LOCAL ENCOUNTERS IN ROBOT SWARMS: FROM LOCALIZATION TO DENSITY REGULATION

A Dissertation
Presented to
The Academic Faculty

By

Siddharth Mayya

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

December 2019

LOCAL ENCOUNTERS IN ROBOT SWARMS: FROM LOCALIZATION TO DENSITY REGULATION

Approved by:

Dr. Magnus Egerstedt, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Seth Hutchinson
School of Interactive Computing
Georgia Institute of Technology

Dr. Yorai Wardi
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Daniel Goldman
School of Physics
Georgia Institute of Technology

Dr. Dylan Shell
Department of Computer Science & Engineering
Texas A&M University

Date Approved: October 22, 2019

कर्मण्येवाधिकारस्ते मा फलेषु कदाचन।
मा कर्मफलहेतुर्भूर्मा ते सङ्गोऽस्त्वकर्मणि ।।

योगस्थः कुरु कर्माणि सङ्गं त्यक्त्वा धनञ्जय।
सिद्ध्यसिद्ध्योः समो भूत्वा समत्वं योग उच्यते ।।

karmanyevadhikaraste ma phalesu kadacana
ma karmaphalahetur bhur ma te sango 'stv akarmani

yogasthah kuru karmani sangam tyaktva dhananjaya
siddhyasiddhyoh samo bhutva samatvam yoga ucyate

Your right is to action alone, never to its fruits at any time. The fruits of your action should not be your motive, nor should your attachment be to inaction.

Perform your activities equipoised, abandoning all attachment to success or failure. Such equanimity is called Yoga.

— **Bhagavad Gita**

To mother and father

ACKNOWLEDGMENTS

This dissertation would not have been possible without various contributions from the people I was fortunate enough to interact with during my time at Georgia Tech.

I am still astonished at the extent to which a research advisor and mentor can profoundly influence one's life. Through these few sentences, I will try to express the gratitude I feel towards my advisor Dr. Magnus Egerstedt. Over the years, his constant guidance, encouragement, and support forged my research career and shaped the way I approach new research questions. I thank him for leading me down a research path that I was truly excited about, thus making my grad school journey an enjoyable and rewarding experience. Outside of our research discussions, his professional conduct served as guidance—and will continue to serve as guidance—on navigating the academic world. "What would Dr. Egerstedt do in this situation?" was a question my labmates and I often found ourselves asking.

I learnt a great deal about robotics, control, biology, and research in general via my interactions with various professors at Georgia Tech. Over discussions which sometimes spanned a whole afternoon, Professor Seth Hutchinson taught me to always ask the important question "What would this algorithm or idea be useful for?", shaping my engineering mindset and teaching me what real robotics entailed. I thank Professor Dan Goldman for introducing me to the exciting world of active matter physics, which changed the course of my research agenda and will act as a central pillar of my research going forward. I owe a lot of my fundamental knowledge in non-linear and optimal control to my interactions and classes with Professor Yorai Wardi. Water cooler interactions with Professors Matthieu Bloch, Eric Feron, and Alenka Zajic always led to new and interesting ideas, which is part of what made TSRB such a stimulating working environment. Outside of Georgia Tech, I had the pleasure of collaborating with Dr. Stephen Pratt, who graciously hosted me for a summer visit at Arizona State University. Conducting experiments on real ants after

reading so much about them was an exciting experience, and I took a lot of new ideas and perspectives from this visit. Over the past year, my collaboration with Dr. Dylan Shell from Texas A&M University has been incredible fun (not to mention incredibly educating), and this has led to some of the work I'm most excited about in my thesis. I look forward to continuing both these collaborations in the future.

It takes a special kind of work environment to make you *want* to come to the lab and be productive. The GRITS Lab was such a place, and what made it special were the people. Firstly, a big shout out to my nearest neighbors María Santos and Gennaro Notomista for demonstrating to me that working and having fun can often be the same thing. The late work nights barely making conference deadlines, numerous trips to Gokul Sweets recovering from the deadlines, navigating our first few conferences together, and lunch/coffee breaks in the lab will be forever imprinted in my memory. Expanding outwards (from my desk), my labmates Yousef Emam, Anqi Li, Chris Banks, Paul Glotfelter, Ian Buckley, Mark Mote, Pietro Pierpaoli, Sean Wilson as well as past lab members Jeremy Cai, Li Wang, Sebastian Ruf, Kyle Slovak, Matt Hale, Tina Setter, Daniel Pickem, Usman Ali, Keith Paarporn helped enrich my PhD experience with their constant support, bright ideas, and smiling faces. I would also like to thank all the senior lab members as well as visiting students and scholars who ensured that good ideas and fresh perspectives kept circulating through the lab.

Grad school evenings would not have been as much fun without Arjun Chakraborty picking me up from TSRB on his way to the pub (or Chipotle). Or without the *chai pe charcha* with Skanda Prasad on the porch at home. Star-gazing trips, game nights, reluctant 10k runs, and gastronomic adventures were an absolute pleasure with the Atlanta posse, who are too numerous to be named here. I thank all of them for pulling me away from work occasionally and teaching me what work-life balance meant. Bhagyashri Telsang and Dheeraj Velicheti have been my inseparable friends for almost a decade and I can say without a doubt that my PhD journey would have been a very different (read: worse)

experience without our regular trips around the American southeast. I'm also very grateful for all the words of encouragement (sufficiently mixed in with mildly hurtful puns) that my MIT and colony friends always had for me. I would be remiss if I didn't mention how Rudra and Mrs. Dog's friendly wags made weekends and evenings a happy time.

Obtaining a doctoral degree while being 9000 miles from home is non-trivial. It took a special kind of family—one that never left my side and injected optimism in my life—to make it happen. My parents supported me through thick and thin, often lending a friendly ear when I provided unsolicited research banter. Jayathe, Jean Luc, and Sanjana made me feel at home away from home: hosting me during my summer internship at Tesla and including me in their vacation plans. I thank the rest of the Mayya family, Pavi and Ram, as well as Mama and Thatha for all their support over the years.

I'm glad to say that I absolutely loved my PhD experience—it was stimulating, educative, and perhaps most importantly, it was *fun*. As I head north to begin the next chapter of my life, I will sorely miss coming to TSRB and interacting with the amazing people at Georgia Tech.

TABLE OF CONTENTS

Acknowledgments v

List of Figures xii

Summary xxii

Chapter 1: Introduction 1

Chapter 2: Background 7

2.1 Encounters in Robotics 7

2.1.1 Spatial Interference 7

2.1.2 Stochastic Swarm Models 9

2.1.3 Spatially Organized Swarm Behaviors 10

2.1.4 Collisional Interactions 11

2.2 Encounters in Nature 11

2.2.1 Living Collectives 12

2.2.2 Non-living Collectives 12

2.3 Conclusions 13

Chapter 3: Localization using Inter-robot Collisions 15

3.1 Stochastic Geometry 17

3.2 Collision Model 19
3.2.1 Deployment at time $k = 0$ 20
3.2.2 Movements at times $k > 0$ 22
3.3 Collisions through Hidden Markov Models 27
3.3.1 The Forward Algorithm 27
3.3.2 Simulation Results 32
3.3.3 Connections with Continuous-time Motion of Robots 35
3.4 Continuous-time Extension 37
3.5 Experimental Results 41
3.5.1 Evaluation Methodology 41
3.5.2 Robot Experiments 43
3.6 Conclusions 46

Chapter 4: Encounter Rates as Information Sources 47
4.1 Motion Model 47
4.2 Modeling the Encounter Rate 48
4.3 Motivating Application: Interference Reduction Using Encounters 52
4.3.1 Problem Setup 54
4.3.2 Continuous-time Markov Chain 57
4.3.3 Decentralized Density Control 61
4.3.4 Experimental Results 64
4.4 Conclusion 65

Chapter 5: Encounters in High-density Robot Swarms 67

5.1 Enksog Model 67
5.1.1 Simulation 71
5.2 Task Allocation Using Encounters 71
5.2.1 Transition Rules 73
5.2.2 Task Allocation Dynamics 74
5.2.3 Equilibrium Analysis 77
5.2.4 Simulation Results 79
5.2.5 Adaptive Transition Rates 82
5.2.6 Experimental Results 91
5.3 Conclusions 93

Chapter 6: Leveraging the Physics of Inter-robot Encounters 99

6.1 Motion and Collision Model 101
6.1.1 Motion Model 101
6.1.2 Inter-Robot Interaction Models 103
6.1.3 Simulation Setup 105
6.2 Motility-Induced Phase Separation 106
6.2.1 Theoretical Analysis 107
6.2.2 Simulations 111
6.3 Deployment and Experiments 113
6.4 Conclusions 115

Chapter 7: Developing Swarm Robotics Platforms 118

7.1 Robotarium 118

7.1.1 The GRITSBot X 119

7.1.2 Safely Handling Intermittent Communication Failures 121

7.2 Brushbots 140

7.2.1 Modeling of Vibration-based Locomotion 141

7.2.2 Brushbots in Swarm Robotics 143

7.3 Conclusions 146

Chapter 8: Conclusions and Future Directions 147

References 149

LIST OF FIGURES

3.1 A mathematical model of the spatial distribution of robots within a cell $D_j \subset D$, which is depicted here as the shaded area. Φ_j is a hard-core point process consisting of infinite points in the space $\mathbb{R}^2$, illustrated by the dots. As described in Definition 4, the location of the robots at time $k = 0$, denoted here by $\mathbb{X}_j(0) = \{x_1, x_2, \ldots, x_9\}$, are points of the hard-core point process Φ_j that happen to lie inside the set D_j, i.e., $\mathbb{X}_j(0) = \Phi_j \cap D_j$. The robots are depicted by circles of radius r. The inhibition distance of the hard-core point process is $2r$. 21

3.2 At time k, the locations of the robots in cell D_j is illustrated by the set $\mathbb{X}_j(k) = \{x_1, \ldots, x_4\}$. The probability of collision experienced by an exogenous robot (whose position is denoted by x_e), dropped at a uniformly random location in cell D_j, is equal to the probability that the random variable denoting the distance between x_e and its first nearest neighbor in the set $\mathbb{X}_j(k)$ is less than $2r$. Furthermore, as specified in Assumption 3, this is also equal to the probability of collision experienced by any robot in $\mathbb{X}_j(k)$. 24

3.3 Comparison of the collision probabilities predicted by the hard-core point process model against the collision rates obtained using a Monte Carlo simulation. For each hard-core process intensity, an exogenous robot is dropped 5000 times into the cell, and the number of collisions is recorded to obtain the Monte Carlo estimate of the probability. In addition, a 95% confidence interval is computed (illustrated by the vertical bars at each data point) for the estimated probabilities according to the Clopper-Pearson confidence interval method. As seen, the theoretical probabilities match the simulation data for varying intensities, and serve as good estimates of the true collision probabilities under the developed collision model. 27

3.4 A circular track with 10 cells of varying widths. In each cell, at time $k = 0$, robots are deployed according to a hard-core point process of intensity $\bar{\lambda}_j$. For times $k > 0$, each robot moves from one cell to another according to a Markovian motion model. 34

3.5 Simulation results of the probabilistic localization technique for a team of robots moving in a 10-cell domain. Figure 3.5a shows the true cell and localization estimates for a randomly chosen robot in the domain. As seen, within a few iterations, the estimates closely track the true cell of the robot. Figure 3.5b illustrates the distribution of the quantized localization error for all the robots, collected over multiple simulation runs. Each simulation was initialized with the same hard-core process intensities $\bar{\lambda}$, and the same Markov transition probabilities P. The localization error is the difference in the cell index between the true cell of the robot $q(k)$ and its PMAP estimate $q^*(k)$. As seen, the estimator perfectly localizes the robot 47% of the time, and the estimate is within one cell of true cell 77% of the time. Note that a localization error of 5 and -5 correspond to the same cell (see Fig. 3.4). . . 35

3.6 Comparison of the empirically obtained pair-correlation function for a given team of robots executing a random walk motion model ($\lambda = 30, r = 0.04, \tau = 0.1$) against the theoretical pair-correlation function for a hard-core point process of the same intensity. The empirical pair correlation was evaluated by noting the positions of the robots at regular intervals of time $t_k, k = 1, 2, \ldots$ and computing the average pair correlation between the points. The similarity in the two curves demonstrates how the random walk motion model yields a spatial distribution of robots which is well-represented by a hard-core point process. 36

3.7 Illustration of the rationale behind the computation of continuous-time collision probabilities, inspired from the kinetic theory of gases. The number of collisions experienced by robot i in a time interval τ, is equal to the number of robots which happen to lie inside the area A_c swept by the robot, as illustrated. To account for the fact that all the other robots are also moving, we assume that robot i is moving with the relative mean speed between the robots, denoted as v_{mean}. This allows us to compute the probability that robot i will collide with another robot in a time interval τ. 39

3.8 Experimental Setup: 18 differential-drive mobile robots are traversing a 4-cell domain on the Robotarium—a remotely accessible multi-robot testbed [46]. The cells of the domain, shown as shaded polygonal areas, are interconnected, allowing the robots to move between them. Each robot executes the random walk algorithm outlined in Section 3.4 and by detecting and utilizing inter-robot collisions, generates estimates of the current cell that it is occupying. We qualitatively demonstrate that collisions among the robots are useful sources of information which can be used for localization. 42

3.9 Performance of the collision filter $\mathcal{C}$ compared to the Markov filter $\mathcal{M}$ in a real-robot experiment. As seen, the tracking performance of the dotted line, denoting the estimate, is significantly better in Fig. 3.9a indicating that collisions are helping the robots localize better in the domain. 44

3.10 Quantitative analysis of the performance of the collision-based localization filter for a real-robot experiment consisting of 18 robots moving in a 4-cell domain for 250 seconds. Figures 3.10a and 3.10b illustrate the distribution of the quantized localization errors of the team for the entire duration of the experiment and the last 30% of the experiment respectively. As seen, the collision filter $\mathcal{C}$ correctly predicts the true cell of the robot more often than the Markov filter, and as shown by Fig. 3.10b, this performance improves as more collision information is incorporated into the estimate. Fig. 3.10c illustrates the average localization error $L(t_k)$, detailed in Definition 11. As the experiment progresses, the collision filter $\mathcal{C}$ performs significantly better than the Markov filter $\mathcal{M}$, illustrating that inter-robot collisions can indeed help robots localize in a densely-packed swarm of robots. 45

4.1 The sensory encounter cross-section area swept by robot i in a time window T. Any robot whose center falls within this region will experience an encounter with robot i. Such robots are marked by the dashed circular outlines. Consequently, the expected number of encounters experienced by robot i in the time window T will be equal to the expected number of robots which happen to fall in this green shaded region, whose area is given by $A_e(T)$. To account for the fact that all robots are moving, the average speed v of robot i is replaced with the mean relative speed v_r. 50

4.2 Distributed Collection Task: A team of robots pick up objects from randomly scattered pick-up locations (denoted by $\square$) and deliver them to drop-off locations (denoted by $\times$). An object can be delivered to any drop-off location. Pick-up and drop-off locations are uniformly and independently distributed with densities λ_p and λ_d, respectively. New robots are stationed outside the domain and enter the domain at a constant rate λ_{in}, symbolized by the arrows. 54

4.3 Histogram of the inter-encounter time τ_c and the encounter resolution time τ_r for a team of simulated robots performing the distributed collection task. In Fig. 4.3a, the inter-encounter time distribution is well-approximated by an exponential random variable with mean given in (4.2). Figure 4.3b shows the best-fit exponential distribution to the encounter resolution times. Best-fit exponential parameter: $\rho = 2.44$. 57

4.4 Each robot can be in four different states when performing the *collection* task: free (f), free and encountering another robot (fe), loaded with an object (l), loaded and encountering a robot (le). State transitions are characterized by a continuous-time Markov chain whose transition rates are illustrated. 58

4.5 The theoretical evolution of $T(\lambda)\lambda$ as the density of robots (measured in robots/m^2) increases is closely matched with the values obtained from simulations with robots employing barrier certificates for collision avoidance. As seen, beyond a certain density, the total output of the swarm increases sub-linearly with the robot density since individual robots spend more time avoiding collisions than transporting objects. Here, we are primarily concerned with the performance of the swarm at intermediate densities—the catastrophic jamming occurring at higher densities is not discussed. 60

4.6 Experimental verification of the Voluntary Retreat Algorithm on a team of 12 real robots operating on the Robotarium. An overhead projector is used to project additional information onto the robot arena. The experiment begins with 8 robots inside the elliptical domain seen in Fig. 4.6a. Robots stationed outside enter at a steady rate λ_{in}. Blue and red circles projected around each robot signify *free (f)* and *loaded (l)* robots, respectively. Green circles indicate that a robot has decided to retreat from the domain. These robots drive out of the elliptical domain via any section of the boundary. The robot population data in Fig. 4.6b is averaged over 5 experimental runs to demonstrate the consistent performance of the algorithm, with one standard deviation depicted by the shaded region. For a different set of initial conditions and simulation parameters, Fig. 4.6c highlights the ability of the Voluntary Retreat Algorithm to maintain the population of robots at the optimal value. 66

5.1 Comparison of empirically obtained estimates of mean inter-encounter times in a robot swarm of varying density against the predictions of two different theoretical models—with and without the Enskog correction factor $\mathcal{X}(\lambda)$ as given by (5.5) and (4.4), respectively. Each robot performs a constant-velocity uniformly ergodic random walk in the domain, while encountering other robots. For the simulation parameters ($v = 0.1, \delta = 0.012, r = 0.018$), the empirical inter-encounter times are obtained by averaging the encounter rates over all the robots in the swarm. As seen, the Enskog correction factor leads to a more accurate prediction of the true encounter rates experienced by the robots. 72

5.2 Performance of the developed task allocation algorithm in a swarm of simulated robots executing three different tasks. For a desired allocation of robots performing active tasks $\rho^* = [0.3, 0.1, 0.4]^T$ and parameters $\bar{\lambda} = 50$ robots/m^2, $\delta = 0.02, r = 1.5\delta, v = 0.1$, the solid lines depict the expectation of the allocation percentages as predicted by Theorem 4, and the three different dashed lines depict the percentage of robots allocated to each task as the simulation progresses. As seen, the swarm achieves the desired allocation, with an expectation predicted by the theoretical curve. The illustrated graphs correspond to a single simulation run. 81

5.3 Robustness of the developed encounter-based task allocation policy to individual robot failures. At time $t = 120s$, 40% of the robots in task T_3 experience failure, and are removed from the simulation. As seen in both Fig. 5.3a and Fig. 5.3b, this disrupts the task allocation of the swarm. The swarm then autonomously regulates the allocation ratios back to the desired levels demonstrating that this strategy is robust to changes in the number of robots. While Fig. 5.3a illustrates the performance of the swarm over one simulation run, Fig. 5.3b averages the allocation over 20 simulations to highlight the effect of the swarm failure at time $t = 120s$. 82

5.4 Standard deviations of the swarm task allocation in the presence of individual robot failures. The error bars in Fig. 5.4a-5.4c correspond to one standard deviation (on each side) of the task allocation levels collected over 20 simulation runs for tasks T_1, T_2, and T_3, respectively. These error bars are overlayed on the averaged task allocation values which were presented in Fig. 5.3b. 83

5.5 The logistic function mapping $\mathcal{F}$ described in (5.31), for a varying set of k, m parameter values and $L = 2$. This mapping is used to specify the response of the controller which modifies the task transition probabilities of the robots based on the difference between the desired and the true allocation ratios. The parameter L modifies the maximum value achieved by the function, m signifies the x-axis point at which the mid-point value of the function is attained, and k modifies the maximum slope of the curve. This gives a high degree of flexibility in designing the response of the controller which can be adjusted based on the size of the swarm, task requirements, noise levels in the allocation estimates etc. 85

5.6 Effect of the closed-loop control of task-transition rates in the developed task allocation algorithm for a swarm of simulated robots with density $\bar{\lambda} = 50$ robots/m^2 in a domain $\mathcal{D}$ with robot parameters ($\delta = 0.02, r = 1.5\delta, v = 0.1$). Figure 5.6a illustrates the true allocation ratios converging to the desired values. Figure 5.6b illustrates how the number of task-transitions per robot per unit time reduces as the true allocation approaches the desired value, indicating that the robots are able to measure the current allocation of the swarm and regulate the transition rates accordingly. 94

5.7 Performance of the closed-loop task allocation algorithm for a swarm of simulated robots in the presence of robot failures. At time $t = 200$s, 40% of the robots allocated to task T_3 experience a failure and are removed from the simulation. As seen in Fig. 5.7a, the swarm regulates the task allocation ratios back to the desired levels. In Fig. 5.7b, the number of transitions reduce as the swarm approaches the desired allocation ratio, but after the failure, the robots detect an increase in the allocation error, leading to a slight increase in the number of transitions after time $t = 200$s. This illustrates the closed-loop nature of the developed task allocation strategy. These results have been averaged over 5 simulation runs to better highlight the effect of the failure. 95

5.8 Performance of the particle filter algorithm in the event of robot failures. The solid lines represent the averaged particle filter estimates of all the robots in the swarm which did not experience a failure. The different dashed lines depict the true allocation levels in the swarm corresponding to each of the three tasks. When 40% of the robots allocated to task T_3 experience a failure, the encounter-based particle filter tracks the changing task allocation levels, thus enabling a closed-loop control of the transition rates. 96

5.9 A swarm of 12 robots are allocated among two tasks on the Robotarium [46], a remotely-accessible swarm robotics testbed. An overhead projector is used to overlay colored circles around each robot depicting the current task being performed by the robot. Green and blue circles represent a robot performing task T_1 and T_2, respectively. Red robots are idle, and waiting to take up a task. In Fig. 5.9b, some robots experience a failure and remain stationary in the domain. These robots are annotated by arrows, and are treated as stationary obstacles. The robots detect encounters when the physical footprint of another robot overlaps with their sensory footprint. The robots use minimally-invasive control barrier certificates [130] for collision avoidance. 96

5.10 Experimental Results for *Case 1: Comparison with and without closed-loop rate control*. For a swarm of 12 robots operating on the Robotarium, Fig. 5.10a plots the allocation results corresponding to the closed-loop task allocation case. As seen, the swarm achieves and maintains the desired allocation ratio. Fig. 5.10b compares the number of task-transitions per robot per unit time for the closed-loop rate controlled allocation algorithm and the open-loop variant. As seen, the inter-robot encounters allow the robots to estimate the current allocation of the swarm and correspondingly reduce the number of task-transitions. 97

5.11 Experimental Results for *Case 2: Closed-loop task allocation with robot failures*. At time $t = 100$s, 3 robots allocated to task T_1 experience a failure and remain stationary in the domain (see robots overlayed with gray circles and annotated with arrows in Fig. 5.9b). As seen in Fig. 5.11a, this failure disturbs the allocation ratios, which is then reattained by the task allocation algorithm. The performance of the particle filters used to estimate the current allocation of the swarm is plotted in Fig. 5.11b averaged over all the robots. The solid lines represent the average estimate whereas the dotted lines represent the true allocation of the swarm in the two tasks T_1 and T_2. As seen, the estimates track the true allocations allowing the swarm to react to changes in the task allocation. Lastly, Fig. 5.11c plots the number of task-transitions corresponding to the closed-loop and open-loop task allocation algorithms. These results were averaged over 3 experimental runs to better illustrate the effect of the failure. 98

6.1 A differential-drive-like brushbot [45], actuated using two vibration motors and moving on two separate flexible bundles of bristles. Given this construction, we model the dynamics of the robot using unicycle dynamics, with translational and rotational additive noise to represent the unpredictability in the motion of the robot. 101

6.2 Validation of the relation between the average speed of robots and the swarm density using the simulation setup described in Section 6.1.3. Average speed measurements are plotted with red dots, while the best-fit line is shown in blue. At low densities, the average speed equals the self-propelled speed of the robots (set at $v_0 = 4$ in this simulation). As the density increases, robots experience higher collision rates which slows them down. Best-fit collision resolution time $\tau_m = 0.177s$ (see (6.8)). 106

6.3 Snapshots illustrating the formation of high as well as low density regions concurrently in a team of simulated robots operating in a domain with periodic boundary conditions. For a set of simulation parameters ($N = 292, r = 1, v_0 = 4, D_r = 1e^{-4}, D_t = 1e^{-5}$), each snapshot represents the configuration of the system at different times. Inter-robot collisions slow down the robots, which cause additional robots to join the clusters. This leads to the formation of dynamic high-density clusters along with the existence of lower density regions in the domain. 111

6.4 Empirically obtained distribution of robot densities evaluated over a grid of size l^2 with $l = 10$. Gaussian kernel-smoothing was applied on the histogram of robot densities collected over multiple simulations at a constant robot density (simulation parameters: $r = 1, D_t = 1e^{-5}, N = 382$). For high activity parameters, the distribution is seen to be distinctly bimodal, due to the formation of high and low robot density regions as predicted in Section 6.2. For lower activity parameters, the swarm does not phase separate and the density distribution over the grid remains unimodal. This presents a mechanism to characterize the simultaneous existence of higher and lower robot densities in the domain. The displayed data were collected over multiple simulation runs. 113

6.5 Dependence of the average aggregation fraction on the activity parameter of the robots (simulation constants: $r = 1, D_t = 1e^{-5}$). The aggregation fraction measures the fraction of robots which belong to high-density clusters (chosen with a cut-off size $N_c = 4$) averaged over time. As seen, at low activity levels, the extent of aggregation remains low regardless of the density of robots in the domain. As the activity is increased, the fraction of robots in high density clusters increases. In the legend, $\bar{\lambda}$ denotes the mean density of robots in the domain. 114

6.6 Dependence of the aggregation behaviors on the mean density of robots in the domain, as predicted by (6.27). Below a certain density, no significant aggregation is seen in the robots regardless of the activity parameter of the robots (simulation constants: $r = 1, D_t = 1e^{-5}$). This is primarily because robots are able to resolve collisions (therefore dissolving clusters) before other robots can join the cluster. 115

6.7 Snapshots for a team of 26 brushbots propelled at a constant speed in a square environment. Reflective boundary conditions were applied by injecting an angular velocity to the robots when they hit the boundaries. As predicted by phase separation theory, collisions cause the robots to slow down which leads to the formation of high density robot clusters that coexist along with regions of lower robot density. These clusters form and dissolve over time in different places. 116

6.8 The fraction of robots belonging to high density robot aggregations for a team of real brushbots. An aggregation is defined as a collection of robots beyond a cut-off size in physical contact with each other. The robots travel randomly in the domain while colliding with each other. The reduction in speed caused due to collisions precipitates the formation of simultaneous regions of low and high robot density as predicted by the phase separation theory in Section 6.2. As seen, the fraction of robots in aggregations remains fairly significant throughout the experiments. 117

7.1 The Robotarium, a remotely accessible swarm robotics tested, allows users to remotely upload code which gets executed on a team of physical differential-drive robots [151]. In addition to these robots, the hardware infrastructure of the Robotarium consists of an elevated arena where the robots operate, a motion capture system seen attached to the roof above the arena, and wireless chargers embedded into the walls of the arena. 119

7.2 The GRITSBot (pictured on the left) was the first robot to operate on the Robotarium and was designed with a focus on low-cost manufacturing and small form factor [152]. It has now been replaced with the GRITSBot X (pictured on the right), which is designed to provide reliable and robust operations for long periods of time without direct human supervision [151]. 120

7.3 The reachable set $\mathcal{R}(t)$ of a robot with dynamics given in (7.1), is depicted for varying time horizons. The robot is represented as a circle at the center. . 125

7.4 The set $conv(\mathcal{R}(t))$ is shown enclosed within $\mathcal{K}(t)$. In particular, it is illustrated how the curve $\mathcal{C}^{++}(t)$, which represents the curved boundary of $conv(\mathcal{R}(t))$ in the first quadrant, can be over-approximated by a circular arc of radius t, which forms the boundary of $\mathcal{K}(t)$ in the first quadrant. . . . 128

7.5 Minimum area ellipse enclosing $\mathcal{K}(t)$ 134

7.6 Evolution of $\mathcal{R}(t), conv(\mathcal{R}(t)), \mathcal{K}(t)$ and $\xi(\mathcal{K}(t))$ for $t = 2, 5, 12, 25$. As predicted by Theorem 8, the dissimilarity between $conv(\mathcal{R}(t)$ and $\xi(\mathcal{K}(t))$ asymptotically goes to zero. The scale for each figure is different. 135

7.7 The safe time horizon represents the longest time duration for which the robot lies outside the ellipsoidal reachable sets of other robots. Thus, the robot experiencing communication failure can execute its last received velocity command for the corresponding safe time horizon and remain safe. Beyond this, the robot stops moving. 139

7.8 Comparison of the motion of robots with and without safe time horizons. Two robots experience communication failure from $t = 3.1s$ to $t = 8.3s$. In the case when safe time horizons are not used (Fig. 7.8a and Fig. 7.8b), the robots exhibit jerky motion behavior, since they abruptly stop during the communication failure. When safe time horizons are used (Fig. 7.8c and Fig. 7.8d), the robots continue moving by executing their last received velocity command for the corresponding safe time horizon, thus demonstrating the ability of the safe time horizon algorithm to effectively handle communication failures. 141

7.9 Examples of brushbots: metallic rods, brushes and toothbrushes are employed to convert energy of vibrations into directed locomotion. 142

7.10 Two regimes of operation of the brushbot: locomotion is achieved by exploiting vibrations in two different ways, depending on the physical characteristics of the robot. 143

7.11 Differential-drive-like brushbot: two sets of brushes are mounted on the opposite sides of the robot body. Desired linear and angular velocities of the robot body can be achieved by varying the speed of vibration motors mounted on top of each set of brushes. 144

7.12 Differential-drive-like brushbot. In Fig. 7.12a, the exploded view of the CAD model shows, from top to bottom: PCB and battery support (black), top body (orange), vibration motors (brown), bottom body (orange), brushes (purple). Figure 7.12b shows a 3D printed prototype of the brushbot: infrared-reflective balls are mounted on top for tracking its pose. 144

7.13 A swarm of 26 differential-drive-like brushbots (like the one shown in Fig. 7.12) performing coverage control [7]. The boundaries of the Voronoi cells corresponding to each robot are shown in grey. 145

SUMMARY

In naturally occurring swarms—living as well as non-living—local proximity encounters among individuals or particles in the collective facilitate a broad range of emergent phenomena. In the context of robot swarms operating with limited sensing and communication capabilities, this thesis demonstrates how the systematic analysis of inter-robot encounters can enable the swarm to perform useful functions without the presence of a central coordinator. We combine ideas from stochastic geometry, statistical mechanics, and biology to develop mathematical models which characterize the nature and frequency of inter-robot encounters occurring in a robot swarm. These models allow the swarm to perform functions like localization, task allocation, and density regulation, while only requiring individual robots to measure the presence of other robots in the immediate vicinity—either via contact sensors or binary proximity detectors. Moreover, the resulting encounter-based algorithms require no communication among the robots or the presence of a central coordinator, and are robust to individual robot failures occurring in the swarm.

In addition to algorithms which *explicitly* make use of inter-robot encounter measurements, we also investigate a scenario where the physical forces experienced by robots during collision-based encounters *implicitly* lead to the predictable formation of non-uniform density regions in the environment. To this end, we take advantage of a lesser-known formal connection between certain types of active-matter systems and equilibrium thermodynamics to clarify the mechanisms underlying this emergent phenomena.

Throughout the thesis, experiments conducted on real robot swarms vindicate the idea that inter-robot encounters can be advantageously leveraged by individuals in the swarm. In addition to the extensive use of the Robotarium, we deploy some of the developed algorithms on a swarm of brushbots, which are custom built vibration-driven robots that are robust to collisions even at relatively high speeds.

CHAPTER 1
INTRODUCTION

Fueled by advances in embedded systems and communications technology, the last decade has seen increased research focus on the deployment of robot swarms for achieving dynamic and complex objectives due to their redundancy, ability to perform different tasks concurrently, and fault tolerance capabilities (e.g., see [1, 2] and references within). However, for large robot swarms, the issue of coordination among the robots becomes a complex one, often requiring extensive communication between robots and/or a central coordinator which has access to global information regarding the swarm [3]. This has motivated the development of decentralized coordinated control algorithms where local interactions among the robots—typically requiring onboard sensing—lead to desirable emergent properties for achieving various objectives e.g., leaderless allocation of tasks [4], collective motion of interacting immobile robots [5], geometric formations [6], and environmental monitoring [7].

Roboticists have often looked at nature for inspiration when designing group-level behaviors for robot swarms [8]. This is not without reason—animal swarms demonstrate remarkable collective behaviors that emerge via local interactions among the individuals as well as between individuals and the environment, e.g., [9, 10]. In particular, insect-colonies perform sophisticated functions such as traffic regulation [11], resource collection [12], and nest construction [13] in a leaderless fashion while being resilient to environmental disturbances. Such behaviors are not just limited to animal swarms—inanimate collectives like colloidal suspensions and artificial nano-swimmers exhibit a wide-range of emergent phenomena [14, 15]. In particular, the study of far-from-equilibrium physical systems has generated a great deal of excitement [16] given its ability to systematically analyze systems as diverse as bacterial swarms, fish schools, and highway traffic. In these *active matter systems*, an interplay between self-propulsion, local inter-agent interactions, and environ-

mental factors leads to a wide variety of emergent behaviors [17, 18].

