_{\mathcal{L}_1}(\bar{\boldsymbol{\Theta}}\boldsymbol{U}_{\kappa+T,k}^j, \bar{\boldsymbol{\Theta}}) \leq$

ε_{lim} for all $k \geq k_{\epsilon,\text{lim}}$ and $j \in \{1, \ldots, m_\kappa\}$. It follows from Theorem 2.7 that for all $k \geq \kappa + T$:

$$D_{\mathcal{L}_1}(\boldsymbol{x}_k^j, \boldsymbol{\Theta}) \leq \left(1 - n_{\text{rec}}(\gamma^j)^{n_{\text{rec}}}\right)^{\left\lfloor \frac{k-\kappa-T}{n_{\text{rec}}} \right\rfloor} D_{\mathcal{L}_1}(\boldsymbol{x}_{\kappa+T}^j, \boldsymbol{\Theta}) \leq \sigma_k \,.$$

Therefore, $|\boldsymbol{x}_k^j[i] - \boldsymbol{\Theta}[i]| \leq \sigma_k$ and $|\mathbb{E}[X_{k,i}^j] - \boldsymbol{\Theta}[i]| \leq \sigma_k$ for all $i \in \{1, \ldots, n_{\text{rec}}\}$ and $k \geq k_{\epsilon,\text{lim}}$.

The swarm distribution in bin $B[i]$ at the k^{th} time instant is given by $\boldsymbol{\mu}_k^\star[i] = \frac{1}{m_\kappa} \sum_{j=1}^{m_\kappa} X_{k,i}^j$. Since these random variables are uncorrelated, we get $\mathbb{E}[\boldsymbol{\mu}_k^\star[i]] = \frac{1}{m_\kappa} \sum_{j=1}^{m_\kappa} \mathbb{E}[X_{k,i}^j]$, $|\mathbb{E}[\boldsymbol{\mu}_k^\star[i]] - \boldsymbol{\Theta}[i]| \leq \sigma_k$, and $\text{Var}(\boldsymbol{\mu}_k^\star[i]) = \frac{1}{m_\kappa}\mathbb{E}[X_{k,i}^j](1 - \mathbb{E}[X_{k,i}^j]) \leq \frac{1}{4m_\kappa}$. It follows from Chebychev's inequality (cf. [143, Theorem 1.6.4, pp. 25]) that for any ε_{bin}, the pointwise error probability for each bin is bounded by:

$$\mathbb{P}\left(|\boldsymbol{\mu}_k^\star[i] - \mathbb{E}[\boldsymbol{\mu}_k^\star[i]]| \geq \varepsilon_{\text{bin}}\right) \leq \frac{1}{4m_\kappa \varepsilon_{\text{bin}}^2} \,.$$

If follows from triangle inequality that:

$$|\boldsymbol{\mu}_k^\star[i] - \mathbb{E}[\boldsymbol{\mu}_k^\star[i]]| + |\mathbb{E}[\boldsymbol{\mu}_k^\star[i]] - \boldsymbol{\Theta}[i]| \geq |\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| \,,$$

$$|\boldsymbol{\mu}_k^\star[i] - \mathbb{E}[\boldsymbol{\mu}_k^\star[i]]| \geq |\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| - |\mathbb{E}[\boldsymbol{\mu}_k^\star[i]] - \boldsymbol{\Theta}[i]| \geq |\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| - \sigma_k \,.$$

Therefore:

$$\mathbb{P}\left(|\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| - \sigma_k \geq \varepsilon_{\text{bin}}\right) \leq \mathbb{P}\left(|\boldsymbol{\mu}_k^\star[i] - \mathbb{E}[\boldsymbol{\mu}_k^\star[i]]| \geq \varepsilon_{\text{bin}}\right)$$

$$\mathbb{P}\left(|\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| \geq \varepsilon_{\text{bin}} + \sigma_k\right) \leq \frac{1}{4m_\kappa \varepsilon_{\text{bin}}^2} \,.$$

The bound on $\mathcal{L}_1$ distance is obtained using Boole's inequality:

$$\mathbb{P}\left(D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \geq \varepsilon_{\text{bin}} + \sigma_k\right) \leq \sum_{i=1}^{n_{\text{rec}}} \mathbb{P}\left(|\boldsymbol{\mu}_k^\star[i] - \boldsymbol{\Theta}[i]| \geq \varepsilon_{\text{bin}} + \sigma_k\right) \leq \frac{n_{\text{rec}}}{4m_\kappa \varepsilon_{\text{bin}}^2} \,.$$

The bound on HD follows from (2.6), i.e., $D_H(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \leq \frac{1}{\sqrt{2}} D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta})^{\frac{1}{2}}$. Therefore:

$$\mathbb{P}\left(D_H(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}) \geq \frac{1}{\sqrt{2}}\sqrt{\varepsilon_{\text{bin}} + \sigma_k}\right) \leq \mathbb{P}\left(\frac{1}{\sqrt{2}} D_{\mathcal{L}_1}(\boldsymbol{\mu}_k^\star, \boldsymbol{\Theta})^{\frac{1}{2}} \geq \frac{1}{\sqrt{2}}\sqrt{\varepsilon_{\text{bin}} + \sigma_k}\right) \leq \frac{n_{\text{rec}}}{4 m_\kappa \varepsilon_{\text{bin}}^2}\,.$$

Since $\lim_{k\to\infty} \boldsymbol{x}_k^j = \boldsymbol{\Theta}$, we get $\lim_{k\to\infty} \mathbb{E}[X_{k,i}^j] = \boldsymbol{\Theta}[i]$ and $\lim_{k\to\infty} \sigma_k = 0$. Setting $\varepsilon_{\text{bin}} = 2\xi_{\text{des}}^2$, we get:

$$\mathbb{P}\left(D_H\left(\lim_{k\to\infty} \boldsymbol{\mu}_k^\star, \boldsymbol{\Theta}\right) \geq \xi_{\text{des}}\right) \leq \frac{n_{\text{rec}}}{16 m_\kappa \xi_{\text{des}}^4}\,.$$

The lower bound on the number of agents is given by $\frac{n_{\text{rec}}}{16 m_\kappa \xi_{\text{des}}^4} \leq \varepsilon_{\text{conv}}$. ■

Remark 2.5. It follows from Theorem 2.8 and the weak law of large numbers [144, pp. 86] that the final swarm distribution $\lim_{k\to\infty} \boldsymbol{\mu}_k^\star$ converges in probability to the desired formation $\boldsymbol{\Theta}$ as the number of agents m_κ tends to infinity. □

Thus, we have proved the convergence of the PSG–IMC algorithm for shape formation. We now discuss its property of robustness.

Remark 2.6. *(Robustness of the PSG–IMC Algorithm)* The PSG–IMC algorithm satisfies the Markov property because the action of each agent depends only on its present bin location and the current swarm distribution. This memoryless property ensures that all the agents re-start their guidance trajectory from their present bin location during every time instant. Thus, the swarm continues to achieve its objective even if agents are added or removed from the swarm or the swarm is damaged by external disturbances or some agents have not reached their target bin during the previous time instant.

Secondly, although the PSG–IMC algorithm uses the feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$, it can tolerate large estimation errors. If the j^{th} agents estimate of the current swarm distribution $\boldsymbol{\mu}_k^j$ has an estimation error ϵ_{est} (2.4), then the distance between

the feedback gains $\xi_k^j = D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^j)$ and $\xi_k^\star = D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}_k^\star)$ is bounded by [134]:

$$\left|\xi_k^j - \xi_k^\star\right| \leq D_H(\boldsymbol{\mu}_k^\star, \boldsymbol{\mu}_k^j) \leq \frac{1}{\sqrt{2}} \epsilon_{\text{est}}^{\frac{1}{2}} . \tag{2.33}$$

Even though ξ_k^j might differ from $\xi_k^\star$ substantially, the resulting Markov matrix $\boldsymbol{M}_k^j$ constructed using ξ_k^j still has $\boldsymbol{\Theta}$ as it stationary distribution. Therefore the agent's pmf vector $\boldsymbol{x}_k^j$ still converges to $\boldsymbol{\Theta}$, and consequently the swam distribution also converges to $\boldsymbol{\Theta}$. Hence the PSG–IMC algorithm can tolerate large errors in the feedback of the current swarm distribution. □

We now present some extensions of the PSG–IMC algorithm.

2.3.2 Multiresolution PSG–IMC Algorithm for Shape Formation

Here the agents take advantage of the multiresolution representation of the desired formation. Such a multiresolution hierarchical approach to guidance or path planning has been used in the Lagrangian framework [145, 146]. Here we present a multiresolution guidance strategy for the Eulerian framework.

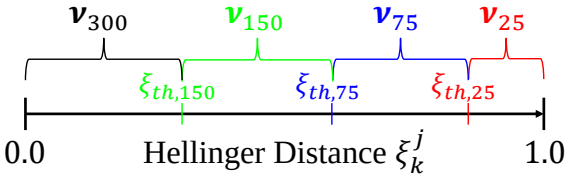


Figure 2.7: This image shows the different thresholds and the corresponding resolution of the desired formation that should be used when ξ_k^j is within those thresholds.

If multiresolution images of the desired formation are available, then the agents can choose the appropriate resolution depending on the current swarm distribution. For example, coarser resolution images ($\boldsymbol{\nu}_{150}$, $\boldsymbol{\nu}_{75}$ and $\boldsymbol{\nu}_{25}$) of the desired formation of the Eiffel Tower $\boldsymbol{\nu}_{300}$ are shown in Fig. 2.4. We select thresholds $\xi_{\text{th},150}$, $\xi_{\text{th},75}$, and $\xi_{\text{th},25}$ so that the agent can use the appropriate resolution of the desired formation when the feedback gain ξ_k^j is within these thresholds as shown in Fig. 2.7. The thresholds are selected such that they are achievable

even in the presence of estimation errors. The main advantages of this multiresolution approach include computational efficiency and simplicity of implementation.

2.3.3 Time-Varying Physical Space of the Swarm

The compact physical space over which the swarm is distributed need not be time-invariant in the global reference frame. The local reference frame of the swarm can follow a pre-defined trajectory in the global reference frame (e.g., an orbit in space or a trajectory in the sea) and the time-varying position of each bin can be easily computed from this known trajectory. Consequently, all the PSG–IMC algorithms discussed in this chapter are also applicable in this scenario.

2.4 Probabilistic Swarm Guidance using Inhomogeneous Markov Chains Algorithm for Area Exploration

In this section, we present an extension of the PSG–IMC algorithm for area exploration in which a swarm of distributed agents are driven to match the unknown target distribution of some physical or artificial phenomena (e.g., oil spill). This problem is commonly called *goal searching* [20].

Definition 2.9. *(Unknown Target Distribution $\mathbf{\Omega}$)* The unknown target distribution $\mathbf{\Omega}$ is a probability (row) vector in $\mathbb{R}^{n_{\text{bin}}}$, where each element $\mathbf{\Omega}[i]$ represents the target distribution in the corresponding bin $B[i]$. Each agent can measure the target distribution in its present bin. □

Each agent independently determines its bin-to-bin trajectory using the PSG–IMC algorithm for area exploration so that the overall swarm converges to this unknown target distribution $\mathbf{\Omega}$. The key idea of this algorithm is that the waiting time in a bin is directly

proportional to the target distribution in that bin. Moreover, the algorithm automatically adapts to changes in the target distribution.

1: Lines 1–2 in Algorithm 2
2: Measure target distribution in present bin $\mathbf{\Omega}[i]$
3: **if** $k - k_0 < \tau_c \mathbf{\Omega}[i]$, **then**
4: Wait in bin $B[i]$
5: **else** Set $\mathbf{\Theta} = \frac{1}{n_{\text{bin}}}\mathbf{1}^T$ and $\xi_k^j = \xi^j$
6: Compute the term $\eta_{k,i}^j$ using (2.34)
7: Lines 9, 11–13 in Algorithm 2, Set $k_0 = k$
8: **end if**

Algorithm 3: PSG–IMC Algorithm for Area Exploration

The pseudo-code for this PSG–IMC algorithm for area exploration is given in Algorithm 3. The j^{th} agent first measures the target distribution in its present bin $\mathbf{\Omega}[i]$ (line 2). The waiting time in bin $B[i]$ is directly proportional to $\mathbf{\Omega}[i]$, where τ_c is the constant of proportionality. The agent checks if it has spent enough time instants in bin $B[i]$ (line 3), where k_0 is is set in line 7. When the algorithm starts (i.e. $k = 1$), we set $k_0 = 1$. If the agent has not spent enough time instants in bin $B[i]$, then it continues to wait in bin $B[i]$ (line 4).

If the agent has spent enough time instants in bin $B[i]$, then it sets the desired formation $\mathbf{\Theta}$ to $\frac{1}{n_{\text{bin}}}\mathbf{1}^T$ because it wants to explore all the bins uniformly (line 5). The agent sets the feedback gain ξ_k^j to some positive constant $\xi^j \in (0, 1)$ because the it does not know the entire target distribution $\mathbf{\Omega}$ (line 5). In order to avoid undesirable transitions, the agent computes the term $\eta_{k,i}^j$ as follows (line 6):

$$\eta_{k,i}^j = \exp(-\tau^j k)\frac{\exp\left(\beta^j(\mathbf{\Omega}[i] - \boldsymbol{\mu}_k^j[i])\right)}{\exp\left(\beta^j|\mathbf{\Omega}[i] - \boldsymbol{\mu}_k^j[i]|\right)}\,. \tag{2.34}$$

Then the agent computes the transition probabilities $\boldsymbol{P}_k^j[i, 1 : n_{\text{bin}}]$ using lines 9 and 11 from Algorithm 2 and selects the next bin using lines 12–13 from Algorithm 2 (line 7). Finally, the agent sets k_0 equal to the current time instant k (line 7). We now discuss the convergence

analysis of this algorithm.

Theorem 2.9. *(Convergence of IMC) If each agent executes the PSG–IMC algorithm given in Algorithm 3, then each agent's time evolution of the pmf vector* $\boldsymbol{x}_k^j$ *converges pointwise to the unknown target distribution* $\boldsymbol{\Omega}$ *irrespective of the initial condition, i.e.,* $\lim_{k\to\infty} \boldsymbol{x}_k^j = \boldsymbol{\Omega}$ *pointwise for all agents.*

Proof: Here, all bins are recurrent bins. Since $\boldsymbol{\Theta} = \frac{1}{n_{\text{bin}}}\mathbf{1}^T$, it follows from Theorem 2.6 that as $k \to \infty$, an agent is equally likely to transition to any bin $B[i]$. But the waiting time in bin $B[i]$ is directly proportional to the target distribution in that bin $\boldsymbol{\Omega}[i]$, due to lines 3–4 in Algorithm 3. Therefore, $\lim_{k\to\infty} \mathbb{E}[X_{k,i}^j] \propto \frac{\boldsymbol{\Omega}[i]}{n_{\text{bin}}}$. Hence, $\lim_{k\to\infty} \boldsymbol{x}_k^j = \boldsymbol{\Omega}$ pointwise for all agents. ■

The remaining convergence analysis straightforwardly follows that of the previous algorithm given in Section 2.3.1.

2.5 Probabilistic Swarm Guidance using Optimal Transport Algorithm for Shape Formation

The key step sin the proposed PSG–OT algorithm for shape formation are shown in Fig. 2.8 and its pseudo code is given in Algorithm 4. In Step 1, each agent determines its present location and bin. During Step 2, each agent estimates the current swarm distribution. Step 3 involves solving the OTP using LP to obtain the bin transition probabilities for the current time step. The target bin is selected by inverse transform sampling the OTP solution, as shown in Step 4. Step 5 involves checking if the proposed target bin satisfies motion constraints. Finally, in Step 6, the agent travels to the target bin while avoiding collisions.

We first state the OTP for transporting the swarm from its estimated current distribution $\boldsymbol{\mu}_k^j$ to the desired formation $\boldsymbol{\Theta}$ while minimizing the cost function $\boldsymbol{C}_k$ and satisfying motion

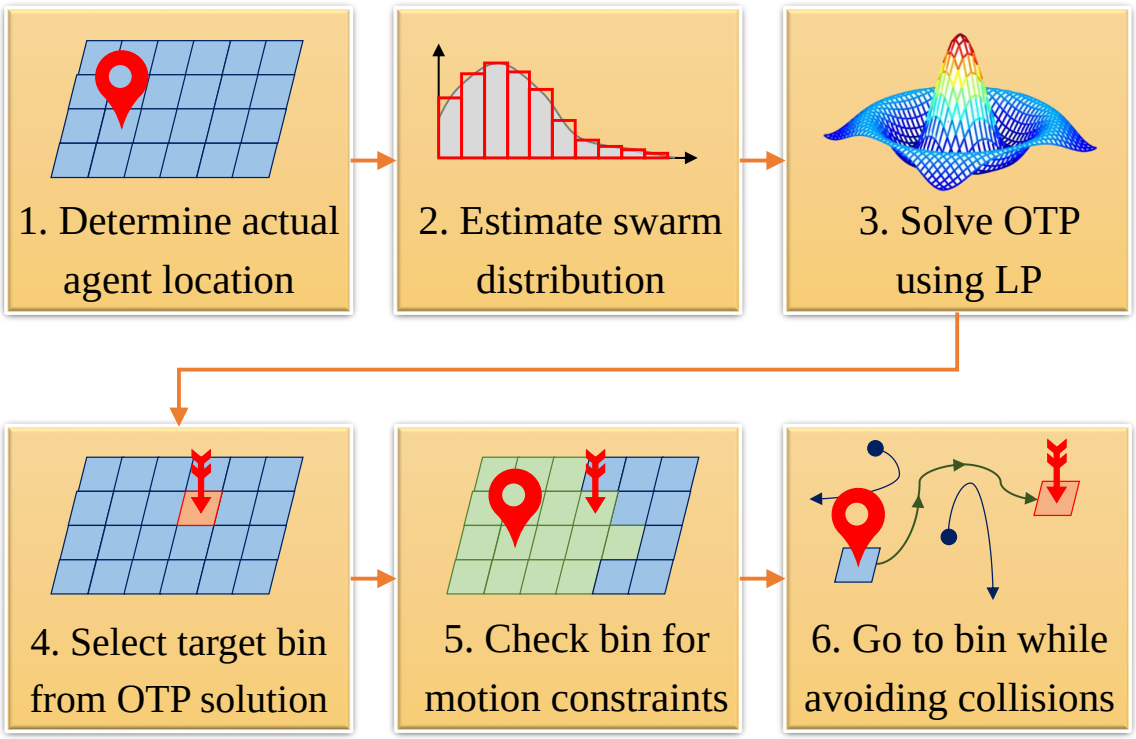


Figure 2.8: Flowchart of PSG–OT describing the key steps for a single agent in a single time step.

constraints. The n_x-dimensional compact convex Euclidean space over which the swarm is distributed is denoted by $\mathcal{B} \subset \mathbb{R}^{n_x}$. Let $\boldsymbol{\kappa}[i] \in \mathbb{R}^{n_x}$ represent the centroid of the convex bin $B[i]$. The Dirac measure on $\boldsymbol{x} \in \mathbb{R}^{n_x}$ is defined as $\delta_{\boldsymbol{x}}(\boldsymbol{\kappa}[i]) = 1$ if $\boldsymbol{x} = \boldsymbol{\kappa}[i]$ and 0 otherwise. Then the measure induced by the j^{th} agent's estimated current swarm distribution $\boldsymbol{\mu}_k^j$ on $\mathcal{B}$ is given by:

$$\rho_k^j(\boldsymbol{x}) = \sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\mu}_k^j[i] \delta_{\boldsymbol{x}}(\boldsymbol{\kappa}[i])\,. \tag{2.35}$$

Similarly, for $\boldsymbol{y} \in \mathbb{R}^{n_x}$, the measure induced by the desired formation $\boldsymbol{\Theta}$ on $\mathcal{B}$ is given by:

$$\sigma_k(\boldsymbol{y}) = \sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\Theta}[i] \delta_{\boldsymbol{y}}(\boldsymbol{\kappa}[i])\,. \tag{2.36}$$

The OTP is represented as the Monge–Kantorovich minimization problem [54]:

$$\boldsymbol{\gamma}_k^j(\boldsymbol{x},\boldsymbol{y}) = \arg \inf_{\varpi(\boldsymbol{x},\boldsymbol{y})} \int_{\mathcal{B}\times\mathcal{B}} \boldsymbol{C}_k(\boldsymbol{x},\boldsymbol{y}) d\varpi(\boldsymbol{x},\boldsymbol{y}) \,, \tag{2.37}$$

where the infimum runs over all joint probability measures $\varpi(\boldsymbol{x},\boldsymbol{y})$ on $\mathcal{B}\times\mathcal{B}$, which have $\rho_k^j(\boldsymbol{x})$ and $\sigma_k(\boldsymbol{y})$ as their marginal measures for $\boldsymbol{x}\in\mathcal{B}$ and $\boldsymbol{y}\in\mathcal{B}$ respectively. The joint measures or couplings that achieve the optimal cost are called optimum transference plans $\boldsymbol{\gamma}_k^j(\boldsymbol{x},\boldsymbol{y})$. The following theorem gives conditions for the existence of optimum transference plans.

Theorem 2.10. *[54, Theorem 4.1, p. 43] (Existence of an optimal coupling) Let $\boldsymbol{a}(\boldsymbol{x}) \in L_1(\rho_k^j)$, $\boldsymbol{b}(\boldsymbol{y}) \in L_1(\sigma_k)$ be two upper semi-continuous functions and $\boldsymbol{C}_k(\boldsymbol{x},\boldsymbol{y})$ be a lower semi-continuous cost function, such that $\boldsymbol{C}_k(\boldsymbol{x},\boldsymbol{y}) \geq \boldsymbol{a}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{y}), \forall \boldsymbol{x},\boldsymbol{y}$. Then there exists an optimal coupling $\boldsymbol{\gamma}_k^j(\boldsymbol{x},\boldsymbol{y})$ which minimizes the total cost $\int_{\mathcal{B}\times\mathcal{B}} \boldsymbol{C}_k(\boldsymbol{x},\boldsymbol{y}) d\varpi(\boldsymbol{x},\boldsymbol{y})$ among all possible couplings $\varpi(\boldsymbol{x},\boldsymbol{y})$ on $\mathcal{B}\times\mathcal{B}$ with marginals $\rho_k^j(\boldsymbol{x})$ and $\sigma_k(\boldsymbol{y})$.*

Two distinct optimal transference trajectories in the optimum transference plan $\boldsymbol{\gamma}_k^j$ do not intersect, except at endpoints [54]. Thus the OTP solution also helps in generating collision-free trajectories for the agents.

2.5.1 Incorporation of Motion Constraints

In this section, we capture the effect of the motion constraints $\boldsymbol{A}_k^j$ (defined in Definition 2.5) by modifying the cost function $\boldsymbol{C}_k$ (defined in Definition 2.3). Let us choose a positive constant $C_{\max}$ such that $C_{\max} \gg \boldsymbol{C}_k[i,\ell]$ for all bins $i,\ell \in \{1,\ldots,n_{\text{bin}}\}$ and for all time instants $k \in \mathbb{N}$. The modified cost function $\tilde{\boldsymbol{C}}_k^j$ that captures the effect of the motion

constraints is given by:

$$\tilde{\boldsymbol{C}}_k^j[i,\ell] = \begin{cases} \boldsymbol{C}_k[i,\ell] & \text{if } \boldsymbol{A}_k^j[i,\ell] = 1 \\ C_{\max} & \text{if } \boldsymbol{A}_k^j[i,\ell] = 0 \end{cases}, \quad \forall i,\ell \in \{1,\ldots,n_{\text{bin}}\},\, \forall k \in \mathbb{N}\,. \tag{2.38}$$

This modified cost function $\tilde{\boldsymbol{C}}_k^j$ is used in the OTP (2.37).

2.5.2 Linear Programming Formulation

Let us now state the discrete OTP that is solved by each agent in the swarm. The discrete OTP can be recast as a LP, where $\boldsymbol{\gamma}_k^j \in \mathbb{R}^{n_{\text{bin}} \times n_{\text{bin}}}$ is the optimum transference plan:

$$\begin{aligned} &\min \sum_{i=1}^{n_{\text{bin}}} \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{\gamma}_k^j[i,\ell] \tilde{\boldsymbol{C}}_k^j[i,\ell]\,, \\ \text{subject to } &\sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{\gamma}_k^j[i,\ell] = \boldsymbol{\mu}_k^j[i], \quad \forall i \in \{1,\ldots,n_{\text{bin}}\}\,, \\ &\sum_{i=1}^{n_{\text{bin}}} \boldsymbol{\gamma}_k^j[i,\ell] = \boldsymbol{\Theta}[\ell], \quad \forall \ell \in \{1,\ldots,n_{\text{bin}}\}\,, \\ &\boldsymbol{\gamma}_k^j[i,\ell] \geq 0, \quad \forall i,\ell \in \{1,\ldots,n_{\text{bin}}\}\,. \end{aligned} \tag{2.39}$$

Each agent independently solves this LP to find the optimum transference plan $\boldsymbol{\gamma}_k^j$ from the initial distribution $\boldsymbol{\mu}_k^j$ to the final desired distribution $\boldsymbol{\Theta}$.

The optimal transference plan $\boldsymbol{\gamma}_k^j$, can also be interpreted in the Markovian sense. For a non-empty bin $B[i]$, the probability of the j^{th} agent transitioning from bin $B[i]$ to bin $B[\ell]$ during the k^{th} time step is given by:

$$\mathbb{P}\left(\boldsymbol{r}_{k+1}^j[\ell] = 1 | \boldsymbol{r}_k^j[i] = 1\right) = \frac{\boldsymbol{\gamma}_k^j[i,\ell]}{\sum_{q=1}^{n_{\text{bin}}} \boldsymbol{\gamma}_k^j[i,q]}\,. \tag{2.40}$$

Inverse transform sampling is used to select the target bin from these bin transition probabilities [51].

Note that the Markov matrix, representing the bin transition probabilities of the j^{th} agent at the k^{th} time step given by (2.40), does not have $\boldsymbol{\Theta}$ as its stationary distribution. Hence this approach is dissimilar from the Markov chain based approaches. But it is due to this Markovian interpretation that PSG–OT is also robust to disturbances or damages to the formation.

2.5.3 Check Target Bins for Motion Constraints

In some cases, it is possible that the target bin, selected from the OTP solution $\boldsymbol{\gamma}_k^j$, does not satisfy motion constraints $\boldsymbol{A}_k^j$. Here, we address this particular situation.

If the agent is in a trapping bin and the selected target bin does not satisfy motion constraints, then the agent transitions according to the transient bin escape condition $\boldsymbol{S}_k^j$ discussed in Section 2.2.3.

If the agent is in a trapping bin and the selected target bin does not satisfy motion constraints, then the agent continues to remain in its present bin. This can also be encapsulated into the matrix $\boldsymbol{S}_k^j$ as:

$$\boldsymbol{S}_k^j[1 : n_{\text{rec}}, 1 : n_{\text{rec}}] = \mathbf{I}\,, \tag{2.41}$$

because the first n_{rec} bins are recurrent bins.

The pseudo code of PSG–OT algorithm for shape formation is shown in Algorithm 4. Each agent determines its current location and current swarm distribution. It determines its next location by solving the OTP and inverse transform sampling. Finally, the agent checks if its target bin satisfies motion constraints and uses the transient bin escape condition.

One iteration of j^{th} agent during k^{th} time instant,
where the j^{th} agent is in bin $B[i]$
Given $\boldsymbol{\Theta}$, $\boldsymbol{C}_k$, $\boldsymbol{A}_k^j$, and $\boldsymbol{\mu}_k^j$
Compute modified cost function $\tilde{\boldsymbol{C}}_k^j$ using (2.38)
Compute optimum transference plan $\boldsymbol{\gamma}_k^j$ using OTP (2.39)
Generate a random number $z \in \text{unif}\left[0; \sum_{\ell=1}^{n_{\text{bin}}} \boldsymbol{\gamma}_k^j[i,\ell]\right]$
Choose bin $B[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{\gamma}_k^j[i,\ell] \leq z < \sum_{\ell=1}^{q} \boldsymbol{\gamma}_k^j[i,\ell]$
if $\boldsymbol{A}_k^j[i,q] = 1$ **then**
    Go to bin $B[q]$
else
    Compute $\boldsymbol{S}_k^j[i, 1:n_{\text{bin}}]$ using (2.18) and (2.41)
    Generate a random number $z \in \text{unif}[0;1]$
    Go to bin $B[q]$ such that $\sum_{\ell=1}^{q-1} \boldsymbol{S}_k^j[i,\ell] \leq z < \sum_{\ell=1}^{q} \boldsymbol{S}_k^j[i,\ell]$
end if

Algorithm 4: PSG–OT Algorithm for Shape Formation

2.6 Collision-free Trajectory Generation using Voronoi Partitions

A collision avoidance algorithms are necessary for determining the collision-free guidance trajectory of each agent from its current location to the target bin. It is desired that all agents should always maintain a minimum collision avoidance distance between each other, which is denoted by r_{col}. In this section, we present a new distributed collision-free trajectory generation algorithm specifically suited for probabilistic swarm guidance and its bin-based architecture by modifying the coverage control algorithm.

The coverage control algorithm, which uses the Voronoi partition based Lloyd's descent algorithm [28], is designed for optimally distributing multiple agents over a given region. This distribution is optimal with respect to a ℓ_2-norm based cost function. An interesting property of these Voronoi partitions is that, if the given region is a convex polytope in an n_x-dimensional Euclidean space, then each Voronoi partition of the given region is itself a n_x-dimensional convex polytope [147].

In the PSG–IMC and PSG–OT algorithms, at every time step, the agents in a particular bin either remain in the same bin or transition to their respective target bins. Let us call the agents that remain in the same bin as stationary agents and those that transition to their target bins as transient agents. Let each agent construct a Voronoi partition of its contiguous bins based on the position of its neighboring agents. The key idea for collision-free trajectory generation, as shown in Fig. 2.9, is to direct each stationary agent to the centroid of its Voronoi partition. Each transient agent is directed to the position in its Voronoi partition that has the minimum distance from the centroid of its target bin.

Let $\boldsymbol{p}^j_{k,\tau}$ denote the actual location of the j^{th} agent at the beginning of the τ^{th} collision-free trajectory generation loop at the k^{th} time step. Let $\hat{\boldsymbol{p}}^j_{k,\tau+1}$ represent the desired location of the j^{th} agent that is generated using the τ^{th} collision-free trajectory generation loop at the k^{th} time step. Let $\tau_{\max}$ denote the number of collision-free trajectory generation loops in each time step.

Let us define $\mathcal{W}(B[i])$ as the set of bins that are contiguous to the bin $B[i]$, i.e., the closure of the bins in $\mathcal{W}(B[i])$ share at least an edge or vertex with the closure of the bin $B[i]$ and $B[i] \in \mathcal{W}(B[i])$. Let $\mathcal{N}_{k,\tau}(\mathcal{W}(B[i]))$ denote the set of all agents in all the bins in $\mathcal{W}(B[i])$ during the τ^{th} collision-free trajectory generation loop at the k^{th} time step. It is assumed that the j^{th} agent in bin $B[i]$ is able to communicate with all the agents in all the bins in $\mathcal{W}(B[i])$. The distributed algorithm for the j^{th} agent in bin $B[i]$ during the τ^{th} collision-free trajectory generation loop at the k^{th} time step is given as follows:

1. The j^{th} agent in bin $B[i]$ transmits its actual location to other agents in $\mathcal{W}(B[i])$ and gathers the actual location of all the agents in $\mathcal{W}(B[i])$. Let $\mathcal{P}_{k,\tau}(\mathcal{W}(B[i])) = \left\{\boldsymbol{p}^\ell_{k,\tau} : \ell \in \mathcal{N}_{k,\tau}(\mathcal{W}(B[i]))\right\}$ represent the actual location of all the agents in $\mathcal{W}(B[i])$.

2. Let $\Xi(\mathcal{W}(B[i]))$ denote the largest convex polytope inside $\mathcal{W}(B[i])$. As the bin $B[i]$ is convex, $B[i] \subseteq \Xi(\mathcal{W}(B[i]))$. Construct the Voronoi partition $\mathcal{V}\left(\mathcal{P}_{k,\tau}(\mathcal{W}(B[i]))\right) = \left\{V^\ell_{k,\tau} : \ell \in \mathcal{N}_{k,\tau}(\mathcal{W}(B[i]))\right\}$ of the region $\Xi(\mathcal{W}(B[i]))$ based on points $\mathcal{P}_{k,\tau}(\mathcal{W}(B[i]))$.

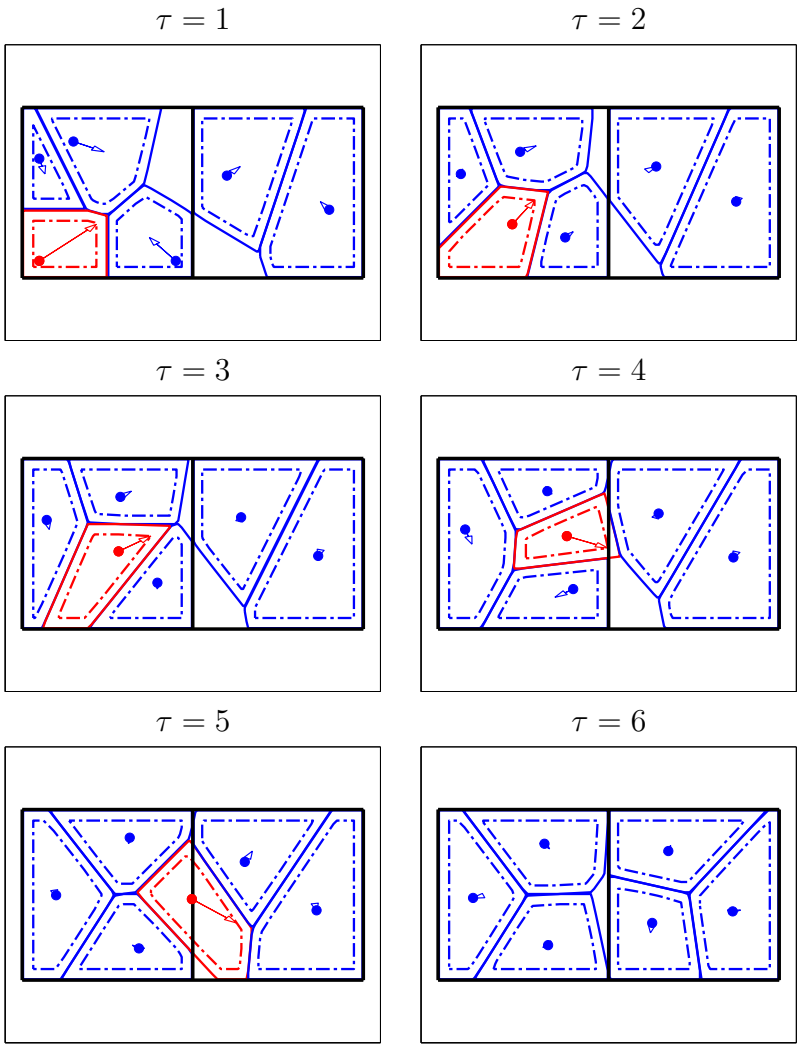


Figure 2.9: Time evolution of six agents in two bins (left and right bin), following the collision-free trajectory generation algorithm. The transient agent (in red) goes from the left bin to the right bin and the remaining stationary agents (in blue) give way to the transient agent. The Voronoi sets $(V_{k,\tau}^j, \tilde{V}_{k,\tau}^j, \check{V}_{k,\tau}^j)$ of all the agents along with their trajectories (denoted using arrows) during each collision-free trajectory generation loop are also shown.

The Voronoi set $V_{k,\tau}^j$ of the j^{th} agent is given by:

$$V_{k,\tau}^j = \left\{ \boldsymbol{x} \in \Xi(\mathcal{W}(B[i])) \,:\, \|\boldsymbol{x} - \boldsymbol{p}_{k,\tau}^j\|_2 \leq \|\boldsymbol{x} - \boldsymbol{p}_{k,\tau}^\ell\|_2,\, \forall \ell \in \mathcal{N}_{k,\tau}(\mathcal{W}(B[i])) \text{ and } j \neq \ell \right\}. \tag{2.42}$$

3. Depending on the type of the agent, set the new location of the j^{th} agent as follows:

(a) If the j^{th} agent is a transient agent, then let $\partial V_{k,\tau}^j$ denote the boundary of the Voronoi set $V_{k,\tau}^j$ obtained using (2.42). Construct a modified Voronoi set $\tilde{V}_{k,\tau}^j$ such that:

$$\tilde{V}_{k,\tau}^j = \left\{ \boldsymbol{x} \in V_{k,\tau}^j \, : \, \|\boldsymbol{x} - \partial V_{k,\tau}^j\|_2 \geq r_{\text{col}} \right\} . \tag{2.43}$$

If $\tilde{V}_{k,\tau}^j = \emptyset$, then set $\hat{\boldsymbol{p}}_{k,\tau+1}^j = \boldsymbol{p}_{k,\tau}^j$. Otherwise, set the desired location of the j^{th} agent as:

$$\hat{\boldsymbol{p}}_{k,\tau+1}^j = \arg \min_{\boldsymbol{x} \in \tilde{V}_{k,\tau}^j} \|\boldsymbol{x} - \boldsymbol{\kappa}_k^j\| , \tag{2.44}$$

where $\boldsymbol{\kappa}_k^j \in \mathcal{R}$ represents the centroid of its target bin.

(b) If the j^{th} agent is a stationary agent, then let $\partial(V_{k,\tau}^j \cap B[i])$ denote the boundary of the set $(V_{k,\tau}^j \cap B[i])$. Construct a modified Voronoi set $\breve{V}_{k,\tau}^j$ such that:

$$\breve{V}_{k,\tau}^j = \left\{ \boldsymbol{x} \in (V_{k,\tau}^j \cap B[i]) \, : \, \|\boldsymbol{x} - \partial(V_{k,\tau}^j \cap B[i])\|_2 \geq r_{\text{col}} \right\} . \tag{2.45}$$

If $\breve{V}_{k,\tau}^j = \emptyset$, then set $\hat{\boldsymbol{p}}_{k,\tau+1}^j = \boldsymbol{p}_{k,\tau}^j$. Otherwise, compute the centroid $\mathcal{C}(\breve{V}_{k,\tau}^j)$ of the modified Voronoi set $\breve{V}_{k,\tau}^j$, which is defined as:

$$\mathcal{C}(\breve{V}_{k,\tau}^j) = \frac{\int_{\breve{V}_{k,\tau}^j} \boldsymbol{x} d\boldsymbol{x}}{\int_{\breve{V}_{k,\tau}^j} d\boldsymbol{x}} , \tag{2.46}$$

where $\boldsymbol{x} \in \breve{V}_{k,\tau}^j$. Set the desired location of the j^{th} agent as the centroid of its modified Voronoi set $\breve{V}_{k,\tau}^j$, i.e., $\hat{\boldsymbol{p}}_{k,\tau+1}^j = \mathcal{C}(\breve{V}_{k,\tau}^j)$.

These steps are illustrated in Fig. 2.9 and in the pseudo-code in Algorithm 5. The computation time for generating Voronoi partitions using Fortune's algorithm is given by $O(n_a \log n_a)$, where n_a is the total number of agents in the region [148]. Moreover, the communication load of this distributed collision-free trajectory generation algorithm is the same as the consensus algorithm.

If the transient agent is close to the boundary of the bin $B[i]$, then these steps help

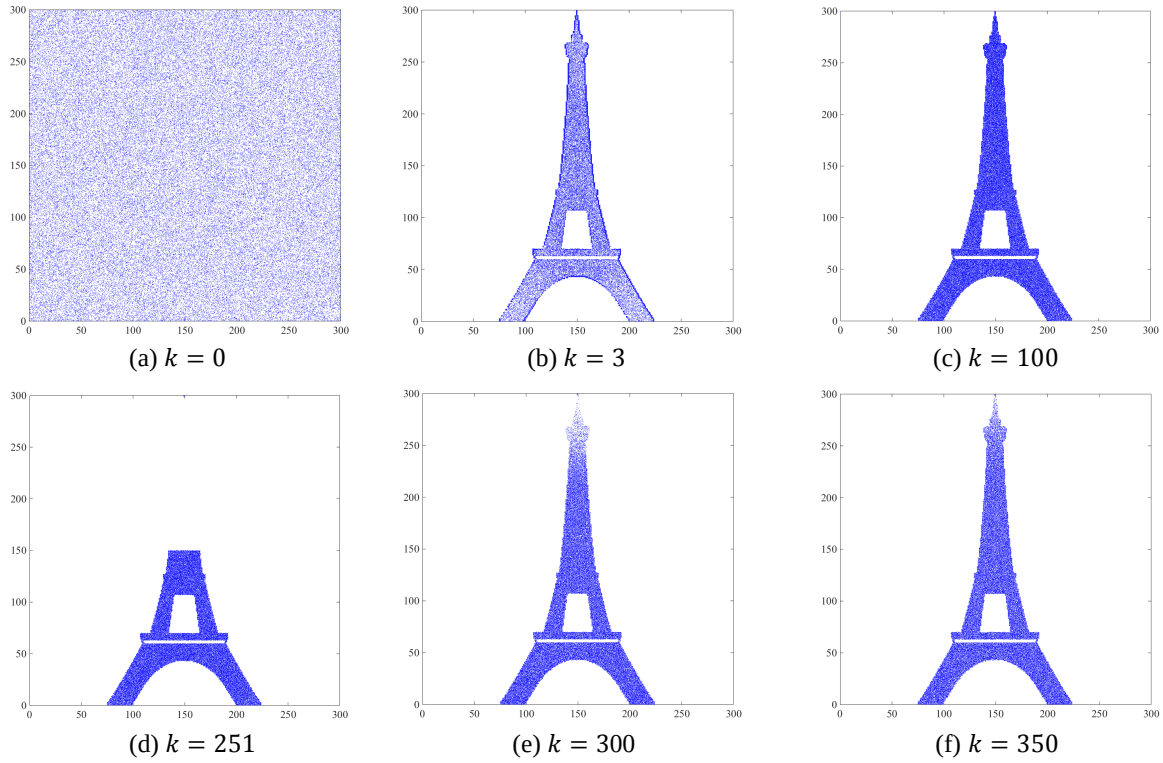
Collision-free trajectory generation loop of j^{th} agent during k^{th} time instant
Given present bin $B[i]$ and target bin $B[q]$
for $\tau = 1$ to $\tau_{\max}$
    Exchange locations $\boldsymbol{p}^{\ell}_{k,\tau}, \forall \ell \in \mathcal{J}^j_k$
    Compute $V^j_{k,\tau}$ using (2.42)
    if $B[q] = B[i]$, **then**
        Compute $\breve{V}^j_{k,\tau}$ using (2.45)
        if $\breve{V}^j_{k,\tau} = \emptyset$, **then**
            Set $\hat{\boldsymbol{p}}^j_{k,\tau+1} = \boldsymbol{p}^j_{k,\tau}$
        else
            Set $\hat{\boldsymbol{p}}^j_{k,\tau+1} = \mathcal{C}(\breve{V}^j_{k,\tau})$ using (2.46)
        end if
    else
        Compute $\tilde{V}^j_{k,\tau}$ using (2.43)
        if $\tilde{V}^j_{k,\tau} = \emptyset$, **then**
            Set $\hat{\boldsymbol{p}}^j_{k,\tau+1} = \boldsymbol{p}^j_{k,\tau}$
        else
            Set $\hat{\boldsymbol{p}}^j_{k,\tau+1}$ using (2.44)
        end if
    end if
end for

Algorithm 5: Generate collision-free trajectory using Voronoi partitions

it to cross the bin boundary and transition from bin $B[i]$ to a contiguous bin which is closer to its target bin. When the transient agent finally reaches its target bin, it becomes a stationary agent in that target bin. The advantage of this distributed collision-free trajectory generation algorithm using Voronoi partitions, where the Voronoi set is modified with the minimum collision avoidance distance, is that it guarantees collision avoidance. Since the region Ξ is convex, the Voronoi set ($V^j_{k,\tau}$) and the modified sets ($\tilde{V}^j_{k,\tau}$, $\breve{V}^j_{k,\tau}$) are also convex [149]. As each agent only moves within its modified Voronoi set, the agents are guaranteed to avoid collisions due to the buffer region of r_{col} along the boundary of each Voronoi set. Hence we have a simple, efficient, and distributed algorithm for collision-free trajectory generation using Voronoi partitions.

2.7 Numerical Simulations and Experiments

In this section, we present extensive numerical simulations and experiments to illustrate the properties of the PSG–IMC algorithms for shape formation and area exploration. We first show that the algorithm is capable of handling complex formation shapes, then show that the algorithm is robust to estimation errors, and finally show that the algorithm can be executed in real-time using quadrotors.



(a) $k = 0$ (b) $k = 3$ (c) $k = 100$

(d) $k = 251$ (e) $k = 300$ (f) $k = 350$

Figure 2.10: This plot shows the swarm distribution at different time instants, in a sample run of the Monte Carlo simulation. Starting from a uniform distribution, the swarm converges to the desired formation of the Eiffel Tower. After 250 time instants, the agents in the top half of the formation are removed and the remaining agents reconfigure to the desired formation.

2.7.1 Numerical Simulations for Shape Formation with Fine Spatial Resolution

In this numerical example, a swarm of 10^5 agents use the PSG–IMC algorithm for shape formation to achieve the desired formation $\boldsymbol{\Theta}$ of the Eiffel Tower (see $\boldsymbol{\nu}_{300}$ in Fig. 2.4), where the physical space is partitioned into 300×300 bins. The cumulative results of 10 Monte Carlo simulations are shown in Fig. 2.11. During each time instant, each agent gets the error-free feedback of the current swarm distribution $\boldsymbol{\mu}_k^\star$ and is allowed to transition to only those bins that are at most 50 steps away. The cost of transition is equal to the ℓ_1 distance between bins, therefore it is symmetric. Here we use the constants $\tau^j = 10^{-3}$ and $\beta^j = 1.8 \times 10^5$ in (2.21).

As shown in Fig. 2.10(a), each simulation starts with the agents uniformly distributed across the physical space. The agents attain the desired formation in 100 time instants. After 250 time instants, approximately 3×10^4 agents are removed from the top half of the formation (as shown in Fig. 2.10(d)) and the remaining agents reconfigure to the desired formation. Thus, the repeatability and robustness properties of the PSG–IMC algorithm for shape formation are evident in these simulation results.

The cumulative results for the HMC-based shape formation algorithm are also shown in Fig. 2.11. Compared to the HMC-based algorithm, the PSG–IMC algorithm provides approximately 2 times improvement in HD, 16 times reduction in the cumulative number of transitions in 500 time instants, and 16 times reduction in the total cost incurred by all the agents in 500 time instants. The key reasons behind the superior performance of the PSG–IMC algorithm are as follows:

(i) In Fig. 2.11, the HD of the HMC algorithm reaches an equilibrium at 0.115 after approximately 40 time instants. The HMC algorithm allows undesirable transitions (i.e., transitions from bins with fewer agents to bins with surplus agents) which increases the HD. Therefore, the HD for the HMC algorithm reaches an equilibrium because these undesirable transitions

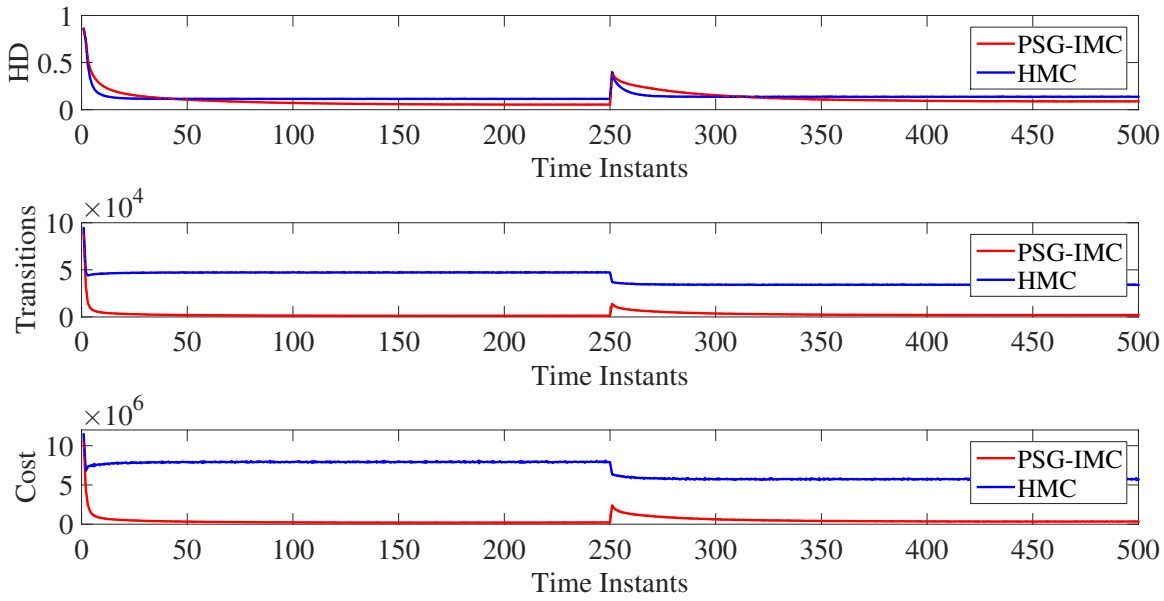


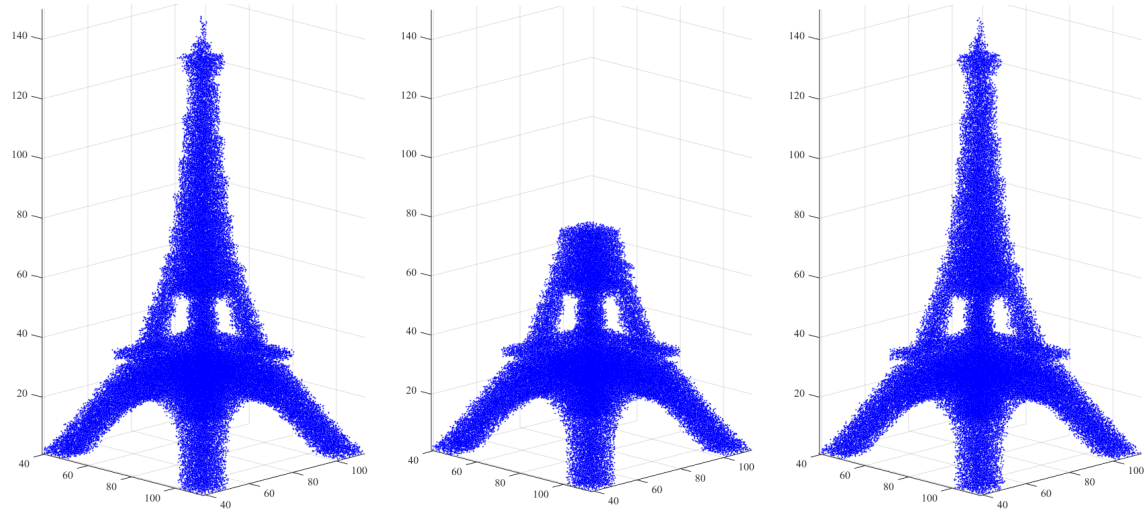
Figure 2.11: The cumulative results of 10 Monte Carlo simulations are shown. The discontinuity after the 250$^{\text{th}}$ time instant is because of the removal of agents from the top half of the formation.

reach an equilibrium with the other favorable transitions. Such undesirable transitions are largely avoided in the PSG–IMC algorithm (due to lines 10–11 in **Method 1**), hence the resulting HD after 250 time instants is 0.055 (i.e., $\approx$ 2 times improvement compared to HMC). The final HD can be further reduced by tuning τ^j and β^j. But such undesirable transitions prevent both these Markovian approaches from truly achieving zero convergence error.

(ii) In the HMC algorithm, there are 1.9×10^6 transitions in the first 40 time instants. This is significantly more than that of the PSG–IMC algorithm (i.e., 5.6×10^5 transitions in 250 time instant). In the PSG–IMC algorithm, the number of transitions at each time instant is proportional to the HD. This helps in achieving faster convergence (when HD is large) while avoiding unnecessary transitions (when HD is small). This also ensures that the agents settle down after the desired formation is achieved. Note that the total number of transitions in the HMC algorithm in 250 time instant is extremely large (i.e., 1.2×10^7 transitions).

Thus, the PSG–IMC algorithm achieves a smaller convergence error than the HMC-based algorithm and significantly reduces the number of transitions for achieving and maintain-

ing the desired formation. Moreover, these two key reasons depend on the feedback and, therefore, cannot be incorporated into HMC-based algorithms.



(a) $k = 4$ HD = 0.026 (b) $k = 5$ HD = 0.256 (c) $k = 10$ HD = 0.020

Figure 2.12: The swarm attains the 3-D shape of the Eiffel Tower. When agents are removed from the top half of the formation, the remaining agents reconfigure to the desired formation.

In the next example, we show that the PSG–IMC algorithm for shape formation can be used to make the 3-D shape of the Eiffel Tower (see Fig. 2.12). Here, the physical space is partitioned into $150 \times 150 \times 150$ bins and there are approximately 5.36×10^4 recurrent bins. Starting from a uniform distribution and no motion constraints, a swarm of 10^5 agents attain the desired formation in a few time instants. When 1.25×10^4 agents are removed from the top half of the formation, the remaining agents reconfigure to the desired formation in a few more time instants.

As another example, we show that the PSG–IMC algorithm can be used to make a desired formation shape with multiple disconnected parts (see Fig. 2.13). The physical space is partitioned into 325×325 bins. A swarm of 10^6 agents start from the left-most bin (located at $(1, 163)$) and attain the desired formation (Fig. 2.13(d)) in 300 time instants.

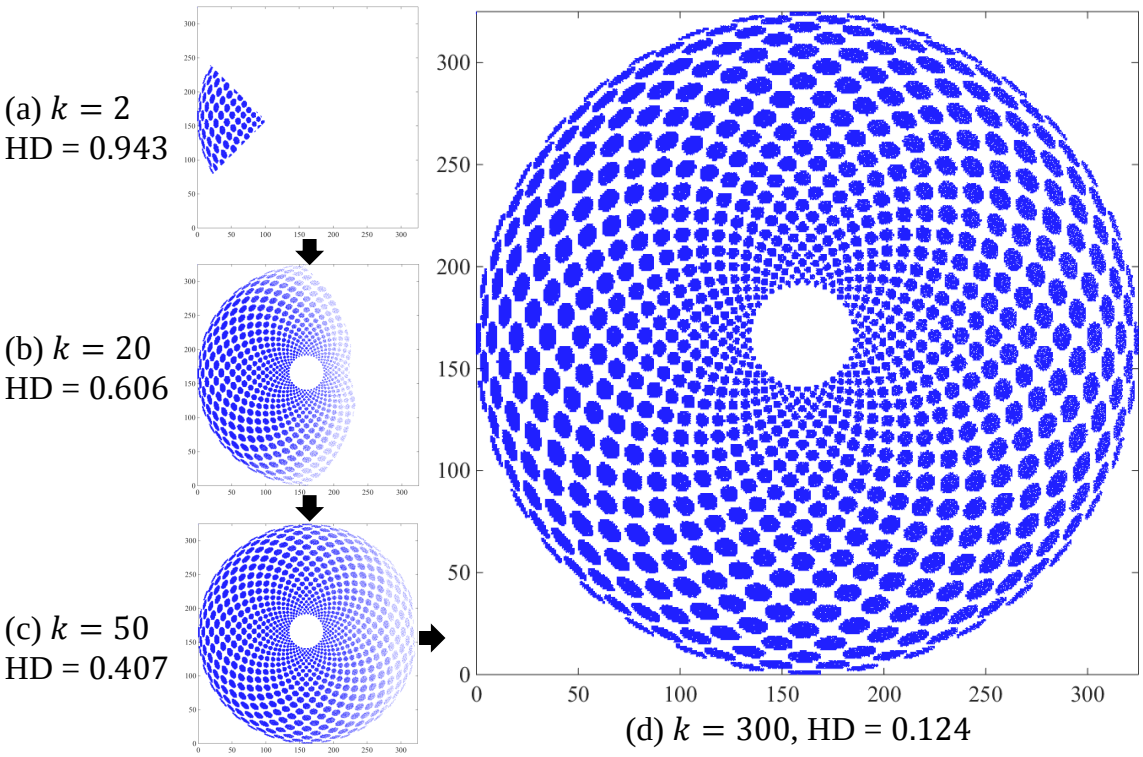


Figure 2.13: This plot shows the swarm distribution at different time instants, where the swarm attains the desired formation shape with multiple disconnected parts.

2.7.2 Numerical Simulations for Shape Formation with Coarse Spatial Resolution and Estimation Errors

The objective of this section is to study the effect of estimation errors on the three Eulerian swarm guidance algorithms, namely PSG–IMC, HMC, and the PSG–OT algorithm. In this numerical example, a swarm of 5000 agents achieve the desired formation $\boldsymbol{\Theta}$ of the Eiffel Tower (see $\boldsymbol{\nu}_{25}$ in Fig. 2.4), where the physical space is partitioned into 25×25 bins. The simulation setup is similar to that in Section 2.7.1. Each agent is allowed to transition to only those bins which are at most 9 steps away.

During each time instant, each agent gets the feedback of the current swarm distribution $\boldsymbol{\mu}_k^j$ with an estimation error ϵ_{est}. The cumulative results of Monte Carlo simulations are shown in Fig. 2.14. The PSG–OT algorithm performs slightly better than the PSG–IMC algorithm in the absence of an estimation error ($\epsilon_{\text{est}} = 0.0$), but such a situation does

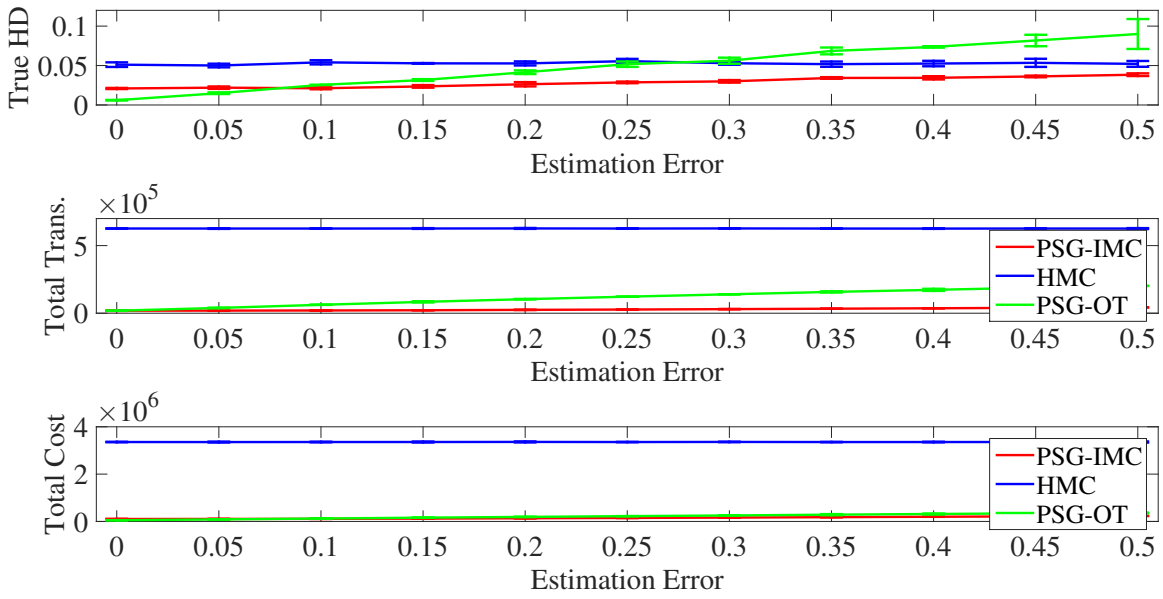


Figure 2.14: The estimation error is varied from 0.0 to 0.5. The performance of the three algorithms, along with 1σ error-bars, are shown for the true HD $D_H(\boldsymbol{\Theta}, \boldsymbol{\mu}^{\star}_{250})$ between the actual swarm distribution after 250 time instants and the desired formation, the cumulative number of transitions in 250 time instants, and the total cost incurred by all the agents in 250 time instants.

not arise in practical scenarios. The PSG–OT algorithm's true HD between the swarm distribution $\boldsymbol{\mu}^{\star}_{250}$ and the desired formation $\boldsymbol{\Theta}$ between drops precipitously in the presence of an estimation error and it performs worse than the open-loop HMC-based algorithm when $\epsilon_{\text{est}} \geq 0.25$. On the other hand, the PSG–IMC algorithm works reliably well for all estimation errors and performs much better than the other two algorithms. Thus, the PSG–IMC algorithm can tolerate large estimation errors in the feedback of the current swarm distribution.

The cumulative results for the three algorithms are shown in Fig. 2.15, where the estimation error ϵ_{est} is equal to 0.25. Compared to the HMC and PSG–OT algorithms, the PSG–IMC algorithm achieves a smaller convergence error with fewer transitions. The results of a few alternative functions for ξ^j_k are also shown in Fig. 2.15 (see Remark 2.4). The two functions $f_1(\xi^j_k) = \tanh(\pi\xi^j_k)$ and $f_2(\xi^j_k) = \sin\left(\cos^{-1}(1-\xi^j_k)\right)$ are always larger than ξ^j_k. The sigmoid function $f_3(\xi^j_k) = \left(\xi^j_k + 0.1\sin(2\pi\xi^j_k)\right)$ is larger than ξ^j_k when $\xi^j_k < 0.5$.

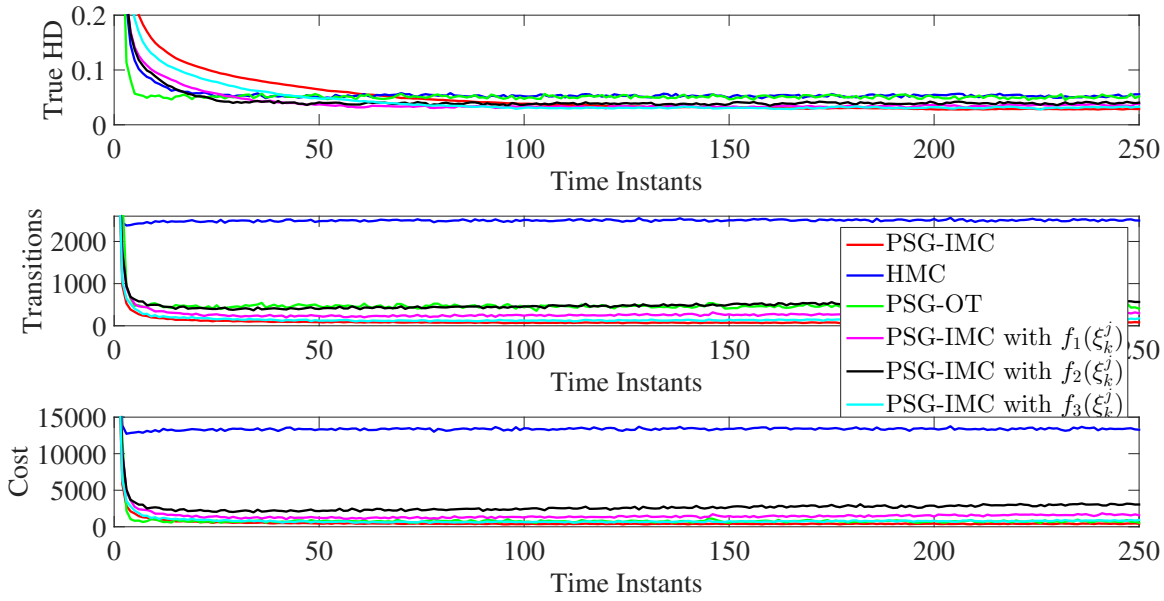


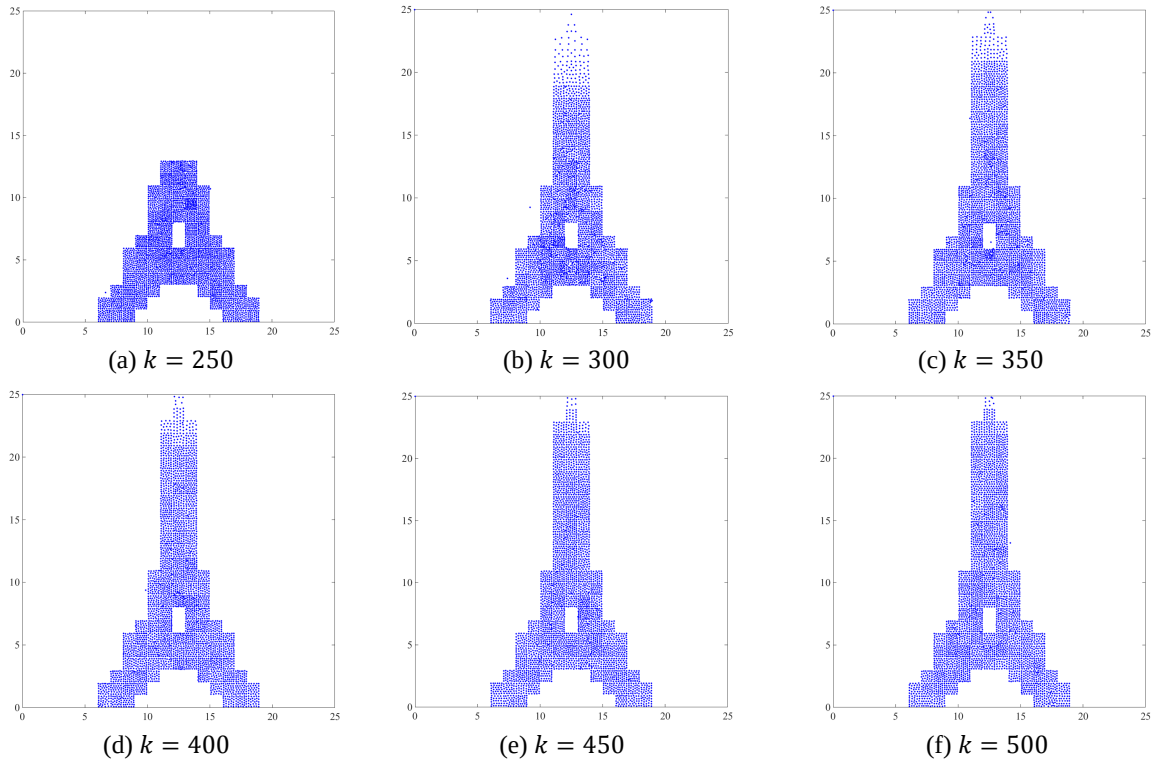
Figure 2.15: The cumulative results of the three algorithms and alternative functions for ξ_k^j are shown, where estimation error $\epsilon_{\text{est}} = 0.25$.

Fig. 2.15 shows that the rate of convergence increases with these functions, but there is also a corresponding increase in the number of transitions.

In Fig. 2.16, we show the swarm distribution at different time instants in a sample run of the Monte Carlo simulation, where the agents execute the PSG–IMC algorithm and the estimation error ϵ_{est} is equal to 0.25. Here the agents track their higher-level bin-to-bin guidance trajectory using the Voronoi partitions based collision-free trajectory generation algorithm presented in Section 2.6.

2.7.3 Experimental Results for Shape Formation to Demonstrate Real-Time Execution

In this section, we show that the PSG–IMC algorithm can be executed in real-time using quadrotors. The formation flying testbed is described in [150, 151]. Three to five quadrotors first generate their higher-level bin-to-bin guidance trajectory using the PSG–IMC algorithm. The quadrotors then track their higher-level bin-to-bin guidance trajectory using the Voronoi partitions based collision-free trajectory generation algorithm presented in Section 2.6. Nine



(a) $k = 250$ (b) $k = 300$ (c) $k = 350$

(d) $k = 400$ (e) $k = 450$ (f) $k = 500$

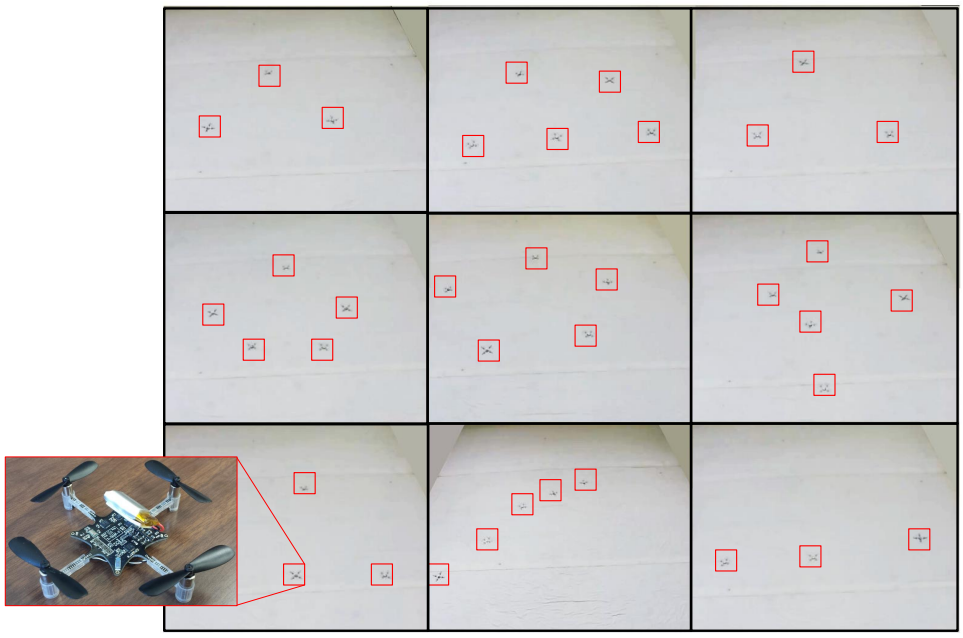
Figure 2.16: These plots show the swarm distribution at different time instants, after agents are removed from the top half of the formation.

separate experiments are shown in Fig. 2.17. The physical space is partitioned into 3×3 bins and the desired formations are shown in Fig. 2.17(b).

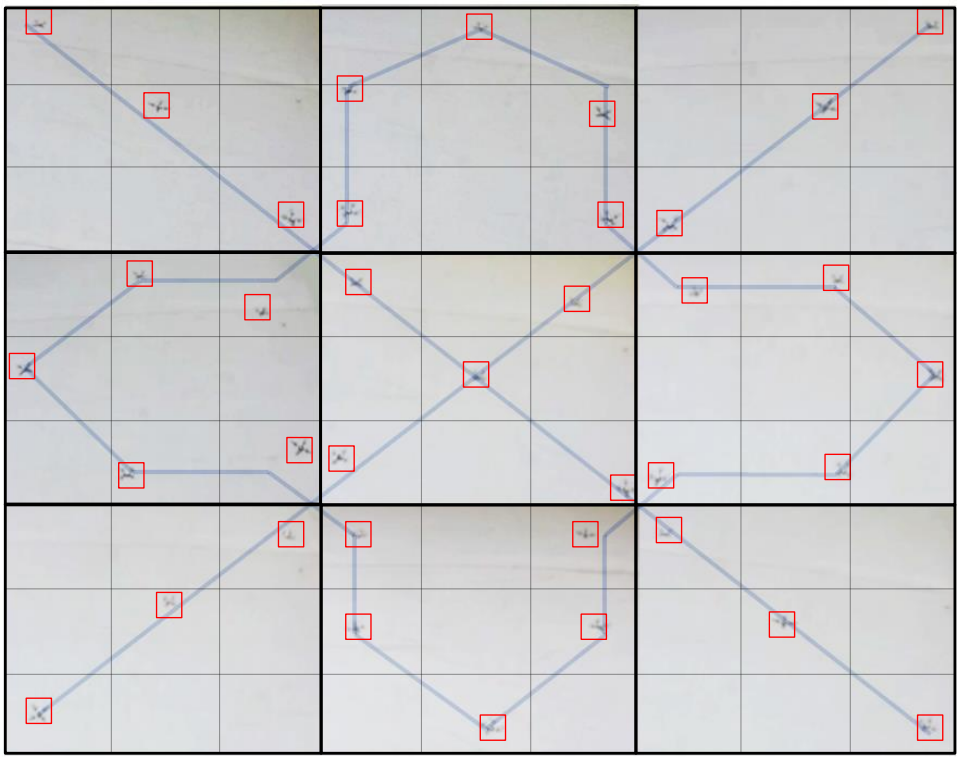
2.7.4 Numerical Simulations for Area Exploration

In this numerical example, a swarm of 10^5 agents use the PSG–IMC algorithm for area exploration to attain the unknown target distribution. The physical space $[0,1] \times [0,1]$ is partitioned into 100×100 bins. The unknown target distribution $\mathbf{\Omega}_1$ for the first 1000 time instants is given by the pmf representation of the multivariate normal distribution $\boldsymbol{\mathcal{N}}\left(\left[\begin{smallmatrix} 0.5 & 0.5 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0.1 & 0.3 \\ 0.3 & 1.0 \end{smallmatrix}\right]\right)$, as shown in the background contour plots in Fig. 2.18(a-c). Similarly, the unknown target distribution $\mathbf{\Omega}_2$ for the next 1000 time instants is given by $\boldsymbol{\mathcal{N}}\left(\left[\begin{smallmatrix} 0.5 & 0.5 \end{smallmatrix}\right], \left[\begin{smallmatrix} 0.1 & -0.3 \\ -0.3 & 1.0 \end{smallmatrix}\right]\right)$. Here we use the constants $\tau^j = 2.5 \times 10^{-3}$ and $\beta^j = 200$ in (2.34).

The cumulative results of 10 Monte Carlo simulations for different values of ξ^j are shown

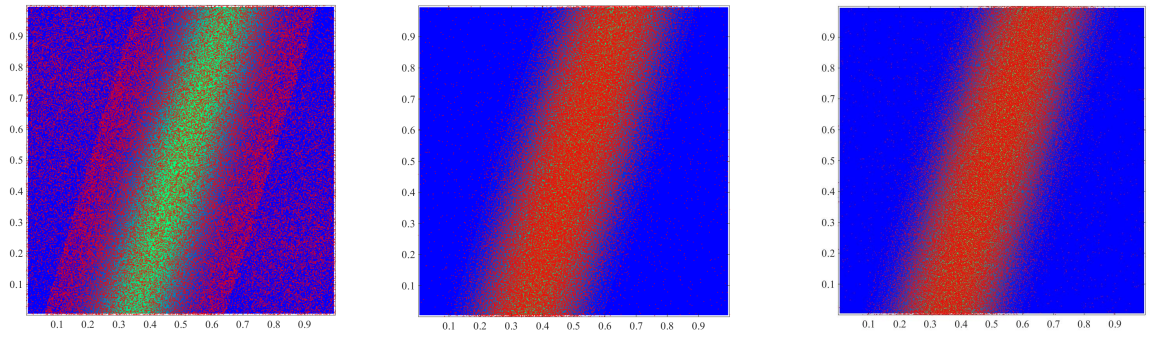


(a) Initial position

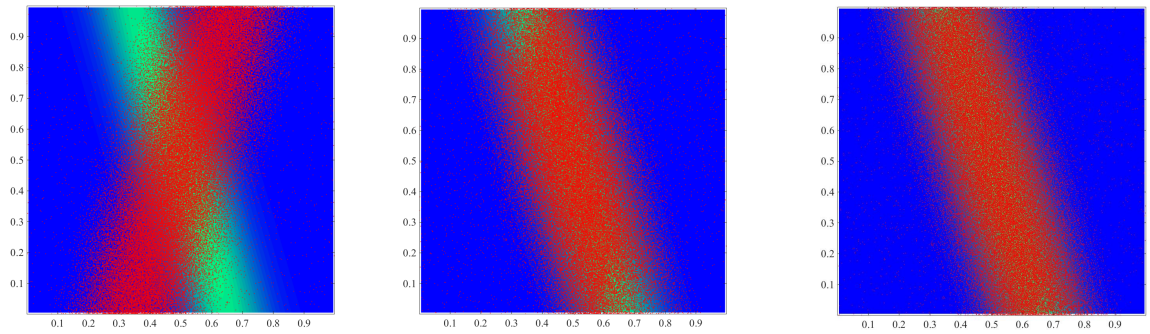


(b) Final position and desired formation (in blue)

Figure 2.17: Quadrotors execute the PSG–IMC algorithm is real-time to achieve the desired formation. The inset figure shows the Crazyflie 1.0 quadrotor used in these experiments.



(a) $k = 0$ (b) $k = 100$ (c) $k = 999$



(d) $k = 1000$ (e) $k = 1100$ (f) $k = 1200$

Figure 2.18: These plots show the swarm distribution of 10^5 agents (in red) and the unknown target distribution (background contour plot), in a sample run of the Monte Carlo simulation. Starting from a uniform distribution, the swarm converges to the unknown target distribution. After 1000 time instants, the unknown target distribution is suddenly changed and the agents reconfigure to this new target distribution.

in Fig. 2.19. A results for the HMC-based area exploration algorithm are also shown in Fig. 2.19. Compared to the HMC-based algorithm, the PSG–IMC algorithm provides approximately 1.5 times improvement in HD, 6 times reduction in the cumulative number of transitions in 2000 time instants, and 6 times reduction in the total cost incurred by the agents in 2000 time instants.

2.7.5 Application to Spacecraft Swarms

In this simulation example, the PSG–OT algorithm is used by a swarm of 5000 agents to attain the desired formation of the letters UIUC. The state space is partitioned into 30×30 bins and each agent is allowed to transition to only those bins which are at most 8 steps

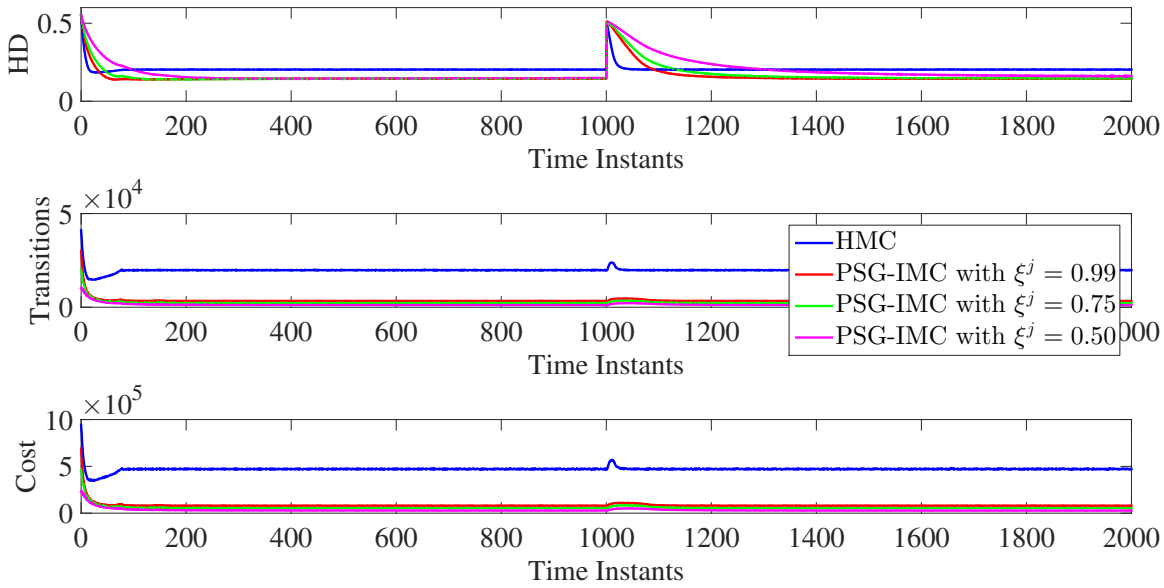


Figure 2.19: The cumulative results of 10 Monte Carlo simulations are shown. The discontinuity or change after the 1000^{th} time instant is because of the sudden change in the unknown target distribution.

away. The cost of transitions is given by the ℓ_1 distance between the bins, and the modified cost function is given by (2.38).

The histogram plots of the swarm distribution at different time steps are shown in Fig. 2.20. Starting from an uniform distribution, the swarm distribution rapidly converges to the desired formation within a couple of time steps using the PSG–OT algorithm. After the 250^{th} time step, the swarm is externally damaged by eliminating approximately 1250 ± 10 agents that form the letter I in the formation. The swarm quickly recovers from this damage and the remaining attain the desired stationary distribution within another couple of time steps.

During each time step, each agent executes the distributed collision-free trajectory generation algorithm to reach its target bin. The trajectories of 5000 agents, executing 20 collision-free trajectory generation loops during the first PSG–OT time step, are shown in Fig. 2.21. After the 250^{th} time step, the swarm is externally damaged and the agents that form the letter I are removed. The trajectories of the remaining 3750 agents for the next two PSG–OT time steps are shown in Fig. 2.22.

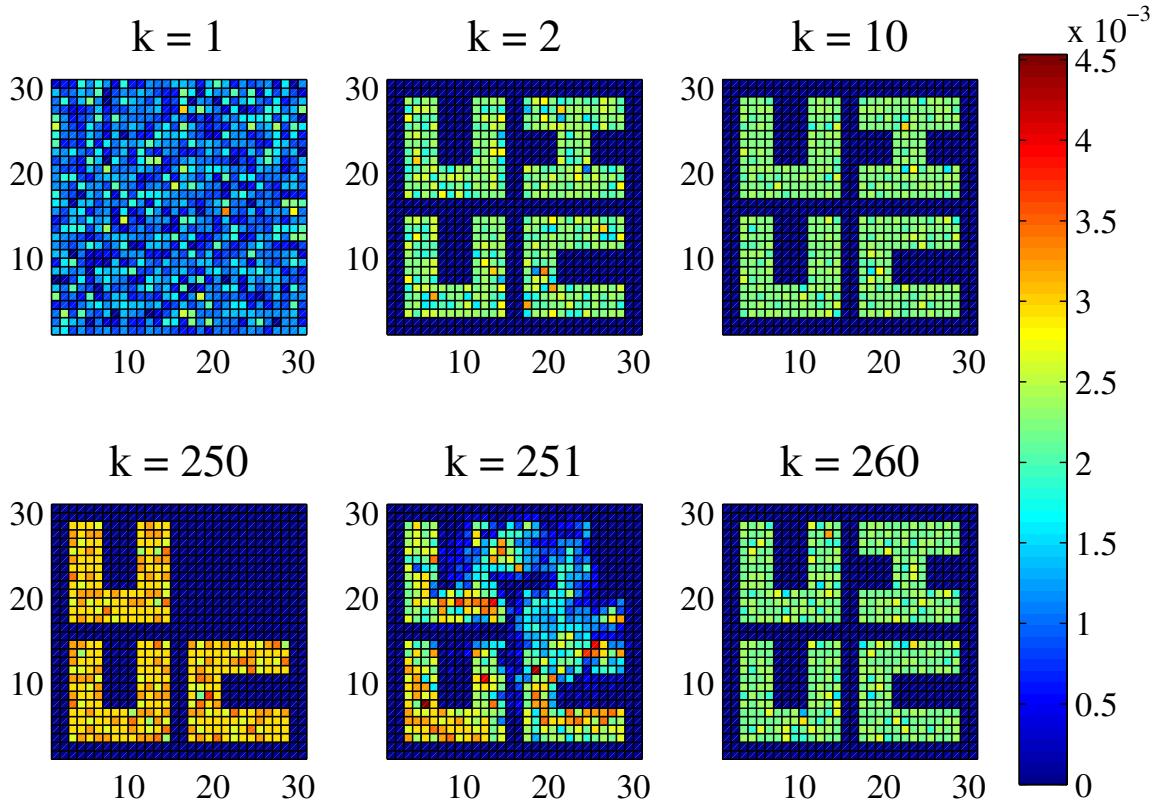


Figure 2.20: Histogram plots of the swarm distribution at different time steps for 5000 agents executing PSG–OT. The colorbar represents the pmf of the swarm distribution. Starting form an uniform distribution, the swarm converges to the desired formation within a couple of time steps. The middle section of the swarm is externally damaged and 1248 agents are removed after the 250^{th} time step, but the swarm autonomously recovers within another couple of time steps.

We now show that the PSG–OT algorithm can be applied to spacecraft swarms in Earth orbit, where each spacecraft in the swarm determines its collision-free desired trajectory so that the overall swarm converges to the desired formation.

We assume that there exists a real/virtual chief spacecraft in the Earth Centered Inertial (ECI) coordinate system, as shown in Fig. 2.23(a). The origin of ECI frame is located at the center of Earth, the $\hat{X}$ axis points towards the vernal equinox and the $\hat{Z}$ axis points towards the north pole. The Local Vertical Local Horizontal (LVLH) coordinate system, also shown in Fig. 2.23(a), is centered at the chief spacecraft and is designed to model the relative motion of all other deputy spacecraft with respect to this chief spacecraft. The $\hat{x}$

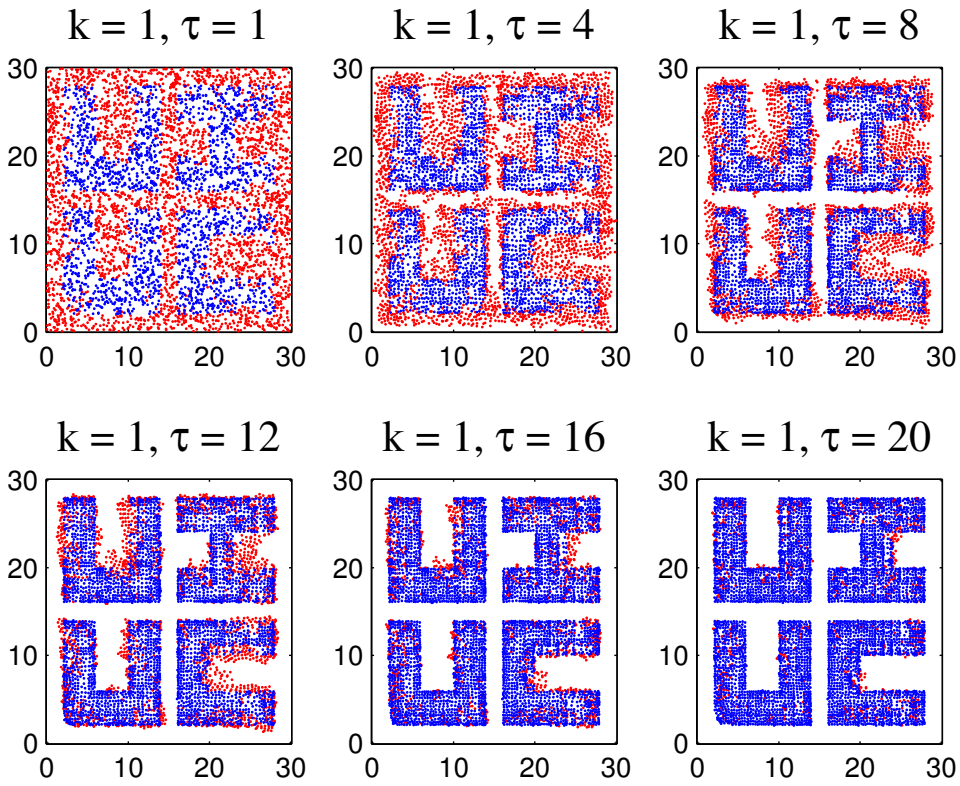


Figure 2.21: Time evolution of the actual location of 5000 agents, during multiple collision-free trajectory generation loops within one PSG–OT time step, is shown. The transient agents are in red and the remaining stationary agents are in blue.

direction is always aligned with the position vector from the center of Earth to the chief spacecraft and points away from the Earth, the $\hat{z}$ direction is aligned with the angular momentum vector, and the $\hat{y}$ direction completes the right–handed coordinate system. The relative states of the j^{th} spacecraft with respect to the chief spacecraft in the LVLH frame is expressed by $\left[x_t^j,\, y_t^j,\, z_t^j,\, \dot{x}_t^j,\, \dot{y}_t^j,\, \dot{z}_t^j\right]$, where t represents continuous time. The linear Hill–Clohessy–Wiltshire (HCW) equations, that describe the evolution of the relative states of the j^{th} deputy spacecraft with respect to the chief spacecraft, are given by [5]:

$$\ddot{x}_t^j - 2n\dot{y}_t^j - 3n^2 x_t^j = 0, \quad \ddot{y}_t^j + 2n\dot{x}_t^j = 0, \quad \ddot{z}_t^j + n^2 z_t^j = 0, \tag{2.47}$$

where the orbital rate n depends on the chief's orbit. The general solution of the HCW

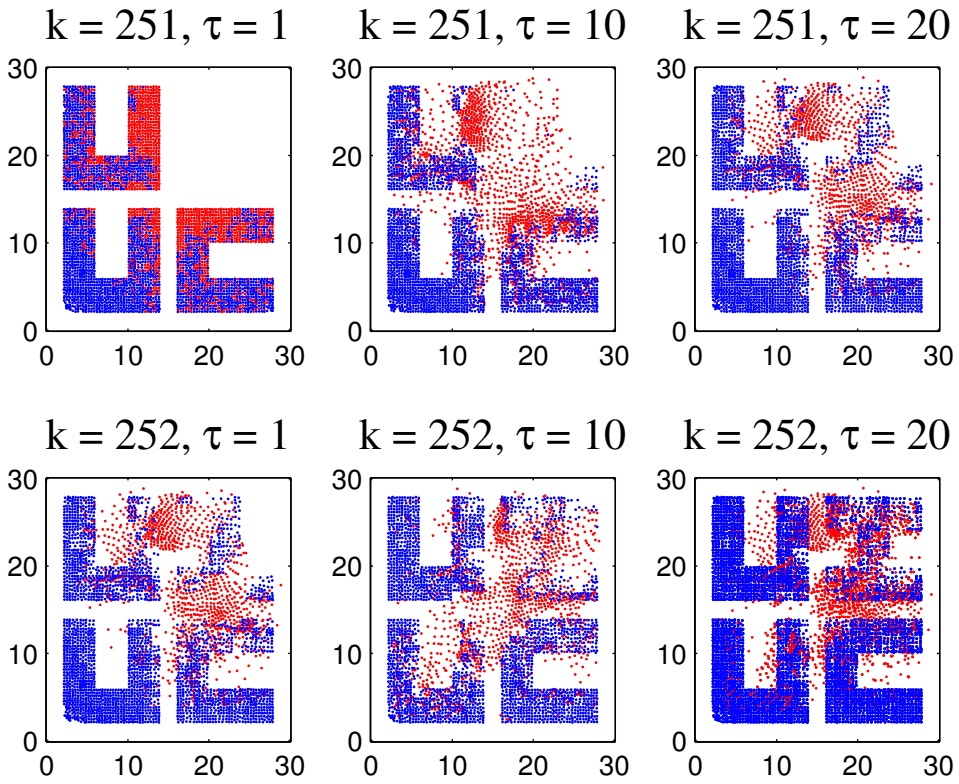


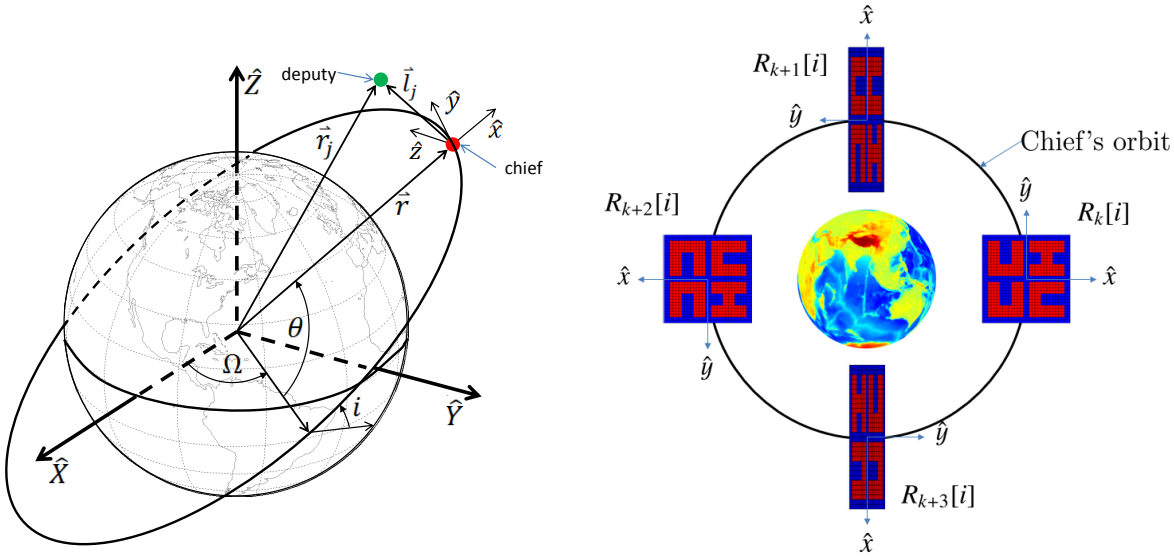
Figure 2.22: After the 250$^{\text{th}}$ time step, the swarm is externally damaged by eliminating the agents that form the letter I. Time evolution of the actual location of the remaining 3750 agents, during multiple collision-free trajectory generation loops within two PSG–OT time steps, is shown. The transient agents are in red and the remaining stationary agents are in blue.

equations is given by [5]:

$$x_t^j = \frac{c_1}{2}\sin(nt + \alpha_0), \quad y_t^j = c_1\cos(nt + \alpha_0) + c_3, \quad z_t^j = c_2\sin(nt + \beta_0), \tag{2.48}$$

where the constants $c_1, c_2, c_3, \alpha_0,$ and β_0 depend on the initial conditions of the deputy spacecraft's position and velocity. Note that the semi-major axis along $\hat{y}$ is twice that of the semi-minor axis along $\hat{x}$. In this chapter we assume that the spacecraft swarm is in the $\hat{x} - \hat{y}$ plane, i.e., $z_t^j = 0$ for all agents.

Passive relative orbits (PROs) for deputy spacecraft are closed elliptical trajectories in LVLH frame that are generated using some initial conditions and no further control input.



(a) (b)

Figure 2.23: (a) Earth Centered Inertial $(\hat{X}, \hat{Y}, \hat{Z})$ and LVLH $(\hat{x}, \hat{y}, \hat{z})$ frames (Reproduced with permission from Ref. [5]) (b) Time-varying bins $(R_k[i], \forall i \in \{1, \ldots, n_{\text{bin}}\})$ designed using linearized concentric Pros. The stretching of the formation at certain time steps is a consequence of the HCW solution.

The initial conditions for generating linearized concentric PROs in the LVLH frame are [5]:

$$\dot{x}_0^j = \frac{1}{2} n y_0^j, \quad \dot{y}_0^j = -2n x_0^j, \quad \dot{z}_0^j = 0\,. \tag{2.49}$$

These initial conditions ensure that the resulting PROs, generated using the HCW equations, are period-matched and concentric elliptical trajectories the $\hat{x} - \hat{y}$ plane of the LVLH frame. The resulting PRO has the following constant terms in the general HCW solution (2.48): $c_1 = \sqrt{(2x_0^j)^2 + (y_0^j)^2}$, $c_2 = c_3 = 0$, and $\alpha_0 = \tan^{-1}\left(\frac{2x_0^j}{y_0^j}\right)$. Time-varying bins $(B_k[i], \forall i \in \{1, \ldots, n_{\text{bin}}\})$ are designed so that the spacecraft continues to coast along the PRO in the LVLH frame while transitioning from $B_k[i]$ to $B_{k+1}[i]$, as shown in Fig. 2.23(b). Similar time-varying bins can be designed for the collision-free trajectory generation loops within each time step.

Motion constraints could arise from the spacecraft dynamics, as it might be prohibitively fuel-expensive for the spacecraft in a certain bin to transition to another bin within a single

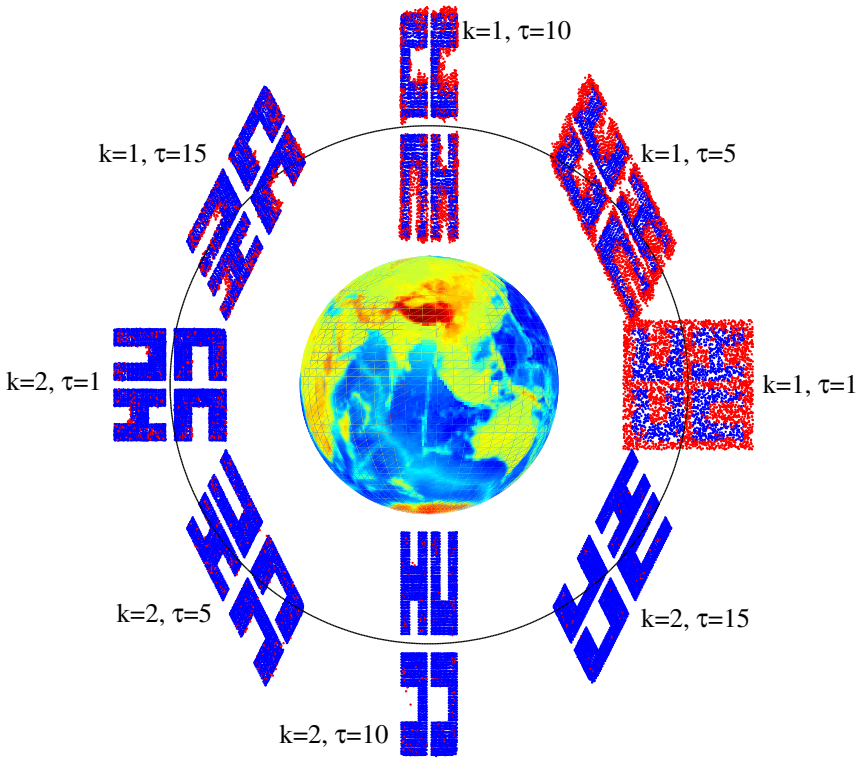


Figure 2.24: Evolution of the shape of the spacecraft swarm in Earth orbit. Starting from an initial distribution, the swarm converges to the desired formation within a few PSG–OT time steps.

time step due to the large distance between these bins. Some constraints could also arise from a limit on the amount of fuel that can be consumed in a single time step or limited control authority. A time-varying motion constraints matrix is designed to handle such motion constraints due to spacecraft dynamics.

If each agent in the spacecraft swam in Earth orbit executes the PSG–OT algorithm, then the swarm converges to the desired formation as the grid-based solution can be mapped to the time-varying bins as shown in Fig. 2.24. This proves the effectiveness of the proposed algorithm for application to spacecraft swarms. Moreover, if some of the spacecraft cannot maintain their desired position due to thruster faults or external disturbances and are lost from the swarm, then the remaining working spacecraft will successfully maintain the formation.

2.8 Chapter Summary

In this chapter, we presented the development of probabilistic swarm guidance algorithms for shape formation and area exploration using the Eulerian framework. Using hardware experiments and numerical simulations, we demonstrated the effectiveness of the proposed algorithms for shape formation, area exploration, and collision-free trajectory generation. To our knowledge, the PSG–IMC algorithm is the first path planning strategy that leverages the idea of constructing IMC in real-time based on state feedback.

Chapter 3

Distributed Estimation

In this chapter, we present discrete-time distributed estimation algorithms for a heterogeneous sensor network that guarantee bounded convergence to the optimal probability distribution of the targets states. The algorithms presented in this chapter have been published in [95, 152].

This chapter is organized as follows. Section 3.1 presents some preliminaries and the problem statement. The LogOP scheme and some convergence results are presented in Section 3.2. The DBF algorithm and its special cases are presented in Section 3.3. The BCF algorithm is presented in Section 3.4. Results of numerical simulations are presented in Section 3.5 and the chapter is concluded in Section 3.6.

3.1 Preliminaries and Problem Statement

In this section, we first introduce the target dynamics and measurement models, and then state the recursive Bayesian filtering algorithm. We then present the communication network topology and finally state the problem statement of this chapter.

The state space of the target's states $\mathcal{X}$ is a compact set in $\mathbb{R}^{n_x}$, where n_x is the dimension of the states of the target. Let $\mathscr{X}$ be the Borel σ–algebra for $\mathcal{X}$. A probability space is defined by the three-tuple $\{\mathcal{X}, \mathscr{X}, \mathbb{P}\}$, where $\mathbb{P}$ is a complete, σ-additive probability measure on all $\mathscr{X}$. Let $p(\boldsymbol{x}) = \frac{d\mathbb{P}(\boldsymbol{x})}{d\mu(\boldsymbol{x})}$ denote the Radon–Nikodým density of the probability distribution $\mathbb{P}(\boldsymbol{x})$ with respect to a measure $\mu(\boldsymbol{x})$. When $\boldsymbol{x} \in \mathcal{X}$ is continuous and $\mu(\boldsymbol{x})$ is a Lebesgue measure, $p(\boldsymbol{x})$ is the probability density function (pdf) [153]. Therefore, the probability of

an event $\mathscr{A} \in \mathscr{X}$ can be written as the Lebesgue–Stieltjes integral $\mathbb{P}(\mathscr{A}) = \int_{\mathscr{A}} p(\boldsymbol{x})\, d\mu(\boldsymbol{x})$. In this paper, we only deal with the continuous case where the function $p(\cdot)$ represents the pdf and $\mu(\cdot)$ is the Lebesgue measure. Let $\Phi(\mathcal{X})$ represent the set of all pdfs over the state space $\mathcal{X}$. The L_1 distance and the KL divergence between the pdfs $\mathcal{P}_1, \mathcal{P}_2 \in \Phi(\mathcal{X})$ are denoted by:

$$D_{L_1}(\mathcal{P}_1, \mathcal{P}_2) = \int_{\mathcal{X}} |\mathcal{P}_1 - \mathcal{P}_2|\ d\mu(\boldsymbol{x})\,,$$
$$D_{\mathrm{KL}}(\mathcal{P}_1 || \mathcal{P}_2) = \int_{\mathcal{X}} \mathcal{P}_1(\boldsymbol{x}) \log\left(\frac{\mathcal{P}_1(\boldsymbol{x})}{\mathcal{P}_2(\boldsymbol{x})}\right)\ d\mu(\boldsymbol{x})\,.$$

In this paper, all the algorithms are presented in discrete time. Let Δ be the time step between any two consecutive time instants.

3.1.1 Target Dynamics and Measurement Models

Let $\boldsymbol{x}_k$ represent the true states of the target at the k^{th} time instant, where $\boldsymbol{x}_k \in \mathcal{X}$ for all $k \in \mathbb{N}$. The dynamics of the target in discrete time is given by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}_k(\boldsymbol{x}_k, \boldsymbol{v}_k)\,,\ \forall k \in \mathbb{N}\,, \tag{3.1}$$

where $\boldsymbol{f}_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ is a possibly nonlinear time-varying function of the state $\boldsymbol{x}_k$ and an independent and identically distributed (i.i.d.) process noise $\boldsymbol{v}_k$, where n_v is the dimension of the process noise vector.

Consider a network of N heterogeneous sensing agents simultaneously tracking this target. Let $\boldsymbol{y}_k^i$ represent the measurement taken by the i^{th} agent at the k^{th} time instant. The measurement model of the agents is given by:

$$\boldsymbol{y}_k^i = \boldsymbol{h}_k^i(\boldsymbol{x}_k, \boldsymbol{w}_k^i),\ \forall i \in \mathcal{V} = \{1, \ldots, N\}\,,\ \forall k \in \mathbb{N}\,, \tag{3.2}$$

where $\boldsymbol{h}_k^i : \mathbb{R}^{n_x} \times \mathbb{R}^{n_{wi}} \to \mathbb{R}^{n_{yi}}$ is a possibly nonlinear time-varying function of the state $\boldsymbol{x}_k$ and an i.i.d. measurement noise $\boldsymbol{w}_k^i$, where n_{yi} and n_{wi} are dimensions of the measurement and measurement noise vectors respectively. Therefore, the measurements are conditionally independent given the target's states. Note that this measurement model is quite general since it accommodates heterogeneous sensors with various sensing and noise characteristics and partial state observation.

3.1.2 Recursive Bayesian Filtering Algorithm

Each agent uses the recursive Bayesian filtering algorithm to estimate the pdf of the states of the target. This algorithm consists of two steps: (i) the prior pdf of the target's states is obtained during the prediction step, and (ii) the posterior pdf of the target's states is updated using the new measurement during the update step [154]–[157]. Let $\boldsymbol{x}_{k|k-1}$ and $\boldsymbol{x}_{k|k}$ represent the predicted and updated states of the target at the k^{th} time instant. Let the pdfs $\mathcal{S}_k^i = p(\boldsymbol{x}_{k|k-1}) \in \Phi(\mathcal{X})$ and $\mathcal{W}_k^i = p(\boldsymbol{x}_{k|k}) \in \Phi(\mathcal{X})$ denote the i^{th} agent's prior and posterior pdfs of the target's states at the k^{th} time instant.

During the prediction step, the prior pdf $\mathcal{S}_k^i = p(\boldsymbol{x}_{k|k-1})$ is obtained from the previous posterior pdf $\mathcal{W}_{k-1}^i = p(\boldsymbol{x}_{k-1|k-1})$ of the $(k-1)^{\text{th}}$ time instant using the Chapman–Kolmogorov equation [153]:

$$\mathcal{S}_k^i = \int_{\mathcal{X}} p(\boldsymbol{x}_{k|k-1}|\boldsymbol{x}_{k-1|k-1}) p(\boldsymbol{x}_{k-1|k-1}) d\mu(\boldsymbol{x}_{k-1|k-1}). \tag{3.3}$$

The probabilistic model of the state evolution $p(\boldsymbol{x}_{k|k-1}|\boldsymbol{x}_{k-1|k-1})$ is defined by the target dynamics model (3.1) and the known statistics of the i.i.d. process noise $\boldsymbol{v}_k$. We assume that the prior pdf is available at the start of the estimation process.

Assumption 3.1. For each agent, the initial prior of the target's states $\mathcal{S}_1^i = p(\boldsymbol{x}_{1|0})$ at the start of the estimation process (i.e., 1^{st} time instant) is available. If no prior knowledge is available, then $p(\boldsymbol{x}_{1|0})$ is assumed to be a uniformly distribution. □

The new measurement $\boldsymbol{y}_k^i$ is used to compute the posterior pdf $\mathcal{W}_k^i = p(\boldsymbol{x}_{k|k})$ during the update step using the Bayes' rule [153]:

$$\mathcal{W}_k^i = \frac{p(\boldsymbol{y}_k^i|\boldsymbol{x}_{k|k-1})\, p(\boldsymbol{x}_{k|k-1})}{\int_{\mathcal{X}} p(\boldsymbol{y}_k^i|\boldsymbol{x}_{k|k-1})\, p(\boldsymbol{x}_{k|k-1})\, d\mu(\boldsymbol{x}_{k|k-1})}\,. \tag{3.4}$$

The likelihood function $p(\boldsymbol{y}_k^i|\boldsymbol{x}_{k|k-1})$ is defined by the i^{th} agent's measurement model (3.2), and the corresponding known statistics of the i.i.d. measurement noise $\boldsymbol{w}_k^i$. It is a function of the variable $\boldsymbol{x}_{k|k-1}$, given the outcome of the new measurement $\boldsymbol{y}_k^i$. Let the pdf $\mathcal{L}_k^i \in \Phi(\mathcal{X})$ represent the normalized likelihood function of the i^{th} agent at the k^{th} time instant, i.e.,

$$\mathcal{L}_k^i = \frac{p(\boldsymbol{y}_k^i|\boldsymbol{x}_{k|k-1})}{\int_{\mathcal{X}} p(\boldsymbol{y}_k^i|\boldsymbol{x}_{k|k-1})\, d\mu(\boldsymbol{x}_{k|k-1})}\,. \tag{3.5}$$

Therefore, the pdf $\mathcal{W}_k^i = \frac{\mathcal{L}_k^i \mathcal{S}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k^i \mathcal{S}_k^i\, d\mu(\boldsymbol{x}_{k|k-1})}$.

If all the sensing agents are hypothetically connected by a complete graph, then the agents can exchange their likelihood functions and each agent can use the multi-sensor Bayesian filtering algorithm to compute the posterior pdf of the target's states. For this centralized Bayesian filter, the posterior pdf $\mathcal{W}_k^{C,i} = p(\boldsymbol{x}_{k|k}) \in \Phi(\mathcal{X})$ is obtained using the Bayes' rule [158]:

$$\mathcal{W}_k^{C,i} = \frac{\left(\prod_{j=1}^{N} p(\boldsymbol{y}_k^j|\boldsymbol{x}_{k|k-1})\right)\, p(\boldsymbol{x}_{k|k-1})}{\int_{\mathcal{X}} \left(\prod_{j=1}^{N} p(\boldsymbol{y}_k^j|\boldsymbol{x}_{k|k-1})\right)\, p(\boldsymbol{x}_{k|k-1})\, d\mu(\boldsymbol{x}_{k|k-1})}\,. \tag{3.6}$$

Bayesian filtering is optimal because this posterior pdf $\mathcal{W}_k^{C,i}$ integrates and uses all the available information expressed by probabilities (assuming they are quantitatively correct) [153]. Moreover, an optimal state estimate with respect to any criterion can be computed from this posterior pdf $\mathcal{W}_k^{C,i}$. The minimum mean-square error (MMSE) estimate and the

maximum a posteriori (MAP) estimate are given by [159]:

$$\hat{\boldsymbol{x}}_{k|k}^{MMSE} = \int_{\mathcal{X}} \boldsymbol{x}\, \mathcal{W}_k^{C,i}\, d\mu(\boldsymbol{x})\,,$$
$$\hat{\boldsymbol{x}}_{k|k}^{MAP} = \arg\max_{\boldsymbol{x}\in\mathcal{X}} \mathcal{W}_k^{C,i}\,.$$

Other potential criteria for optimality, such as maximum likelihood, minimum conditional KL divergence, and minimum free energy, are discussed in [153, 159]. Let the pdf $\mathcal{L}_k^C \in \Phi(\mathcal{X})$ represent the normalized joint likelihood function of all the sensing agents at the k^{th} time instant, i.e.,

$$\mathcal{L}_k^C = \frac{\left(\prod_{j=1}^N p(\boldsymbol{y}_k^j|\boldsymbol{x}_{k|k-1})\right)}{\int_{\mathcal{X}} \left(\prod_{j=1}^N p(\boldsymbol{y}_k^j|\boldsymbol{x}_{k|k-1})\right)\, d\mu(\boldsymbol{x}_{k|k-1})}\,. \tag{3.7}$$

Therefore, the pdf $\mathcal{W}_k^{C,i} = \frac{\mathcal{L}_k^C \mathcal{S}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k^C \mathcal{S}_k^i\, d\mu(\boldsymbol{x}_{k|k-1})}$.

The main advantage of Bayesian filters is that no approximation is needed during the filtering process; i.e., the complete information about the dynamics and uncertainties of the model can be incorporated in the filtering algorithm. However, Bayesian filtering is computationally expensive. Practical implementation of these algorithms, in their most general form, is achieved using particle filtering [157, 160] and Bayesian programming [161, 162].

3.1.3 Problem Statement

Let the pdf $\mathcal{T}_k^i \in \Phi(\mathcal{X})$ denote the estimated likelihood function of the i^{th} agent at the k^{th} time instant. The aim of this paper is to design a discrete-time distributed estimation algorithm, over the communication network topology given in Section 3.1.4, so that each agent's estimated likelihood function $\mathcal{T}_k^i$ converges to an error ball around the joint likelihood function $\mathcal{L}_k^C$ (3.7), i.e., after κ^{th} time instant:

$$D_{L_1}\left(\mathcal{T}_k^i, \mathcal{L}_k^C\right) \leq \delta\,, \quad \forall k \geq \kappa\,, \forall i \in \mathcal{V}\,, \tag{3.8}$$

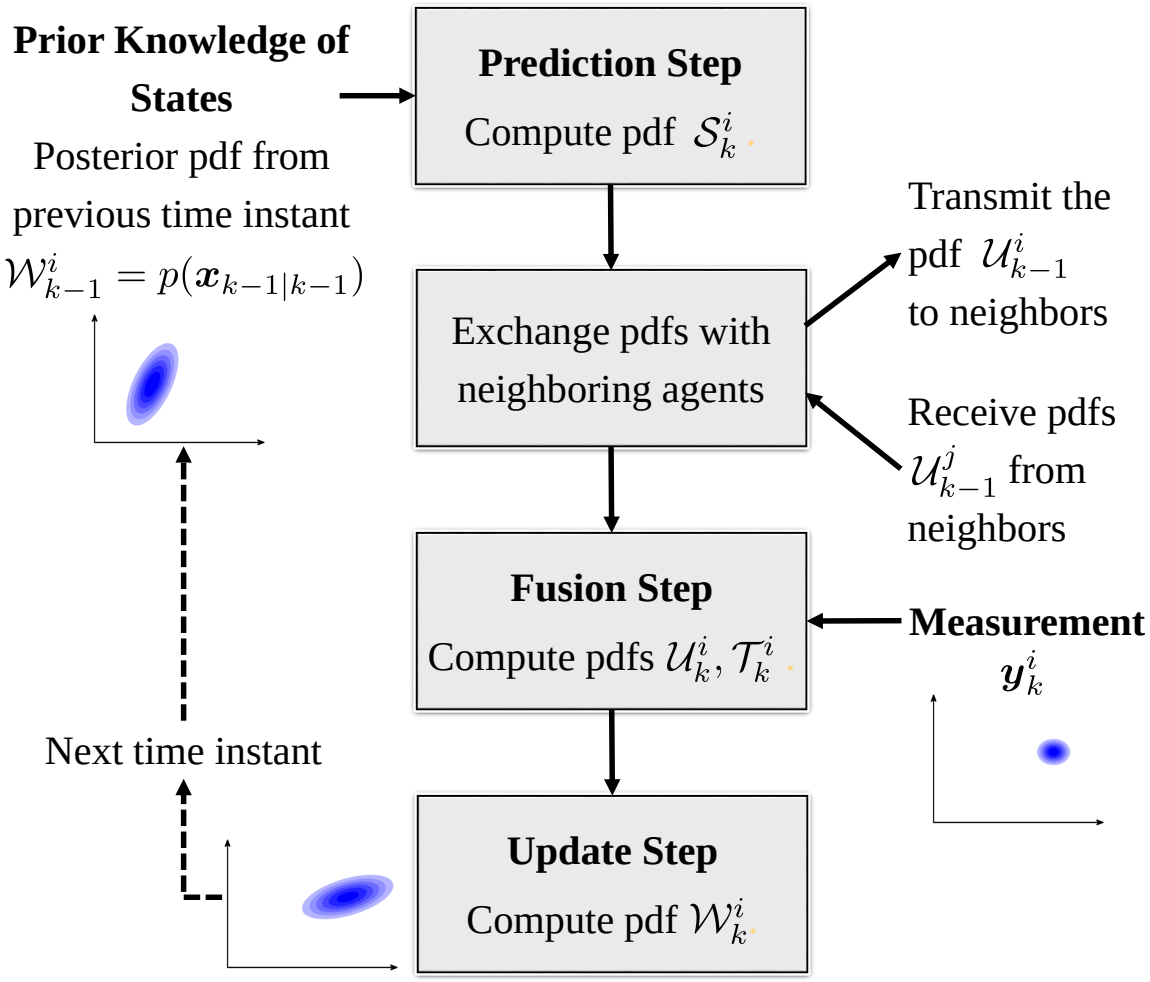


Figure 3.1: Flowchart of the DBF algorithm (for the i^{th} agent at the k^{th} time instant)

where δ is the desired convergence error.

The DBF algorithm, shown in Fig. 3.1, achieves the objective of (3.8). After the prediction step, the agent's exchange their estimated pdfs with their neighboring agents to estimate the joint likelihood function. Then the agents compute the posterior pdf during the update step. The pseudo-code of the DBF algorithm is presented in Algorithm 6.

3.1.4 Communication Network Topology

The time-varying communication network topology of the sensor network is given by the directed graph $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ with the edge set $\mathcal{E}_k \subset \mathcal{V} \times \mathcal{V}$ and $(i, j) \in \mathcal{E}_k$ if and only if

the i^{th} agent receives information from the j^{th} agent at the k^{th} time instant. The inclusive neighbors of the i^{th} agent at k^{th} time instant are denoted by $\mathcal{J}_k^i = \{j \in \mathcal{V} : (i,j) \in \mathcal{E}_k\} \cup \{i\}$. The matrix $\mathcal{A}_k \in \mathbb{R}^{N \times N}$ represents the adjacency matrix of $\mathcal{G}_k$, where $\mathcal{A}_k[i,j] \neq 0$ if and only if $j \in \mathcal{J}_k^i$. The following assumption on the communication network topology has been widely used in the literature [103]–[163].

Assumption 3.2. *(Periodic Strong Connectivity with Balanced Adjacency Matrix)* The time-varying communication network topology $\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)$ and its adjacency matrix $\mathcal{A}_k$ satisfy the following properties:

(i) There is some positive integer $\mathfrak{b} \geq 1$ such that, for all time instants $k \in \mathbb{N}$, the directed graph $(\mathcal{V}, \mathcal{E}_k \cup \mathcal{E}_{k+1} \cup \ldots \cup \mathcal{E}_{k+\mathfrak{b}-1})$ is strongly connected. If $\mathcal{G}_k$ is strongly connected for all time instants, then $\mathfrak{b} = 1$.

(ii) There exists a constant $\gamma > 0$ such that $\mathcal{A}_k[i,j] \in [\gamma, 1]$ for all $j \in \mathcal{J}_k^i \backslash \{i\}$ and $\mathcal{A}_k[i,i] = 1 - \sum_{j \neq i} \mathcal{A}_k[i,j] \geq \gamma$ for all $k \in \mathbb{N}$.

(iii) The matrix $\mathcal{A}_k$ is doubly stochastic, i.e., $\mathbf{1}^T \mathcal{A}_k = \mathbf{1}^T$ and $\mathcal{A}_k \mathbf{1} = \mathbf{1}$ for all $k \in \mathbb{N}$.