These observations provide vital insights into the development of scalable algorithms for coordinating large robot swarms, especially in situations where the deployment scenario precludes the presence of a central coordinator and imposes restrictions on the exteroceptive sensing as well as communication capabilities of individual robots. For instance, robots might have to operate in harsh and unpredictable GPS-denied environments where sensing ranges are extremely limited [19]. Alternatively, in micro-robotics applications [20], size limitations can pose restrictions on the exteroceptive sensing capabilities of individual robots. This motivates the following question: *how can robot swarms equipped with very limited sensing and communication capabilities perform useful functions without the presence of a central coordinator?*

This thesis provides an answer to this question by demonstrating how robots in a swarm can leverage inter-robot encounters to perform useful tasks. We focus on the proximity encounters which occur among robots as a natural byproduct of the motion of robots in a common physical space [21], e.g., during collision avoidance maneuvers or direct physical contact. We demonstrate that such encounters—typically measured by the robots either via onboard contact sensors or binary proximity detectors—can allow the robots to gather information about their surroundings and perform useful tasks like localization [22, 23], task allocation [24], and density regulation [25]. These motivating applications were chosen to highlight how the resulting encounter-based algorithms are decentralized, require extremely limited robot capabilities, and are robust to individual robot failures.

Indeed, plenty of direct evidence suggests that encounters play a crucial role in the functioning of animals swarms, particularly insect colonies [26]. *Solenopsis invicta* ants frequently engage in head-to-head contact to regulate traffic flow and ensure efficient transport of resources in tunnels and narrow trails [11, 27]. *Temnothorax rugatulus* ants have been shown to switch between different emigration strategies based on the rate of encountering other ants within the new nest [28]. Bees have been shown to use aggregation behav-

iors, facilitated by collisions, to find the highest temperature spot in a varying temperature field [29]. This bee colony phenomenon inspired the development of a distributed multi-robot algorithm where the robots collectively aggregate near a light source using inter-robot collisions to trigger measurements from on-board light sensors [30]. Work has also been done on analyzing persistent collisional interactions between a robot and obstacles in the environment [17], planning collisions with the environment for generating optimal trajectories [31], and allowing a robot to localize itself in an environment using a minimal set of sensors, including a contact sensor [32].

As a first step, we take these types of specific observations about collisions and investigate whether or not the collisions can be used as sufficiently rich sources of information to enable the robots to localize themselves in a particular environment. To this end, Chapter 3 envisions a collection of robots moving around somewhat randomly in a domain with known characteristics, equipped with no other sensors than binary, tactile collision sensors. The domain is assumed to be partitioned into a set of connected cells and the task is for the individual robots to establish what cells they are currently occupying by counting collisions. We use the theory of *spatial point processes* [33] to model the distribution of robots in the environment and analytically compute the probability of a robot experiencing a collision in a given cell. A probabilistic localization technique then allows each robot to compute a probability distribution over the different cells that tells the robot how likely it is that it is currently occupying a given cell. Experimental results conducted on real robots validate the idea that inter-robot collisions can indeed facilitate localization.

The science of understanding and modeling collisions among individuals in a collective is an old one. The kinetic theory of gases—which attempts to derive the macroscopic properties of gases by considering their molecular composition and motion—develops models for the frequency of inter-molecular interactions between randomly moving molecules [34]. In contrast to the discrete-time treatment of collisions developed in Chapter 3, adopting these models allows us to develop a *continuous-time* analytical characterization of the

frequency of inter-robot encounters in Chapter 4. Furthermore, we demonstrate that not just physical inter-robot collisions, but also proximity encounters among the robots—e.g., when two robots come close enough to detect each other's presence but do not necessarily collide—can act as sources of information for robots.

In many swarm robotics applications, e.g., in the context of foraging [35] or collection tasks [36, 37], an increase in the frequency of inter-robot encounters typically implies a decrease in the efficiency of the swarm [38]. This is because, with increasing robot density, robots tend to spend more time and effort avoiding collisions with each other, thereby hurting the overall productivity of the swarm. In Section 4.3, we demonstrate that the developed continuous-time encounter model enables an analytical characterization of this interference phenomena. As a motivating application, we consider a swarm of robots performing a distributed collection task and analytically derive the fraction of time that each robot spends performing the primary task as opposed to avoiding collisions with other robots. This leads to the computation of an optimal robot density which achieves a trade-off between the deployed swarm size and the performance of the swarm at the collection task. We develop a decentralized algorithm which enables the robot swarm to achieve this optimal density using only binary encounter measurements.

While the encounter model developed in Chapter 4 provides a continuous-time characterization of encounters, it implicitly assumes that each robot has a zero footprint area—an assumption that contributes to the inaccuracy of the model at higher robot densities. To this end, Chapter 5 uses ideas from the Enskog theory of high density gases [39, 40] as well as stochastic geometry to explicitly account for the finite footprint area of the robots. This is manifested as a correction factor introduced into the previously developed encounter model, and is shown to yield a more accurate prediction of the measured encounter rates experienced by robots.

We develop this corrected encounter model in the context of a motivating application which highlights how leveraging encounter-based information can facilitate leaderless

swarm behaviors which are robust to individual robot failures. To this end, we are partially inspired by red harvester ants, *Pogonomyrmex barbatus* which have been shown to take up midden work (sorting the refuse pile within the colony nest) when encountering other ants performing midden work [41]—an instance where division of labor within the colony is facilitated via encounters. Motivated by such behaviors, Section 5.2 develops a decentralized task allocation mechanism to allocate a set of pre-defined tasks among a swarm of robots by leveraging the spatial interactions occurring among the robots as they move around the domain. We propose a strategy where individual robots switch between different tasks when they encounter other robots in the region. By appropriately designing an encounter-based probabilistic scheme for each robot to transition between tasks, we allow the swarm to achieve the desired task allocation. Furthermore, the inter-robot encounters enable the robots to measure the current allocation of the swarm, enabling a closed-loop regulation of task transitions in a decentralized and leaderless fashion.

Designing collective behaviors for minimally equipped robot swarms can become especially challenging when considering tasks which require them to spread across an environment, e.g., for distributed sensing or environmental surveillance applications [42]. Under these circumstances, the robots might require spatial proximity to facilitate information exchange, while simultaneously performing task related actions [43]. In Chapter 6, we highlight a mechanism to achieve coexisting regions of low and high robot density in swarms with severe constraints on sensing and communication. However, in contrast to the *explicit* use of encounter measurements in Chapters 3, 4 and 5, we demonstrate how the physical forces experienced by robots during collision-based encounters can *implicitly* lead to formation of these regions of non-uniform density [44].

Concepts from statistical mechanics provide a theoretical justification for the formation of these regions of low and high density. We demonstrate that, under certain conditions, a system of self-propelled colliding robots can be mapped to a system of particles at equilibrium, thus allowing us to inherit the rich theory of equilibrium thermodynamics. This

enables us to analytically determine the conditions on the motion parameters of the robots required to display such phase separating behaviors. We validate our ideas on a team of vibration-driven robots, called brushbots [45], which sit atop bristles and achieve directed motion by vibrating them.

Throughout this thesis, we primarily rely on two different robotic platforms to validate the developed encounter-driven algorithms. Chapters 3, 4 and 5 use the Robotarium, a remotely-accessible swarm robotics testbed located at the Georgia Institute of Technology [46], whereas Chapter 6 uses the brushbots: a swarm of vibration-driven robots, ideal for testing the emergence of collision-based behaviors. In Chapter 7, we highlight the design details of both these swarm robotics platforms, while also highlighting the steps taken to ensure their reliable operations. Chapter 8 concludes this thesis by highlighting the primary contributions and outlining future work.

CHAPTER 2
BACKGROUND

We begin this chapter by highlighting some of the existing work in the robotics literature that has focused on proximity-based interactions among robots in a swarm or between robots and the environment. The focus then shifts to naturally occurring collectives—extensively studied in both the biology and physics literature—where encounters have been shown to play a fundamental role in the emergence of organized behaviors.

2.1 Encounters in Robotics

Since robots are embedded in a physical space where they interact with each other and the environment, a vast literature has emerged on the study of their spatial interactions—both in the single-robot and multi-robot setting. In addition to providing a survey of the relevant literature, the following sections aim to provide a broad classification of the major research threads in this area.

2.1.1 Spatial Interference

In the field of swarm robotics, spatial interference—characterized as the competition for physical space among robots—has been an active area of research (e.g., see [47, 48] and references within). A significant portion of antecedent research has considered spatial interference as an undesirable phenomenon—it can lead to increased design complexity in coordination algorithms, waste of resources as robots spend time avoiding each other, and even physical damage due to inter-robot collisions. One approach to curtailing spatial interference is to choose the right number of robots to perform a specific task [49, 50]. To this end, [38] presents a quantitative analysis of the trade-off between robot team size and efficiency in the context of collective search tasks. Another approach is to limit deadlocks

by allowing the robots to decide who gets priority by performing "aggression maneuvers", e.g., [51, 52].

Different strategies for ameliorating the negative effects of spatial interference have been studied in the context of decentralized multi-robot foraging tasks, where the robots find and collect target objects before bringing them back to a pre-designated area, e.g., [53, 54, 55]. In order to prevent robots from getting in each others way, some approaches [56, 57] opt for a bucket-brigading technique where individual robots are restricted to remain in a finite search area, and transport objects to and from their neighbors' search area. In [58], the authors present an adaptive bucket-brigading technique where robots modify their search radii based on the level of interference they experience.

Inspired by interference regulation mechanisms observed in insect colonies, many multi-robot foraging and task allocation techniques allow an unequal workload distribution to emerge among the robots with the aim of regulating robot densities, e.g., [59, 60]. In [11], the authors conduct robophysical experiments to demonstrate that behaviors such as idleness and individual retreating adopted by *Solenopsis invicta* ants can prevent the formation of flow-stopping clogs in narrow and confined environments. In the prey-retrieval task studied in [59], the authors demonstrate via robotic experiments that allowing robots to modify their tendency to participate in the primary task leads to reduced interference. In particular, approaches developed in [61] and [62] allow individual robots to count the number of collisions or measure their own performance, respectively, in order to decide whether to continue foraging or rest. Similar studies predefine participatory tendencies in robots [63], or allow them to adjust it online [35], demonstrating an improvement in efficiency by running experiments.

The effects of interference on the efficiency of a swarm have been studied experimentally e.g., [36, 53, 64], and by developing analytical models [65, 66, 67]. There have also been studies on how interference among robots can be used as a tool to guide the design of multi-robot coordination controllers, e.g., [21]. The work presented in this thesis analyti-

cally characterizes the spatial interference among robots in a swarm as a function of their motion parameters and density. For robots performing a collection task, Chapter 4 derives a continuous-time Markov chain based analytical characterization of the fraction of time spent by robots avoiding collisions as opposed to engaging in productive activities. This leads to a decentralized mechanism via which individual robots can take decisions so as to achieve a desired trade-off between team size and productivity. The development of a Markov chain based framework is borne out of necessity: in order to allow robots to make decisions at an individual level, we require more than a mean-field characterization of the effects of interference.

2.1.2 Stochastic Swarm Models

A major challenge in the modeling and control of large-scale robot swarms is to develop techniques which scale well with the size of the swarm. In situations where the robots are homogeneously distributed, robot-robot as well as robot-environment interactions can be modeled as being analogous to chemical reactions between molecules, thus allowing the use of chemical reaction network (CRN) models [68]. By characterizing the local interactions experienced by robots using such models, many approaches then develop a macroscopic view of the swarm behavior to accomplish functions such as aggregation [69], redistribution of agents among different sites [70], and collaborative manipulation [71]. For instance, [72] enables the decentralized assembly of parts using a swarm of homogeneously distributed robots by modeling their interactions with the parts as CRNs. In [73, 74], the authors develop stochastic descriptions of encounter-based switching among predefined behaviors in individual robots—with applications in boundary coverage and the collective transport of objects using robot swarms.

Such techniques, combined with inspirations from fluid dynamics and biological models, have paved the way for stochastic descriptions of swarm movement, where the swarm is treated as a continuum, e.g., [75, 76]. Beginning with stochastic models to describe

the motion of individual robots, such techniques use the Kolmogorov forward equation to describe the evolution of the probability density of the entire swarm. In particular, this framework provides a natural way to incorporate state feedback-based control techniques to manipulate the behavior of the swarm e.g., [77] and has been applied to various swarm robotics applications such as coverage [78] and task allocation [79].

2.1.3 Spatially Organized Swarm Behaviors

A significant part of the relevant literature has focused on the synthesis of organized behaviors in minimally-equipped robot swarms, e.g., for creating robot aggregates [80], segregation of robots into groups that share a common property [81], clustering of objects [82], or self-assembly [83]. The problem of aggregation becomes especially relevant when the robots have limited sensing capabilities and basic computational resources, since physical proximity is then essential to enable more sophisticated swarm behaviors [84]. In [85], the authors investigate the ability of robots with access to minimal but unrestricted range information to aggregate.

In [86], robots segregate into different groups based on the concept of different encounter radii. Recent work has demonstrated a segregation mechanism which only requires each robot to detect the binary presence of a robot close to it as well as whether the detected robot is of the same type [87]. By allowing the robots to move differently based on such a classification, segregation of the robot team is achieved.

The development of algorithms requiring minimal sensory capabilities of the robots helps us to understand what fundamental behaviors are possible using basic sensing and computational capabilities on the robots [88]. For instance, collections of individually configurable immotile modules interacting with each other mechanically in a confined space can lead to collective motion as demonstrated in [5, 89]. Furthermore, such algorithms enable the deployment of robots in harsh environments—where sensing capabilities are severely restricted, and also help enable novel applications in micro-scale robot swarms.

2.1.4 Collisional Interactions

The robotics literature is scattered with examples where collision-based encounters among robots and the environment are leveraged to achieve specific goals. This is especially true in the context of small and slow robots, where physical collisions can be tolerated. In recent work, the authors in [17] experimentally and numerically study the collisional interactions between a sensorless snake-robot and an array of immovable posts, with the aim of designing control principles for robots navigating cluttered and complex environments. In [90], the authors demonstrate how a collision-tolerant pico-quadrotor can fly through cluttered obstacle-laden environments while sustaining multiple collisions—some of which are leveraged to quickly change direction. Along a similar vein, [31] incorporates planned collisions with the environment into the generation of optimal robot trajectories, and [91] investigates the use of contact-based sensing to localize a single robot in a region with obstacles. In [32], the authors investigate the ability of a robot equipped with a minimal set of sensors, including a contact sensor, to localize itself within a region. However, the contact sensor is used as a boundary detector, and does not provide sensory observations as such.

The work in [30] develops a distributed multi-robot algorithm where the robots use inter-robot collisions to trigger measurements from on-board light sensors, thus enabling collective decision making. This algorithm was directly inspired by the ability of bees to utilize collisions for creating large clusters around an optimal temperature spot, a feat which cannot be achieved by isolated bees on their own [29].

2.2 Encounters in Nature

As discussed in Chapter 1, there are plenty of examples in nature where encounters play a crucial role in the emergence of collective behaviors. Below, we highlight some of the representative literature on this topic, which have served as a motivation for the ideas pursued in this thesis.

2.2.1 Living Collectives

As elucidated in Chapter 1, there is plenty of evidence to suggest that local interactions among individuals in animal swarms contribute to the collective cognition of the swarm [9]. For instance, collisional interactions are often fruitfully leveraged by animals to locomote over complex terrain, as in the case of the snakes navigating through obstacle-laden environments [92], or cockroaches passing through irregular terrain [93].

In particular, ant colonies have demonstrated remarkably complex behaviors made possible via encounters—defined as physical proximity or antennation—occurring among the ants [94]. For example, colonies of the ant species *Leptothorax tuberointerruptus* have been shown to regulate nest size through local interactions between the ants surrounding the brood, new building materials, and the current walls of the nest [13]. As mentioned in Chapter 1, red harvester ants take up midden work (sorting the refuse pile within the colony nest) when they encounter other ants performing midden work.

Encounters play a crucial role in the successful emigration of *Temnothorax rugatulus* ants from an unsuitable nest to a suitable one. As the population size in a candidate nest surpasses a threshold, the ants switch from a slow recruitment-based emigration strategy to a fast transport-based one. It has been shown that the detection of this quorum occurs via the rate of encounters experienced by an ant when she enters the candidate nest [28]. In other ant colonies, encounter rates serve as critical cues which allow the regulation of densities in narrow tunnels and ensure the smooth transportation of materials [11, 27].

2.2.2 Non-living Collectives

There has been a significant amount of interest in the physics community to understand the mechanisms underlying the origins of collective motion in nature. This is partially due to the ubiquity of such behaviors across a broad range of scales, from molecular motion to large animal swarms [95]. While thermodynamics provides well-established literature on deriving macroscopic properties emerging from microscopic phenomena, a vast amount of

the literature only applies to systems at equilibrium—thus limiting its direct applicability in the study of robot swarms.

On the other hand, the physics of *active matter* deals with the study of self-driven particles that convert stored or ambient energy into organized movement, and the associated models have been used to describe systems such as fish schools, colloidal suspensions, and bacterial colonies (see surveys [15, 18] for a discussion on collective phenomena in active matter systems). Such systems are *far-from-equilibrium* due to the interplay between self-propulsion, persistent collisional interactions, and random fluctuations. In a now canonical example, the authors in [96] demonstrated the onset of collective motion when individual particles in the ensemble tended to align with the average direction of motion of the particles in their local neighborhood.

In Chapter 6, we specifically focus on the study of motility induced phase separation [97] which describes how active matter systems consisting of self-propelled particles under purely repulsive interactions can spontaneously phase separate. In particular, it has been shown that, under suitable modeling assumptions, a direct connection can be made between such a system and a passive simple fluid at equilibrium with attractive interactions among the molecules. Systems of the latter kind are well understood [98], thus allowing a large body of analysis to be transferred for the study of active self-propelled systems, as discussed in [97, 99, 100]. We leverage this connection to demonstrate how a swarm of brushbots [45] with severely constrained sensing capabilities can phase separate into regions of high and low robot density.

2.3 Conclusions

This chapter served to highlight the diversity of literature on the study of encounter-based interactions among agents in a collective as well as between agents and the environment. We specifically provided a review of relevant work from the robotics literature, where such interactions have been studied in both single-robot and multi-robot scenarios. This interest

partially stems from remarkable trends in nature where encounters—either collisional or proximity-based—enable useful behaviors. Consequently, the latter part of the chapter focused on specific examples from the physics and biology literature where such interactions have been studied.

Despite a common thread of investigation, perspectives among these studies drastically vary, both in terms of the end goal as well as the tools used to analyze the collective. In this thesis, we focus on (i) the development of mathematical models to systematically analyze proximity encounters occurring in robot swarms, and (ii) demonstrate the potential such models unlock by considering certain motivating applications where robot swarms perform tasks in a completely decentralized manner. This is enabled by combining distinct perspectives and modeling choices from the statistical mechanics, biology, and robotics literature.

CHAPTER 3

LOCALIZATION USING INTER-ROBOT COLLISIONS

Collision avoidance constitutes one of the key mechanisms for making multi-robot systems operate in a safe and orderly manner—to keep the robots safe, collisions should be avoided at all costs. This certainly makes a lot of sense in a number of applications, where collisions have the potential to be catastrophic, e.g., for fleets of unmanned aerial vehicles or platoons of self-driving trucks. Regardless of the algorithms used, by necessity, collision avoidance becomes increasingly dominant as the robot density increases. This is because more and more robots occupy the same space. Consequently, the robots spend more of their time avoiding each other as opposed to progressing the overall team mission (see Section 2.1.1 for a discussion on this phenomena).

In this chapter, we sidestep this issue completely and *embrace collisions as a feature of the multi-robot system*. In fact, as momentum is velocity times mass, collisions become less of an issue as the robots get smaller and slower. As a consequence, for truly large teams of small robots, collisions can no longer be considered catastrophic, as observed in [90]. This fact is moreover observed in naturally occurring swarms, where "mild" collisions are known to happen, as discussed in Chapter 2.

Once one embraces the somewhat unorthodox position that robots are allowed to collide, one can observe that there is actual information to be extracted from the collisions. For example, if a robot collides frequently with other robots, it should be able to infer something about the robot density. This chapter aims to demonstrate how inter-robot collisions can act as an information source in robot swarms and enable the robots to localize themselves in a given environment. We consider a scenario where a swarm of robots, equipped with no sensors other than binary, tactile collision sensors, are moving around somewhat randomly in a domain which has been partitioned into a set of connected cells. The robots

are tasked with using the obtained binary collision measurements to establish which cell they are currently present in.

To develop such a localization technique, we need mobility models to describe the motion of the robots between the cells and collision models to analyze the likelihood of a robot experiencing a collision in a given cell. And, it should be noted already at this point that some of the mathematical models pursued to this end are somewhat simplistic. However, these simplifications should be understood in light of the underlying ambition to harness the power of collisions, and they can ultimately be justified by showing that they can indeed be used as generators of successful, adaptive localization schemes. To this end, we assume that the motion of each robot from one cell to another is derived from a static Markov chain, where the set of cells represents the states of the Markov chain. The Markovian motion model encodes the uncertainty associated with the motion of the each robot, as well as the interactions between the robots which may affect their motion in unpredictable ways. Since each robot does not know its current cell, the states of the Markov chain are hidden. Representing the localization problem in a Hidden Markov Model (HMM) [101] framework allows each robot to update the probability distribution over all the states by incorporating sensory collision information at each time step. The emission probabilities of the HMM correspond to the probability of a robot experiencing a collision in a given cell. A *pointwise maximum a-posteriori probability (PMAP)* estimator [102] is used to compute the best guess of the robot's current cell based on the updated probability distributions.

The rest of this chapter is organized as follow: Section 3.1 introduces some results from stochastic geometry which are required for the development of the collision model. Section 3.2 uses mean-field approximations and a spatial point process to determine the probability of a robot experiencing collisions. Section 3.3 subsequently introduces the Markov chain-based motion model of the robots and develops the localization algorithm. Simulation results illustrate the efficacy of the proposed algorithm itself for robots satisfying the

Markovian motion model. Section 3.4 extends this idea to a continuous-time setting. Section 3.5 concludes the chapter with a deployment of the localization scheme on a team of real colliding robots.

3.1 Stochastic Geometry

Stochastic geometry is the field of mathematics dealing with random spatial patterns. More specifically, spatial point processes are used to study random point patterns and the properties that they satisfy (for e.g., [33]). Point processes are most commonly used as a mechanism to interpret patterns of geometrical objects as a system of points in a suitable space, e.g., for representing trees in a forest ('statistical ecology' [103]), the home addresses of patients with a particular disease ('spatial epidemiology' [104]), the positions of stars and galaxies ('astrostatistics' [105]), or the locations of sensors in a sensor network [106]. For a survey of the applications of spatial point processes, see [107, 108].

In this chapter, we use a homogeneous point process to model the spatial distribution of a team of robots in a given region. A representative point for each robot (which also describes its location in the space) is denoted as a point in the point process. By using existing statistical techniques for computing the density of points and inter-point distance distributions, we develop a collision model for the robots.

This section introduces some basic notation regarding point processes which will be used throughout this chapter. First, we discuss spatial Poisson processes, which exhibit the property of *complete spatial randomness*, implying that the location of one point has no influence on the location of another. This is followed by a discussion of hard-core point processes, where the points face mutual repulsion between each other, and cannot lie closer than a certain "inhibition" distance. This mutual repulsion property is useful in capturing the fact that the locations of the robots cannot be closer than a minimum distance owing to their physical footprint.

Definition 1. *A point process* Φ *in* $\mathbb{R}^d$ *is a random set of points* $\Phi = \{x_1, x_2, \ldots\}$. *If* B *is*

a Borel set, then $\Phi \cap B$ represents the random set of points of Φ which also belong to B.

The simplest point process is called the Poisson point process. There is extensive literature on Poisson processes, e.g., [109, 110], but a basic definition is given below.

Definition 2. *A point process Φ is Poisson, if the random number of points of Φ in a bounded Borel set B, denoted as $\mathcal{N}_\Phi(B)$, has a Poisson distribution with mean $\lambda\mathcal{V}(B)$,*

$$\mathcal{N}_\Phi(B) \sim Poiss(\lambda\mathcal{V}(B)), \tag{3.1}$$

where λ is the intensity of the Poisson process and represents the expected number of points of the process in a set of unit volume. $\mathcal{V}(B)$ denotes the volume of the set B.

Algorithm 1 briefly describes how a Poisson point process within a compact region B can be generated in simulation. However, the point process used to describe the locations

Algorithm 1 Poisson point process

1: Obtain the number of points to be placed inside B, denoted by $\mathcal{N}_\Phi(B)$, according to (3.1).
2: Determine the positions of these $\mathcal{N}_\Phi(B)$ points by selecting uniformly random locations within the set B.

of the robots should capture the fact that the robots have a physical footprint and cannot inter-penetrate. Consequently, we now provide the definition of a hard-core point process, which accurately captures this phenomenon.

Definition 3. *A hard-core point process is a point process in which the constituent points are forbidden to lie closer than a certain positive minimum distance. Mathematically, a point process Φ is hard-core if*

$$\Phi = \{x_1, x_2, \ldots : \|x_i - x_j\| \geq \delta, \forall i \neq j\}, \tag{3.2}$$

where δ is called the inhibition distance.

We consider a homogeneous team of robots, each with a circular footprint of certain radius. This circular footprint can represent the shape of the robots or an impermeable region around each robot. Consequently, the spatial distribution of the robots is modeled as a hard-core point process with an inhibition distance equal to twice the radius of the robots. This ensures that no two robot footprints can overlap.

A common method of generating hard-core point processes is to perform a dependent-thinning procedure on an underlying "parent" spatial Poisson process, where points are successively removed until the desired hard-core conditions are satisfied by all the remaining points [33]. In this chapter, we generate the popular *Matern II* hard-core process [111], which has an analytically tractable intensity, and whose corresponding thinning procedure is simple and efficient. Algorithm 2 outlines the steps for generating a *Matern II* hard-core point process with an inhibition distance δ. Using the concept of a hard-core point process, the next section develops a mathematical model for the collisions occurring between robots.

Algorithm 2 *Matern II* hard-core point process with inhibition distance δ

1: Generate a parent Poisson point process Φ_p of intensity λ_p within the desired set B, using Algorithm 1.
2: Assign a uniformly independent number in $(0, 1)$ (called as a mark) to each point in Φ_p. Denote the mark of a point x in Φ_p as $m(x)$.
3: Retain a point x of Φ_p with mark $m(x)$, if the ball $B(x, \delta)$ contains no point of Φ_p with marks smaller than $m(x)$. A point that has been thinned still has a role to play in the rest of the thinning process. The intensity of the obtained hard-core process is given by,

$$\lambda = \frac{1 - exp(\lambda_p \mathcal{V}_b(\delta))}{\mathcal{V}_b(\delta)}, \tag{3.3}$$

where $\mathcal{V}_b(\delta)$ is the volume of a ball of radius δ.

3.2 Collision Model

In order to use collision information to localize the robots, we first must establish a model for how often collisions occur among the robots. This section introduces both the underly-

ing localization problem and provides estimates for the probability of a robot experiencing a collision at a given time as a function of the distribution of robots over the domain of interest.

Consider a team of robots deployed over a compact and connected planar domain D, which is divided into M connected and non-overlapping cells $D_1, D_2, \ldots, D_M$, such that,

$$D = \bigcup_{j=1}^{M} D_j, \tag{3.4}$$

with the corresponding environmental index set being $\mathcal{M} = \{1, \ldots, M\}$.

In order to mathematically model the spatial distribution of the robots and utilize this to compute the frequency of collisions, we break our analysis into two parts: for time $k = 0$ when the robots are deployed, and for times $k > 0$, when the robots move from one cell to another based on a motion model.

3.2.1 Deployment at time $k = 0$

As described in Section 3.1, we model the geometric distribution of the robots as a spatial point process, which are useful tools for interpreting patterns of objects and studying their properties. A representative point of each robot denotes a point in the point process.

Definition 4. *Within each cell D_j, at time $k = 0$, the set describing the locations of the robots, $\mathbb{X}_j(0)$, is the intersection of a homogeneous point process Φ_j and the set D_j,*

$$\mathbb{X}_j(0) = \Phi_j \cap D_j, \;\; \forall j \in \{1, \ldots, M\}. \tag{3.5}$$

Each robot in the domain has a physical footprint, which enforces a minimum separation between the points in the point process—since the robots cannot physically interpenetrate, the points in the point process cannot be closer than a certain distance. This physical constraint can be captured by using hard-core point processes [111], where the

points are separated by a minimum "inhibition" distance (see Section 3.1 for a formal definition).

Definition 5. *Let the location of each robot i be $x_i \in D$, and let the footprint of the robot be a ball $B(x_i, r)$ centered at x_i with radius r. Then, each point process $\Phi_j, j = 1, \ldots, M$ is a hard-core point process with an inhibition distance $2r$ (see Fig. 3.1).*

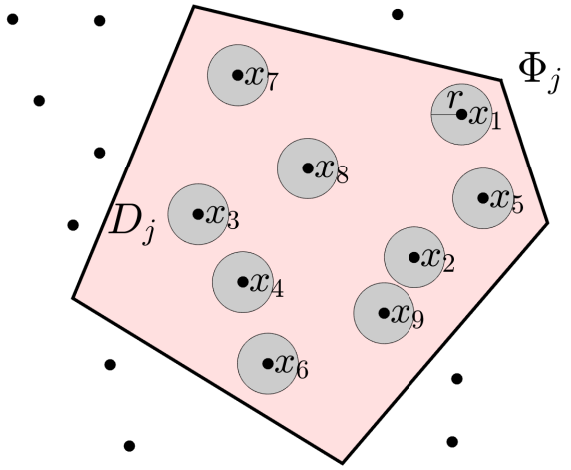


Figure 3.1: A mathematical model of the spatial distribution of robots within a cell $D_j \subset D$, which is depicted here as the shaded area. Φ_j is a hard-core point process consisting of infinite points in the space $\mathbb{R}^2$, illustrated by the dots. As described in Definition 4, the location of the robots at time $k = 0$, denoted here by $\mathbb{X}_j(0) = \{x_1, x_2, \ldots, x_9\}$, are points of the hard-core point process Φ_j that happen to lie inside the set D_j, i.e., $\mathbb{X}_j(0) = \Phi_j \cap D_j$. The robots are depicted by circles of radius r. The inhibition distance of the hard-core point process is $2r$.

Note that in Definition 5, the time dependence of the location of robot i has been hidden for the sake of notational clarity. Each hard-core point process Φ_j can be characterized by an intensity $\bar{\lambda}_j$, which is a measure of the density of robots within the cell D_j. We now define the intensity of each point process in terms of the expected number of robots in the cell.

Definition 6. *Let $\bar{N}_j = \mathcal{N}_{\Phi_j}(D_j) = |\mathbb{X}_j(0)|$ denote the number of robots in cell D_j. The intensity of the point process Φ_j, denoted as $\bar{\lambda}_j$, is the expected number of robots per unit*

area in cell D_j,

$$\bar{\lambda}_j = \frac{\mathbb{E}[\bar{N}_j]}{\mathcal{V}(D_j)}, \tag{3.6}$$

where $\mathcal{V}(D_j)$ represents the area of cell D_j.

This allows us to define the expected total number of robots deployed at time $k = 0$.

Definition 7. *The expected total number of robots in the arena, denoted as N_{total}, is given as,*

$$N_{total} = \sum_{j=1}^{M} \mathbb{E}[\bar{N}_j] = \sum_{j=1}^{M} \bar{\lambda}_j \mathcal{V}(D_j). \tag{3.7}$$

Note that since hard-core point processes—which represent robot locations— are obtained by performing a probabilistic thinning procedure (see Algorithm 2), the number of robots deployed in each cell at time $k = 0$ is a random variable with a known expectation.

3.2.2 Movements at times $k > 0$

After the deployment of robots at time $k = 0$, each robot moves from one cell to another, so that at a given time k, let the number of robots in each cell be denoted by the vector $N(k) = [N_1(k), N_2(k), \ldots, N_M(k)]^T$, and the intensity of the points within a cell be represented as $\lambda(k) = [\lambda_1(k), \lambda_2(k), \lambda_3(k), \ldots, \lambda_M(k)]^T$. The initial intensities are given by $\lambda(0) = \bar{\lambda} = [\bar{\lambda}_1, \ldots, \bar{\lambda}_M]^T$, and the number of robots in each cell as $N(0) = \bar{N} = [\bar{N}_1, \ldots, \bar{N}_M]^T$, which have been defined in Section 3.2.1. For times $k > 0$, the total number of robots in the domain does not change, i.e.,

$$\sum_{j=1}^{M} N_j(k) = \sum_{j=1}^{M} \bar{N}_j, \ \forall k > 0. \tag{3.8}$$

In each cell, the robots may collide with other robots in that cell. Since the number of robots in each cell differs, and so does the size of the cell, the rate of collisions experienced by the robots will be different.

The path planning literature contains several ways in which the probability of collisions

can be calculated under motion and sensory uncertainty e.g., [112, 113]. In many cases, Monte Carlo based techniques have been used to obtain estimates of the collision probabilities, e.g., [114, 74], while the majority of the analytical techniques developed compute the probability of collision along a specific path taken through the workspace in the presence of stationary or moving obstacles, e.g., [115, 116].

Neither of these approaches quite serve the purpose here, where the collisions experienced by a robot in a given cell will depend on the motion of all the other robots in the cell. Luckily, mean-field theory [79, 117, 77] provides an elegant way to simplify such complex interactions by allowing us to replace the effect of other robots on a single robot by an averaged effect. To perform the mean-field approximation, we simply assume that the effect of the other robots on a given robot can be approximated with a single collision probability. Thus, at each time step k, we will compute the probability of a robot experiencing a collision in a particular cell D_j, $j \in \mathcal{M}$, which we will denote by $\phi_j(k)$. (Note that the index here only refers to the cell since this probability is the same for all robots under the mean-field approximation.)

In order to develop a stochastic collision model that is rich enough to be relevant yet simple enough to be analytically tractable, we make the following assumptions:

Assumption 1. *At each time k, the locations of the robots in each cell D_j, described by the set $\mathbb{X}_j(k)$, can be modeled as points belonging to a hard-core point process of intensity $\lambda_j(k)$ and inhibition distance $2r$.*

Assumption 2. *Collisions occur when the footprints of the robots overlap, i.e., when $B(x_i, r) \cap B(x_j, r) \neq \emptyset,\ i \neq j$. This is obviously physically not valid, but it both allows for mathematical simplicity yet is, as we will see, empirically justified using Monte Carlo simulations.*

Assumption 3. ***[Mean-field approximation]*** *The probability of any robot experiencing a collision in cell D_j at time k, is equal to the probability that an "exogenous" robot dropped*

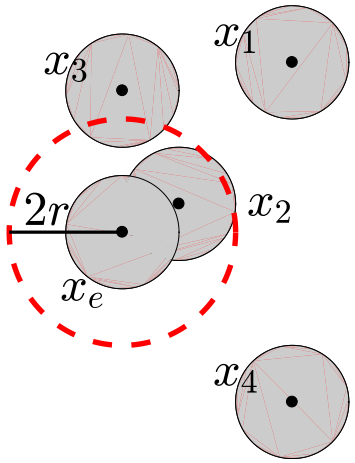


Figure 3.2: At time k, the locations of the robots in cell D_j is illustrated by the set $\mathbb{X}_j(k) = \{x_1, \ldots, x_4\}$. The probability of collision experienced by an exogenous robot (whose position is denoted by x_e), dropped at a uniformly random location in cell D_j, is equal to the probability that the random variable denoting the distance between x_e and its first nearest neighbor in the set $\mathbb{X}_j(k)$ is less than $2r$. Furthermore, as specified in Assumption 3, this is also equal to the probability of collision experienced by any robot in $\mathbb{X}_j(k)$.

at a uniformly random location within cell D_j collides with a robot belonging to the set $\mathbb{X}_j(k)$.

Under this model, a collision is said to occur if this *exogenous* robot, whose position is denoted as x_e, is dropped at an uniformly random location within cell D_j at time k, and spatially overlaps with a robot in the cell belonging to set $\mathbb{X}_j(k)$. The probability of such an event occurring is equal to the probability that the random variable denoting the distance between x_e and its first nearest neighbor in the set $\mathbb{X}_j(k)$ is less than $2r$ (see Fig. 3.2). We formalize this distance metric in the following definition.

Definition 8. *The contact distance, denoted as $d(x, \mathbb{X}_j(k))$, is a random variable which denotes the shortest distance from an independently chosen uniformly random point $x \in D_j$ and the set of robot locations $\mathbb{X}_j(k)$.*

Using the definition of the contact distance, combining this with Assumption 3, and utilizing results from [118], the following theorem presents an analytical expression for the probability of collision experienced by a robot in a given cell at time k.

Theorem 1. *The probability of a robot experiencing a collision in cell D_j at time k, is given by,*

$$\phi_j(k) = 1 - exp\Big(- \int_0^{2r} 2\pi\lambda_{p,j}(k) q \eta(q, 2r) dq \Big), \tag{3.9}$$

where,

$$\eta(q, \delta) = \frac{1 - exp[-\lambda_{p,j}(k)(\pi\delta^2 - l_2(q, \delta))]}{\lambda_{p,j}(k)(\pi\delta^2 - l_2(q, \delta))}, \tag{3.10}$$

$$l_2(q, \delta) = \begin{cases} \pi q^2 : 0 < q < \frac{\delta}{2} \\ q^2 cos^{-1}(1 - \frac{\delta^2}{2q^2}) + \\ \delta^2 cos^{-1}(\frac{\delta}{2q}) - \frac{1}{2}\delta\sqrt{4q^2 - \delta^2} : q \geq \frac{\delta}{2} \end{cases}, \tag{3.11}$$

and $\lambda_{p,j}(k)$ is the intensity of the parent Poisson point process which is described in Section 3.1 and can be computed using (3.3) *for a desired hard-core process intensity $\lambda_j(k)$.*

Proof. From Definition 8, we know that the probability of an *exogenous* robot, whose position is denoted as x_e, colliding with any robot in cell D_j at time k, can be expressed as the probability that the random variable $d(x_e, \mathbb{X}_j(k))$ takes a value less that $2r$ (see Fig. 3.2). Furthermore, Assumption 3 states that, given the mean-field approximation, this probability is equal to the probability of any robot in cell D_j experiencing a collision at time k. This implies that,

$$\phi_j(k) = \mathcal{F}_{e \to \mathbb{X}_j(k)}(2r), \tag{3.12}$$

where,

$$\mathcal{F}_{e \to \mathbb{X}_j(k)}(2r) = Prob(d(x_e, \mathbb{X}_j(k)) \leq 2r). \tag{3.13}$$

$\mathcal{F}_{e \to \mathbb{X}_j(k)}(l)$ is the cumulative distribution function of the random variable $d(x_e, \mathbb{X}_j(k))$

evaluated at some distance l.

Results presented in [118] give us an analytical expression for the cumulative distribution function of the distance between an independently and uniformly chosen point in the domain, and its nearest neighbor in a hard-core point process. Since, according to Assumption 1, $\mathbb{X}_j(k)$ is considered to be part of a hard-core point process, we can directly utilize this result to compute $\mathcal{F}_{e\to\mathbb{X}_j(k)}(l)$. Evaluating the function for $l = 2r$, gives us the probability of collision experienced by a robot in cell D_j at time k. □

We validate (3.9) by comparing the derived expression for the probability of collision in a given region against a Monte Carlo simulation which computes the collision rate by repeatedly dropping an exogenous robot, in an uniformly random manner into the region and then checking for collisions (see Fig. 3.3). For a particular intensity λ, an empirical collision rate is estimated, along with a 95% confidence interval, which is computed using the Clopper-Pearson method [119]. The presence of a slight positive bias in the Monte Carlo estimates in Fig. 3.3, especially at high densities can also be observed in the results presented in [118], and likely arises from approximations made when analytically deriving an expression for $\mathcal{F}_{e\to\mathbb{X}_j(k)}(l)$ (see [118]).

As the robots experience collisions while moving from one cell to another, we fix our attention on a particular robot—it does not matter which since they are all identical—and we will in subsequent sections suppress the subscript denoting the identity of the robot. To this end, we associate a binary variable with the collision events experienced by this robot, i.e.,

$$\gamma(k) = \begin{cases} 1 \text{ if the robot experiences a collision at time } k, \\ 0 \text{ otherwise.} \end{cases} \tag{3.14}$$

The question then becomes, given a string of such observations $\Gamma(k) = [\gamma(k), \ldots, \gamma(1)]$, can the robot figure out which cell it currently is in? In order to answer this question, we need a motion model, which is developed in the next section.

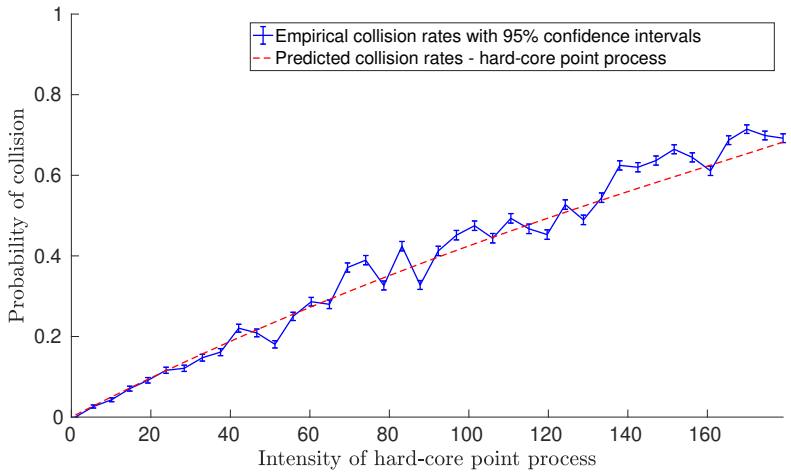


Figure 3.3: Comparison of the collision probabilities predicted by the hard-core point process model against the collision rates obtained using a Monte Carlo simulation. For each hard-core process intensity, an exogenous robot is dropped 5000 times into the cell, and the number of collisions is recorded to obtain the Monte Carlo estimate of the probability. In addition, a 95% confidence interval is computed (illustrated by the vertical bars at each data point) for the estimated probabilities according to the Clopper-Pearson confidence interval method. As seen, the theoretical probabilities match the simulation data for varying intensities, and serve as good estimates of the true collision probabilities under the developed collision model.

3.3 Collisions through Hidden Markov Models

3.3.1 The Forward Algorithm

In the previous section, we developed estimates for the probability of a robot experiencing collisions in a given cell. In order to utilize this information to localize the robots, we need a mathematical model for the motion of the robots between the cells. Due to the lack of locational information and detailed sensing, as well as the constant interaction between robots, the motion of the robots between cells is stochastic.

This lends itself naturally to a Markov chain-based motion model, where the robots move between the cells based on transition probabilities. This approach is commonly used when dealing with uncertainty in robot motion since it efficiently models the stochasticity inherent in the system, e.g., [120, 121]. Thus, the robots move from one cell to another based on some transition probabilities which are assumed to be time invariant. In this

section, we exploit this Markovian motion model and develop a corresponding localization algorithm.

Definition 9. *Let the current cell of robot r at time k be denoted by $q_r(k)$, i.e., if robot r is in cell D_j at time k, then $q_r(k) = j$. The motion of the robot from one cell to another is Markovian, and can be represented by a transition probability matrix P_r, whose elements are given by,*

$$P_{ij,r} = \text{Prob}\left(q_r(k+1) = i \mid q_r(k) = j\right),\ i, j \in \mathcal{M}. \tag{3.15}$$

Similar to Section 3.2, we apply the mean-field approximation and assume that all the robots are subject to the same effects from the environment and other robots. This implies that all the robots can be assumed to move according to the same transition probabilities. So, (3.15) can be replaced by the following equation, where we suppress the index of the robot, since this now holds for all robots,

$$P_{ij} = \text{Prob}\left(q(k+1) = i \mid q(k) = j\right). \tag{3.16}$$

The resulting transition probability matrix P is chosen to reflect the physical accessibility of one cell from another as well as other mobility considerations. In Section 3.3.2, a particular choice of transition probabilities is made but for now, we simply assume that these probabilities are somehow made available to us.

In order to complete the definition of the Markov chain, we also define the initial probability distribution of each robot over the different cells. Owing to the mean-field approximation made earlier, this initial probability distribution, denoted as $\mu = [\mu_1, \mu_2, \ldots, \mu_M]^T$ will be the same for all the robots,

$$\mu_j = \frac{\bar{N}_j}{N_{total}} = \frac{\bar{\lambda}_j \mathcal{V}(D_j)}{N_{total}}, j \in \mathcal{M}, \tag{3.17}$$

where N_{total}, defined in (3.8), represents the expected total number of robots in the domain

D. The Markov chain is now formally defined using the transition probability matrix P, and the initial distribution μ.

We now use this Markovian motion model to derive how the intensities $\lambda_j(k)$ evolve with time.

Lemma 1. *As the robots move from one cell to another, let* $\lambda(k) = [\lambda_1(k), \lambda_2(k), \ldots, \lambda_M(k)]^T$ *represent the expected number of robots per unit area in each cell. Given the transition probability matrix* P*, the intensities evolve according to the following equation,*

$$\lambda(k+1) = (\mathcal{D} \odot P)\lambda(k), \tag{3.18}$$

where $\mathcal{D}$ *is a matrix comprising of elements*

$$\mathcal{D}_{ij} = \frac{\mathcal{V}(D_j)}{\mathcal{V}(D_i)}, \tag{3.19}$$

and $\odot$ *symbolizes the Hadamard product.*

Proof. At time $k+1$, the number of robots in cell D_j will simply depend on how many robots left and entered the cell in one time step,

$$N_j(k+1) = N_j(k) + N_{added} - N_{removed}, \tag{3.20}$$

$$\mathbb{E}[N_j(k+1)] = \mathbb{E}[N_j(k)] + \sum_{i=1, i\neq j}^{M} P_{ji}\mathbb{E}[N_i(k)] - \sum_{i=1, i\neq j}^{M} P_{ij}\mathbb{E}[N_j(k)]. \tag{3.21}$$

Adding and subtracting the term $P_{jj}\mathbb{E}[N_j(k)]$,

$$\mathbb{E}[N_j(k+1)] = \mathbb{E}[N_j(k)](1 - \sum_{i=1}^{M} P_{ij}) + \sum_{i=1}^{M} P_{ji}\mathbb{E}[N_i(k)], \tag{3.22}$$

$$\mathbb{E}[N_j(k+1)] = \sum_{i=1}^{M} P_{ji}\mathbb{E}[N_i(k)]. \tag{3.23}$$

From (3.6), we know that,

$$\lambda_j(k+1) = \frac{\mathbb{E}[N_j(k+1)]}{\mathcal{V}(D_j)} = \frac{1}{\mathcal{V}(D_j)} \sum_{i=1}^{M} P_{ji}\lambda_i(k)\mathcal{V}(D_i). \tag{3.24}$$

This gives us the desired result. □

Note that the states of the Markov chain are hidden, since the robots do not know which cell they are currently in. However, they are making observations in the form of collision measurements over the set $\{\Gamma_1, \Gamma_2\} = \{0, 1\}$. The observation probabilities—the probability of observing a 0 or a 1 in cell D_j—thus becomes,

$$G_{lj}(k) = \text{Prob}\left(\gamma(k) = \Gamma_l \mid q(k) = j\right),\ l = 1, 2,\ j \in \mathcal{M}. \tag{3.25}$$

Since the robots are homogeneous, we already know that the probability of observing a collision in cell D_j at time k is $\phi_j(k)$, i.e.,

$$G_{1j}(k) = 1 - \phi_j(k) \tag{3.26}$$

$$G_{2j}(k) = \phi_j(k), \quad j \in \mathcal{M}. \tag{3.27}$$

Note that even though we are focusing our attention on a single robot, all the other robots are moving around at the same time and it is the statistics of this motion that ultimately determines the collision probabilities.

We have now finally obtained the hidden Markov model $\mathcal{H} = (P, G, \mu)$. This construction allows each robot to compute and update the probability of being in a particular cell at time k given observations up to and including time k.

Let $\delta_k(i)$ denote the probability of the chosen robot being in cell D_i at time k, given the sequence of observations $\Gamma(k) = [\gamma(k), \ldots, \gamma(1)]$. This is given by the posterior probabil-

ity in the context of the hidden Markov model $\mathcal{H}$,

$$\delta_k(i) = \text{Prob}\left(q(k) = i \mid \Gamma(k), \mathcal{H}\right). \tag{3.28}$$

One way to estimate the current cell of the robot is to simply pick the state corresponding to the highest posterior probability,

$$q^*(k) = \underset{i \in \mathcal{M}}{\text{argmax}}\, \delta_k(i). \tag{3.29}$$

Such an estimator is called the *pointwise maximum a-posteriori probability (PMAP)* estimator [101], since it picks the state with the highest posterior probability. This estimator is *optimally accurate* in the sense that it maximizes the expected number of correct estimates, e.g., [122].

In order to compute $\delta_k(i)$ at each time instant k, we define a variable $\alpha_k(i)$ which denotes the joint probability of obtaining a sequence of observations $\Gamma(k)$ and being in state D_i at time k,

$$\alpha_k(i) = \text{Prob}\left(\Gamma(k), q(k) = i \mid \mathcal{H}\right). \tag{3.30}$$

The probability of obtaining a sequence of observations is now simply the sum of $\alpha_k(i)$ over all the states,

$$\text{Prob}\left(\Gamma(k) \mid \mathcal{H}\right) = \sum_{j=1}^{M} \alpha_k(j). \tag{3.31}$$

Thus, $\delta_k(i)$ can be expressed as,

$$\delta_k(i) = \frac{\text{Prob}\left(\Gamma(k), q(k) = i \mid \mathcal{H}\right)}{\text{Prob}\left(\Gamma(k) \mid \mathcal{H}\right)},$$

or, equivalently,

$$\delta_k(i) = \frac{\alpha_k(i)}{\sum_{j=1}^{M} \alpha_k(j)}. \tag{3.32}$$

We can now use the recursive forward algorithm [101] to compute values of $\delta_i(k)$. Algorithm 3 outlines the steps required to generate the PMAP estimates.

Algorithm 3 PMAP Estimation - The Forward Algorithm

1: $k = 0$, $\alpha_0(i) = G_{\gamma(0)i}(0)\mu_i$, $i \in \mathcal{M}$
2: **while** `true` **do**
3: $\quad \delta_k(i) = \frac{\alpha_k(i)}{\sum_{j=1}^{M} \alpha_k(j)}$, $i \in \mathcal{M}$
4: $\quad q^*(k) = \text{argmax}_{i \in \mathcal{M}}\, \delta_k(i)$
5: $\quad \gamma(k+1) \leftarrow \{0, 1\}$ (collision measurement)
6: $\quad \alpha_{k+1}(i) = (\sum_{j=1}^{M} \alpha_k(j) P_{ij}) G_{\gamma(k+1)i}(k+1)$, $i \in \mathcal{M}$
7: $\quad k = k + 1$
8: **end while**

Step 1 initializes the forward probabilities at time $k = 0$, as the joint probability of being in cell D_i and making an observation $\gamma(0)$. The posterior probabilities are computed in step 3 using (3.32). The PMAP estimator simply picks the state with the highest posterior probability in step 4. Step 6 outlines the induction step of the forward algorithm. The product $\alpha_k(j)P_{ij}$ gives the joint probability of reaching cell D_i at time $k+1$ via cell D_j at time k and making the partial observation $\Gamma(k)$. By summing over all the possible states, $\sum_{j=1}^{M} \alpha_k(j)P_{ij}$ gives us the joint probability of reaching cell D_i at time $k+1$ and making observations $\Gamma(k)$. Finally, $\alpha_{k+1}(i)$ is obtained by multiplying $\sum_{j=1}^{M} \alpha_k(j)P_{ij}$ with $G_{\gamma(k+1)i}(k+1)$, which takes into account the latest collision measurement $\gamma(k+1)$.

Algorithm 3 assumes that all robots know the initial distribution of robots μ, the initial hard-core intensities $\bar{\lambda}$, the size of each cell $\mathcal{V}(D_j), j \in \mathcal{M}$, and the Markov transition matrix P. There is no communication needed between the robots before or during the execution of the algorithm.

3.3.2 Simulation Results

In order to illustrate the capabilities of Algorithm 3, we consider a scenario where robots are moving around a circular track comprised of M segments each of width w_j, $j \in \mathcal{M}$ (see Fig. 3.4). The Markov matrix P is chosen in such a way that at each time step k,

the robots can either remain in the same segment or move to the next segment. Assuming that the transition between cells occurs in only one direction, the width of each segment w_j not only affects the collision probabilities but also the congestion in each cell. We model this congestion by letting the probability of a robot staying in the same cell be inversely proportional to the width of the cell: the narrower the segment, the higher is the probability of getting stuck,

$$P(i,i) = \epsilon\frac{1}{w_i}, P(i+1,i) = 1 - P(i,i),\ i = 1,..,M-1 \tag{3.33}$$

$$P(M,M) = \epsilon\frac{1}{w_M}, P(1,M) = 1 - P(M,M), \tag{3.34}$$

where ϵ is chosen so as to restrict the probabilities between 0 and 1.

At time $k = 0$, based on the hard-core process intensities $\bar{\lambda} = [\bar{\lambda}_1, \ldots, \bar{\lambda}_M]^T$, we first generate the corresponding Poisson point process in each cell, and use the probabilistic dependent thinning procedure shown in Algorithm 2, to simulate the locations of the robots in each cell. Thus, the total number of robots deployed at time $k = 0$, is a random number with an expectation N_t, as described in Section 3.2.

For times $k > 0$, each robot moves from one cell to another according to the Markov chain, and is placed at a random non-overlapping position within the cell. The resulting spatial disribution of robots in a cell is not a strict hard-core point process—however, as stated in Assumption 1, it is explicitly assumed that the spatial distribution can be modeled as a hard-core point process. The assumed equivalence between the simulated distribution of robots and a true hard-core point process within the cell is justified in that it allows the total number of robots in the domain to be constant for all times $k > 0$. Moreover, the collision probabilities experienced by the robots in the simulation and the collision probabilities corresponding to an equivalent true hard-core point process as prescribed by Theorem 1, are sufficiently similar.

In the simulation scenario, we developed a circular track with 10 individual cells of

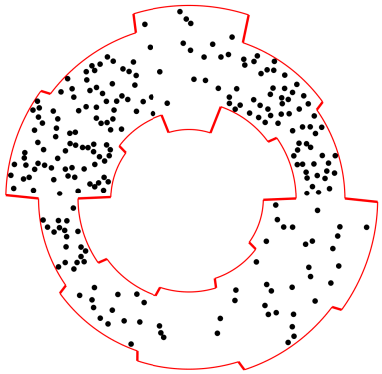
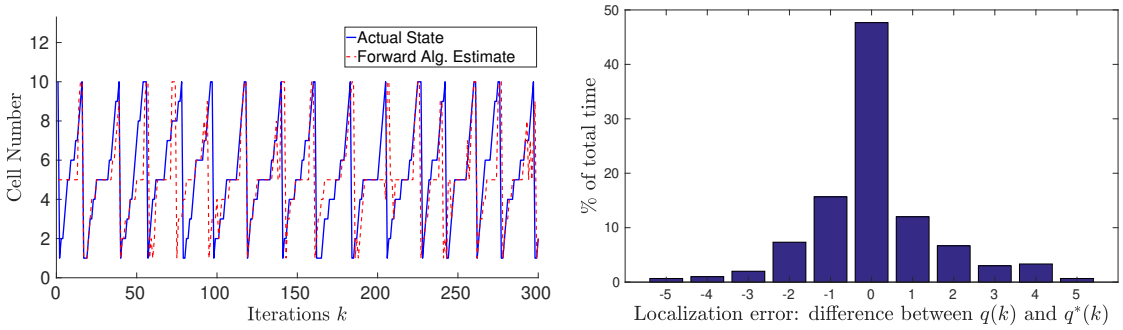


Figure 3.4: A circular track with 10 cells of varying widths. In each cell, at time $k = 0$, robots are deployed according to a hard-core point process of intensity $\bar{\lambda}_j$. For times $k > 0$, each robot moves from one cell to another according to a Markovian motion model.

widths $[0.18, 0.25, 0.47, 0.32, 0.39, 0.15, 0.22, 0.29, 0.36, 0.43]$, and $\epsilon = 0.07$ (see Fig. 3.4). The simulations were conducted for 300 iterations, and the true cell of each robot along with the estimated cell was recorded.

Figure 3.5a shows the true cell and the cell estimates for a randomly chosen robot in the team of robots traversing the circular track. As seen, using only collision measurements, the PMAP estimator generates estimates which closely track the true cell of the robot as the robot moves around the domain.

In order to rigorously analyze the tracking capabilities of the developed localization filter, we ran multiple simulations keeping the initial hard-core process intensities $\bar{\lambda}$ and the Markov probabilities P fixed. Figure 3.5b quantitatively analyzes the localization error for all the robots over the multiple simulations. At any given time k, the localization error of the filter for a given robot is computed as the difference between the true cell index of the robot $q(k)$, and its PMAP estimate $q^*(k)$. Thus, an error of 0 implies perfect localization, $\{+1, -1\}$ implies that the cell adjacent to the true cell was picked, etc. As seen, for majority of the time, the estimator either performs perfect localization, or localizes the robot to a cell adjacent to the true one.



(a) Localization Results

(b) Average Localization Error

Figure 3.5: Simulation results of the probabilistic localization technique for a team of robots moving in a 10-cell domain. Figure 3.5a shows the true cell and localization estimates for a randomly chosen robot in the domain. As seen, within a few iterations, the estimates closely track the true cell of the robot. Figure 3.5b illustrates the distribution of the quantized localization error for all the robots, collected over multiple simulation runs. Each simulation was initialized with the same hard-core process intensities $\bar{\lambda}$, and the same Markov transition probabilities P. The localization error is the difference in the cell index between the true cell of the robot $q(k)$ and its PMAP estimate $q^*(k)$. As seen, the estimator perfectly localizes the robot 47% of the time, and the estimate is within one cell of true cell 77% of the time. Note that a localization error of 5 and -5 correspond to the same cell (see Fig. 3.4).

3.3.3 Connections with Continuous-time Motion of Robots

Until now, we considered a discrete-time motion model for the robots, where each robot transitioned between cells at discrete time steps. At each time k, we computed the probability of a robot experiencing a collision. It is worth noting that the developed collision model is agnostic to the underlying motion of the robots as long as the spatial distribution of the robots in each cell at discrete time epochs k can be described by a hard-core point process. Thus, the discrete-time formulation significantly reduces the complexity of modeling interactions between robots, and allows us to develop a collision model which applies to any motion pattern satisfying this spatial distribution constraint. To illustrate this point, we now introduce a particular continuous-time motion model which yields a spatial distribution of robots that can be well-approximated as a hard-core point process.

We model the planar motion of each robot using single-integrator dynamics,

$$\dot{x} = u, \tag{3.35}$$

where u is the applied control velocity, and as before, the index of the robot is suppressed. Since the robots do not know where they are based on traditional sensing modalities, we simply let each robot travel with a fixed speed v in a randomly chosen direction until it collides with another robot or with the walls of the domain. Essentially, each robot performs a Brownian-like random walk in the domain [96].

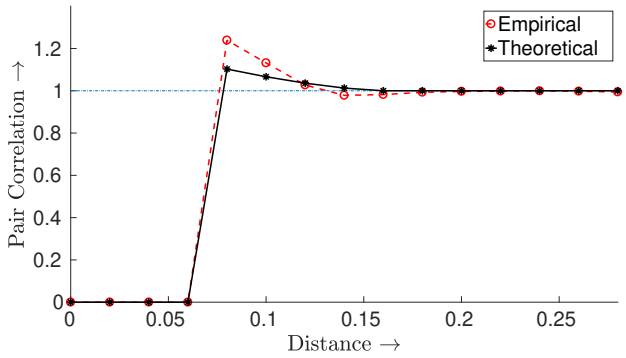


Figure 3.6: Comparison of the empirically obtained pair-correlation function for a given team of robots executing a random walk motion model ($\lambda = 30, r = 0.04, \tau = 0.1$) against the theoretical pair-correlation function for a hard-core point process of the same intensity. The empirical pair correlation was evaluated by noting the positions of the robots at regular intervals of time $t_k, k = 1, 2, \ldots$ and computing the average pair correlation between the points. The similarity in the two curves demonstrates how the random walk motion model yields a spatial distribution of robots which is well-represented by a hard-core point process.

We demonstrate that the resulting spatial distribution of robots executing the random walk motion is well-approximated by a hard-core point process. For a simulated team of robots executing the random walk, we note the locations of the robots at time epochs $t_k, k = 1, 2, \ldots$, and analyze the resulting spatial distribution of the set of points. This is done by empirically computing the average pair-correlation function [33] corresponding to the set of robot locations. The pair-correlation function measures the relative likelihood of finding another point at a certain distance with respect to a given point and is used

to identify the spatial point process which best represents a given set of points. Fig. 3.6 plots the average pair-correlation function empirically computed from the simulation ($\lambda = 30, r = 0.04, \tau = 0.1$) and plots it against the theoretical pair correlation function of a Matern Type-II hard core point process of the same intensity (see [123] for the analytical expression). The similarity in the two curves illustrates that the spatial distribution of robots executing the random walk motion is well represented by a hard-core point process. In general, a variety of factors such as changes in the motion of a robot following a collision, spatio-temporal properties of the trajectories followed by the robots, and the discretization time interval will determine whether a particular motion model can be abstracted by a hard-core point process model.

3.4 Continuous-time Extension

We now demonstrate how the localization algorithm developed in the previous section can be applied to the continuous-time motion of robots in the domain. As introduced in Section 3.3.3, we allow the robots to perform a random walk motion in the domain. This random walk implies that the transitions between cells is stochastic in nature. Consequently, we model these transitions as a particular realization of a stationary Markov chain with transition probability matrix P, where the states of the chain are the different cells of the domain. At regular intervals of time $t_k = k\tau, k = 1, 2, \ldots$, the cell occupied by each robot is noted, and this information is used to generate the transition probability matrix P. Similar to the approach described in [124] and applied in [121], we compute a maximum likelihood estimate of the transition matrix, denoted as $\hat{P}$, by conducting repeated simulations of the random walk of the robots in the domain and noting the cell transitions made by the robots.

We have now obtained a discrete Markov chain, evaluated at discrete times $t_k, k = 1, 2, \ldots$, which is used as a simplified and abstract representation of the motion of the robots between the cells. Similar to Section 3.3, the states of the Markov chain are hidden since the robots do not know which cell they currently occupy. Let $q(t_k)$ represent the cell

of a chosen robot at time t_k—again, the index of the robot is suppressed. Let $c(t)$ denote a binary variable associated with the collision faced by the robot at time t,

$$c(t) = \begin{cases} 1 \text{ if the robot experiences collision at time } t, \\ 0 \text{ otherwise} \end{cases} . \tag{3.36}$$

Thus, the observations from the Markov chain can be redefined as,

$$\gamma(t_k) = \begin{cases} 1, \text{ if } \exists\, t \in [t_k, t_{k+1}) \;\; s.t. \;\; c(t) = 1, \\ 0, \text{ otherwise} \end{cases} . \tag{3.37}$$

As a result, $\gamma(t_k)$ encodes whether the robot experienced any collisions over the time interval $[t_k, t_{k+1})$ and can be used as an observation variable in the hidden Markov model. We now compute the observation probabilities $G_{lj}(t_k),\ l = 1, 2$, as defined in (3.25). Similar to Section 3.2, collisions occur between robots when their footprints overlap. In order to compute the collision probabilities, we draw parallels between our problem setup and kinetic theory of gases, where the frequency of collisions between randomly moving molecules has been rigorously analyzed, e.g., [98, 125].

For a robot with radius r, the collision cross-section is a circle of radius $2r$ centered at the robot (see Fig. 3.7). If we assume that all robots except robot i are stationary, the number of collisions experienced by robot i in a time duration τ, would be equal to the number of robots which lie in the area swept by the robot in the time interval τ, as illustrated in Fig. 3.7. This region can be approximated as a rectangle of area,

$$A_c = 2(2r)v\tau, \tag{3.38}$$

where v is the fixed speed of the robot. But, in order to adjust for the fact that all other robots are also moving with the same speed v, we replace the velocity magnitude v in (3.38) with the mean relative speed between the robots [98].

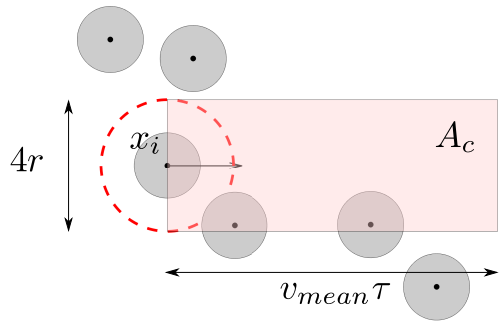


Figure 3.7: Illustration of the rationale behind the computation of continuous-time collision probabilities, inspired from the kinetic theory of gases. The number of collisions experienced by robot i in a time interval τ, is equal to the number of robots which happen to lie inside the area A_c swept by the robot, as illustrated. To account for the fact that all the other robots are also moving, we assume that robot i is moving with the relative mean speed between the robots, denoted as v_{mean}. This allows us to compute the probability that robot i will collide with another robot in a time interval τ.