Therefore, the digraph $\mathcal{G}_k$ is periodically strongly connected and the matrix $\mathcal{A}_k$ is non-degenerate and balanced. □

3.2 Logarithmic Opinion Pool and Convergence Results

In this section, we first state the LogOP scheme for combining probability distributions and then present some convergence results. Let the pdf $\mathcal{P}_k^i \in \Phi(\mathcal{X})$ denote the i^{th} agent's pdf at the k^{th} time instant. The LinOP and LogOP schemes for combining the pdfs $\mathcal{P}_k^i$ of all the

agents are given by [97]:

$$\mathcal{P}_k^{\text{LinOP}}(\boldsymbol{x}) = \sum_{i=1}^{N} \alpha_k^i \mathcal{P}_k^i(\boldsymbol{x}) \,, \tag{3.9}$$

$$\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{x}) = \frac{\Pi_{i=1}^{N} \left(\mathcal{P}_k^i(\boldsymbol{x})\right)^{\alpha_k^i}}{\int_{\mathcal{X}} \Pi_{i=1}^{N} \left(\mathcal{P}_k^i(\bar{\boldsymbol{x}})\right)^{\alpha_k^i} \, d\mu(\bar{\boldsymbol{x}})} \,, \tag{3.10}$$

where the weights α_k^i are such that $\sum_{i=1}^{N} \alpha_k^i = 1$ and the integral in the denominator of (3.10) is finite. Thus, the combined pdf obtained using LinOP and LogOP gives the weighted algebraic and geometric average of the individual pdfs respectively. As shown in Fig. 3.2, the combined pdf obtained using LogOP typically preserves the exact multimodal or unimodal nature of the original individual pdfs [99]. The most compelling reason for using the LogOP scheme is that it is externally Bayesian; i.e., the LogOP combination step commutes with the process of updating the pdfs by multiplying with a commonly agreed likelihood pdf $\mathcal{L}_k \in \Phi(\mathcal{X})$:

$$\frac{\mathcal{L}_k \, \mathcal{P}_k^{\text{LogOP}}}{\int_{\mathcal{X}} \mathcal{L}_k \, \mathcal{P}_k^{\text{LogOP}} d\mu(\tilde{\boldsymbol{x}})} = \frac{\Pi_{i=1}^{N} \left(\frac{\mathcal{L}_k \, \mathcal{P}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k \, \mathcal{P}_k^i \, d\mu(\bar{\boldsymbol{x}})} \right)^{\alpha_k^i}}{\int_{\mathcal{X}} \Pi_{i=1}^{N} \left(\frac{\mathcal{L}_k \, \mathcal{P}_k^i}{\int_{\mathcal{X}} \mathcal{L}_k \, \mathcal{P}_k^i \, d\mu(\bar{\boldsymbol{x}})} \right)^{\alpha_k^i} d\mu(\tilde{\boldsymbol{x}})}.$$

Therefore, the LogOP scheme is ideal for combining pdfs in distributed estimation algorithms.

Due to the multiplicative nature of the LogOP scheme, each agent has veto power [99]. That is, if $\mathcal{P}_k^i(\boldsymbol{x}) = 0$ for some $\boldsymbol{x} \in \mathcal{X}$ and some agent $i \in \mathcal{V}$ with $\alpha_k^i > 0$, then $\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{x}) = 0$ in the combined pdf irrespective of the pdfs of the other agents. In order to avoid this veto condition, we enforce the following assumption which has been used in the literature [88, 99].

Assumption 3.3. *(Zero Probability Property)* In this paper, all pdfs are strictly positive everywhere in the compact set $\mathcal{X}$. Moreover, all pdfs are upper bounded by some large value. Note that the actual lower and upper bounds are not used in the convergence analysis. □

We actually need the pdfs to be positive only on the support of the probability distri-

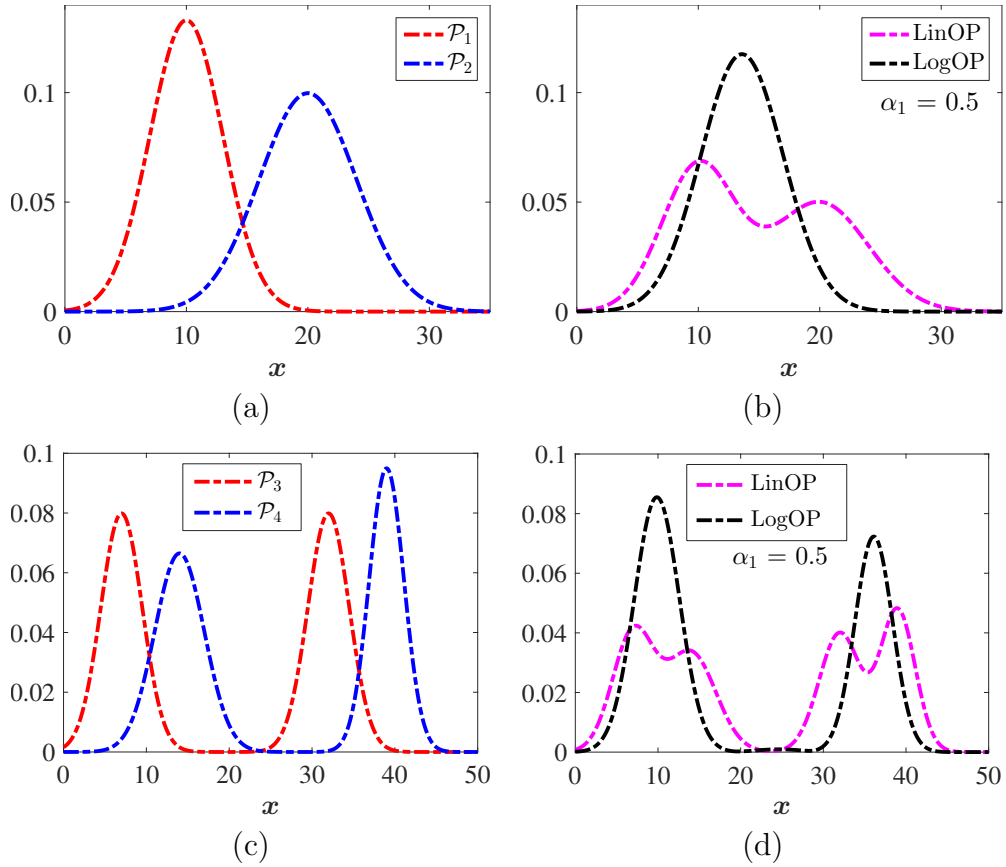


Figure 3.2: In (a), two unimodal pdfs $\mathcal{P}_1$ and $\mathcal{P}_2$ are shown. In (b), these pdfs are combined using the LinOP and LogOP using the weight $\alpha_1 = \alpha_2 = 0.5$. Note that the LinOP solution is multimodal while the LogOP solution is unimodal. In (c), the pdfs $\mathcal{P}_3$ and $\mathcal{P}_4$ have bimodal nature. In (d), the LogOP solution preserves this bimodal nature.

bution of the target's states. Since we do not know this support beforehand, therefore we enforce the pdfs to be positive everywhere in $\mathcal{X}$.

In order to analyze the LogOP scheme with general probability distributions, we use the following function that is designed to state the LogOP scheme as a linear equation and remove the effect of the normalizing constants.

Definition 3.1. For any constant $\boldsymbol{\psi} \in \mathcal{X}$, it follows from Assumption 3.3 that $\mathcal{P}_k^i(\boldsymbol{\psi}) > 0$ for all agents and $\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{\psi}) > 0$. Using simple algebraic manipulation of (3.10), we get

[100]:

$$\mathscr{P}_k^{\text{LogOP}}(\boldsymbol{x}) := \ln\left[\frac{\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{x})}{\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{\psi})}\right] = \sum_{i=1}^{N} \alpha_k^i \mathscr{P}_k^i(\boldsymbol{x})\,, \tag{3.11}$$

$$\text{where } \mathscr{P}_k^i(\boldsymbol{x}) := \ln\left[\frac{\mathcal{P}_k^i(\boldsymbol{x})}{\mathcal{P}_k^i(\boldsymbol{\psi})}\right]\,,\ \forall i \in \mathcal{V}\,, \tag{3.12}$$

$$\frac{\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{x})}{\mathcal{P}_k^{\text{LogOP}}(\boldsymbol{\psi})} = \frac{\left(\frac{\Pi_{i=1}^N \left(\mathcal{P}_k^i(\boldsymbol{x})\right)^{\alpha_k^i}}{\int_{\mathcal{X}} \Pi_{i=1}^N \left(\mathcal{P}_k^i(\bar{\boldsymbol{x}})\right)^{\alpha_k^i} d\mu(\bar{\boldsymbol{x}})}\right)}{\left(\frac{\Pi_{i=1}^N \left(\mathcal{P}_k^i(\boldsymbol{\psi})\right)^{\alpha_k^i}}{\int_{\mathcal{X}} \Pi_{i=1}^N \left(\mathcal{P}_k^i(\bar{\boldsymbol{x}})\right)^{\alpha_k^i} d\mu(\bar{\boldsymbol{x}})}\right)}\,. \tag{3.13}$$

Under Assumption 3.3, the functions $\mathscr{P}_k^i(\boldsymbol{x})$ and $\mathscr{P}_k^{\text{LogOP}}(\boldsymbol{x})$ are all well-defined functions. Thus, we have represented the LogOP scheme (3.10) as a linear equation using these functions. The actual value of the constant $\boldsymbol{\psi}$ is chosen during the convergence analysis.

We now define pointwise convergence and then show that the functions in Definition 3.1 can be used to prove convergence of their corresponding pdfs.

Definition 3.2. *(Pointwise Convergence)* Let the pdf $\mathcal{P}^\star \in \Phi(\mathcal{X})$ denote the limiting pdf. The pdf $\mathcal{P}_k^i$ converges pointwise to the pdf $\mathcal{P}^\star$, if and only if $\lim_{k\to\infty} \mathcal{P}_k^j(\boldsymbol{x}) = \mathcal{P}^\star(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$.

Lemma 3.1. *We define the function* $\mathscr{P}^\star(\boldsymbol{x}) := \ln\left[\frac{\mathcal{P}^\star(\boldsymbol{x})}{\mathcal{P}^\star(\boldsymbol{\psi})}\right]$ *for any* $\boldsymbol{\psi} \in \mathcal{X}$*. Under Assumption 3.3, if the function* $\mathscr{P}_k^i$ *from (3.12) converges pointwise to the function* $\mathscr{P}^\star$*, then the corresponding pdf* $\mathcal{P}_k^i$ *also converges pointwise to the pdf* $\mathcal{P}^\star$*. Moreover, there exists* $\boldsymbol{\psi} \in \mathcal{X}$ *such that* $\lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{\psi}) = \mathcal{P}^\star(\boldsymbol{\psi})$.

Proof: If $\lim_{k\to\infty} \mathscr{P}_k^i(\boldsymbol{x}) = \mathscr{P}^\star(\boldsymbol{x})$ pointwise, then:

$$\lim_{k\to\infty} \left(\ln \mathcal{P}_k^i(\boldsymbol{x}) - \ln \mathcal{P}_k^i(\boldsymbol{\psi})\right) = \ln \mathcal{P}^\star(\boldsymbol{x}) - \ln \mathcal{P}^\star(\boldsymbol{\psi}). \tag{3.14}$$

We first show that there exists $\boldsymbol{\psi} \in \mathcal{X}$ such that $\lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{\psi}) = \mathcal{P}^\star(\boldsymbol{\psi})$. If this claim is untrue, then either $0 < \lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{x}) < \mathcal{P}^\star(\boldsymbol{x})$ or $0 < \mathcal{P}^\star(\boldsymbol{x}) < \lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{x})$ for all

$\boldsymbol{x} \in \mathcal{X}$ because all pdfs satisfy Assumption 3.3. Hence either $\int_{\mathcal{X}} \lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{x}) d\mu(\boldsymbol{x}) = 1 < \int_{\mathcal{X}} \mathcal{P}^\star(\boldsymbol{x}) d\mu(\boldsymbol{x})$ or $\int_{\mathcal{X}} \mathcal{P}^\star(\boldsymbol{x}) d\mu(\boldsymbol{x}) < \int_{\mathcal{X}} \lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{x}) d\mu(\boldsymbol{x}) = 1$, which results in contradiction since $\int_{\mathcal{X}} \mathcal{P}^\star(\boldsymbol{x}) d\mu(\boldsymbol{x}) = 1$. Hence, such a $\boldsymbol{\psi} \in \mathcal{X}$ must exist.

Substituting $\lim_{k\to\infty} \mathcal{P}_k^i(\boldsymbol{\psi}) = \mathcal{P}^\star(\boldsymbol{\psi})$ into (3.14) gives $\lim_{k\to\infty} \ln \mathcal{P}_k^i(\boldsymbol{x}) = \ln \mathcal{P}^\star(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$. This implies that $\lim_{k\to\infty} \mathcal{P}_k^j(\boldsymbol{x}) = \mathcal{P}^\star(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$ since logarithm is a monotonic function. ■

We now define convergence in total variation (TV) and present a connection with pointwise convergence.

Definition 3.3. *(Convergence in TV)* The measure $\mu_{\mathcal{P}_k^i}$ is defined as the measure induced by the pdf $\mathcal{P}_k^i$ on $\mathscr{X}$, where $\mu_{\mathcal{P}_k^i}(\mathscr{A}) = \int_{\mathscr{A}} \mathcal{P}_k^i \, d\mu(\boldsymbol{x})$ for any event $\mathscr{A} \in \mathscr{X}$. Similarly, let $\mu_{\mathcal{P}^\star}$ denote the measure induced by the pdf $\mathcal{P}^\star$ on $\mathscr{X}$.

The TV distance is defined as $\|\mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^\star}\|_{\text{TV}} := \sup_{\mathscr{A}\in\mathscr{X}} |\mu_{\mathcal{P}_k^i}(\mathscr{A}) - \mu_{\mathcal{P}^\star}(\mathscr{A})|$. The measure $\mu_{\mathcal{P}_k^i}$ converges to the measure $\mu_{\mathcal{P}^\star}$ in TV, if and only if $\| \lim_{k\to\infty} \mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^\star}\|_{\text{TV}} = 0$.

Lemma 3.2. *(Pointwise Convergence implies Convergence in TV) If the pdf $\mathcal{P}_k^i$ converges to the pdf $\mathcal{P}^\star$ pointwise, then the measure $\mu_{\mathcal{P}_k^i}$ converges in TV to the measure $\mu_{\mathcal{P}^\star}$. Moreover, $\|\mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^\star}\|_{\text{TV}} = \frac{1}{2} D_{L_1}(\mathcal{P}_k^i, \mathcal{P}^\star)$.*

Proof: Similar to the proof of Scheffé's theorem [143, pp. 84], under Assumption 3.3, using the dominated convergence theorem (cf. [143, Theorem 1.5.6, pp. 23]) for any event $\mathscr{A} \in \mathscr{X}$ gives:

$$\lim_{k\to\infty} \int_{\mathscr{A}} \mathcal{P}_k^i d\mu(\boldsymbol{x}) = \int_{\mathscr{A}} \lim_{k\to\infty} \mathcal{P}_k^i d\mu(\boldsymbol{x}) = \int_{\mathscr{A}} \mathcal{P}^\star d\mu(\boldsymbol{x}) \,.$$

This relation between measures implies that $\| \lim_{k\to\infty} \mu_{\mathcal{P}_k^i} - \mu_{\mathcal{P}^\star}\|_{\text{TV}} = 0$. The relationship between TV error and L_1 distance follows from [164, pp. 48]. ■

Another reason for using the LogOP scheme is that it gives the KL-divergence-minimizing pdf as shown in the following lemma.

Lemma 3.3. *[94, 95] The pdf $\mathcal{P}_k^{\mathrm{KL}} \in \Phi(\mathcal{X})$ that globally minimizes the sum of KL divergences with the pdfs $\mathcal{P}_k^i$ for all agents is given by:*

$$\mathcal{P}_k^{\mathrm{KL}} = \underset{\rho \in \Phi(\mathcal{X})}{\arg\min} \sum_{i=1}^{N} D_{\mathrm{KL}}\left(\rho || \mathcal{P}_k^i\right) = \frac{\prod_{i=1}^{N} \left(\mathcal{P}_k^i\right)^{\frac{1}{N}}}{\int_{\mathcal{X}} \prod_{i=1}^{N} \left(\mathcal{P}_k^i\right)^{\frac{1}{N}} d\mu(\bar{\boldsymbol{x}})}\,. \tag{3.15}$$

Note that the pdf $\mathcal{P}_k^{\mathrm{KL}}$ is equivalent to the pdf $\mathcal{P}_k^{\mathrm{LogOP}}$ (3.10) obtained using the LogOP scheme with weights $\alpha_k^i = \frac{1}{N}$ for all agents.

Proof: The sum of the KL divergences of a pdf $\rho \in \Phi(\mathcal{X})$ with the locally estimated posterior pdfs is given by:

$$\sum_{j=1}^{N} D_{\mathrm{KL}}\left(\rho || \mathcal{P}_k^j\right) = \sum_{j=1}^{N} \int_{\mathcal{X}} \left(\rho \ln(\rho) - \rho \ln(\mathcal{P}_k^j)\right) d\mu(\boldsymbol{x}). \tag{3.16}$$

Under Assumption 3.3, $D_{\mathrm{KL}}\left(\rho || \mathcal{P}_k^j\right)$ is well defined for all agents. Differentiating (3.16) with respect to ρ using Leibniz integral rule [143, Theorem A.5.1, pp. 372], and equating it to zero gives:

$$\sum_{j=1}^{N} \int_{\mathcal{X}} \left(\ln(\rho) + 1 - \ln(\mathcal{P}_k^j)\right)\, d\mu(\boldsymbol{x}) = 0\,, \tag{3.17}$$

where $\rho^{\mathrm{KL}} = \frac{1}{e} \prod_{j=1}^{N} (\mathcal{P}_k^j)^{1/N}$ is the solution of (3.17). The projection of ρ^{KL} on the set $\Phi(\mathcal{X})$, which is obtained by normalizing ρ^{KL}, is the pdf $\mathcal{P}_k^{\mathrm{KL}} \in \Phi(\mathcal{X})$ given by (3.15).

The KL divergence is a convex function of pdf pairs [165, Theorem 2.7.2, pp. 30], hence the sum of KL divergences (3.16) is a convex function of ρ. If $\rho_1, \rho_2, \ldots, \rho_n \in \Phi(\mathcal{X})$ and $\eta_1, \eta_2, \ldots, \eta_n \in [0, 1]$ such that $\sum_{i=1}^{n} \eta_i = 1$, then $\rho^\dagger = \sum_{i=1}^{n} \eta_i \rho_i \in \Phi(\mathcal{X})$; because
(i) since $\rho_i > 0, \forall \boldsymbol{x} \in \mathcal{X}, \forall i \in \{1, \ldots, n\}$ therefore $\rho^\dagger > 0, \forall \boldsymbol{x} \in \mathcal{X}$; and
(ii) since $\int_{\mathcal{X}} \rho_i d\mu(\boldsymbol{x}) = 1, \forall i \in \{1, \ldots, n\}$ therefore $\int_{\mathcal{X}} \rho^\dagger d\mu(\boldsymbol{x}) = 1$.
Moreover, since $\mathcal{X}$ is a compact set, therefore $\Phi(\mathcal{X})$ is a closed set. Hence $\Phi(\mathcal{X})$ is a closed convex set. Hence, (3.15) is a convex optimization problem.

The gradient of $\sum_{j=1}^{N} D_{\mathrm{KL}}\left(\rho||\mathcal{P}_k^j\right)$ evaluated at $\mathcal{P}_k^{\mathrm{KL}}$ is a constant, i.e.,

$$\frac{d}{d\rho}\sum_{j=1}^{N} D_{\mathrm{KL}}\left(\rho||\mathcal{P}_k^j\right)\Big|_{\rho=\mathcal{P}_k^{\mathrm{KL}}} = N\,\ln\frac{e}{\int_{\mathcal{X}}\prod_{j=1}^{N}\left(\mathcal{P}_k^j\right)^{\frac{1}{N}}\,d\mu(\boldsymbol{x})}\,. \tag{3.18}$$

This indicates that for further minimizing the convex cost function, we have to change the normalizing constant of $\mathcal{P}_k^{\mathrm{KL}}$, which will result in exiting the set $\Phi(\mathcal{X})$. Hence $\mathcal{P}_k^{\mathrm{KL}}$ is the global minimum of the convex cost function (3.15) in the convex set $\Phi(\mathcal{X})$.

Another proof approach involves taking the logarithm, in the KL divergence formula, to the base $c := \left(\int_{\mathcal{X}}\prod_{j=1}^{N}\left(\mathcal{P}_k^j\right)^{\frac{1}{N}}\,d\mu(\boldsymbol{x})\right)$. Then differentiating $\sum_{j=1}^{N} D_{\mathrm{KL}}\left(\rho||\mathcal{P}_k^j\right)$ with respect to ρ gives:

$$\sum_{j=1}^{N}\int_{\mathcal{X}}\left(\log_c(\rho)+1-\log_c(\mathcal{P}_k^j)\right)\,d\mu(\boldsymbol{x}) = 0\,, \tag{3.19}$$

which is minimized by $\mathcal{P}_k^{\mathrm{KL}}$. Hence $\mathcal{P}_k^{\mathrm{KL}}$ is indeed the global minimum of the convex optimization problem in (3.15). ■

Since the KL divergence is the measure of the information lost during the process of combining the pdfs $\mathcal{P}_k^i$ of all agents, the pdf $\mathcal{P}_k^{\mathrm{KL}}$ minimizes the information lost in this combination process. We can now focus on analyzing distributed estimation algorithms that use the LogOP scheme.

3.3 Distributed Bayesian Filtering Algorithm

In this section, we present the main DBF algorithm and its application to special cases.

3.3.1 Main Algorithm

The normalized joint likelihood function $\mathcal{L}_k^C$ (3.7) in the multi-sensor Bayesian filtering algorithm in Section 3.1.2 is also given by:

$$\mathcal{L}_k^C = \frac{\prod_{j=1}^N \mathcal{L}_k^j}{\int_X \prod_{j=1}^N \mathcal{L}_k^j \, d\mu(\bar{\boldsymbol{x}})} = \frac{\left(\mathcal{L}_k^{\mathrm{KL}}\right)^N}{\int_X \left(\mathcal{L}_k^{\mathrm{KL}}\right)^N \, d\mu(\bar{\boldsymbol{x}})}\,, \tag{3.20}$$

$$\text{where } \mathcal{L}_k^{\mathrm{KL}} = \frac{\prod_{j=1}^N \left(\mathcal{L}_k^j\right)^{\frac{1}{N}}}{\int_{\mathcal{X}} \prod_{j=1}^N \left(\mathcal{L}_k^j\right)^{\frac{1}{N}} \, d\mu(\bar{\boldsymbol{x}})}\,. \tag{3.21}$$

We now state the assumption on the time-varying nature of the pdfs $\mathcal{L}_k^i$ (3.5) for all agents.

Assumption 3.4. *(Relatively Bounded Normalized Likelihood Functions)* For any time step $\Delta > 0$, there exists a time-invariant constant $\theta_L > 0$ such that for all agents:

$$e^{-\Delta\theta_L} \leq \frac{\mathcal{L}_k^i(\boldsymbol{x})}{\mathcal{L}_{k-1}^i(\boldsymbol{x})} \leq e^{\Delta\theta_L}\,,\ \forall \boldsymbol{x} \in \mathcal{X},\ \forall k \in \mathbb{N}\,. \tag{3.22}$$

The necessary conditions for satisfying (3.22) are given by:

$$D_{\mathrm{KL}}\left(\mathcal{L}_k^i || \mathcal{L}_{k-1}^i\right) \leq \Delta\theta_L,\ D_{\mathrm{KL}}\left(\mathcal{L}_{k-1}^i || \mathcal{L}_k^i\right) \leq \Delta\theta_L\,.$$

A conservative θ_L always exists because Assumption 3.3 ensures that all pdfs are bounded. □

Note that Assumption 3.4 directly couples the target dynamics and measurement models with the time step of the distributed estimation algorithm. We now state the main result of this section. Let the pdfs $\mathcal{U}_k^i \in \Phi(\mathcal{X})$ and $\mathcal{T}_k^i \in \Phi(\mathcal{X})$ respectively denote the estimated KL-divergence-minimizing pdf and the estimated likelihood function of the i^{th} agent at the k^{th} time instant. We show that the pdf $\mathcal{U}_k^i$ converges to the pdf $\mathcal{L}_k^{\mathrm{KL}}$ (3.21) and the pdf $\mathcal{T}_k^i$ converges to the pdf $\mathcal{L}_k^C$ (3.20).

Theorem 3.4. *Under Assumptions 3.1–3.4, the convergence error between the pdf $\mathcal{T}_k^i$ and the pdf $\mathcal{L}_k^C$ (3.7) is bounded by $\delta \in (0,2)$ after κ^{th} time instant:*

$$\max_{i\in\mathcal{V}} D_{L_1}\left(\mathcal{T}_k^i, \mathcal{L}_k^C\right) \leq \delta\,, \quad \forall k \geq \kappa\,, \tag{3.23}$$

$$\text{where } \mathfrak{D}_1 = 2\ln\left(\max_{\ell,j\in\mathcal{V}} \max_{\boldsymbol{x}\in\mathcal{X}} \frac{\mathcal{L}_1^\ell(\boldsymbol{x})}{\mathcal{L}_1^j(\boldsymbol{x})}\right)\,, \tag{3.24}$$

$$\kappa = \left\lceil \frac{2\mathfrak{b}(N-1)\log\left(\frac{\eta\delta}{\mathfrak{D}_1 2(1+\eta)N\sqrt{N}-\delta}\right)}{\log\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)} \right\rceil + 1\,, \tag{3.25}$$

$\mathfrak{b}$ and γ are defined in Assumption 3.2 and $\eta \in (0,1)$ is a design parameter, if each agent updates its pdfs $\mathcal{U}_k^i$ and $\mathcal{T}_k^i$ using the following fusion rule:

$$\mathcal{U}_k^i = \begin{cases} \mathcal{L}_1^i & \text{if } k=1 \\ \dfrac{\Lambda\,\mathcal{L}_k^i\left(\mathcal{L}_{k-1}^i\right)^{-1}}{\int_{\mathcal{X}} \Lambda\,\mathcal{L}_k^i\left(\mathcal{L}_{k-1}^i\right)^{-1} d\mu(\bar{\boldsymbol{x}})} & \text{if } k\geq 2 \end{cases}\,, \tag{3.26}$$

$$\text{where } \Lambda = \prod_{j\in\mathcal{J}_k^i} (\mathcal{U}_{k-1}^j)^{\mathcal{A}_k[i,j]}\,,$$

$$\mathcal{T}_k^i = \frac{(\mathcal{U}_k^i)^N}{\int_{\mathcal{X}} (\mathcal{U}_k^i)^N d\mu(\bar{\boldsymbol{x}})}\,, \tag{3.27}$$

and the time step Δ has an upper bound of $\Delta_{\max}$:

$$\Delta_{\max} := \frac{\delta\left(1-\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}\right)}{4(1+\eta)\mathfrak{b}N(N-1)\sqrt{N}\theta_L}\,, \tag{3.28}$$

where θ_L is defined in Assumption 3.4. The TV error between the measures induced by the pdfs $\mathcal{T}_k^i$ and $\mathcal{L}_k^C$ is also bounded by $\frac{\delta}{2}$ after κ^{th} time instant, i.e., $\|\mu_{\mathcal{T}_k^i} - \mu_{\mathcal{L}_k^C}\|_{\mathrm{TV}} \leq \frac{\delta}{2}$ for all $k \geq \kappa$.

Moreover, the steady-state convergence error between the pdfs $\mathcal{T}_k^i$ and $\mathcal{L}_k^C$ is bounded by $\frac{\delta}{1+\eta}$, i.e., $\lim_{k\to\infty} \max_{i\in\mathcal{V}} D_{L_1}\left(\mathcal{T}_k^i, \mathcal{L}_k^C\right) \leq \frac{\delta}{1+\eta}$.

Proof: The following functions are defined using Definition 3.1:

$$\mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x}) = \ln\left[\frac{\mathcal{L}_k^{\mathrm{KL}}(\boldsymbol{x})}{\mathcal{L}_k^{\mathrm{KL}}(\boldsymbol{\psi})}\right], \qquad \mathscr{L}_k^{C}(\boldsymbol{x}) = \ln\left[\frac{\mathcal{L}_k^{C}(\boldsymbol{x})}{\mathcal{L}_k^{C}(\boldsymbol{\psi})}\right],$$
$$\mathscr{L}_k^{i}(\boldsymbol{x}) = \ln\left[\frac{\mathcal{L}_k^{i}(\boldsymbol{x})}{\mathcal{L}_k^{i}(\boldsymbol{\psi})}\right], \qquad \mathscr{U}_k^{i}(\boldsymbol{x}) = \ln\left[\frac{\mathcal{U}_k^{i}(\boldsymbol{x})}{\mathcal{U}_k^{i}(\boldsymbol{\psi})}\right],$$
$$\mathscr{T}_k^{i}(\boldsymbol{x}) = \ln\left[\frac{\mathcal{T}_k^{i}(\boldsymbol{x})}{\mathcal{T}_k^{i}(\boldsymbol{\psi})}\right], \qquad \forall i \in \mathcal{V}.$$

Step 1. We first show that the pdf $\mathcal{U}_k^i$ (3.26) converges to the pdf $\mathcal{L}_k^{\mathrm{KL}}$ (3.21). Equation (3.26) can be re-written using these functions as:

$$\mathscr{U}_k^i = \begin{cases} \mathscr{L}_1^i & \text{if } k = 1 \\ \sum_{j=1}^N \mathcal{A}_k[i,j]\mathscr{U}_{k-1}^j + \mathscr{L}_k^i - \mathscr{L}_{k-1}^i & \text{if } k \geq 2 \end{cases}, \tag{3.29}$$

because $\mathcal{A}_k[i,j] = 0$ if $j \notin \mathcal{J}_k^i$. Since $\mathcal{A}_k$ is doubly stochastic, (3.29) satisfies the conservation property:

$$\begin{aligned} \sum_{i=1}^N \mathscr{U}_k^i &= \sum_{i=1}^N \sum_{j=1}^N \mathcal{A}_k[i,j]\mathscr{U}_{k-1}^j + \sum_{i=1}^N \left(\mathscr{L}_k^i - \mathscr{L}_{k-1}^i\right), \\ &= \sum_{i=1}^N \left(\sum_{j=1}^N \mathcal{A}_k[j,i]\right)\mathscr{U}_{k-1}^i + \sum_{i=1}^N \left(\mathscr{L}_k^i - \mathscr{L}_{k-1}^i\right), \\ &= \sum_{i=1}^N \left(\mathscr{U}_1^i - \mathscr{L}_1^i\right) + \sum_{i=1}^N \mathscr{L}_k^i = N\mathscr{L}_k^{\mathrm{KL}}. \end{aligned} \tag{3.30}$$

This shows that if the functions $\mathscr{U}_k^i$ converge towards each other, then they will converge to

the function $\mathscr{L}_k^{\mathrm{KL}}$. For any $\boldsymbol{x} \in \mathcal{X}$, (3.29) can be written in matrix form as follows:

$$\begin{aligned} \mathfrak{U}_k(\boldsymbol{x}) &= \mathcal{A}_k \mathfrak{U}_{k-1}(\boldsymbol{x}) + \mathfrak{L}_k(\boldsymbol{x}) - \mathfrak{L}_{k-1}(\boldsymbol{x}) \,, \\ \text{where } \mathfrak{U}_k(\boldsymbol{x}) &= \begin{bmatrix} \mathscr{U}_k^1(\boldsymbol{x}) & \dots & \mathscr{U}_k^i(\boldsymbol{x}) & \dots & \mathscr{U}_k^N(\boldsymbol{x}) \end{bmatrix}^T \,, \\ \mathfrak{L}_k(\boldsymbol{x}) &= \begin{bmatrix} \mathscr{L}_k^1(\boldsymbol{x}) & \dots & \mathscr{L}_k^i(\boldsymbol{x}) & \dots & \mathscr{L}_k^N(\boldsymbol{x}) \end{bmatrix}^T \,. \end{aligned} \tag{3.31}$$

We define $\mathfrak{L}_k^{\mathrm{KL}}(\boldsymbol{x}) = \mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x})\mathbf{1}$. Therefore, we get:

$$\begin{aligned} \mathfrak{U}_k(\boldsymbol{x}) - \mathfrak{L}_k^{\mathrm{KL}}(\boldsymbol{x}) &= \mathcal{A}_k \left(\mathfrak{U}_{k-1}(\boldsymbol{x}) - \mathfrak{L}_{k-1}^{\mathrm{KL}}(\boldsymbol{x}) \right) + \boldsymbol{\Omega}_{k,k}(\boldsymbol{x}) \,, \\ \text{where } \boldsymbol{\Omega}_{k,k}(\boldsymbol{x}) &= \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) \left(\mathfrak{L}_k(\boldsymbol{x}) - \mathfrak{L}_{k-1}(\boldsymbol{x}) \right) \,, \end{aligned}$$

and $\mathcal{A}_k \mathfrak{L}_{k-1}^{\mathrm{KL}}(\boldsymbol{x}) = \mathfrak{L}_{k-1}^{\mathrm{KL}}(\boldsymbol{x})$ and $\mathfrak{L}_k^{\mathrm{KL}}(\boldsymbol{x}) = \frac{\mathbf{1}\mathbf{1}^T}{N} \mathfrak{L}_k(\boldsymbol{x})$. The overall evolution of the error vector $\boldsymbol{e}_k(\boldsymbol{x}) = (\mathfrak{U}_k(\boldsymbol{x}) - \mathfrak{L}_k^{\mathrm{KL}}(\boldsymbol{x}))$ after $\mathfrak{b}$ time instants is given by:

$$\begin{aligned} \boldsymbol{e}_{k+\mathfrak{b}-1}(\boldsymbol{x}) &= \mathcal{A}_{k,k+\mathfrak{b}-1} \boldsymbol{e}_{k-1}(\boldsymbol{x}) + \boldsymbol{\Omega}_{k,k+\mathfrak{b}-1}(\boldsymbol{x}) \,, \\ \text{where } \mathcal{A}_{k,k+\mathfrak{b}-1} &= \left(\prod_{\tau=k}^{k+\mathfrak{b}-1} \mathcal{A}_\tau \right) \,, \\ \boldsymbol{\Omega}_{k,k+\mathfrak{b}-1}(\boldsymbol{x}) &= \begin{cases} \sum_{\tau=k}^{k+\mathfrak{b}-2} \left(\mathcal{A}_{\tau+1,k+\mathfrak{b}-1} \boldsymbol{\Omega}_{\tau,\tau}(\boldsymbol{x}) \right) + \boldsymbol{\Omega}_{k+\mathfrak{b}-1,k+\mathfrak{b}-1}(\boldsymbol{x}) & \text{if } \mathfrak{b} \geq 2 \\ \boldsymbol{\Omega}_{k,k}(\boldsymbol{x}) & \text{if } \mathfrak{b} = 1 \end{cases} . \end{aligned} \tag{3.32}$$

It follows from Assumption 3.2 that the matrix $\mathcal{A}_{k,k+\mathfrak{b}-1}$ is irreducible. Therefore, the matrix $\mathcal{A}_{k,k+\mathfrak{b}-1}$ is primitive [141, Lemma 8.5.4, pp. 516] and $\lambda_{N-1}\left(\mathcal{A}_{k,k+\mathfrak{b}-1}\right) < 1$, where λ_{N-1} denotes the second largest eigenvalue of the matrix.

Note that $\mathbf{1}^T \boldsymbol{e}_k(\boldsymbol{x}) = 0$ because of (4.24) and $\mathbf{1}^T \boldsymbol{\Omega}_{k,k+\mathfrak{b}-1}(\boldsymbol{x}) = 0$ because $\mathbf{1}^T \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N} \right) = 0$. Therefore, we investigate the convergence of $\boldsymbol{e}_k(\boldsymbol{x})$ along all directions that are orthogonal to $\mathbf{1}^T$. Let $V_{\mathrm{tr}} = \left[\frac{1}{\sqrt{N}} \mathbf{1}, V_{\mathrm{s}} \right]$ be the orthonormal matrix of eigenvectors of the symmetric

primitive matrix $\mathcal{A}_{1,\mathfrak{b}}^T\mathcal{A}_{1,\mathfrak{b}}$. By spectral decomposition [12], we get:

$$V_{\text{tr}}^T\mathcal{A}_{1,\mathfrak{b}}^T\mathcal{A}_{1,\mathfrak{b}}V_{\text{tr}} = \begin{bmatrix} 1 & \mathbf{0}^{1\times(N-1)} \\ \mathbf{0}^{(N-1)\times 1} & V_{\text{s}}^T\mathcal{A}_{1,\mathfrak{b}}^T\mathcal{A}_{1,\mathfrak{b}}V_{\text{s}} \end{bmatrix},$$

where $\frac{1}{N}\mathbf{1}^T\mathcal{A}_{1,\mathfrak{b}}^T\mathcal{A}_{1,\mathfrak{b}}\mathbf{1} = 1$, $\frac{1}{\sqrt{N}}\mathbf{1}^T\mathcal{A}_{1,\mathfrak{b}}^T\mathcal{A}_{1,\mathfrak{b}}V_{\text{s}} = \mathbf{0}^{1\times(N-1)}$, and $V_{\text{s}}^T\mathcal{A}_{1,\mathfrak{b}}^T\mathcal{A}_{1,\mathfrak{b}}\mathbf{1}\frac{1}{\sqrt{N}} = \mathbf{0}^{(N-1)\times 1}$ are used. Since the eigenvectors are orthonormal, we have $V_{\text{s}}V_{\text{s}}^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T = \mathbf{I}$. Left-multiplying (3.32) with V_{s}^T gives:

$$\begin{aligned} V_{\text{s}}^T\boldsymbol{e}_{k+\mathfrak{b}-1}(\boldsymbol{x}) &= V_{\text{s}}^T\mathcal{A}_{k,k+\mathfrak{b}-1}\left(V_{\text{s}}V_{\text{s}}^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\boldsymbol{e}_{k-1}(\boldsymbol{x}) + V_{\text{s}}^T\boldsymbol{\Omega}_{k,k+\mathfrak{b}-1}(\boldsymbol{x}), \\ &= V_{\text{s}}^T\mathcal{A}_{k,k+\mathfrak{b}-1}V_{\text{s}}V_{\text{s}}^T\boldsymbol{e}_{k-1}(\boldsymbol{x}) + V_{\text{s}}^T\boldsymbol{\Omega}_{k,k+\mathfrak{b}-1}(\boldsymbol{x}). \end{aligned} \tag{3.33}$$

We first investigate the stability of this system without the disturbance term $V_{\text{s}}^T\boldsymbol{\Omega}_{k,k+\mathfrak{b}-1}(\boldsymbol{x})$. Let $\|V_{\text{s}}^T\boldsymbol{e}_{k+\mathfrak{b}-1}(\boldsymbol{x})\|_2$ be a candidate Lyapunov function for this system. Therefore, we get:

$$\begin{aligned} \|V_{\text{s}}^T\boldsymbol{e}_{k+\mathfrak{b}-1}(\boldsymbol{x})\|_2 &= \|V_{\text{s}}^T\mathcal{A}_{k,k+\mathfrak{b}-1}V_{\text{s}}V_{\text{s}}^T\boldsymbol{e}_{k-1}(\boldsymbol{x})\|_2 \\ &\leq \|V_{\text{s}}^T\mathcal{A}_{k,k+\mathfrak{b}-1}V_{\text{s}}\|_2\|V_{\text{s}}^T\boldsymbol{e}_{k-1}(\boldsymbol{x})\|_2 \\ &\leq \sigma_{\max}(\mathcal{A}_{k,k+\mathfrak{b}-1}V_{\text{s}})\|V_{\text{s}}^T\boldsymbol{e}_{k-1}(\boldsymbol{x})\|_2, \end{aligned}$$

where $\sigma_{\max}$ denotes the largest singular value of the matrix. Since the symmetric matrix $\left(\mathcal{A}_{k,k+\mathfrak{b}-1}^T\mathcal{A}_{k,k+\mathfrak{b}-1}\right)$ is primitive, we get using spectral decomposition:

$$\left[\frac{1}{\sqrt{N}}\mathbf{1},\, V_{\text{s},k,k+\mathfrak{b}-1}\right]^T\mathcal{A}_{k,k+\mathfrak{b}-1}^T\mathcal{A}_{k,k+\mathfrak{b}-1}\left[\frac{1}{\sqrt{N}}\mathbf{1},\, V_{\text{s},k,k+\mathfrak{b}-1}\right] = \begin{bmatrix} 1 & \mathbf{0}^{1\times(N-1)} \\ \mathbf{0}^{(N-1)\times 1} & \mathcal{D}_{k,k+\mathfrak{b}-1} \end{bmatrix},$$

where $\mathcal{D}_{k,k+\mathfrak{b}-1} = V_{\text{s},k,k+\mathfrak{b}-1}^T\left(\mathcal{A}_{k,k+\mathfrak{b}-1}^T\mathcal{A}_{k,k+\mathfrak{b}-1}\right)V_{\text{s},k,k+\mathfrak{b}-1}$,

where $V_{\text{s},k,k+\mathfrak{b}-1}^T$ is orthonormal and also orthogonal to $\mathbf{1}^T$ and the diagonal matrix $\mathcal{D}_{k,k+\mathfrak{b}-1}$ contains all the other eigenvalues other than 1. Since both $V_{\text{s},k,k+\mathfrak{b}-1}^T$ and V_{s}^T are orthogonal

to $\mathbf{1}^T$, there exists a constant transformation matrix $\mathfrak{I}$ such that $V_{\mathrm{s}} = V_{\mathrm{s},k,k+\mathfrak{b}-1}\mathfrak{I}$. The existence of $\mathfrak{I}$ is guaranteed since both $V^T_{\mathrm{s},k,k+\mathfrak{b}-1}$ and V_{s}^T are orthonormal vectors spanning the subspace that is orthogonal to $\mathbf{1}^T$, hence a linear transformation matrix between them must exist. We know $\left(\mathfrak{I}^T\mathfrak{I}\right) = \mathbf{I}$ because $\mathbf{I} = V_{\mathrm{s}}^T V_{\mathrm{s}} = \mathfrak{I}^T V^T_{\mathrm{s},k,k+\mathfrak{b}-1} V_{\mathrm{s},k,k+\mathfrak{b}-1}\mathfrak{I} = \mathfrak{I}^T\mathfrak{I}$. Therefore, we get from [141, Corollary 1.3.4, pp. 45] that:

$$\begin{aligned}\lambda_{\max}\left(V_{\mathrm{s}}^T \mathcal{A}^T_{k,k+\mathfrak{b}-1}\mathcal{A}_{k,k+\mathfrak{b}-1}V_{\mathrm{s}}\right) &= \lambda_{\max}\left(V^T_{\mathrm{s},k,k+\mathfrak{b}-1}\left(\mathcal{A}^T_{k,k+\mathfrak{b}-1}\mathcal{A}_{k,k+\mathfrak{b}-1}\right)V_{\mathrm{s},k,k+\mathfrak{b}-1}\right)\\ &= \lambda_{N-1}\left(\mathcal{A}^T_{k,k+\mathfrak{b}-1}\mathcal{A}_{k,k+\mathfrak{b}-1}\right) < 1.\end{aligned}$$

Hence, $\sigma_{\max}(\mathcal{A}_{k,k+\mathfrak{b}-1}V_{\mathrm{s}}) = \sigma_{N-1}(\mathcal{A}_{k,k+\mathfrak{b}-1}) < 1$, where σ_{N-1} denotes the second largest singular value of the matrix. Therefore, the error vector $V_{\mathrm{s}}^T\boldsymbol{e}_k(\boldsymbol{x})$ is globally exponentially stable in absence of the disturbance term.

The second largest eigenvalue of a matrix is upper bounded by the coefficient of ergodicity of that matrix [135, pp. 137]. Since the matrix $\mathcal{A}_{k,k+\mathfrak{b}-1}$ is irreducible, the matrix $\mathcal{A}_{k,k+\mathfrak{b}(N-1)-1}$ is a positive matrix because the maximum path length between any two agents is less than or equal to $\mathfrak{b}(N-1)$. Since the positive elements in the matrix $\mathcal{A}^T_{k,k+\mathfrak{b}(N-1)-1}\mathcal{A}_{k,k+\mathfrak{b}(N-1)-1}$ is lower bounded by $\gamma^{2\mathfrak{b}(N-1)}$, we have $\sigma_{N-1}(\mathcal{A}_{k,k+\mathfrak{b}(N-1)-1}) \leq \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}$ [121]. Moreover, it follows from Assumption 3.4 that $\|\mathfrak{L}_k(\boldsymbol{x}) - \mathfrak{L}_{k-1}(\boldsymbol{x})\|_2 \leq$

$2\sqrt{N}\Delta\theta_L$. Therefore, we have:

$$
\begin{aligned}
&\|V_{\mathrm{s}}^T \boldsymbol{\Omega}_{k,k+\mathfrak{b}(N-1)-1}(\boldsymbol{x})\|_2 \\
&= \left\| V_{\mathrm{s}}^T \left(\sum_{\tau=k}^{k+\mathfrak{b}(N-1)-2} \left(\mathcal{A}_{\tau+1,k+\mathfrak{b}(N-1)-1} \boldsymbol{\Omega}_{\tau,\tau}(\boldsymbol{x})\right) + \boldsymbol{\Omega}_{k+\mathfrak{b}(N-1)-1,k+\mathfrak{b}(N-1)-1}(\boldsymbol{x}) \right) \right\|_2 \\
&= \left\| \sum_{\tau=k}^{k+\mathfrak{b}(N-1)-2} \left(V_{\mathrm{s}}^T \mathcal{A}_{\tau+1,k+\mathfrak{b}(N-1)-1} \left(\mathfrak{L}_\tau(\boldsymbol{x}) - \mathfrak{L}_{\tau-1}(\boldsymbol{x})\right)\right) + V_{\mathrm{s}}^T \left(\mathfrak{L}_{k+\mathfrak{b}(N-1)-1}(\boldsymbol{x}) - \mathfrak{L}_{k+\mathfrak{b}(N-1)-2}(\boldsymbol{x})\right) \right\|_2 \\
&\leq \left(\sum_{\tau=k}^{k+\mathfrak{b}(N-1)-2} \|V_{\mathrm{s}}^T\|_2 \|\mathcal{A}_{\tau+1,k+\mathfrak{b}(N-1)-1}\|_2 \|\mathfrak{L}_\tau(\boldsymbol{x}) - \mathfrak{L}_{\tau-1}(\boldsymbol{x})\|_2 \right) \\
&\quad + \|V_{\mathrm{s}}^T\|_2 \|\mathfrak{L}_{k+\mathfrak{b}(N-1)-1}(\boldsymbol{x}) - \mathfrak{L}_{k+\mathfrak{b}(N-1)-2}(\boldsymbol{x})\|_2 \\
&\leq 2\mathfrak{b}(N-1)\sqrt{N}\Delta\theta_L
\end{aligned}
$$

Hence, in the presence of the disturbance term, we get:

$$
\begin{aligned}
\|V_{\mathrm{s}}^T \boldsymbol{e}_{k+\mathfrak{b}(N-1)-1}(\boldsymbol{x})\|_2 &= \left\|V_{\mathrm{s}}^T \mathcal{A}_{k,k+\mathfrak{b}(N-1)-1} V_{\mathrm{s}} V_{\mathrm{s}}^T \boldsymbol{e}_{k-1}(\boldsymbol{x}) + V_{\mathrm{s}}^T \boldsymbol{\Omega}_{k,k+\mathfrak{b}(N-1)-1}(\boldsymbol{x})\right\|_2 \\
&\leq \left\|V_{\mathrm{s}}^T \mathcal{A}_{k,k+\mathfrak{b}(N-1)-1} V_{\mathrm{s}} V_{\mathrm{s}}^T \boldsymbol{e}_{k-1}(\boldsymbol{x})\right\|_2 + \left\|V_{\mathrm{s}}^T \boldsymbol{\Omega}_{k,k+\mathfrak{b}(N-1)-1}(\boldsymbol{x})\right\|_2 \\
&\leq \sigma_{N-1}(\mathcal{A}_{k,k+\mathfrak{b}(N-1)-1}) \|V_{\mathrm{s}}^T \boldsymbol{e}_{k-1}(\boldsymbol{x})\|_2 + \|V_{\mathrm{s}}^T \boldsymbol{\Omega}_{k,k+\mathfrak{b}(N-1)-1}(\boldsymbol{x})\|_2 \\
&\leq \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}} \|V_{\mathrm{s}}^T \boldsymbol{e}_{k-1}(\boldsymbol{x})\|_2 + 2\mathfrak{b}(N-1)\sqrt{N}\Delta\theta_L \,.
\end{aligned}
$$

Using the discrete Gronwall lemma [166, pp. 9] we obtain:

$$
\begin{aligned}
\|V_{\mathrm{s}}^T \boldsymbol{e}_k(\boldsymbol{x})\|_2 \leq \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}\left\lfloor \frac{k-1}{\mathfrak{b}(N-1)} \right\rfloor} \|V_{\mathrm{s}}^T \boldsymbol{e}_1(\boldsymbol{x})\|_2 \\
+ \frac{1 - \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}\left\lfloor \frac{k-1}{\mathfrak{b}(N-1)} \right\rfloor}}{1 - \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}} 2\mathfrak{b}(N-1)\sqrt{N}\Delta\theta_L \,.
\end{aligned}
$$

For a given η, we define Ξ as:

$$\Xi = \frac{(1+\eta)2\mathfrak{b}(N-1)\sqrt{N}\Delta\theta_L}{1-(1-N\gamma^{2\mathfrak{b}(N-1)})^{\frac{1}{2}}}\,. \tag{3.34}$$

Then, we calculate κ as follows:

$$\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}\left\lfloor\frac{\kappa-1}{\mathfrak{b}(N-1)}\right\rfloor}\sqrt{N}\mathfrak{D}_1 + \frac{1-\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}\left\lfloor\frac{\kappa-1}{\mathfrak{b}(N-1)}\right\rfloor}}{1-(1-N\gamma^{2\mathfrak{b}(N-1)})^{\frac{1}{2}}}2\mathfrak{b}(N-1)\sqrt{N}\Delta\theta_L = \Xi\,.$$

It follows that for all $\boldsymbol{x}\in\mathcal{X}$:

$$\max_{i\in\mathcal{V}}|\mathscr{U}_k^i(\boldsymbol{x})-\mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x})|\le\Xi\,,\quad\forall k\ge\kappa\,, \tag{3.35}$$

$$\lim_{k\to\infty}\max_{i\in\mathcal{V}}|\mathscr{U}_k^i(\boldsymbol{x})-\mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x})|\le\frac{\Xi}{1+\eta}\,, \tag{3.36}$$

$$\text{where }\kappa=\left\lceil\frac{2\mathfrak{b}(N-1)\log\left(\mathfrak{D}_2\right)}{\log\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)}\right\rceil+1\,, \tag{3.37}$$

$$\mathfrak{D}_2=\frac{\eta}{\frac{\mathfrak{D}_1\left(1-\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}\right)}{2\mathfrak{b}(N-1)\Delta\theta_L}-1}\,.$$

and $\mathfrak{D}_1$ is defined in (3.24). Thus, the function $\mathscr{U}_k^i$ converges to the function $\mathscr{L}_k^{\mathrm{KL}}$ with convergence error Ξ. Note that our proof technique is substantially different from that in [163].

Step 2. We now show that the pdf $\mathcal{T}_k^i$ (3.27) converges to the pdf $\mathcal{L}_k^C$ (3.20). Equations (3.20) and (3.27) can be re-written as:

$$\mathscr{L}_k^C(\boldsymbol{x})=N\mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x})\,,\quad\mathscr{T}_k^i(\boldsymbol{x})=N\mathscr{U}_k^i(\boldsymbol{x})\,,\,\forall i\in\mathcal{V}\,.$$

Therefore, it follows from (3.35) that:

$$\max_{i\in\mathcal{V}}|\mathscr{T}_k^i(\boldsymbol{x})-\mathscr{L}_k^C(\boldsymbol{x})|\le N\Xi\,,\quad\forall k\ge\kappa\,. \tag{3.38}$$

Thus, the function $\mathscr{T}_k^i$ converges to the function $\mathscr{L}_k^C$ with convergence error $N\Xi$. We now focus on the convergence error between the pdfs $\mathcal{T}_k^i$ and $\mathcal{L}_k^C$. We have:

$$\max_{i\in\mathcal{V}} \left| \ln\left[\frac{\mathcal{T}_k^i(\boldsymbol{x})}{\mathcal{T}_k^i(\boldsymbol{\psi})}\right] - \ln\left[\frac{\mathcal{L}_k^C(\boldsymbol{x})}{\mathcal{L}_k^C(\boldsymbol{\psi})}\right] \right| \leq N\Xi\,,\ \forall k \geq \kappa\,.$$

We select $\boldsymbol{\psi} \in \mathcal{X}$ such that $\mathcal{T}_k^i(\boldsymbol{\psi}) = \mathcal{L}_k^C(\boldsymbol{\psi})$. Using steps similar to that in the proof of Lemma 3.1, we can show that such a $\boldsymbol{\psi}$ always exists. Therefore,

$$\begin{aligned}
\max_{i\in\mathcal{V}} \left| \ln\left[\frac{\mathcal{T}_k^i(\boldsymbol{x})}{\mathcal{L}_k^C(\boldsymbol{x})}\right] \right| &\leq N\Xi\,,\ \forall k \geq \kappa\,,\\
e^{-N\Xi} \leq \max_{i\in\mathcal{V}} \left(\frac{\mathcal{T}_k^i(\boldsymbol{x})}{\mathcal{L}_k^C(\boldsymbol{x})}\right) &\leq e^{N\Xi}\,,\ \forall k \geq \kappa\,.
\end{aligned}$$

Since $2N\Xi < \delta < 2$, we know $1 - e^{-N\Xi} \leq e^{N\Xi} - 1 \leq 2N\Xi$. Hence we get:

$$\max_{i\in\mathcal{V}} \left|\mathcal{T}_k^i(\boldsymbol{x}) - \mathcal{L}_k^C(\boldsymbol{x})\right| \leq 2N\Xi\mathcal{L}_k^C(\boldsymbol{x})\,,\ \forall k \geq \kappa\,.$$

Since $\boldsymbol{x} \in \mathcal{X}$ can be any point, therefore:

$$\begin{aligned}
\max_{i\in\mathcal{V}} D_{L_1}\left(\mathcal{T}_k^i, \mathcal{L}_k^C\right) &= \max_{i\in\mathcal{V}} \int_{\mathcal{X}} \left|\mathcal{T}_k^i - \mathcal{L}_k^C\right|\, d\mu(\boldsymbol{x})\\
&\leq 2N\Xi \int_{\mathcal{X}} \mathcal{L}_k^C\, d\mu(\boldsymbol{x}) = 2N\Xi\,,\ \forall k \geq \kappa\,. \qquad (3.39)
\end{aligned}$$

Hence the convergence error between the pdfs is bounded by $2N\Xi$. Since we want $2N\Xi \leq \delta$, we get from (3.34) that we should ensure $\Delta \leq \Delta_{\max}$ (3.28). We also get κ (3.25) by substituting $\Delta_{\max}$ in (3.37). Finally, the constraint on TV error follows from Lemma 3.2. ■

Theorem 3.4 explicitly bounds the time step Δ of the distributed estimation algorithm with the time-scale of the target dynamics using Assumption 3.4. Moreover, after κ time instants, each agent's estimated likelihood function $\mathcal{T}_k^i$ converges to an error ball centered on the normalized joint likelihood function $\mathcal{L}_k^C$ used in the multi-sensor Bayesian filtering

algorithm (3.6). Therefore, the i^{th} agent uses its estimated likelihood function $\mathcal{T}_k^i$ to compute the posterior pdf $\mathcal{W}_k^i$ during the update step of the Bayesian filtering algorithm:

$$\mathcal{W}_k^i = p(\boldsymbol{x}_{k|k}) = \frac{\mathcal{T}_k^i \, \mathcal{S}_k^i}{\int_{\mathcal{X}} \mathcal{T}_k^i \, \mathcal{S}_k^i \, d\mu(\bar{\boldsymbol{x}})} \,. \tag{3.40}$$

The pseudo-code of the DBF algorithm is given in Algorithm 6, whose steps are shown in Fig. 3.1.

1. (i^{th} agent's steps at k^{th} time instant)
2. Compute prior pdf $\mathcal{S}_k^i = p(\boldsymbol{x}_{k|k-1})$ using (3.3)
3. Obtain local measurement $\boldsymbol{y}_k^i$
4. Compute normalized likelihood function $\mathcal{L}_k^i$ (3.5)
5. Receive pdfs $\mathcal{U}_{k-1}^j$ from agents $j \in \mathcal{J}_k^i$
6. Compute pdfs $\mathcal{U}_k^i$ and $\mathcal{T}_k^i$ using (3.26)–(3.27)
7. Compute posterior pdf $\mathcal{W}_k^i = p(\boldsymbol{x}_{k|k})$ using (3.40)

Algorithm 6: Distributed Bayesian Filtering Algorithm

Remark 3.1. A key advantage of the DBF algorithm is that it does not require all the sensors to observe the target. If an agent does not observe the target, then it sets its normalized likelihood function as the uniform distribution, i.e., $\mathcal{L}_k^i(\boldsymbol{x}) = \frac{1}{\int_{\mathcal{X}} d\mu(\boldsymbol{x})}$. Then this agent's likelihood function does not influence the joint likelihood function and the final converged pdf because of the geometric nature of the fusion rule.

It should be noted that the effectiveness of the DBF algorithm is predicated on Assumption 3.4. Moreover, the upper bound on the time step $\Delta_{\max}$ (3.28) decreases with increasing number of agents. □

In the following corollary, we provide sharper bounds for κ (3.25) and $\Delta_{\max}$ (3.28) for a special case of the communication network topology.