Lemma 2. *Given a team of robots, which move in (uniformly) random directions, with a constant velocity magnitude v, the average relative speed between any given robot, and all other robots is,*

$$v_{mean} = \frac{4v}{\pi}. \tag{3.39}$$

Proof. Let $\vec{v_j}(t)$ represent the velocity vector corresponding to robot j at any given time t. Since the obtained result holds for all times, we suppress the time index for the rest of the proof.

For a given robot i, the relative mean speed, with respect to all other robots j, is given as

$$v_{mean} = \mathbb{E}[\sqrt{< \vec{v_{ij}}, \vec{v_{ij}} >}], \tag{3.40}$$

where $\vec{v_{ij}}$ = $\vec{v_i}$ - $\vec{v_j}$. This gives us,

$$v_{mean} = \mathbb{E}[\sqrt{< \vec{v_i} - \vec{v_j}, \vec{v_i} - \vec{v_j} >}] \tag{3.41}$$

$$v_{mean} = \mathbb{E}[\sqrt{< \vec{v_i}, \vec{v_i} > + < \vec{v_j}, \vec{v_j} > -2 < \vec{v_i}, \vec{v_j} >}] \tag{3.42}$$

$$v_{mean} = \sqrt{2}v\mathbb{E}[\sqrt{1 - \cos\theta}]. \tag{3.43}$$

Here, θ represents the angle between two uncorrelated random vectors $\vec{v_i}$ and $\vec{v_j}$, and thus, is a uniform random variable in the range $[0, 2\pi)$. Taking the mean over this range, we get the desired result.

□

This result holds for all robots, since they are all identical, and it does not matter which robot we pick. So, the effective area swept by a chosen robot in time τ is now given as,

$$A_c = 4r\frac{4}{\pi}v\tau. \tag{3.44}$$

Consequently, under this model, the number of collisions experienced by any robot in a time interval τ is the number of robots which lie in the corresponding collision area A_c. The probability of collision experienced by any robot in cell D_j, in any time horizon of length τ, is the probability that at least one other robot is present in an area A_c swept by the robot in the time horizon τ.

These constructions give us both the transition probability matrix P and the emission probability matrix G. As a result, we have constructed the hidden Markov model as $\mathcal{H} = (P, G, \bar{\mu})$. Proceeding as described in Section 3.3, we can utilize Algorithm 3 to obtain estimates $q^*(t_k)$ of the cell that the robot is occupying in the time interval $[t_k, t_{k+1})$.

It is worth noting that the steady-state distribution of robots in each cell will be directly impacted by how the domain is partitioned into the different cells. This in-turn has a direct impact on the performance of the localization algorithm. The more distinguishable the

steady-state robot densities within the different cells are from each other, the better the localization algorithm can be expected to perform. For example, if the domain is divided into a large number of similar looking smaller cells, the localization algorithm might not be able to distinguish between cells whose stationary density and physical accessibility are very similar to each other.

3.5 Experimental Results

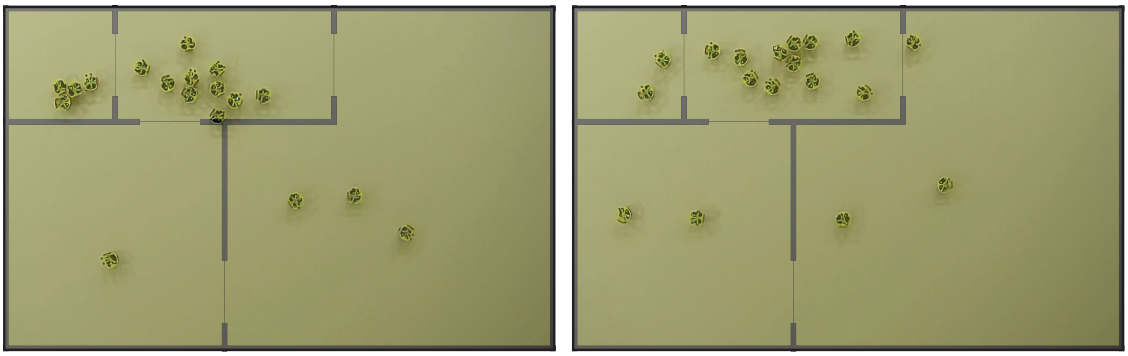
In this section, we deploy the collision-based localization algorithm on a team of real-world robots, and quantitatively analyze the accuracy of localization obtained. We consider a scenario where a team of robots are operating in an interconnected system of pipes. The domain is divided into 4 distinct cells, illustrated in Fig. 3.8. Each robot in the team independently determines which cell it is currently operating in using only collisions as a sensing mechanism.

As described in Section 3.4, the robots travel in straight lines until they collide with another robot or with the walls of the domain. The Markov transition matrix P corresponding to the motion of the robots is estimated by conducting extensive simulations of the robots operating in the domain.

3.5.1 Evaluation Methodology

In order to investigate the information content provided by collisions towards localization, we introduce a Markov filter which ignores collision measurements and generates estimates solely based on the state distribution of the Markov chain describing the motion of the robots (see Section 3.3). The Markov filter, denoted by $\mathcal{M}$, is defined as follows.

Definition 10. *Let P denote the known transition probability matrix corresponding to the motion of the robots, as described in Definition 9, and let $\mu = [\mu_1, \ldots, \mu_M]^T$ denote the initial distribution of the robots. Then, for each robot i, the Markov filter $\mathcal{M}$ generates cell estimates by picking a random cell based on the state distribution of the Markov chain at*



(a) (b)

Figure 3.8: Experimental Setup: 18 differential-drive mobile robots are traversing a 4-cell domain on the Robotarium—a remotely accessible multi-robot testbed [46]. The cells of the domain, shown as shaded polygonal areas, are interconnected, allowing the robots to move between them. Each robot executes the random walk algorithm outlined in Section 3.4 and by detecting and utilizing inter-robot collisions, generates estimates of the current cell that it is occupying. We qualitatively demonstrate that collisions among the robots are useful sources of information which can be used for localization.

time t_k*,*

$$q^*_{i,m}(t_k) \sim dist(\pi(t_k)), \tag{3.45}$$

where $\pi(t_k) = [\pi_1(t_k), \ldots, \pi_M(t_k)]^T$ *represents the state distribution of the Markov chain at time* t_k*. Under the mean-field approximation, for any robot* r*,*

$$\pi_j(t_k) = \textit{Prob}\left(q_r(t_k) = j\right), j \in \mathcal{M}, \tag{3.46}$$

where $\pi(t_k)$ *evolves according to* $\pi(t_k) = P\pi(t_{k-1})$ *with* $\pi(0) = \mu$*.*

Let $\mathcal{C}$ denote the collision-based localization filter (see Algorithm 3) developed in Section 3.3. The collision filter utilizes information from both the Markovian motion of the robots and inter-robot collisions to generate an estimate of the cell a robot is present in. On the other hand, the Markov filter $\mathcal{M}$ ignores collisional information and generates estimates only based on the Markovian motion of the robots. Thus, comparing the performance of the two filters provides an insight into the information content provided by inter-robot collisions and whether collisions help the robots localize better in the domain.

The following definition introduces a metric to help evaluate the performance of the collision filter $\mathcal{C}$ and compare it against the performance of the Markov filter $\mathcal{M}$.

Definition 11. *The non-negative localization error experienced by robot i at time t_k, represented as $d(q_i(t_k), q_i^*(t_k))$ is defined as the number of cell transitions that are required in order to reach robot i's true cell $q_i(t_k)$ from its cell estimate $q_i^*(t_k)$. Using this metric, the average localization error of the robot team at time t_k is given as,*

$$L(t_k) = \frac{1}{N} \sum_{i=1}^{N} d(q_i(t_k), q_i^*(t_k)), \tag{3.47}$$

where N is the number of robots in the swarm.

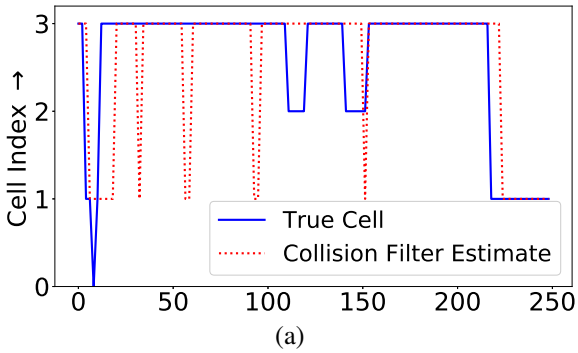
We now present the localization results for a team of real robots operating on a multi-robot testbed. The results are compared against the performance of the Markov filter $\mathcal{M}$.

3.5.2 Robot Experiments

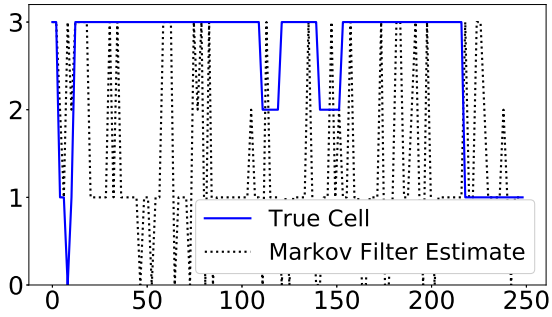
The probabilistic localization algorithm was implemented on actual, physical robots on the Robotarium [46], which is a remotely accessible swarm robotics testbed where code is uploaded via a web-interface and the experimental data is returned after the experiments. The experiment involved 18 differential-drive mobile robots traversing a 4-cell domain, which was overlayed onto the Robotarium arena floor (see Fig. 3.8). Position data of the robots was collected using an overhead tracking system. Robots were distributed among the cells according to an initial distribution at time $t = 0$.

Figure 3.9 depicts a sample path of a robot as it traverses the domain. As seen, the collision filter $\mathcal{C}$—which incorporates collision information at regular intervals of time—actively tracks the true cell of the robot, and performs significantly better than the Markov filter $\mathcal{M}$.

As seen in Fig. 3.10c, the average localization error $L(t_k)$ corresponding to the collision filter $\mathcal{C}$ decreases with time, as the robots move around the domain and use collisions to



(a)

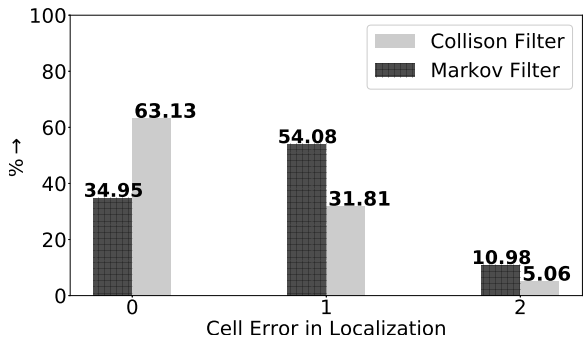


(b)

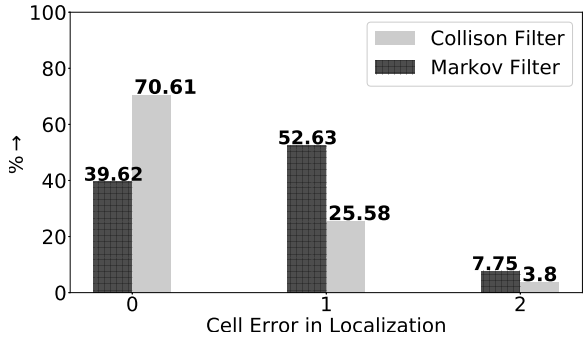
Figure 3.9: Performance of the collision filter $\mathcal{C}$ compared to the Markov filter $\mathcal{M}$ in a real-robot experiment. As seen, the tracking performance of the dotted line, denoting the estimate, is significantly better in Fig. 3.9a indicating that collisions are helping the robots localize better in the domain.

estimate their current location. In particular, the average localization error for the filter $\mathcal{C}$ drops below 1 cell within a few seconds.

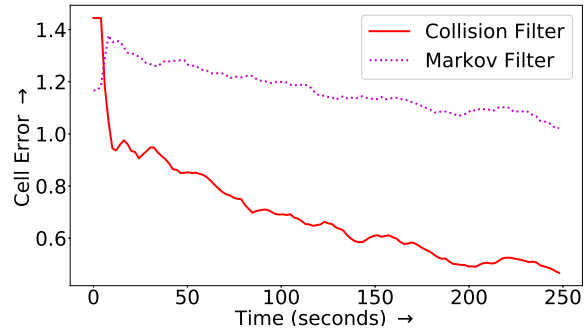
Figures 3.10a and 3.10b depict a histogram of the quantized localization error $d(q_i, q_i^*)$, as introduced in Definition 11, averaged over all the robots. The statistics in Fig. 3.10a considers the average errors over the entire duration of the experiment. As seen, the collision filter correctly predicts the true cell of the robot 63% of the time. Furthermore, if we only consider the data from the last 30% of the experiment, this performance measure improves to 70% indicating that additional collision information is helping the robots localize better as the experiment progresses (see Fig. 3.10b). In both cases, the percentage of correct predictions are higher compared to the baseline Markov filter. Figures 3.9 and 3.10 illustrate that collisions provide the robots with information which can be utilized for localization.



(a)



(b)



(c)

Figure 3.10: Quantitative analysis of the performance of the collision-based localization filter for a real-robot experiment consisting of 18 robots moving in a 4-cell domain for 250 seconds. Figures 3.10a and 3.10b illustrate the distribution of the quantized localization errors of the team for the entire duration of the experiment and the last 30% of the experiment respectively. As seen, the collision filter $\mathcal{C}$ correctly predicts the true cell of the robot more often than the Markov filter, and as shown by Fig. 3.10b, this performance improves as more collision information is incorporated into the estimate. Fig. 3.10c illustrates the average localization error $L(t_k)$, detailed in Definition 11. As the experiment progresses, the collision filter $\mathcal{C}$ performs significantly better than the Markov filter $\mathcal{M}$, illustrating that inter-robot collisions can indeed help robots localize in a densely-packed swarm of robots.

3.6 Conclusions

In this chapter, we took the first steps towards analyzing inter-robot encounters in robot swarms, which is one of the primary goals of this thesis. We argued that for certain classes of robot swarms, physical collisions among robots can not only be considered as a feature, but can also be viewed as a sensing modality. To this end, we used ideas from stochastic geometry to compute the probability of a robot experiencing a collision at a given point in time. This model was developed in the context of a localization scheme, where a team of robots moved around a partitioned domain equipped with only binary tactile collision sensors, and used the information derived from these sensors to localize themselves in the domain.

CHAPTER 4
ENCOUNTER RATES AS INFORMATION SOURCES

The previous chapter illustrated how physical inter-robot collisions can act as a source of information which can be utilized by a robot swarm. In this chapter, we demonstrate that not just physical inter-robot collisions but also proximity encounters among the robots can act as a source of information. Furthermore, we develop a continuous-time characterization of the nature and frequency of encounters experienced by robots in the environment.

Since the analytical development of such an encounter model is heavily dependent on the motion of the robots, Section 4.1 specifies a motion model for the robots, which, although somewhat restrictive, allows us to make mathematical statements regarding encounters. Following this, Section 4.2 provides the formal definition of an inter-robot encounter, and then develops a continuous-time model to describe the rate of inter-robot encounters. In Section 4.3, we demonstrate how the developed encounter model enables an analytical characterization of the effects of spatial interference on a robot swarm performing a distributed collection task. In particular, our model analytically captures the sublinearity in the increase of overall productivity of the swarm with increasing density, which is precipitated by the increasing amount of time spent by robots avoiding collisions with each other as opposed to performing the task at hand. We develop a decentralized algorithm which allows the swarm to achieve an optimal density which represents a desired trade-off between team size and productivity.

4.1 Motion Model

The rate of encounters experienced by the robots will depend on their motion in the domain. In order to make concrete statements regarding inter-robot encounters, we first restrict the motion characteristics of the robots in the environment. Consider a swarm of N identi-

cal robots operating in a planar and compact domain $\mathcal{D} \subset \mathbb{R}^2$. The following definition characterizes the motion of the robots.

Definition 12. *For any given robot $i \in \{1, \ldots, N\}$, let $p_i(t) \in \mathcal{D}$ denote the position of the robot at time t. Then, for any given set $S \subseteq \mathcal{D}$, the average time spent by the robot in that set is equal to the measure of the set, i.e.,*

$$\lim_{T\to\infty} \int_{t=0}^{T} \frac{1}{T} \mathbb{1}\left(p_i(t) \in S\right)\, dt = \frac{|S|}{|\mathcal{D}|}, \quad \forall i \in \{1, \ldots, N\}, \tag{4.1}$$

where $\mathbb{1}$ is the indicator function and $|\mathcal{D}|$ denotes the area of domain $\mathcal{D}$. Such a uniformly ergodic trajectory of the robot implies that all regions of the domain are equally visited by the robot over long time horizons.

The restrictiveness of the uniform ergodicity assumption is justified by demonstrating that it allows us to systematically analyze inter-robot encounters which ultimately enables successful decentralized density regulation and task allocation schemes as outlined in this chapter and in Chapter 5. This ergodic model allows us to be general about the kinds of motion patterns that result from the execution of the task—the developed encounter model is applicable as long as the ergodicity assumptions are satisfied. Notably, the uniformly ergodic trajectories of the robots will imply that their density over the domain $\mathcal{D}$ at any given time is also uniform.

4.2 Modeling the Encounter Rate

Analyzing the frequency of encounters between robots by tracking the positions of all the robots at all times is a computationally intensive task. Consequently, similar to the approach taken in Chapter 3, we simplify this problem by performing a mean-field approximation. This allows us to replace the effect of all the other robots on a single robot with an averaged effect. Consequently, we study the encounter characteristics of a single robot—it does not matter which one since they are all identical under the mean-field approximation.

Let r denote the physical radius of each robot and $\delta > r$ denote the sensory radius of each robot. This is the maximum distance until which the robot can detect the presence of another robot (see Fig. 4.1). We define two robots i and j as having an *encounter* when each robot is detected within the other's sensing radius, i.e., $\|p_i - p_j\| < r + \delta$ where p_i represents the position of robot i. Upon encountering another robot, each robot executes a collision avoidance maneuver to resolve the encounter and continue along its trajectory. We assume that all robots traverse the environment with an average speed v.

As seen in Fig. 4.1, the effective "proximity" cross-section of each robot is a circle of radius $r + \delta$ centered at the robot. Similar to the ideas developed for the molecular analogue of our problem [125], if we assume that all the other robots are stationary, the number of encounters experienced by a robot in a time window T is the number of robots which happen to fall in the sensory area swept by the robot in time window T, given by

$$A_e(T) = 2(r + \delta)vT, \tag{4.2}$$

where v is the average speed of the robot (see Fig. 4.1). But, in order to account for the fact that all other robots are also moving with the same average speed v, we replace the velocity magnitude v in (4.2) with the mean relative speed between the robots, denoted as v_r, as defined in (3.39).

Consequently, the expected number of encounters experienced by any robot in unit time, denoted by Ω, is the number of robots which are expected to lie in a region of area $A_e(1)$:

$$\Omega(\lambda) = A_e(1)\lambda = 2(r + \delta)v_r\lambda = 2(r + \delta)\frac{4}{\pi}v\lambda, \tag{4.3}$$

where $\lambda = N/|\mathcal{D}|$ is the density of the robots in the domain $\mathcal{D}$, and $|\mathcal{D}|$ denotes the area of the domain.

While Ω tells us how many encounters a robot can expect, it does not tell us anything about when the encounters occur. Luckily for us, models of inter-collision times in gaseous

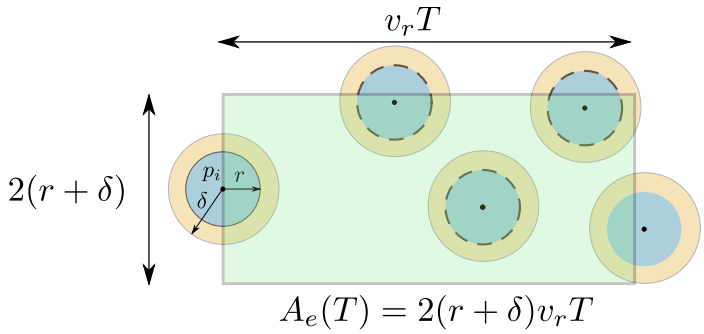


Figure 4.1: The sensory encounter cross-section area swept by robot i in a time window T. Any robot whose center falls within this region will experience an encounter with robot i. Such robots are marked by the dashed circular outlines. Consequently, the expected number of encounters experienced by robot i in the time window T will be equal to the expected number of robots which happen to fall in this green shaded region, whose area is given by $A_e(T)$. To account for the fact that all robots are moving, the average speed v of robot i is replaced with the mean relative speed v_r.

systems have been extensively studied, e.g., [34, 98]. Directly applying methods from [34], the following observation derives the distribution of a random variable denoting the time between encounters experienced by a robot in the swarm.

Observation 1 ([34])**.** *Let $\tau(\lambda)$ be the random variable denoting the time duration between encountering robots in a swarm of density λ within domain $\mathcal{D}$. Under the uniformly ergodic motion model of the robots, and using the assumption that inter-robot encounters are independent events (similar to [126]), $\tau(\lambda)$ can be modeled as an exponentially distributed random variable with mean,*

$$\mathbb{E}[\tau(\lambda)] = \frac{1}{\Omega(\lambda)}, \tag{4.4}$$

where Ω is the expected number of encounters in unit time, given by (4.3)*.*

Proof. Under the conditions mentioned in the statement of the observation, a robot can stay encounter free for a time duration $t + \delta t$ if it stays encounter free for a time duration t and does not encounter another robot in the time δt,

$$\text{Prob}(\tau(\lambda) \geq t + \delta t) = \text{Prob}(\tau(\lambda) \geq t)\ \text{Prob}(\text{no encounter in time } \delta t). \tag{4.5}$$

For convenience of notation, we denote Prob$(\tau(\lambda) \geq t) = P_\lambda(t)$. Note that (4.3) represents the *encounter rate* of the swarm, i.e.,

$$\Omega(\lambda)\delta t = \text{Prob(collision in a small time interval } \delta t). \tag{4.6}$$

Plugging this into (4.5), we get,

$$P_\lambda(t + \delta t) = P_\lambda(t)(1 - \Omega(\lambda)\delta t). \tag{4.7}$$

This gives us,

$$P_\lambda(t) + \frac{dP_\lambda}{dt}(t)\delta t = P_\lambda(t)(1 - \Omega(\lambda)\delta t), \tag{4.8}$$

$$\frac{1}{P_\lambda(t)}\frac{dP_\lambda}{dt}(t) = -\Omega(\lambda). \tag{4.9}$$

Solving this differential equation, and applying the boundary condition $P_\lambda(0) = 1$, we get,

$$\text{Prob}(\tau(\lambda) \geq t) = e^{-\Omega(\lambda)t}. \tag{4.10}$$

This proves Observation 1.

□

We have now obtained an analytical expression for the expected rate of inter-robot encounters as well as a characterization of the time-interval between encounters among a team of robots obeying a specific motion model. In the next section, we consider a motivating application which illustrates how the developed encounter model can help analytically characterize the effect of inter-robot spatial interference on the productivity of a robot swarm.

4.3 Motivating Application: Interference Reduction Using Encounters

As discussed in Section 2.1, swarm robotics systems are being increasingly deployed in dynamic real-world environments without the need for a central coordinator overseeing the mission. As the size of these swarms increases, robots tend to spend more time and effort avoiding collisions between each other, thereby hurting the overall performance of the swarm [21]. This phenomenon—an inevitable consequence of the competition for physical space—is called multi-robot interference and has been studied in a wide number of contexts (see Section 2.1 for a review of relevant literature on this subject).

In the context of multi-robot *foraging* [35, 55] and *collection* tasks [36, 37], it has been shown that while adding more robots to the swarm at intermediate robot densities increases the total number of objects collected, the increase in performance is sub-linear—caused by a significant decrease in individual robot performance, e.g., [56, 65]. Owing to the increased operational cost of deploying large robot teams, one would like to obtain the desired overall productivity while utilizing as few robots as possible [38]. Achieving such a trade-off is especially challenging in situations where relatively simplistic robots operate in dynamic environments without a central coordinator.

For a swarm of robots executing a *distributed collection* task, this section develops a decentralized algorithm which enables individual robots to decide between staying in the region and contributing to the task, or retreating from the region when the negative effects of interference outweigh the increased productivity of a larger swarm. In order to allow individual robots to make these decisions, we first develop an analytical model which describes the fraction of time that a robot spends performing the primary task as opposed to avoiding collisions with other robots. This allows us to compute an optimal robot density that minimizes a cost function in order to achieve a trade-off between the deployed team size and the overall performance of the swarm.

We envision a team of robots collecting objects from randomly scattered "pick-up"

points and depositing them at "drop-off" points. The locations of these points are unknown to the robots. Such a distributed collection task is of relevance to multi-robot applications such as search & rescue, mine-clearing, moving waste to recycling stations, transporting passengers to destinations, etc. Furthermore, we assume (i) a constant influx of robots which join the collection task, and (ii) no explicit communication between the robots, thus inspiring the need for a dynamic, yet decentralized mechanism to regulate interference in the domain.

As discussed in Section 2.2, entomological studies have shown that many social insect colonies possess the ability to regulate interference using decentralized techniques, e.g. [127, 128]. In [11], the authors study the ability of *Solenopsis Invicta* ants to regulate densities in narrow tunnels and prevent the formation of flow-stopping clogs. This is partially attributed to the tendency of individual ants to reverse out of crowded tunnels thus reducing the density. Crucially, it has been shown that ants use inter-ant encounters as the primary sensory mechanism to regulate density and perform functions like task allocation and division of labor [26].

Inspired by these observations from biology, *we allow robots to use local inter-robot encounter measurements as the only sensory mechanism to regulate interference*. Using the analytical encounter model developed in Section 4.2, we allow robots to measure the density of the swarm by counting interactions. The rest of this chapter is organized as follows. Section 4.3.1 introduces the distributed collection task and characterizes the rate of encountering objects and other robots in the domain. Section 4.3.2 leverages these analytical models to compute the optimal density of robots for achieving a desired trade-off between the size of the swarm and the overall performance. In Section 4.3.3, we develop an algorithm which allows each robot to leave the domain if the measured density of robots in the swarm is above the optimal value. In Section 4.3.4, the algorithm is deployed on a team of real robots. Section 4.4 concludes the chapter.

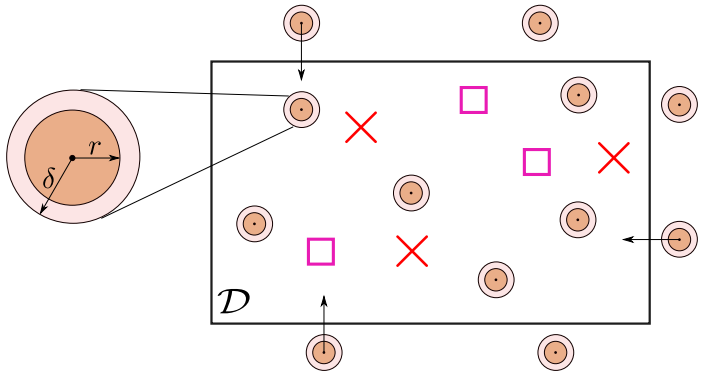


Figure 4.2: Distributed Collection Task: A team of robots pick up objects from randomly scattered pick-up locations (denoted by $\square$) and deliver them to drop-off locations (denoted by $\times$). An object can be delivered to any drop-off location. Pick-up and drop-off locations are uniformly and independently distributed with densities λ_p and λ_d, respectively. New robots are stationed outside the domain and enter the domain at a constant rate λ_{in}, symbolized by the arrows.

4.3.1 Problem Setup

We consider a homogeneous group of robots—each with a circular footprint of radius r and a sensing skirt of radius $\delta \geq r$ around it to detect objects and other robots (see Fig. 4.2). Let v denote the average speed of the robots as they move through the domain.

As illustrated in Fig. 4.2, the domain $\mathcal{D} \subset \mathbb{R}^2$ contains a set of randomly, but uniformly distributed pick-up locations (marked as $\square$) and drop-off locations (marked as $\times$) from where robots can pick up objects and deliver them, respectively. Each robot picks up or drops off an object if it gets within the sensing distance δ of the corresponding location.

We consider the scenario where picked up objects do not have a specific destination but can be dropped off at any of the drop-off locations. A robot picks up only one object at a time, and delivers it to a destination before picking up another object. We assume that the positions of the pick-up and drop-off points are unknown to the robots, but they know the corresponding non-zero uniform densities, denoted as λ_p and λ_d. Furthermore, new robots initially outside domain $\mathcal{D}$ join the collection task at a constant rate λ_{in}, as illustrated in Fig. 4.2.

Encounter Models

As robots explore the domain, they cross paths with each other. Similar to the encounter model described in Section 4.2, we define two robots i and j as having an "encounter" when they come within each other's sensing radius, i.e., when $\|p_i - p_j\| \leq r + \delta$ where p_i represents the location of robot i. At this point, each robot executes a collision avoidance maneuver (e.g., [129, 130]) to resolve the encounter and continues on its intended path. There are two relevant questions which arise in this context: how often does a given robot encounter other robots in the domain and how long does it take to resolve an encounter? Since the models concerning the first question were developed in Section 4.2, we focus on the second question next.

When robots encounter each other, they execute collision avoidance maneuvers. Let τ_r be the random variable which denotes the time it takes for a robot to resolve an encounter with another robot. τ_r will depend on a variety of factors such as the collision avoidance maneuvers used by the robots, velocities before collision, presence of domain boundaries, congestion, etc. The uncertainty associated with performing multiple sub-tasks in robotic/vehicular systems has been modeled using sums of exponential random variables [79, 131]. Similar to these formulations, we model τ_r as an exponentially distributed random variable with a parameter ρ that will vary based on the type of robots and collision avoidance mechanisms used,

$$\tau_r \sim exp(\rho). \tag{4.11}$$

In practice, the parameter ρ can be estimated by fitting the empirical encounter resolution times to the distribution in (4.11). Later in the section, we illustrate using simulations that τ_r is indeed well-approximated by an exponential distribution for a given collision avoidance algorithm.

Object Pick-Up and Drop-Off Rates

Let λ_p and λ_d denote the uniform density of object pick-up and drop-off locations in the domain, respectively. As the robots move through the domain, they pick-up/drop-off objects when they enter within a sensing distance δ of the respective location. Similar to the analysis done in Section 4.2, the expected number of locations encountered per unit time will be,

$$\Omega_p(\lambda_p) = 2\delta v \lambda_p \ ; \ \Omega_d(\lambda_d) = 2\delta v \lambda_d, \tag{4.12}$$

where Ω_p and Ω_d correspond to the expectations for pick-up and drop-off locations, respectively. Furthermore, since transport locations are uniformly random, the time elapsed between encountering two such locations can be modeled as an exponentially distributed random variable. Let τ_p and τ_d denote the time between encountering two successive pick-up and drop-off locations, respectively,

$$\tau_p(\lambda_p) \sim exp(\Omega_p(\lambda_p)) \ ; \ \tau_d(\lambda_d) \sim exp(\Omega_d(\lambda_d)). \tag{4.13}$$

The above developed models play a crucial role in the development of a decentralized mechanism to regulate interference, which is the ultimate goal of this section. We now introduce a simulation setup where the robots solve the distributed collection task while using a particular collision avoidance algorithm.

Simulation Setup

We validate the theoretical encounter models developed in this chapter using a simulated team of differential-drive robots solving the distributed collection task while performing a uniformly ergodic random walk in the domain. The robots deploy minimally-invasive barrier certificates [130] to compute safe velocities when they encounter other robots in the domain.

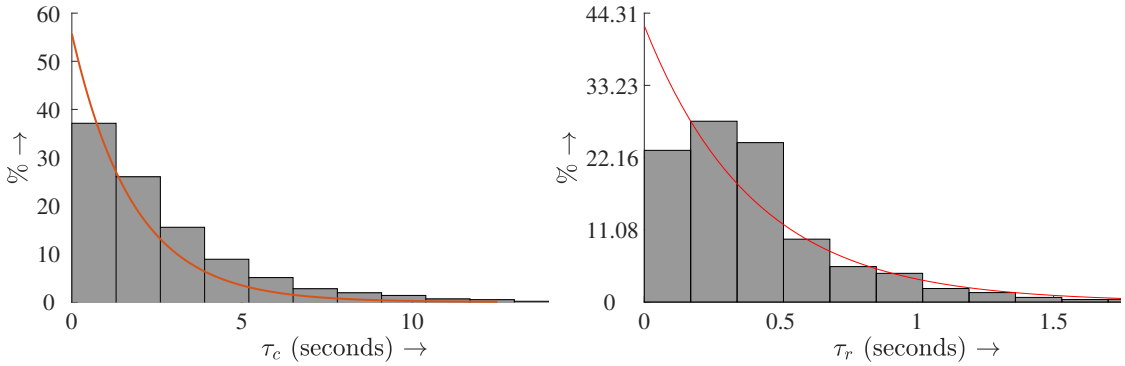


Figure 4.3: Histogram of the inter-encounter time τ_c and the encounter resolution time τ_r for a team of simulated robots performing the distributed collection task. In Fig. 4.3a, the inter-encounter time distribution is well-approximated by an exponential random variable with mean given in (4.2). Figure 4.3b shows the best-fit exponential distribution to the encounter resolution times. Best-fit exponential parameter: $\rho = 2.44$.