Corollary 3.5. *In Theorem 3.4, if the communication network topology is strongly connected*

and time-invariant, then κ *(3.25) and* $\Delta_{\max}$ *(3.28) are given by:*

$$\kappa = \left\lceil \frac{\log\left(\frac{\eta\delta}{2\mathfrak{D}_1(1+\eta)N\sqrt{N}-\delta}\right)}{\log\left(\sigma_{N-1}(\mathcal{A})\right)} \right\rceil + 1\,, \tag{3.41}$$

$$\Delta_{\max} = \frac{\delta\left(1-\sigma_{N-1}(\mathcal{A})\right)}{4(1+\eta)N\sqrt{N}\theta_L}\,, \tag{3.42}$$

where $\mathcal{A}$ *is the time-invariant adjacency matrix.*

Proof: In this case, in the presence of the disturbance term, we get:

$$\|V_{\mathrm{s}}^T \boldsymbol{e}_k(\boldsymbol{x})\|_2 \leq \sigma_{N-1}(\mathcal{A})\|V_{\mathrm{s}}^T \boldsymbol{e}_{k-1}(\boldsymbol{x})\|_2 + 2\sqrt{N}\Delta\theta_L\,.$$

Using the discrete Gronwall lemma [166, pp. 9] we obtain:

$$\|V_{\mathrm{s}}^T \boldsymbol{e}_k(\boldsymbol{x})\|_2 \leq \left(\sigma_{N-1}(\mathcal{A})\right)^{k-1}\|V_{\mathrm{s}}^T \boldsymbol{e}_1(\boldsymbol{x})\|_2 + \frac{1-\left(\sigma_{N-1}(\mathcal{A})\right)^{k-1}}{1-\sigma_{N-1}(\mathcal{A})} 2\sqrt{N}\Delta\theta_L\,. \tag{3.43}$$

Hence, we get $\max_{i\in\mathcal{V}} |\mathscr{U}_k^i(\boldsymbol{x}) - \mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x})| \leq \Xi$ for all $k \geq \kappa$, where

$$\Xi = \frac{(1+\eta)2\sqrt{N}\Delta\theta_L}{1-\sigma_{N-1}(\mathcal{A})}\,,$$

$$\kappa = \left\lceil \frac{\log\left(\frac{\eta 2\Delta\theta_L}{\mathfrak{D}_1(1-\sigma_{N-1}(\mathcal{A}))-2\Delta\theta_L}\right)}{\log\left(\sigma_{N-1}(\mathcal{A})\right)} \right\rceil + 1\,.$$

We get (3.42) using $2N\Xi \leq \delta$ and (3.41) by substituting $\Delta_{\max}$ into κ. ■

3.3.2 Robustness Analysis

In this section, we study the robustness of the DBF algorithm. In order to implement the DBF algorithm, the agents need to communicate their estimated pdfs with their neighboring agents (see line 5 in Algorithm 6). The information theoretic approach for communicating pdfs is discussed in [167]. If particle filters are used to implement the Bayesian filter and

combine the pdfs [160, 168], then the resampled particles represent the agent's estimated pdf. Hence communicating pdfs is equivalent to transmitting these resampled particles. Another approach involves approximating the pdf by a weighted sum of Gaussian pdfs [169, pp. 213] and then transmitting this approximate distribution. Several techniques for estimating the Gaussian parameters are discussed in the Gaussian mixture model literature [168]–[171].

Each agent can misrepresent its normalized likelihood function $\mathcal{L}_k^i$ due to modeling errors. Let the pdf $\hat{\mathcal{L}}_k^i \in \Phi(\mathcal{X})$ represent the normalized likelihood function that is corrupted with modeling errors. Let the pdfs $\bar{\mathcal{U}}_k^i \in \Phi(\mathcal{X})$ and $\bar{\mathcal{T}}_k^i \in \Phi(\mathcal{X})$ respectively denote the corrupted versions of the pdfs $\mathcal{U}_k^i$ and $\mathcal{T}_k^i$. We first state the assumptions on these modeling errors and then state the main result of this section.

Assumption 3.5. *(Relatively Bounded Modeling Errors)* There exists a time-invariant constant $\varepsilon_L \geq 0$ such that for all agents $i \in \mathcal{V}$:

$$e^{-\varepsilon_L} \leq \frac{\hat{\mathcal{L}}_k^i(\boldsymbol{x})}{\mathcal{L}_k^i(\boldsymbol{x})} \leq e^{\varepsilon_L}\,, \qquad \forall \boldsymbol{x} \in \mathcal{X},\, \forall k \in \mathbb{N}\,. \tag{3.44}$$

□

Theorem 3.6. *Under Assumptions 3.1–3.5, the convergence error between the pdf $\bar{\mathcal{T}}_k^i$ and the pdf $\mathcal{L}_k^C$ (3.7) is bounded by $\delta \in (0, 2)$ after κ^{th} time instant:*

$$\max_{i \in \mathcal{V}} D_{L_1}\left(\bar{\mathcal{T}}_k^i, \mathcal{L}_k^C\right) \leq \delta\,, \quad \forall k \geq \kappa\,, \tag{3.45}$$

$$\text{where } \kappa = \left\lceil \frac{2\mathfrak{b}(N-1)\log\left(\mathfrak{D}_3\right)}{\log\left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)} \right\rceil + 1\,, \tag{3.46}$$

$$\mathfrak{D}_3 = \frac{\eta}{\frac{(\mathfrak{D}_1 + 2\varepsilon_L)2N(1+\eta)\sqrt{N}}{\left(\delta - 4N\sqrt{N}\varepsilon_L\right)} - 1}\,, \tag{3.47}$$

and ε_L is defined in Assumption 3.5, if each agent updates its pdfs $\bar{\mathcal{U}}_k^i$ and $\bar{\mathcal{T}}_k^i$ using the

following fusion rule:

$$\bar{\mathcal{U}}_k^i(\boldsymbol{x}) = \begin{cases} \hat{\mathcal{L}}_1^i & \text{if } k = 1 \\ \dfrac{\bar{\Lambda}\,\hat{\mathcal{L}}_k^i \left(\hat{\mathcal{L}}_{k-1}^i\right)^{-1}}{\int_{\mathcal{X}} \bar{\Lambda}\hat{\mathcal{L}}_k^i \left(\hat{\mathcal{L}}_{k-1}^i\right)^{-1} d\mu(\bar{\boldsymbol{x}})} & \text{if } k \geq 2 \end{cases}, \tag{3.48}$$

$$\text{where } \bar{\Lambda} = \prod_{j \in \mathcal{J}_k^i} (\bar{\mathcal{U}}_{k-1}^j)^{\mathcal{A}_k[i,j]},$$

$$\bar{\mathcal{T}}_k^i(\boldsymbol{x}) = \frac{(\bar{\mathcal{U}}_k^i)^N}{\int_{\mathcal{X}} (\bar{\mathcal{U}}_k^i)^N d\mu(\bar{\boldsymbol{x}})}, \tag{3.49}$$

and the time step Δ has an upper bound of $\Delta_{\max}$:

$$\Delta_{\max} := \frac{\left(\delta - 4N\sqrt{N}\varepsilon_L\right)\left(1 - \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}\right)}{4(1+\eta)\mathfrak{b}N(N-1)\sqrt{N}\theta_L} - \frac{2\varepsilon_L}{\theta_L}. \tag{3.50}$$

The TV error between the measures induced by the pdfs $\bar{\mathcal{T}}_k^i$ and $\mathcal{L}_k^C$ is also bounded by $\frac{\delta}{2}$ after κ^{th} time instant.

Proof: The following functions are defined using Definition 3.1:

$$\bar{\mathscr{U}}_k^i(\boldsymbol{x}) = \ln\left[\frac{\bar{\mathcal{U}}_k^i(\boldsymbol{x})}{\bar{\mathcal{U}}_k^i(\boldsymbol{\psi})}\right], \qquad \hat{\mathscr{L}}_k^i(\boldsymbol{x}) = \ln\left[\frac{\hat{\mathcal{L}}_k^i(\boldsymbol{x})}{\hat{\mathcal{L}}_k^i(\boldsymbol{\psi})}\right],$$

$$\bar{\mathscr{T}}_k^i(\boldsymbol{x}) = \ln\left[\frac{\bar{\mathcal{T}}_k^i(\boldsymbol{x})}{\bar{\mathcal{T}}_k^i(\boldsymbol{\psi})}\right], \qquad \forall i \in \mathcal{V}.$$

Equation (3.48) can be re-written using these functions as:

$$\bar{\mathscr{U}}_k^i = \begin{cases} \hat{\mathscr{L}}_1^i & \text{if } k = 1 \\ \sum_{j=1}^N \mathcal{A}_k[i,j]\bar{\mathscr{U}}_{k-1}^j + \hat{\mathscr{L}}_k^i - \hat{\mathscr{L}}_{k-1}^i & \text{if } k \geq 2 \end{cases}, \tag{3.51}$$

and it satisfies the conservation property $\sum_{i=1}^N \bar{\mathscr{U}}_k^i = \sum_{i=1}^N \hat{\mathscr{L}}_k^i = N\hat{\mathscr{L}}_k^{\mathrm{KL}}$. For any $\boldsymbol{x} \in \mathcal{X}$,

(3.51) can be written in matrix form as follows:

$$\begin{aligned}
&\bar{\mathfrak{U}}_k(\boldsymbol{x}) - \hat{\mathfrak{L}}_k^{\mathrm{KL}}(\boldsymbol{x}) = \mathcal{A}_k \left(\bar{\mathfrak{U}}_{k-1}(\boldsymbol{x}) - \hat{\mathfrak{L}}_{k-1}^{\mathrm{KL}}(\boldsymbol{x})\right) + \hat{\boldsymbol{\Omega}}_{k,k}(\boldsymbol{x})\,,\\
&\text{where } \hat{\boldsymbol{\Omega}}_{k,k}(\boldsymbol{x}) = \left(\mathbf{I} - \frac{\mathbf{1}\mathbf{1}^T}{N}\right)\left(\hat{\mathfrak{L}}_k(\boldsymbol{x}) - \hat{\mathfrak{L}}_{k-1}(\boldsymbol{x})\right)\,,\\
&\bar{\mathfrak{U}}_k = \left[\bar{\mathscr{U}}_k^1 \ldots \bar{\mathscr{U}}_k^N\right]^T\,,\ \hat{\mathfrak{L}}_k = \left[\hat{\mathscr{L}}_k^1 \ldots \hat{\mathscr{L}}_k^N\right]^T\,.
\end{aligned}$$

It follows from Assumptions 3.4 and 3.5 that $\|\hat{\mathfrak{L}}_k(\boldsymbol{x}) - \hat{\mathfrak{L}}_{k-1}(\boldsymbol{x})\|_2 \leq 2\sqrt{N}\Delta\theta_L + 4\sqrt{N}\varepsilon_L$. We define the error vector $\bar{\boldsymbol{e}}_k(\boldsymbol{x}) = (\bar{\mathfrak{U}}_k(\boldsymbol{x}) - \hat{\mathfrak{L}}_k^{\mathrm{KL}}(\boldsymbol{x}))$. Using steps similar to those in the proof of Theorem 3.4, we get:

$$\begin{aligned}
\|V_{\mathrm{s}}^T\bar{\boldsymbol{e}}_k(\boldsymbol{x})\|_2 \leq{}& \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}\left\lfloor\frac{k-1}{\mathfrak{b}(N-1)}\right\rfloor}\|V_{\mathrm{s}}^T\bar{\boldsymbol{e}}_1(\boldsymbol{x})\|_2\\
&+ \frac{1 - \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}\left\lfloor\frac{k-1}{\mathfrak{b}(N-1)}\right\rfloor}}{1 - \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}} 2\mathfrak{b}(N-1)\sqrt{N}(\Delta\theta_L + 2\varepsilon_L)\,.
\end{aligned}$$

Since $\|\boldsymbol{e}_k(\boldsymbol{x})\|_2 \leq \|\bar{\boldsymbol{e}}_k(\boldsymbol{x})\|_2 + \|\mathfrak{L}_k^{\mathrm{KL}}(\boldsymbol{x}) - \hat{\mathfrak{L}}_k^{\mathrm{KL}}(\boldsymbol{x})\|_2$, we get $\max_{i\in\mathcal{V}}|\bar{\mathscr{U}}_k^i(\boldsymbol{x}) - \mathscr{L}_k^{\mathrm{KL}}(\boldsymbol{x})| \leq \Xi$ for all $k \geq \kappa$, where

$$\begin{aligned}
\Xi &= \frac{(1+\eta)2\mathfrak{b}(N-1)\sqrt{N}(\Delta\theta_L + 2\varepsilon_L)}{1 - \left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}} + 2\sqrt{N}\varepsilon_L\,,\\
\kappa &= \left\lceil \frac{2\mathfrak{b}(N-1)\log\left(\mathfrak{D}_4\right)}{\log\left(1 - N\gamma^{2\mathfrak{b}(N-1)}\right)} \right\rceil + 1\,,\\
\mathfrak{D}_4 &= \frac{\eta}{\frac{(\mathfrak{D}_1+2\varepsilon_L)\left(1-\left(1-N\gamma^{2\mathfrak{b}(N-1)}\right)^{\frac{1}{2}}\right)}{2\mathfrak{b}(N-1)(\Delta\theta_L+2\varepsilon_L)} - 1}\,.
\end{aligned}$$

We get (3.50) using $2N\Xi \leq \delta$ and (3.46) by substituting $\Delta_{\max}$ into κ. ■

It follows from Theorem 3.6 that in order to generate satisfactory estimates using the DBF algorithm, the bound ε_L should be substantially smaller than δ.

3.3.3 Special Case: DBF-Kalman Information Filter

In this section, we apply the DBF algorithm to the special case where the target dynamics and measurement models are given by linear systems with additive Gaussian noise. The linear dynamics of the target in discrete time is given by:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{v}_k \,, \tag{3.52}$$

where the process noise $\boldsymbol{v}_k$ is a zero mean multivariate normal distribution with covariance $\boldsymbol{Q}_k$. The linear measurement model of the i^{th} agent is given by:

$$\boldsymbol{y}_k^i = \boldsymbol{H}_k^i \boldsymbol{x}_k + \boldsymbol{w}_k^i \,, \quad \forall i \in \mathcal{V} \,, \tag{3.53}$$

where the measurement noise $\boldsymbol{w}_k^i$ is a zero mean multivariate normal distribution with covariance $\boldsymbol{R}_k^i$. For each agent, the initial prior of the target's states $p(\boldsymbol{x}_{1|0})$ is a multivariate normal distribution with mean $\hat{\boldsymbol{x}}_0^j$ and covariance $\boldsymbol{P}_0^j$.

The prior and posterior pdfs are given by:

$$\mathcal{S}_k^i = p(\boldsymbol{x}_{k|k-1}) = \mathcal{N}(\hat{\boldsymbol{x}}_{k|k-1}^i, \boldsymbol{P}_{k|k-1}^i) \,, \tag{3.54}$$

$$\mathcal{W}_k^i = p(\boldsymbol{x}_{k|k}) = \mathcal{N}(\hat{\boldsymbol{x}}_{k|k}^i, \boldsymbol{P}_{k|k}^i) \,, \tag{3.55}$$

where $\hat{\boldsymbol{x}}_{k|k-1}^i$ and $\hat{\boldsymbol{x}}_{k|k}^i$ denote the means and $\boldsymbol{P}_{k|k-1}^i$ and $\boldsymbol{P}_{k|k}^i$ denote the covariance matrices of the two multivariate normal distributions. Here we adopt the information filter-based representation. We define the following information vectors, information matrices, and trans-

formed measurements [172, 173]:

$$\hat{\boldsymbol{z}}^i_{k|k} = (\boldsymbol{P}^i_{k|k})^{-1}\hat{\boldsymbol{x}}^i_{k|k}\,, \qquad \boldsymbol{Z}^i_{k|k} = (\boldsymbol{P}^i_{k|k})^{-1}\,,$$

$$\hat{\boldsymbol{z}}^i_{k|k-1} = (\boldsymbol{P}^i_{k|k-1})^{-1}\hat{\boldsymbol{x}}^i_{k|k-1}\,, \qquad \boldsymbol{Z}^i_{k|k-1} = (\boldsymbol{P}^i_{k|k-1})^{-1}\,,$$

$$\boldsymbol{i}^i_k = (\boldsymbol{H}^i_k)^T(\boldsymbol{R}^i_k)^{-1}\boldsymbol{y}^i_k\,, \tag{3.56}$$

$$\boldsymbol{I}^i_k = (\boldsymbol{H}^i_k)^T(\boldsymbol{R}^i_k)^{-1}\boldsymbol{H}^i_k\,. \tag{3.57}$$

The prediction step (3.3) is given by:

$$\boldsymbol{M}^i_k = (\boldsymbol{F}^{-1}_k)^T\boldsymbol{Z}^i_{k-1|k-1}\boldsymbol{F}^{-1}_k\,,$$

$$\boldsymbol{C}^i_k = \boldsymbol{M}^i_k\left(\boldsymbol{M}^i_k + \boldsymbol{Q}^{-1}_k\right)^{-1}\,, \qquad \boldsymbol{L}^i_k = \mathbf{I} - \boldsymbol{C}^i_k\,,$$

$$\boldsymbol{Z}^i_{k|k-1} = \boldsymbol{L}^i_k\boldsymbol{M}^i_k(\boldsymbol{L}^i_k)^T + \boldsymbol{C}^i_k\boldsymbol{Q}^{-1}_k(\boldsymbol{C}^i_k)^T\,, \tag{3.58}$$

$$\hat{\boldsymbol{z}}^i_{k|k-1} = \boldsymbol{L}^i_k(\boldsymbol{F}^{-1}_k)^T\hat{\boldsymbol{z}}^i_{k-1|k-1}\,. \tag{3.59}$$

The effect on the update step of the information filter due to the weights on the likelihood function is shown in Appendix B. The fusion rule (3.26)–(3.27) is given by:

$$\boldsymbol{u}^i_k = \boldsymbol{i}^i_k - \boldsymbol{i}^i_{k-1} + \sum_{j\in\mathcal{J}^i_k}\mathcal{A}_k[i,j]\boldsymbol{u}^j_{k-1}\,, \tag{3.60}$$

$$\boldsymbol{U}^i_k = \boldsymbol{I}^i_k - \boldsymbol{I}^i_{k-1} + \sum_{j\in\mathcal{J}^i_k}\mathcal{A}_k[i,j]\boldsymbol{U}^j_{k-1}\,, \tag{3.61}$$

$$\boldsymbol{t}^i_k = N\boldsymbol{u}^i_k\,, \qquad \boldsymbol{T}^i_k = N\boldsymbol{U}^i_k\,, \tag{3.62}$$

with initial condition $\boldsymbol{u}^i_1 = \boldsymbol{i}^i_1$ and $\boldsymbol{U}^i_1 = \boldsymbol{I}^i_1$ for the 1$^{\text{st}}$ time instant. Finally, the update step (3.40) is given by:

$$\hat{\boldsymbol{z}}^i_{k|k} = \hat{\boldsymbol{z}}^i_{k|k-1} + \boldsymbol{t}^j_k\,, \qquad \boldsymbol{Z}^i_{k|k} = \boldsymbol{Z}^i_{k|k-1} + \boldsymbol{T}^j_k\,. \tag{3.63}$$

The mean and covariance matrix can be found using reverse transformations (e.g., $\boldsymbol{P}^i_{k|k} = (\boldsymbol{Z}^i_{k|k})^{-1}$ and $\hat{\boldsymbol{x}}^i_{k|k} = \boldsymbol{P}^i_{k|k}\hat{\boldsymbol{z}}^i_{k|k}$). The pseudo-code of the DBF-Kalman information filtering algorithm for linear-Gaussian models is given in Algorithm 7.

1. (i^{th} agent's steps at k^{th} time instant)
2. Compute prior $\boldsymbol{Z}^i_{k|k-1}$ and $\hat{\boldsymbol{z}}^i_{k|k-1}$ using (3.58)–(3.59)
3. Obtain local measurement $\boldsymbol{y}^i_k$
4. Compute $\boldsymbol{I}^i_k$ and $\boldsymbol{i}^i_k$ using (3.56)–(3.57)
5. Receive $\boldsymbol{U}^j_{k-1}$ and $\boldsymbol{u}^j_{k-1}$ from agents $j \in \mathcal{J}^i_k$
6. Compute $\boldsymbol{U}^i_k$, $\boldsymbol{u}^i_k$, $\boldsymbol{T}^i_k$, and $\boldsymbol{t}^i_k$ using (3.60)–(3.62)
7. Compute posterior $\boldsymbol{Z}^i_{k|k}$ and $\hat{\boldsymbol{z}}^i_{k|k}$ using (3.63)

Algorithm 7: DBF-Kalman Information Filtering Algorithm

3.3.4 Special Case: Multiple Consensus Loops within Each Time Instant

In this section, we show that the proposed DBF algorithm can be easily extended to recursively combine local likelihood functions using multiple consensus loops within each time instant (as opposed to a single consensus step at each time instant) so that each agent's estimated likelihood function converges to the joint likelihood function $\mathcal{L}^C_k$ (3.20). Then, the resultant DBF algorithm is equivalent to the Bayesian consensus algorithms [92, 93], whose requirement of multiple consensus loops within each time step significantly reduces the practicality of such algorithms. Let the pdfs $\mathcal{U}^i_{k,\nu} \in \Phi(\mathcal{X})$ and $\mathcal{T}^i_{k,\nu} \in \Phi(\mathcal{X})$ denote to the local pdfs of the i^{th} agent during the ν^{th} consensus loop at the k^{th} time instant. Since the pdf $\mathcal{L}^i_k$ is not updated during the k^{th} time instant, we define the pdfs $\mathcal{L}^i_{k,\nu} = \mathcal{L}^i_k$ for all

$\nu \in \mathbb{N}$. If each agent updates its local pdfs $\mathcal{U}^i_{k,\nu}$ and $\mathcal{T}^i_{k,\nu}$ using our DBF algorithm as follows:

$$\mathcal{U}^i_{k,\nu} = \begin{cases} \mathcal{L}^i_{k,1} & \text{if } \nu = 1 \\ \dfrac{\tilde{\Lambda}\, \mathcal{L}^i_{k,\nu} \left(\mathcal{L}^i_{k,\nu-1}\right)^{-1}}{\int_{\mathcal{X}} \tilde{\Lambda}\, \mathcal{L}^i_{k,\nu} \left(\mathcal{L}^i_{k,\nu-1}\right)^{-1} d\mu(\bar{\boldsymbol{x}})} & \text{if } \nu \geq 2 \end{cases}, \tag{3.64}$$

$$\text{where } \tilde{\Lambda} = \prod_{j \in \mathcal{J}^i_k} (\mathcal{U}^j_{k,\nu-1})^{\mathcal{A}_k[i,j]},$$

$$\mathcal{T}^i_k = \frac{(\mathcal{U}^i_{k,\nu})^N}{\int_{\mathcal{X}} (\mathcal{U}^i_{k,\nu})^N d\mu(\bar{\boldsymbol{x}})}, \tag{3.65}$$

then each pdf $\mathcal{T}^i_{k,\nu}$ globally exponentially converges pointwise to the normalized joint likelihood function $\mathcal{L}^C_k$ (3.20) at a rate faster or equal to the second-largest singular value of the doubly stochastic matrix $\mathcal{A}_k$, where the communication network topology is strongly connected. The proof of global exponentially convergence follows from [95]. Thus, the distributed estimation algorithm in [92, 93] is a special case of our DBF algorithm.

3.4 Bayesian Consensus Filtering Algorithm

In this section, we present the BCF algorithm and also presents some extensions. In this section, we assume that multiple consensus loops are executed within each time instant. Let the pdf $\mathcal{F}^j_{k,\nu}$ represent the estimated pdf of the j^{th} agent during the ν^{th} consensus loop at the k^{th} time instant. We now state the main result of this section.

Theorem 3.7. *Under Assumptions 3.1–3.3 and* $\mathfrak{b} = 1$, *if each agent updates its estimated pdf* $\mathcal{F}^i_{k,\nu}$ *using the following fusion rule:*

$$\mathcal{F}^i_{k,\nu} = \begin{cases} \dfrac{\prod\limits_{j \in \mathcal{J}^i_k} (\mathcal{F}^j_{k,\nu-1})^{\mathcal{A}_k[j,i]}}{\int_{\mathcal{X}} \prod\limits_{j \in \mathcal{J}^i_k} (\mathcal{F}^j_{k,\nu-1})^{\mathcal{A}_k[j,i]}\, d\mu(\boldsymbol{x})} & \textit{if } \nu \geq 2 \\ \mathcal{W}^i_k & \textit{if } \nu = 1 \end{cases}, \tag{3.66}$$

then each $\mathcal{F}^j_{k,\nu}$ *globally exponentially converges pointwise to the pdf* $\mathcal{W}^{KL}_k$ *(3.15) defined in Lemma 3.3 at a rate faster or equal to* $\sqrt{\lambda_{N-1}(\mathcal{A}_k^T\mathcal{A}_k)} = \sigma_{N-1}(\mathcal{A}_k)$. *Furthermore, their induced measures globally exponentially converge in TV, i.e.,* $\lim_{\nu\to\infty}\mu_{\mathcal{F}^i_{k,\nu}} \xrightarrow{T.V.} \mu_{\mathcal{W}^{KL}_k}$ *for all agents.*

Proof: The matrix $\mathcal{A}_k$ is a nonnegative, doubly stochastic, and irreducible matrix. Since all the diagonal entries of $\mathcal{A}_k$ are positive, it is a primitive matrix [141, Lemma 8.5.4, pp. 516]. It follows from the Perron–Frobenius theorem (cf. [135, pp. 3]) that $\lim_{\nu\to\infty}\mathcal{A}_k^\nu = \frac{1}{N}\mathbf{1}\mathbf{1}^T$.

The functions $\mathscr{F}^i_{k,\nu}(\boldsymbol{x}) = \ln\left[\frac{\mathcal{F}^i_{k,\nu}(\boldsymbol{x})}{\mathcal{F}^i_{k,\nu}(\boldsymbol{\psi})}\right]$ for all agents and are defined using Definition 3.1. Then (3.66) can be re-written as:

$$\mathscr{F}^i_{k,\nu} = \sum_{j=1}^{N}\mathcal{A}_k[j,i]\mathscr{F}^j_{k-1,\nu}\,,\forall\nu\geq 2\,, \tag{3.67}$$

because $\mathcal{A}_k[j,i]=0$ if $j\notin\mathcal{J}^i_k$. Obviously, (3.67) generalizes to:

$$\mathscr{H}_{k,\nu} = \mathcal{A}_k\mathscr{H}_{k,\nu-1}\,, \tag{3.68}$$

$$\mathscr{H}_{k,\nu} = \mathcal{A}_k^{(\nu-1)}\mathscr{H}_{k,1}\,, \tag{3.69}$$

$$\text{where}\qquad \mathscr{H}_{k,\nu} = \left[\mathscr{F}^1_{k,\nu},\ldots,\mathscr{F}^N_{k,\nu}\right]^T\,,$$

$$\mathscr{H}^\star_k = \begin{bmatrix}\mathscr{W}^{KL}_k\\ \vdots\\ \mathscr{W}^{KL}_k\end{bmatrix} = \frac{1}{N}\mathbf{1}\mathbf{1}^T\begin{bmatrix}\mathscr{F}^1_{k,\nu}\\ \vdots\\ \mathscr{F}^N_{k,\nu}\end{bmatrix}\,.$$

It follows from (3.69) that $\lim\nu\to\infty\mathscr{H}_{k,\nu} = \mathscr{H}^\star_k$. We now focus on the rate of convergence.

Let $V_{\text{tr}} = \left[\frac{1}{\sqrt{N}}\mathbf{1},\, V_{\text{s}}\right]$ be the orthonormal matrix of eigenvectors of the symmetric primitive matrix $\mathcal{A}_k^T\mathcal{A}_k$. By spectral decomposition [12], we get:

$$V_{\text{tr}}^T\mathcal{A}_k^T\mathcal{A}_kV_{\text{tr}} = \begin{bmatrix}1 & \mathbf{0}^{1\times(N-1)}\\ \mathbf{0}^{(N-1)\times 1} & V_{\text{s}}^T\mathcal{A}_k^T\mathcal{A}_kV_{\text{s}}\end{bmatrix}\,,$$

where $\frac{1}{N}\mathbf{1}^T\mathcal{A}_k^T\mathcal{A}_k\mathbf{1} = 1$, $\frac{1}{\sqrt{N}}\mathbf{1}^T\mathcal{A}_k^T\mathcal{A}_k V_\mathrm{s} = \mathbf{0}^{1\times(N-1)}$, and $V_\mathrm{s}^T\mathcal{A}_k^T\mathcal{A}_k\mathbf{1}\frac{1}{\sqrt{N}} = \mathbf{0}^{(N-1)\times 1}$ are used. Since the eigenvectors are orthonormal, $V_sV_s^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T = \mathbf{I}$. The rate at which $\mathscr{H}_{k,\nu}$ synchronizes to $\frac{1}{\sqrt{N}}\mathbf{1}$ (or $\mathscr{H}_k^\star$) is equal to the rate at which $V_\mathrm{s}^T\mathscr{H}_{k,\nu} \to \mathbf{0}^{(N-1)\times 1}$. Pre-multiplying (3.68) by V_s^T and substituting $V_s^T\mathbf{1} = 0$ results in:

$$\begin{aligned} V_\mathrm{s}^T\mathscr{H}_{k,\nu} &= V_\mathrm{s}^T\mathcal{A}_k\left(V_sV_s^T + \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)\mathscr{H}_{k,\nu-1} \\ &= V_\mathrm{s}^T\mathcal{A}_kV_sV_s^T\mathscr{H}_{k,\nu-1}\,. \end{aligned}$$

Let $\mathbf{z}_{k,\nu} = V_\mathrm{s}^T\mathscr{H}_{k,\nu}$. The corresponding virtual dynamics is represented by $\mathbf{z}_{k,\nu} = (V_\mathrm{s}^T\mathcal{A}_kV_s)\mathbf{z}_{k,\nu-1}$, which has both $V_\mathrm{s}^T\mathcal{U}_{k,\nu}$ and $\mathbf{0}$ as particular solutions. Let $\Phi_{k,\nu} = \mathbf{z}_{k,\nu}^T\mathbf{z}_{k,\nu}$ be a candidate Lyapunov function for this dynamics. Expanding this gives:

$$\Phi_{k,\nu} = \mathbf{z}_{k,\nu-1}^TV_\mathrm{s}^T\mathcal{A}_k^T\mathcal{A}_kV_s\mathbf{z}_{k,\nu-1} \leq \left(\lambda_{\max}(V_\mathrm{s}^T\mathcal{A}_k^T\mathcal{A}_kV_s)\right)\Phi_{k,\nu-1}.$$

Note that $V_\mathrm{s}^T\mathcal{A}_k^T\mathcal{A}_kV_s$ contains all the eigenvalues of $\mathcal{A}_k^T\mathcal{A}_k$ other than 1. Hence $\lambda_{\max}(V_\mathrm{s}^T\mathcal{A}_k^T\mathcal{A}_kV_s) = \lambda_{N-1}(\mathcal{A}_k^T\mathcal{A}_k) < 1$ and $\Phi_{k,\nu}$ globally exponentially vanishes with a rate faster or equal to $\lambda_{N-1}(\mathcal{A}_k^T\mathcal{A}_k)$. Hence each $\mathscr{F}_{k,\nu}^i$ globally exponentially converges pointwise to $\mathscr{W}_k^{KL}$ with a rate faster or equal to $\sqrt{\lambda_{N-1}(\mathcal{A}_k^T\mathcal{A}_k)} = \sigma_{N-1}(\mathcal{A}_k)$.

Next, we need to find the rate of convergence of $\mathcal{F}_{k,\nu}^i$ to $\mathcal{W}_k^{KL}$. From the exponential convergence of $\mathscr{F}_{k,\nu}^i$ to $\mathscr{W}_k^{KL}$, we get:

$$\left|\ln\left[\frac{\mathcal{F}_{k,\nu}^i(\boldsymbol{x})}{\mathcal{F}_{k,\nu}^i(\boldsymbol{\psi})}\frac{\mathcal{W}_k^{KL}(\boldsymbol{\psi})}{\mathcal{W}_k^{KL}(\boldsymbol{x})}\right]\right| \leq \sigma_{N-1}(\mathcal{A}_k)\left|\ln\left[\frac{\mathcal{F}_{k,\nu-1}^i(\boldsymbol{x})}{\mathcal{F}_{k,\nu-1}^i(\boldsymbol{\psi})}\frac{\mathcal{W}_k^{KL}(\boldsymbol{\psi})}{\mathcal{W}_k^{KL}(\boldsymbol{x})}\right]\right|. \tag{3.70}$$

Let us define the function $\alpha_{k,\nu}^i(\boldsymbol{x})$ such that $\alpha_{k,\nu}^i(\boldsymbol{x}) = \left[\frac{\mathcal{F}_{k,\nu}^i(\boldsymbol{x})}{\mathcal{F}_{k,\nu}^i(\boldsymbol{\psi})}\frac{\mathcal{W}_k^{KL}(\boldsymbol{\psi})}{\mathcal{W}_k^{KL}(\boldsymbol{x})}\right]$ if $\mathcal{F}_{k,\nu}^i(\boldsymbol{x})\mathcal{W}_k^{KL}(\boldsymbol{\psi}) \geq \mathcal{F}_{k,\nu}^i(\boldsymbol{\psi})\mathcal{W}_k^{KL}(\boldsymbol{x})$ and $\alpha_{k,\nu}^i(\boldsymbol{x}) = \left[\frac{\mathcal{F}_{k,\nu}^i(\boldsymbol{\psi})}{\mathcal{F}_{k,\nu}^i(\boldsymbol{x})}\frac{\mathcal{W}_k^{KL}(\boldsymbol{x})}{\mathcal{W}_k^{KL}(\boldsymbol{\psi})}\right]$ otherwise. Note that $\alpha_{k,\nu}^i(\boldsymbol{x})$ is a continuous function since it is a product of continuous functions. Since $\alpha_{k,\nu}^i(\boldsymbol{x}) \geq 1$ and $\ln\left(\alpha_{k,\nu}^i(\boldsymbol{x})\right) \geq$

0, $\forall \boldsymbol{x} \in \mathcal{X}$, (3.70) simplifies to:

$$\begin{aligned} \ln\left(\alpha^i_{k,\nu}(\boldsymbol{x})\right) &\leq \sigma_{N-1}(\mathcal{A}_k) \ln\left(\alpha^i_{k,\nu-1}(\boldsymbol{x})\right) . \\ \alpha^i_{k,\nu}(\boldsymbol{x}) &\leq \left(\alpha^i_{k,1}(\boldsymbol{x})\right)^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}} . \end{aligned} \tag{3.71}$$

Since $\mathcal{F}^i_{k,\nu}(\boldsymbol{x})$ tends to $\mathcal{W}^{KL}_k(\boldsymbol{x})$, i.e., $\lim_{\nu\to\infty} \alpha^i_{k,\nu}(\boldsymbol{x}) = 1$, we can write (3.71) as:

$$\alpha^i_{k,\nu}(\boldsymbol{x}) - 1 \leq \left(\alpha^i_{k,1}(\boldsymbol{x})\right)^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}} - 1^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}} . \tag{3.72}$$

Using the mean value theorem (cf. [174]), the right hand side of (3.72) can be simplified to (3.73), for some $c \in [1, \alpha^i_{k,1}(\boldsymbol{x})]$ as follows:

$$\left(\alpha^i_{k,1}(\boldsymbol{x})\right)^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}} - 1^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}} = (\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)} \left(c^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}-1}\right) \left(\alpha^i_{k,1}(\boldsymbol{x}) - 1\right) . \tag{3.73}$$

As $\sigma_{N-1}(\mathcal{A}_k) < 1$, the maximum value of $\left(c^{(\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)}-1}\right)$ is 1. Substituting this into (3.72) gives:

$$\alpha^i_{k,\nu}(\boldsymbol{x}) - 1 \leq (\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)} \left(\alpha^i_{k,1}(\boldsymbol{x}) - 1\right) . \tag{3.74}$$

Hence $\alpha^i_{k,\nu}(\boldsymbol{x})$ exponentially converges to 1 with a rate faster or equal to $\sigma_{N-1}(\mathcal{A}_k)$. Irrespective of the orientation of $\alpha^i_{k,\nu}(\boldsymbol{x})$ and $\alpha^i_{k,1}(\boldsymbol{x})$, (3.74) can be written as (3.75) by multiplying with $\frac{1}{\alpha^i_{k,\nu}(\boldsymbol{x})}$ or $\frac{1}{\alpha^i_{k,1}(\boldsymbol{x})}$, and then with $\mathcal{W}^{KL}_k(\boldsymbol{x})$.

$$\left| \frac{\mathcal{W}^{KL}_k(\boldsymbol{\psi})}{\mathcal{F}^i_{k,\nu}(\boldsymbol{\psi})} \mathcal{F}^i_{k,\nu}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x}) \right| \leq (\sigma_{N-1}(\mathcal{A}_k))^{(\nu-1)} \left| \frac{\mathcal{W}^{KL}_k(\boldsymbol{\psi})}{\mathcal{F}^i_{k,1}(\boldsymbol{\psi})} \mathcal{F}^i_{k,1}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x}) \right| . \tag{3.75}$$

We can choose $\bar{\boldsymbol{\psi}} \in \mathcal{X}$ such that $\mathcal{F}^i_{k,1}(\bar{\boldsymbol{\psi}}) = \mathcal{W}^{KL}_k(\bar{\boldsymbol{\psi}})$. Now we discuss two cases to reduce

the left hand side of (3.75) to $\left|\mathcal{F}^i_{k,\nu}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x})\right|$.

$$
\begin{aligned}
&\left|\frac{\mathcal{W}^{KL}_k(\bar{\boldsymbol{\psi}})}{\mathcal{F}^i_{k,\nu}(\bar{\boldsymbol{\psi}})}\mathcal{F}^i_{k,\nu}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x})\right| \\
&= \begin{cases} \left|\mathcal{F}^i_{k,\nu}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x}) + \left(\frac{\mathcal{W}^{KL}_k(\bar{\boldsymbol{\psi}})}{\mathcal{F}^i_{k,\nu}(\bar{\boldsymbol{\psi}})} - 1\right)\mathcal{F}^i_{k,\nu}(\boldsymbol{x})\right| & \text{if } \frac{\mathcal{W}^{KL}_k(\bar{\boldsymbol{\psi}})}{\mathcal{F}^i_{k,\nu}(\bar{\boldsymbol{\psi}})} \geq 1 \\ \left|\mathcal{W}^{KL}_k(\boldsymbol{x}) - \mathcal{F}^i_{k,\nu}(\boldsymbol{x}) + \left(1 - \frac{\mathcal{W}^{KL}_k(\bar{\boldsymbol{\psi}})}{\mathcal{F}^i_{k,\nu}(\bar{\boldsymbol{\psi}})}\right)\mathcal{F}^i_{k,\nu}(\boldsymbol{x})\right| & \text{if } \frac{\mathcal{W}^{KL}_k(\bar{\boldsymbol{\psi}})}{\mathcal{F}^i_{k,\nu}(\bar{\boldsymbol{\psi}})} < 1 \end{cases} \\
&\geq \left|\mathcal{W}^{KL}_k(\boldsymbol{x}) - \mathcal{F}^i_{k,\nu}(\boldsymbol{x})\right| .
\end{aligned}
$$

Hence, for both the cases, we are able to simplify (3.75) to:

$$
\left|\mathcal{F}^i_{k,\nu}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x})\right| \leq \left(\sigma_{N-1}(\mathcal{A}_k)\right)^{(\nu-1)} \left|\mathcal{F}^i_{k,1}(\boldsymbol{x}) - \mathcal{W}^{KL}_k(\boldsymbol{x})\right| .
$$

Thus each $\mathcal{F}^i_{k,\nu}(\boldsymbol{x})$ globally exponentially converges to $\mathcal{W}^{KL}_k(\boldsymbol{x})$ with a rate faster or equal to $\sigma_{N-1}(\mathcal{A}_k)$. ■

1. (i^{th} agent's steps at k^{th} time instant)
2. Compute prior pdf $\mathcal{U}^i_k = p(\boldsymbol{x}_{k|k-1})$ using (3.3)
3. Compute posterior pdf $\mathcal{W}^i_k = p(\boldsymbol{x}_{k|k})$ using (3.4)
4. **for** $\nu = 1$ to n_{loop}
5. **if** $\nu = 1$, **then**
6. Set estimated pdf $\mathcal{F}^i_{k,\nu} = \mathcal{W}^i_k$
7. **else**
8. Exchange estimated pdf $\mathcal{F}^i_{k,\nu-1}$ with neighboring agents $\mathcal{J}^j_k$
9. Compute estimated pdf $\mathcal{F}^i_{k,\nu}$ using (3.66)
10. **end if**
11. **end for**
12. Set posterior pdf $\mathcal{W}^i_k = \mathcal{F}^i_{k,n_{\text{loop}}}$

Algorithm 8: Bayesian Consensus Filtering Algorithm

The pseudocode of the BCF algorithm is given in Algorithm 8. According to [175], this strategy of first updating the local estimate and then combining these local estimates to achieve a consensus is stable and gives the best performance in comparison with other update–combine strategies.

We now state that the methods for recursively combining probability distributions to reach a consensual distribution are limited to LinOP, LogOP, and their affine combinations.

Remark 3.2. The LinOP and LogOP methods for combining probability distributions can be generalized by the g–Quasi–Linear Opinion Pool (g–QLOP), which is described by the following equation:

$$\mathcal{F}_{k,\nu}^{j} = \frac{g^{-1}(\sum_{\ell\in\mathcal{J}_k^j} \alpha_{k,\nu-1}^{j\ell} g(\mathcal{F}_{k,\nu-1}^{\ell}))}{\int_{\mathcal{X}} g^{-1}(\sum_{\ell\in\mathcal{J}_k^j} \alpha_{k,\nu-1}^{j\ell} g(\mathcal{F}_{k,\nu-1}^{\ell}))\, d\mu(\boldsymbol{x})}, \forall j \in \{1,\dots,N\}, \forall \nu \in \mathbb{N}, \tag{3.76}$$

where g is a continuous, strictly monotone function. It is shown in [100] that, other than the linear combination of LinOP and LogOP, there is no function g for which the final consensus can be expressed by the following equation:

$$\lim_{\nu\to\infty} \mathcal{F}_{k,\nu}^{j} = \frac{g^{-1}(\sum_{j=1}^{N} \pi_j g(\mathcal{F}_{k,0}^{\ell}))}{\int_{\mathcal{X}} g^{-1}(\sum_{j=1}^{N} \pi_j g(\mathcal{F}_{k,0}^{\ell}))\, d\mu(\boldsymbol{x})}, \forall j \in \{1,\dots,N\}, \forall \nu \in \mathbb{N}, \tag{3.77}$$

where $\boldsymbol{\pi}$ is the unique stationary solution. Moreover, the function g is said to be k-Markovian if the scheme for combining probability distribution (3.76) yields the consensus (3.77) for every regular communication network topology and for all initial positive densities. It is also shown that g is k-Markovian if and only if the g–QLOP is either LinOP or LogOP [100]. □

3.4.1 Number of Consensus Loops

In this section, we compute the number of consensus loops (n_{loop} in Algorithm 8) needed to reach a satisfactory consensus estimate across the network and discuss the convergence of this algorithm.

Definition 3.4. (Disagreement vector $\boldsymbol{\theta}_{k,\nu}$) Let us define $\boldsymbol{\theta}_{k,\nu} := \left(\theta_{k,\nu}^{1},\dots,\theta_{k,\nu}^{N}\right)^{T}$, where $\theta_{k,\nu}^{j} := \|\mathcal{F}_{k,\nu}^{j} - \mathcal{W}_{k}^{KL}\|_{\mathcal{L}_1}$. Since the $\mathcal{L}_1$ distances between pdfs is bounded by 2, the ℓ_2 norm of the disagreement vector ($\|\boldsymbol{\theta}_{k,\nu}\|_{\ell_2}$) is upper bounded by $2\sqrt{N}$. □

This conservative bound is used to obtain the minimum number of consensus loops

for achieving ε-consensus across the network, while tracking a moving target. Let us now quantify the divergence of the local pdfs during the Bayesian filtering stage.

Definition 3.5. (Error propagation dynamics $\mathbf{\Gamma}(\cdot)$) Let us assume that the dynamics of the ℓ_2 norm of the disagreement vector during the Bayesian filtering stage can be obtained from the target dynamics (3.1) and measurement models (3.2). The error propagation dynamics $\mathbf{\Gamma}(\cdot)$ estimates the maximum divergence of the local pdfs during the Bayesian filtering stage, i.e., $\|\boldsymbol{\theta}_{k,1}\|_{\ell_2} \leq \mathbf{\Gamma}\left(\|\boldsymbol{\theta}_{k-1,n_{\text{loop}}}\|_{\ell_2}\right)$, where $\|\boldsymbol{\theta}_{k-1,n_{\text{loop}}}\|_{\ell_2}$ is the disagreement vector with respect to $\mathcal{W}_{k-1}^{KL}$ at the end of the consensus stage during the $(k-1)^{\text{th}}$ time instant; and $\|\boldsymbol{\theta}_{k,1}\|_{\ell_2}$ is the disagreement vector with respect to $\mathcal{W}_{k}^{KL}$ after the update stage during the k^{th} time instant. □

Next we obtain the minimum number of consensus loops for achieving ε-consensus across the network and also derive conditions on the communication network topology for a given number of consensus loops.

Theorem 3.8. *Each agent tracks the target using the BCF algorithm. For some acceptable consensus error* $\varepsilon_{\text{consensus}} > 0$ *and* $\gamma_k = \min\left(\mathbf{\Gamma}\left(\|\boldsymbol{\theta}_{k-1,n_{\text{loop}}}\|_{\ell_2}\right), 2\sqrt{N}\right)$*:*
(i) if the number of consensus loops is at least $n_{\text{loop}} \geq 1 + \left\lceil \frac{\ln(\varepsilon_{\text{consensus}}/\gamma_k)}{\ln \sigma_{N-1}(\mathcal{A}_k)} \right\rceil$ *for a given* $\mathcal{A}_k$*; or*
(ii) if the communication network topology $(\mathcal{A}_k)$ *during the* k^{th} *time instant is such that* $\sigma_{N-1}(\mathcal{A}_k) \leq \left(\frac{\varepsilon_{\text{consensus}}}{\gamma_k}\right)^{\frac{1}{(n_{\text{loop}}-1)}}$ *for a given* n_{loop}*;*
then the ℓ_2 *norm of the disagreement vector at the end of the consensus stage is less than* $\varepsilon_{\text{consensus}}$*, i.e.,* $\|\boldsymbol{\theta}_{k,n_{\text{loop}}}\|_{\ell_2} \leq \varepsilon_{\text{consensus}}$.

Proof: Theorem 3.7 states that the local estimated pdfs $\mathcal{F}_{k,\nu}^{j}$ globally exponentially converges pointwise to the pdf $\mathcal{W}_{k}^{KL}$ (3.15) with a rate of $\sigma_{N-1}(\mathcal{A}_k)$. If $\boldsymbol{\theta}_{k,1}$ is the initial disagreement vector at the start of the consensus stage, then $\|\boldsymbol{\theta}_{k,n_{\text{loop}}}\|_{\ell_2} \leq \left(\sigma_{N-1}(\mathcal{A}_k)\right)^{(n_{\text{loop}}-1)} \|\boldsymbol{\theta}_{k,1}\|_{\ell_2} \leq \left(\sigma_{N-1}(\mathcal{A}_k)\right)^{(n_{\text{loop}}-1)} \gamma_k$. Thus, we get the conditions on n_{loop} or $\sigma_{N-1}(\mathcal{A}_k)$ from the inequality $\left(\sigma_{N-1}(\mathcal{A}_k)\right)^{(n_{\text{loop}}-1)} \gamma_k \leq \varepsilon_{\text{consensus}}$. ■

3.4.2 Hierarchical Bayesian Consensus Filtering

In this section, we assume that only N_1 agents can observe the target at the k^{th} time instant. The remaining agents are not able to observe the target at this time instant. Without loss of generality, we assume that the first N_1 agents are the tracking agents. The objective of hierarchical consensus algorithm is to guarantee that posterior pdfs of only the agents tracking the target contribute to the consensual pdf. We first make the following assumption on the communication network topology.

Assumption 3.6. Let $\mathcal{D}_k$ represent the communication network topology of only the tracking agents.

(i) The communication network topologies $\mathcal{G}_k$ and $\mathcal{D}_k$ are both SC.

(ii) The matrix $\mathcal{A}_k$ can be decomposed into four parts $\mathcal{A}_k = \left[\begin{smallmatrix} \mathcal{A}_{k1} & \mathcal{A}_{k2} \\ \mathcal{A}_{k3} & \mathcal{A}_{k4} \end{smallmatrix}\right]$, where $\mathcal{A}_{k1} \in \mathbb{R}^{N_1 \times N_1}$, $\mathcal{A}_{k2} = \mathbb{R}^{N_1 \times (N-N_1)}$, $\mathcal{A}_{k3} \in \mathbb{R}^{(N-N_1)\times N_1}$, and $\mathcal{A}_{k4} \in \mathbb{R}^{(N-N_1)\times(N-N_1)}$. We assume that $\mathcal{A}_{k1}$ is balanced and $\mathcal{A}_{k2} = \mathbf{0}^{N_1 \times (N-N_1)}$. The matrix $\mathcal{A}_k$ is row stochastic.

(iii) If $i \in \{1, \ldots, N_1\}$, then $\mathcal{A}_k[j,i] > 0$ if and only if either $(j,i) \in \mathcal{E}_k$ and $j \in \{1, \ldots, N_1\}$ or $i = j$. If $i \in \{N_1 + 1, \ldots, N\}$, then $\mathcal{A}_k[j,i] > 0$ if and only if either $(j,i) \in \mathcal{E}_k$ or $i = j$. $\square$

We now state the main result of this section.

Theorem 3.9. *Under Assumptions 3.1, 3.3, and 3.6, if each agent updates its estimated pdf $\mathcal{F}^i_{k,\nu}$ using the fusion rule (3.66), then each pdf $\mathcal{F}^i_{k,\nu}$ globally exponentially converges pointwise to the pdf $\mathcal{F}^{KL}_k$ given by:*

$$\mathcal{F}^{KL}_k = \frac{\prod_{i=1}^{N_1} (\mathcal{W}^i_k)^{\frac{1}{N_1}}}{\int_{\mathcal{X}} \prod_{i=1}^{N_1} (\mathcal{W}^i_k)^{\frac{1}{N_1}} \, d\mu(\boldsymbol{x})} \tag{3.78}$$

at a rate faster or equal to $\sqrt{\lambda_{N_1-1}(\mathcal{A}^T_{k1}\mathcal{A}_{k1})} = \sigma_{N_1-1}(\mathcal{A}_{k1})$. Only the initial estimates of the tracking agents contribute to the consensual pdf $\mathcal{F}^{KL}_k$. Furthermore, their induced measures converge in total variation, i.e., $\lim_{\nu\to\infty} \mu_{\mathcal{F}^j_{k,\nu}} \xrightarrow{T.V.} \mu_{\mathcal{F}^{KL}_k}$ for all agents.

Proof: The matrix $\mathcal{A}_{k1}$ conforms to the balanced digraph $\mathcal{D}_k$. Let $\mathbf{1}_{N_1} = [1, 1, \ldots, 1]^T$, with

N_1 elements. Similar to the proof of Theorem 3.7, we get $\mathcal{A}_{k1}$ is a primitive matrix and $\lim_{\nu\to\infty} \mathcal{A}_{k1}^{\nu} = \frac{1}{N_1}\mathbf{1}_{N_1}\mathbf{1}_{N_1}^T$.

Next, we decompose $\mathscr{H}_{k,\nu}$ (3.68) into two parts $\mathscr{H}_{k,\nu} = [\mathscr{Y}_{k,\nu}; \mathscr{Z}_{k,\nu}]$, where $\mathscr{Y}_{k,\nu} = \left(\mathcal{F}_{k,\nu}^1, \ldots, \mathcal{F}_{k,\nu}^{N_1}\right)^T$ and $\mathscr{Z}_{k,\nu} = \left(\mathcal{F}_{k,\nu}^{N_1+1}, \ldots, \mathcal{F}_{k,\nu}^{N}\right)^T$. Since $\mathcal{A}_{k2}$ is a zero matrix, (3.69) generalizes and hierarchically decomposes to:

$$\mathscr{Y}_{k,\nu+1} = \mathcal{A}_{k1}^{\nu}\mathscr{Y}_{k,0}\,, \qquad \forall \nu \in \mathbb{N}\,, \tag{3.79}$$

$$\mathscr{Z}_{k,\nu+1} = \mathcal{A}_{k3}\mathscr{Y}_{k,\nu} + \mathcal{A}_{k4}\mathscr{Z}_{k,\nu}\,, \qquad \forall \nu \in \mathbb{N}\,. \tag{3.80}$$

Combining equation (3.79) with the previous result gives $\lim_{\nu\to\infty} \mathscr{Y}_{k,\nu} = \frac{1}{N_1}\mathbf{1}_{N_1}\mathbf{1}_{N_1}^T\mathscr{Y}_{k,1}$. Thus $\lim_{\nu\to\infty} \mathscr{F}_{k,\nu}^j = \mathscr{F}_k^{KL} = \frac{1}{N_1}\mathbf{1}_{N_1}^T\mathscr{Y}_{k,1} = \frac{1}{N_1}\sum_{i=1}^{N_1}\mathscr{F}_{k,1}^i$ for all agents $j \in \{1, \ldots, N_1\}$. From the proof of Theorem 3.7, we get the pdfs $\mathcal{F}_{k,\nu}^j, \forall j \in \{1, \ldots, m_1\}$ globally exponentially converges pointwise to $\mathcal{F}_k^{KL}$ (3.78) with a rate faster or equal to $\sigma_{N_1-1}(\mathcal{A}_{k1})$.

Since $\mathcal{G}(k)$ is strongly connected, information from the tracking agents reach the non-tracking agents. Taking the limit of equation (3.80) and substituting the above result gives:

$$\lim_{\nu\to\infty} \mathscr{Z}_{k,\nu+1} = \frac{1}{N_1}\mathcal{A}_{k3}\mathbf{1}_{N_1}\mathbf{1}_{N_1}^T\mathscr{Y}_{k,1} + \mathcal{A}_{k4}\lim_{\nu\to\infty}\mathscr{Z}_{k,\nu} \tag{3.81}$$

Let $\mathbf{1}_{N_2} = [1, 1, \ldots, 1]^T$, with $N_2(= N - N_1)$ elements. Since $\mathcal{A}_k$ is row stochastic, we get $\mathcal{A}_{k3}\mathbf{1}_{N_1} = [\mathbf{I} - \mathcal{A}_{k4}]\mathbf{1}_{N_2}$. Hence, from equation (3.81), we get $\lim_{\nu\to\infty} \mathscr{Z}_{k,\nu} = \frac{1}{N_1}\mathbf{1}_{N_2}\mathbf{1}_{N_1}^T\mathscr{Y}_{k,1}$. Moreover, the inessential states die out geometrically fast [135, pp. 120]. Hence, $\lim_{\nu\to\infty} \mathcal{F}_{k,\nu}^j = \mathcal{F}_k^{KL} = \frac{1}{N_1}\mathbf{1}_{N_1}^T\mathscr{Y}_{k,1}$ for all agents $j \in \{N_1 + 1, \ldots, N\}$. Hence, the estimates of the non-tracking agents $\mathcal{F}_{k,\nu}^j, \forall j \in \{N_1 + 1, \ldots, N\}$ also converge pointwise geometrically fast to the same consensual pdf $\mathcal{F}_k^{KL}$ (3.78). ■

Note that Theorem 3.8 can be directly applied from Section 3.4.1 to compute the minimum number of consensus loops n_{loop} for achieving ϵ-convergence in a given communication network topology or for designing the $\mathcal{A}_{k1}$ matrix for a given number of consensus loops.

A simulation example of Hierarchical BCF algorithm for tracking orbital debris in space is discussed in the next section.

3.5 Numerical Simulations

In this section, we compare the performance of the DBF algorithms with centralized multi-sensor estimation algorithms using the benchmark example studied in [85, 94, 176]. The target's motion is shown in Fig. 3.3. The target dynamics is modeled by a linear (nearly constant velocity) model:

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \boldsymbol{v}_k \,, \qquad \text{where } \boldsymbol{Q} = \begin{bmatrix} \frac{\Delta^3}{3} & \frac{\Delta^2}{2} & 0 & 0 \\ \frac{\Delta^2}{2} & \Delta & 0 & 0 \\ 0 & 0 & \frac{\Delta^3}{3} & \frac{\Delta^2}{2} \\ 0 & 0 & \frac{\Delta^2}{2} & \Delta \end{bmatrix}$$

is the covariance matrix of the process noise $\boldsymbol{v}_k$, Δ is the time step, and the state vector $\boldsymbol{x}_k$ denotes the position and velocity components along the coordinate axes, i.e., $\boldsymbol{x}_k = [x_k,\, \dot{x}_k,\, y_k,\, \dot{y}_k]^T$. As shown in Fig. 3.3, 50 sensing agents are distributed over the given region and are able to communicate with their neighboring agents. The undirected communication network topology is assumed to be time-invariant. Local-degree weights are used to compute the doubly stochastic adjacency matrix $\mathcal{A}_k$ as:

$$\mathcal{A}_k[i,j] = \frac{1}{\max(d_i, d_j)} \,, \qquad \forall j \in \mathcal{J}_k^i \text{ and } i \neq j \,,$$
$$\mathcal{A}_k[i,i] = 1 - \sum_{j \in \mathcal{V} \setminus \{i\}} \mathcal{A}_k[i,j] \,,$$

where d_i denotes the degree of the i^{th} agent.

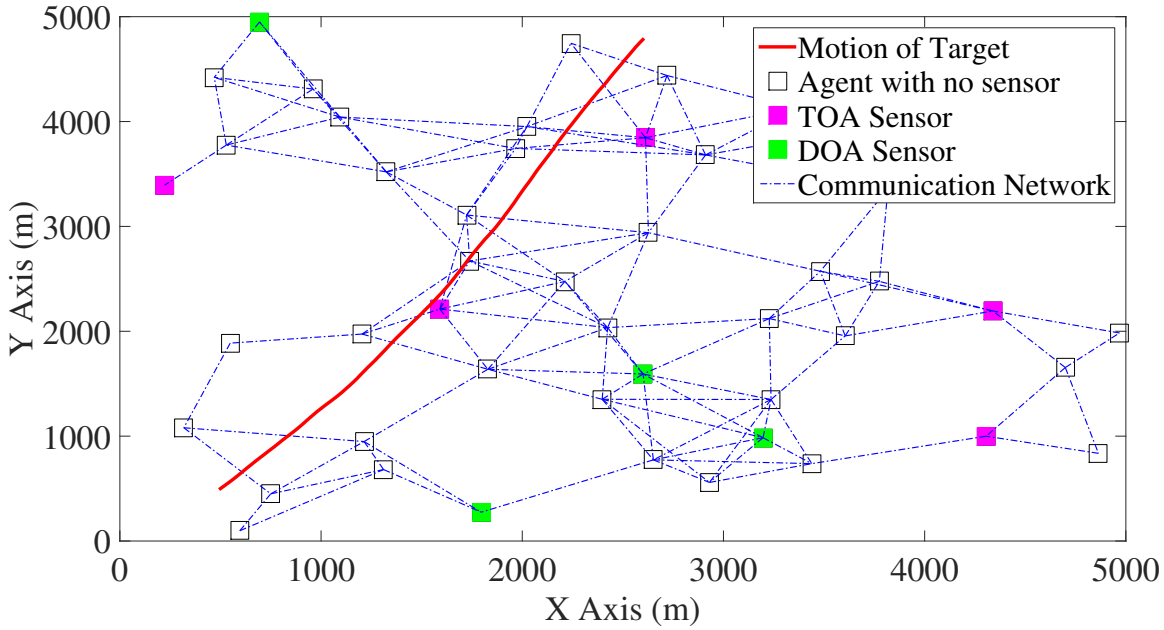


Figure 3.3: The motion of the target, the position of sensing agents (5 TOA sensors, 5 DOA sensors, and 40 agents with no sensors), and their communication network topology are shown.