For a set of simulation parameters ($\lambda = 0.99$, $\delta = 0.11$, $v = 1$), Fig. 4.3a plots the histogram of the empirically obtained inter-encounter times, alongside the theoretical values corresponding to the same density given by (4.4). Figure 4.3b plots a histogram of the time taken by robots to resolve encounters and plots that against a best-fit exponential curve (MATLAB `fitdist`: $\rho = 2.44$). As seen in both cases, for the given choice of collision-avoidance algorithm, the empirical distributions are well-approximated by the developed encounter models.

4.3.2 Continuous-time Markov Chain

At a particular time t, any robot can be described as belonging to one of four distinct states defined by $\mathcal{V} = \{f, l, fe, le\}$: free and ready to pick-up an object (f), free and encountering another robot (fe), carrying an object (l), and experiencing an inter-robot encounter while carrying an object (le). The possible transitions between states is illustrated in Fig. 4.4.

Owing to the stochastic nature of the motion of robots and distribution of transport locations, the transition of a robot from one state to another will be stochastic. Let $X(t)$ be the stochastic variable which represents the current state of a chosen robot at time t.

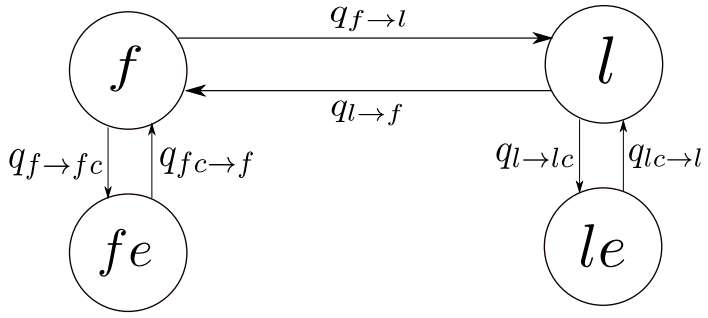


Figure 4.4: Each robot can be in four different states when performing the *collection* task: free (f), free and encountering another robot (fe), loaded with an object (l), loaded and encountering a robot (le). State transitions are characterized by a continuous-time Markov chain whose transition rates are illustrated.

It does not matter which robot we pick since they are all identical under the mean-field approximation (see Section 4.2). The memoryless properties of the inter-robot encounter and object-transport models developed in (4.4), (4.11), and (4.13) imply that the stochastic variable $X(t)$ evolves according to a continuous-time Markov process [132] over the possible states $\mathcal{V} = \{f, l, fe, le\}$.

The probability per unit time of making a transition from state $i \to j,\ i, j \in \mathcal{V}$ is characterized by the transition rate $q_{i\to j}$. Impossible transitions are characterized by $q_{i\to j} = 0$. The following theorem derives expressions for the transition rates and computes the fraction of time spent by each robot in the different states $\mathcal{V}$.

Theorem 2. *Consider a team of robots performing the distributed collection task as defined in Section 4.3.1. Let $X(t) \in \mathcal{V}$ be the stochastic variable denoting the current state of a given robot at time t. Utilizing the encounter as well as object transport models developed in* (4.4)*,* (4.11)*, and* (4.13)*, the fraction of time spent by each robot in a state $j \in \mathcal{V}$ can be computed by evaluating the unique stationary distributions*

$$\pi_j = \lim_{t\to\infty} Prob(X(t) = j), j \in \mathcal{V}, \tag{4.14}$$

whose expressions are given as,

$$\pi_f = \frac{\lambda_d/\lambda_p}{(1+\lambda_d/\lambda_p)\left(1+\frac{c\lambda}{\rho}\right)}, \tag{4.15}$$

$$\pi_l = \frac{\lambda_p}{\lambda_d}\pi_f \ ; \ \pi_{fe} = \frac{c\lambda}{\rho}\pi_f \ ; \ \pi_{le} = \frac{c\lambda}{\rho}\pi_l, \tag{4.16}$$

where $c = 2(r+\delta)v_r$.

Proof. The transition rates characterizing the CTMC illustrated in Fig. 4.4 can be defined as follows.

- $q_{f\to fe}$ and $q_{l\to le}$ represent the rates at which a free (or loaded) robot encounters another robot in the domain, which is given by (4.3).
- $q_{fe\to f}$ and $q_{le\to l}$ represent the rates at which a robot resolves an encounter when it is either free or loaded. From (4.11), this is simply the parameter ρ.
- $q_{f\to l}$ and $q_{l\to f}$ represent the rate at which a robot picks up objects and drops them off, respectively, given by (4.12).

Given the transition rates $q_{i\to j}, i, j \in \mathcal{V}$, we can compute the stationary distribution of the CTMC by constructing the generator matrix G [132], which is given as,

$$G = \begin{bmatrix} -(c\lambda + m\lambda_p) & m\lambda_p & c\lambda & 0 \\ m\lambda_d & -(c\lambda + m\lambda_d) & 0 & c\lambda \\ \rho & 0 & -\rho & 0 \\ 0 & \rho & 0 & -\rho \end{bmatrix}, \tag{4.17}$$

where $c = 2(r+\delta)v_r$ and $m = 2\delta v$. The stationary distribution $\pi = (\pi_f, \pi_l, \pi_{fe}, \pi_{le})^T$ can be computed by solving the equation $\pi^T G = 0$. The uniqueness of the stationary distribution π follows from the irreducibility and positive-recurrence properties of the embedded discrete-time Markov chain corresponding to the CTMC illustrated in Fig. 4.4. □

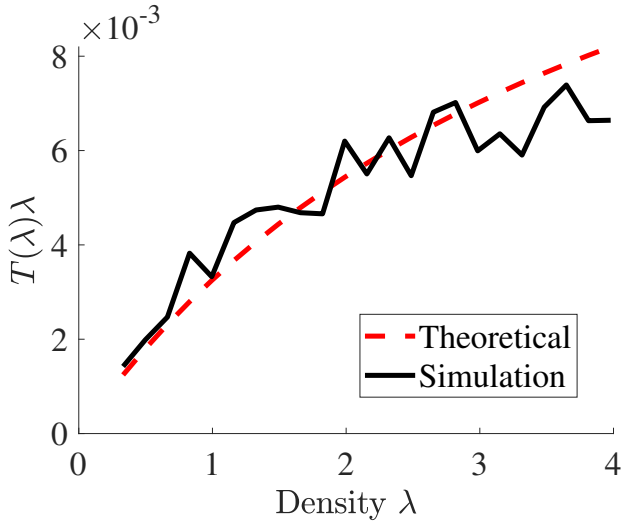


Figure 4.5: The theoretical evolution of $T(\lambda)\lambda$ as the density of robots (measured in robots/m^2) increases is closely matched with the values obtained from simulations with robots employing barrier certificates for collision avoidance. As seen, beyond a certain density, the total output of the swarm increases sub-linearly with the robot density since individual robots spend more time avoiding collisions than transporting objects. Here, we are primarily concerned with the performance of the swarm at intermediate densities—the catastrophic jamming occurring at higher densities is not discussed.

Since $q_{l\to f}$ represents the rate at which robots deliver objects and become free, the expected number of objects delivered per robot per unit time, denoted by T, is given as,

$$T(\lambda) = q_{l\to f}\pi_l = 2\delta v \lambda_d \pi_l(\lambda). \tag{4.18}$$

For the simulation setup introduced in Section 4.3.1, Fig. 4.5 plots the total number of objects delivered by the robots per unit time per unit area $T(\lambda)\lambda$, against the density λ (parameters: $v = 0.1$, $\delta = 0.11$, $\lambda_p = 0.82$, $\lambda_d = 0.66$, $\rho = 2.44$). As seen, the actual productivity of the swarm closely matches the theoretically expected output, barring deviations due to edge effects and the stochastic nature of inter-robot encounters. This section is primarily concerned with the performance of the swarm at intermediate densities—Fig. 4.5 does not consider very large densities where complete robot jamming can be expected, e.g., [64].

Inspired by the sub-linearity in the total output of the swarm at intermediate densities

(see Fig. 4.5), we would like to compute the optimal density of deployed robots so as to achieve a balance between the swarm size (characterized by λ) and the total output of the swarm (characterized by $T(\lambda)\lambda$). Such a trade-off can be encoded in the following optimization problem,

$$\underset{\lambda \geq 0}{\operatorname{argmin}} J(\lambda) = \frac{C}{T(\lambda)\lambda} + \lambda, \tag{4.19}$$

where λ is the density of robots deployed in the domain, and C is chosen to trade-off the cost of deploying robots and the effectiveness of the swarm at performing the task. Setting the gradient of $J(\lambda)$ to zero and solving analytically, we get the optimal density of robots,

$$\lambda^* = \underset{\lambda \geq 0}{\operatorname{argmin}} J(\lambda) = \sqrt{\frac{C(1 + \frac{\lambda_d}{\lambda_p})}{2\delta v \lambda_d}}. \tag{4.20}$$

In the next section, we allow a robot swarm to autonomously regulate its density to the optimal value by enabling individual robots to retreat from the domain when they sense a higher-than-optimal density.

4.3.3 Decentralized Density Control

The primary aim of this section is to develop an algorithm that allows individual robots to retreat from the domain when they detect a higher-than-optimal density. The first part of this section develops an ensemble-level control framework for reducing the robot density. A decentralized implementation is later discussed. We also develop an estimation algorithm that allows each robot in the domain to measure the time-varying density of robots $\lambda(t)$. We assume that robots enter and exit the domain by driving through any section of the boundary of domain $\mathcal{D}$ (as depicted in Fig. 4.2).

Ensemble Closed-Loop Control

Given the density of pick-up and drop-off locations λ_p and λ_d, if the density of robots at time t, denoted as $\lambda(t)$, is higher than the optimal λ^*, we would like a certain fraction of

the robots to leave the domain (we ignore the degenerate case when $\lambda^* = 0$). Let $\epsilon\lambda$ be the number of robots leaving per unit area per unit time. Then, the population dynamics in the domain can be expressed as,

$$\dot{\lambda}(t) = \lambda_{in} - \epsilon(\lambda(t))\lambda(t), \tag{4.21}$$

where λ_{in} denotes the known and constant number of robots added to the swarm per unit area per unit time (as discussed in Section 4.3.1). The following choice of ϵ feedback-linearizes the above system while driving the density of the swarm to λ^* using a "one-sided" proportional controller,

$$\epsilon(\lambda(t)) = \begin{cases} \frac{1}{\lambda(t)}[\lambda_{in} - K_p(\lambda^* - \lambda(t))], & \lambda > \lambda^* \\ 0 & , \text{ otherwise} \end{cases}, \tag{4.22}$$

where K_p is an appropriately chosen proportional gain. A proportional controller was chosen primarily as a simple demonstration of the control mechanism; in general, based on the desired performance requirements, any controller can be used. Applying (4.22) in (4.21), we get the ensemble closed loop population dynamics,

$$\dot{\lambda}(t) \equiv \begin{cases} K_p(\lambda^* - \lambda(t)), & \lambda(t) > \lambda^* \\ \lambda_{in} & , \text{ otherwise} \end{cases}. \tag{4.23}$$

The control parameter $\epsilon(\lambda(t))$ can be interpreted as the probability per unit time with which a robot should leave the domain as a function of the density $\lambda(t)$. Thus, in order for each robot to individually compute ϵ, it must estimate the density of robots in the domain.

Estimation

The robots performing the distributed collection task are not equipped with sophisticated sensors, but can detect encounters with other robots as they move through the domain, as defined in Section 4.3.1. Let $\boldsymbol{e}_i(t)$ denote the total number of robots currently in an encounter with robot i. New encounters experienced by robot i can be characterized by an index variable,

$$H_i(t) = \begin{cases} 1, \text{if } \exists\ t_1 < t\ s.t.,\ \forall x \in [t_1, t),\ \boldsymbol{e}_i(x) < \boldsymbol{e}_i(t) \\ 0,\ \text{otherwise} \end{cases} . \tag{4.24}$$

Denote $y_{iL}(t)$ as the total number of encounters experienced by robot i in a time interval $[t-L, t]$, $t \geq L$,

$$y_{iL}(t) = \int_{t-L}^{L} H_i(t)dt. \tag{4.25}$$

Given this measurement and the encounter model described by (4.4), the Maximum-likelihood Estimate (MLE) [133] of the swarm density for robot i at time $t > L$ is,

$$\hat{\lambda}_i(t) = \frac{y_{iL}(t)}{2(r+\delta)\frac{4}{\pi}vL}. \tag{4.26}$$

The measurement horizon L should be chosen so as to achieve a trade-off between accuracy of estimate and the ability of the robot to track changing densities.

Decentralized Implementation

Using the maximum likelihood density estimate $\hat{\lambda}_i(t)$, each robot computes its probability of leaving the domain $\epsilon(\hat{\lambda}_i(t))$ using (4.22). The operations of each robot, executed every dt seconds in software, is illustrated in Algorithm 4.

Before deployment, each robot computes the optimal density λ^* using the density of transport locations and influx rate of robots. After an initial wait of L seconds (during

Algorithm 4 Voluntary Retreat Algorithm

1: Initialize $\hat{\lambda}_i = 0, k = 0$
2: Given λ_p, λ_d, λ_{in}, and C, compute λ^*
3: **for** each time $t = kdt$, $t > L$ **do**
4: Compute $y_{iL}(t)$ from (4.25) $\leftarrow$ *detecting encounters*
5: Compute $\hat{\lambda}_i(t)$ from (4.26)
6: Compute $\epsilon(\hat{\lambda}_i(t))$ from (4.22)
7: Leave the domain with probability $\epsilon(\hat{\lambda}_i(t))dt$
8: $k = k + 1$
9: **end for**

which time robots are collecting encounter information), each robot computes the MLE estimate in step 5 and the resulting ϵ in step 6. Finally, the robot flips a biased coin to leave the domain with a probability $\epsilon(\hat{\lambda}_i(t))dt$.

Each robot i will have a different estimate of the density; consequently, $\epsilon(\hat{\lambda}_i(t))$ will vary from one robot to another. While this implies that robots behave differently from each other, we demonstrate in the next section that, for a team of real world robots, the algorithm indeed allows the robots to regulate the density of the swarm to the optimal value.

4.3.4 Experimental Results

The developed algorithm was deployed on the Robotarium [46], a remotely accessible swarm robotics testbed. The experiment was initialized with 8 robots starting inside the elliptical domain (with $1.2m$ and $0.8m$ as the semi-major and semi-minor axis length) shown in Fig. 4.6a, deploying minimally-invasive barrier certificates [130] for collision avoidance. An additional 4 robots were initially placed outside and entered the domain at a steady rate. Pick-up ($\square$) and drop-off ($\times$) locations are marked on the domain.

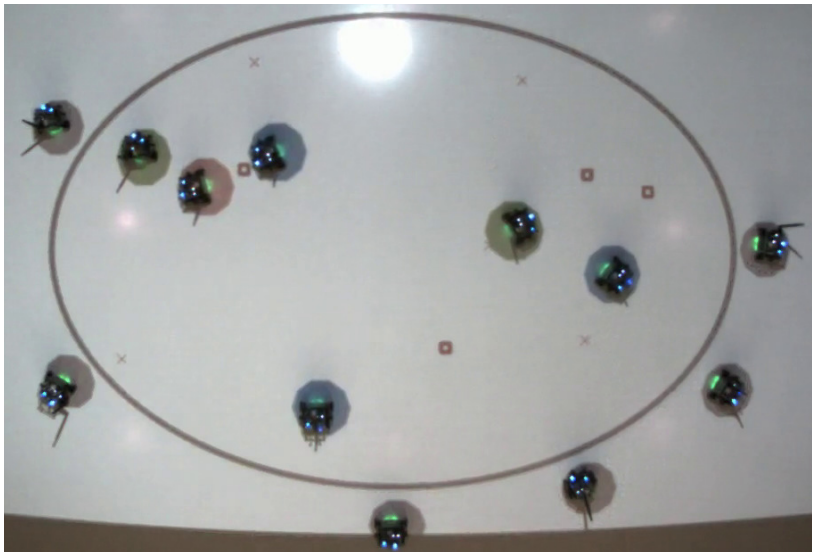
For the following parameters: $\lambda_p = 1.32$, $\lambda_d = 1.32$, $C = 0.03$, $\delta = 0.1$, $v = 0.1$, each robot computes the optimal swarm density using (4.20). The robots perform an unbiased random walk in the domain while encountering other robots. Using Algorithm 4 (with parameters $L = 90$, $dt = 0.03$, $K_p = 0.03$, $\lambda_{in} = 0.02$), each robot flips a biased coin to decide whether it should stay or retreat.

The experiment was repeated 5 times to average the results, and demonstrate the consistent performance of the voluntary retreat algorithm. Figure 4.6b plots the mean of the true robot population against the optimal value, along with the standard deviation (represented by the shaded region). As seen, the rate at which robots retreat reduces as the true population approaches the optimal value, in accordance with the closed-loop dynamics given in (4.23). Even with variations in the density estimates of the robots as well as the constant influx of new robots (indicated with upwards jumps in Fig. 4.6b), the closed loop algorithm allows the population to settle around the optimal value.

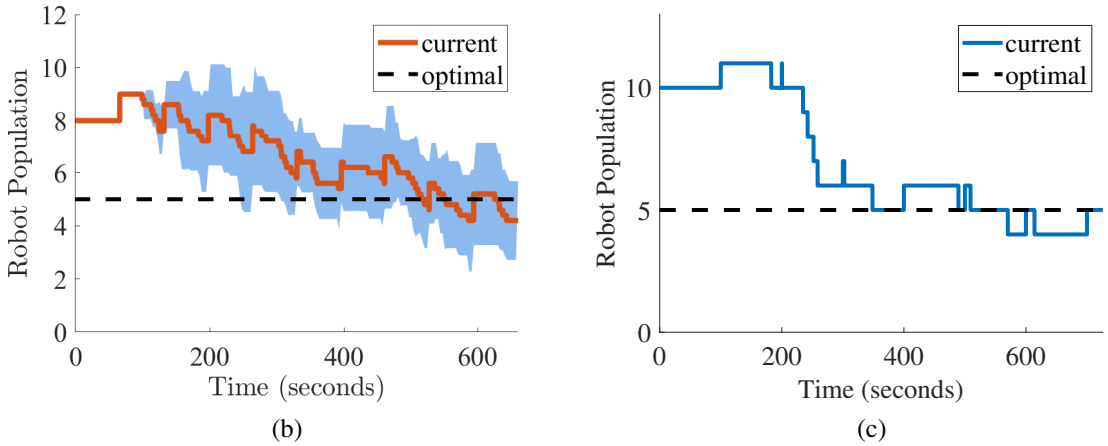
4.4 Conclusion

In this chapter we saw that the proximity encounters naturally occurring among robots operating in a given environment can help characterize the efficiency of the robot swarm at performing a distributed collection task, while also enabling the swarm to maximize this efficiency in a decentralized and communication-free manner. To this end, we allowed individual robots to estimate the current robot density using encounter measurements and voluntarily leave the domain if the density was higher than the optimal value.

Using ideas from the kinetic theory of gases, we saw the development of a higher-fidelity continuous-time encounter model as compared to the discrete-time version developed in Chapter 3. The uniform ergodicity assumption—upon which the validity of the encounter model hinges—allows us to remain general about the exact motion of the robots in the environment; the developed models work as long as the uniform ergodicity assumption is satisfied.



(a)



(b) (c)

Figure 4.6: Experimental verification of the Voluntary Retreat Algorithm on a team of 12 real robots operating on the Robotarium. An overhead projector is used to project additional information onto the robot arena. The experiment begins with 8 robots inside the elliptical domain seen in Fig. 4.6a. Robots stationed outside enter at a steady rate λ_{in}. Blue and red circles projected around each robot signify *free (f)* and *loaded (l)* robots, respectively. Green circles indicate that a robot has decided to retreat from the domain. These robots drive out of the elliptical domain via any section of the boundary. The robot population data in Fig. 4.6b is averaged over 5 experimental runs to demonstrate the consistent performance of the algorithm, with one standard deviation depicted by the shaded region. For a different set of initial conditions and simulation parameters, Fig. 4.6c highlights the ability of the Voluntary Retreat Algorithm to maintain the population of robots at the optimal value.

CHAPTER 5
ENCOUNTERS IN HIGH-DENSITY ROBOT SWARMS

The previous chapter saw the development of a continuous-time encounter model which characterized the expected rate of inter-robot encounters as a function of the robot size, speed and swarm density. Central to this model, was the notion of an effective encounter area swept by a given robot as it traversed the environment (see Fig. 4.1). Under the assumption that robots were uniformly distributed in a given region, we computed the expected number of encounters per unit time for each robot.

In this section we highlight how, due to the fact that individual robots have a finite footprint area, such an assumption may introduce inaccuracies in the encounter model—especially at higher robot densities. As also discussed in Chapter 3, such inaccuracies can negatively affect the performance of the encounter-based algorithms developed in this thesis, particularly in the context of densely packed swarms. In order to explicitly capture the fact that real robots have a finite footprint which cannot overlap with other robots, we use ideas from the *Enskog theory of high-density gases* [39] to analytically derive a correction factor accounting for the area occupied by each robot. The efficacy of this continuous-time encounter model is then demonstrated by developing an encounter-based task allocation algorithm where the swarm achieves a desired task distribution by switching between tasks when they encounter other robots.

5.1 Enksog Model

For a team of N robots moving in a planar and compact domain $\mathcal{D}$ with uniformly ergodic trajectories, Section 4.2 used ideas from the kinetic theory of gases to characterize the encounter rates experienced by the robots. However, as observed in many studies on the nature of inter-particle collisions (for e.g., see [134, 135]), the expression for the encounter

rates given in (4.3) is not accurate when the total area occupied by the robots is a significant fraction of the domain $\mathcal{D}$. This phenomenon is attributed to the fact that, in situations of high density, it is incorrect to assume that the local density of robots in the effective encounter area represented by A_e is λ (see Fig. 4.1). The following two factors contribute to a variation in the local density within A_e, as observed in [40]:

1. Given one robot, it is unlikely to find another robot center within a distance of $(r+\delta)$ since, as discussed in Chapter 4, the robots execute collision avoidance maneuvers when their sensors detect each other.

2. For a slightly larger separation distance than $(r+\delta)$, there is an increased probability of finding another robot (compared to examining a random section of the domain). This is due to the fact that two nearby robots encounter other robots from all sides except the sides facing each other, resulting in an effective "attraction" that tends to keep the robots from separating.

Since the encounter-based algorithms developed so far rely on some form of comparison between encounter measurements and what is predicted by the encounter model, an inaccurate model will affect the performance of such an algorithm.

In order to account for these inaccuracies, the molecular physics literature has introduced a multiplicative density correction factor $\mathcal{X}(\lambda) > 1$ [39], which theoretically models these variations in local density and can be used to address the same phenomena in a swarm of robots. Using this correction factor, the density λ in (4.3) would be replaced with $\mathcal{X}(\lambda)\lambda$. Note that the correction factor applies for the robot density measured in the small region with area A_e (as depicted in Fig. 4.1), and *not* for the average density in the entire domain $\mathcal{D}$.

The Enskog correction factor $\mathcal{X}(\lambda)$ represents the ratio of the local density of robots centers at the point of inter-robot contact, to the average density in the domain [135]. A crucial result shown in [136] allows the interpretation of $\mathcal{X}$ as the pair correlation func-

tion [123] for a pair of disks, evaluated just outside their point of contact. Given a set of points (representing robot centers in this thesis), the pair correlation function $g(d)$ evaluates the relative likelihood of finding a point at a distance d from a given point when compared to a completely uniform distribution of points.

Under this interpretation, we obtain an analytical expression for $\mathcal{X}(\lambda)$ by analyzing the spatial distribution of robots at any time instant t using ideas from stochastic geometry—the study of random point patterns and the properties that they satisfy (e.g. [33, 110]). Similar to the models developed in Chapter 3, we use hard-core point processes to capture the fact that the robots always maintain a distance $r + \delta$ with other robots (see Definition 3 for a formal description of hard-core point processes). Motivated by the empirical simulations presented in Section 3.3.3, we make the following observation about the instantaneous spatial distribution of robot centers in the domain.

Observation 2. *For a swarm of N robots operating in domain $\mathcal{D}$ under the uniformly ergodic motion model given in Section 4.1, the robot locations at any time t can be well-approximated by a Matern type-II hard-core point process [111] within domain $\mathcal{D}$ with density $\lambda = N/|\mathcal{D}|$, and an inhibition distance equal to the distance maintained between the robots, i.e., $\Delta = r + \delta$.*

The above observation allows for the derivation of an analytical expression for $\mathcal{X}(\lambda)$ as seen in the following theorem. Furthermore, when applied as a correction factor for computing mean inter-encounter times, it yields valid results compared to empirical estimates obtained from swarm robot simulations.

Theorem 3. *For a swarm of N robots each with a physical footprint of radius r and a sensory radius δ operating in a domain $\mathcal{D}$, variations in the local density of robots in the region represented by area A_e (see Fig. 4.1) can be modeled using the Enskog correction*

factor which is parameterized by the average density $\lambda = N/\mathcal{D}$, and is given as,

$$\mathcal{X}(\lambda) = \frac{2c}{(c-k_d)(1-e^{-\lambda_p c})}\left[1 - \frac{1-e^{-\lambda_p(2c-k_d)}}{(2c-k_d)(1-e^{-\lambda_p c})}c\right], \tag{5.1}$$

where

$$c = \pi(r+\delta)^2, \quad k_d = \left(\frac{8\pi}{3} - \sqrt{3}\right)(r+\delta)^2, \quad \lambda_p = \frac{1-e^{-\lambda c}}{c}. \tag{5.2}$$

Proof. Similar to [136], the Enskog correction factor can be interpreted as the pair correlation function between two robots evaluated just outside their point of contact. Utilizing Observation 2, the correction factor can be given as,

$$\mathcal{X}(\lambda) = \lim_{\mathrm{d}r \to 0} g_\lambda(r + \delta + \mathrm{d}r), \tag{5.3}$$

where $g_\lambda(\cdot)$ is the pair correlation function for a Matern type-II hard-core point process.

Plugging the expression for the pair correlation function of a Matern type-II hard core process from [123] into (5.3), we obtain the expression in (5.1). □

Theorem 3 allows us to redefine the expected number of encounters per unit time and the inter-encounter times to account for the non-zero area covered by the footprint of the robots. The corresponding corrected variables are now represented using bold characters $\boldsymbol{\Omega}$ and $\boldsymbol{\tau}$ respectively. Thus,

$$\boldsymbol{\Omega}(\lambda) = 2(r+\delta)\frac{4}{\pi} v \mathcal{X}(\lambda)\lambda, \tag{5.4}$$

and the time between encountering robots in the domain is an exponentially distributed random variable $\boldsymbol{\tau}(\lambda)$ with an adjusted mean

$$\mathbb{E}[\boldsymbol{\tau}(\lambda)] = \frac{1}{\boldsymbol{\Omega}(\lambda)}. \tag{5.5}$$

5.1.1 Simulation

In this section, we demonstrate the validity of the Enskog-corrected encounter model developed on a swarm of simulated robots operating in a rectangular domain $\mathcal{D}$. The planar motion of each robot is modeled using single-integrator dynamics:

$$\dot{x} = u, \tag{5.6}$$

where u is the control velocity applied to the robot. These dynamics are propagated forward using standard Euler integration techniques. As specified by the motion model in Section 4.1, the robots perform a uniformly ergodic random walk in the domain. More specifically, the robots travel in straight lines at constant speed until they encounter other robots or the walls of the domain. When an encounter is detected between robots, their velocity components along the axis of collision are exchanged. When robots encounter a wall, they turn around with a reflected velocity vector. For all the robots, the mean time between encountering other robots is measured.

For a set of simulation parameters ($v = 0.1, \delta = 0.012, r = 0.018$) and for a range of robot densities λ, Fig. 5.1 compares the mean of the inter-encounter time intervals predicted by the density-corrected version in (5.5) and the uncorrected version in (4.4) against the empirically measured mean of inter-encounter times experienced by the robots. As seen, inclusion of the Enskog correction factor in the calculations leads to a more accurate prediction of the true encounter rates experienced by the robots. The accuracy of the resulting model is higher than what is typical in molecular system simulations (see [135]).

5.2 Task Allocation Using Encounters

As a motivating application to highlight the salient features of the developed encounter model, this section formulates a stochastic task-switching policy which allows a swarm of robots to autonomously allocate themselves among a set of pre-defined tasks using inter-

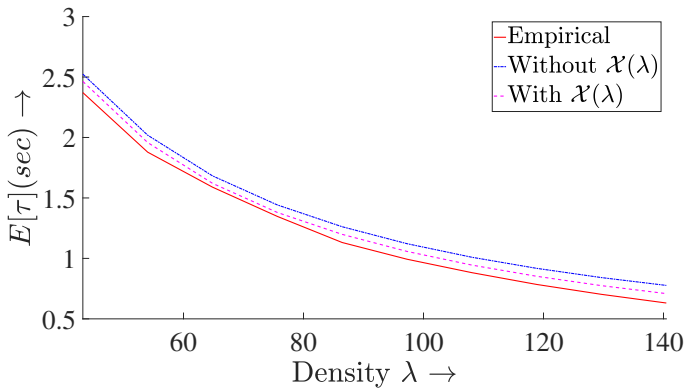


Figure 5.1: Comparison of empirically obtained estimates of mean inter-encounter times in a robot swarm of varying density against the predictions of two different theoretical models—with and without the Enskog correction factor $\mathcal{X}(\lambda)$ as given by (5.5) and (4.4), respectively. Each robot performs a constant-velocity uniformly ergodic random walk in the domain, while encountering other robots. For the simulation parameters ($v = 0.1, \delta = 0.012, r = 0.018$), the empirical inter-encounter times are obtained by averaging the encounter rates over all the robots in the swarm. As seen, the Enskog correction factor leads to a more accurate prediction of the true encounter rates experienced by the robots.

robot encounters as the trigger for switching between tasks. We assume that each robot has the sensory capability to detect an encounter as well as the current task assigned to the robot that it is encountering. This form of local information exchange is frequently seen in insect colonies [137], and can be realized in many ways, e.g., using near-field communication, color/infrared sensors, chemical signaling, etc.

In order to allow robots to transition from performing one task to another while only being equipped with such short-range parsimonious sensors, we allow some robots in the domain to be idle—they move around the domain but do not perform a designated task—instead they act as a *resource*. Idle robots can take on new tasks when they encounter robots performing active tasks and active robots can enter this idle state when they encounter other robots performing the same task. This task-transition mechanism was partially inspired by similar observations in ant-colonies [138], where idle ants take up tasks based on local interactions with other ants.

The transition of each robot between tasks is regulated via a set of probabilities which act as control knobs. More specifically, for each task, we assign two transition probabil-

ities: one which determines the affinity of the robot to take up the task and one which determines the affinity of the robot to stop performing the task and become idle. These transition probabilities affect the system performance in two ways—the ratio of the probabilities determine the final allocation of the swarm while their magnitudes determine the *rate* at which the swarm approaches a given allocation.

We now focus on the first of these two aspects—we illustrate how the ratio of the transition probabilities can be chosen to determine the equilibrium allocation levels achieved by the swarm. To that end, we derive the dynamics for the expected number of robots performing each task, and demonstrate how the desired equilibrium allocation levels can be attained by appropriately choosing the ratio of these transition probabilities.

5.2.1 Transition Rules

Consider a swarm of N robots operating in a planar and compact domain $\mathcal{D} \subset \mathbb{R}^2$. The swarm has to be allocated to K tasks denoted as $T_1, T_2, \ldots, T_K$ according to a desired allocation, which is specified by $\rho^* = [\rho_1^*, \ldots, \rho_K^*]$ where ρ_j^* represents the desired fraction of robots performing task T_j. At any given time, some robots in the swarm are not assigned to any task, but are simply moving around the domain, waiting to take up a task. We categorize these *idle* robots under the "task" T_0. The following definition specifies the rules according to which robots switch between tasks.

Definition 13. *The transition of each robot between tasks is triggered by encounters with other robots. An encounter between robot i and j is said to occur when they are detected within each other's sensing radius, i.e., when $\|p_i - p_j\| \leq r + \delta$, where p_i represents the position of robot i. At this point, each robot is able to detect an encounter as well as detect the current task allocated to the robot it is encountering. The task transitions occur according to the following rules:*

1. *When an idle robot i encounters an active robot j executing task T_m, $m \in \{1, \ldots, K\}$, robot i takes up task T_m with probability $0 \leq p_{m0} \leq 1$.*

2. *If robot i encounters robot j, and they are both assigned to the same task $T_m,\ m \in \{1, \ldots, K\}$, both robots choose to leave their tasks, independently of one another, and become idle with probability $0 < p_{0m} \leq 1$.*

3. *All other encounters result in no task switching.*

The control probabilities p_{m0} and p_{0m}, $m \in \{1, \ldots, K\}$ determine the task allocation levels as well as the rate of task-transitions occurring in the swarm. In particular, we will demonstrate how the *ratio* of the control probabilities p_{m0} and p_{0m} determines the desired allocation levels for task T_m. These ratios are transmitted to the robots prior to deployment. It should be noted that the presence of idle robots is necessary to allow the swarm to achieve a desired allocation ratio, i.e. the condition $p_{0m} > 0$ must hold for all $m \in \{1, \ldots, K\}$. Next, we study how the task allocation of the swarm evolves based on the transition probabilities and the encounters occurring among the robots.

5.2.2 Task Allocation Dynamics

As described above, task switching between the robots is triggered by inter-robot encounters. The switching is determined based on the task of the encountering robots and certain transition probabilities, as described in Definition 13. Since the occurrence of inter-robot encounters is stochastic in nature, the number of robots performing a task at any given time is a random number. However, by leveraging the encounter model developed in Section 5.1, we can derive dynamical equations for the expected number of robots performing a task at any given time.

Let $\boldsymbol{N_j}(t)$ represent the expected number of robots assigned to task $T_j,\ j \in \{0, \ldots, K\}$ at time t. Here, the bold character represents the expectation of a random variable. In this chapter, we assume that no robots enter or leave the collective, and hence, the total number of robots is conserved:

$$\sum_{j=0}^{K} \boldsymbol{N_j}(t) = N,\ \forall\, t \geq 0. \tag{5.7}$$

The expected density of robots performing each task is $\boldsymbol{\lambda_j}(t) = \boldsymbol{N_j}(t)/|\mathcal{D}|, j \in \{0, \ldots, K\}$. Denote $\bar{\lambda} = N/|\mathcal{D}|$ to be the total density of robots in the domain.

We assume that at time $t = 0$, the robots are randomly assigned some task $T_j,\ j \in \{0, \ldots, K\}$, according to an initial distribution $\mu = [\mu_0, \ldots, \mu_K]$ where $\sum_{j=0}^{K} \mu_j = 1$, and $\mu_j > 0, \forall j \in \{0, \ldots, K\}$ ensuring that each task is allocated to at least one robot within the swarm. Thus, the expected densities of robots assigned to each task at time $t = 0$ are,

$$\boldsymbol{\lambda_j}(0) = \mu_j \bar{\lambda},\ \forall j \in \{0, \ldots, K\}. \tag{5.8}$$

The following theorem derives a differential equation for the expected density of robots executing task T_j at any given time t, given the initial conditions in (5.8).

Theorem 4. *The evolution of the expected density of robots performing tasks $T_j,\ j \in \{1, \ldots, K\}$ is given by the system of differential equations*

$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{j}}(t) = 2(r + \delta)\frac{4}{\pi} v \mathcal{X}(\boldsymbol{\lambda_j}(t))\boldsymbol{\lambda_j}(t) p_{0j} \left[r_j \boldsymbol{\lambda_0}(t) - \boldsymbol{\lambda_j}(t)\right],\ \ j \in \{1, \ldots, K\} \tag{5.9}$$

where r_j is the ratio of control probabilities corresponding to task T_j:

$$r_j = \frac{p_{j0}}{p_{0j}}, \tag{5.10}$$

and the initial conditions are given by (5.8)*.*

Proof. According to the task transition rules described in Definition 13, the expected number of robots assigned to task $T_j, j \in \{1, \ldots, K\}$ at time $t + dt$ is given by the equation,

$$\boldsymbol{N_j}(t + dt) = \boldsymbol{N_j}(t) + \boldsymbol{N_{0 \to j}}(dt) - \boldsymbol{N_{j \to 0}}(dt),\ j \in \{1, \ldots, K\}, \tag{5.11}$$

where $\boldsymbol{N_{i \to j}}(dt)$ denotes the expected number of robots which switch from task T_i to task T_j in time dt.

At time t, under the mean-field approximation and the motion model developed in Section 4.1, the time duration between encountering two robots which are allocated to task T_j is characterized by the exponential random variable $\boldsymbol{\tau}(\boldsymbol{\lambda_j}(t)), \forall j \in \{0, \ldots, K\}$ (see Observation 1 and Theorem 3).

Applying the rules from Definition 13, $\forall j \in \{1, \ldots, K\}$, we get,

$$\boldsymbol{N_{0\to j}}(dt) = \text{Prob}(\boldsymbol{\tau}(\boldsymbol{\lambda_j}) \leq dt)\ p_{j0}\boldsymbol{N_0}(t), \tag{5.12}$$

$$\boldsymbol{N_{j\to 0}}(dt) = \text{Prob}(\boldsymbol{\tau}(\boldsymbol{\lambda_j}) \leq dt)\ p_{0j}\boldsymbol{N_j}(t). \tag{5.13}$$

Invoking (4.10) with the density-corrected terms from (5.4),

$$\text{Prob}(\boldsymbol{\tau}(\boldsymbol{\lambda_j}) \leq dt) = 1 - \exp(-\boldsymbol{\Omega}(\boldsymbol{\lambda_j})dt). \tag{5.14}$$

Substituting (5.12), (5.13), (5.14) in (5.11) we get,

$$\begin{aligned}\boldsymbol{N_j}(t+dt) - \boldsymbol{N_j}(t) = (1 - \exp(-\boldsymbol{\Omega}(\boldsymbol{\lambda_j})dt))p_{j0}\boldsymbol{N_0}(t) \\ - (1 - \exp(-\boldsymbol{\Omega}(\boldsymbol{\lambda_j})dt))p_{0j}\boldsymbol{N_j}(t).\end{aligned} \tag{5.15}$$

Dividing by dt and taking the limit $dt \to 0$,

$$\begin{aligned}\dot{\boldsymbol{N_j}}(t) = \lim_{dt\to 0} \frac{(1 - \exp(-\boldsymbol{\Omega}(\boldsymbol{\lambda_j})dt))}{dt} p_{j0}\boldsymbol{N_0}(t) \\ - \lim_{dt\to 0} \frac{(1 - \exp(-\boldsymbol{\Omega}(\boldsymbol{\lambda_j})dt))}{dt} p_{0j}\boldsymbol{N_j}(t),\end{aligned} \tag{5.16}$$

we get,

$$\dot{\boldsymbol{N_j}}(t) = \boldsymbol{\Omega}(\boldsymbol{\lambda_j})p_{j0}\boldsymbol{N_0}(t) - \boldsymbol{\Omega}(\boldsymbol{\lambda_j})p_{0j}\boldsymbol{N_j}(t). \tag{5.17}$$

Dividing this equation by $|\mathcal{D}|$, we get,

$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{j}}(t) = \boldsymbol{\Omega}(\boldsymbol{\lambda}_{\boldsymbol{j}})(p_{j0}\boldsymbol{\lambda}_{\mathbf{0}}(t) - p_{0j}\boldsymbol{\lambda}_{\boldsymbol{j}}(\boldsymbol{t})), \forall j \in \{1, \ldots, K\}. \tag{5.18}$$

Denoting the ratio of the control probabilities as in (5.10), we get

$$\dot{\boldsymbol{\lambda}}_{\boldsymbol{j}}(t) = \boldsymbol{\Omega}(\boldsymbol{\lambda}_{\boldsymbol{j}})p_{0j}\Big[r_j\boldsymbol{\lambda}_{\mathbf{0}}(t) - \boldsymbol{\lambda}_{\boldsymbol{j}}(t)\Big], \tag{5.19}$$

which proves Theorem 4. □

Note that, since the total number of robots is conserved, the expected density of idle robots is simply given by

$$\boldsymbol{\lambda}_{\mathbf{0}}(\boldsymbol{t}) = \bar{\lambda} - \sum_{j=1}^{K} \boldsymbol{\lambda}_{\boldsymbol{j}}(\boldsymbol{t}), \tag{5.20}$$

where $\bar{\lambda} = N/|\mathcal{D}|$. Equation 5.9 describes the evolution of the expected density of robots performing each task under the developed task allocation paradigm. By deriving the equilibrium points of this dynamical system, and analyzing their stability, we can allow the swarm to achieve any desired task allocation ratio by modifying the transition probabilities.

5.2.3 Equilibrium Analysis

We now derive the equilibrium densities for the dynamical system presented in Theorem 4. This allows us to choose the transition probabilities described in Definition 13 based on a desired task allocation ρ^*.

Theorem 5. *The task allocation algorithm is said to have reached an equilibrium allocation, represented by expected equilibrium densities* $\boldsymbol{\lambda}_{\boldsymbol{j,eq}}$*, when* $\dot{\boldsymbol{\lambda}}_{\boldsymbol{j}}(t) = 0, \forall j \in \{0, \ldots, K\}$*. If the system has achieved equilibrium allocation, then for each* $j \in \{1, \ldots, K\}$*,*

$$\boldsymbol{\lambda}_{\boldsymbol{j,eq}} = 0, \quad or \quad \boldsymbol{\lambda}_{\boldsymbol{j,eq}} = r_j\boldsymbol{\lambda}_{\mathbf{0,eq}} \tag{5.21}$$

where r_j is given by (5.10)*. Furthermore, the first equilibrium point in* (5.21) *is unstable and the second one is locally asymptotically stable.*

Proof. For an active task $j \in \{1, \ldots, K\}$, setting the derivative in (5.17) to zero, we get,

$$\mathcal{X}(\boldsymbol{\lambda_{j,eq}})\boldsymbol{\lambda_{j,eq}}\Big[p_{j0}\boldsymbol{\lambda_{0,eq}} - p_{0j}\boldsymbol{\lambda_{j,eq}}\Big] = 0, \tag{5.22}$$

giving us the equilibrium points in (5.21).

Next, we analyze the stability of the equilibrium points, by linearizing the non-linear dynamics in (5.17) around the equilibrium points, and analyzing the stability of the obtained linear system. For the system of dynamical equations $\dot{\lambda}_j = f(\lambda_j)$, a small perturbation from the equilibrium can be approximated as,

$$f(\boldsymbol{\lambda_{j,eq}} + \epsilon) = f(\boldsymbol{\lambda_{j,eq}}) + \frac{\partial f}{\partial \lambda_j}(\boldsymbol{\lambda_{j,eq}})\epsilon. \tag{5.23}$$

Since the Enskog correction factor $\mathcal{X}(\lambda_j) > 1$ varies very slowly with λ_j (see [123]),

$$\frac{\partial \mathcal{X}}{\partial \lambda_j}(\lambda_j) \approx 0, \tag{5.24}$$

we treat the term $2(r+\delta)v_r\mathcal{X}(\boldsymbol{\lambda_{j,eq}}) > 0$ as a constant denoted by $\bar{c}$. Then,

$$\frac{\partial f}{\partial \lambda_j}(\boldsymbol{\lambda_{j,eq}}) = \bar{c}\Big[p_{j0}\boldsymbol{\lambda_{0,eq}} - 2p_{0j}\boldsymbol{\lambda_{j,eq}}\Big]. \tag{5.25}$$

As discussed in Section 5.2.1, the presence of idle robots is necessary for the proper functioning of the developed method, i.e., $\lambda_{0,eq} > 0$. Thus, for the first equilibrium point, we get,

$$\frac{\partial f}{\partial \lambda_j}(0) = \bar{c}p_{j0}\boldsymbol{\lambda_{0,eq}} > 0, \tag{5.26}$$

implying instability, and for the second equilibrium point we get,

$$\frac{\partial f}{\partial \lambda_j}\Big(\frac{p_{j0}}{p_{0j}}\boldsymbol{\lambda}_{\mathbf{0},\boldsymbol{eq}}\Big) = -\bar{c}p_{j0}\boldsymbol{\lambda}_{\mathbf{0},\boldsymbol{eq}} < 0, \tag{5.27}$$

implying local asymptotic stability. This proves the theorem. □

The second equilibrium point in (5.21) demonstrates the relation via which we can achieve a desired task allocation of the swarm by appropriately selecting the ratio of the control probabilities p_{0j} and $p_{j0}, j \in \{1, \ldots, K\}$.

5.2.4 Simulation Results

We first demonstrate the performance of the developed task allocation strategy on a swarm of simulated robots operating in a rectangular domain $\mathcal{D}$. The robots are to be allocated among three different tasks T_1, T_2, and T_3, with the following distribution: $\rho^* = [0.3, 0.1, 0.4]^T$. This implies that the remaining 20% of the robots are expected to remain idle at any given time, waiting to take up tasks. Each task satisfies the uniform ergodicity conditions laid out in Chapter 4.1. When robots encounter each other, they deploy a collision avoidance algorithm to resolve the encounter and continue along their intended trajectories. We use control barrier certificates [130] as the collision avoidance algorithm since they minimally alter the control inputs of the robots to ensure safety and can be executed in real time.

Given the desired allocation specifications ρ^* and the total density of robots in the domain $\bar{\lambda}$, we can express the desired density of robots performing active tasks and remaining

idle (indicated by task T_0),

$$\begin{bmatrix} \boldsymbol{\lambda_{0,des}} \\ \boldsymbol{\lambda_{1,des}} \\ \boldsymbol{\lambda_{2,des}} \\ \boldsymbol{\lambda_{3,des}} \end{bmatrix} = \begin{bmatrix} 1 - \mathbf{1}^T \rho^* \\ \rho^* \end{bmatrix} \bar{\lambda}, \tag{5.28}$$

where $\mathbf{1}$ represents a vector of ones. In this equation, $\boldsymbol{\lambda_{0,des}}$ is the expected density of robots that are idle in the swarm and are acting like a resource.

We can now use (5.21) to compute the desired ratio of the control probabilities

$$r_j = \frac{p_{j0}}{p_{0j}} = \frac{\boldsymbol{\lambda_{j,des}}}{\boldsymbol{\lambda_{0,des}}}, j = 1, 2, 3. \tag{5.29}$$

These ratios corresponding to the desired allocation are transmitted to all the robots prior to deployment. As an example, each robot can then compute the control probabilities as follows: let $p_{0j} = 0.9$, and compute $p_{j0} = r_j p_{0j},\ j = 1, 2, 3$. In the next section, we demonstrate how each robot can select appropriate values for the control probabilities to regulate the *rate* at which robots switch between different tasks.

Once deployed, the robots achieve the desired distribution among the tasks by using inter-robot encounters as the trigger for switching between tasks and simply applying the rules presented in Definition 13. For a chosen set of parameters $\delta = 0.02, r = 1.5\delta, v = 0.1$, Fig. 5.2 shows the evolution of the task allocation ratios for a swarm with density $\bar{\lambda} = 50$ robots/m^2. On average, the swarm achieves the allocation ratios predicted by the theoretical curves plotted alongside (see Theorem 4). It is worth noting that the simulation results in Fig. 5.2 correspond to a single simulation run.

In order to illustrate the self-regulatory nature of this algorithm, Fig. 5.3 considers a scenario where 40% of the robots currently allocated to task T_3 suddenly experience failure. At a certain point in time, these robots are removed from the simulation. This

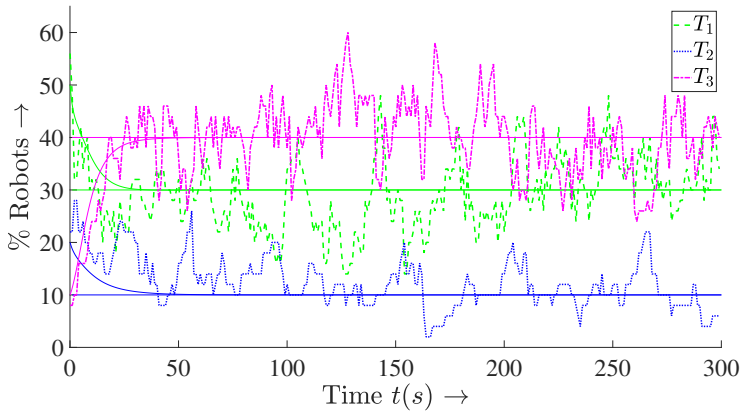
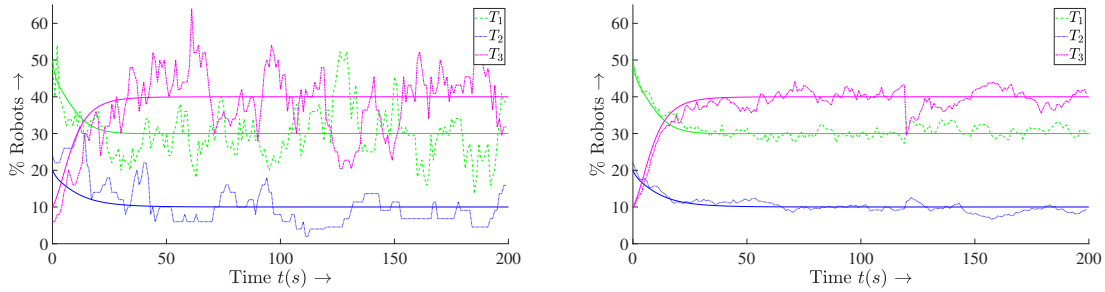


Figure 5.2: Performance of the developed task allocation algorithm in a swarm of simulated robots executing three different tasks. For a desired allocation of robots performing active tasks $\rho^* = [0.3, 0.1, 0.4]^T$ and parameters $\bar{\lambda} = 50$ robots/m^2, $\delta = 0.02, r = 1.5\delta, v = 0.1$, the solid lines depict the expectation of the allocation percentages as predicted by Theorem 4, and the three different dashed lines depict the percentage of robots allocated to each task as the simulation progresses. As seen, the swarm achieves the desired allocation, with an expectation predicted by the theoretical curve. The illustrated graphs correspond to a single simulation run.

implies that the allocation ratios are disturbed. Fig. 5.3a illustrates the performance of the task allocation algorithm for one simulation run, and Fig. 5.3b averages the performance of the swarm over 20 simulations to better highlight the response of the allocation algorithm to the failure. As seen, the swarm autonomously regulates the task allocation back to the designed levels, and is robust to changes in the number of robots operating at any given time, as long as a high enough population is maintained. While Fig. 5.3 illustrates the average performance of the task allocation algorithm over 20 simulation runs, it does not provide any information about the distribution of the performance over individual runs. To this end, Fig. 5.4 presents errors bars corresponding to one standard deviation of the data collected over 20 simulation runs. These are overlayed on the graphs for the average allocation of robots in each task.



(a) 1 simulation run

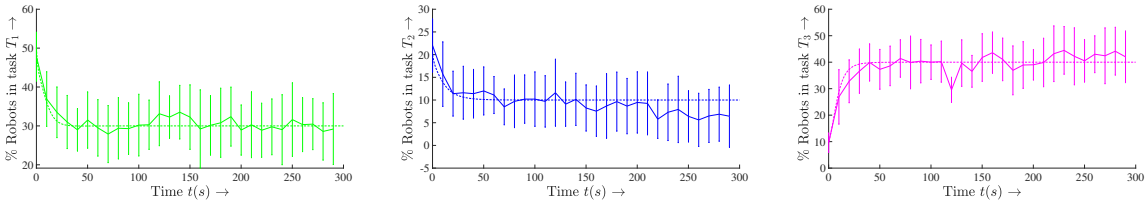
(b) 20 simulation runs

Figure 5.3: Robustness of the developed encounter-based task allocation policy to individual robot failures. At time $t = 120s$, 40% of the robots in task T_3 experience failure, and are removed from the simulation. As seen in both Fig. 5.3a and Fig. 5.3b, this disrupts the task allocation of the swarm. The swarm then autonomously regulates the allocation ratios back to the desired levels demonstrating that this strategy is robust to changes in the number of robots. While Fig. 5.3a illustrates the performance of the swarm over one simulation run, Fig. 5.3b averages the allocation over 20 simulations to highlight the effect of the swarm failure at time $t = 120s$.

5.2.5 Adaptive Transition Rates

So far, we developed a decentralized task allocation algorithm which allowed the robot swarm to divide itself among a set of tasks using inter-robot encounters as a trigger for switching between tasks. Since no robot knows the global allocation state of the swarm, robots continue to switch between tasks—as a function of the steady rate of inter-robot encounters—even after the desired allocation has been reached. Consequently, this constant task-switching reduces the effectiveness of the robots at performing any one task. For example, frequent switching between a transportation task and a surveillance task might render the robot ineffective at performing either function. Partially, this behavior is attributed to the *open-loop* nature of the task allocation algorithm—robots switch between tasks regardless of the current state of the allocation.

We now demonstrate that inter-robot encounters can enable the robots to *measure* the current allocation of the swarm and enable a closed-loop control of the rate of task transitions. We develop a feedback mechanism which allows each robot to measure the current allocation of the swarm and then regulate the transition rates p_{j0} and p_{0j} while maintaining



(a) Errorbars for task T_1 (b) Errorbars for task T_2 (c) Errorbars for task T_3

Figure 5.4: Standard deviations of the swarm task allocation in the presence of individual robot failures. The error bars in Fig. 5.4a-5.4c correspond to one standard deviation (on each side) of the task allocation levels collected over 20 simulation runs for tasks T_1, T_2, and T_3, respectively. These error bars are overlayed on the averaged task allocation values which were presented in Fig. 5.3b.

a constant allocation ratio r_j (see (5.10)). Thus, the robots can regulate the ingress and egress from each task without affecting the overall task allocation specifications. Such a mechanism allows the swarm to prioritize uninterrupted task execution when the desired allocation levels have been reached—by reducing the task-transition probabilities.

On the other hand, when the current allocation is far from the desired, this feedback mechanism can allow the robots to transition between tasks at a higher rate, thus helping the swarm achieve the desired task allocation faster. This also helps the disturbance rejection capabilities of the algorithm, by allowing the swarm to increase task transition rates in response to a detected deviation in the allocation levels. By adaptively varying the task-transition rates as a function of the measured task allocation of the swarm, we achieve a faster response of the task allocation algorithm as well as better steady state tracking of the desired task allocation.

The first part of this section develops an ensemble level controller which regulates the rate of task transitions occurring in the swarm as a function of the current allocation. A decentralized implementation of the controller is later discussed, for which we develop an estimation algorithm to allow each robot to measure the current allocation of the swarm using inter-robot encounters.

Swarm-level Control

Let $\rho(t) = [\rho_1, \ldots, \rho_K]$ represent the task allocation ratios of the robot swarm at a given time t:

$$\rho_j(t) = \frac{\lambda_j(t)}{\bar{\lambda}},\ j \in \{1, \ldots, K\}. \tag{5.30}$$

Given a desired allocation ρ^*, it would be desirable to increase the rate of transitions between tasks when $\rho(t)$ is farther away from ρ^* and reduce it when $\rho(t)$ is closer to ρ^*. One way to achieve this behavior is to use a proportional controller which modifies the rate according to the error in the allocation. However, instead of modifying the rate linearly with the Euclidean distance between the true and the desired allocation, we use a logistic function mapping [139], $\mathcal{F} : [0, 1] \to \mathbb{R}_+$, defined as,

$$\mathcal{F}(x) = \frac{L}{1 + e^{-k(x-m)}} \tag{5.31}$$

where $L > 0$ denotes the maximum value achieved by the function, $m \in [0, 1]$ denotes the x-axis point at which the function achieves its mid-point value, and k modifies the slope of the curve (see Fig. 5.5). The logistic curve formulation allows for a high-degree of flexibility in adjusting the sensitivity of the controller to noise in the error as well as deciding the response characteristic of the controller itself.

Using the logistic functional, we define a *logistic-proportional* controller for each task T_j, $j \in \{1, \ldots, K\}$ as,

$$p_{0j} = \mathcal{F}_j(|\rho_j^* - \rho_j|), j \in \{1, \ldots, K\}, \tag{5.32}$$

with parameters L_j, m_j and k_j which can be adjusted for each task. Once the value of p_{0j} has been selected, the transition probability p_{j0} can be simply computed using (5.10):

$$p_{j0} = r_j p_{0j}. \tag{5.33}$$

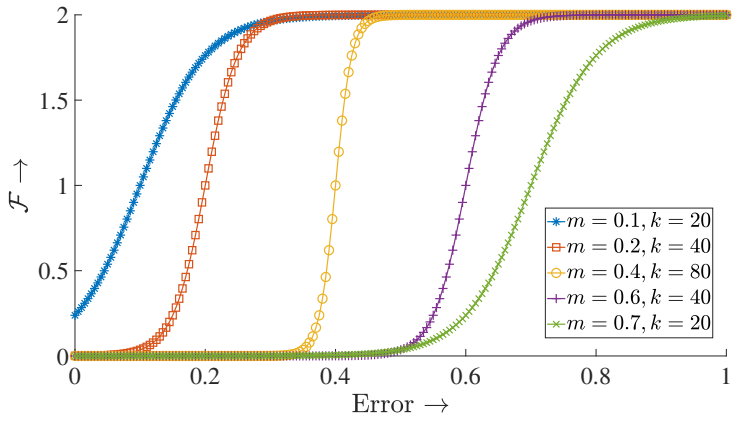


Figure 5.5: The logistic function mapping $\mathcal{F}$ described in (5.31), for a varying set of k, m parameter values and $L = 2$. This mapping is used to specify the response of the controller which modifies the task transition probabilities of the robots based on the difference between the desired and the true allocation ratios. The parameter L modifies the maximum value achieved by the function, m signifies the x-axis point at which the mid-point value of the function is attained, and k modifies the maximum slope of the curve. This gives a high degree of flexibility in designing the response of the controller which can be adjusted based on the size of the swarm, task requirements, noise levels in the allocation estimates etc.

From (5.33), the following relation holds, and ensures that p_{0j} and p_{j0} are valid transition probabilities satisfying the conditions given in Definition 13:

$$p_{0j} \in \left[0, min(\frac{1}{r_j}, 1)\right] \implies p_{j0} \in [0, 1]. \tag{5.34}$$

For (5.34) to hold true, the parameter L_j in the logistic-proportional controller given in (5.32) is constrained as,

$$L_j \in \left[0, min(\frac{1}{r_j}, 1)\right],\ j \in \{1, \ldots, K\}. \tag{5.35}$$

Note that a proportional controller was used primarily for simplicity; in general, more complex control algorithms can be used to regulate the transition probabilities.

Estimating Current Allocation

Executing the control law given by (5.32) requires the robots to have knowledge of the current allocation ratios of the swarm. We now develop an estimation algorithm which

allows individual robots to use the mean-field dynamics given in (5.9) and measurements made via inter-robot encounters to generate estimates of the current allocation of the swarm. This ultimately facilitates a decentralized implementation of the control law given in (5.32).

The robots are interested in estimating $\lambda(t) = [\lambda_0(t), \ldots, \lambda_K(t)]$ which represents the true density of the robots allocated to each task at time t. Estimating the density of robots allocated to each task will allow the robots to estimate the allocation ratios directly. Let $\lambda_i^*(t) = [\lambda_{i,0}^*(t), \ldots, \lambda_{i,K}^*(t)]$ denote the estimate generated by robot i at time t. Furthermore, let $p_{0j}^i(t)$ and $p_{j0}^i(t)$, $j \in \{1, \ldots, K\}$ represent the transition probabilities obtained by robot i from applying the controller in (5.32) as a function of the allocation ratios obtained from $\lambda_i^*(t)$.

As described in the beginning of this section, individual robots can measure the task allocated to the robots they encounter in the domain. Since inter-robot encounters occur in a random fashion, these measurements are inherently stochastic in nature (see Section 5.1). Additionally, the robots know the non-linear mean-field dynamics obeyed by the density of robots executing each task, as specified in (5.9). One way to estimate a variable of interest—especially in the presence of non-linear dynamics and measurement noise which does not obey a Gaussian distribution—is to use a *particle filter*, e.g., [140, 141]. A particle filter approximates the posteriori distribution of interest using a set of weighted samples. At each iteration, the particles are updated based on a given prediction model and are assigned weights based on the measurements.

Each robot i deploys a particle filter to track the density of robots performing tasks $T_j, j \in \{0, \ldots, K\}$. For the particle filter corresponding to robot i, let $\big(\hat{\lambda}_i^h \in \mathbb{R}^{K+1}, w_i^h \in [0, 1]\big)$ be a tuple representing the h^{th} particle in the filter, where $h = \{1, \ldots, H\}$. A *particle* is represented by the sample estimate $\hat{\lambda}_i^h = [\hat{\lambda}_{i,0}^h, \hat{\lambda}_{i,1}^h, \ldots, \hat{\lambda}_{i,K}^h]^T$ and its corresponding weight w_i^h which is a relative measure of the importance of the particle. Here, H denotes the total number of particles in the filter. Computing an estimate for the density of robots allocated to each task can be broken down into two phases: *predict* and *update*. In the

predict phase, the particles evolve according to the model given by (5.9). This is followed by the *update* phase which uses the encounter measurements to update the weights of the different particles.

Owing to the stochastic nature of the task allocation algorithm, the true density of robots allocated to each task will vary from the expected densities given by the mean-field model in (5.9). Thus, in the *predict* phase, we model the evolution of the true robot densities by adding a noise term to (5.9). For robot i, the j^{th} component of particle $h \in \{1, \ldots, H\}$ evolves as

$$\dot{\hat{\lambda}}^h_{i,j}(t) = \Omega(\hat{\lambda}^h_{i,j}(t))p^i_{0j}(t)[r_j\hat{\lambda}_{i,0}(t) - \hat{\lambda}^h_{i,j}(t)] + n(t), j \in \{1, \ldots, K\}, \tag{5.36}$$

where $n(t)$ is a zero-mean noise signal representing the deviation of the true intensities from the mean-field model, and similar to (5.20), the density $\hat{\lambda}^h_{i,0}$ evolves as,

$$\dot{\hat{\lambda}}^h_{i,0}(t) = -\sum_{j=1}^{K} \dot{\hat{\lambda}}^h_{i,j}(t). \tag{5.37}$$

In Section 5.2.5, we make specific design choices for the distribution and variance of $n(t)$, but for now, we simply assume that a noise model is available to us. Note that while the transition probabilities $p^i_{0j}(t)$ and $p^i_{j0}(t)$ are chosen by robot i at each point in time, their ratio, represented by r_j is fixed before deployment and remains constant for all tasks $T_1, \ldots, T_K$.

As mentioned earlier, robots encounter other robots as they traverse the domain and are able to measure the current task allocated to the robots they are encountering. For robot i, let $e^j_i(t)$ be the binary variable denoting an encounter with another robot which is currently

allocated to task T_j:

$$e_i^j(t) = \begin{cases} 1, \text{if robot } i \text{ encounters a robot allocated to task } T_j \\ 0, \text{otherwise} \end{cases} \tag{5.38}$$

For robot i, let $y_i^j(t)$ denote the total number of encounters with robots executing task T_j over the time interval $[t-T, t], t \geq T$:

$$y_i^j(t) = \int_{t-T}^{t} e_i^j(t)dt. \tag{5.39}$$

Thus, via encounters, each robot obtains the measurement vector $y_i(t) = [y_i^0, y_i^1, \ldots, y_i^M]$ representing the encounters experienced over the time interval $[t-T, t]$, $t \geq T$. Given the encounter model developed in Section 5.1, the probability that robot i makes a measurement $y_i^j(t)$ given the density $\hat{\lambda}_{i,j}^h(t)$ represented by particle $h \in \{1, \ldots, H\}$ is,

$$\mathcal{L}_{i,j}^h(t) = Prob(y_i^j(t)|\hat{\lambda}_{i,j}^h(t)) = \frac{\Omega(\hat{\lambda}_{i,j}^h(t)T)^{y_i^j(t)} exp\big(-\Omega(\hat{\lambda}_{i,j}^h(t)T)\big)}{y_i^j(t)!}. \tag{5.40}$$

These posteriori probabilities are used to update the weights corresponding to each particle in the *update* phase.