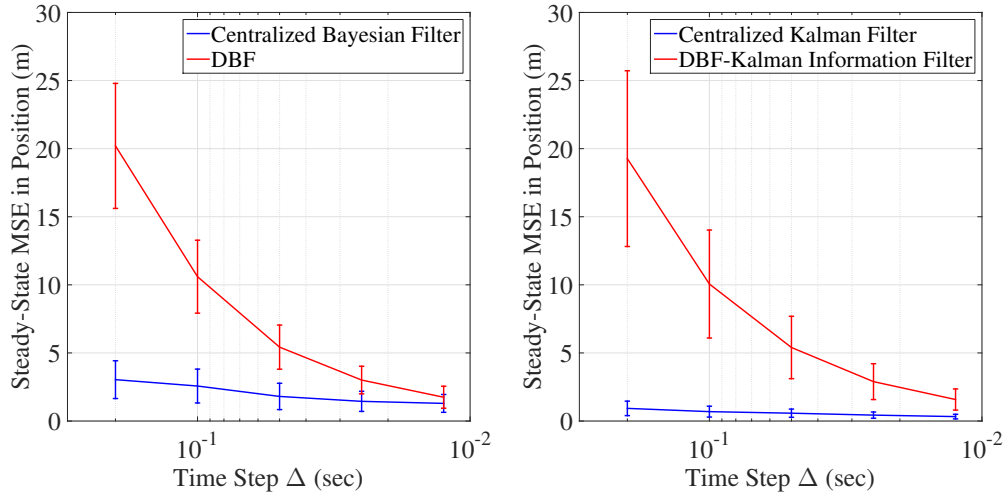


Figure 3.4: Variation of steady-state MSE in position with respect to time step Δ is shown for (a) the centralized Bayesian filtering algorithm and the DBF algorithm in Scenario 1 and (b) the centralized Kalman filtering algorithm and the DBF algorithm for linear-Gaussian models in Scenario 2.

3.5.1 Scenario 1: Nonlinear Measurement Models

In the first scenario, five of these agents are equipped with nonlinear position sensors that can measure their distance to the target using Time of Arrival (TOA) sensors. Another five agents are equipped with Direction of Arrival (DOA) sensors that can measure the bearing angle between the target and themselves. The remaining agents do not have any sensors. The measurement models for these sensors are given by:

$$\boldsymbol{h}_k^i(\boldsymbol{x}_k, \boldsymbol{w}_k^i) = \begin{cases} \text{atan2}(x_k - x^i,\, y_k - y^i) + \boldsymbol{w}_k^i & \text{if } i \text{ has DOA sensor} \\ \sqrt{(x_k - x^i)^2 + (y_k - y^i)^2} + \boldsymbol{w}_k^i & \text{if } i \text{ has TOA sensor} \\ \boldsymbol{w}_k^i & \text{if } i \text{ has no sensor} \end{cases},$$

where (x^i, y^i) denotes the position of the i^{th} agent and atan2 is the 4-quadrant inverse tangent function. The measurement noise $\boldsymbol{w}_k^i$ is a zero mean Gaussian distribution with variance $\sigma_r = 10$ m for the TOA sensor and $\sigma_\theta = 2^\circ$ for the DOA sensor.

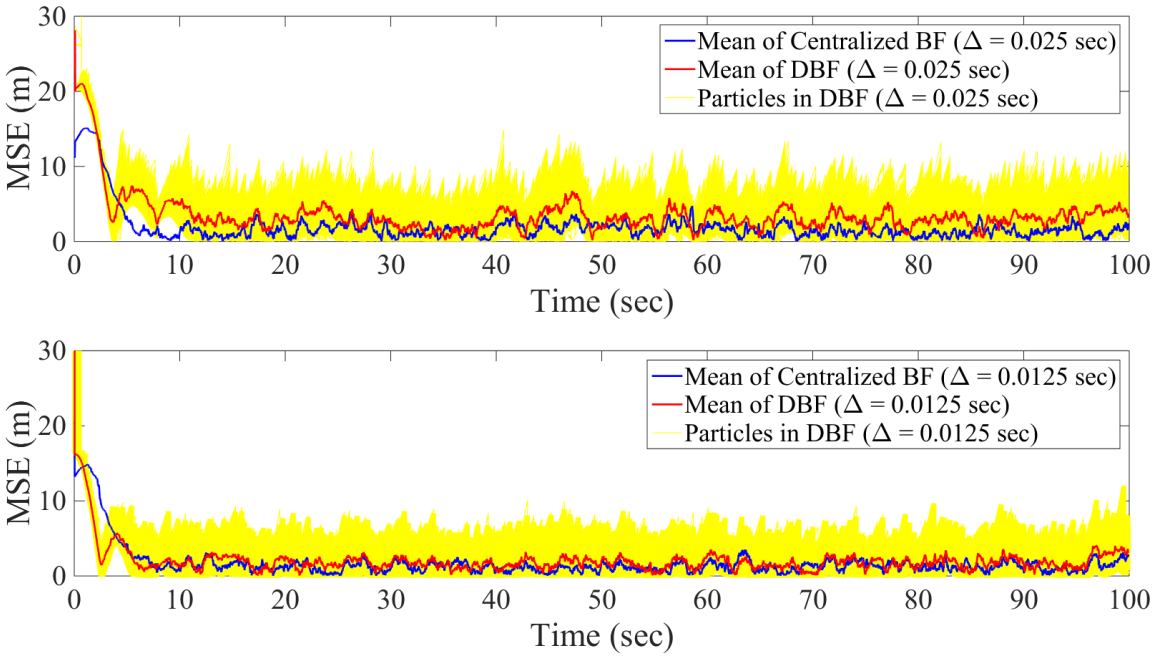


Figure 3.5: The trajectories of the MSE in position for the centralized Bayesian filtering algorithm and the DBF algorithm are shown for two time steps.

Particle filters [177] are used to execute the centralized Bayesian filtering algorithm and the DBF algorithm. Fig. 3.4(a) shows the cumulative results from multiple simulations with varying time steps (Δ) decreasing from 0.2 sec to 0.0125 sec. The same target motion, shown in Fig. 3.3, is used for all simulations. We see that the DBF algorithm's steady-state mean-square-error (MSE) in position converges to that of the centralized Bayesian filtering algorithm as the time step Δ decreases (i.e., the steady-state MSE is smaller than 5 m when the time step $\Delta \leq 0.05$ sec). Note that the MSE of the centralized Bayesian filter does not change much with change in time step because it is constrained by the measurement noise intensities.

Fig. 3.5 shows the trajectories of the MSE in position for two simulation cases. When the time step $\Delta = 0.025$ sec, the steady-state MSE in position for the DBF algorithm converges to that of the centralized Bayesian filter. This is also evident from the results shown in Fig. 3.6. This shows that the performance of the DBF algorithm approaches the performance of the centralized Bayesian filter as the time step is reduced. Finally, Fig. 3.7 shows that the L_1 distances between the estimated likelihood functions and the joint likelihood function are bounded by δ from Theorem 3.4.

3.5.2 Scenario 2: Linear Measurement Models

In the second scenario, the same ten agents (having DOA or TOA sensors) have linear position sensors $\boldsymbol{h}_k^i(\boldsymbol{x}_k, \boldsymbol{w}_k^i) = \left[\begin{smallmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{smallmatrix}\right] \boldsymbol{x}_k + \boldsymbol{w}_k^i$, where the measurement noise $\boldsymbol{w}_k^i$ is a zero mean Gaussian distribution with covariance matrix $R_k^i = 15\mathbf{I}$.

Fig. 3.4(b) shows the cumulative results for the centralized Kalman filtering algorithm and the DBF-Kalman information filtering algorithm for linear-Gaussian models. Note that the difference between the steady-state MSEs of the two algorithms is less than 3 m when the time step Δ decreases below 0.05 sec. This is also evident from the results shown in Fig. 3.8. Hence, the performance of the DBF-Kalman information filtering algorithm for linear-Gaussian models approaches the performance of the centralized Kalman filtering

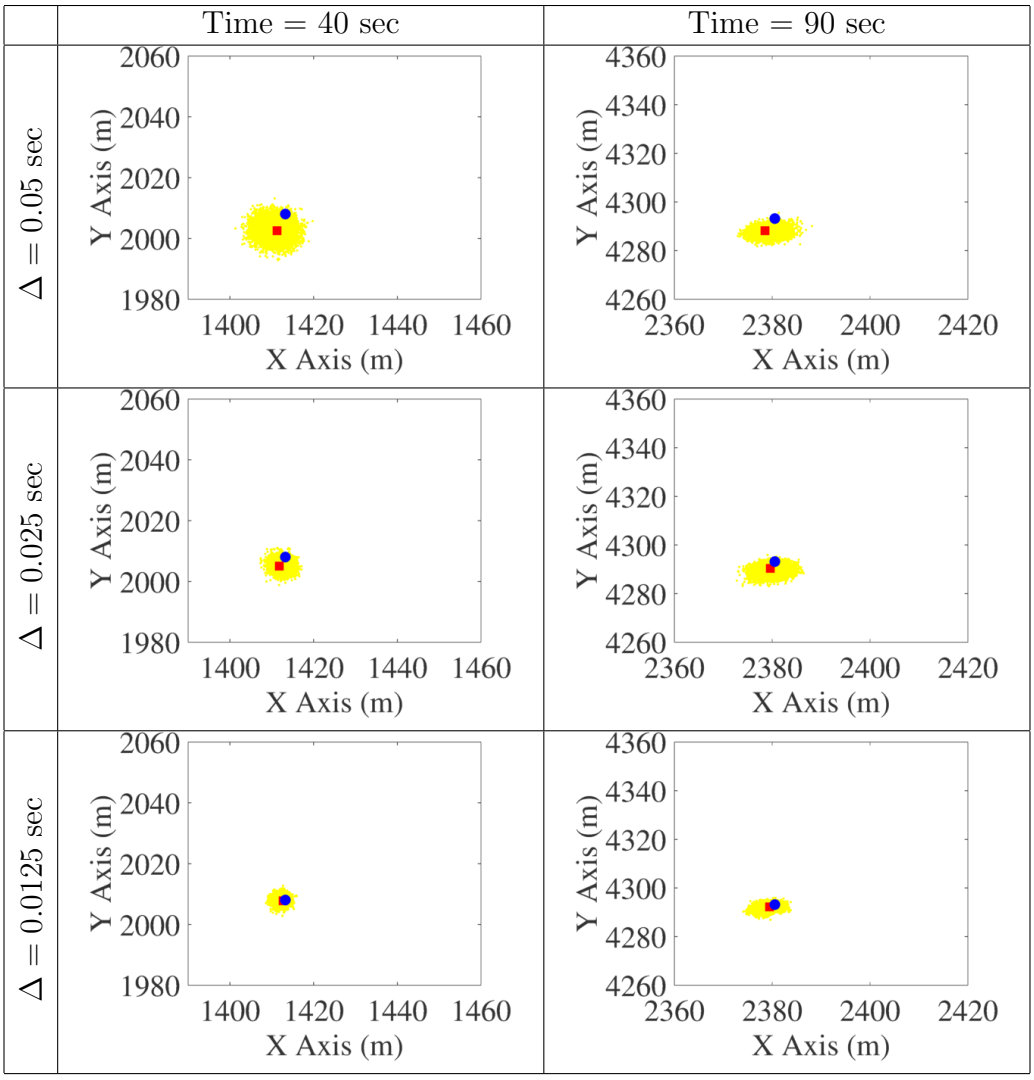


Figure 3.6: All the particles in the estimated pdf of the DBF algorithm are shown, where ● is the actual position of the target and ■ is the mean of the estimated pdf.

algorithm as the time step is reduced. The differences in the simulation results in Section 3.5.1 and Section 3.5.2 arise from the nonlinear and linear measurement models used in these sections respectively.

3.5.3 Tracking Orbital Debris

Currently, there are over ten thousand objects in Earth orbit, of size 0.5 cm or greater, and almost 95% of them are nonfunctional space debris. These debris pose a significant

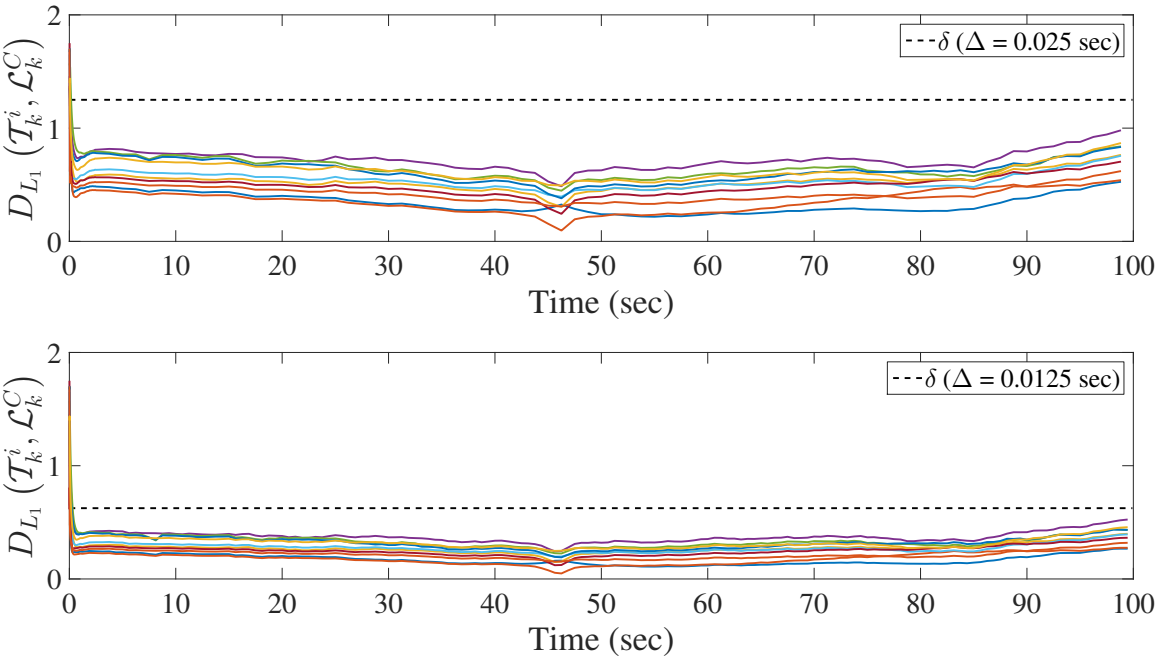


Figure 3.7: The trajectories of the L_1 distances between the estimated likelihood functions and the joint likelihood function for the ten sensing agents are shown.

threat to functional spacecraft and satellites in orbit. The US has established the Space Surveillance Network (SSN) for ground based observations of the orbital debris using radars and optical telescopes [178, 179]. In February 2009, the Iridium–33 satellite collided with the Kosmos–2251 satellite and a large number of debris fragments were created. In this section, we use the Hierarchical BCF Algorithm to track one of the Iridium–33 debris created in this collision. The orbit of this debris around Earth and the location of SSN sensors are shown in Fig. 3.9.

The actual two-line element set (TLE) of the Iridium–33 debris was accessed from North American Aerospace Defense Command (NORAD) on 4$^{\text{th}}$ Dec 2013. The nonlinear Simplified General Perturbations (SGP4) model, which uses an extensive gravitational model and accounts for the drag effect on mean motion [180, 181], is used as the target dynamics model. The communication network topology of the SSN is assumed to be a static SC balanced graph, as shown in Fig. 3.9. If the debris is visible above the sensor's horizon, then it is assumed to create a single measurement during each time step of one minute. The

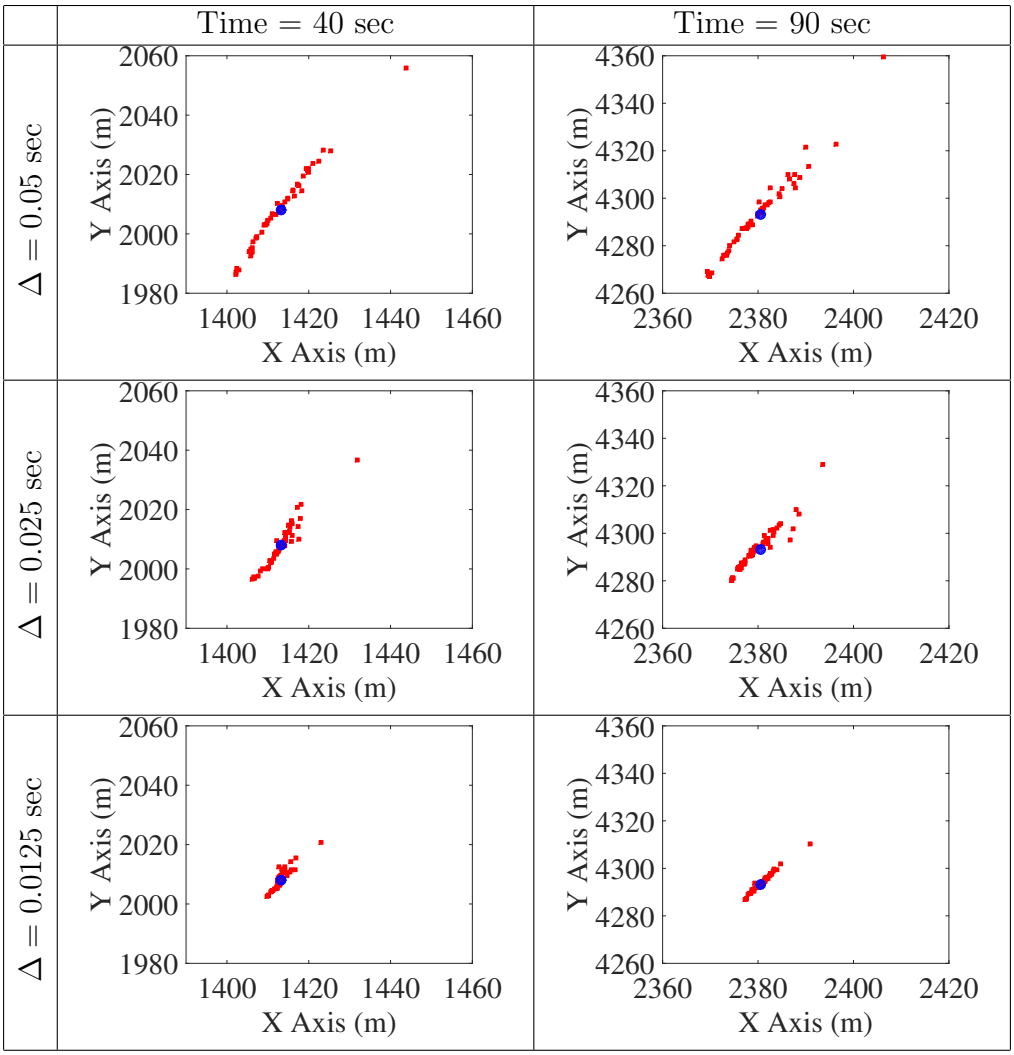


Figure 3.8: The estimated means of all the agents executing the DBF algorithm for linear-Gaussian models are shown, where ● is the actual position of the target and ■ is the mean of the estimated pdf.

heterogeneous measurement model of the j^{th} sensor is given by:

$$\boldsymbol{y}_k^j = \boldsymbol{x}_k + \boldsymbol{w}_k^j, \text{ where } \boldsymbol{w}_k^j = \mathcal{N}\left(0, (1000 + 50j) \times \mathbf{I}\right),$$

where $\boldsymbol{x}_k \in \mathbb{R}^3$ is the actual location of the debris and the additive Gaussian measurement noise depends on the sensor number. We compare the performance of our Hierarchical BCF algorithm against the Hierarchical BCF–LinOP algorithm, where the LinOP is used during the consensus stage.

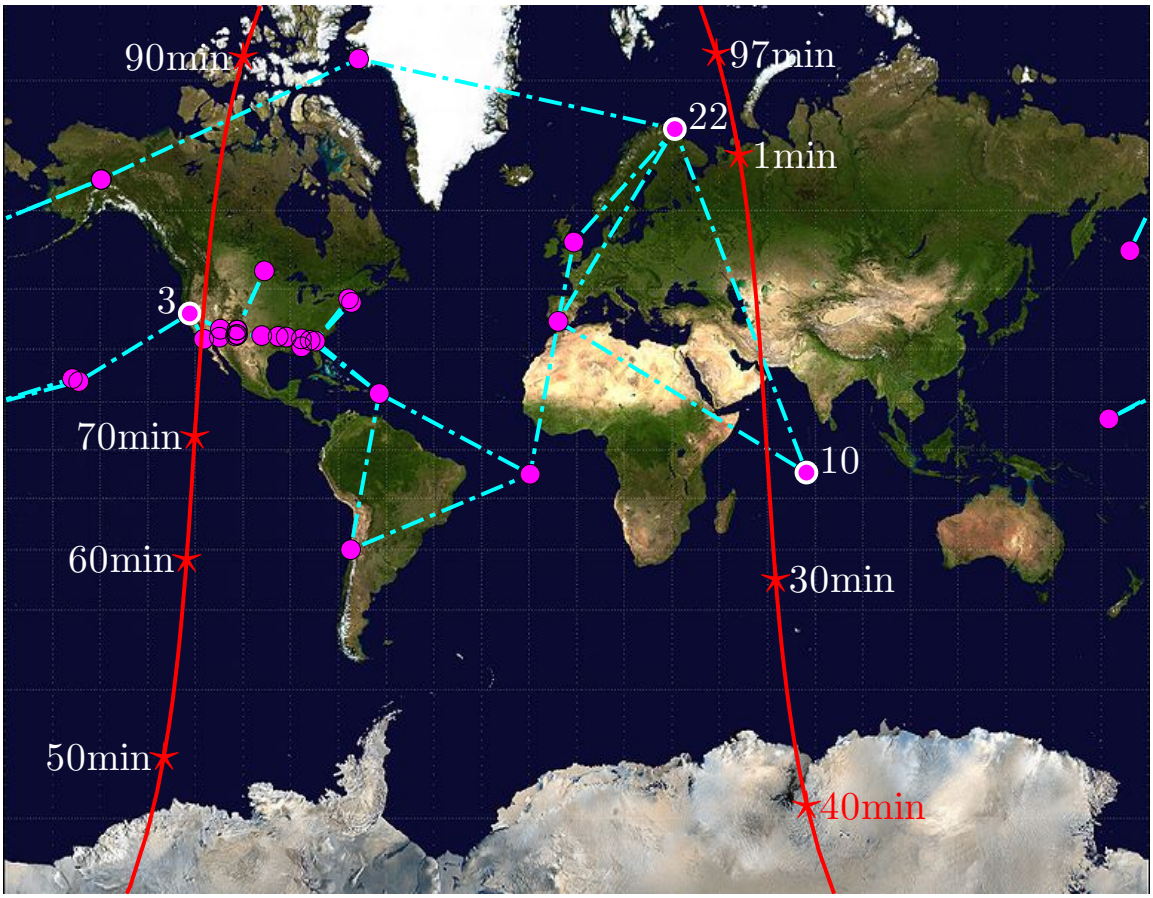


Figure 3.9: The SSN locations are shown along with their static SC balanced communication network topology. The orbit of the Iridium–33 debris is shown in red, where ⋆ marks its actual position during particular time instants.

In this simulation example, we simplify the debris tracking problem by assuming only the mean motion (n) of the debris is unknown. The objective of this simulation example is to estimate n of the Iridium–33 debris within 100 minutes. Hence, each sensor knows the other TLE parameters of the debris and an uniform prior distribution is assumed. Note that at any time instant, only a few of the SSN sensors can observe the debris, as shown in Fig 3.10(a). The results of three stand-alone Bayesian filters, implemented using particle filters with resampling [160], are shown in Fig 3.10(b-d). Note that the estimates of the 22^{nd} and 10^{th} sensors initially do not converge due to large measurement error, in spite of observing the debris for some time. The estimates of the 3^{rd} sensor does converge when it is able to observe the debris after 70 minutes. Hence we propose to use the Hierarchical

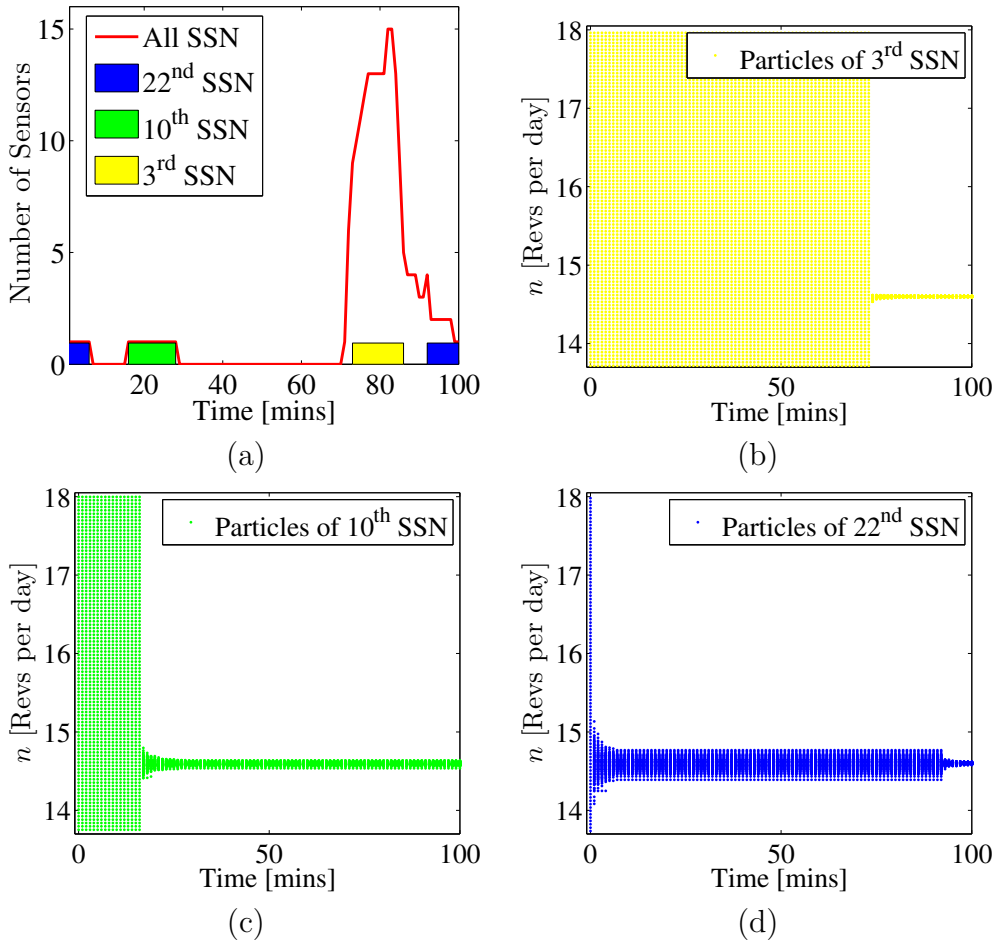


Figure 3.10: (a) Number of SSN sensors observing debris. Trajectories of particles for stand-alone Bayesian filters for (b) 3rd, (c) 10th, and (d) 22nd SSN sensor.

BCF algorithm where the consensual distribution is updated as and when sensors observe the debris.

Particle filters with resampling are used to evaluate the Bayesian filters and communicate pdfs in the Hierarchical BCF algorithms. 100 particles are used by each sensor and 10 consensus loops are executed during each time step of one minute. The trajectories of all the particles of the sensors in the Hierarchical BCF algorithm using LinOP and LogOP and their respective consensual probability distributions at different time instants are shown in Fig. 3.10(a-d). As expected, all the sensors converge on the correct value of n of 14.6 revs per day. The Hierarchical BCF–LinOP estimates are multimodal for the first 90 minutes.

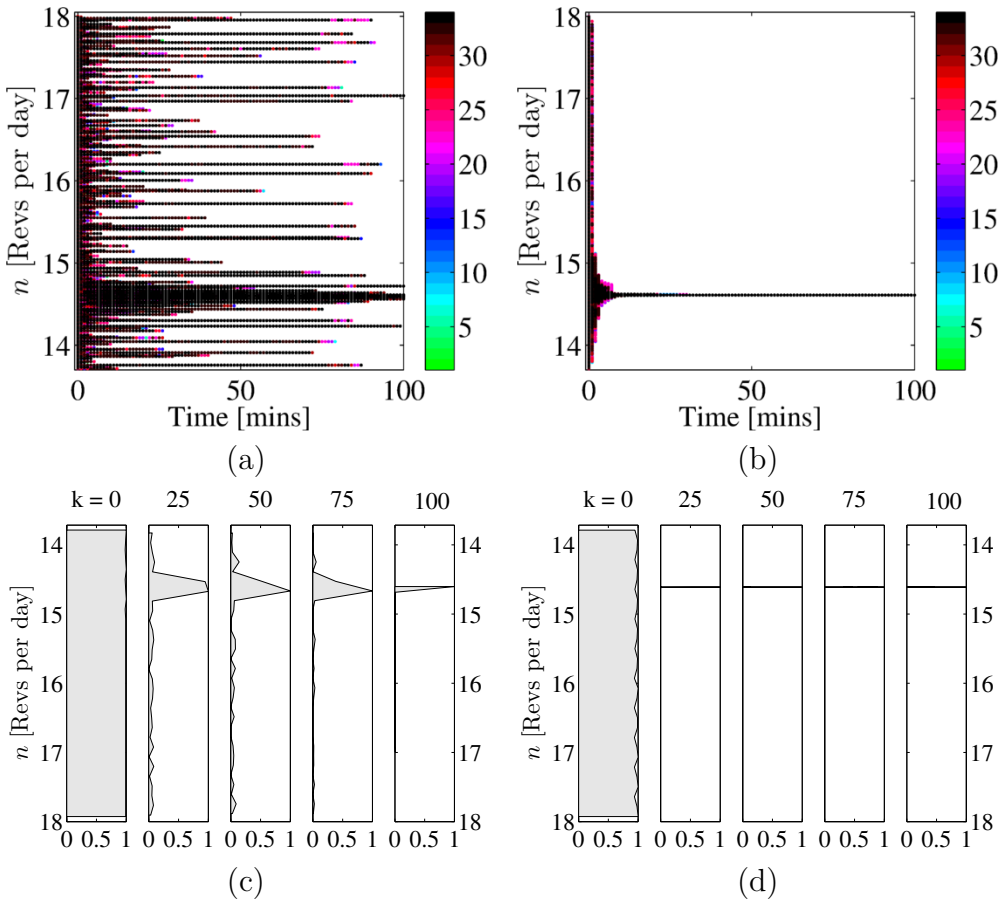


Figure 3.11: Trajectories of particles of all sensors for (a) Hierarchical BCF–LinOP and (b) Hierarchical BCF–LogOP. The color-bar on the right denotes the 33 SSN sensors. Evolution of the consensual probability distribution for (c) Hierarchical BCF–LinOP and (d) Hierarchical BCF–LogOP.

On the other hand, the Hierarchical BCF estimates converges to the correct value within the first 10 minutes because the LogOP algorithm efficiently communicates the best consensual estimate to other sensors during each time step and achieves consensus across the network.

3.6 Chapter Summary

In this chapter, we presented distributed estimation algorithms and rigorously analyzed their convergence and robustness. We illustrated the properties of the proposed algorithms using numerical simulations. The novel proof techniques presented in this chapter can also be used

in other distributed estimation algorithms which rely on the LogOP scheme.

Chapter 4

Attitude Control of Spacecraft with a Captured Object

In this chapter, we present novel control algorithms for stabilizing the attitude of a spacecraft that has captured a significantly larger object. The algorithms presented in this chapter have been published in [182, 183, 184].

This chapter is organized as follows. Section 4.1 discusses the ARM mission, the problem statement, and some preliminaries. The new robust nonlinear tracking control law and its extensions are presented in Section 4.2. The design of desired attitude trajectories is discussed in Section 4.3. Numerical simulations are presented in Section 4.4 and this chapter is concluded in Section 4.5.

4.1 Preliminaries and Problem Statement

In this section, we first present a brief introduction of the conceptual ARM spacecraft and potential NEO asteroid targets in Section 4.1.1. We then present the attitude kinematics and dynamics of the combined system in Section 4.1.2 and state the main control problem in Section 4.1.3.

4.1.1 Conceptual ARM Spacecraft Design and NEO Asteroid Parameters

According to some preliminary feasibility studies [1, 185, 186], the most desirable asteroids are the carbonaceous C–type asteroids because samples from these asteroids can return to Earth without any restriction. The densities of asteroids can range from 1 g cm^{-3} for a

high-porosity carbonaceous chondrite to 8 g cm^{-3} for solid nickel-iron meteorites, but the majority of NEO asteroids have densities between 1.9-3.8 g cm^{-3}. It is estimated that C–type NEO asteroids with 7-10 m diameter would be in the 2.5-13 $\times 10^5$ kg mass range. Table 4.1 lists the nominal range of NEO asteroid parameters that the ARM spacecraft should be capable of handling.

Table 4.1: Nominal range of NEO asteroid parameters considered for the ARM mission [1]

Parameter	Range
Mass (m_{ast})	2.5-13 $\times 10^5$ kg
Density (ρ_{ast})	1-8 g cm^{-3} (majority within 1.9-3.8 g cm^{-3})
Diameter	2-10 m
Initial Angular Velocity ($\boldsymbol{\omega}_{\text{initial}}$)	$\leq$ 0.5 rotations per minute (rpm)
Initial Attitude ($\boldsymbol{\beta}_{\text{initial}}$)	any attitude ($\boldsymbol{\beta}_{\text{initial}} \in \mathbb{S}^3$)

The constraint on the initial angular velocity (spin rate) of the tumbling asteroid results from the technological capability of the sensors and angular momentum capacity of the actuators to be used on board the ARM spacecraft. In order to generate realistic models of asteroids for numerical simulations, we use the shape models of the asteroids 433 Eros [6] and 25143 Itokawa [7] shown in Fig. 4.1(a,b). The diameters of these realistic asteroid models are shown in Fig. 4.1(c,d). Fig. 4.1(c,d) also show the nominal range of NEO asteroid parameters considered for this ARM mission concept, i.e., the asteroid's mass is within 2.5-13 $\times 10^5$ kg, the asteroid's density is within 1.9-3.8 g cm^{-3}, and the asteroid's diameter varies in the range of 2-10 m. Asteroids with large mass and small density have large diameters and consequently have large moments of inertia. These realistic models of asteroids are used for Monte Carlo simulations in Section 4.4 for comparing the various control laws discussed in this chapter.

A conceptual design of the ARM spacecraft is shown in Fig. 4.2(a), which is used as the nominal ARM spacecraft design in this chapter. This conceptual ARM spacecraft, with dry mass of 5,500 kg and wet mass of 15,500 kg, could carry 13 $\times 10^3$ kg of Xenon propellant for the 40 kW SEP system and an additional 900 kg of liquid propellant for the roll control

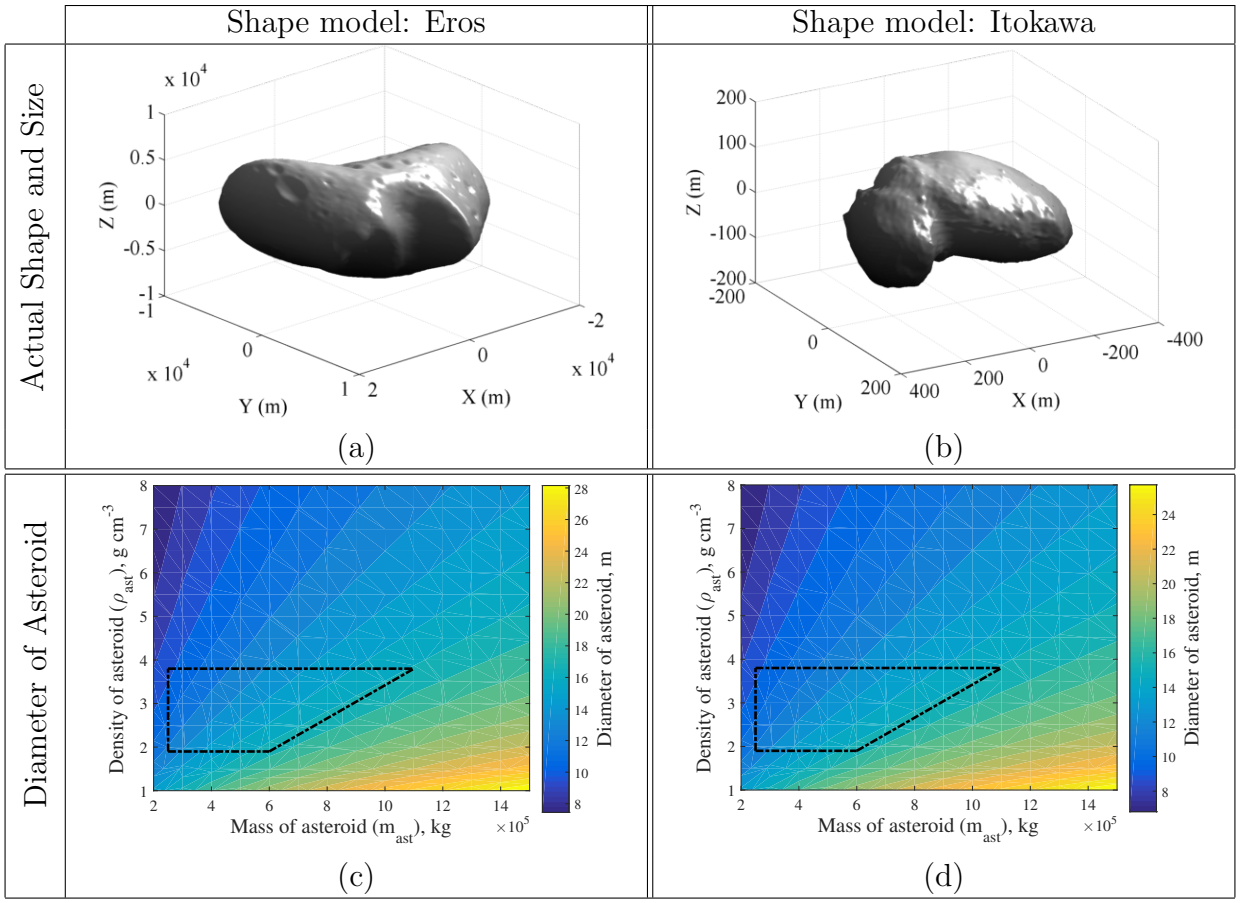
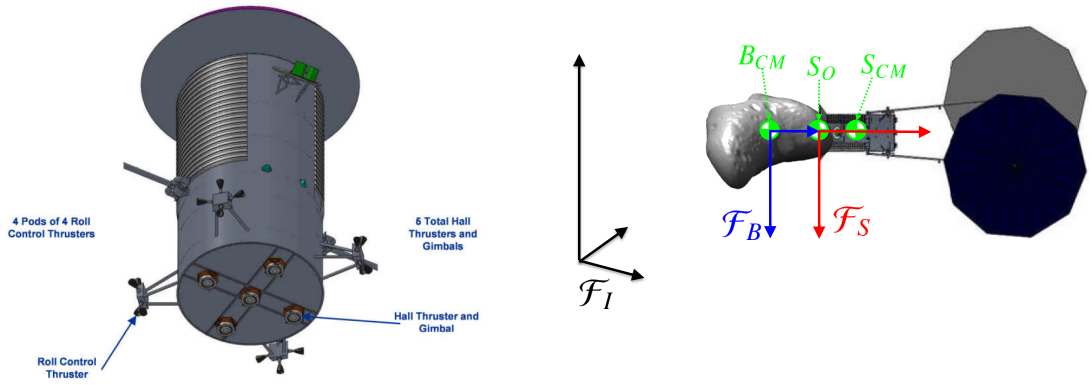


Figure 4.1: Shape models of asteroids (a) 433 Eros [6] and (b) 25143 Itokawa [7] are used for generating realistic models of asteroids. In (c,d), the diameters of realistic asteroid models are shown, which are obtained by sizing the shape models of Eros and Itokawa respectively for the given masses and densities. The inset black trapezium shows the nominal range of NEO asteroid parameters.

thrusters [187]. The various propulsion systems on board the conceptual ARM spacecraft are shown in Fig. 4.2(a). The SEP system includes five Hall thrusters and power processor units. The SEP system will be used by the conceptual ARM spacecraft for the journey to the chosen NEO asteroid and also for bringing the captured asteroid to the Earth–Moon system [188, 189]. Attitude control during the SEP thrusting stage will be achieved by gimballing the Hall thrusters.

The Reaction Control Subsystem (RCS) will be used for attitude control of the tumbling asteroid and spacecraft combination. The RCS uses hypergolic bipropellant, comprising of



(a) (b)

Figure 4.2: (a) Bottom view of the conceptual ARM spacecraft with the five Hall thrusters and the four RCS thruster pods (image credit: NASA [1]). (b) The conceptual ARM spacecraft [1] is shown with a captured asteroid. The inertial frame $\mathcal{F}_I$, the body fixed frame $\mathcal{F}_B$, and the spacecraft frame $\mathcal{F}_S$ are shown. B_{CM} is the center of mass of the combined system.

monomethylhydrazine and nitrogen tetroxide with a gaseous pressured nitrogen. The RCS includes four pods of four thrusters as shown in Fig. 4.2(a). Each thruster has a maximum thrust output of 200 N and the fuel's specific impulse is 287 sec. The position, direction, and control influence matrix of the thrusters in the spacecraft frame $\mathcal{F}_S$ (see Fig. 4.2(b)) are shown in Table 4.2, where an opposing pair of thrusters in a pod are represented by a single thruster capable of producing thrust between +200 N to −200 N. Note that during the capture and detumbling stage, the ARM-spacecraft's solar arrays will be folded back to facilitate the despinning process.

Table 4.2: RCS Thruster Position and Direction (generated from Fig. 4.2(a) and [1])

Thruster	Pod	Position of Thruster (m)	Direction of Thrust	Control Influence Matrix ($\boldsymbol{B}$)
u_1	P_1	$\boldsymbol{r}^{P_1/S_O} = 1.4\hat{s}_x + 5.9\hat{s}_z$	$\pm\hat{s}_y$	$\boldsymbol{B}(:,1) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_1/S_O}) \times (\hat{s}_y)$
u_2	P_1	$\boldsymbol{r}^{P_1/S_O} = 1.4\hat{s}_x + 5.9\hat{s}_z$	$\pm\hat{s}_z$	$\boldsymbol{B}(:,2) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_1/S_O}) \times (\hat{s}_z)$
u_3	P_2	$\boldsymbol{r}^{P_2/S_O} = 1.4\hat{s}_y + 5.9\hat{s}_z$	$\pm\hat{s}_x$	$\boldsymbol{B}(:,3) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_2/S_O}) \times (\hat{s}_x)$
u_4	P_2	$\boldsymbol{r}^{P_2/S_O} = 1.4\hat{s}_y + 5.9\hat{s}_z$	$\pm\hat{s}_z$	$\boldsymbol{B}(:,4) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_2/S_O}) \times (\hat{s}_z)$
u_5	P_3	$\boldsymbol{r}^{P_3/S_O} = -1.4\hat{s}_x + 5.9\hat{s}_z$	$\pm\hat{s}_y$	$\boldsymbol{B}(:,5) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_3/S_O}) \times (\hat{s}_y)$
u_6	P_3	$\boldsymbol{r}^{P_3/S_O} = -1.4\hat{s}_x + 5.9\hat{s}_z$	$\pm\hat{s}_z$	$\boldsymbol{B}(:,6) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_3/S_O}) \times (\hat{s}_z)$
u_7	P_4	$\boldsymbol{r}^{P_4/S_O} = -1.4\hat{s}_y + 5.9\hat{s}_z$	$\pm\hat{s}_x$	$\boldsymbol{B}(:,7) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_4/S_O}) \times (\hat{s}_x)$
u_8	P_4	$\boldsymbol{r}^{P_4/S_O} = -1.4\hat{s}_y + 5.9\hat{s}_z$	$\pm\hat{s}_z$	$\boldsymbol{B}(:,8) = (\boldsymbol{r}^{S_O/B_{CM}} + \boldsymbol{r}^{P_4/S_O}) \times (\hat{s}_z)$

4.1.2 Attitude Dynamics and Kinematics of the Combined System

In this section, we present the attitude dynamics and kinematics equations that are used in this chapter. We assume that the combined spacecraft and captured object form a rigid body.

As shown in Fig. 4.2(b), the center of mass of the combined system (B_{CM}) is the origin of the body fixed frame $\mathcal{F}_B$. Let S_O, which is the point of contact between the spacecraft and the object, denote the origin of the spacecraft frame $\mathcal{F}_S$. We assume that attitude orientation of $\mathcal{F}_B$ with respect to $\mathcal{F}_I$ is the same as that of $\mathcal{F}_S$ with respect to $\mathcal{F}_I$; i.e., the rotation matrix from $\mathcal{F}_S$ to $\mathcal{F}_B$ is an identity matrix.

Attitude Dynamics with Uncertainty

Let $\boldsymbol{J}_{\text{obj}}^{B_{CM}}$ be the unknown, constant, positive-definite inertia tensor of the captured object at B_{CM} and expressed in $\mathcal{F}_B$. Let $\boldsymbol{J}_{\text{sc}}^{S_{CM}}$ be the known, constant, positive-definite inertia tensor of the spacecraft at the center of mass of the spacecraft (S_{CM}) and expressed in $\mathcal{F}_S$. Let $\boldsymbol{r}^{S_O/B_{CM}}$ denote the unknown vector from B_{CM} to S_O. The combined inertia tensor of the system at B_{CM}, expressed in $\mathcal{F}_B$, is determined using the parallel axis theorem:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}} = \boldsymbol{J}_{\text{obj}}^{B_{CM}} + \boldsymbol{J}_{\text{sc}}^{S_{CM}} + m_{\text{sc}} \left[\left(\boldsymbol{r}^{S_{CM}/B_{CM}}\right)^T \left(\boldsymbol{r}^{S_{CM}/B_{CM}}\right) \mathbf{I} - \left(\boldsymbol{r}^{S_{CM}/B_{CM}}\right) \left(\boldsymbol{r}^{S_{CM}/B_{CM}}\right)^T \right], \tag{4.1}$$

where $\boldsymbol{r}^{S_{CM}/B_{CM}} = \boldsymbol{r}^{S_{CM}/S_O} + \boldsymbol{r}^{S_O/B_{CM}}$, $\boldsymbol{r}^{S_{CM}/S_O}$ is the known vector from S_O to S_{CM}, m_{sc} is the mass of the spacecraft, and the rotation matrix from the spacecraft frame to the body frame is an identity matrix.

Let $\boldsymbol{\omega} \in \mathbb{R}^3$ be the angular velocity of the system in the body fixed frame $\mathcal{F}_B$ with respect to the inertial frame $\mathcal{F}_I$ and expressed in the frame $\mathcal{F}_B$. Let $\boldsymbol{u} \in \mathbb{R}^{n_t}$ be the outputs

of n_t actuators and $\boldsymbol{B} \in \mathbb{R}^{3\times n_t}$ be the corresponding control influence matrix. The attitude dynamics of the rigid combination is given by:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}} = \left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\boldsymbol{\omega}\right)\times\boldsymbol{\omega} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{d}_{\text{ext}}\,, \tag{4.2}$$

where $\boldsymbol{d}_{\text{ext}}$ represents the external torque acting on the system.

We now study the effect of modeling uncertainties in $\boldsymbol{J}_{\text{obj}}^{B_{CM}}$ and $\boldsymbol{r}^{S_O/B_{CM}}$, measurement errors in $\boldsymbol{\omega}$, and actuator errors in $\boldsymbol{u}$ on the attitude dynamics of the system (4.2). Let $\boldsymbol{J}_{\text{obj}}^{B_{CM}} = \hat{\boldsymbol{J}}_{\text{obj}}^{B_{CM}} + \Delta\boldsymbol{J}_{\text{obj}}^{B_{CM}}$, where $\hat{(\cdot)}$ and $\Delta(\cdot)$ refer to the known and unknown parts respectively. Similarly, let $\boldsymbol{r}^{S_O/B_{CM}} = \hat{\boldsymbol{r}}^{S_O/B_{CM}} + \Delta\boldsymbol{r}^{S_O/B_{CM}}$, $\boldsymbol{\omega} = \hat{\boldsymbol{\omega}} + \Delta\boldsymbol{\omega}$, and $\boldsymbol{u} = \hat{\boldsymbol{u}} + \Delta\boldsymbol{u}$. Due to these uncertainties, the combined inertia tensor is given by:

$$\begin{aligned}
\boldsymbol{J}_{\text{tot}}^{B_{CM}} &= \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} + \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\,, \qquad (4.3)\\
\text{where } \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} &= \hat{\boldsymbol{J}}_{\text{obj}}^{B_{CM}} + \boldsymbol{J}_{\text{sc}}^{S_{CM}} + m_{\text{sc}}[(\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})^T(\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})\mathbf{I}\\
&\quad - (\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})\times(\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})^T]\,,\\
\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}} &= \Delta\boldsymbol{J}_{\text{obj}}^{B_{CM}} + m_{\text{sc}}[2(\Delta\boldsymbol{r}^{S_O/B_{CM}})^T(\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})\mathbf{I}\\
&\quad - (\Delta\boldsymbol{r}^{S_O/B_{CM}})(\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})^T - (\boldsymbol{r}^{S_{CM}/S_O} + \hat{\boldsymbol{r}}^{S_O/B_{CM}})(\Delta\boldsymbol{r}^{S_O/B_{CM}})^T]\,.
\end{aligned}$$

Here we have neglected the second order error terms. Note that the terms $\Delta\boldsymbol{r}^{S_O/B_{CM}}$ and $\Delta\boldsymbol{J}_{\text{obj}}^{B_{CM}}$ appear independently in $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}$.

Similarly, the control influence matrix can be decomposed into $\boldsymbol{B} = \hat{\boldsymbol{B}} + \Delta\boldsymbol{B}$ because it

depends on $\boldsymbol{r}^{S_O/B_{CM}}$. Simplifying the dynamics of the system (4.2) gives:

$$(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} + \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}})\dot{\hat{\boldsymbol{\omega}}} - (\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} + \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}})\hat{\boldsymbol{\omega}} \times \hat{\boldsymbol{\omega}} = \boldsymbol{u}_c + \boldsymbol{d}_{\text{res}}\,, \tag{4.4}$$

where $\boldsymbol{u}_c = \hat{\boldsymbol{B}}\hat{\boldsymbol{u}}\,,$

$$\boldsymbol{d}_{\text{res}} = \left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} + \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\right)\Delta\boldsymbol{\omega} \times \hat{\boldsymbol{\omega}} + \left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} + \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\right)\hat{\boldsymbol{\omega}} \times \Delta\boldsymbol{\omega}$$
$$+ \Delta\boldsymbol{B}\hat{\boldsymbol{u}} + \hat{\boldsymbol{B}}\Delta\boldsymbol{u} - \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\Delta\dot{\boldsymbol{\omega}} + \boldsymbol{d}_{\text{ext}}\,.$$

Note that $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}$ is the only unknown parameter in the left hand side of Eq. (4.4). The remaining unknown terms are grouped into the resultant disturbance term $\boldsymbol{d}_{\text{res}}$ in Eq. (4.4). We use Eq. (4.4) to analyze the stability of control laws presented in this chapter.

Impact of Feed-Forward Cancellation on Resultant Disturbance Torque

In this section, we compare the resultant disturbance torques of different attitude tracking control laws. If a linear control law (e.g., proportional-derivative control) is used, then the resultant disturbance torque is given by $\boldsymbol{d}_{\text{res}}$ in Eq. (4.4). However, a linear control law does not achieve global exponential stability for attitude tracking. If the following feedback linearization based control law is used:

$$\boldsymbol{u}_c = \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}} \times \hat{\boldsymbol{\omega}} - \boldsymbol{K}_f(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)\,, \tag{4.5}$$

where $\boldsymbol{K}_f \in \mathbb{R}^{3\times3}$ is a positive definite constant matrix and $\boldsymbol{\omega}_r$ is is the desired reference trajectory, then this control law globally exponentially stabilizes the left-hand side of the following closed-loop system:

$$\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}(\dot{\hat{\boldsymbol{\omega}}} - \dot{\boldsymbol{\omega}}_r) + \boldsymbol{K}_f(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r) = \underbrace{\left[\boldsymbol{d}_{\text{res}} - \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\hat{\boldsymbol{\omega}}} - \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}} \times \hat{\boldsymbol{\omega}}\right]}_{\boldsymbol{d}_{\text{res},1}}\,, \tag{4.6}$$

and the term $\boldsymbol{d}_{\text{res},1}$ appears as a resultant disturbance torque.

Suppose there exists a control law that globally exponentially stabilizes the following closed-loop system despite the disturbance in the left-hand side of the equation:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}(\dot{\hat{\boldsymbol{\omega}}}-\dot{\boldsymbol{\omega}}_r)-(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}})\times(\hat{\boldsymbol{\omega}}-\boldsymbol{\omega}_r)+\boldsymbol{K}_f(\hat{\boldsymbol{\omega}}-\boldsymbol{\omega}_r)=\underbrace{\left[\boldsymbol{d}_{\text{res}}+\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r-(\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}})\times\boldsymbol{\omega}_r\right]}_{\boldsymbol{d}_{\text{res},2}}. \tag{4.7}$$

Then the term $\boldsymbol{d}_{\text{res},2}$ appears as a resultant disturbance torque. We show later that Eq. (4.7) is the closed loop system of our proposed nonlinear tracking control law.

Example 4.1. In Table 4.3, we now numerically compare the magnitude of some of the terms in the resultant disturbance torques $\boldsymbol{d}_{\text{res}}$ in Eq. (4.4), $\boldsymbol{d}_{\text{res},1}$ in Eq. (4.6), and $\boldsymbol{d}_{\text{res},2}$ in Eq. (4.7) using the following values:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}=10^6\times\begin{bmatrix}1.2652 & 0.4397 & 0.0015\\ 0.4397 & 3.8688 & 0.0002\\ 0.0015 & 0.0002 & 3.5440\end{bmatrix}\ \text{kg m}^2,\ \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}=10^6\times\begin{bmatrix}1&0&0\\0&3&0\\0&0&3\end{bmatrix}\ \text{kg m}^2,\ \|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\|_2\approx 10^6\text{kg m}^2,$$

$$\boldsymbol{\omega}=\begin{bmatrix}0.01 & 0.02 & 0.03\end{bmatrix}\ \text{rad sec}^{-1},\ \hat{\boldsymbol{\omega}}=\begin{bmatrix}0.010 & 0.020 & 0.0299\end{bmatrix}\ \text{rad sec}^{-1},\ \Delta\boldsymbol{\omega}=10^{-4}\times\begin{bmatrix}-0.44 & 0.09 & 0.91\end{bmatrix}\ \text{rad sec}^{-1},$$

and we have neglected $\dot{\hat{\boldsymbol{\omega}}}$ and $\dot{\boldsymbol{\omega}}_r$ because they are much smaller than $\hat{\boldsymbol{\omega}}$.

Table 4.3: Magnitude of some of the disturbance terms in the resultant disturbance torques

Disturbance term	Magnitude (ℓ_2-norm)	Disturbance term present in		
		$\boldsymbol{d}_{\text{res}}$ in Eq. (4.4)	$\boldsymbol{d}_{\text{res},1}$ in Eq. (4.6)	$\boldsymbol{d}_{\text{res},2}$ in Eq. (4.7)
$\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\hat{\boldsymbol{\omega}}$	372.8 Nm	✗	✓	✗
$\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\boldsymbol{\omega}_r$	depends on $\boldsymbol{\omega}_r$	✗	✗	✓
$\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\Delta\boldsymbol{\omega}\times\hat{\boldsymbol{\omega}}$	7.8 Nm	✓	✓	✓
$\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\Delta\boldsymbol{\omega}\times\hat{\boldsymbol{\omega}}$	6.3 Nm	✓	✓	✓
$\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\Delta\boldsymbol{\omega}$	10.1 Nm	✓	✓	✓
$\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\Delta\boldsymbol{\omega}$	7.3 Nm	✓	✓	✓

The magnitude of the disturbance term $(\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\hat{\boldsymbol{\omega}})$ in Table 4.3 is significantly larger than the magnitude of other disturbance terms because of unprecedentedly large modeling uncertainties in the captured object. Moreover, this resultant disturbance torque is so large that it is comparable to the maximum control torque that the spacecraft can generate. Hence,

Table 4.4: Properties of Attitude Representations (adapted from [2])

Attitude Representation	Range, Transformation	Global?	Unique?
Euler Angles	ϕ, θ, $\psi \in [-\pi, \pi]$	No (Singularity at $\theta = \pm\pi/2$)	No
Euler Axis of Rotation and Angle	$\boldsymbol{e} \in \mathbb{S}^2$, $\Phi \in [-\pi, \pi]$	Yes	No ($\boldsymbol{e}^S = -\boldsymbol{e}$, $\Phi^S = 2\pi - \Phi$)
Quaternions $\boldsymbol{\beta} \in \mathbb{S}^3$,	$\beta_i = e_i \sin\frac{\Phi}{2}, i \in \{1,2,3\}$, $\beta_4 = \cos\frac{\Phi}{2}$	Yes	No ($\boldsymbol{\beta}^S = -\boldsymbol{\beta}$)
Classical Rodrigues Parameters $\boldsymbol{\sigma} \in \mathbb{R}^3$	$\boldsymbol{\sigma} = \boldsymbol{e} \tan\frac{\Phi}{2}$	No (Singularity at $\Phi = \pm\pi$)	Yes (When $\Phi \neq \pm\pi$)
Modified Rodrigues Parameters $\boldsymbol{q} \in \mathbb{R}^3$	$\boldsymbol{q} = \boldsymbol{e} \tan\frac{\Phi}{4}$	No (Singularity at $\Phi = \pm 2\pi$)	No ($\boldsymbol{q}^S = -\boldsymbol{e} \tan\frac{2\pi-\Phi}{4}$)
Rotation Matrix $\boldsymbol{R} \in \mathrm{SO}(3)$, $\det(\boldsymbol{R}) = 1$	$\boldsymbol{R}\boldsymbol{R}^T = \mathbf{I}$, $\boldsymbol{R}^T\boldsymbol{R} = \mathbf{I}$, $\boldsymbol{R} = \mathbf{I}\cos\Phi + \boldsymbol{S}(\boldsymbol{e})\sin\Phi + (1-\cos\Phi)\boldsymbol{e}\boldsymbol{e}^T$	Yes	Yes

control laws that have the disturbance term $(\Delta \boldsymbol{J}_{\text{tot}}^{B_{CM}} \hat{\boldsymbol{\omega}} \times \hat{\boldsymbol{\omega}})$ in their resultant disturbance torque (like the feed-forward cancellation based control law Eq. (4.6)) are not suitable for this control problem.