Algorithm 5 illustrates the operations of the particle filter on each robot executed every dt seconds in software. The step-by-step descriptions of the different operations are as follows. In Step 1, the particles are initialized with the density of robots allocated to each task at time $t = 0$, as given by (5.8). For each particle, a noise term is added to the density estimate to spread out the particles. The noise term also represents stochasticity in the allocation process. All particles are initially assigned an equal weight of $1/H$. As the robots move around performing tasks, Step 5 illustrates how each robot accumulates encounter measurements over a time window T. As specified by the *predict* phase, the density estimates represented by the particles are updated in Step 8 according to the dy-

Algorithm 5 Particle Filter for Estimating Current Allocation – Robot i

1: Initialize H particles: $\hat{\lambda}_i^h(0) = \lambda_0 + n(0)$ and the weights $w_i^h(0) = 1/H$, $h = \{1, \ldots, H\}$.
2: Iteration variable: $k = 1$
3: **for** time $t = kdt$ **do**
4: **if** $t > T$ **then**
5: Compute $y_i^j(t)$ from (5.39) $\forall j \in \{0, \ldots, K\} \leftarrow measurements$
6: **end if**
7: **for** each particle $h \in \{1, \ldots, H\}$ **do**
8: Compute $\hat{\lambda}_{i,j}^h(t)$ using (5.36) and (5.37), $\forall j \in \{0, \ldots, K\} \leftarrow predict$
9: **if** $t > T$ **then**
10: Compute $w_i^h(t) = w_i^h(t - dt) \prod_{m=0}^{K} \mathcal{L}_{i,m}^h(t - dt)$ using (5.40) $\leftarrow update$
11: **end if**
12: **end for**
13: Normalize weights so that $\sum_{h=1}^{H} w_i^h(t) = 1$
14: Best particle: $h_i^*(t) = \text{argmax}_{h \in \{1,\ldots,H\}} w_i^h(t)$
15: Estimate for density of robots allocated to task $T_0, \ldots, T_K$: $\lambda_i^*(t) = \hat{\lambda}_i^{h_i^*}(t)$
16: k = k + 1
17: **end for**

namical model given in (5.36), along with the added noise term. Following this, in the *update* phase, the weights corresponding to each particle are updated using the encounter measurements stored in $y_i(t)$ (see Step 10). More specifically, the particle weights are recursively multiplied with the likelihood of making encounter measurements corresponding to each of the density estimates $\hat{\lambda}_{i,j}^h(t)$, $j \in \{0, \ldots, K\}$. Robot i's estimate for the density of robots allocated to each task at time t, denoted as $\lambda_i^*(t)$, is computed in step 15 as the value of the particle corresponding to the highest weight. The measurement time horizon T is chosen so as to achieve a trade-off between accuracy of measurements and the ability of the filter to track time-varying densities of robots in each task. Furthermore, to tackle issues of sample degeneracy, common to particle filters [142], we implement a re-sampling strategy to draw new particles at regular intervals of time.

As mentioned above, given an estimate of the density of robots performing different tasks $\lambda_i^*(t)$, each robot $i \in \{1, \ldots, N\}$ computes a local version of the control probabilities $p_{j0}^i(t)$ and $p_{0j}^i(t)$ using the controller described in (5.32). These are used by the robots

to switch roles when they encounter other robots in the swarm. In the next section, we illustrate the efficacy of the proposed rate-controlled task allocation algorithm.

Simulations

We now illustrate the deployment of the feedback-based task allocation algorithm on a swarm of simulated robots operating on a rectangular domain $\mathcal{D}$. The robots are to be divided among three tasks (denoted as T_1, T_2, T_3) with specifications $\rho^* = [0.3; 0.1; 0.4]^T$, each of which involves the robots executing a uniformly ergodic trajectory in the domain. This implies that 20% of the robots are expected to be idle at any time, waiting to take up a task. The total density of robots is $\bar{\lambda} = 50$ robots/m^2. Similar to Section 5.2.4, the robots deploy minimally-invasive barrier certificates [130] to avoid collisions between each other.

The desired ratio of control probabilities, computed using (5.28) and (5.29) is transmitted to all the robots before deployment. Once the swarm is deployed, each robot uses inter-robot encounters to generate estimates of the current allocation of the swarm using Algorithm 5, and computes the control probabilities p_{j0} and p_{0j} using the logistic-proportional controller given by (5.32) while ensuring that the ratio of the control probabilities r_j remains constant. The process noise $n(t)$ is modeled as a Gaussian signal with zero-mean and a variance experimentally chosen according to the total density of robots in the domain. The parameters of the logistic-proportional controller are chosen based on the noise-levels in the estimates, the requirements on the controller performance, and size of the swarm.

For the simulation parameters ($\delta = 0.02, r = 1.5\delta, v = 0.1, k_j = 50, m_j = 0.16, L_j = \frac{1}{r_j}, j \in \{1, \ldots, K\}$), Fig. 5.6 illustrates the performance of the task allocation algorithm with the encounter-based time-varying transition rates. As the true allocation of the swarm approaches the desired levels, the number of task-transitions per robot per unit time occurring in the swarm reduces as illustrated in Fig. 5.6b.

Next, we demonstrate the adaptive nature of the task-transition rates during individual robot failures occurring in the swarm. As before, at time $t = 200s$, 40% of the robots

currently performing task T_3 experience a failure and are removed from the simulation. As seen in Fig. 5.7a, the swarm regulates the task allocation levels back to the desired values. Furthermore, following the failure at time $t = 200s$, the closed-loop rate controllers on the robots increase the task-transition rate temporarily so as to re-attain the desired allocation levels, as seen in Fig. 5.7b. Note that the plots in Fig. 5.7 have been averaged over 5 simulation runs to better illustrate the effect of the failure.

For the same simulation, Fig. 5.8 illustrates the performance of the particle filters during and after the individual robot failures occur. The estimates of all the particle filters on the robots have been averaged (solid lines) and plotted against the true allocation (three different dashed lines). The estimates track the true values illustrating that the inter-robot encounters allow the robots to obtain reliable estimates of the current allocation levels in the swarm. This in turn enables the closed-loop control of task transition rates.

5.2.6 Experimental Results

The closed-loop task allocation algorithm was implemented on the Robotarium, which is a remotely accessible swarm robotics testbed where code is uploaded via a web-interface and experimental data is returned after the experiments [46]. As seen in Fig. 5.9, the robot swarm consists of 12 robots which are to be allocated among two different tasks T_1 and T_2 according to the specifications $\rho^* = [0.5, 0.25]^T$. The rest of the robots are idle, waiting to take up an active task, as specified in Section 5.2.1. A colored circle is projected over each robot using an overhead projector to designate the current task that it is performing: green signifies task T_1, blue signifies task T_2, and red circles designate robots which are idle. In Fig. 5.9b, robots overlayed with black circles and designated with red arrows experience a failure and remain stationary in the domain for the rest of the experiment.

The tasks to be performed satisfy the assumptions stated in Section 4.1 and result in uniformly ergodic trajectories of the robots. The robots detect an encounter when they come with a certain distance of each other. At this point, they deploy minimally-invasive

control barrier certificates [130] to avoid collisions among each other, and continue along their trajectories. The robots switch tasks during encounters as described in Sections 5.2.1 and 5.2.5. We perform two different experiments to illustrate the salient features of the developed task allocation algorithm.

Case 1: Comparison with and without closed-loop rate control: In this scenario, we allow the robots to deploy the task allocation algorithm with and without the feedback-based rate control mechanism developed in Section 5.2.5. In particular, the transition probabilities corresponding to the maximum error in the rate controller (given in (5.32)) is set to be equal to the constant time-invariant rate corresponding to the open-loop controller. For a chosen set of parameters ($k = 50, m = 0.22, L_j = \frac{1}{r_j}, j \in \{1, \ldots, K\}$), Fig. 5.10 plots the allocation ratios of the swarm corresponding to the closed-loop task allocation run. As seen, the swarm achieves and maintains the desired allocation ratio. Figure 5.10b compares the number of transitions per robot per unit time in the case with and without the closed-loop rate control. Once the swarm is closer to the desired allocation, the number of transitions are significantly lower in the case of closed-loop rate control. This illustrates the efficacy of the algorithm in regulating the task-transition rates based on how far the swarm is from the desired allocation.

Case 2: Closed-loop task allocation with robot failures: In this scenario, 3 robots performing task T_1 experience a failure and stop moving in the domain. They remain as stationary obstacles in the domain and are not counted towards the task allocation. This failure occurs at time $t = 100$s and disturbs the allocation ratios in the swarm as seen in Fig. 5.11a. The inherently stable task allocation algorithm then re-attains the desired allocation ratios. Figure 5.11b plots the allocation estimates generated by the particle filters averaged over all the robots against the true allocation. As seen, the inter-robot encounters enable the robots to estimate the current allocation of the swarm, in order to regulate the rate of task-transitions. The transition rates decrease after the allocation has been achieved, diminishing the role of the task allocation and allowing task execution to occur (see Fig. 5.11c). To better illus-

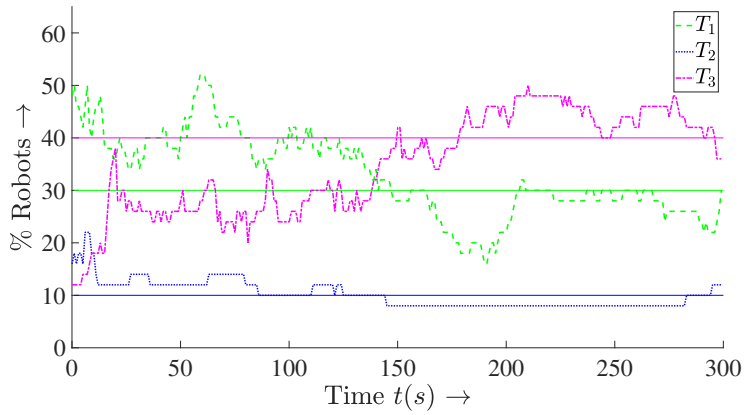
trate the effect of the failure, the results shown in Fig. 5.11 have been averaged over 3 experimental runs.

The experiments demonstrate the inherently-stable as well as adaptive nature of the developed task allocation algorithm. By constantly tracking the current allocation of the swarm using inter-robot encounters, each robot is able to switch between tasks at a rate proportional to the discrepancy between desired and current allocation levels. The robots use local encounter measurements to facilitate the task allocation without the need for extensive communication or a central coordinator.

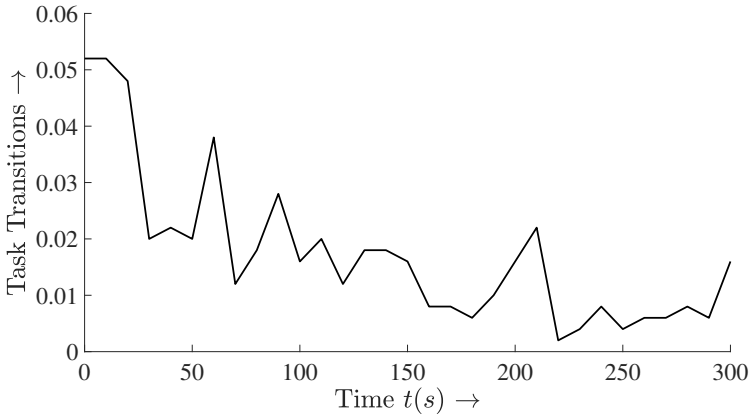
5.3 Conclusions

In this chapter, we developed a stochastic task allocation policy which allows a swarm of robots to autonomously distribute themselves among a set of tasks using inter-robot encounters as the primary interaction modality. The developed method allows individual robots to switch between tasks with certain transition probabilities when they encounter other robots in the domain. By appropriately choosing these transition probabilities, we demonstrate that the swarm can achieve a desired allocation in an inherently stable manner.

Building upon the encounter model developed in Chapter 4, we used ideas from the Enskog theory of high density gases to explicitly take into account the footprint area of the robots. This leads to a more accurate prediction of encounter rates in a swarm of robots.

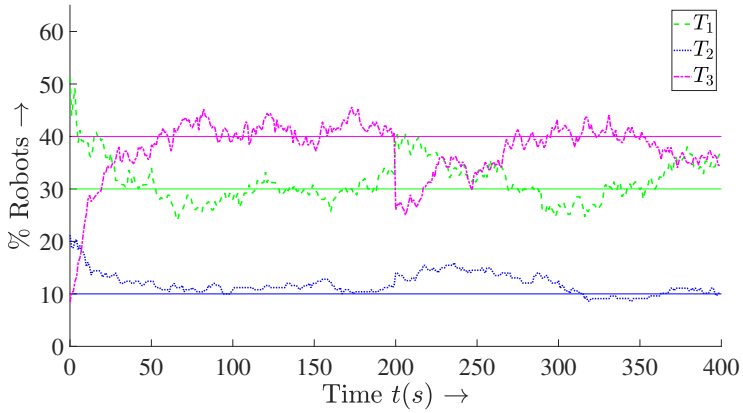


(a) Rate Controlled Task Allocation

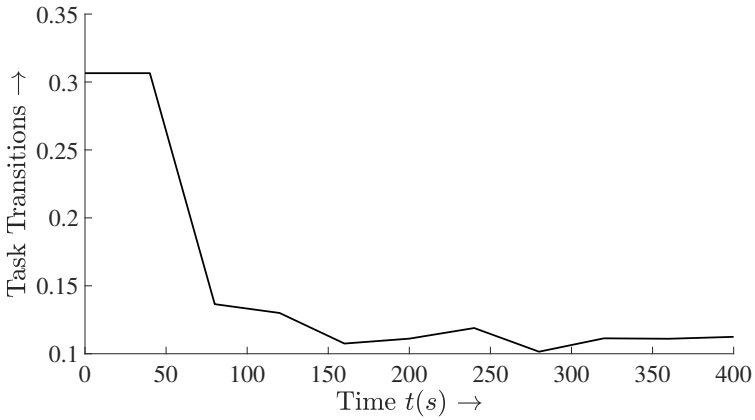


(b) Number of Task-Transitions

Figure 5.6: Effect of the closed-loop control of task-transition rates in the developed task allocation algorithm for a swarm of simulated robots with density $\bar{\lambda} = 50$ robots/m^2 in a domain $\mathcal{D}$ with robot parameters $(\delta = 0.02, r = 1.5\delta, v = 0.1)$. Figure 5.6a illustrates the true allocation ratios converging to the desired values. Figure 5.6b illustrates how the number of task-transitions per robot per unit time reduces as the true allocation approaches the desired value, indicating that the robots are able to measure the current allocation of the swarm and regulate the transition rates accordingly.



(a) Rate Controlled Task Allocation during failure



(b) Number of Task-Transitions

Figure 5.7: Performance of the closed-loop task allocation algorithm for a swarm of simulated robots in the presence of robot failures. At time $t = 200$s, 40% of the robots allocated to task T_3 experience a failure and are removed from the simulation. As seen in Fig. 5.7a, the swarm regulates the task allocation ratios back to the desired levels. In Fig. 5.7b, the number of transitions reduce as the swarm approaches the desired allocation ratio, but after the failure, the robots detect an increase in the allocation error, leading to a slight increase in the number of transitions after time $t = 200$s. This illustrates the closed-loop nature of the developed task allocation strategy. These results have been averaged over 5 simulation runs to better highlight the effect of the failure.

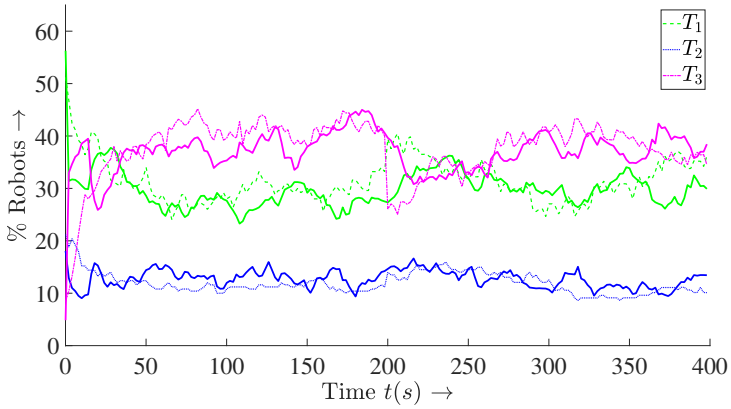
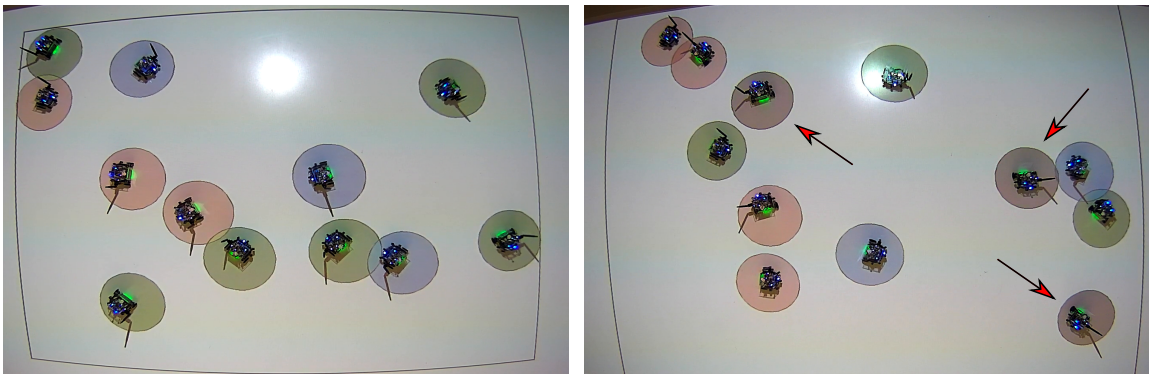
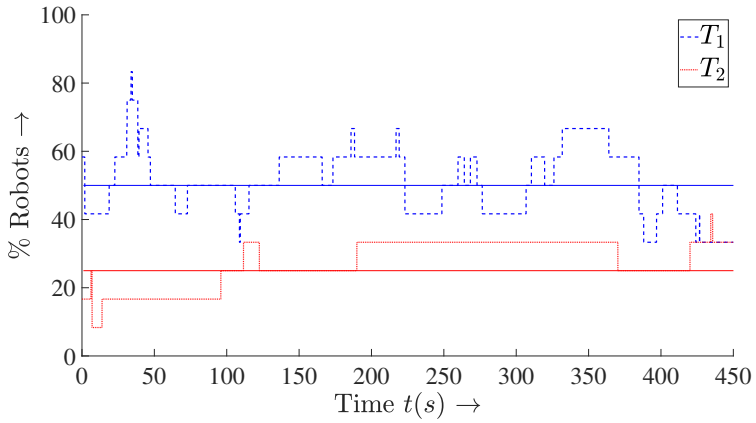


Figure 5.8: Performance of the particle filter algorithm in the event of robot failures. The solid lines represent the averaged particle filter estimates of all the robots in the swarm which did not experience a failure. The different dashed lines depict the true allocation levels in the swarm corresponding to each of the three tasks. When 40% of the robots allocated to task T_3 experience a failure, the encounter-based particle filter tracks the changing task allocation levels, thus enabling a closed-loop control of the transition rates.

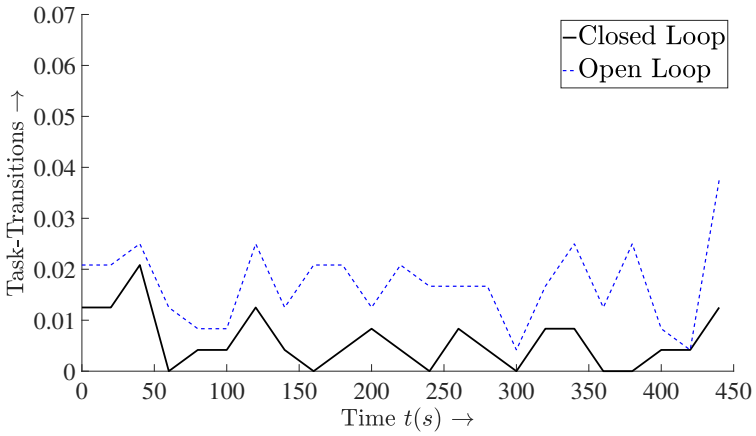


(a) (b)

Figure 5.9: A swarm of 12 robots are allocated among two tasks on the Robotarium [46], a remotely-accessible swarm robotics testbed. An overhead projector is used to overlay colored circles around each robot depicting the current task being performed by the robot. Green and blue circles represent a robot performing task T_1 and T_2, respectively. Red robots are idle, and waiting to take up a task. In Fig. 5.9b, some robots experience a failure and remain stationary in the domain. These robots are annotated by arrows, and are treated as stationary obstacles. The robots detect encounters when the physical footprint of another robot overlaps with their sensory footprint. The robots use minimally-invasive control barrier certificates [130] for collision avoidance.

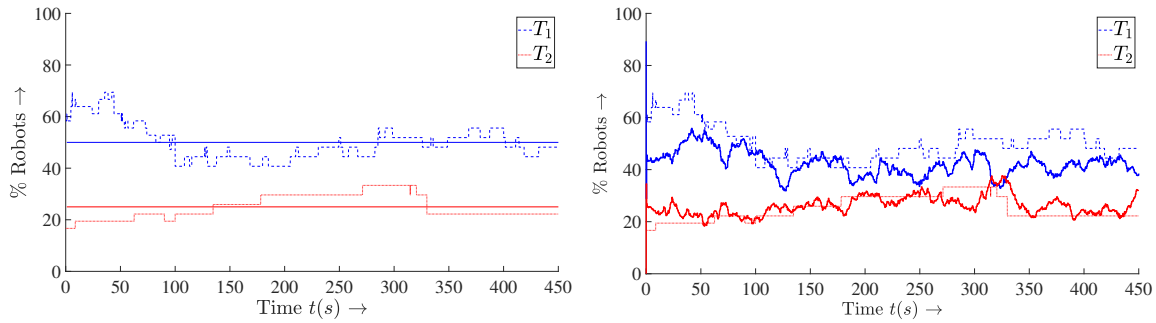


(a) Task Allocation



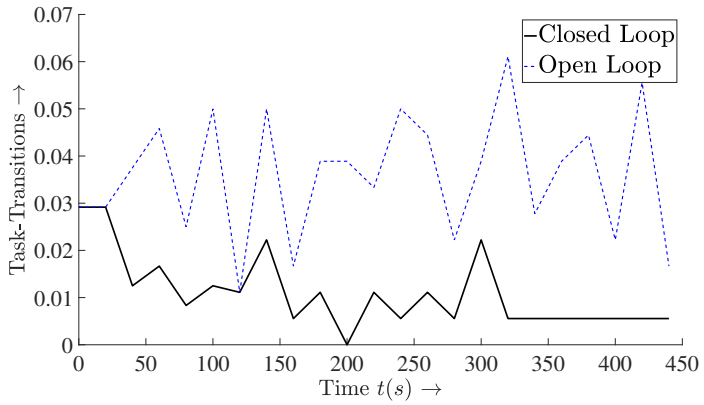
(b) Number of Transitions

Figure 5.10: Experimental Results for *Case 1: Comparison with and without closed-loop rate control*. For a swarm of 12 robots operating on the Robotarium, Fig. 5.10a plots the allocation results corresponding to the closed-loop task allocation case. As seen, the swarm achieves and maintains the desired allocation ratio. Fig. 5.10b compares the number of task-transitions per robot per unit time for the closed-loop rate controlled allocation algorithm and the open-loop variant. As seen, the inter-robot encounters allow the robots to estimate the current allocation of the swarm and correspondingly reduce the number of task-transitions.



(a) Task Allocation

(b) Particle Filter



(c) Number of Transitions

Figure 5.11: Experimental Results for *Case 2: Closed-loop task allocation with robot failures*. At time $t = 100$s, 3 robots allocated to task T_1 experience a failure and remain stationary in the domain (see robots overlayed with gray circles and annotated with arrows in Fig. 5.9b). As seen in Fig. 5.11a, this failure disturbs the allocation ratios, which is then reattained by the task allocation algorithm. The performance of the particle filters used to estimate the current allocation of the swarm is plotted in Fig. 5.11b averaged over all the robots. The solid lines represent the average estimate whereas the dotted lines represent the true allocation of the swarm in the two tasks T_1 and T_2. As seen, the estimates track the true allocations allowing the swarm to react to changes in the task allocation. Lastly, Fig. 5.11c plots the number of task-transitions corresponding to the closed-loop and open-loop task allocation algorithms. These results were averaged over 3 experimental runs to better illustrate the effect of the failure.

CHAPTER 6

LEVERAGING THE PHYSICS OF INTER-ROBOT ENCOUNTERS

Motivated by the multitude of examples where animal swarms use encounters to trigger different behaviors, Chapters 3, 4, and 5 used inter-robot encounters as an *explicit* source of information—measured either via contact-based or proximity sensors on the robots—which was incorporated into the function or task being performed by the robots. In contrast, the fundamental macro-scale properties exhibited by non-living collectives in nature are partially attributed to the physical forces experienced by individual particles during encounters [14].

In this chapter, we highlight a mechanism where robots swarms can *implicitly* leverage the forces experienced during inter-robot collisions to predictably achieve coexisting regions of low and high robot density. This work is motivated by the scenario where robots with extremely limited sensing and communication capabilities perform tasks which require them to spread across an environment, e.g., for distributed sensing or environmental surveillance applications [42, 143]. Under these circumstances, the robots might require spatial proximity to facilitate information exchange, while simultaneously performing task related actions [43, 144].

We demonstrate these behaviors on a team of vibration-driven robots, called *brushbots* [45] (see Chapter 7 for a detailed discussion on the brushbots). In particular, these robots do not possess sensors to detect other robots and simply traverse the environment while colliding with other robots. We illustrate that the mechanisms underlying the obtained density distributions can be explained using results from statistical mechanics, which investigates how macroscopic phenomena observed in physical systems can be related to the microscopic behaviors of constituent particles [14].

While equilibrium statistical mechanics provides an extensive vocabulary to describe

macroscopic behaviors, much of the classical theory deals with idealized interactions among particles, limiting its applicability in swarm robotics systems [145]. However, the study of physical systems that are far from equilibrium has shown promise in systematically analyzing complex collectives in nature [18]. In these *active matter* systems, an interplay between self-propulsion, inter-particle effects, and environmental forces leads to a wide-variety of emergent behaviors [15, 96]. This chapter takes advantage of a lesser-known *formal connection between certain types of active matter systems and equilibrium thermodynamics* to develop a microscopic description for a group of self-propelled brushbots, while retaining the extensive benefits of the classical thermodynamic theory.

We envision a team of brushbots moving randomly within a closed domain, while colliding with each other. The simultaneous formation of regions with lower and higher robot density is intuitively supported by two observations. Firstly, a given robot's speed decreases with increasing robot density around it—a direct consequence of the inter-robot collisions experienced by the robot. Secondly, a system of particles—which are embodied by robots in this context—tend to accumulate in regions where they move more slowly [146]. This phenomenon, known as *motility induced phase separation* (MIPS) in the physics literature [97], allows us to explore the conditions on the motion characteristics of the robots required to display such behaviors [44].

A team of brushbots provides an ideal platform for achieving such variable density behaviors, since their minimalist construction makes them robust to the force experienced during inter-robot collisions even at relatively high speeds (see Chapter 7). Additionally, the inherently noisy dynamics of the brushbots ensures that—similar to active matter systems—high-density robot clusters do not persist forever.

The outline of the rest of this chapter is as follows. Section 6.1 first introduces the stochastic differential equation describing the dynamics of each brushbot and uses the encounter models developed in Chapter 4 to derive a density-dependent speed profile for each robot. In Section 6.2, this model is leveraged to discuss the conditions under which motility

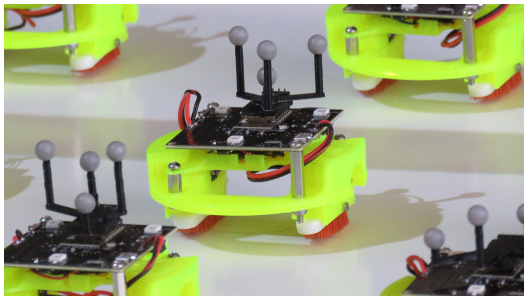


Figure 6.1: A differential-drive-like brushbot [45], actuated using two vibration motors and moving on two separate flexible bundles of bristles. Given this construction, we model the dynamics of the robot using unicycle dynamics, with translational and rotational additive noise to represent the unpredictability in the motion of the robot.

induced phase separation can occur by drawing connections with an equivalent system of particles at thermal equilibrium. Simulations confirm the formation of robot aggregations for varying parameter ranges. In Section 6.3, the mechanism is deployed on a team of real differential-drive–like brushbots to illustrate the formation of high and low robot densities.

6.1 Motion and Collision Model

6.1.1 Motion Model

In this section, we first briefly motivate the dynamical model for the brushbot operating in an environment with no other robots present. Following this, we analyze the effects of collisional interactions between multiple robots and characterize the velocity of the robots as a function of swarm density.

The differential-drive–like construction of our brushbots (see Fig. 6.1) allows the motion of the robot to be expressed using unicycle dynamics with linear and angular velocity as the inputs. For a group of N robots, let $z_i = (x_i, y_i)$ and θ_i denote the position and orientation of robot $i \in \{1, \ldots, N\} \triangleq \mathcal{N}$, respectively. Each robot has a circular footprint of radius r and operates in a closed and bounded domain $\mathcal{V} \subset \mathbb{R}^2$.

As will be elaborated in more detail in Chapter 7, the presence of manufacturing differences as well as the nature of bristle-based movement underline the need for a stochastic

model to describe the motion of the brushbots. To this end, we introduce additive noise terms which affect the translational and rotational motion of the robots, with diffusive coefficients D_t and D_r, respectively. Under this model, the state (z_i, θ_i) evolves according to standard Langevin dynamics, which is a stochastic differential equation, given componentwise as

$$\begin{aligned} dx_i &= v_0 \cos\theta_i + \sqrt{2D_t}\Delta W_x, \\ dy_i &= v_0 \sin\theta_i + \sqrt{2D_t}\Delta W_y, \\ d\theta_i &= \sqrt{2D_r}\Delta W_\theta, \end{aligned} \tag{6.1}$$

where v_0 is the constant self-propelled speed of the robot and $\Delta W_x, \Delta W_y, \Delta W_\theta$ denote Wiener process increments representing white Gaussian noise. The total distance traveled by the robot will depend on the diffusion parameters (D_t, D_r) and the speed v_0 [147]. The effective diffusion coefficient of the robot [18], measured on time scales longer than the time required for the orientation of the robot to uncorrelate (defined as $\tau_r = D_r^{-1}$), can be quantified as,

$$D = \frac{v_0^2}{2D_r} + D_t. \tag{6.2}$$

An important predictor of phase separation behavior in interacting systems is the *activity* parameter, e.g., [99, 100], which we define here as

$$\mathcal{A} = \frac{v_0}{2rD_r}. \tag{6.3}$$

The activity parameter is proportional to the distance traveled by a robot before its direction uncorrelates completely [99] and will be used to characterize the proportion of high and low robot density regions in Section 6.2. In Section 6.3, we use robotic experiments to empirically determine the diffusion coefficients D_t and D_r of the brushbot, but for now, we simply assume that these parameters are available to us and are constant in the swarm.

6.1.2 Inter-Robot Interaction Models

In the previous section, we described the dynamics of a single robot moving with a self-propelled speed v_0 through the domain. Given a group of N brushbots moving randomly according to these dynamics in the domain $\mathcal{V}$, we now develop a model to describe the effects of inter-robot collisions on the average speed of the robots.

For a team of colliding robots, average robot speeds reduce with increasing density in the region around the robot—a direct consequence of the density-dependent collision rates [23]. The distribution of robots over the domain can be described in terms of the coarse-grained density $\lambda : \mathcal{V} \to \mathbb{R}_+$, obtained by overlaying a suitable smoothing function, e.g. [99], over the microscopic density measure,

$$\sum_{k=1}^{N} \delta(z - z_k), \tag{6.4}$$

defined at each point $z \in \mathcal{V}$. Here δ denotes the Dirac delta measure. We defer the actual computational details of the coarse-grained density to Section 6.2.2, which discusses the simulation results.

A given robot i will experience varying collision rates depending on the density of robots surrounding it. The developed model is agnostic to which robot we pick since the following mean-field analysis applies to any robot in the domain. Then, the speed of a given robot i is a function of its location as well as the robot density in the environment λ. The following assumption simplifies the dependence of the robot speed on the robot density.

Assumption 4. *Assume that the coarse-grained density of robots λ varies slowly over the domain,*

$$\left\| \frac{\partial \lambda}{\partial z} \right\| \ll 1, \forall z \in \mathcal{V}. \tag{6.5}$$

Furthermore, let the speed of robot i depend only on the densities in some neighborhood of the robot location z_i. Then, the speed of the robot can be assumed to depend only on the

density at the robot's location.

Consequently, we denote the speed of robot i as $v(\lambda(z_i))$. This assumption allows us to develop an analytical expression for the speed of a robot at a given location, as a function of the robot density at that location. Using the inter-robot collision model developed in Chapter 4, the expected time between collisions experienced by robot i at its current location z_i is given as,

$$\tau_c(\lambda(z_i))^{-1} = 4r\frac{4}{\pi}v_0\lambda(z_i), \tag{6.6}$$

where r denotes the radius of each robot. The modified speed term $(4/\pi)v_0$ is equal to the mean relative speed between all the robots (see Lemma 2).

Furthermore, let τ_m denote the expected time spent by a robot in a collision with another robot before re-attaining its self-propelled speed v_0. This process occurs via forces acting on the robots as well as the rotational diffusion of each robot (as dictated by the rotational diffusion coefficient D_r in (6.1)). For instance, the rotational diffusion might cause the robots to eventually move in different directions, effectively resolving the collision. In Section 6.1.3, for a choice of diffusion parameters, we use a team of simulated brushbots to determine the average speed empirically and to estimate τ_m as the parameter value which best fits the speed data.