Clearly, $\boldsymbol{d}_{\text{res}}$ is the smallest resultant disturbance torque because it does not contain the terms $(\Delta \boldsymbol{J}_{\text{tot}}^{B_{CM}} \hat{\boldsymbol{\omega}} \times \hat{\boldsymbol{\omega}})$ and $(\Delta \boldsymbol{J}_{\text{tot}}^{B_{CM}} \hat{\boldsymbol{\omega}} \times \boldsymbol{\omega}_r)$. The magnitude of the resultant disturbance torque $\boldsymbol{d}_{\text{res},2}$ depends on $\boldsymbol{\omega}_r$, which can be made smaller than $\hat{\boldsymbol{\omega}}$, since it depends on the desired attitude trajectory. □

We show later that our proposed nonlinear tracking control law makes use of smaller resultant disturbance torque $\boldsymbol{d}_{\text{res},2}$ while retaining the superior tracking performance.

Attitude Representation and Kinematics

The attitude orientation of the body frame $\mathcal{F}_B$ with respect to the inertial frame $\mathcal{F}_I$ can be represented by various attitude representations as shown in Table 4.4. An attitude representation is global if it can represent any possible orientation. The attitude representation is unique if there is only one attitude state for every possible orientation. In Table 4.4, $(\cdot)^S$ denotes the shadow point representation of the same attitude. Note that classical Rodrigues

parameters are unique [190] (when $\Phi \neq \pm\pi$) because $\boldsymbol{\sigma}^S = -\boldsymbol{e}\tan\frac{2\pi-\Phi}{2} = \boldsymbol{e}\tan\frac{\Phi}{2} = \boldsymbol{\sigma}$.

The attitude kinematics of the rigid combination using quaternions ($\boldsymbol{\beta}_v = [\beta_1, \beta_2, \beta_3]$), modified Rodrigues parameters (MRP), and rotation matrix on SO(3) are given respectively by [115, 191, 192]:

$$\dot{\boldsymbol{\beta}}_v = \frac{1}{2}(\beta_4\boldsymbol{\omega} + \boldsymbol{\beta}_v \times \boldsymbol{\omega})\,, \qquad \dot{\beta}_4 = -\frac{1}{2}\boldsymbol{\beta}_v^T\boldsymbol{\omega}\,, \tag{4.8}$$

$$\dot{\boldsymbol{q}} = \boldsymbol{Z}(\boldsymbol{q})\boldsymbol{\omega}\,, \quad \text{where} \quad \boldsymbol{Z}(\boldsymbol{q}) = \frac{1}{2}\left[\mathbf{I}\left(\frac{1-\boldsymbol{q}^T\boldsymbol{q}}{2}\right) + \boldsymbol{q}\boldsymbol{q}^T + \boldsymbol{S}(\boldsymbol{q})\right]\,, \quad \boldsymbol{S}(\boldsymbol{q}) = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix}\,, \tag{4.9}$$

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\boldsymbol{S}(\boldsymbol{\omega})\,. \tag{4.10}$$

The attitude kinematics equations using Euler angles (ϕ, θ, ψ), classical Rodrigues parameters ($\boldsymbol{\sigma}$), and the first three elements of a quaternion vector ($\boldsymbol{\beta}_v$) can also be written in the form of $\dot{\boldsymbol{q}} = \boldsymbol{Z}(\boldsymbol{q})\boldsymbol{\omega}$ (like Eq. (4.9)) with a different definition of $\boldsymbol{Z}(\boldsymbol{q})$ [193]. We show later that our proposed nonlinear control law permits the use of any attitude representation.

4.1.3 Problem Statement: Attitude Control of the Combined System

The salient features of the attitude control problem discussed in this chapter are as follows: (i) The rigid combined system, comprising the spacecraft and the captured object, is tumbling. The tumbling rate can be nonuniform due to the cross-terms in the moment of inertia tensor. (ii) The object's inertia tensor, mass, center of mass, and center of gravity have large uncertainties (approximately 10% of the nominal value). (iii) The object is non-collaborative; i.e., no actuators are placed on the object. All actuators are on board the spacecraft.

Let $\boldsymbol{q}_{\text{final}}$ denote the desired attitude orientation of the stabilized system. The attitude control objective is to stabilize the system, in the presence of uncertain physical parameters,

bounded disturbances, measurement errors, and actuator saturations, such that for some appropriate $\varepsilon_{\text{trans}} > 0$, $\varepsilon_{\text{ss}} > 0$, and $T \gg 0$:

$$\|\boldsymbol{\omega}(t)\|_2 \leq \varepsilon_{\text{trans}}, \qquad \forall t > 0\,, \tag{4.11}$$

$$\|\boldsymbol{q}(t) - \boldsymbol{q}_{\text{final}}\|_2 \leq \varepsilon_{\text{ss}}, \qquad \forall t > T\,. \tag{4.12}$$

The transient error bound $\varepsilon_{\text{trans}}$ is imposed on the angular velocity $\boldsymbol{\omega}(t)$ in Eq. (4.11) in order to ensure that the system is always within the technological capability of the sensors and actuators on board the spacecraft. It is desired that after time T, the system should achieve the desired attitude orientation $\boldsymbol{q}_{\text{final}}$ as shown in the steady-state condition (4.12). Note that if the system has to hold its attitude within the given steady-state error bound ε_{ss}, then the desired angular velocity $\boldsymbol{\omega}_{\text{final}}$ of the stabilized system should be sufficiently close to $\mathbf{0}$ rad sec^{-1}.

In this chapter, a control law that guarantees global exponential convergence or a contracting closed-loop dynamics in the sense of Lemma C.1 is derived to achieve the objectives in Eqs. (4.11-4.12) by using Lemma C.2 (see Appendix C). Hence, in the presence of disturbances, such a globally exponentially stabilizing control law yields finite-gain $\mathcal{L}_p$ stability and input-to-state stability [194]. If a control law that only yields global asymptotic convergence (without any disturbance), then the error in the system's trajectory may not be bounded for a certain class of disturbance and proving robustness is more involved [194].

4.2 Control Laws for Nonlinear Attitude Control

In this section, we present the new nonlinear attitude tracking control laws that are deemed suitable for satisfying the control problem statement. We first present a novel robust nonlinear tracking control law that guarantees globally exponential convergence of the system's attitude trajectory to the desired attitude trajectory. In order to highlight the advantages of

this new control law, we also present several extensions of this attitude tracking control law, like augmenting it with an integral control term and deriving an exponentially-stabilizing tracking control law on SO(3).

4.2.1 Robust Nonlinear Tracking Control Law with Global Exponential Stability

The following theorem states the proposed robust nonlinear tracking control law. Note that this control law does not cancel the term $\boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\hat{\boldsymbol{\omega}}$ exactly, in contrast with most conventional nonlinear tracking control laws using feed-forward cancellation. Although this control law is written for MRP, it can also be used with other attitude representations like Euler angles, classical Rodrigues parameters, and the quaternion vector, by changing the definition of $\boldsymbol{Z}(\boldsymbol{q})$.

Theorem 4.1. *For the given desired attitude trajectory $\boldsymbol{q}_d(t)$, and positive definite constant matrices $\boldsymbol{K}_r \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Lambda}_r \in \mathbb{R}^{3\times3}$, we define the following control law:*

$$\begin{aligned} \boldsymbol{u}_c &= \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_r - \boldsymbol{K}_r(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)\,, \\ \textit{where}\quad \boldsymbol{\omega}_r &= \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\dot{\boldsymbol{q}}_d(t) + \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\boldsymbol{\Lambda}_r(\boldsymbol{q}_d(t) - \hat{\boldsymbol{q}})\,. \end{aligned} \tag{4.13}$$

This control law stabilizes the combined system (4.4) with the following properties:
(i) In the absence of resultant disturbance torque $\boldsymbol{d}_{\text{res},2}$, this control law guarantees global exponential convergence of the system's trajectory to the desired trajectory $\boldsymbol{q}_d(t)$.
(ii) In the presence of bounded resultant disturbance torque $\boldsymbol{d}_{\text{res},2}$, this control law guarantees that the tracking error ($\boldsymbol{q}_e = \hat{\boldsymbol{q}} - \boldsymbol{q}_d$) globally exponentially converges to the following ball

$$\lim_{t\to\infty}\int_0^{\boldsymbol{q}_e}\|\delta\boldsymbol{q}_e\|_2 \le \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{\Lambda}_r)\lambda_{\min}(\boldsymbol{K}_r)\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}\left(\sup_t \sigma_{\max}(\boldsymbol{Z}(\hat{\boldsymbol{q}}))\right)\left(\sup_t \|\boldsymbol{d}_{\text{res},2}\|_2\right)\,.$$

Hence, this control law is finite-gain $\mathcal{L}_p$ stable and input-to-state stable (ISS), which are

sufficient conditions for satisfying the control problem statement Eqs. (4.11-4.12).

Proof: The closed-loop dynamics, which is obtained by substituting $\boldsymbol{u}_c$ from Eq. (4.13) into Eq. (4.4), becomes

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_e - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_e + \boldsymbol{K}_r\boldsymbol{\omega}_e = \underbrace{\left[\boldsymbol{d}_{\text{res}} + \Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \boldsymbol{S}\left(\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_r\right]}_{\boldsymbol{d}_{\text{res},2}}, \tag{4.14}$$

where $\boldsymbol{\omega}_e = (\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)$. We first show that the control law indeed globally exponentially stabilizes the closed-loop system without the resultant disturbance $\boldsymbol{d}_{\text{res},2}$. The virtual dynamics of $\boldsymbol{y}$, derived from Eq. (4.14) without $\boldsymbol{d}_{\text{res},2}$, is given as

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{y}} - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{y} + \boldsymbol{K}_r\boldsymbol{y} = \boldsymbol{0}\,, \tag{4.15}$$

where $\boldsymbol{y}$ has $\boldsymbol{y} = \boldsymbol{\omega}_e$ and $\boldsymbol{y} = \boldsymbol{0}$ as its two particular solutions. After we obtain the dynamics of the infinitesimal displacement at fixed time, $\delta\boldsymbol{y}$ from (4.15), we perform the squared-length analysis (see Appendix C):

$$\frac{d}{dt}\left(\delta\boldsymbol{y}^T\boldsymbol{J}_{\text{tot}}^{B_{CM}}\delta\boldsymbol{y}\right) = -2\delta\boldsymbol{y}^T\boldsymbol{K}_r\delta\boldsymbol{y} \leq \frac{-2\lambda_{\min}(\boldsymbol{K}_r)}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}\left(\delta\boldsymbol{y}^T\boldsymbol{J}_{\text{tot}}^{B_{CM}}\delta\boldsymbol{y}\right)\,, \tag{4.16}$$

where we exploited the skew-symmetric property of the matrix $\boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)$. Hence, it follows from the contraction analysis (Lemma C.1 in Appendix C) that all system trajectories of (4.15) converge exponentially fast to a single trajectory (i.e., $\delta\boldsymbol{y} \to \boldsymbol{0}$ and $\boldsymbol{\omega}_e \to \boldsymbol{0}$) at a rate of $\frac{\lambda_{\min}(\boldsymbol{K}_r)}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}$.

In the presence of bounded resultant disturbance $\boldsymbol{d}_{\text{res},2}$, it follows from Lemma C.2 in Appendix C that:

$$\lim_{t\to\infty}\int_0^{\boldsymbol{\omega}_e}\|\delta\boldsymbol{y}\|_2 \leq \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{K}_r)\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}\sup_t\|\boldsymbol{d}_{\text{res},2}\|_2 \tag{4.17}$$

Hence the dynamics of the closed-loop system is bounded in the presence of bounded resul-

tant disturbance $\boldsymbol{d}_{\text{res},2}$. We now prove that convergence of $\boldsymbol{\omega}_e \to \mathbf{0}$ implies convergence of the system's trajectory to the desired trajectory ($\hat{\boldsymbol{q}} \to \boldsymbol{q}_d$). It follows from the definition of $\boldsymbol{\omega}_r$ that:

$$\boldsymbol{\omega}_e = \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})(\dot{\hat{\boldsymbol{q}}} - \dot{\boldsymbol{q}}_d) + \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\boldsymbol{\Lambda}_r(\hat{\boldsymbol{q}} - \boldsymbol{q}_d) = \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})(\dot{\boldsymbol{q}}_e + \boldsymbol{\Lambda}_r \boldsymbol{q}_e)\,, \tag{4.18}$$

where $\boldsymbol{q}_e = (\hat{\boldsymbol{q}} - \boldsymbol{q}_d)$. In the absence of $\boldsymbol{\omega}_e$, all system trajectories of $\delta\boldsymbol{q}_e$ will converge exponentially fast to a single trajectory ($\delta\boldsymbol{q}_e \to 0$) with a rate of $\lambda_{\min}(\boldsymbol{\Lambda}_r)$, where the virtual displacement $\delta\boldsymbol{q}_e$ is an infinitesimal displacement at fixed time. In the presence of $\boldsymbol{\omega}_e$, it follows from Lemma C.2 in Appendix C that:

$$\begin{aligned}\lim_{t\to\infty} \int_0^{\boldsymbol{q}_e} \|\delta\boldsymbol{q}_e\|_2 &\leq \frac{1}{\lambda_{\min}(\boldsymbol{\Lambda}_r)} \sup_t \|\boldsymbol{Z}(\hat{\boldsymbol{q}})\boldsymbol{\omega}_e\|_2 \\ &\leq \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{\Lambda}_r)\lambda_{\min}(\boldsymbol{K}_r)\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})} \left(\sup_t \sigma_{\max}(\boldsymbol{Z}(\hat{\boldsymbol{q}}))\right) \left(\sup_t \|\boldsymbol{d}_{\text{res},2}\|_2\right).\end{aligned} \tag{4.19}$$

Hence we have shown, by constructing a hierarchically-combined closed-loop system of $\boldsymbol{\omega}_e$ and $\boldsymbol{q}_e$, that the attitude trajectory $\boldsymbol{q}$ will globally exponentially converge to a bounded error ball around the desired trajectory $\boldsymbol{q}_d(t)$. Moreover, it follows from Lemma C.2 in Appendix C that this control law is finite-gain $\mathcal{L}_p$ stable and input-to-state stable. Hence the control gains $\boldsymbol{K}_r$ and $\boldsymbol{\Lambda}_r$ can be designed such that the error bounds $\varepsilon_{\text{trans}}$ and ε_{ss} in Eqs. (4.11-4.12) are satisfied. ■

The desired attitude trajectory $\boldsymbol{q}_d(t)$ can be any reference trajectory that we would like the system to track. We discuss methods for designing these desired attitude trajectories in Section 4.3.

4.2.2 Relation to Nonlinear Tracking Control using Euler-Lagrangian Systems

In this section, we compare the robust nonlinear tracking control law Eq. (4.13) with the well-known robust nonlinear tracking control for Euler-Lagrangian (EL) systems [117]. We first state the EL system with uncertainty, which is a combined representation of the attitude kinematics and dynamics of the system:

$$\hat{\boldsymbol{M}}(\hat{\boldsymbol{q}})\ddot{\hat{\boldsymbol{q}}} + \hat{\boldsymbol{C}}(\hat{\boldsymbol{q}},\dot{\hat{\boldsymbol{q}}})\dot{\hat{\boldsymbol{q}}} = \hat{\boldsymbol{\tau}}_c + \boldsymbol{\tau}_{\text{res}}\,, \tag{4.20}$$

where $\hat{\boldsymbol{\tau}}_c = \boldsymbol{Z}^{-T}(\hat{\boldsymbol{q}})\boldsymbol{u}_c\,, \qquad \hat{\boldsymbol{M}}(\hat{\boldsymbol{q}}) = \boldsymbol{Z}^{-T}(\hat{\boldsymbol{q}})\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\,,$

$$\hat{\boldsymbol{C}}(\hat{\boldsymbol{q}},\dot{\hat{\boldsymbol{q}}}) = -\boldsymbol{Z}^{-T}(\hat{\boldsymbol{q}})\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\dot{\boldsymbol{Z}}(\hat{\boldsymbol{q}})\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}}) - \boldsymbol{Z}^{-T}(\hat{\boldsymbol{q}})\,\boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\dot{\hat{\boldsymbol{q}}}\right)\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\,,$$

and $\boldsymbol{\tau}_{\text{res}}$ is the resultant disturbance torque acting on the EL system. Note that $\dot{\hat{\boldsymbol{M}}}(\hat{\boldsymbol{q}}) - 2\hat{\boldsymbol{C}}(\hat{\boldsymbol{q}},\dot{\hat{\boldsymbol{q}}})$ in Eq. (4.20) is a skew-symmetric matrix, and this property is essential to the stability proof. We use a slight modification of the original robust nonlinear tracking control law Eq. (4.13), which is given by:

$$\boldsymbol{u}_c = \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_r - \boldsymbol{Z}^{T}(\hat{\boldsymbol{q}})\boldsymbol{K}_\ell\boldsymbol{Z}(\hat{\boldsymbol{q}})(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r) \tag{4.21}$$

where $\boldsymbol{K}_\ell \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Lambda}_\ell \in \mathbb{R}^{3\times3}$ are positive definite constant matrices. Substituting $\boldsymbol{\omega}_r$ into (4.21), using the identity $\dot{\boldsymbol{Z}}^{-1}(\hat{\boldsymbol{q}}) = -\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\dot{\boldsymbol{Z}}(\hat{\boldsymbol{q}})\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})$, and multiplying both sides with $\boldsymbol{Z}^{-T}(\hat{\boldsymbol{q}})$ gives us:

$$\hat{\boldsymbol{\tau}}_c = \hat{\boldsymbol{M}}(\hat{\boldsymbol{q}})\ddot{\boldsymbol{q}}_r + \hat{\boldsymbol{C}}(\hat{\boldsymbol{q}},\dot{\hat{\boldsymbol{q}}})\dot{\boldsymbol{q}}_r - \boldsymbol{K}_\ell(\dot{\hat{\boldsymbol{q}}} - \dot{\boldsymbol{q}}_r)\,, \tag{4.22}$$

where $\dot{\boldsymbol{q}}_r = \dot{\boldsymbol{q}}_d(t) + \boldsymbol{\Lambda}_\ell(\boldsymbol{q}_d(t) - \hat{\boldsymbol{q}})$.

Remark 4.1. (*Advantages of (4.13) over the control law for EL systems (4.22)*): First, the control law for EL system (4.22) extensively uses the measured attitude $\hat{\boldsymbol{q}}$ and its rate $\dot{\hat{\boldsymbol{q}}}$ but

does not explicitly use the measured angular velocity $\hat{\boldsymbol{\omega}}$. Moreover, the matrices $\boldsymbol{Z}(\hat{\boldsymbol{q}})$ and $\boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})$, which might be susceptible to large fluctuations due to measurement errors in $\hat{\boldsymbol{q}}$, are used multiple times in Eq. (4.22). For example, the actual control input $\boldsymbol{u}_c$ depends on the computed control signal $\hat{\boldsymbol{\tau}}_c$ in Eq. (4.22) through the relation $\boldsymbol{u}_c = \boldsymbol{Z}^T(\hat{\boldsymbol{q}})\hat{\boldsymbol{\tau}}_c$ as shown in Eq. (4.20). On the other hand, the original control law (4.13) directly computes $\boldsymbol{u}_c$.

Second, as shall be seen in Sec. 4.2.3, the stability proof is constructed using a constant matrix $\boldsymbol{J}_{\text{tot}}^{B_{CM}}$, not the nonlinear matrix $\hat{\boldsymbol{M}}(\hat{\boldsymbol{q}})$, thereby allowing for an integral control formulation. Third, in Eqs. (4.21) and (4.22), the terms $\boldsymbol{Z}^T(\hat{\boldsymbol{q}})\boldsymbol{K}_\ell\boldsymbol{Z}(\hat{\boldsymbol{q}})$, $\hat{\boldsymbol{M}}(\hat{\boldsymbol{q}})$, and $\hat{\boldsymbol{C}}(\hat{\boldsymbol{q}},\dot{\hat{\boldsymbol{q}}})$ strongly couple the three axes motions using the highly non-diagonal, non-symmetric matrix $\boldsymbol{Z}(\hat{\boldsymbol{q}})$. This strong coupling of the three-axis rotational motions might be undesirable. For example, initially, there might be an error in only one axis, but this coupling will subsequently introduce errors in all three axes. Depending on the inertia matrix, this strong coupling of three-axis motions can be avoided in the proposed control law Eq. (4.13). □

4.2.3 Robust Nonlinear Tracking Control Law with Integral Control

Another benefit of the original robust nonlinear tracking control law Eq. (4.13) is that it can be augmented with an integral control term in a straight-forward manner to eliminate any constant external disturbance while ensuring exponential convergence of the system's attitude trajectory to the desired attitude trajectory.

Theorem 4.2. *For the given desired attitude trajectory $\boldsymbol{q}_d(t)$, positive definite constant matrices $\boldsymbol{K}_m \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Lambda}_m \in \mathbb{R}^{3\times3}$, and (possibly time-varying) uniformly positive definite diagonal matrix $\boldsymbol{K}_I(t) \in \mathbb{R}^{3\times3}$, we define the following control law:*

$$\boldsymbol{u}_c = \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_r - \boldsymbol{K}_m(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r) - \int_0^t \boldsymbol{K}_I(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)dt\,, \tag{4.23}$$

where $\boldsymbol{\omega}_r = \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\dot{\boldsymbol{q}}_d(t) + \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\boldsymbol{\Lambda}_m(\boldsymbol{q}_d(t) - \hat{\boldsymbol{q}})\,.$

This control law has the following properties:

(i) This control law guarantees global exponential convergence of the system's trajectory to $\boldsymbol{q}_d(t)$ *for any constant external disturbance (constant bias) acting on the system.*

(ii) In the presence of time-varying disturbance $\boldsymbol{d}_{\text{res},2}$ *with a bounded rate* $\dot{\boldsymbol{d}}_{\text{res},2}$*, this control law guarantees that* $\boldsymbol{q}(t)$ *will globally exponentially converge to an error ball around* $\boldsymbol{q}_d(t)$*, whose size is determined by* $\dot{\boldsymbol{d}}_{\text{res},2}$ *(i.e., finite-gain* $\mathcal{L}_p$ *stable and ISS with respect to disturbance inputs with bounded rates).*

Proof: The closed-loop dynamics is given by:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_e - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_e + \boldsymbol{K}_m\boldsymbol{\omega}_e + \int_0^t \boldsymbol{K}_I\boldsymbol{\omega}_e dt = \boldsymbol{d}_{\text{res},2}\,, \tag{4.24}$$

where $\boldsymbol{\omega}_e = (\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)$ and $\boldsymbol{d}_{\text{res},2}$ is defined in Eq. (4.14). We first show that this control law can eliminate a constant external disturbance, hence replacing $\boldsymbol{d}_{\text{res},2}$ in Eq. (4.24) with a constant disturbance term $\boldsymbol{d}_{\text{const}}$ gives us:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_e - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_e + \boldsymbol{K}_m\boldsymbol{\omega}_e + \int_0^t \boldsymbol{K}_I\boldsymbol{\omega}_e dt = \boldsymbol{d}_{\text{const}}\,. \tag{4.25}$$

Differentiating Eq. (4.25) with respect to time and setting $\dot{\boldsymbol{d}}_{\text{const}} = 0$, we get:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\ddot{\boldsymbol{\omega}}_e + \left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right)\dot{\boldsymbol{\omega}}_e + \left(\boldsymbol{K}_I - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\hat{\boldsymbol{\omega}}}\right)\right)\boldsymbol{\omega}_e = 0\,. \tag{4.26}$$

If we show that Eq. (4.26) is contracting, then we prove our claim (i) that the given control law can successfully eliminate any constant external disturbance acting on the system. In order to prove Eq. (4.26) is globally exponentially stable, we consider two cases which depend on the time-varying nature of the matrix $\boldsymbol{K}_I$.

We first consider the case where $\boldsymbol{K}_I$ is a constant positive definite diagonal matrix. The matrix $\boldsymbol{K}_I$ can be decomposed into $\boldsymbol{K}_I = \boldsymbol{K}_I^{\frac{1}{2}}\boldsymbol{K}_I^{\frac{1}{2}}$, where the matrix $\boldsymbol{K}_I^{\frac{1}{2}}$ is also a constant positive definite diagonal matrix. We introduce the term $\boldsymbol{y}_1$, where $\dot{\boldsymbol{y}}_1$ is defined

as $\dot{\boldsymbol{y}}_1 = \boldsymbol{K}_I^{\frac{1}{2}}\boldsymbol{\omega}_e$. Then we can write $\dot{\boldsymbol{\omega}}_e$ as:

$$\dot{\boldsymbol{\omega}}_e = -(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right)\boldsymbol{\omega}_e - (\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}}\boldsymbol{y}_1\,. \tag{4.27}$$

Note that differentiating Eq. (4.27) with respect to time and substituting $\dot{\boldsymbol{y}}_1$ gives us Eq. (4.26). Therefore, these equations can be written in matrix form as:

$$\begin{bmatrix}\dot{\boldsymbol{\omega}}_e\\ \dot{\boldsymbol{y}}_1\end{bmatrix} = \begin{bmatrix}-(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right) & -(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}}\\ \boldsymbol{K}_I^{\frac{1}{2}} & \boldsymbol{0}\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}_e\\ \boldsymbol{y}_1\end{bmatrix} = \boldsymbol{F}\begin{bmatrix}\boldsymbol{\omega}_e\\ \boldsymbol{y}_1\end{bmatrix}\,. \tag{4.28}$$

We define the positive definite matrix $\boldsymbol{\Xi} = \begin{bmatrix}\boldsymbol{J}_{\text{tot}}^{B_{CM}} & b\mathbf{I}\\ b\mathbf{I} & \mathbf{I}\end{bmatrix}$, where b is a constant between $0 < b < \lambda_{\max}^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})$. The symmetric matrix $(\boldsymbol{\Xi F})_{\text{sym}} = \frac{1}{2}\left((\boldsymbol{\Xi F}) + (\boldsymbol{\Xi F})^T\right)$ is given by:

$$(\boldsymbol{\Xi F})_{\text{sym}} = -\begin{bmatrix}\frac{\boldsymbol{K}_m+\boldsymbol{K}_m^T}{2} - b\boldsymbol{K}_I^{\frac{1}{2}} & \frac{b}{2}\left[(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right)\right]^T\\ \frac{b}{2}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right) & \frac{b}{2}\left((\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}} + \boldsymbol{K}_I^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\right)\end{bmatrix}\,.$$

The sufficient conditions for the matrix $(\boldsymbol{\Xi F})_{\text{sym}}$ to be negative definite are [195]:

$$-\frac{\boldsymbol{K}_m + \boldsymbol{K}_m^T}{2} + b\boldsymbol{K}_I^{\frac{1}{2}} < 0\,, \qquad -\frac{b}{2}\left((\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}} + \boldsymbol{K}_I^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\right) < 0\,, \tag{4.29}$$

$$\frac{\lambda_{\max}\left(-\frac{\boldsymbol{K}_m+\boldsymbol{K}_m^T}{2} + b\boldsymbol{K}_I^{\frac{1}{2}}\right)\lambda_{\max}\left(-\frac{b}{2}\left((\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}} + \boldsymbol{K}_I^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\right)\right)}{\sigma_{\max}^2\left(-\frac{b}{2}\left[(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right)\right]^T\right)} > 1\,. \tag{4.30}$$

Equation (4.29) is satisfied by $0 < b < \frac{\lambda_{\min}(\boldsymbol{K}_m+\boldsymbol{K}_m^T)}{2\lambda_{\max}(\boldsymbol{K}_I^{\frac{1}{2}})}$. Equation (4.30) is satisfied by $b < b_3$, where b_3 is given by:

$$b_3 = \frac{\lambda_{\max}\left(\frac{\boldsymbol{K}_m+\boldsymbol{K}_m^T}{2}\right)\lambda_{\min}\left((\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}} + \boldsymbol{K}_I^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\right)}{\frac{1}{2}\sigma_{\max}^2\left[(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\right)\right]^T + \lambda_{\min}\left(\boldsymbol{K}_I^{\frac{1}{2}}\right)\lambda_{\min}\left((\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\boldsymbol{K}_I^{\frac{1}{2}} + \boldsymbol{K}_I^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})^{-1}\right)}\,. \tag{4.31}$$

Therefore, the matrix $(\boldsymbol{\Xi F})_{\text{sym}}$ is negative definite if b is chosen such that $0 < b < \min(\lambda_{\max}^{\frac{1}{2}}(\boldsymbol{J}_{\text{tot}}^{B_{CM}}), \frac{\lambda_{\min}(\boldsymbol{K}_m+\boldsymbol{K}_m^T)}{2\lambda_{\max}(\boldsymbol{K}_I^{\frac{1}{2}})}, b_3)$. We define the generalized virtual displacement $\delta\boldsymbol{z} =$

$[\delta\boldsymbol{\omega}_e,\, \delta\boldsymbol{y}_1]^T$, where $\delta\boldsymbol{\omega}_e$ and $\delta\boldsymbol{y}_1$ are infinitesimal displacements at fixed time. Therefore,

$$\frac{d}{dt}\left(\delta\boldsymbol{z}^T\boldsymbol{\Xi}\delta\boldsymbol{z}\right) = \delta\boldsymbol{z}^T\left((\boldsymbol{\Xi}\boldsymbol{F}) + (\boldsymbol{\Xi}\boldsymbol{F})^T\right)\delta\boldsymbol{z} \leq 2\lambda_{\max}((\boldsymbol{\Xi}\boldsymbol{F})_{\text{sym}})\|\delta\boldsymbol{z}\|_2^2$$
$$\leq \frac{2\lambda_{\max}((\boldsymbol{\Xi}\boldsymbol{F})_{\text{sym}})}{\lambda_{\max}(\boldsymbol{\Xi})}\left(\delta\boldsymbol{z}^T\boldsymbol{\Xi}\delta\boldsymbol{z}\right). \tag{4.32}$$

Hence, it follows from the contraction analysis (Lemma C.1 in Appendix C) that all system trajectories converge exponentially fast to a single trajectory ($\delta\boldsymbol{z} \to 0$ and $\delta\boldsymbol{\omega}_e \to 0$) at a rate of $\frac{-\lambda_{\max}((\boldsymbol{\Xi}\boldsymbol{F})_{\text{sym}})}{\lambda_{\max}(\boldsymbol{\Xi})}$. Moreover, in the presence of bounded time-varying resultant disturbance $\boldsymbol{d}_{\text{res},2}$ with bounded $\dot{\boldsymbol{d}}_{\text{res},2}$, we get from Lemma C.2 in Appendix C:

$$\lim_{t\to\infty}\int_0^{\boldsymbol{\omega}_e}\|\delta\boldsymbol{\omega}_e\|_2 \leq \frac{(b+1)\lambda_{\max}(\boldsymbol{\Xi})}{-\lambda_{\max}((\boldsymbol{\Xi}\boldsymbol{F})_{\text{sym}})}\left(\sup_t \lambda_{\max}(\boldsymbol{K}_I^{-\frac{1}{2}})\right)\left(\sup_t\|\dot{\boldsymbol{d}}_{\text{res},2}\|_2\right). \tag{4.33}$$

where $\|\delta\boldsymbol{\omega}_e\|_2 \leq \|\delta\boldsymbol{z}\|_2$ and $\lambda_{\min}(\boldsymbol{\Xi}) > 1$ are used. Also, note that the disturbance term in the righthand side of Eq. (4.28) is $(\boldsymbol{0}; -\boldsymbol{K}_I^{-\frac{1}{2}}\dot{\boldsymbol{d}}_{\text{res},2})$. The fact that convergence of $\boldsymbol{\omega}_e \to \boldsymbol{0}$ implies convergence of the system's trajectory to the desired trajectory ($\hat{\boldsymbol{q}} \to \boldsymbol{q}_d$) is already presented in the proof of Theorem 4.1. This completes the proof.

If both $\boldsymbol{K}_I$ and $\dot{\boldsymbol{K}}_I$ are uniformly positive definite diagonal matrices, there exits a simpler proof, which is presented here. The matrix $\dot{\boldsymbol{K}}_I$ can also be decomposed into $\dot{\boldsymbol{K}}_I = \dot{\boldsymbol{K}}_I^{\frac{1}{2}}\dot{\boldsymbol{K}}_I^{\frac{1}{2}}$. We introduce another term $\boldsymbol{y}_2$, where:

$$\dot{\boldsymbol{y}}_2 = \boldsymbol{K}_I^{\frac{1}{2}}\boldsymbol{\omega}_e - \boldsymbol{K}_I^{-\frac{1}{2}}\dot{\boldsymbol{K}}_I^{\frac{1}{2}}\boldsymbol{y}_2\,. \tag{4.34}$$

Once again, $\dot{\boldsymbol{\omega}}_e$ can be written in a form similar to that of Eq. (4.27). The matrix form of these equations is given by:

$$\begin{bmatrix}\dot{\boldsymbol{\omega}}_e\\ \dot{\boldsymbol{y}}_2\end{bmatrix} = \begin{bmatrix}-(\boldsymbol{J}_{\text{tot}}^{BCM})^{-1}\left(\boldsymbol{K}_m - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{BCM}\hat{\boldsymbol{\omega}}\right)\right) & -(\boldsymbol{J}_{\text{tot}}^{BCM})^{-1}\boldsymbol{K}_I^{\frac{1}{2}}\\ \boldsymbol{K}_I^{\frac{1}{2}} & -\boldsymbol{K}_I^{-\frac{1}{2}}\dot{\boldsymbol{K}}_I^{\frac{1}{2}}\end{bmatrix}\begin{bmatrix}\boldsymbol{\omega}_e\\ \boldsymbol{y}_2\end{bmatrix} = \tilde{\boldsymbol{F}}\begin{bmatrix}\boldsymbol{\omega}_e\\ \boldsymbol{y}_2\end{bmatrix}. \tag{4.35}$$

Clearly, the symmetric part of the matrix $\boldsymbol{\Xi}\tilde{\boldsymbol{F}}$ is negative definite. Therefore,

$$\frac{d}{dt}\left(\delta \boldsymbol{z}^T \boldsymbol{\Xi} \delta \boldsymbol{z}\right) = \delta \boldsymbol{z}^T \left((\boldsymbol{\Xi}\tilde{\boldsymbol{F}}) + (\boldsymbol{\Xi}\tilde{\boldsymbol{F}})^T\right) \delta \boldsymbol{z} \leq 2\lambda_{\max}((\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}}) \|\delta \boldsymbol{z}\|_2^2 \leq \frac{2\lambda_{\max}((\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}})}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})} \left(\delta \boldsymbol{z}^T \boldsymbol{\Xi} \delta \boldsymbol{z}\right), \tag{4.36}$$

where $(\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}} = \frac{(\boldsymbol{\Xi}\tilde{\boldsymbol{F}}) + (\boldsymbol{\Xi}\tilde{\boldsymbol{F}})^T}{2}$. Also, $\lambda_{\max}((\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}}) < 0$ and is bounded as $\lambda_{\max}((\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}}) \leq -\min(\lambda_{\min}(\boldsymbol{K}_m), \inf_t(\lambda_{\min}(\boldsymbol{K}_I^{-\frac{1}{2}} \dot{\boldsymbol{K}}_I^{\frac{1}{2}})))$. Hence, it follows from the contraction analysis that all system trajectories converge exponentially fast to a single trajectory at a rate of $\frac{-\lambda_{\max}((\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}})}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}$. Moreover, in the presence of bounded $\boldsymbol{d}_{\text{res},2}$ and $\dot{\boldsymbol{d}}_{\text{res},2}$, we get from Lemma C.2 in Appendix C that:

$$\lim_{t\to\infty} \int_0^{\boldsymbol{\omega}_e} \|\delta \boldsymbol{\omega}_e\|_2 \leq \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{-\lambda_{\max}((\boldsymbol{\Xi}\tilde{\boldsymbol{F}})_{\text{sym}})} \left(\sup_t \lambda_{\max}(\boldsymbol{K}_I^{-\frac{1}{2}})\right) \left(\sup_t \|\dot{\boldsymbol{d}}_{\text{res},2}\|_2\right). \tag{4.37}$$

where $\|\delta \boldsymbol{\omega}_e\|_2 \leq \|\delta \boldsymbol{z}\|_2$ and $\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}}) > 1$ are used. Also, note that the disturbance term in the righthand side of Eq. (4.35) is $(\boldsymbol{0}; \boldsymbol{K}_I^{-\frac{1}{2}} \dot{\boldsymbol{d}}_{\text{res},2})$. ■

Remark 4.2. Note that the second block diagonal matrix of the Jacobin $\boldsymbol{F}$ in Eq. (4.28) is $\boldsymbol{0}$, which usually yields a semi-contracting system with global asymptotic stability. For example, $\boldsymbol{F}$ from Eq. (4.28) and $\boldsymbol{\Theta} = \begin{bmatrix} \boldsymbol{J}_{\text{tot}}^{B_{CM}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$ results in a semi-contracting system due to $\frac{1}{2}\left((\boldsymbol{\Theta}\boldsymbol{F}) + (\boldsymbol{\Theta}\boldsymbol{F})^T\right) = \begin{bmatrix} -\boldsymbol{K}_m & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}$. Similarly, the following adaptive control law also yields global asymptotic stability. In contrast, Theorem 4.2 presents a stronger result with global exponential stability. □

4.2.4 Nonlinear Adaptive Control

Let the parameter $\hat{\boldsymbol{a}}$ capture the six uncertain terms in the inertia tensor $\boldsymbol{J}_{\text{tot}}^{B_{CM}}$. The resulting adaptive nonlinear tracking control law and the tuning law are given by [117]:

$$\boldsymbol{u}_c = \boldsymbol{Y}\hat{\boldsymbol{a}} - \boldsymbol{K}_r(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)\,, \qquad \dot{\hat{\boldsymbol{a}}} = -\boldsymbol{\Gamma}_r \text{Proj}\left(\hat{\boldsymbol{a}}, \boldsymbol{Y}^T(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)\right)\,, \tag{4.38}$$

where $\boldsymbol{Y}\hat{\boldsymbol{a}} = \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_r$, $\boldsymbol{\omega}_r$ is defined in Eq. (4.13), and $\boldsymbol{\Gamma}_r \in \mathbb{R}^{6\times 6}$ is a positive-definite diagonal matrix. For some boundary function $f(\boldsymbol{\theta})$ (e.g., $f(\boldsymbol{\theta}) = \frac{(\boldsymbol{\theta}^T\boldsymbol{\theta} - \theta_{\max}^2)}{\epsilon_\theta \theta_{\max}^2}$), the projection operator is given by $\text{Proj}(\boldsymbol{\theta}, \boldsymbol{x}) = \boldsymbol{x} - \frac{\nabla f(\boldsymbol{x})\nabla f(\boldsymbol{\theta})^T}{\|\nabla f(\boldsymbol{\theta})\|^2}\boldsymbol{x} f(\boldsymbol{\theta})$ if $f(\boldsymbol{\theta}) > 0$, $\nabla f(\boldsymbol{\theta})^T\boldsymbol{x} > 0$; and $\boldsymbol{x}$ otherwise.

The proof of global asymptotic stability of using Eq. (4.38) for the disturbance-free system, derived from (4.4), is straightforward. The stability result of adaptive control is only globally asymptotic because its closed-loop system of the states $(\boldsymbol{\omega}_e, \hat{\boldsymbol{a}})^T$ yields a negative semidefinite Jacobian matrix $\left[\begin{smallmatrix} -\boldsymbol{K}_r & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{smallmatrix}\right]$ (also, see Eq. (4.28)). However, the use of a projection operator in Eq. (4.38) permits ISS, as shown in [12].

4.2.5 Robust Nonlinear Tracking Control Law on SO(3)

It is shown in Table 4.4 that the rotation matrix ($\boldsymbol{R} \in \text{SO}(3)$) is a global and unique attitude representation. In this section, we present a variation of Eq. (4.13) that exponentially stabilizes the attitude dynamics from almost all initial conditions on SO(3), i.e., all initial conditions except for those starting from a two-dimensional subset of SO(3).

It is shown in [196] that even global asymptotic convergence is not possible for any continuous feedback control law in SO(3). An almost-globally asymptotically stabilizing control law on SO(3) is discussed in [2]. In this chapter, we present a novel control law that guarantees exponential convergence to the desired trajectory for almost all initial conditions on SO(3). Another control law that also guarantees almost-global exponential convergence

is presented in [197], but our control law and proof techniques are substantially different from the Lyapunov-based approach used in [197].

Let $\boldsymbol{R}_d(t) \in \mathrm{SO}(3)$ denote the desired attitude trajectory, which is obtained from the desired attitude trajectory $\boldsymbol{q}_d(t)$ using the transformations given in Table 4.4. Let the inverse of the $\boldsymbol{S}(\cdot)$ map be the $\vee(\cdot)$ map, whose input is a skew-symmetric matrix and is defined as $\vee(\boldsymbol{S}(\boldsymbol{\omega})) = \boldsymbol{\omega}$. We now define the following notations [197]:

$$\boldsymbol{e}_{\hat{\boldsymbol{R}}} = \frac{1}{2\sqrt{1+\mathrm{tr}(\boldsymbol{R}_d^T\hat{\boldsymbol{R}})}}\left(\vee\left(\boldsymbol{R}_d^T\hat{\boldsymbol{R}} - \hat{\boldsymbol{R}}^T\boldsymbol{R}_d\right)\right), \qquad \boldsymbol{e}_{\hat{\boldsymbol{\omega}}} = \hat{\boldsymbol{\omega}} - \hat{\boldsymbol{R}}^T\boldsymbol{R}_d\left(\vee\left(\boldsymbol{R}_d^T\dot{\boldsymbol{R}}_d\right)\right), \tag{4.39}$$

where $\mathrm{tr}(\cdot)$ is the trace of the matrix. Here $\boldsymbol{e}_{\hat{\boldsymbol{R}}}$ represents the attitude error vector between the current measured attitude $\hat{\boldsymbol{R}}$ and the desired attitude $\boldsymbol{R}_d$. For any $\boldsymbol{R}_d^T\hat{\boldsymbol{R}}$, its trace is bounded by $-1 \leq \mathrm{tr}(\boldsymbol{R}_d^T\hat{\boldsymbol{R}}) \leq 3$. Hence $\boldsymbol{e}_{\hat{\boldsymbol{R}}}$ is not defined only on the two-dimensional subset of $\mathrm{SO}(3)$ where $\mathrm{tr}(\boldsymbol{R}_d^T\hat{\boldsymbol{R}}) = -1$, i.e., $\hat{\boldsymbol{R}} = \boldsymbol{R}_d\exp(\pm\pi\boldsymbol{S}(\boldsymbol{\kappa}))$, where $\boldsymbol{\kappa} \in \mathbb{S}^2$ [197]. Finally, we define the matrix $\boldsymbol{E}(\hat{\boldsymbol{R}}, \boldsymbol{R}_d)$ as follows [197]:

$$\frac{d\boldsymbol{e}_{\hat{\boldsymbol{R}}}}{dt} = \left[\frac{1}{2\sqrt{1+\mathrm{tr}(\boldsymbol{R}_d^T\hat{\boldsymbol{R}})}}\left(\mathrm{tr}(\hat{\boldsymbol{R}}^T\boldsymbol{R}_d)\mathbf{I} - \hat{\boldsymbol{R}}^T\boldsymbol{R}_d + 2\boldsymbol{e}_{\hat{\boldsymbol{R}}}\boldsymbol{e}_{\hat{\boldsymbol{R}}}^T\right)\right]\boldsymbol{e}_{\hat{\boldsymbol{\omega}}} = \boldsymbol{E}(\hat{\boldsymbol{R}}, \boldsymbol{R}_d)\boldsymbol{e}_{\hat{\boldsymbol{\omega}}}\,. \tag{4.40}$$

Theorem 4.3. *For the desired attitude trajectory $\boldsymbol{R}_d(t) \in \mathrm{SO}(3)$ and positive definite matrices $\boldsymbol{K}_e \in \mathbb{R}^{3\times3}$ and $\boldsymbol{\Lambda}_e \in \mathbb{R}^{3\times3}$, we define the following control law:*

$$\begin{aligned} \boldsymbol{u}_c &= \hat{\boldsymbol{J}}_{\mathrm{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r - \boldsymbol{S}\left(\hat{\boldsymbol{J}}_{\mathrm{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\boldsymbol{\omega}_r - \boldsymbol{K}_e(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)\,, \\ \textit{where} \quad \boldsymbol{\omega}_r &= \hat{\boldsymbol{R}}^T\boldsymbol{R}_d\left(\vee\left(\boldsymbol{R}_d^T\dot{\boldsymbol{R}}_d\right)\right) - \boldsymbol{\Lambda}_e\boldsymbol{E}^T(\hat{\boldsymbol{R}}, \boldsymbol{R}_d)\boldsymbol{e}_{\hat{\boldsymbol{R}}}\,. \end{aligned} \tag{4.41}$$

In the absence of disturbances or uncertainties, this control law guarantees exponential convergence of the system's trajectory $\boldsymbol{R}(t) \in \mathrm{SO}(3)$ to the desired trajectory $\boldsymbol{R}_d(t)$ for almost all initial conditions, i.e., all initial conditions that are not on the two-dimensional subset

of SO(3) where $\hat{\boldsymbol{R}}(0) = \boldsymbol{R}_d(0)\exp(\pm\pi \boldsymbol{S}(\boldsymbol{\kappa}))$*, where* $\boldsymbol{\kappa} \in \mathbb{S}^2$*. Moreover, in the presence of bounded disturbances or uncertainties, this control law guarantees that* $\boldsymbol{R}(t)$ *will exponentially converge to a bounded error ball around* $\boldsymbol{R}_d(t)$*.*

Proof: The closed-loop dynamics obtained by substituting $\boldsymbol{u}_c$ from Eq. (4.41) into Eq. (4.4) is the same as Eq. (4.14) in the proof of Theorem 4.1. Hence we can directly conclude from that proof that all system trajectories of $\boldsymbol{\omega}_e$ converge exponentially fast to a single trajectory ($\boldsymbol{\omega}_e \to 0$) at a rate of $\frac{\lambda_{\min}(\boldsymbol{K}_e)}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}$. Moreover, in the presence of bounded resultant disturbance $\boldsymbol{d}_{\text{res},2}$, $\lim_{t\to\infty} \int_0^{\boldsymbol{\omega}_e} \|\delta \boldsymbol{y}\|_2$ is bounded by Eq. (4.17).

Now we show that convergence of $\boldsymbol{\omega}_e$ implies convergence of the system's trajectory to the desired trajectory ($\boldsymbol{e}_{\hat{\boldsymbol{R}}} \to 0$). It follows from the definition of $\boldsymbol{\omega}_r$ that:

$$\boldsymbol{\omega}_e = \hat{\boldsymbol{\omega}} - \hat{\boldsymbol{R}}^T \boldsymbol{R}_d \left(\vee \left(\boldsymbol{R}_d^T \dot{\boldsymbol{R}}_d\right)\right) + \boldsymbol{\Lambda}_e \boldsymbol{E}^T \boldsymbol{e}_{\hat{\boldsymbol{R}}} = \boldsymbol{E}^{-1}\left(\dot{\boldsymbol{e}}_{\hat{\boldsymbol{R}}} + \boldsymbol{E}\boldsymbol{\Lambda}_e \boldsymbol{E}^T \boldsymbol{e}_{\hat{\boldsymbol{R}}}\right). \tag{4.42}$$

In the absence of $\boldsymbol{\omega}_e$, all system trajectories of $\delta \boldsymbol{e}_{\hat{\boldsymbol{R}}}$ will converge exponentially fast to a single trajectory ($\delta \boldsymbol{e}_{\hat{\boldsymbol{R}}} \to 0$) with a rate of $\lambda_{\min}(\boldsymbol{E}\boldsymbol{\Lambda}_e \boldsymbol{E}^T)$, where $\boldsymbol{E}\boldsymbol{\Lambda}_e \boldsymbol{E}^T$ is also a positive definite matrix. In the presence of $\boldsymbol{\omega}_e$, it follows from Lemma C.2 in Appendix C that:

$$\lim_{t\to\infty} \int_0^{\boldsymbol{e}_{\hat{\boldsymbol{R}}}} \|\delta \boldsymbol{e}_{\hat{\boldsymbol{R}}}\|_2 \leq \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{E}\boldsymbol{\Lambda}_e \boldsymbol{E}^T)\lambda_{\min}(\boldsymbol{K}_e)\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})} \left(\sup_t \sigma_{\max}(\boldsymbol{E})\right)\left(\sup_t \|\boldsymbol{d}_{\text{res},2}\|_2\right). \tag{4.43}$$

Note that $\|\boldsymbol{e}_{\hat{\boldsymbol{R}}}\|_2 \to \infty$ if $\hat{\boldsymbol{R}} \to \boldsymbol{R}_d \exp(\pm\pi \boldsymbol{S}(\boldsymbol{\kappa}))$, where $\boldsymbol{\kappa} \in \mathbb{S}^2$. On the other hand, for any valid initial condition, $\|\boldsymbol{e}_{\hat{\boldsymbol{R}}}\|_2$ is always bounded and exponentially decreasing till it reaches the error ball. This implies that once the system starts from a valid initial condition, it can never go towards the two-dimensional subset of SO(3) due to exponential convergence. Hence we have shown, using a hierarchical closed-loop system, that the attitude error vector $\boldsymbol{e}_{\hat{\boldsymbol{R}}}$ exponentially converges to the error bound for almost all initial conditions (except for those initial conditions in the two-dimensional subset of SO(3)). ■

4.3 Design of Desired Attitude Trajectory

In this section, we discuss techniques for computing a reference fuel-optimal trajectory and resultant disturbance torque used for the proposed attitude tracking control law in Section 4.2. We also outline a framework for minimizing the resultant disturbance torque for the tracking control law.

4.3.1 Design of Fuel-Optimal Desired Attitude Trajectory

In this section, we design the desired (reference) attitude trajectory $\boldsymbol{q}_d(t)$ so that the system reaches the desired attitude orientation $\boldsymbol{q}_{\text{final}}$ in a fuel-optimal fashion. The original nonlinear optimal control problem is given by:

$$\min_{\boldsymbol{q}_d(t),\,\boldsymbol{\omega}_d(t),\,\boldsymbol{u}_d(t)} \int_0^{t_{\text{final}}} \|\boldsymbol{u}_d(t)\|_1 dt\,, \tag{4.44}$$

$$\text{subject to} \quad \hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} \dot{\boldsymbol{\omega}}_d(t) - \left(\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}} \boldsymbol{\omega}_d(t)\right) \times \boldsymbol{\omega}_d(t) - \hat{\boldsymbol{B}} \boldsymbol{u}_d(t) = 0\,, \tag{4.45}$$

$$\dot{\boldsymbol{q}}_d(t) = \boldsymbol{Z}(\boldsymbol{q}_d(t))\boldsymbol{\omega}_d(t), \qquad \boldsymbol{q}_d(0) = \boldsymbol{q}_{\text{init}}\,, \qquad \boldsymbol{q}_d(t_{\text{final}}) = \boldsymbol{q}_{\text{final}}\,, \tag{4.46}$$

$$\|\boldsymbol{u}_d(t)\|_\infty \leq u_{\max}\,, \qquad \|\boldsymbol{\omega}_d(t)\|_2 \leq \varepsilon_{\text{trans}}\,, \qquad \boldsymbol{\omega}_d(0) = \boldsymbol{\omega}_{\text{init}}\,, \qquad \boldsymbol{\omega}_d(t_{\text{final}}) = \boldsymbol{0} \tag{4.47}$$

where $\boldsymbol{\omega}_d(t)$ and $\boldsymbol{u}_d(t)$ are the fuel-optimal angular velocity and thruster input trajectories. Since all the thrusters generate thrust independently (and there is no gimballing of thrusters), we use the ℓ_1 vector norm in the $\mathcal{L}_1$ cost function in Eq. (4.61) [198]. In [111, 199, 200, 201, 202], a number of optimization strategies are discussed for solving this problem.

We show later that a relatively negligible amount of fuel is needed for driving the attitude of the system to the desired value after the angular velocity of the system is satisfactorily stabilized (see Fig. 4.4(e) in Section 4.4). Therefore, we first find the fuel-optimal angular velocity trajectory $\boldsymbol{\omega}_d(t)$ that reduces the system's attitude rate to a sufficiently small value and then design a full attitude trajectory $\boldsymbol{q}_d(t)$. The desired fuel-optimal angular velocity

trajectory $\boldsymbol{\omega}_d(t)$ is obtained by solving the following reduced optimal control problem:

$$\min_{\boldsymbol{\omega}_d(t),\, \boldsymbol{u}_d(t)} \int_0^{t_{\text{final}}} \|\boldsymbol{u}_d(t)\|_1 dt\,, \qquad \text{subject to} \quad \text{Eqs. (4.45) and (4.47)} \tag{4.48}$$

Since the reduced optimal control problem of $\boldsymbol{\omega}_d(t)$ in Eq. (4.48) has fewer optimization constraints than the full optimal control problem of finding both $\boldsymbol{q}_d(t)$ and $\boldsymbol{\omega}_d(t)$ in Eq. (4.44), the solution of the reduced problem in Eq. (4.48) consumes less fuel than the full problem in Eq. (4.44). Once $\boldsymbol{\omega}_d(t)$ is computed from Eq. (4.48), $\boldsymbol{q}_d(t)$ is then obtained using the following equations:

$$\dot{\boldsymbol{q}}_d(t) = Z(\boldsymbol{q}_d(t))\boldsymbol{\omega}_d(t)\,, \qquad \ddot{\boldsymbol{q}}_d(t) = \dot{Z}(\boldsymbol{q}_d(t))\boldsymbol{\omega}_d(t) + Z(\boldsymbol{q}_d(t))\dot{\boldsymbol{\omega}}_d(t)\,. \tag{4.49}$$

Note that the desired attitude trajectory $\boldsymbol{q}_d(t)$ obtained using Eq. (4.49) only stabilizes the angular velocity of the system.

Once the angular velocity of the system is sufficiently close to zero, the desired angular velocity trajectory $\boldsymbol{\omega}_d(t)$ is augmented with a position error term so that the system's attitude converges to the desired attitude:

$$\tilde{\boldsymbol{\omega}}_d(t) = \boldsymbol{\omega}_d(t) - k_{qd} Z(\boldsymbol{q}_d(t))^{-1}(\boldsymbol{q}_d(t) - \boldsymbol{q}_{\text{final}})\,, \tag{4.50}$$

where $k_{qd} > 0$. The desired attitude trajectory $\boldsymbol{q}_d(t)$ is then obtained from the augmented angular velocity $\tilde{\boldsymbol{\omega}}_d(t)$ using the following equations:

$$\dot{\boldsymbol{q}}_d(t) = Z(\boldsymbol{q}_d(t))\tilde{\boldsymbol{\omega}}_d(t) = Z(\boldsymbol{q}_d(t))\boldsymbol{\omega}_d(t) - k_{qd}(\boldsymbol{q}_d(t) - \boldsymbol{q}_{\text{final}})\,, \tag{4.51}$$

$$\ddot{\boldsymbol{q}}_d(t) = \frac{d}{dt}\left[Z(\boldsymbol{q}_d(t))\tilde{\boldsymbol{\omega}}_d(t)\right] = \dot{Z}(\boldsymbol{q}_d(t))\boldsymbol{\omega}_d(t) + Z(\boldsymbol{q}_d(t))\dot{\boldsymbol{\omega}}_d(t) - k_{qd}\dot{\boldsymbol{q}}_d(t)\,. \tag{4.52}$$

These equations are initialized and periodically reset using the current attitude and angular velocity measurements.

4.3.2 Desired Attitude Trajectory using Derivative plus Proportional-Derivative Control

In this section, we first state the derivative plus proportional-derivative (D+PD) control strategy and then design another desired attitude trajectory $\boldsymbol{q}_d(t)$ based on the D+PD control strategy.

In the D+PD control strategy, we first use the derivative (rate damping) linear control law for despinning the tumbling system. Once the angular velocity (spin rate) of the system is sufficiently close to zero, the D+PD control strategy switches to a linear PD control law to stabilize the attitude of the system in the desired orientation.

Theorem 4.4. *[113, 190, 192] (i) For the positive-definite symmetric matrix $\boldsymbol{K}_d \in \mathbb{R}^{3\times 3}$, the derivative (rate damping) control law is given by:*

$$\boldsymbol{u}_c = -\boldsymbol{K}_d \hat{\boldsymbol{\omega}}\,, \tag{4.53}$$

In the absence of disturbances or uncertainties, this control law guarantees global exponential convergence of the system's angular velocity to $\mathbf{0}$ rad sec^{-1}. In the presence of resultant disturbance torque, this control law guarantees the system's angular velocity trajectory will globally exponentially converge to a bounded error ball around $\mathbf{0}$ rad sec^{-1}.
(ii) For the positive-definite symmetric matrix $\boldsymbol{K}_d \in \mathbb{R}^{3\times 3}$ and the constant $k_p > 0$, the proportional-derivative control law is given by:

$$\boldsymbol{u}_c = -k_p \boldsymbol{\beta}_{\text{error},v} - \boldsymbol{K}_d \hat{\boldsymbol{\omega}}\,, \tag{4.54}$$

where the error quaternion $(\boldsymbol{\beta}_{\text{error},v}, \boldsymbol{\beta}_{\text{error},4}) \in \mathbb{R}^3 \times \mathbb{R}$ represents the orientation error of $\mathcal{F}_B$ with respect to the desired target attitude $\boldsymbol{\beta}_{\text{final}}$. This control law only guarantees global asymptotic convergence of the system's trajectory to the desired trajectory $\boldsymbol{q}_d(t)$ in the absence

of disturbances or uncertainties. Hence, the error in the system's trajectory may not be bounded for a certain class of disturbances [194].

Proof: Although the proof techniques of this theorem are well-known, we present a short proof that will be used in the subsequent sections for analysis of robustness in light of Lemma C.2. The closed-loop dynamics from Eq. (4.53) and Eq. (4.4), are given by:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\hat{\boldsymbol{\omega}}} - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\hat{\boldsymbol{\omega}} + \boldsymbol{K}_d\hat{\boldsymbol{\omega}} = \boldsymbol{d}_{\text{res}}\,. \tag{4.55}$$

Let us use the Lyapunov function $V_d = \hat{\boldsymbol{\omega}}^T\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}$. In the presence of disturbance $\boldsymbol{d}_{\text{res}}$ bounded by $\sup_t \|\boldsymbol{d}_{\text{res}}\|_2 \leq d_{\text{res,max}}$, differentiating V_d with respect to time and applying the comparison lemma [194] gives:

$$\|\hat{\boldsymbol{\omega}}(t)\|_2 \leq \sqrt{\frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}}e^{-\frac{\lambda_{\min}(\boldsymbol{K}_d)}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}t}\|\hat{\boldsymbol{\omega}}(0)\|_2 + \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}\frac{d_{\text{res,max}}}{\lambda_{\min}(\boldsymbol{K}_d)}\left(1 - e^{-\frac{\lambda_{\min}(\boldsymbol{K}_d)}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}t}\right). \tag{4.56}$$

Thus, in the absence of $\boldsymbol{d}_{\text{res}}$, the system's angular velocity converges exponentially fast to $\boldsymbol{0}$ rad sec^{-1}, regardless of initial conditions, with the convergence rate of $\frac{\lambda_{\min}(\boldsymbol{K}_d)}{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}$. Moreover, the solution of the perturbed system exponentially converges to the error ball $\|\hat{\boldsymbol{\omega}}(t)\|_2 \leq \frac{\lambda_{\max}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}{\lambda_{\min}(\boldsymbol{J}_{\text{tot}}^{B_{CM}})}\frac{d_{\text{res,max}}}{\lambda_{\min}(\boldsymbol{K}_d)}$ (see Lemma C.2).

Let us now prove claim (ii). The closed-loop system (without $\boldsymbol{d}_{\text{res}}$) can be written as:[106]

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\hat{\boldsymbol{\omega}}} - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\hat{\boldsymbol{\omega}} + k_p\boldsymbol{\beta}_{\text{error},v} + \boldsymbol{K}_d\hat{\boldsymbol{\omega}} = 0\,. \tag{4.57}$$

Let the Lyapunov function be $V_{pd} = \hat{\boldsymbol{\omega}}^T\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}} + 2k_p\boldsymbol{\beta}_{\text{error},v}^T\boldsymbol{\beta}_{\text{error},v} + 2k_p(1 - \boldsymbol{\beta}_{\text{error},4})^2$. Differentiating V_{pd} with respect to time gives $\dot{V}_{pd} = -\hat{\boldsymbol{\omega}}^T(\boldsymbol{K}_d^T + \boldsymbol{K}_d)\hat{\boldsymbol{\omega}}$. Since $\dot{V}_{pd}$ is negative semi-definite, application of LaSalle's invariant set theorem [117, 194] proves that $\lim_{t\to\infty}\hat{\boldsymbol{\omega}} = \boldsymbol{0}$ rad sec^{-1} and that $\lim_{t\to\infty}\boldsymbol{\beta}_{\text{error},v} = \boldsymbol{0}$ with $\lim_{t\to\infty}\boldsymbol{\beta}_{\text{error},4} = \pm 1$, Note that the error quaternions $\boldsymbol{\beta}_{\text{error},4} = \pm 1$ represent the same attitude orientation. ■

The closed-loop dynamics from Eq. (4.53) and Eq. (4.54), are given by:

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\hat{\boldsymbol{\omega}}} - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\hat{\boldsymbol{\omega}} + \boldsymbol{K}_d\hat{\boldsymbol{\omega}} = \boldsymbol{d}_{\text{res}}\,, \tag{4.58}$$

$$\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\hat{\boldsymbol{\omega}}} - \boldsymbol{S}\left(\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\right)\hat{\boldsymbol{\omega}} + k_p\boldsymbol{\beta}_{\text{error},v} + \boldsymbol{K}_d\hat{\boldsymbol{\omega}} = \boldsymbol{d}_{\text{res}}\,. \tag{4.59}$$

It is seen in Section 4.1.2 that the D+PD control strategy experiences a smaller resultant disturbance torque even in the presence of large $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}$. But the D+PD control strategy does not guarantee global exponential stability (in the absence of disturbances), which is a sufficient condition for satisfying the control problem statement. Hence, we now present the design of a resultant disturbance minimizing desired attitude trajectory for the nonlinear attitude tracking control law Eq. (4.13).

The desired trajectory is basically broken into two phases. In the first phase, similar to the D+PD control strategy, the desired attitude trajectory is such that $\boldsymbol{\omega}_r = \boldsymbol{0}$ in Eq. (4.13) if the magnitude of the system's angular velocity is large. This ensures that the robust nonlinear tracking control law Eq. (4.13) effectively reduces to the linear derivative control law Eq. (4.53) with the same global exponential tracking stability and resultant disturbance torque.

In the second phase, once the angular velocity of the system is sufficiently close to zero, we use the following desired attitude trajectory for the nonlinear tracking control law Eq. (4.13):

$$\boldsymbol{q}_d(t) = \boldsymbol{q}_{\text{final}}\,, \qquad \dot{\boldsymbol{q}}_d(t) = \boldsymbol{0}\,, \qquad \therefore \boldsymbol{\omega}_r = \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\boldsymbol{\Lambda}_r(\boldsymbol{q}_{\text{final}} - \hat{\boldsymbol{q}}). \tag{4.60}$$

Note that this ensures that the system's attitude globally exponentially converges to the desired final attitude and the system is robust to disturbances. Since the actual angular velocity of the system is small, the resultant disturbance torque is also small even in the presence of large modeling error in $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}$.

The following proposition provides a framework for choosing the desired attitude trajec-

tory so that the resultant disturbance torque $\boldsymbol{d}_{\text{res},2}$ is as small as $\boldsymbol{d}_{\text{res}}$.

Proposition 4.5. *Compared to $\boldsymbol{d}_{\text{res}}$, the extra terms in $\boldsymbol{d}_{\text{res},2}$ (i.e., $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r$ and $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\boldsymbol{\omega}_r$) depend on $\boldsymbol{\omega}_r$, which in turn depends on the desired attitude trajectory. Therefore, the desired attitude trajectory is chosen as follows: (i) If the modeling error in $\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}$ is small (i.e., $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\|_2 \leq 10^4$ kg m^2), then select the fuel-optimal desired attitude trajectory from Section 4.3.1. (ii) Otherwise, select the desired attitude trajectory based on the D+PD control strategy given in Section 4.3.2.*

This will ensure that $\|\boldsymbol{d}_{\text{res},2}\|_2 \approx \|\boldsymbol{d}_{\text{res}}\|_2$, consequently minimizing the resultant disturbance torque for the robust nonlinear tracking control law Eq. (4.13).

Proof: Let the worst case angular velocity of the system be bounded by 0.5 rpm ($\approx 5\times10^{-2}$ rad sec^{-1}) as shown in Table 4.5. If the fuel-optimal desired trajectory is used, then $\|\boldsymbol{\omega}_r\|_2 \approx \|\hat{\boldsymbol{\omega}}\|_2$. If the modeling error is small (i.e., $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\|_2 \leq 10^4$ kg m^2), then $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\boldsymbol{\omega}_r\|_2 \leq$ 25 Nm. Neglecting $\dot{\boldsymbol{\omega}}_r$, which is significantly smaller than $\boldsymbol{\omega}_r$ or $\hat{\boldsymbol{\omega}}$, we see that $\|\boldsymbol{d}_{\text{res},2}\|_2 \approx \|\boldsymbol{d}_{\text{res}}\|_2$.

If the D+PD control strategy based desired attitude trajectory is used, then $\boldsymbol{\omega}_r = \boldsymbol{0}$ when $\hat{\boldsymbol{\omega}}$ is large, therefore $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\boldsymbol{\omega}_r\|_2 = 0$ Nm and $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\dot{\boldsymbol{\omega}}_r\|_2 = 0$ Nm. If $\hat{\boldsymbol{\omega}}$ is sufficiently close to $\boldsymbol{0}$ (i.e., $\|\hat{\boldsymbol{\omega}}\|_2 \leq 5\times10^{-4}$ rad sec^{-1} as shown in Table 4.7 and $\|\boldsymbol{\omega}_r\|_2 \approx \|\hat{\boldsymbol{\omega}}\|_2$), and even if the modeling error is very large (i.e., $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\|_2 \leq 10^7$ kg m^2), we still get $\|\Delta\boldsymbol{J}_{\text{tot}}^{B_{CM}}\hat{\boldsymbol{\omega}}\times\boldsymbol{\omega}_r\|_2 \leq 2.5$ Nm. Neglecting $\dot{\boldsymbol{\omega}}_r$ again, we see that $\|\boldsymbol{d}_{\text{res},2}\|_2 \approx \|\boldsymbol{d}_{\text{res}}\|_2$.
■

4.4 Simulation Results

In this section, we apply our proposed control law to the ARM attitude control problem of carrying a large unknown object. We first numerically compare the performance of multiple attitude control laws in Section 4.4.1. We then present a detailed sensitivity analysis of

various parameters used in the robust nonlinear tracking control law Eq. (4.13) and the D+PD based desired attitude trajectory in Section 4.3.2.

4.4.1 Comparison of Control Laws for Nonlinear Attitude Control

We use the nominal design of the conceptual ARM spacecraft given in [1] and shown in Fig. 4.2(a). Here an opposing pair of thrusters in a pod are represented by a single thruster capable of producing thrust between +200 N to −200 N. We use the Moore–Penrose pseudoinverse of $\hat{\boldsymbol{B}}$ to allocate thrusts to the eight thrusters in the spacecraft, i.e., $\hat{\boldsymbol{u}} = \hat{\boldsymbol{B}}^T \left(\hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^T\right)^{-1} \boldsymbol{u}_c$. Note that we use the right-pseudoinverse since the matrix $\hat{\boldsymbol{B}}$ has full row rank and the matrix inverse $\left(\hat{\boldsymbol{B}}\hat{\boldsymbol{B}}^T\right)^{-1}$ is well defined. We do not use the left-pseudoinverse since the matrix $\left(\hat{\boldsymbol{B}}^T\hat{\boldsymbol{B}}\right)$ is usually near singular and hence its inverse may not be defined.

The fuel consumed by the spacecraft, from time t_0 to t_f, is computed using the following equation:

$$\text{Fuel consumed} = \frac{1}{I_{sp}\, g_0} \int_{t_0}^{t_f} \|\boldsymbol{u}\|_1 dt\,, \tag{4.61}$$

where I_{sp} is the specific impulse of the fuel (i.e., 287 sec for the spacecraft[1]) and g_0 is the nominal acceleration due to the gravity (i.e., 9.8 m sec^{-2}). Since all the thrusters generate thrust independently (and there is no gimballing of thrusters), we use the ℓ_1 cost function in Eq. (4.61) [198].

The shape models of asteroids 433 Eros [6] and 25143 Itokawa [7], shown in Fig. 4.1(a,b), are used for generating realistic models of asteroids. We assume that the 16-metric-ton spacecraft has captured an 1200-metric-ton asteroid. The objective is to stabilize the rigid asteroid and spacecraft combination from the given initial conditions to reach the desired final conditions. The simulation parameters, which are the same for all simulation cases, are given in Table 4.5.

Table 4.5: Simulation parameters (that are same for all simulation cases)

Type of Parameter	**Value**
Spacecraft Parameters	$m_{\text{sc}} = 1.6 \times 10^4$ kg, $\boldsymbol{J}_{\text{sc}}^{S_{CM}} = 10^4 \times \begin{bmatrix} 5.584 & 0 & 0 \\ 0 & 5.584 & 0 \\ 0 & 0 & 1.568 \end{bmatrix}$ kg m^2, $\boldsymbol{r}^{S_{CM}/S_O} = [\ 0 \quad 0 \quad 3.0\ ]$ m,
Asteroid Parameters	$m_{\text{obj}} = 1.2 \times 10^6$ kg, $\rho_{\text{obj}} = 1.9$ g cm^{-3}, Shape model: Eros, $\boldsymbol{J}_{\text{obj}}^{B_{CM}} = 10^7 \times \begin{bmatrix} 0.8658 & 0.4432 & -0.0005 \\ 0.4432 & 3.4900 & 0.0002 \\ -0.0005 & 0.0002 & 3.5579 \end{bmatrix}$ kg m^2, $\boldsymbol{r}^{S_O/B_{CM}} = [\ -0.0495 \quad -0.0004 \quad 3.5456\ ]$ m,
External Disturbance Actuator Error	$\|\boldsymbol{d}_{\text{ext}}\|_2 \approx 1$ Nm, $\Delta\boldsymbol{u} = 0$ N,
Initial Conditions	$\boldsymbol{q}_{\text{initial}} = [\ 0.05 \quad 0.04 \quad 0.03\ ]$, $\boldsymbol{\omega}_{\text{initial}} = [\ 0.01 \quad 0.02 \quad 0.03\ ]$ rad sec^{-1},
Desired Final Conditions Eqs. (4.11,4.12)	$\|\boldsymbol{\omega}(t)\|_2 \leq 0.5$ rpm, $\forall t \in \mathbb{R}$, $\boldsymbol{q}_{\text{final}} = [\ 0 \quad 0 \quad 0\ ]$, $\|\boldsymbol{q}(t) - \boldsymbol{q}_{\text{final}}\|_2 \leq 10^{-2}$, $\forall t > 10^5$ sec, $\|\boldsymbol{\omega}(t)\|_2 \leq 10^{-4}$ rad sec^{-1}, $\forall t > 10^5$ sec,

Table 4.6: Time-varying simulation parameters for the simulation cases

Case	**Modeling Uncertainties**		**Measurement Errors**		**Actuator Sat.**
	$\|\Delta\boldsymbol{J}_{\text{obj}}^{B_{CM}}\|_2$ (kg m^2)	$\|\Delta\boldsymbol{r}^{S_O/B_{CM}}\|_2$ (m)	$\mathcal{P}(\Delta\boldsymbol{\omega})$ (rad^2 sec^{-2})	$\mathcal{P}(\Delta\boldsymbol{q})$	$u_{\max}$ (N)
1.	0	0	0	0	200
2.	10^5	10^{-2}	0	0	200
3.	10^6	10^{-1}	0	0	200
4.	10^7	1	0	0	1000
5.	0	0	10^{-12}	10^{-8}	200
6.	0	0	10^{-10}	10^{-6}	200
7.	10^7	1	10^{-10}	10^{-6}	200
8.	10^6	1	10^{-10}	10^{-6}	200
9.	10^7	10^{-1}	10^{-10}	10^{-6}	200
10.	10^7	1	10^{-12}	10^{-6}	200
11.	10^7	1	10^{-10}	10^{-8}	200

In Table 4.6, we state the eleven simulation cases considered in this study. These simulation cases are based on varying levels of: (i) modeling uncertainties in the estimated inertia tensor of the asteroid ($\Delta\boldsymbol{J}_{\text{obj}}^{B_{CM}}$), (ii) modeling uncertainties in the vector from the spacecraft's body to the center of mass of the system ($\Delta\boldsymbol{r}^{S_O/B_{CM}}$), (iii) measurement errors in the system's angular velocity ($\Delta\boldsymbol{\omega}$), (iv) measurement errors in the system's attitude represented using MRP ($\Delta\boldsymbol{q}$), and (v) actuator saturations ($u_{\max}$). Each simulation is executed for 10^5 sec ($\approx$ 28 hours). The additive measurement errors ($\Delta\boldsymbol{\omega}$, $\Delta\boldsymbol{q}$) are simulated using band-limited white noise where $\mathcal{P}(\cdot)$ specifies the height of the power spectral density of the

white noise, which is the same for each axis. Note that in Case 4, the maximum thrust magnitude of each thruster ($u_{\max}$) is increased to 1000 N to avoid actuator saturation.

Table 4.7: Control law parameters and desired attitude trajectory parameters

Type of Parameter	**Value**
Robust NTCL Eq. (4.13)	$\boldsymbol{K}_r = 10^4\mathbf{I}$, $\boldsymbol{\Lambda}_r = 10^{-3}\mathbf{I}$,
Adaptive RNTCL Eq. (4.38)	$\boldsymbol{K}_r = 10^4\mathbf{I}$, $\boldsymbol{\Lambda}_r = 10^{-3}\mathbf{I}$, $\boldsymbol{\Gamma}_r = 10^{12}\mathbf{I}$
D+PD Control Strategy Eqs. (4.53,4.54)	$\boldsymbol{K}_d = 10^4\mathbf{I}$, $k_p = 10$, Switch from derivative to proportional-derivative when $\|\hat{\boldsymbol{\omega}}(t)\|_2 \leq 5 \times 10^{-4}$ rad sec^{-1}.
Fuel-optimal desired attitude trajectory (Section 4.3.1)	Desired angular velocity $\boldsymbol{\omega}_d(t)$ is obtained by solving Eq. (4.48) using the GPOPS-II numerical solver[203] Desired trajectory $\boldsymbol{q}_d(t)$ obtained using Eq. (4.49), When $\|\hat{\boldsymbol{\omega}}(t)\|_2 \leq 5 \times 10^{-4}$ rad sec^{-1}, switch to angular velocity $\tilde{\boldsymbol{\omega}}_d(t)$ in Eq. (4.50) with $k_{qd} = 10^{-4}$, Desired trajectory $\boldsymbol{q}_d(t)$ obtained using Eqs. (4.51,4.52).
D+PD control strategy based desired attitude trajectory (Section 4.3.2)	Start with $\boldsymbol{\omega}_r = \mathbf{0}$, When $\|\hat{\boldsymbol{\omega}}(t)\|_2 \leq 5 \times 10^{-4}$ rad sec^{-1}, switch to desired trajectory $\boldsymbol{q}_d(t) = \boldsymbol{q}_{\text{final}}$ in Eq. (4.60), therefore $\boldsymbol{\omega}_r = \boldsymbol{Z}^{-1}(\hat{\boldsymbol{q}})\boldsymbol{\Lambda}_r(\boldsymbol{q}_{\text{final}} - \hat{\boldsymbol{q}})$.

In this section, we compare the performance of the following attitude control laws: (i) Robust nonlinear tracking control law (Robust NTCL) Eq. (4.13), (ii) Adaptive version of the robust nonlinear tracking control law (Adaptive RNTCL) Eq. (4.38), and (iii) Derivative plus proportional-derivative (D+PD) control Eqs. (4.53,4.54). For the tracking control laws, both the fuel-optimal desired attitude trajectory (Section 4.3.1) and D+PD control based desired attitude trajectory (Section 4.3.2) are considered. The control law parameters and the parameters for these two desired attitude trajectories are given in Table 4.7.

The performance of these control laws for the eleven simulation cases given in Table 4.6 are shown in Table 4.8 and Fig. 4.3. Some of the notations used in Table 4.8 and Fig. 4.3 are as follows: (i) The angular velocity convergence time $t_{\omega,\text{conv}}$ denotes the least time instant after which the system's angular velocity $\boldsymbol{\omega}(t)$ is always below the given threshold of 10^{-4} rad sec^{-1}, i.e., $\|\boldsymbol{\omega}(t)\|_2 \leq 10^{-4}$ rad sec^{-1}, $\forall t > t_{\omega,\text{conv}}$. (ii) The attitude convergence time $t_{q,\text{conv}}$ denotes the least time instant after which the error in the system's attitude $\|\boldsymbol{q}(t) - \boldsymbol{q}_{\text{final}}\|_2$ is always below the given threshold of 10^{-2}, i.e., $\|\boldsymbol{q}(t) - \boldsymbol{q}_{\text{final}}\|_2 \leq 10^{-2}$, $\forall t > t_{q,\text{conv}}$. Note that after time $t_{q,\text{conv}}$, the attitude control law can be switched off because the asteroid and

Table 4.8: Performance of control laws for the simulation cases, where the fuel-optimal and D+PD based desired attitude trajectories are used

		Fuel-optimal trajectory		D+PD based trajectory		
Case	**Convergence Time and Fuel Consumed**	**Robust NTCL Eq. (4.13)**	**Adaptive RNTCL Eq. (4.38)**	**Robust NTCL Eq. (4.13)**	**Adaptive RNTCL Eq. (4.38)**	**D+PD Control Eqs.(4.53,4.54)**
1.	$t_{\omega,\text{conv}}$ (sec)	5.23×10^4	5.32×10^4	2.34×10^4	2.34×10^4	3.08×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)	82.5	106.4	135.2	135.1	120.2
	$t_{q,\text{conv}}$(sec)	8.57×10^4	8.56×10^4	3.01×10^4	2.93×10^4	4.31×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)	82.9	106.9	135.5	135.3	121.4
2.	$t_{\omega,\text{conv}}$ (sec)	5.23×10^4	5.32×10^4	2.35×10^4	2.35×10^4	3.16×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)	82.7	113.0	134.3	134.1	120.6
	$t_{q,\text{conv}}$(sec)	8.57×10^4	8.57×10^4	3.03×10^4	2.95×10^4	4.31×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)	83.1	113.4	134.5	134.3	121.7
3.	$t_{\omega,\text{conv}}$ (sec)	2.30×10^4	2.37×10^4	2.38×10^4	2.38×10^4	3.41×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)	138.9	130.2	127.9	127.8	121.0
	$t_{q,\text{conv}}$(sec)	5.56×10^4	5.59×10^4	3.07×10^4	2.97×10^4	4.84×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)	139.4	130.8	128.2	128.1	121.6
4.	$t_{\omega,\text{conv}}$ (sec)	2.86×10^4	3.06×10^4	2.07×10^4	2.07×10^4	2.80×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)	1220.1	814.7	115.1	115.2	116.1
	$t_{q,\text{conv}}$(sec)	6.04×10^4	6.14×10^4	2.61×10^4	2.52×10^4	4.16×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)	1220.3	814.9	115.6	115.6	117.5
5.	$t_{\omega,\text{conv}}$ (sec)	5.32×10^4	5.31×10^4	2.34×10^4	2.34×10^4	3.08×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)	83.2	86.0	135.3	135.1	120.2
	$t_{q,\text{conv}}$(sec)	8.60×10^4	8.51×10^4	3.01×10^4	2.93×10^4	4.31×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)	83.6	86.4	135.5	135.3	121.5
6.	$t_{\omega,\text{conv}}$ (sec)	5.32×10^4	5.31×10^4	2.34×10^4	2.34×10^4	3.08×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)	83.1	87.5	135.2	135.0	120.2
	$t_{q,\text{conv}}$(sec)	8.56×10^4	8.48×10^4	3.01×10^4	2.93×10^4	4.32×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)	83.7	88.9	135.5	135.3	121.5
7.	$t_{\omega,\text{conv}}$ (sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.05×10^4	2.05×10^4	2.76×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)			114.6	114.6	115.9
	$t_{q,\text{conv}}$(sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.59×10^4	2.52×10^4	4.11×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)			115.1	115.1	117.3
8.	$t_{\omega,\text{conv}}$ (sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.08×10^4	2.08×10^4	2.80×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)			115.5	115.5	116.6
	$t_{q,\text{conv}}$(sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.81×10^4	2.74×10^4	4.19×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)			116.1	116.1	118.1
9.	$t_{\omega,\text{conv}}$ (sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.36×10^4	2.36×10^4	3.35×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)			129.5	129.4	121.4
	$t_{q,\text{conv}}$(sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.77×10^4	2.67×10^4	4.70×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)			129.8	130.0	122.1
10.	$t_{\omega,\text{conv}}$ (sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.05×10^4	2.05×10^4	2.76×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)			114.6	114.7	115.9
	$t_{q,\text{conv}}$(sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.59×10^4	2.52×10^4	4.11×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)			115.1	115.2	117.3
11.	$t_{\omega,\text{conv}}$ (sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.05×10^4	2.05×10^4	2.76×10^4
	Fuel at $t_{\omega,\text{conv}}$ (kg)			114.6	114.6	115.9
	$t_{q,\text{conv}}$(sec)	$\mathcal{NC}^{\ddagger\ddagger}$	$\mathcal{NC}^{\ddagger\ddagger}$	2.59×10^4	2.52×10^4	4.11×10^4
	Fuel at $t_{q,\text{conv}}$ (kg)			115.1	115.1	117.3

spacecraft combination has been three-axis stabilized in the final desired orientation. The fuel consumed up to time $t_{\omega,\text{conv}}$ and $t_{q,\text{conv}}$ are also shown in Table 4.8. (iii) The symbol $\mathcal{NC}$ or "Not Converged" refers to the case when the control law is not able to stabilize the system due to actuator saturation.

In the absence of measurement errors and modeling uncertainties (Case 1), Fig. 4.4(a,b,c) shows a result of the nonlinear tracking control law (Robust NTCL) tracking the fuel-optimum reference trajectory. Note that Fig. 4.4(c) also shows the fuel consumption level for the case where the fuel-optimal $\boldsymbol{\omega}_d(t)$ trajectory is not augmented (i.e., $k_{qd} = 0$) and

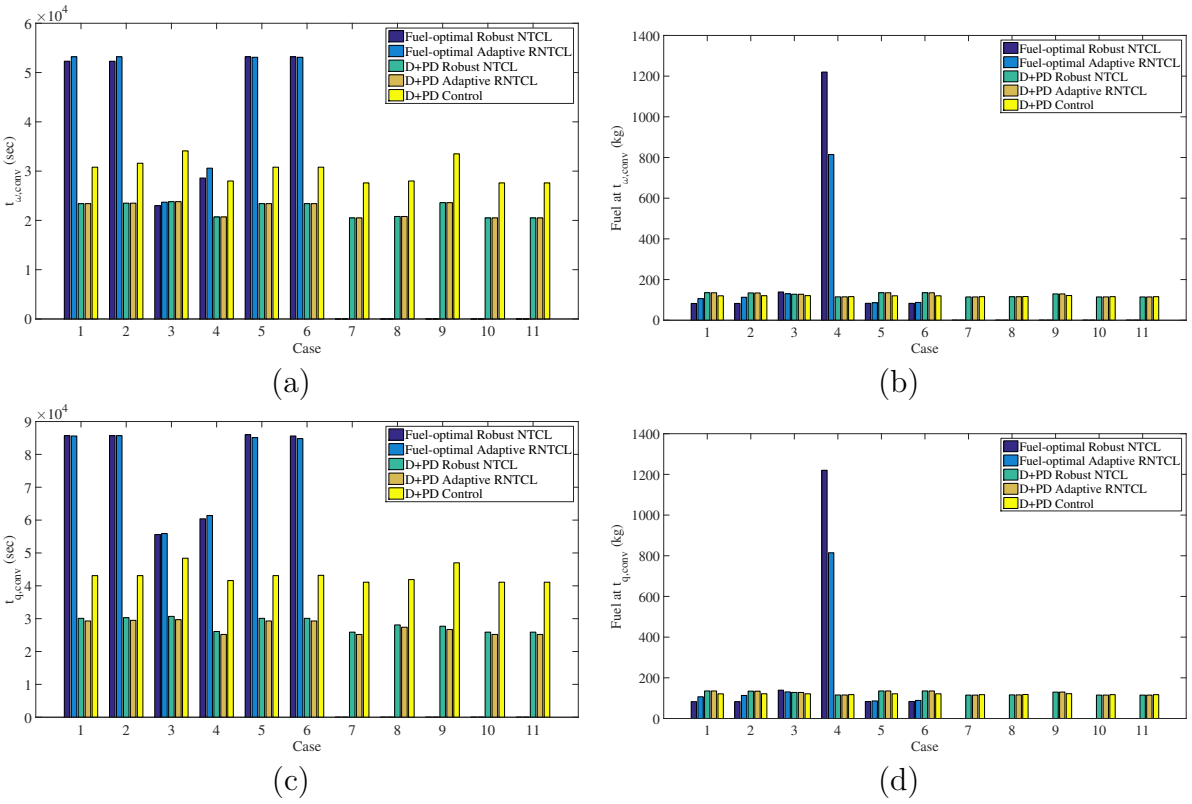


Figure 4.3: Visualization of the results in Table 4.8 for (a) $t_{\omega,\text{conv}}$, (b) fuel at $t_{\omega,\text{conv}}$, (c) $t_{q,\text{conv}}$, and (d) fuel at $t_{q,\text{conv}}$.

consequently only the angular velocity of the system converges. We can infer from this plot that a relatively negligible amount of fuel (≈ 3 kg) is used for stabilizing the attitude of the asteroid and spacecraft combination using the augmented angular velocity $\tilde{\boldsymbol{\omega}}_d(t)$ in Eq. (4.50).

We conclude from Cases 1 and 2 that in the absence of measurement errors and under smaller modeling uncertainties, which can be achieved using online system identification techniques, the method of Robust NTCL tracking the fuel optimal trajectory is the best strategy because it guarantees exponential convergence to the desired trajectory, thereby consuming the least fuel. One caveat of using this control law is that the reduced nonlinear optimal control problem in Eq. (4.48) should be solved in real time for the given initial angular velocity $\boldsymbol{\omega}_{\text{initial}}$ and the estimated inertia tensor of the combined system $\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}$.

Cases 3 and 4 in Fig. 4.3 show that the nonlinear tracking control laws, which track the

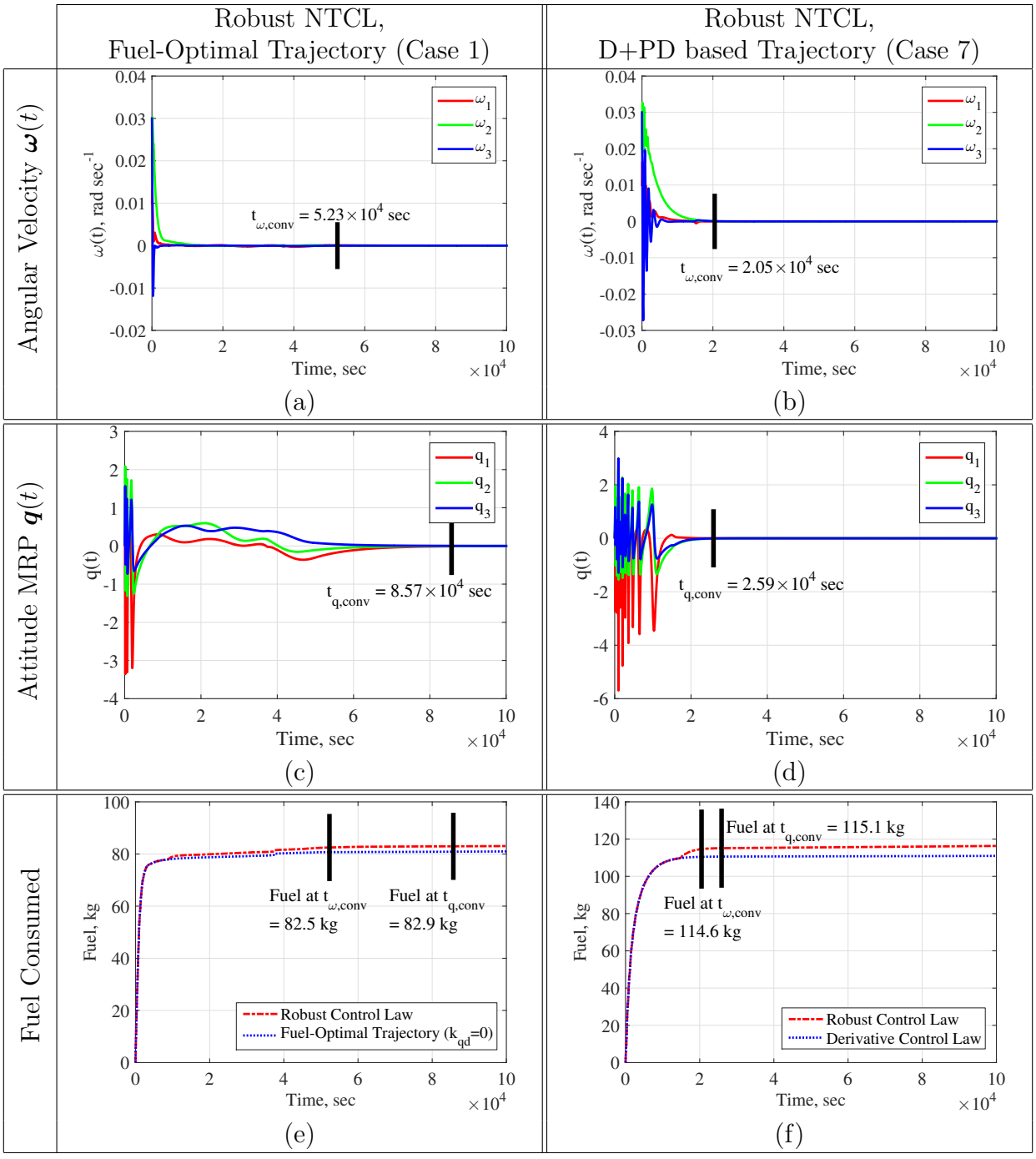


Figure 4.4: Simulation results of the Robust NTCL for Case 1 and Case 7 in Table 4.8 are shown. The plots show the trajectories of the angular velocity $\boldsymbol{\omega}(t)$, the attitude represented using MRP $\boldsymbol{q}(t)$, and the fuel consumed with respect to time. The angular velocity convergence time $t_{\omega,\text{conv}}$, the attitude convergence time $t_{q,\text{conv}}$, and the corresponding fuel consumption are also shown.

fuel-optimal desired attitude trajectory, consume more fuel than the same nonlinear tracking control laws that track the D+PD control strategy based desired attitude trajectory, because of the large resultant disturbance torque discussed in Section 4.1.2. Note that the angular velocity convergence times ($t_{\omega,\text{conv}}$) of the nonlinear control laws for Cases 1–4 are different because different values of $\hat{\boldsymbol{J}}_{\text{tot}}^{B_{CM}}$ are used in the nonlinear optimal control problem (4.48) to obtain the fuel-optimal desired attitude trajectories ($\boldsymbol{\omega}_d(t)$, $\boldsymbol{q}_d(t)$) given in Section 4.3.1.

In Cases 5–11, a simple filtering algorithm is used to remove the additive noise from the measured states. In this filtering algorithm, the states ($\boldsymbol{\omega}$, $\boldsymbol{q}$) are first predicted using the nonlinear dynamics and kinematics equations and state values from the previous time instant. Then the errors between the measured states and the predicted states are filtered using a low-pass filter (first-order filter with transfer function $\frac{\omega_{\text{cutoff}}}{s+\omega_{\text{cutoff}}}$, where $\omega_{\text{cutoff}} = 0.02\pi$ rad sec^{-1}) to remove the high frequency components arising from the noise. Finally, the filtered errors are added to the predicted states to retrieve the estimated states ($\hat{\boldsymbol{\omega}}$, $\hat{\boldsymbol{q}}$).

Cases 5 and 6 show that the Robust NTCL consumes less fuel while tracking the fuel-optimal trajectory compared to the D+PD control strategy based trajectory in the presence of small measurement errors. In contrast, Case 7 shows that this Robust NTCL cannot stabilize the system in the presence of both large modeling errors and measurement errors, because of actuator saturation caused by the large resultant disturbance torque discussed in Section 4.1.2.

If we use the resultant disturbance torque minimizing, D+PD control strategy based desired attitude trajectory, then the Robust NTCL can stabilize the system in the presence of both large measurement errors and large modeling errors. Moreover, the fuel consumed and the time of convergence do not change much with uncertainties and errors, as seen in Cases 1–7 in Fig. 4.3. Moreover, Case 7 shows the worst case measurement errors for the desired convergence bounds because if the measurement errors (noise levels) increase above these values, then the instantaneous magnitude of the measurement errors become comparable to the desired convergence bounds in Table 4.5 and the spacecraft expends fuel

continuously to counter these errors. Therefore these uncertainty and error limits determine the required technical capabilities of the sensors and actuators on board the spacecraft.

The simulation results (trajectories) of the Robust NTCL for Case 7 are shown in Fig. 4.4(d,e,f). Note that the net fuel consumed ($\approx$ 120 kg) after 10^5 sec is comfortably within the fuel capacity of the spacecraft (i.e., 900 kg [1]). Figure 4.4(f) also shows the fuel consumption for the case where only the derivative (rate damping) control law Eq. (4.53) is used for the entire time and consequently only the angular velocity of the system converges. We can infer from this plot that a comparatively negligible amount of fuel ($\approx$ 5 kg) is used by the proportional term in Eq. (4.60) for stabilizing the attitude of the asteroid and spacecraft combination.

Case 7 also gives the worst case modeling errors because $\|\Delta \boldsymbol{J}_{\text{obj}}^{B_{CM}}\|_2 \approx \|\boldsymbol{J}_{\text{obj}}^{B_{CM}}\|_2$ and $\|\Delta \boldsymbol{r}^{S_O/B_{CM}}\|_2 \approx \|\boldsymbol{r}^{S_O/B_{CM}}\|_2$. In Cases 8–11, we study the effect of each of these uncertainties by individually reducing them from their worst case bounds. Note that the control laws tracking the fuel-optimal desired trajectory are unable to stabilize the system because of actuator saturation. The Robust NTCL, which tracks the D+PD based reference trajectory, gives satisfactory performance for these cases too.

4.4.2 Sensitivity Analysis of the Robust Nonlinear Tracking Control Law

We now present detailed sensitivity analysis of the Robust NTCL and the D+PD control strategy based desired attitude trajectory, by varying the asteroid parameters, the initial conditions, and the control law parameters. The parameters that are not explicitly specified are taken from Tables 4.5 and 4.7 and from Case 7 in Table 4.6. Figure 4.5 shows results of numerical simulations over a wide range of asteroid parameters. We observe that the Robust NTCL performs relatively well and the fuel consumed by the spacecraft is upper bounded by 300 kg for the nominal range of asteroid parameters.

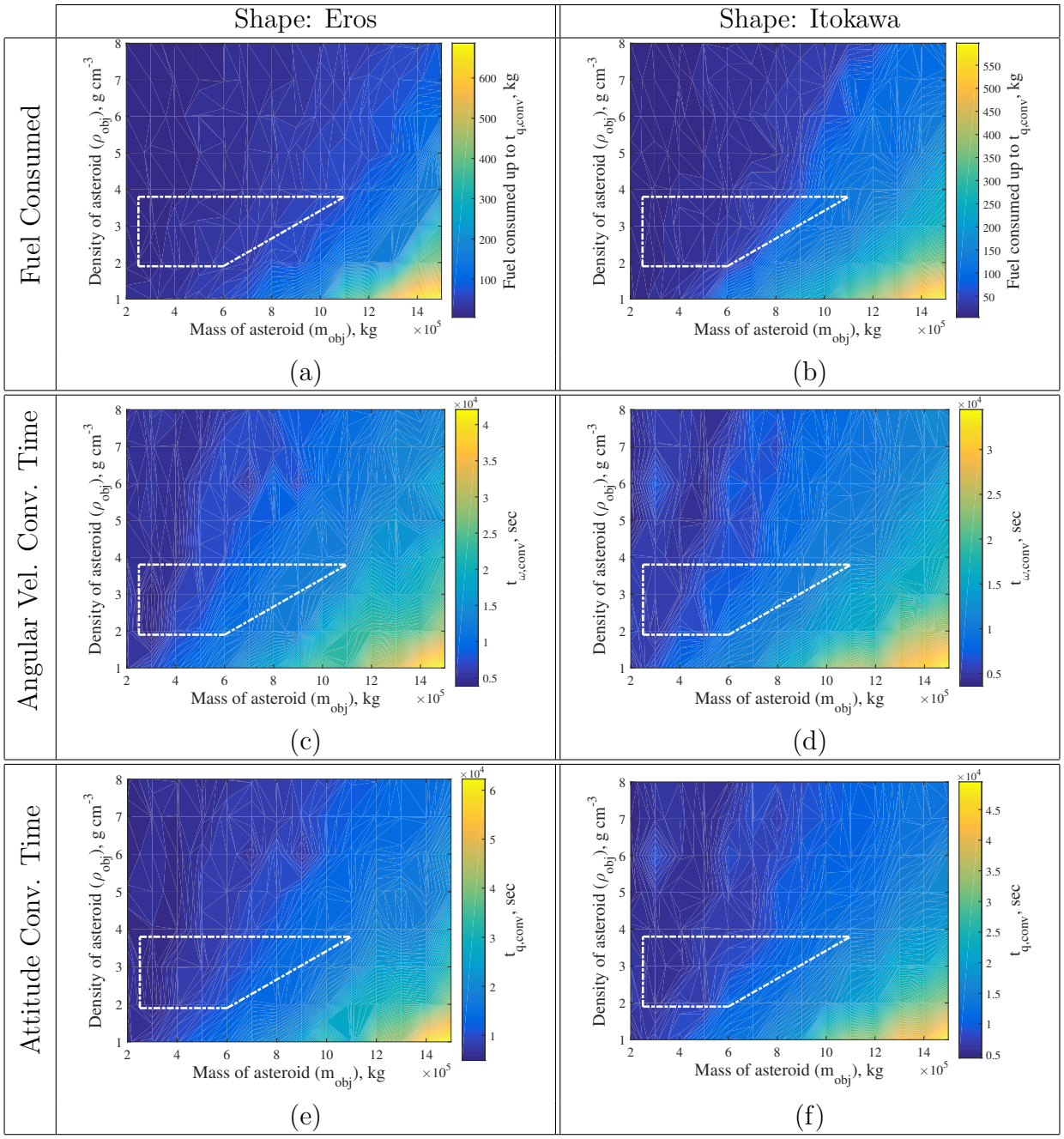


Figure 4.5: Numerical simulations show the performance of the Robust NTCL over a wide range of asteroid parameters for the shape models of Eros and Itokawa respectively. The plots show the variation of the convergence time of the angular velocity ($t_{\omega,\text{conv}}$), the convergence time of the attitude ($t_{q,\text{conv}}$), and the fuel consumed up to time $t_{q,\text{conv}}$ with respect to the mass and density of the model asteroid. The inset white trapezium shows the nominal range of NEO asteroid parameters, i.e., the asteroid's mass is within 2.5-13 $\times 10^5$ kg, the asteroid's density is within 1.9-3.8 g cm^{-3}, and the asteroid's diameter is less than 15 m.

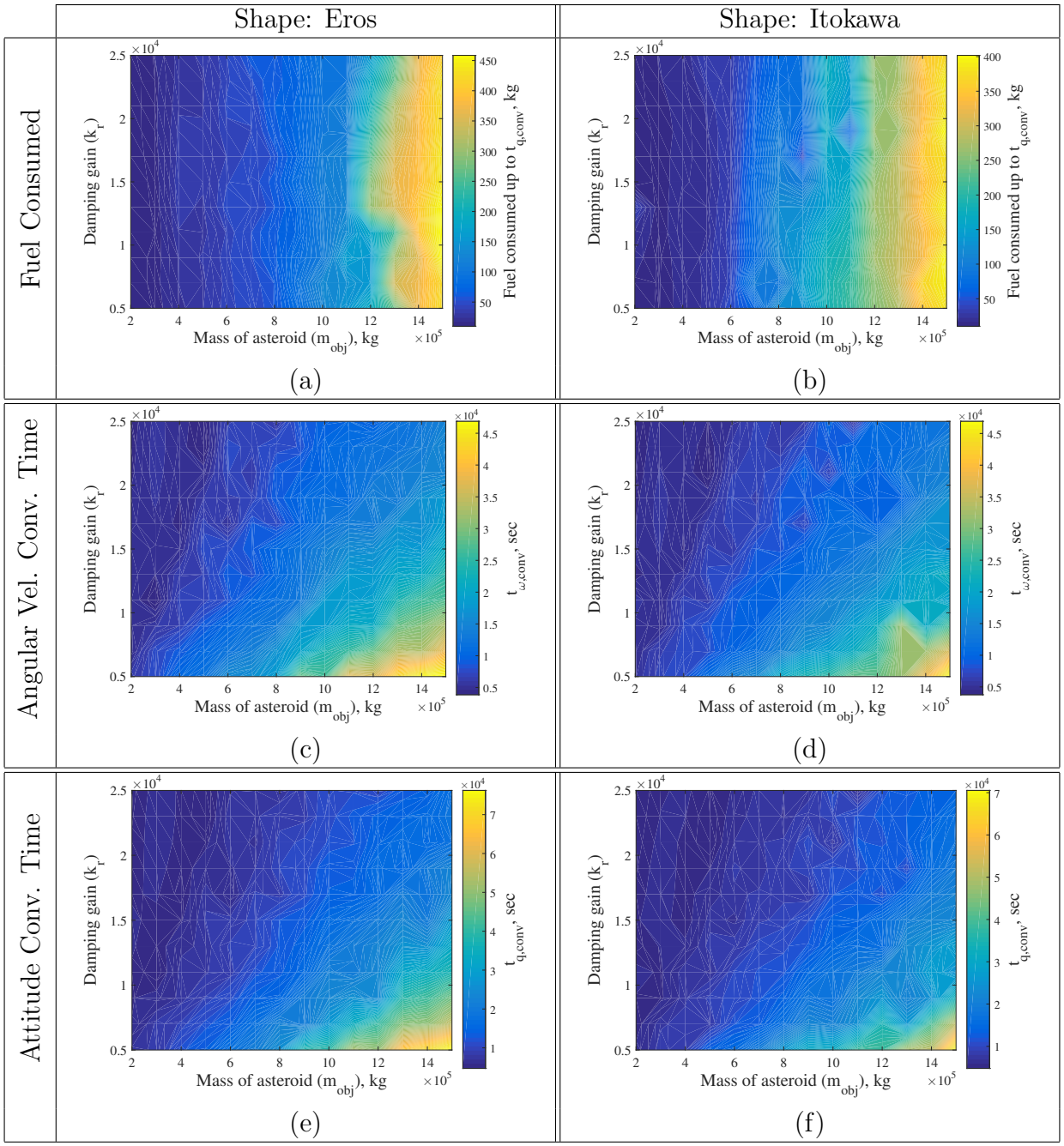


Figure 4.6: Sensitivity plots show the effect of the damping gain $\boldsymbol{K}_r = k_r\mathbf{I}$ on the convergence time of the angular velocity ($t_{\omega,\text{conv}}$), the convergence time of the attitude ($t_{q,\text{conv}}$), and the fuel consumed up to time $t_{q,\text{conv}}$ for the two shape models of Eros and Itokawa respectively.

Previously, we inferred from Fig. 4.4(f) that the damping term $-\boldsymbol{K}_r(\hat{\boldsymbol{\omega}} - \boldsymbol{\omega}_r)$ in the Robust NTCL Eq. (4.13) dictates the fuel consumption and that the effect of the proportional term in Eq. (4.60) is negligible. The effect of this damping gain, which is given by $\boldsymbol{K}_r = k_r\mathbf{I}$,

on the fuel consumption and the convergence time is shown in Fig. 4.6. Even though k_r is varied from 0.5-2.5 $\times 10^4$, its effect on the fuel consumed to stabilize the system is minimal as shown in Fig. 4.6(a,d). On the other hand, increasing k_r reduces the convergence time of the angular velocity as shown in Fig. 4.6(b,e) and the convergence time of the attitude as shown in Fig. 4.6(c,f). If k_r is chosen to be less than 0.5×10^4, then the angular velocity convergence time increases beyond 10^5 sec, which is not desirable. If k_r is chosen to be greater than 2.5×10^4, then the system converges quickly but the control action becomes very sensitive to angular velocity measurement errors. Hence the damping gain of $k_r \approx 10^4$ is ideal for this mission.

The effect of the tuning parameter, which is given by $\mathbf{\Lambda}_r = \lambda_r \mathbf{I}$, on the fuel consumption and the convergence time is shown in Fig. 4.7. As expected, its effect is minimal because the tuning parameter is only used after the angular velocity of the system is sufficiently close to zero in the D+PD based reference trajectory. Therefore, we recommend using any tuning parameter within the range of 0.5-2.5 $\times 10^{-3}$.

It is shown in Fig. 4.5(a,d) that the asteroid with mass $m_{\text{obj}} = 1.1 \times 10^6$ kg and density $\rho_{\text{obj}} = 1.9$ g cm^{-3} consumes the maximum fuel among all asteroids in the nominal range. The effect of the initial angular velocity $\boldsymbol{\omega}_{\text{initial}}$ on the fuel consumption and convergence time for this nominal asteroid is shown in Fig. 4.8. We observe that the fuel consumed by the spacecraft-asteroid combination is upper bounded by 300 kg for all initial conditions, as shown in Fig. 4.8(a,d). Hence we conclude that the fuel consumed by the spacecraft-asteroid combination using the Robust NTCL Eq. (4.13), which tracks the D+PD control strategy based desired attitude trajectory (Section 4.3.2), is upper bounded by 300 kg for the nominal range of asteroid parameters (i.e., the asteroid's mass is within 2.5-13 $\times 10^5$ kg, the asteroid's density is within 1.9-3.8 g cm^{-3}, and the asteroid's diameter is less than 15 m). Note that the convergence times of the angular velocities and the attitudes are satisfactory for all initial conditions. Moreover, the effect of the initial attitude $\boldsymbol{q}_{\text{initial}}$ on the fuel consumption and convergence time is negligible.

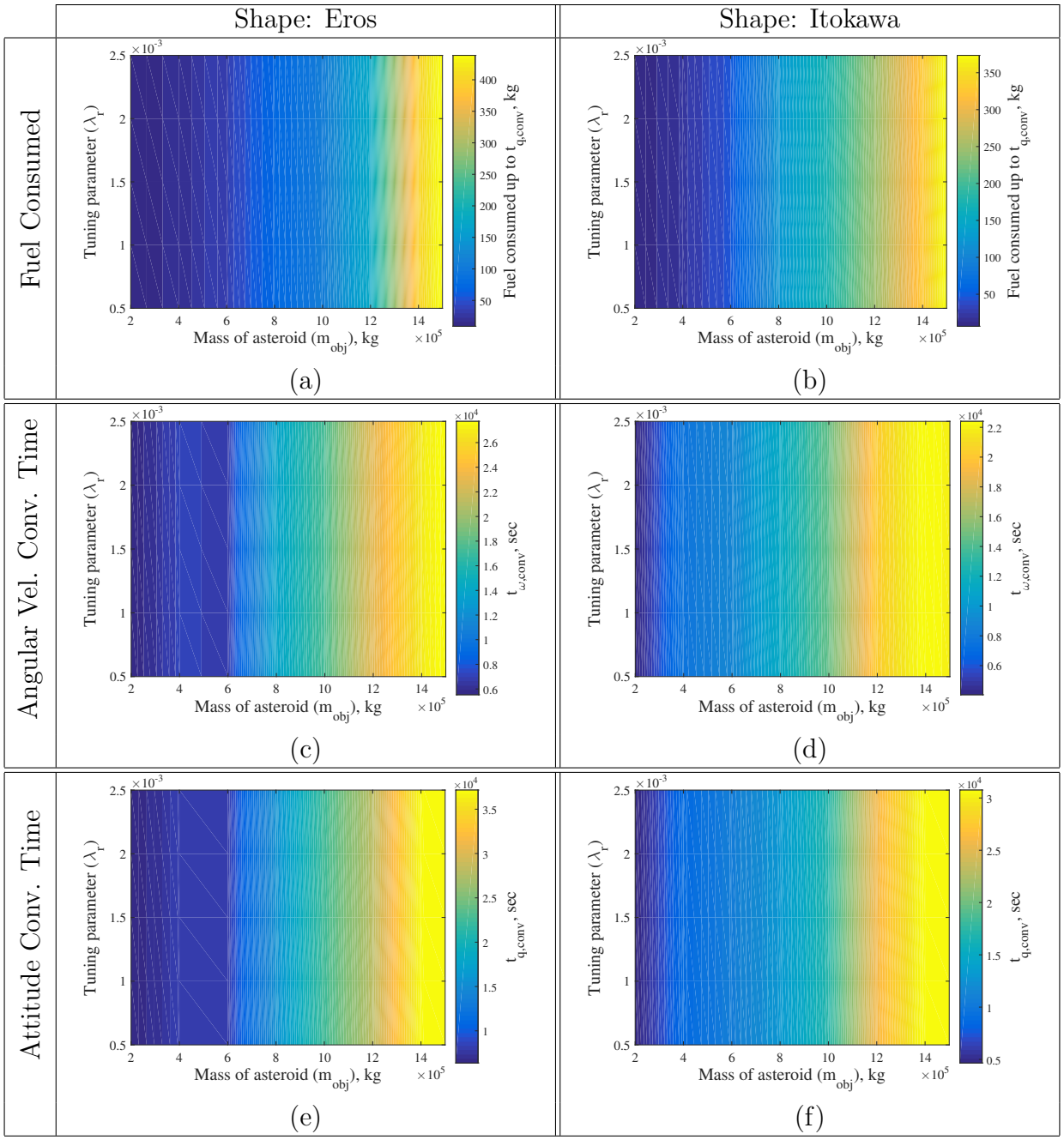


Figure 4.7: Sensitivity plots show the effect of the tuning parameter $\mathbf{\Lambda}_r = \lambda_r \mathbf{I}$ on the convergence time of the angular velocity ($t_{\omega,\text{conv}}$), the convergence time of the attitude ($t_{q,\text{conv}}$), and the fuel consumed up to time $t_{q,\text{conv}}$ for the two shape models of Eros and Itokawa respectively.

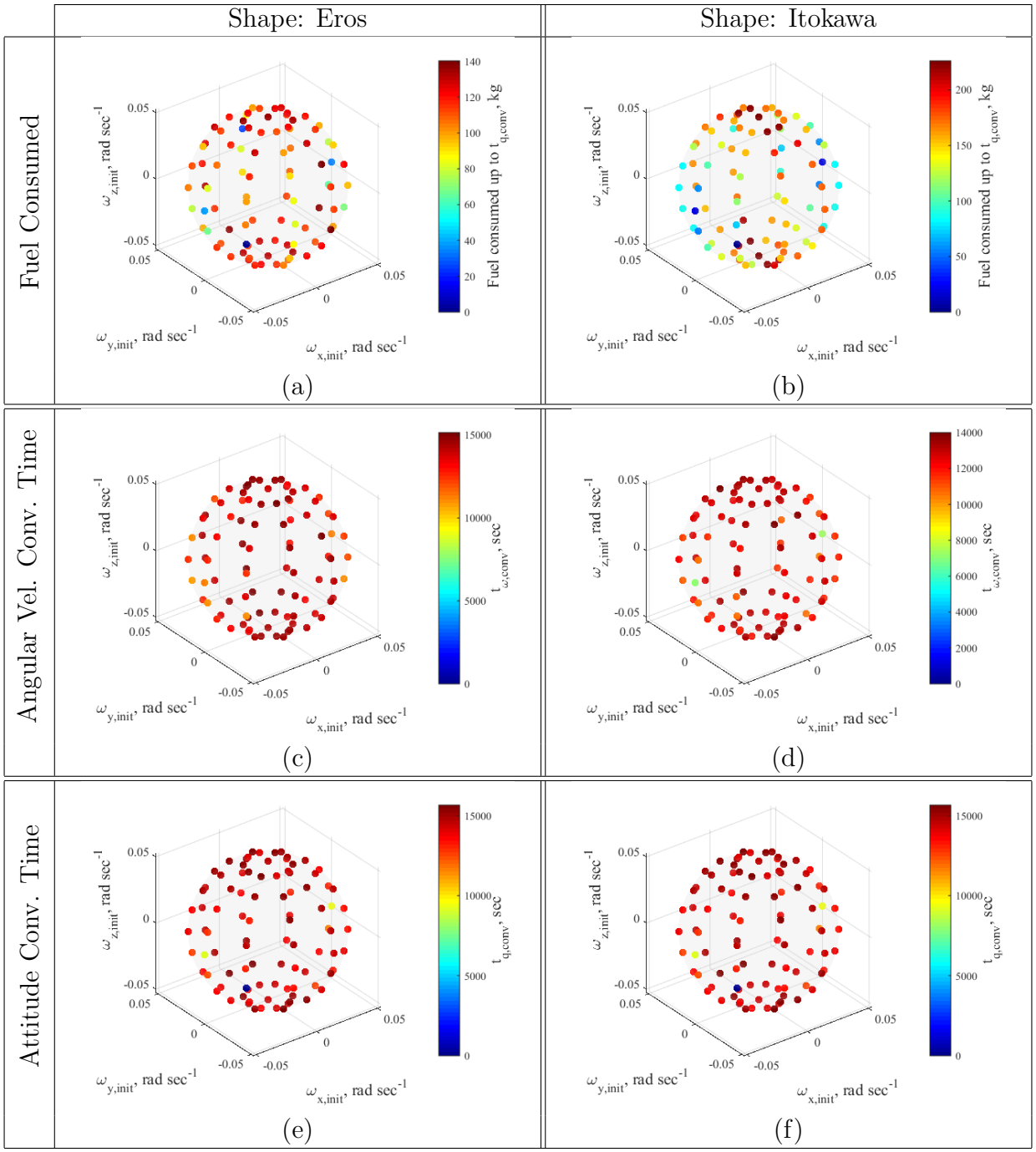


Figure 4.8: Sensitivity plots show the effect of the initial angular velocity $\boldsymbol{\omega}_{\text{initial}}$ on the convergence time of the angular velocity ($t_{\omega,\text{conv}}$), the convergence time of the attitude ($t_{q,\text{conv}}$), and the fuel consumed up to time $t_{q,\text{conv}}$ for the two shape models of Eros and Itokawa respectively, where the asteroid's mass is $m_{\text{obj}} = 1.1 \times 10^6$ kg and the asteroid's density is $\rho_{\text{obj}} = 1.9$ g cm^{-3}. All initial angular velocities are on the sphere with radius of 0.5 rpm.

4.5 Chapter Summary

In this chapter, we presented the development of a new robust nonlinear tracking control law and its extensions. We numerically compared the performance of multiple control laws for the ARM mission type and recognized the best control strategies under varying uncertainties. We envisage that the design guidelines presented in this chapter can be useful for a future asteroid capture or redirect mission.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this dissertation, we presented novel algorithms for shape formation, distributed estimation, and attitude control of multi-agent systems.

In Chapter 2, we presented path planning algorithms for large-scale swarms by using the Eulerian framework. Our highly scalable, robust, and versatile PSG–IMC algorithm ensures that a swarm converges to the desired formation or the unknown target distribution. In our PSG–IMC algorithm, time-inhomogeneous Markov matrices with a desired stationary distribution are systematically constructed using the Hellinger-distance-based feedback gain, which incorporates feedback from the current swarm distribution. These Markov matrices satisfy suitable motion constraints, minimize the expected cost of transitions, and circumvent transitions from bins that are deficient in the number of agents. We also presented a rigorous convergence analysis of the PSG–IMC algorithm. To our knowledge, this PSG–IMC algorithm we have developed is the first path planning strategy that leverages the idea of constructing IMC in real-time based on state feedback.

We have carried out numerical simulations which show that the PSG–IMC algorithm achieves $\approx 6-16$ times reduction in total cost of transitions and $\approx 1.5-2$ times reduction in HD, as compared to existing HMC-based algorithms for shape formation and area exploration applications. This is because the PSG–IMC algorithm avoids undesirable transitions, and the number of transitions at each time instant is proportional to the HD. In the presence of estimation errors, our PSG–IMC algorithm also outperforms the PSG–OT algorithm,

because the pmf of the predicted position of each agent converges to the desired formation regardless of estimation errors. We have also demonstrated hardware experiments using multiple quadrotors, where our Voronoi partition-based collision-free trajectory generation algorithm is used to track the higher-level trajectory generated by the PSG–IMC algorithm. The reconfiguration of spacecraft swarms in Earth orbit shows that these algorithms are also suitable for application to spacecraft swarms.

In Chapter 3, we presented two novel, discrete-time distributed estimation algorithms, namely the DBF algorithm and the BCF algorithm. The DBF algorithm ensures that each agent's estimated likelihood function converges to an error ball around the joint likelihood function of the centralized multi-sensor Bayesian filtering algorithm. We have rigorously proven the convergence properties of this algorithm. We have shown an explicit connection between the time step of the distributed estimation algorithm and the time-scale of the target dynamics. We also presented the DBF-Kalman information filtering algorithm for the special case of linear-Gaussian models. The properties of these algorithms are illustrated using numerical examples. We envisage that the novel proof techniques presented in Chapter 3 can also be used in other distributed estimation algorithms which rely on the LogOP scheme.

In Chapter 4, we presented a new robust nonlinear tracking control law for attitude control of a spacecraft with large uncertainty, which guarantees both global exponential convergence to the desired attitude trajectory and bounded tracking errors (in the sense of finite-gain $\mathcal{L}_p$ stability and ISS) in the presence of uncertainties and disturbances. The benefits of this new attitude tracking control law include superior robustness due to no feed-forward cancellation and straightforward extensions to integral control and various attitude representations such as MRPs and SO(3). We presented a comparison of the resultant disturbance torques produced by various types of attitude control laws and concluded that the proposed control law could produce a small resultant disturbance torque if the desired trajectory was designed appropriately. We also discussed techniques for obtaining fuel-optimal or resultant disturbance torque minimizing desired attitude trajectories for these

nonlinear tracking control laws.

We then numerically compared the performance of multiple control laws, such as the proposed robust nonlinear tracking control law, nonlinear adaptive control, and the D+PD linear control strategy, for a spacecraft-asteroid combination with large modelling uncertainty. We illustrated that in the presence of small measurement errors and small modeling uncertainties, which could be achieved using online system identification, the robust nonlinear tracking control law that tracked a fuel-optimal reference trajectory was the best strategy because it consumed the least amount of fuel. We also showed that a comparatively negligible amount of fuel was needed for driving the attitude of the combined system to the desired orientation after the angular velocity of the system was stabilized. One caveat of using both optimal control and nonlinear tracking control is that the spacecraft should have sufficient computational power for online system identification and real time fuel-optimal trajectory generation.

On the other hand, in the presence of large modeling uncertainties, measurement errors, and actuator saturations, or in the absence of sufficient computational power on board the spacecraft, the simple linear D+PD control strategy resulted in good performance. This performance was further enhanced with properties of superior robustness and tracking convergence if the robust nonlinear tracking control law was used to globally exponentially track a desired attitude trajectory that was generated using the D+PD linear control strategy. We envisage that the design guidelines presented in this chapter can be useful for a future asteroid capture or redirect mission.

5.2 Future Work

The algorithms introduced in this dissertation can also be used for other applications. Potential avenues for future research are as follows:

- The PSG–IMC and PSG–OT algorithms presented in Chapter 2 can be adapted to

solve other cooperative control tasks, such as surveillance, task allocation, and coverage, since such problems can also be cast as shape formation or area exploration problems. Future research could also focus on tightly integrating a lower-level guidance and control algorithm with these algorithms. We envisage that the proposed algorithms will facilitate the development of autonomous swarm robotic systems that are capable of performing a variety of complex tasks, by providing a versatile, robust, and scalable path planning strategy.

- In this dissertation, we have independently pursued the development of motion planning and distributed estimation algorithms. The errors introduced by the distributed estimation algorithm for estimating the current swarm distribution affects the motion planning algorithms. Similarly the shape of the formation affects the error in the estimated current swarm distribution through the communication network topology. But the effect of these errors and methods for minimizing the cumulative error have not been investigated. Future research could focus on the development of a holistic algorithm that incorporates distributed estimation, guidance, navigation, and control algorithms for multi-agent systems.

- The distributed estimation algorithms in Chapter 3 can only track a single target. Future research could focus on extending them to multi-target tracking by solving the data-association problem using the extra information contained in the probability distributions.

- The nonlinear attitude control law in Chapter 4 assumes that the combined spacecraft and captured object form a rigid body. In practical scenarios, there is slippage between the two bodies and multiple bending and torsional modes are introduced due to structural flexibility. Future research could focus on augmenting this control law so that it performs under such practical constraints.

Appendix A

Background Results on Products of Stochastic Matrices

In this section, we first state some definitions and results used in the proofs of Theorems 2.6 and 2.7.

Definition A.1. [135, pp. 92, 149] *(Asymptotic Homogeneity)* A sequence of stochastic matrices $\boldsymbol{P}_k$, $k \geq 1$ is said to be asymptotically homogeneous (with respect to $\boldsymbol{d}$) if there exists a probability (row) vector $\boldsymbol{d}$ such that $\lim_{k\to\infty} \boldsymbol{d}\boldsymbol{P}_k = \boldsymbol{d}$. □

Definition A.2. [135, pp. 92, 149] *(Strong Ergodicity)* The forward matrix product $\boldsymbol{U}_{T,r} := \boldsymbol{P}_T \boldsymbol{P}_{T+1} \cdots \boldsymbol{P}_{r-1}$, formed from a sequence of stochastic matrices $\boldsymbol{P}_k$, $k \geq 1$, is said to be strongly ergodic if for each i, ℓ, T, we get $\lim_{r\to\infty} \boldsymbol{U}_{T,r}[i,\ell] = \boldsymbol{v}[\ell]$, where $\boldsymbol{v}$ is a probability vector and the element $\boldsymbol{v}[\ell]$ is independent of i. Therefore, $\boldsymbol{v}$ is the unique limit vector and $\lim_{r\to\infty} \boldsymbol{U}_{T,r} = \mathbf{1}\boldsymbol{v}$. □

Theorem A.1. *[135, pp. 150] If (i) the forward matrix product $\boldsymbol{U}_{T,r}$ is primitive and (ii) there exists γ (independent of k) such that:*

$$0 < \gamma \leq \min_{i,\ell}{}^{+} \boldsymbol{P}_k[i,\ell]\,, \tag{A.1}$$

where $\min^+$ refers to the minimum of the positive elements, then asymptotic homogeneity of $\boldsymbol{P}_k$ is necessary and sufficient for strong ergodicity of $\boldsymbol{U}_{T,r}$.

Theorem A.2. *[135, pp. 149] If (i) all $\boldsymbol{P}_k$, $k \geq 1$ are irreducible and (ii) there exists γ (independent of k) such that (A.1) is satisfied, then asymptotic homogeneity of $\boldsymbol{P}_k$ (with respect to $\boldsymbol{d}$) is equivalent to $\lim_{k\to\infty} \boldsymbol{e}_k \to \boldsymbol{e}$, where $\boldsymbol{e}_k$ is the unique stationary distribution vector corresponding to $\boldsymbol{P}_k$, and $\boldsymbol{e}$ is a limit vector. Moreover, $\boldsymbol{d} = \boldsymbol{e}$.*

Corollary A.3. *[135, pp. 150] Under the prior conditions (i) and (ii) of Theorem A.2 and if (iii) $\boldsymbol{U}_{T,r}$ is strongly ergodic with unique limit vector $\boldsymbol{v}$, then $\boldsymbol{v} = \boldsymbol{e}$.*

Lemma A.4. *The matrix $\boldsymbol{U}^j_{k,k+n_{rec}-1}$ is a positive matrix.*

Proof: Proof by Contradiction. Let us assume that there exists $i, \ell \in \{1, \ldots, n_{\text{rec}}\}$ and $i \neq \ell$ such that $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$.

Step 1. In order to satisfy the assumption $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$, we need $\boldsymbol{P}^j_{q,\text{sub}}[i,\ell] = 0$ for all $q \in \{k, \ldots, k+n_{\text{rec}}-2\}$. This immediately follows from:

$$\begin{aligned}\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] \geq & \boldsymbol{P}^j_{k,\text{sub}}[i,\ell]\Big(\prod_{s=k+1}^{k+n_{\text{rec}}-2} \boldsymbol{P}^j_{s,\text{sub}}[\ell,\ell]\Big) + \Big(\prod_{s=k}^{k+n_{\text{rec}}-3} \boldsymbol{P}^j_{s,\text{sub}}[i,i]\Big)\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}[i,\ell] \\ & + \sum_{q=k+1}^{k+n_{\text{rec}}-3}\left(\Big(\prod_{s=k}^{q-1}\boldsymbol{P}^j_{s,\text{sub}}[i,i]\Big)\boldsymbol{P}^j_{q,\text{sub}}[i,\ell]\Big(\prod_{s=q+1}^{k+n_{\text{rec}}-2}\boldsymbol{P}^j_{s,\text{sub}}[\ell,\ell]\Big)\right). \qquad \text{(A.2)}\end{aligned}$$

since all the diagonal elements of all matrices $\boldsymbol{P}^j_{k,\text{sub}}, \ldots, \boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}$ are positive. Therefore, $\boldsymbol{P}^j_{q,\text{sub}}[i,\ell] = 0$ for all $q \in \{k, \ldots, k+n_{\text{rec}}-2\}$ is a necessary condition for $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$.

Moreover, $\boldsymbol{U}^j_{k,t}[i,\ell] = 0$ for all $t \in \{k+1, \ldots, k+n_{\text{rec}}-1\}$ is a necessary condition for $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$. Otherwise, if $\boldsymbol{U}^j_{k,t'}[i,\ell] > 0$ for some $t' \in \{k+1, \ldots, k+n_{\text{rec}}-2\}$, then it follows that:

$$\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] \geq \boldsymbol{U}^j_{k,t'}[i,\ell]\Big(\prod_{s=t'}^{k+n_{\text{rec}}-2}\boldsymbol{P}^j_{s,\text{sub}}[\ell,\ell]\Big) > 0\,. \qquad \text{(A.3)}$$

Hence, it is not possible to reach from bin $B[i]$ to bin $B[\ell]$ within any of the $n_{\text{rec}}-1$ time instants under the assumption $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$.

Step 2. Since the matrix $\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}$ is irreducible, there is at least one bin $B[s_1]$ such that

$\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}[s_1,\ell] > 0$, where $s_1 \in \{1, \ldots, n_{\text{rec}}\}\backslash\{i,\ell\}$.

Under the assumption $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] = 0$, we need $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-2}[i,s_1] = 0$. Otherwise, if $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-2}[i,s_1] > 0$, then $\boldsymbol{U}^j_{k,k+n_{\text{rec}}-1}[i,\ell] \geq \boldsymbol{U}^j_{k,k+n_{\text{rec}}-2}[i,s_1]\boldsymbol{P}^j_{k+n_{\text{rec}}-2,\text{sub}}[s_1,\ell] > 0$. It follows from the previous step that $\boldsymbol{U}^j_{k,t}[i,s_1] = 0$ for all $t \in \{k+1, \ldots, k+n_{\text{rec}}-2\}$ is a

necessary condition for $\boldsymbol{U}^j_{k,k+n_{\rm rec}-2}[i,s_1]=0$.

Hence, it is not possible to reach from bin $B[i]$ to bin $B[\ell]$ or bin $B[s_1]$ within any of the $n_{\rm rec}-2$ time instants under the assumption $\boldsymbol{U}^j_{k,k+n_{\rm rec}-1}[i,\ell]=0$.

Step 3. Since the matrix $\boldsymbol{P}^j_{k+n_{\rm rec}-3,{\rm sub}}$ is irreducible, there is at least one bin $B[s_2]$ such that either $\boldsymbol{P}^j_{k+n_{\rm rec}-3,{\rm sub}}[s_2,\ell]>0$ or $\boldsymbol{P}^j_{k+n_{\rm rec}-3,{\rm sub}}[s_2,s_1]>0$, where $s_2\in\{1,\ldots,n_{\rm rec}\}\backslash\{i,\ell,s_1\}$.

Under the assumption $\boldsymbol{U}^j_{k,k+n_{\rm rec}-1}[i,\ell]=0$ and condition $\boldsymbol{U}^j_{k,k+n_{\rm rec}-2}[i,s_1]=0$, we need $\boldsymbol{U}^j_{k,k+n_{\rm rec}-3}[i,s_2]=0$. It follows from the previous step that $\boldsymbol{U}^j_{k,t}[i,s_2]=0$ for all $t\in\{k+1,\ldots,k+n_{\rm rec}-3\}$ is a necessary condition for $\boldsymbol{U}^j_{k,k+n_{\rm rec}-3}[i,s_2]=0$.

Hence, it is not possible to reach from bin $B[i]$ to bin $B[\ell]$ or bin $B[s_1]$ or bin $B[s_2]$ within any of the $n_{\rm rec}-3$ time instants under the assumption $\boldsymbol{U}^j_{k,k+n_{\rm rec}-1}[i,\ell]=0$.

Step 4. If we continue the above argument till the first time instant, we get that under assumption $\boldsymbol{U}^j_{k,k+n_{\rm rec}-1}[i,\ell]=0$, it is not possible to reach from bin $B[i]$ to any other bin during the first time instant. But this is a contradiction since the matrix $\boldsymbol{P}^j_{k,{\rm sub}}$ is irreducible. Therefore our assumption $\boldsymbol{U}^j_{k,k+n_{\rm rec}-1}[i,\ell]=0$ is incorrect. ■

Appendix B

Effect of Weight on the Likelihood Function

The effect on the update step of the information filter due to the weight α on the likelihood function is shown here. We start with the update step for linear-Gaussian model:

$$\mathcal{W}_k^i(\boldsymbol{x}) = p(\boldsymbol{x}_{k|k}) \propto$$
$$\exp\left(-\frac{\alpha}{2}(\boldsymbol{y}_k^i - \boldsymbol{H}_k^i\boldsymbol{x}_k)^T(\boldsymbol{R}_k^i)^{-1}(\boldsymbol{y}_k^i - \boldsymbol{H}_k^i\boldsymbol{x}_k) - \frac{1}{2}(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}^i)^T(\boldsymbol{P}_{k|k-1}^i)^{-1}(\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}^i)\right) \tag{B.1}$$

The MAP estimate of the state is defined by:

$$\left.\frac{\partial \log p(\boldsymbol{x}_{k|k})}{\partial \boldsymbol{x}_k}\right|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k^{MAP}} = 0\,. \tag{B.2}$$

We get from (B.1) that:

$$\frac{\partial \log p(\boldsymbol{x}_{k|k})}{\partial \boldsymbol{x}_k} = \alpha(\boldsymbol{y}_k^i - \boldsymbol{H}_k^i\boldsymbol{x}_k)^T(\boldsymbol{R}_k^i)^{-1}\boldsymbol{H}_k^i - (\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{k|k-1}^i)^T(\boldsymbol{P}_{k|k-1}^i)^{-1}\,.$$

Substituting into (B.2), we get:

$$(\boldsymbol{x}_k)^T\left((\boldsymbol{H}_k^i)^T\alpha(\boldsymbol{R}_k^i)^{-1}\boldsymbol{H}_k^i + (\boldsymbol{P}_{k|k-1}^i)^{-1}\right) = (\boldsymbol{y}_k^i)^T\alpha(\boldsymbol{R}_k^i)^{-1}\boldsymbol{H}_k^i + (\hat{\boldsymbol{x}}_{k|k-1}^i)^T(\boldsymbol{P}_{k|k-1}^i)^{-1}\,,$$
$$\boldsymbol{x}_k = \left((\boldsymbol{H}_k^i)^T\alpha(\boldsymbol{R}_k^i)^{-1}\boldsymbol{H}_k^i + (\boldsymbol{P}_{k|k-1}^i)^{-1}\right)^{-1}\left((\boldsymbol{H}_k^i)^T\alpha(\boldsymbol{R}_k^i)^{-1}\boldsymbol{y}_k^i + (\boldsymbol{P}_{k|k-1}^i)^{-1}\hat{\boldsymbol{x}}_{k|k-1}^i\right)\,.$$

It follows from the matrix inverse lemma[1] that:

$$\boldsymbol{x}_k = \hat{\boldsymbol{x}}^i_{k|k-1} + \boldsymbol{P}^i_{k|k-1}(\boldsymbol{H}^i_k)^T \left(\alpha^{-1}\boldsymbol{R}^i_k + \boldsymbol{H}^i_k \boldsymbol{P}^i_{k|k-1}(\boldsymbol{H}^i_k)^T\right)^{-1} \left(\boldsymbol{y}^i_k - \boldsymbol{H}^i_k \hat{\boldsymbol{x}}^i_{k|k-1}\right) .$$

Therefore, we get:

$$\begin{aligned}
\boldsymbol{K}^i_k &= \boldsymbol{P}^i_{k|k-1} \left(\boldsymbol{H}^i_k\right)^T \left(\boldsymbol{H}^i_k \boldsymbol{P}^i_{k|k-1} \left(\boldsymbol{H}^i_k\right)^T + \alpha^{-1}\boldsymbol{R}^i_k\right)^{-1} , \\
\hat{\boldsymbol{x}}^i_{k|k} &= \hat{\boldsymbol{x}}^i_{k|k-1} + \boldsymbol{K}^i_k \left(\boldsymbol{y}^i_k - \boldsymbol{H}^i_k \hat{\boldsymbol{x}}^i_{k|k-1}\right) , \\
\boldsymbol{P}^i_{k|k} &= \left(\mathbf{I} - \boldsymbol{K}^i_k \boldsymbol{H}^i_k\right) \boldsymbol{P}^i_{k|k-1} .
\end{aligned}$$

We get the following new update rules for Kalman information filter:

$$\hat{\boldsymbol{z}}^i_{k|k} = \hat{\boldsymbol{z}}^i_{k|k-1} + \alpha \boldsymbol{i}^i_k = (\boldsymbol{P}^i_{k|k-1})^{-1}\hat{\boldsymbol{x}}^i_{k|k-1} + \alpha(\boldsymbol{H}^i_k)^T(\boldsymbol{R}^i_k)^{-1}\boldsymbol{y}^i_k , \tag{B.3}$$

$$\boldsymbol{Z}^i_{k|k} = \boldsymbol{Z}^i_{k|k-1} + \alpha \boldsymbol{I}^i_k = (\boldsymbol{P}^i_{k|k-1})^{-1} + \alpha(\boldsymbol{H}^i_k)^T(\boldsymbol{R}^i_k)^{-1}\boldsymbol{H}^j_k . \tag{B.4}$$

[1]The inverse of the matrix $A = B^{-1} + CD^{-1}C^T$, is given by the matrix $A^{-1} = B - BC(D + C^T BC)^{-1}C^T B$.

Appendix C

Contraction Theory

In this chapter, we use contraction theory to prove the stability of control laws. In this section, we present some results on contraction theory from [204, 12]. Readers are referred to these references for detailed descriptions and proofs for the following theorems.

Lemma C.1. *[204] (Contraction Analysis) We consider a smooth nonlinear non-autonomous system*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t), \qquad \boldsymbol{x}(t) \in \mathbb{R}^n \,. \tag{C.1}$$

A virtual displacement $\delta\boldsymbol{x}$ is defined as an infinitesimal displacement at fixed time, and $\boldsymbol{\Theta}(\boldsymbol{x}, t)$ is a smooth coordinate transformation of the virtual displacement such that $\delta\boldsymbol{z} = \boldsymbol{\Theta}\delta\boldsymbol{x}$. Then if there exists a positive λ and a uniformly positive definite metric, $\boldsymbol{M}(\boldsymbol{x}, t) = \boldsymbol{\Theta}(\boldsymbol{x}, t)^T\boldsymbol{\Theta}(\boldsymbol{x}, t)$, such that

$$\frac{d}{dt}\Big(\delta\boldsymbol{z}^T\delta\boldsymbol{z}\Big) = \frac{d}{dt}\Big(\delta\boldsymbol{x}^T\boldsymbol{M}(\boldsymbol{x}, t)\delta\boldsymbol{x}\Big) = \delta\boldsymbol{x}^T\Big(\dot{\boldsymbol{M}} + \Big(\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\Big)^T\boldsymbol{M} + \boldsymbol{M}\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\Big)\delta\boldsymbol{x} \leq -2\lambda\delta\boldsymbol{x}^T\boldsymbol{M}(\boldsymbol{x}, t)\delta\boldsymbol{x}\,, \tag{C.2}$$

then all system trajectories converge exponentially fast to a single trajectory regardless of the initial conditions ($\delta\boldsymbol{z},\ \delta\boldsymbol{x} \to \boldsymbol{0}$) at a rate of λ (i.e., contracting), and λ is the largest eigenvalue of the symmetric part of $(\dot{\boldsymbol{\Theta}} + \boldsymbol{\Theta}\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}})\boldsymbol{\Theta}^{-1}$.

We define the $\mathcal{L}_p$ norm in the extended space $\mathcal{L}_{pe}, p \in [1, \infty]$ as follows [12]:

$$\|(\boldsymbol{u})_\tau\|_{\mathcal{L}_p} = \left(\int_0^\tau \|\boldsymbol{u}(t)\|_2^p \, dt \right)^{1/p} < \infty, \quad p \in [1, \infty), \qquad \|(\boldsymbol{u})_\tau\|_{\mathcal{L}_\infty} = \sup_{t \geq 0} \|(\boldsymbol{u}(t))_\tau\|_2 < \infty, \tag{C.3}$$

where $(\boldsymbol{u})_\tau$ is a truncation of $\boldsymbol{u}(t)$, i.e., $(\boldsymbol{u}(t))_\tau = 0$ for $t \geq \tau$, $\tau \in [0, \infty)$ while $(\boldsymbol{u}(t))_\tau = \boldsymbol{u}(t)$ for $0 \leq t \leq \tau$.

Lemma C.2. *[12] (Robust Contraction and Link to $\mathcal{L}_p$ Stability and ISS) Let $P_1(t)$ be a solution of the contracting system (C.1), globally exponentially tending to a single trajectory at a contraction rate of λ. Eq. (C.1) is now perturbed as:*

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), t) + \boldsymbol{d}(\boldsymbol{x}(t), t), \tag{C.4}$$

and $P_2(t)$ denotes the trajectory of Eq. (C.4). Then the smallest path integral (i.e., distance) $R(t) = \int_{P_1}^{P_2} \|\delta \boldsymbol{z}(t)\|_2 = \int_{P_1}^{P_2} \|\boldsymbol{\Theta}(\boldsymbol{x}, t) \delta \boldsymbol{x}(t)\|_2, \forall t \geq 0$ exponentially converges to the following error ball[204]

$$\lim_{t \to \infty} R(t) \leq \sup_{\boldsymbol{x}, t} \frac{\|\boldsymbol{\Theta}(\boldsymbol{x}, t) \boldsymbol{d}(\boldsymbol{x}(t), t)\|_2}{\lambda}, \tag{C.5}$$

with $\boldsymbol{\Theta} \boldsymbol{d} \in \mathcal{L}_\infty$. Furthermore, if $\boldsymbol{d}(\boldsymbol{x}(t), t) \in \mathcal{L}_{pe}$, then Eq. (C.4) is finite-gain $\mathcal{L}_p$ stable with $p \in [1, \infty]$ for an output function $\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{d}, t)$ with $\int_{Y_1}^{Y_2} \|\delta \boldsymbol{y}\|_2 \leq \eta_1 \int_{P_1}^{P_2} \|\delta \boldsymbol{x}\|_2 + \eta_2 \|\boldsymbol{d}\|_2$, $\exists \eta_1, \eta_2 \geq 0$, since

$$\left\| \left(\int_{Y_1}^{Y_2} \|\delta \boldsymbol{y}\|_2 \right)_\tau \right\|_{\mathcal{L}_p} \leq \frac{\eta_1 \zeta R(0)}{\sqrt{\lambda_{\min}(\boldsymbol{M})}} + \left(\frac{\eta_1}{\lambda} + \eta_2 \right) \frac{\|(\boldsymbol{\Theta} \boldsymbol{d})_\tau\|_{\mathcal{L}_p}}{\sqrt{\lambda_{\min}(\boldsymbol{M})}}, \quad \forall \tau \in [0, \infty), \tag{C.6}$$

where $Y_1(t)$ and $Y_2(t)$ denote the output trajectories of the original contracting system and its perturbed system respectively and $\zeta = 1$ if $p = \infty$ or $\zeta = 1/(\lambda p)^{1/p}$ if $p \in [1, \infty)$. The perturbed system (C.4) is also input-to-state stable (ISS).

References

[1] J. Brophy, F. Culick, and L. Friedman, "Asteroid retrieval feasibility study," tech. rep., Keck Institute for Space Studies, California Institute of Technology, Pasadena, CA, April 2012.

[2] N. A. Chaturvedi, A. K. Sanyal, and N. H. McClamroch, "Rigid–body attitude control," *IEEE Control Syst. Mag.*, vol. 31, no. 3, pp. 30–51, 2011.

[3] "National Aeronautics and Space Administration, "NASA/Advanced Concepts Lab"." URL: http://www.nasa.gov/multimedia/imagegallery/image_feature_2520.html [cited January 21, 2016].

[4] "NASA Identifying Candidate Asteroids for Redirect Mission." URL: http://www.nasa.gov/content/nasa-identifying-candidate-asteroids-for-redirect-mission/ [cited January 21, 2016].

[5] D. Morgan, S.-J. Chung, L. Blackmore, B. Acikmese, D. Bayard, and F. Y. Hadaegh, "Swarm-keeping strategies for spacecraft under J2 and atmospheric drag perturbations," *J. Guid. Control Dyn.*, vol. 35, no. 5, pp. 1492 – 1506, 2012.

[6] R. W. Gaskell, "Gaskell Eros Shape Model V1.0. NEAR-A-MSI-5-EROSSHAPE-V1.0.," tech. rep., NASA Planetary Data System, 2008.

[7] R. Gaskell, J. Saito, M. Ishiguro, T. Kubota, T. Hashimoto, N. Hirata, S. Abe, O. Barnouin-Jha, and D. Scheeres, "Gaskell Itokawa Shape Model V1.0. HAY-A-AMICA-5-ITOKAWASHAPE-V1.0.," tech. rep., NASA Planetary Data System, 2008.

[8] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[9] R. Gross, M. Bonani, F. Mondada, and M. Dorigo, "Autonomous self-assembly in swarm-bots," *IEEE Trans. Robotics*, vol. 22, pp. 1115–1130, Dec. 2006.

[10] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

[11] P. Yang, R. Freeman, and K. M. Lynch, "Multi-agent coordination by decentralized estimation and control," *IEEE Trans. Autom. Control*, vol. 53, no. 11, pp. 2480–2496, 2008.

[12] S.-J. Chung, S. Bandyopadhyay, I. Chang, and F. Y. Hadaegh, "Phase synchronization control of complex networks of Lagrangian systems on adaptive digraphs," *Automatica*, vol. 49, pp. 1148–1161, May 2013.

[13] D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Model predictive control of swarms of spacecraft using sequential convex programming," *J. Guid. Control Dyn.*, vol. 37, no. 6, pp. 1725–1740, 2014.

[14] L. Barnes, W. Alvis, M.-A. Fields, K. Valavanis, and W. Moreno, "Heterogeneous swarm formation control using bivariate normal functions to generate potential fields," in *IEEE Workshop Distributed Intell. Syst.: Collective Intelligence and Its Applicat.*, pp. 85–94, 2006.

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

[16] J. S. Shamma, *Cooperative Control of Distributed Multi–Agent Systems*. Chichester, West Sussex, England: John Wiley & Sons, 2007.

[17] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, *et al.*, "ARGoS: a modular, multi-engine simulator for heterogeneous swarm robotics," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, (San Francisco, CA), pp. 5027–5034, Sept. 2011.

[18] I. Chattopadhyay and A. Ray, "Supervised self-organization of homogeneous swarms using ergodic projections of Markov chains," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1505–1515, 2009.

[19] B. Açikmeşe and D. S. Bayard, "A Markov chain approach to probabilistic swarm guidance," in *Proc. Amer. Control Conf.*, pp. 6300–6307, June 2012.

[20] J. M. Hereford, "Analysis of a new swarm search algorithm based on trophallaxis," in *IEEE Congr. Evol. Computation*, (Barcelona, Spain), July 2010.

[21] J. M. Hereford and M. A. Siebold, *Swarm Robotics from Biology to Robotics*, ch. Bio-Inspired Search Strategies for Robot Swarms, pp. 1–25. Rijeka, Croatia: InTech, 2010.

[22] A. R. Mesquita, J. P. Hespanha, and K. Åström, "Optimotaxis: A stochastic multi-agent on site optimization procedure," in *Proc. Hybrid Systems: Computation and Control*, (St. Louis, MO), Apr. 2008.

[23] D. J. Bruemmer, D. D. Dudenhoeffer, M. D. McKay, and M. O. Anderson, "A robotic swarm for spill finding and perimeter formation," in *Spectrum*, (Reno, NV), 2002.

[24] J. E. Hurtado, R. D. Robinett, C. R. Dohrmann, and S. Y. Goldsmith, "Decentralized control for a swarm of vehicles performing source localization," *J. Intell. Robotic Syst.*, vol. 41, no. 1, pp. 1–18, 2004.

[25] W. Liu, A. Winfield, J. Sa, J. Chen, and L. Dou, *Swarm robotics*, ch. Strategies for energy optimisation in a swarm of foraging robots, pp. 14–26. Springer, 2007.

[26] N. Correll and A. Martinoli, "Robust distributed coverage using a swarm of miniature robots," in *IEEE Int. Conf. Robotics Automation*, (Rome, Italy), Apr. 2007.

[27] E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, and M. Dorigo, "Self-organized flocking with a mobile robot swarm: a novel motion control method," *Adaptive Behavior*, Oct. 2012.

[28] J. Cortés, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robotics and Automation*, vol. 20, pp. 243–255, April 2004.

[29] N. Napp and E. Klavins, "Load balancing for multi-robot construction," in *IEEE Int. Conf. Robotics Automation*, (Shanghai, China), May 2011.

[30] S. Berman, A. Halasz, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[31] A. Cornejo, A. J. Lynch, E. Fudge, S. Bilstein, M. Khabbazian, and J. McLurkin, "Scale-free coordinates for multi-robot systems with bearing-only sensors," *Int. J. Robotics Research*, vol. 32, no. 12, pp. 1459–1474, 2013.

[32] M. Turpin, N. Michael, and V. Kumar, "CAPT: Concurrent assignment and planning of trajectories for multiple robots," *Int. J. Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.

[33] R. M. Murray, "Murray recent research in cooperative control of multivehicle systems," *ASME J. Dyn. Syst. Meas. Control*, vol. 129, pp. 571–583, May 2007.

[34] J. Grace and J. Baillieul, "Stochastic strategies for autonomous robotic surveillance," in *Proc. IEEE Conf. Decision and Control*, (Seville, Spain), pp. 2200–2205, Dec. 2005.

[35] F. Y. Hadaegh, S.-J. Chung, and H. M. Manohara, "On development of 100-gram-class spacecraft for swarm applications," *IEEE Syst. J.*, 2014. DOI: 10.1109/JSYST.2014.2327972.

[36] S. Bandyopadhyay, G. P. Subramanian, D. Morgan, R. Foust, S.-J. Chung, and F. Y. Hadaegh, "A review of impending small satellite formation flying missions," in *53rd AIAA Aerospace Sciences Meeting*, (Kissimee, FL), Jan. 2015. AIAA 2015-1623.

[37] S. Bandyopadhyay, R. Foust, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "A review of formation flying and constellation missions using nanosatellites," *J. Spacecraft and Rockets*, 2016. to appear.

[38] G. P. Subramanian, R. Foust, S. Chan, Y. Taleb, D. Rogers, J. Kokkatk, S. Bandyopadhyay, D. Morgan, S.-J. Chung, and F. Y. Hadaegh, "Information-driven systems engineering study of a formation flying demonstration mission using six CubeSats," in *53rd AIAA Aerospace Sciences Meeting*, (Kissimee, FL), pp. 1–30, Jan. 2015. AIAA 2015-2043.

[39] Y. Tsuda, M. Yoshikawa, M. Abe, H. Minamino, and S. Nakazawa, "System design of the Hayabusa–2 asteroid sample return mission to 1999JU3," *Acta Astronautica*, vol. 91, pp. 356–362, 2013.

[40] A. Harris, M. Barucci, J. Cano, A. Fitzsimmons, M. Fulchignoni, S. Green, D. Hestroffer, V. Lappas, W. Lork, P. Michel, D. Morrison, D. Payson, and F. Schaeffer, "The european union funded NEOSHIELD project: A global approach to near-earth object impact threat mitigation," *Acta Astronautica*, vol. 90, no. 1, pp. 80–84, 2013.

[41] S.-I. Nishida, S. Kawamoto, Y. Okawa, F. Terui, and S. Kitamura, "Space debris removal system using a small satellite," *Acta Astronautica*, vol. 65, no. 1, pp. 95–102, 2009.

[42] C. G. Henshaw, "The darpa phoenix spacecraft servicing program: Overview and plans for risk reduction," in *Proceedings of 12th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, (Montreal, Canada), 2014.

[43] V. Gazi and K. M. Passino, "Stability analysis of social foraging swarms," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.,*, vol. 34, no. 1, pp. 539–557, 2004.

[44] C. Canuto, F. Fagnani, and P. Tilli, "A Eulerian approach to the analysis of rendezvous algorithms," in *Proc. IFAC World Congr.*, 2008.

[45] P. K. Menon, G. D. Sweriduk, and K. D. Bilimoria, "New approach for modeling, analysis, and control of air traffic flow," *J. Guid. Control Dyn.*, vol. 27, no. 5, pp. 737–744, 2004.

[46] P. Kingston and M. Egerstedt, "Index-free multiagent systems: An Eulerian approach," in *Proc. 2nd IFAC Workshop on Distributed Estimation and Control in Networked Systems*, (Annecy, France), pp. 215–220, 2010.

[47] D. J. G. Pearce and M. S. Turner, "Density regulation in strictly metric-free swarms," *New Journal of Physics*, vol. 16, 2014.

[48] R. A. Brooks and T. Lozano-Perez, "A subdivision algorithm in configuration space for findpath with rotation," *IEEE Trans. Syst. Man Cybern.*, vol. 15, no. 2, pp. 224–233, 1985.

[49] R. V. Cowlagi and P. Tsiotras, "Beyond quadtrees: Cell decompositions for path planning using wavelet transforms," in *Proc. IEEE Conf. Decision and Control*, (New Orleans, LA), pp. 1392–1397, Dec. 2007.

[50] P. Agharkar, R. Patel, and F. Bullo, "Robotic surveillance and Markov chains with minimal first passage time," in *Proc. IEEE Conf. Decision and Control*, (Los Angeles, CA), Dec. 2014.

[51] L. Devroye, *Non-Uniform Random Variate Generation*. New York, NY: Springer-Verlag, 1986.

[52] D. P. Bertsekas, *Linear network optimization: algorithms and codes*. MIT Press, 1991.

[53] L. Kantorovitch, "On the translocation of masses," *Management Science*, vol. 5, no. 1, pp. 1–4, 1958.

[54] C. Villani, *Optimal transport: Old and New*. Springer Verlag, 2008.

[55] S. T. Rachev and L. Ruschendorf, *Mass Transportation Problems. Vol. I: Theorey*. Springer, 1998.

[56] Y. Zemel, "Optimal transportation: Continuous and discrete," Master's thesis, École polytechnique fédérale de Lausanne, 2012.

[57] S. Daneri and G. Savaré, "Lecture notes on gradient flows and optimal transport," in *Seminaires et Congres SMF*, pp. 1–27, 2010.

[58] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *J. Guid. Control Dyn.*, vol. 25, no. 4, pp. 755–764, 2002.

[59] J. DeMot and E. Feron, "Optimal two-agent navigation with local environment information," in *Proc. Allerton Conf. Comm. Control Computing*, (Allerton, IL), Oct. 2004.

[60] M. G. Earl and R. D'Andrea, "Iterative MILP methods for vehicle-control problems," *IEEE Trans. Robotics*, vol. 21, no. 6, pp. 1158–1167, 2005.

[61] M. Movafaghpour and E. Masehian, "A linear programming approach for probabilistic robot path planning with missing information of outcomes," in *IEEE Int. Conf. Automation Science and Engineering*, (Trieste, Italy), pp. 126–132, Aug 2011.

[62] L. Yang, J. Qi, and J. Han, "Path planning methods for mobile robots with linear programming," in *Proc. Int. Conf. Modelling, Identification and Control*, (Wuhan, China), June 2012.

[63] G. C. Chasparis and J. S. Shamma, "Linear-programming-based multi-vehicle path planning with adversaries," in *Amer. Control Conf.*, (Portland, OR), June 2005.

[64] G. Notarstefano and F. Bullo, "Network abstract linear programming with application to minimum-time formation control," in *Proc. IEEE Conf. Decision and Control*, (New Orleans, LA), pp. 927–932, Dec. 2007.

[65] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer, "A distributed simplex algorithm for degenerate linear programs and multi-agent assignments," *Automatica*, vol. 48, pp. 2298–2304, 2012.

[66] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas, "A distributed auction algorithm for the assignment problem," in *IEEE Conf. Decision Control*, (Cancun, Mexico), pp. 1212–1217, Dec. 2008.

[67] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[68] D. Milutinović and P. Lima, "Modeling and optimal centralized control of a large-size robotic population," *IEEE Trans. Robotics*, vol. 22, no. 6, pp. 1280–1285, 2006.

[69] G. Foderaro, S. Ferrari, and T. A. Wettergren, "Distributed optimal control for multi-agent trajectory optimization," *Automatica*, vol. 50, no. 1, pp. 149–154, 2014.

[70] J. Qi, R. Vazquez, and M. Krstic, "Multi-agent deployment in 3-D via PDE control," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 891–906, 2015.

[71] M. R. Pac, A. M. Erkmen, and I. Erkmen, "Control of robotic swarm behaviors based on smoothed particle hydrodynamics," in *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, (San Diego, CA), Oct. 2007.

[72] D. A. Dolgov and E. H. Durfee, "The effects of locality and asymmetry in large-scale multiagent MDPs," in *Coordination of Large-Scale Multiagent Systems* (P. Scerri, R. Vincent, and R. Mailler, eds.), pp. 3–25, Springer, 2005.

[73] Y. F. Chen, N. K. Ure, G. Chowdhary, J. P. How, and J. Vian, "Planning for large-scale multiagent problems via hierarchical decomposition with applications to UAV health management," in *Proc. Amer. Control Conf.*, pp. 1279–1285, 2014.

[74] A. Mogilner and L. Edelstein-Keshet, "A non-local model for a swarm," *J. Math. Biology*, vol. 38, no. 6, pp. 534–570, 1999.

[75] C. C. Cheah, S. P. Hou, and J. J. E. Slotine, "Region-based shape control for a swarm of robots," *Automatica*, vol. 45, no. 10, pp. 2406–2411, 2009.

[76] S. Zhao, S. Ramakrishnan, and M. Kumar, "Density-based control of multiple robots," in *Proc. Amer. Control Conf.*, (San Francisco, CA), 2011.

[77] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," *Robotica*, pp. 1–35, March 2014.

[78] D. Morgan, G. P. Subramanian, S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic guidance of distributed systems using sequential convex programming," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, (Chicago, IL), pp. 3850–3857, Sept. 2014.

[79] M. Lindhe, P. Ogren, and K. H. Johansson, "Flocking with obstacle avoidance: A new distributed coordination algorithm based on Voronoi partitions," in *IEEE Int. Conf. Robotics Automation*, (Barcelona, Spain), Apr. 2005.

[80] S. Garrido, L. Moreno, D. Blanco, and P. Jurewicz, "Path planning for mobile robot navigation using Voronoi diagram and fast marching," *Int. J. Robotics Automation*, vol. 2, no. 1, pp. 42–64, 2011.

[81] V. Borkar and P. Varaiya, "Asymptotic agreement in distributed estimation," *IEEE Trans. Autom. Control*, vol. 27, no. 3, pp. 650 – 655, 1982.

[82] L. Chen, P. O. Arambel, and R. K. Mehra, "Estimation under unknown correlation: covariance intersection revisited," *IEEE Trans. Autom. Control*, vol. 47, no. 11, pp. 1879–1882, 2002.

[83] M. Kamgarpour and C. Tomlin, "Convergence properties of a decentralized Kalman filter," in *IEEE Conf. Decision Control*, pp. 3205–3210, IEEE, 2008.

[84] R. Olfati-Saber, "Kalman-consensus filter : Optimality, stability, and performance," in *IEEE Conf. Decision Control*, (Shanghai, China), pp. 7036–7042, December 2009.

[85] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering," *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1410–1415, 2015.

[86] G. Pavlin, P. Oude, M. Maris, J. Nunnink, and T. Hood, "A multi-agent systems approach to distributed Bayesian information fusion," *Inform. Fusion*, vol. 11, pp. 267–282, 2010.

[87] A. Jadbabaie, P. Molavi, A. Sandroni, and A. Tahbaz-Salehi, "Non-Bayesian social learning," *Games and Economic Behavior*, vol. 76, pp. 210–225, 2012.

[88] A. Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *arXiv preprint arXiv:1508.05161*, 2015.

[89] T. Bailey, S. Julier, and G. Agamennoni, "On conservative fusion of information with unknown non-Gaussian dependence," in *Int. Conf. Information Fusion*, (Singapore), pp. 1876–1883, July 2012.

[90] N. Ahmed, J. Schoenberg, and M. Campbell, "Fast weighted exponential product rules for robust general multi-robot data fusion," in *Robotics: Science and Systems VIII* (N. Roy, P. Newman, and S. Srinivasa, eds.), pp. 9–16, MIT Press, 2013.

[91] C. S. R. Fraser, L. F. Bertuccelli, H.-L. Choi, and J. P. How, "A hyperparameter consensus method for agreement under uncertainty," *Automatica*, vol. 48, no. 2, pp. 374 – 380, 2012.

[92] O. Hlinka, O. Slučiak, F. Hlawatsch, P. M. Djuric, and M. Rupp, "Likelihood consensus and its application to distributed particle filtering," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4334–4349, 2012.

[93] O. Hlinka, F. Hlawatsch, and P. M. Djuric, "Consensus-based distributed particle filtering with distributed proposal adaptation," *IEEE Trans. Signal Process.*, vol. 62, no. 12, pp. 3029–3041, 2014.

[94] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability," *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.

[95] S. Bandyopadhyay and S.-J. Chung, "Distributed estimation using Bayesian consensus filtering," in *Proc. Amer. Control Conf.*, (Portland, OR), pp. 634–641, June 2014.

[96] M. H. DeGroot, "Reaching a consensus," *J. Amer. Statistical Assoc.*, vol. 69, no. 345, pp. 688 – 704, 1960.

[97] M. Bacharach, "Normal Bayesian dialogues," *J. Amer. Statistical Assoc.*, vol. 74, no. 368, pp. 837 – 846, 1979.

[98] S. French, "Consensus of opinion," *European J. Operational Research*, vol. 7, pp. 332 – 340, 1981.

[99] C. Genest and J. V. Zidek, "Combining probability distributions: A critique and an annotated bibliography," *Statistical Sci.*, vol. 1, no. 1, pp. 114 – 135, 1986.

[100] G. L. Gilardoni and M. K. Clayton, "On reaching a consensus using DeGroot's iterative pooling," *Ann. Stat.*, vol. 21, no. 1, pp. 391 – 401, 1993.

[101] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803 – 812, 1986.

[102] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520 – 1533, 2004.

[103] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Trans. Autom. Control*, vol. 48, no. 6, pp. 988 – 1001, 2003.

[104] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 33–55, 2009.

[105] A. Olshevsky, "Average consensus in nearly linear time and its implications for decentralized optimization and multi-agent control," *arXiv preprint arXiv:1411.4186*, 2014.

[106] W. Luo, Y.-C. Chu, and K.-V. Ling, "Inverse optimal adaptive control for attitude tracking of spacecraft," *IEEE Trans. Autom. Control*, vol. 50, no. 11, pp. 1639–1654, 2005.

[107] J. Ahmed, V. T. Coppola, and D. Bernstein, "Adaptive asymptotic tracking of spacecraft attitude motion with inertia matrix identification," *J. Guid. Control Dyn.*, vol. 21, no. 5, pp. 684–691, 1998.

[108] J. L. Junkins, M. R. Akella, and R. D. Robinett, "Nonlinear adaptive control of spacecraft maneuvers," *J. Guid. Control Dyn.*, vol. 20, no. 6, pp. 1104–1110, 1997.

[109] B.-S. Chen, C.-S. Wu, and Y.-W. Jan, "Adaptive fuzzy mixed H2/H∞ attitude control of spacecraft," *IEEE Trans. Aerospace Electron. Syst.*, vol. 36, no. 4, pp. 1343–1359, 2000.

[110] J. L. Crassidis and F. L. Markley, "Sliding mode control using modified Rodrigues parameters," *J. Guid. Control Dyn.*, vol. 19, no. 6, pp. 1381–1383, 1996.

[111] W. Kang, "Nonlinear H∞ control and its application to rigid spacecraft," *IEEE Trans. Autom. Control*, vol. 40, no. 7, pp. 1281–1285, 1995.

[112] F. L. Markley and J. L. Crassidis, *Fundamentals of Spacecraft Attitude Determination and Control*. New York, NY: Springer, 2014.

[113] P. Tsiotras, "Stabilization and optimality results for the attitude control problem," *J. Guid. Control Dyn.*, vol. 19, no. 4, pp. 772–779, 1996.

[114] Y. D. Song and W. Cai, "Quaternion observer-based model-independent attitude tracking control of spacecraft," *J. Guid. Control Dyn.*, vol. 32, no. 5, pp. 1476–1482, 2009.

[115] S.-J. Chung, U. Ahsun, and J.-J. E. Slotine, "Application of synchronization to formation flying spacecraft: Lagrangian approach," *J. Guid. Control Dyn.*, vol. 32, pp. 512–526, Mar.-Apr. 2009.

[116] A. Weiss, I. Kolmanovsky, D. S. Bernstein, and A. Sanyal, "Inertia-free spacecraft attitude control using reaction wheels," *J. Guid. Control Dyn.*, vol. 36, no. 5, pp. 1425–1439, 2013.

[117] J.-J. E. Slotine and W. Li, *Applied Nonlinear Control*, vol. 199. NJ: Prentice-Hall Englewood Cliffs, 1991.

[118] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Inhomogeneous Markov chain approach to probabilistic swarm guidance algorithm," in *Proc. 5th Int. Conf. Spacecraft Formation Flying Missions and Technologies*, (Munich, Germany), 2013.

[119] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using optimal transport," in *Proc. IEEE Conf. Control Applicat.*, (Antibes, France), pp. 498–505, Oct. 2014.

[120] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Feedback-based inhomogeneous Markov chain approach to probabilistic swarm guidance," in *8th International Workshop on Satellite Constellations and Formation Flying*, (Delft, Netherlands), June 2015.

[121] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "A probabilistic Eulerian approach for path planning of a large-scale swarm of agents," *IEEE Trans. Robotics*, 2015. under review.

[122] P. R. Halmos, *Naive set theory*. N.J.: Princeton University Press, 1960. p. 28.

[123] A. Olshevsky, *Efficient Information Aggregation Strategies for Distributed Control and Signal Processing*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, Sept. 2010. p. 99.

[124] G. Habibi and J. McLurkin, "Maximum-leaf spanning trees for efficient multi-robot recovery with connectivity guarantees," in *Distributed Autonomous Robotic Systems*, pp. 275–289, Springer, 2014.

[125] M. Jackson, *Social and economic networks*. Princeton, NJ: Princeton University Press, 2008.

[126] J. Yu, S.-J. Chung, and P. G. Voulgaris, "Target assignment in robotic networks: Distance optimality guarantees and hierarchical strategies," *IEEE Trans. Autom. Control*, vol. 60, pp. 327–341, Feb. 2015.

[127] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," *Wireless Networks*, vol. 10, pp. 169–181, Mar. 2004.

[128] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, pp. 65 – 78, 2004.

[129] B. Gharesifard and J. Cortés, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," *European J. Control*, vol. 18, no. 6, pp. 539–557, 2012.

[130] A. Rikos, T. Charalambous, and C. N. Hadjicostis, "Distributed weight balancing over digraphs," *IEEE Trans. Control Network Syst.*, vol. 1, no. 2, pp. 190–201, 2014.

[131] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Probabilistic swarm guidance using inhomogeneous Markov chains," http://arxiv.org/abs/1403.4134).

[132] E. Torgerson, *Comparison of Statistical Experiments*. Cambridge University Press, 1991.

[133] S. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models and Methods in Appl. Sci.*, vol. 1, no. 4, pp. 300–307, 2007.

[134] D. Pollard, "Asymptopia," 2000. Manuscript in progress, available at http://www.stat.yale.edu/~pollard.

[135] E. Seneta, *Non-negative Matrices and Markov Chains*. New York, NY: Springer-Verlag, 2006.

[136] S. Chib and E. Greenberg, "Understanding the Metropolis–Hastings algorithm," *American Statistician*, vol. 49, no. 4, pp. 327–335, 1995.

[137] L. J. Billera and P. Diaconis, "A geometric interpretation of the Metropolis–Hastings algorithm," *Statistical Science*, pp. 335–339, 2001.

[138] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing Markov chain on a graph," *SIAM review*, vol. 46, no. 4, pp. 667–689, 2004.

[139] O. Cihan and M. Akar, "Fastest mixing reversible Markov chains on graphs with degree proportional stationary distributions," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 227–232, 2015.

[140] D. B. West, *Introduction to graph theory*, vol. 2. Prentice Hall, 2001.

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

[142] B. Touri and A. Nedić, "On ergodicity, infinite flow, and consensus in random models," *IEEE Trans. Autom. Control*, vol. 56, no. 7, pp. 1593–1605, 2011.

[143] R. Durrett, *Probability: Theory and Examples*. Thomson Brooks, 2005.

[144] P. Billingsley, *Probability and Measure*. New York: J. Wiley & Sons, 1995.

[145] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE J. Robotics and Automation*, vol. 2, no. 3, pp. 135–145, 1986.

[146] R. V. Cowlagi and P. Tsiotras, "Hierarchical motion planning with dynamical feasibility guarantees for mobile robotic vehicles," *IEEE Trans. Robotics*, vol. 28, no. 2, pp. 379–395, 2012.

[147] Q. Du, V. Faber, and M. Gunzburger, "Centroidal Voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, pp. 637–676, Dec. 1999.

[148] M. deBerg, O. Cheong, M. vanKreveld, and M. Overmars, *Computational geometry: algorithms and applications*. Berlin: Springer, 2008. p. 151.

[149] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2009.

[150] D. Morgan, G. P. Subramanian, S.-J. Chung, and F. Y. Hadaegh, "Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming," *Int. J. Robotics Research*, 2016. to appear.

[151] G. P. Subramanian, "Nonlinear control strategies for quadrotors and CubeSats," Master's thesis, University of Illinois at Urbana-Champaign, 2015.

[152] S. Bandyopadhyay and S.-J. Chung, "Distributed bayesian filtering algorithm for dynamic sensor networks," *Automatica*, 2016. under review.

[153] Z. Chen, "Bayesian filtering: From Kalman filters to particle filters, and beyond," *Statistics*, vol. 182, no. 1, pp. 1–69, 2003.

[154] M. H. DeGroot, *Probability and Statistics*. Cambridge, Massachusetts: Addison-Wesley, 1975.

[155] H. Jeffreys, *Theory of Probability*. Oxford: Clarendon Press, 1961.

[156] K. Subrahmaniam, *A Primer in Probability*. New York, NY: M. Dekker, 1979.

[157] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[158] H. Durrant-Whyte and T. C. Henderson, *Springer Handbook of Robotics*, ch. Multisensor Data Fusion, pp. 585–610. Springer, 2008.

[159] N. Gordon, B. Ristic, and S. Arulampalam, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, London, 2004.

[160] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. Signal Process.*, vol. 50, pp. 174–188, February 2002.

[161] O. Lebeltel, P. Bessiere, J. Diard, and E. Mazer, "Bayesian robot programming," *Autonomous Robots*, vol. 16, pp. 49–79, January 2004.

[162] M.-H. Chen, "Bayesian computation: From posterior densities to Bayes factors, marginal likelihoods, and posterior model probabilities," in *Bayesian Thinking, Modeling and Computation* (D. K. Dey and C. R. Rao, eds.), Handbook of Statistics, ch. 15, pp. 437 – 457, Amsterdam: Elsevier, 2005.

[163] M. Zhu and S. Martínez, "Discrete-time dynamic average consensus," *Automatica*, vol. 46, no. 2, pp. 322–329, 2010.

[164] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Soc., 2009.

[165] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: Wiley, 1991.

[166] A. Stuart and A. R. Humphries, *Dynamical Systems and Numerical Analysis*. Cambridge, England: Cambridge University Press, 1998.

[167] G. Kramer and S. A. Savari, "Communicating probability distributions," *IEEE Trans. Inf. Theory*, vol. 53, pp. 518–525, February 2007.

[168] J. H. Kotecha and P. M. Djuric, "Gaussian sum particle filtering," *IEEE Trans. Signal Process.*, vol. 51, pp. 2602–2612, Oct. 2003.

[169] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Mineola, New York: Dover Publications, 2005.

[170] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications to Clustering*. New York, N.Y.: M. Dekker, 1988.

[171] D. A. Reynolds, "Gaussian mixture models," *Encyclopedia of Biometric Recognition*, February 2008.

[172] A. G. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC press, 1998.

[173] H. Fourati, *Multisensor Data Fusion: From Algorithms and Architectural Design to Applications*. Series: Devices, Circuits, and Systems, CRC Press, Taylor & Francis Group LLC, 2015.

[174] K. A. Ross, *Elementary Analysis: The Theory of Calculus*. Springer, 1980.

[175] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 60, pp. 6217–6234, Dec. 2012.

[176] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software*. John Wiley & Sons, 2004.

[177] D. A. A. Marín, "Particle filter tutorial." (http://www.mathworks.com/matlabcentral/fileexchange/35468-particle-filter-tutorial) [retrieved on January 21, 2016].

[178] M. E. P. Chatters and M. B. J. Crothers, *AU-18 Space Primer*, ch. Space Surveillance Network, pp. 249–258. Air University Press, Maxwell Air Force Base, Alabama, 2009.

[179] D. A. Vallado and J. D. Griesbach, "Simulating space surveillance networks," in *AAS/AIAA Astrodynamics Specialist Conf.*, (Girdwood), 2012. Paper AAS 11-580.

[180] F. R. Hoots and R. L. Roehrich, "Spacetrack report number 3: Models for propagation of NORAD element sets," tech. rep., U.S. Air Force Aerospace Defense Command, Colorado Springs, CO., 1980.

[181] D. Vallado and P. Crawford, "SGP4 orbit determination," in *AIAA/AAS Astrodynamics Specialist Conf.*, 2008.

[182] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Attitude stabilization of a captured asteroid for the asteroid redirect mission," in *AIAA Guidance, Navigation, and Control*, (Kissimee, FL), Jan. 2015. AIAA 2015-0596.

[183] S. Bandyopadhyay and S.-J. Chung, "Nonlinear attitude control of spacecraft with a captured asteroid," in *8th International Workshop on Satellite Constellations and Formation Flying*, (Delft, Netherlands), June 2015.

[184] S. Bandyopadhyay, S.-J. Chung, and F. Y. Hadaegh, "Nonlinear attitude control of spacecraft with a large captured object," *J. Guid. Control Dyn.*, 2016. to appear (DOI: 10.2514/1.G001341).

[185] J. R. Brophy, R. Gershman, D. Landau, J. Polk, C. Porter, D. Yeomans, C. Allen, W. Williams, and E. Asphaug, "Asteroid return mission feasibility study," in *AIAA/ASME/SAE/ASEE Joint Propulsion Conf.*, (San Diego, CA), July–August 2011.

[186] J. R. Brophy, L. Friedman, and F. Culick, "Asteroid retrieval feasibility," in *IEEE Aerospace Conference*, (Big Sky, MT), pp. 1–16, March 2012.

[187] J. R. Brophy and S. Oleson, "Spacecraft conceptual design for returning entire near-earth asteroids," in *AIAA/ASME/SAE/ASEE Joint Propulsion Conf.*, (Atlanta, GA), July–August 2012.

[188] D. Landau, J. Dankanich, N. Strange, J. Bellerose, P. Llanos, and M. Tantardini, "Trajectories to nab a NEA (near-Earth asteroid)," in *AAS/AIAA Spaceflight Mechanics Meeting*, (Kauai, HI), Feb. 2013.

[189] N. Strange, D. Landau, T. McElrath, G. Lantoine, and T. Lam, "Overview of mission design for NASA asteroid redirect robotic mission concept," in *Int. Electric Propulsion Conf.*, (Washington, D.C.), Oct. 2013.

[190] H. Schaub and J. L. Junkins, "Stereographic orientation parameters for attitude dynamics: A generalization of the Rodrigues parameters," *Journal of the Astronautical Sciences*, vol. 44, pp. 1–19, Jan.–Mar. 1996.

[191] M. J. Sidi, *Spacecraft Dynamics and Control*. Cambridge, U.K.: Cambridge Univ. Press, 1997.

[192] B. Wie, *Space Vehicle Dynamics and Control*. Reston, VA, US: American Institute of Aeronautics and Astronautics, 1998.

[193] M. D. Shuster, "A survey of attitude representations," *Navigation*, vol. 8, no. 9, pp. 439–517, 1993.

[194] H. K. Khalil, *Nonlinear Systems*. New York: Macmillan Pub. Co., 1992.

[195] W. Wang and J.-J. E. Slotine, "On partial contraction analysis for coupled nonlinear oscillators," *Biological cybernetics*, vol. 92, no. 1, pp. 38–53, 2005.

[196] S. P. Bhat and D. S. Bernstein, "A topological obstruction to continuous global stabilization of rotational motion and the unwinding phenomenon," *Systems & Control Letters*, vol. 39, no. 1, pp. 63–70, 2000.

[197] T. Lee, "Exponential stability of an attitude tracking control system on SO(3) for large-angle rotational maneuvers," *Syst. Control Lett.*, vol. 61, no. 1, pp. 231–237, 2012.

[198] I. M. Ross, "Space trajectory optimization and l1-optimal control problems," *Elsevier Astrodynamics Series*, vol. 1, pp. 155–VIII, 2007.

[199] M. Krstic and P. Tsiotras, "Inverse optimal stabilization of a rigid spacecraft," *IEEE Trans. Autom. Control*, vol. 44, no. 5, pp. 1042–1049, 1999.

[200] R. Sharma and A. Tewari, "Optimal nonlinear tracking of spacecraft attitude maneuvers," *IEEE Trans. Control Systems Technology*, vol. 12, no. 5, pp. 677–682, 2004.

[201] H. Schaub, J. L. Junkins, and R. D. Robinett, "New penalty functions and optimal control formulation for spacecraft attitude control problems," *J. Guid. Control Dyn.*, vol. 20, no. 3, pp. 428–434, 1997.

[202] D. Garg, M. Patterson, W. W. Hager, A. V. Rao, D. A. Benson, and G. T. Huntington, "A unified framework for the numerical solution of optimal control problems using pseudospectral methods," *Automatica*, vol. 46, no. 11, pp. 1843–1851, 2010.

[203] M. A. Patterson and A. V. Rao, "GPOPS-II: A MATLAB software for solving multiple-phase optimal control problems using hp-adaptive gaussian quadrature collocation methods and sparse nonlinear programming," *ACM Transactions on Mathematical Software*, vol. 41, pp. 1–37, Oct. 2014.

[204] W. Lohmiller and J. E. Slotine, "On contraction analysis for nonlinear systems," *Automatica*, vol. 34, no. 6, pp. 683 – 696, 1998.