With the robot traveling at speed v_0 between collisions and effectively remaining stationary during a collision, the average speed may be expressed as

$$v(\lambda(z_i)) = v_0\left(1 - \frac{\tau_m}{\tau_c(\lambda(z_i)) + \tau_m}\right). \tag{6.7}$$

Under the simplifying assumption that the time to resolve collisions is smaller than inter-collision time intervals, i.e., $\tau_m \ll \tau_c$, (which is valid at low as well as intermediate densities), and substituting from (6.6) the expression for τ_c, (6.7) can be cast into the general

form

$$v(\lambda(z_i)) = v_0 \left(1 - \frac{\lambda(z_i)}{\lambda^*}\right), \tag{6.8}$$

where

$$\lambda^* = \left(4r\frac{4}{\pi}v_0\tau_m\right)^{-1} \tag{6.9}$$

represents the extrapolated density at which the speed becomes zero, and can be interpreted as the packing density of robots in the domain [97]. Next, we introduce a simulation setup which verifies the linear dependency of velocity on local robot density (as predicted by (6.6), (6.8) and (6.9)), and gives numerical estimates for the collision resolution time τ_m.

6.1.3 Simulation Setup

We validate the density-dependent speed model using a simulated team of brushbots operating in a closed and bounded rectangular domain $\mathcal{V}$. The simulation consists of N disks of a fixed radius r moving according to the dynamics described by (6.1). During collisions, excluded volume constraints—a consequence of the fact that robots are not inter-penetrable—are imposed via a force acting on the robots, as is common in the physics literature, e.g., [100]. In order to avoid the influence of boundary effects, periodic boundary conditions are applied, i.e., when robots exit from one side of the domain, they reappear on the other side. In Section 6.3, when performing experiments on the actual brushbots, we allow the robots to turn around when they reach the boundaries of the domain.

The simulation is performed for a uniform distribution of robots over the domain to prevent the averages from being affected by variable high and low density regions. A uniform distribution was ensured by selecting the activity parameter $\mathcal{A}$, defined in (6.3), to be well below the values which would lead to phase separation (see supplementary material for [99]). The average speed of the swarm at each point in time was computed by projecting the instantaneous velocity of each robot along its current orientation vector

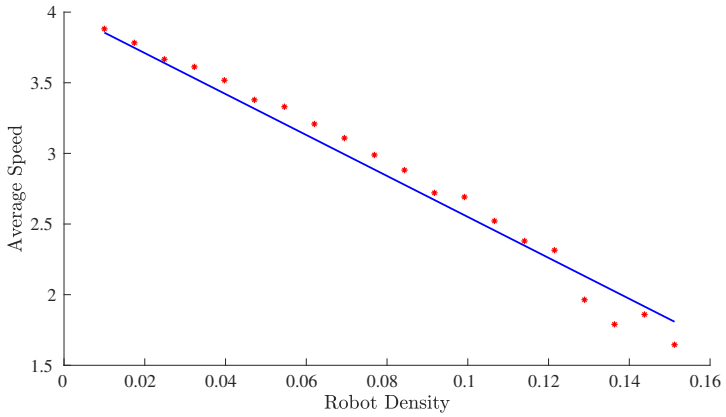


Figure 6.2: Validation of the relation between the average speed of robots and the swarm density using the simulation setup described in Section 6.1.3. Average speed measurements are plotted with red dots, while the best-fit line is shown in blue. At low densities, the average speed equals the self-propelled speed of the robots (set at $v_0 = 4$ in this simulation). As the density increases, robots experience higher collision rates which slows them down. Best-fit collision resolution time $\tau_m = 0.177s$ (see (6.8)).

$\Theta_i = [\cos\theta_i, \sin\theta_i]^T$,

$$\hat{v}(t) = \frac{1}{N}\sum_{i=1}^{N} \dot{z}_i(t)^T \Theta_i(t). \tag{6.10}$$

For different robot densities, Fig. 6.2 depicts the average robot speeds empirically computed over the duration of a simulation. As seen, the average robot speed decreases linearly with density, as predicted by (6.6) and (6.8). Deviations at high densities can be justified by the violation of the assumption that $\tau_m \ll \tau_c$. In the next section, we illustrate how the slowdown of robots due to collisions can actually lead to phase separated regions of high and low robot densities, as predicted by equilibrium thermodynamics.

6.2 Motility-Induced Phase Separation

In the previous section, we analyzed the velocity profile of robots interacting purely via inter-robot collisions. In this section, we provide a brief summary of important results in the active matter literature that establish an equivalence between such a swarm of robots and a passive Brownian molecular system at equilibrium (see [97] for further details). Such an equivalence allows us to specify the system parameters under which the swarm can be

expected to phase separate and form regions of unequal robot density.

6.2.1 Theoretical Analysis

We begin the analysis by studying the probability distribution of a single particle whose speed, denoted as $v(\lambda(z))$, varies spatially over the domain according to the robot density. For a single robot obeying the Langevin dynamics in (6.1), the probability density ϕ describing the location of a robot over the domain evolves according to the following equation, as given in [148],

$$\dot{\phi}(z) = \nabla \cdot J, \tag{6.11}$$

$$J = -D(\lambda(z))\nabla\phi(z) + V(\lambda(z))\phi(z), \forall z \in \mathcal{V}, \tag{6.12}$$

where $\nabla\cdot$ denotes the divergence operator, ∇ denotes the gradient, and D is now the spatially varying diffusion coefficient of the robot, modified from (6.2) as

$$D(\lambda(z)) = \frac{v^2(\lambda(z))}{2D_r} + D_t. \tag{6.13}$$

The drift-velocity V satisfies the relation

$$\frac{V(\lambda(z))}{D(\lambda(z))} = -\left(1 + \frac{2D_tD_r}{v^2(\lambda(z))}\right)^{-1} \nabla \ln v(\lambda(z)). \tag{6.14}$$

From these equations, the coarse-grain density of robots interacting with each other has been shown [97] to evolve according to the following stochastic differential equation

$$\dot{\lambda}(z) = -\nabla \cdot \Big(- D\nabla\lambda(z) + V\lambda(z) + \sqrt{2D\lambda(z)}\Lambda\Big), \tag{6.15}$$

where Λ represents white Gaussian noise interpreted in the Itô sense [97]. We suppress the explicit dependence of D and V on $\lambda(z)$ for readability.

The steady state probability distribution of the density, $\mathcal{P}_{\text{eq}}(\lambda)$, can be obtained by solving for the equilibrium solution of the corresponding Fokker–Planck equation,

$$\left[V\lambda - D\nabla\lambda - D\lambda\Big(\nabla\frac{\partial}{\partial\lambda}\Big)\right]\mathcal{P}_{\text{eq}}(\lambda) = 0. \tag{6.16}$$

The following theorem, initially presented in [149] and summarized in [97], illustrates how a swarm of robots interacting via collision interactions can be mapped to a system of passive Brownian particles at equilibrium.

Theorem 6. *([97]) A team of robots operating in a domain $\mathcal{V}$, satisfying the dynamics in* (6.1) *and interacting purely via inter-robot collisions, can be expressed as a system of passive Brownian particles with an attractive potential if the following conditions are satisfied:*

1. *Assumption 4 is valid.*

2. *The coarse-graining of the microscopic density measure given by* (6.4) *is valid.*

Under these conditions, the free energy density of the system can be expressed as,

$$f(\lambda(z)) = \lambda(z)(\ln(\lambda(z)) - 1) + \int_0^{\lambda(z)} \frac{1}{2}\ln\Big(\frac{v^2(s)}{D_r} + 2D_t\Big)ds,\ \lambda(z) \leq \lambda^* \tag{6.17}$$

where λ^* is given by (6.9)*.*

Proof. An equivalence between the self-propelled robot swarm and a passive Brownian particle system at equilibrium is made by expressing the steady-state solution of (6.16) as obeying the equilibrium Boltzmann distribution [98] over the domain, given as

$$\mathcal{P}_{eq} \propto exp(-\mathcal{F}), \tag{6.18}$$

where $\mathcal{F} = \mathcal{F}_{\text{ent}} + \mathcal{F}_{\text{vel}}$ is the free energy of the system, expressed in terms of the free

energy density functional f,

$$\mathcal{F} = \int_{z \in \mathcal{V}} f(\lambda(z))\, dz. \tag{6.19}$$

The free energy density, given by,

$$f(\lambda(z)) = \lambda(z)(\ln \lambda(z) - 1) + f_{\text{vel}}(\lambda(z)), \tag{6.20}$$

comprises of two parts: an ideal entropy contribution (contributing to $\mathcal{F}_{\text{ent}}$) and an excess energy density f_{vel} (contributing to $\mathcal{F}_{\text{vel}}$). The latter, which would stem from the attractive potential in a passive Brownian system, in fact stems from the density dependent velocity profiles of the robots. To satisfy (6.18), the following integrability condition results (discussed in [149]),

$$\frac{V(\lambda(z))}{D(\lambda(z))} = -\nabla \frac{\partial \mathcal{F}_{\text{vel}}}{\partial \lambda}. \tag{6.21}$$

Substituting from (6.14), and observing that $\nabla \ln v = v^{-1} \nabla v$, this condition can be re-written as,

$$\frac{v \nabla v \tau_r}{v^2 \tau_r + 2D_t} = \frac{\partial \mathcal{F}_{\text{vel}}}{\partial \lambda}, \tag{6.22}$$

where $\tau_r = D_r^{-1}$. Given Assumption 4, which implies that the speed of each robot only depends on the local density, (6.22) is satisfied by the following speed-derived free energy density

$$f_{\text{vel}}(\lambda) = \int_0^{\lambda} \frac{1}{2} \ln \left(\frac{v^2(s)}{D_r} + 2D_t \right) ds, \tag{6.23}$$

under the constraint that the density λ can never exceed the close packed value given by (6.9). This leads to the desired result. □

Given the equivalence established by Theorem 6, we can analyze the phase separation properties of the robot swarm by analyzing the free-energy density functional in (6.17). The following observations summarize the conditions under which such a system can be expected to phase separate.

Observation 3. *In classical thermodynamics [98], spontaneous separation of a particle system into regions of high and low density occurs when concavities exist in the free energy density f defined in* (6.17)*. More specifically, when the local density $\lambda(z)$ at a point $z \in \mathcal{V}$ is such that $f''(\lambda(z)) < 0$ where $'$ denotes a derivative with respect to the density λ, small fluctuations cause the system to phase separate into regions with densities which result in a reduction in free energy density. This process is called spinodal decomposition.*

Given (6.17), it is a simple exercise to verify the conditions on the density-dependent velocity of the robots which favor occurrence of spinodal decomposition in the system [97],

$$f''(\lambda(z)) < 0 \iff \frac{v(\lambda(z))^2}{D_r}\left(1 + \lambda(z)\frac{v'(\lambda(z))}{v(\lambda(z))}\right) < -2D_t. \tag{6.24}$$

Substituting the expression of density dependent speed from (6.8), we can make the following further observation with regards to the *conditions on robot density and parameters* under which the swarm can be expected to achieve non-uniform density distributions.

Observation 4. *For a team of brushbots with velocity dependent speed given in* (6.8)*, the spinodal densities (at which $f''(\lambda) = 0$), represented as $\lambda_s^{\pm}$, are given as*

$$\lambda_s^{\pm} = \frac{\lambda^*}{4}\left(3 \pm \sqrt{1 - 16D_tD_r/v_0^2}\right), \tag{6.25}$$

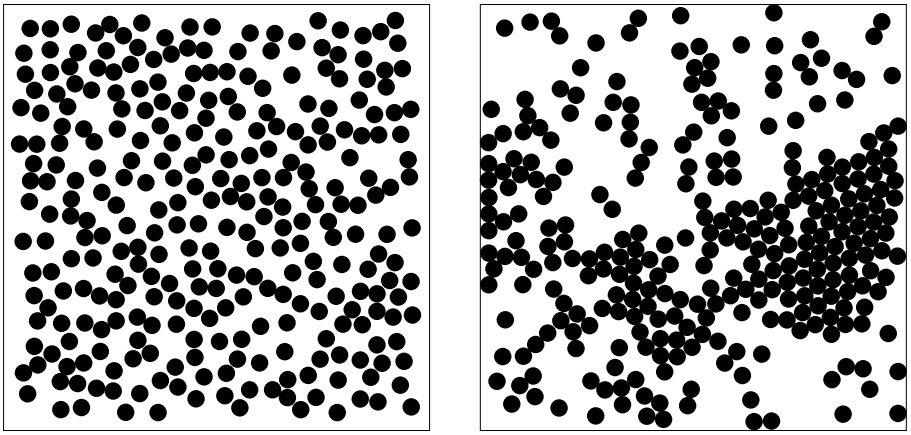
where λ^* is specified in (6.9)*. Consequently, these spinodal points only exist when the following condition is satisfied by the robot parameters:*

$$v_0 > \sqrt{16D_tD_r}. \tag{6.26}$$

Because robot densities cannot exceed the packing density λ^*, the following ranges of densities,

$$\lambda_s^- \leq \lambda(z) \leq \min(\lambda_s^+, \lambda^*), \quad z \in \mathcal{V} \tag{6.27}$$

favor spontaneous phase separation.



(a) $t = 0s$ (b) $t = 15s$

Figure 6.3: Snapshots illustrating the formation of high as well as low density regions concurrently in a team of simulated robots operating in a domain with periodic boundary conditions. For a set of simulation parameters ($N = 292, r = 1, v_0 = 4, D_r = 1e^{-4}, D_t = 1e^{-5}$), each snapshot represents the configuration of the system at different times. Inter-robot collisions slow down the robots, which cause additional robots to join the clusters. This leads to the formation of dynamic high-density clusters along with the existence of lower density regions in the domain.

This implies that, when the local density in a region of the domain lies in the range specified by (6.27), the swarm can be expected to spontaneously phase separate into regions with low and high robot densities.

6.2.2 Simulations

We observe the formation of dynamic robot clusters in a rectangular domain with periodic boundary conditions using the simulation setup described in Section 6.1.3. Figure 6.3 illustrates the intermediate-time snapshots of the phase separating swarm robotic system. The robots are placed according to a uniform distribution in the domain at time $t = 0$ (see Fig. 6.3a) and simply move according to the Langevin dynamics described in Section 6.1 while experiencing collisions among each other. Figure 6.3b depicts how an initially uniformly distributed swarm of robots can form regions of high and low robot density, primarily caused by the density-dependent slow down of robots which precipitates the formation of dynamic high-density robot clusters.

We characterize the simultaneous existence of regions with higher and lower robot densities by computing the coarse-grained density λ over the domain,

$$\lambda(z) = \sum_{i=1}^{N} w(\|z - z_i\|), \forall z \in \mathcal{V}, \tag{6.28}$$

where the weighting function w is given as,

$$w(d) = \exp\big(- d_c^2/(d_c^2 - d^2)\big). \tag{6.29}$$

Here, d_c is the cutoff distance at which $w(r) \rightarrow 0$. In simulation, the coarse-grained density was evaluated on a grid of l^2 lattice points, with cutoff distance $d_c = 0.8l$. For a grid size $l = 10$, Fig. 6.4 plots the empirically obtained distribution of the coarse-grained densities evaluated at the lattice points. These values were obtained by applying Gaussian kernel-smoothing on the histogram of coarse-grained densities over the grid. For each set of activity parameters, data points were collected from multiple simulations of the robot swarm, to average out spurious effects caused by initial conditions. As seen, the distribution of robot densities is unimodal at low activity levels and becomes distinctly bimodal at higher activities, indicating the formation of low and high robot densities in the domain.

To quantify the formation of clusters in the swarm, we measure the time-averaged fraction of robots that belong to high-density aggregations in the swarm. An *aggregation* is defined as robots in physical contact with each other, whose total size exceeds a minimum cut-off value N_c. Figure 6.5 illustrates the impact of the activity parameter $\mathcal{A}$ of the robots, by plotting the average aggregation fraction for increasing activity values and for different mean robot densities in the domain (denoted as $\bar{\lambda} = N/|\mathcal{V}|$ where $|\mathcal{V}|$ denotes the area of domain $\mathcal{V}$). As seen, no significant robot aggregation occurs below a certain activity threshold of the robots, regardless of the density. This observation is in agreement with the lower activity thresholds for observed phase separation in studies, e.g., [97]. Figure 6.6 illustrates the need for a sufficient robot density to see phase separation behaviors, as

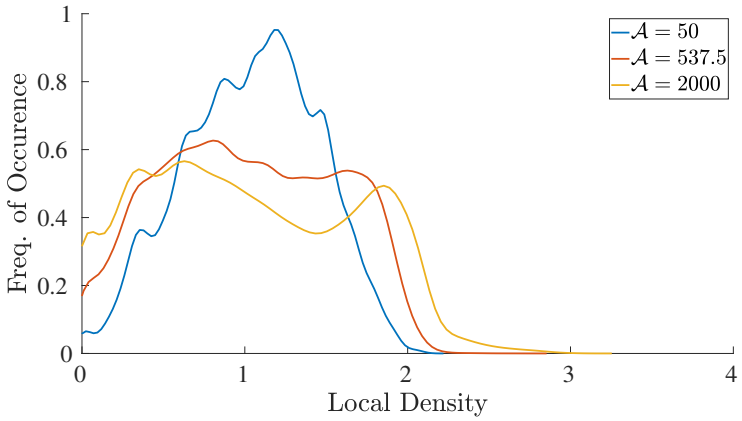


Figure 6.4: Empirically obtained distribution of robot densities evaluated over a grid of size l^2 with $l = 10$. Gaussian kernel-smoothing was applied on the histogram of robot densities collected over multiple simulations at a constant robot density (simulation parameters: $r = 1, D_t = 1e^{-5}, N = 382$). For high activity parameters, the distribution is seen to be distinctly bimodal, due to the formation of high and low robot density regions as predicted in Section 6.2. For lower activity parameters, the swarm does not phase separate and the density distribution over the grid remains unimodal. This presents a mechanism to characterize the simultaneous existence of higher and lower robot densities in the domain. The displayed data were collected over multiple simulation runs.

predicted by (6.27).

6.3 Deployment and Experiments

In this section, we first identify the noise parameters of a single brushbot and discuss the implications of these parameters on the phase separation properties of the swarm. Following this, we illustrate the emergence of dynamic regions of low and high robot density using a team of 26 brushbots operating in a confined rectangular environment.

In order to identify the translational and rotational diffusion coefficient of the brushbots, we excited one brushbot with a constant self-propelled linear speed and zero angular velocity in a confined rectangular space. Position and orientation data were collected using an overhead tracking system. When a robot encountered the walls of the environment, the data collection was terminated until the robot turned around to traverse the environment again. Data were collected for a total time of 300 sec at a self-propelled speed $v_0 = 6$ cm/sec.

The rotational diffusion coefficient can be computed by measuring the mean-square

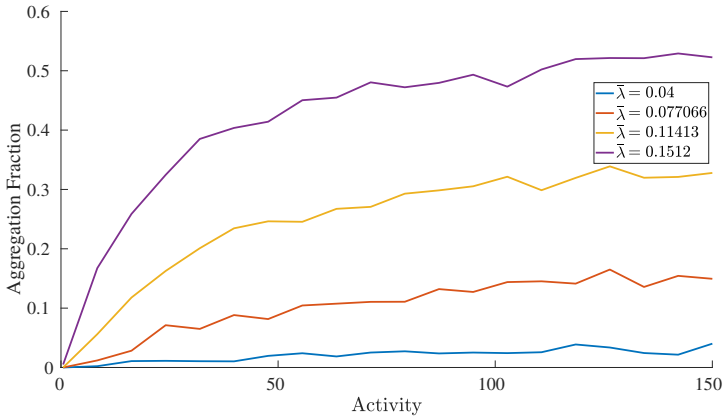


Figure 6.5: Dependence of the average aggregation fraction on the activity parameter of the robots (simulation constants: $r = 1, D_t = 1e^{-5}$). The aggregation fraction measures the fraction of robots which belong to high-density clusters (chosen with a cut-off size $N_c = 4$) averaged over time. As seen, at low activity levels, the extent of aggregation remains low regardless of the density of robots in the domain. As the activity is increased, the fraction of robots in high density clusters increases. In the legend, $\bar{\lambda}$ denotes the mean density of robots in the domain.

variation in the orientation of a robot [150]. We numerically estimate D_r by performing a linear regression on the mean-square variations in the orientation of the robot for varying time-intervals (estimated value $D_r = 0.0041$). It is noteworthy that the value for the translational diffusion coefficient D_t as computed from these experiments was negligible within numerical precision (computed using the Green–Kubo method [150]).

These estimates of the noise characteristics of a brushbot, allow us to make the following observation regarding the ability of a swarm of brushbots to spontaneously phase separate

Observation 5. *Based on the robot noise parameters identified ($D_r = 0.014, D_t = 0$),* (6.26) *is satisfied always. Consequently, a swarm of brushbots can be expected to spontaneously form regions of low and high robot density as long as the number of robots are high enough to achieve the local densities described in* (6.27).

It should be noted that the identification of diffusion parameters for the brushbots performed above is not meant to serve as a quantitative analysis of the noise properties of the brushbots, but to understand the qualitative effects of these values on the formation of lower

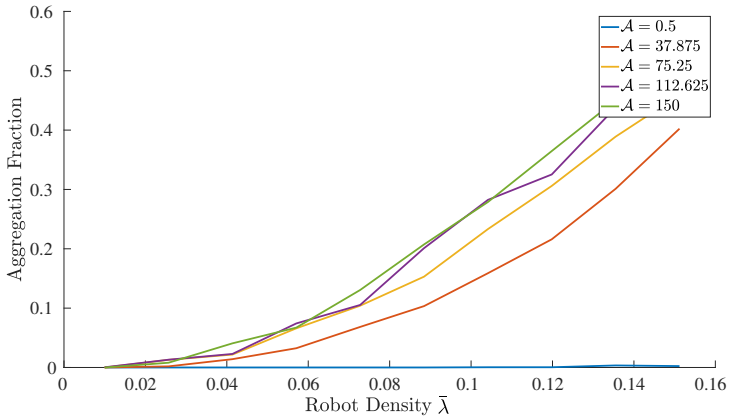


Figure 6.6: Dependence of the aggregation behaviors on the mean density of robots in the domain, as predicted by (6.27). Below a certain density, no significant aggregation is seen in the robots regardless of the activity parameter of the robots (simulation constants: $r = 1, D_t = 1e^{-5}$). This is primarily because robots are able to resolve collisions (therefore dissolving clusters) before other robots can join the cluster.

and higher density regions in the swarm. Indeed, since the effects from translational noise are minimal, the exact values of the noise parameters do not play a role in the prediction of motility-induced phase separation.

Figure 6.7 illustrates intermediate-time snapshots of 26 brushbots achieving regions of varying robot density in an enclosed square domain. As expected, collisions caused the robots to slow down, which precipitated the formation of high density regions. Robot clusters were identified based on physical contact among the robots, beyond a minimum threshold size (N_c was chosen as 4). We quantitatively analyzed the formation of these clusters by plotting the aggregation fraction in the swarm over time. This represents the fraction of robots which belong to a cluster. As seen in Fig. 6.8, at least 20% of the robots were clustered in aggregations throughout the experiment, while the rest of the robots moved freely through the domain.

6.4 Conclusions

This chapter differentiates itself from the previous chapters by *implicitly* taking advantage of the physics of inter-robot collisions to achieve a desirable emergent property. We envi-

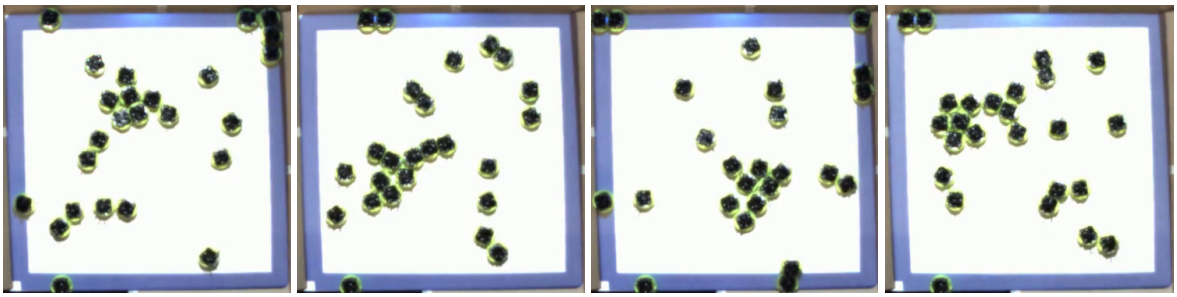


Figure 6.7: Snapshots for a team of 26 brushbots propelled at a constant speed in a square environment. Reflective boundary conditions were applied by injecting an angular velocity to the robots when they hit the boundaries. As predicted by phase separation theory, collisions cause the robots to slow down which leads to the formation of high density robot clusters that co-exist along with regions of lower robot density. These clusters form and dissolve over time in different places.

sion a scenario where a swarm of colliding vibration-driven brushbots form high-density robot aggregates that coexist with lower robot densities in space. Theoretical techniques from the study of far-from-equilibrium collectives and statistical mechanics clarify the mechanisms underlying the formation of these high and low density regions. Specifically, we capitalize on a transformation that connects the collective properties of a system of self-propelled particles with that of a well-studied molecular fluid system, thereby inheriting the rich theory of equilibrium thermodynamics. This connection is a relatively new one, and is previous unexplored in the context of robotics. We use real robot experiments as well as simulations to illustrate how inter-robot collisions can precipitate the formation of non-uniform robot densities in a closed and bounded region.

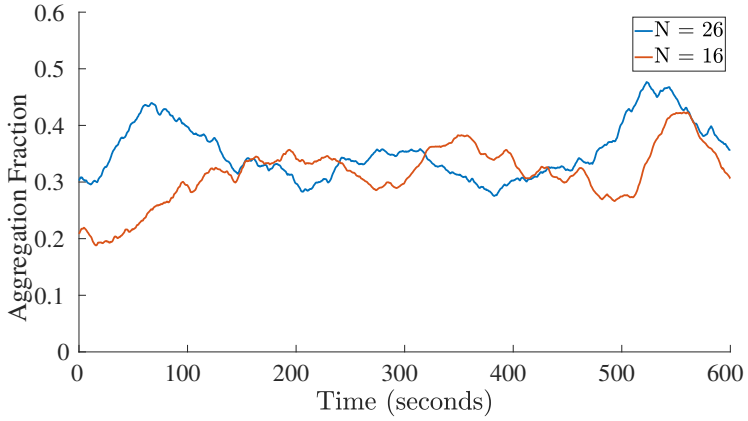


Figure 6.8: The fraction of robots belonging to high density robot aggregations for a team of real brushbots. An aggregation is defined as a collection of robots beyond a cut-off size in physical contact with each other. The robots travel randomly in the domain while colliding with each other. The reduction in speed caused due to collisions precipitates the formation of simultaneous regions of low and high robot density as predicted by the phase separation theory in Section 6.2. As seen, the fraction of robots in aggregations remains fairly significant throughout the experiments.

CHAPTER 7
DEVELOPING SWARM ROBOTICS PLATFORMS

The encounter-based methods developed throughout this thesis have been validated and tested on two robotic platforms: (i) the *Robotarium* [46, 151] and (ii) the *brushbots* [45]. The Robotarium—which is a remotely accessible swarm robotics testbed, where code is uploaded remotely and experimental results are returned to the user—is occupied by a team of wheeled differential-drive robots. In contrast, the brushbots achieve directed motion by vibrating flexible bundles of bristles. Their locomotion modality and construction makes them ideal for experiments which involve collisions among the robots even at relatively high speeds. This chapter details the design and development of both these swarm robotics platforms, which have been conceptualized and built at the Georgia Institute of Technology. Given the larger purpose served by the Robotarium as an open-access robotics testbed, Section 7.1.2 highlights some measures taken to ensure the reliable and safe operations of its constituent robots.

7.1 Robotarium

The Robotarium is a remotely accessible swarm-robotics testbed, designed to help users from all across the world quickly prototype and validate their distributed control strategies through implementation on physical robots without the overhead of setting up their own hardware [151]. With an explicit ambition to democratize access to world-class research infrastructure, the platform is free to be used by anyone from anywhere in the world for academic research or educational purposes. The testbed also serves to facilitate the performance comparison of different algorithms by deploying them on a standardized hardware environment.

The Robotarium, pictured in Fig. 7.1, resides in a refurbished classroom on the cam-

Figure 7.1: The Robotarium, a remotely accessible swarm robotics tested, allows users to remotely upload code which gets executed on a team of physical differential-drive robots [151]. In addition to these robots, the hardware infrastructure of the Robotarium consists of an elevated arena where the robots operate, a motion capture system seen attached to the roof above the arena, and wireless chargers embedded into the walls of the arena.

pus of the Georgia Institute of Technology, consisting of a 12 ft $\times$ 14 ft elevated platform where the robots operate. The walls of the platform are outfitted with wireless chargers which allow the robots to charge without human supervision. Eight motion capture cameras are mounted above the perimeter of the testbed that provide position and orientation information of the robots for data acquisition and control purposes. Each robot is tracked through a unique, nonsymmetrical pattern of reflective markers. While ensuring the robust and reliable operation of the Robotarium involved closely-intertwined developments on the hardware, software and theory front, the remainder of this section focuses on the robots which operate on the Robotarium, and highlights some theory which has been developed to improve the safe and reliable operations of the robots.

7.1.1 The GRITSBot X

The original robot that inhabited the Robotarium was the GRITSBot, whose design focused on minimizing cost and form factor [152]. More recently, these robots were replaced with the GRITSBot X, whose design explicitly focuses on reliable long-term operations, ease of manufacturing as well as maintenance. Below we outline some of these design considera-

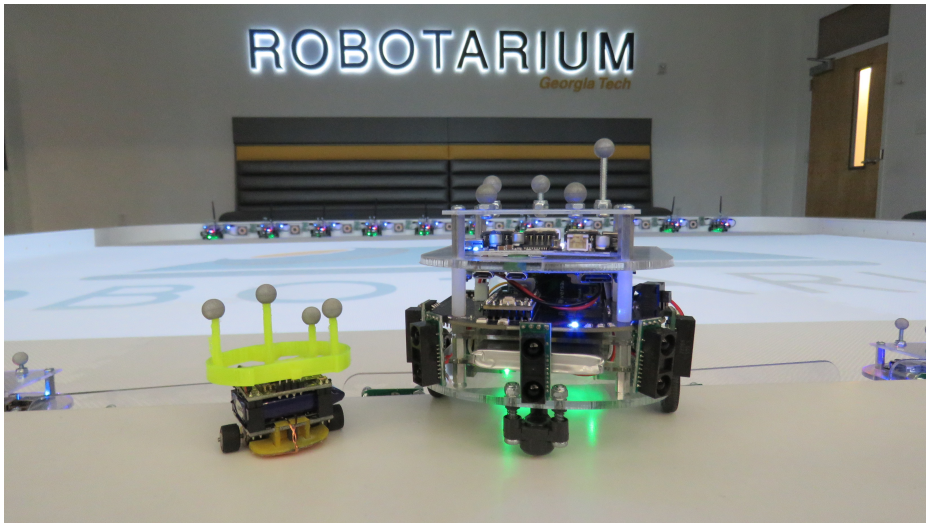


Figure 7.2: The GRITSBot (pictured on the left) was the first robot to operate on the Robotarium and was designed with a focus on low-cost manufacturing and small form factor [152]. It has now been replaced with the GRITSBot X (pictured on the right), which is designed to provide reliable and robust operations for long periods of time without direct human supervision [151].

tions in further detail, and highlight the actions taken to address them. Figure 7.2 illustrates the GRITSBot and its newer variant, the GRITSBot X, on the Robotarium arena.

Standardized Manufacturing With the aim of reducing manufacturing time and improving the ease of manufacturing, the GRITSBot X chassis is comprised of two custom, laser-cut 0.31 cm thick acrylic plates. This chassis is robust enough to handle limited collisions and can be rapidly replicated using a laser cutter. The motors, custom PCB, battery, and tracking markers are attached to the acrylic chassis through screws and threaded standoffs. All electronics are attached through standard Japan Solderless Terminal (JST) connectors. Together, these choices enable rapid production of the GRITSBot X with low variance between the quality of each robot's mechanical assembly and electrical connections.

Robust Actuation To achieve reliable actuation, the GRITSBot X is equipped with micro metal gear motors, which dictate the 9.5cm $\times$ 8cm footprint of the GRITSBot X. These motors are widely available commercially with quadrature encoders (0.25° resolution) mounted directly to the motor shaft. This hardware enables a sufficiently high-resolution

feedback to maintain reliable actuator performance even when constant use degrades the hardware and unknown disturbances, like foreign particles, perturb the ideal output.

Repairability Since extended use of the robots will inevitably lead to failure among individual components, the two motors of the GRITSBot X are attached to the robot electronically through JST connections and mechanically by two mounting screws. These design choices allow for a quick replacement of the motors when they fail. The laser cut acrylic chassis, battery holder, and charging plate, may be quickly and inexpensively replicated and replaced with the removal of a few screws and plugs.

Autonomous Charging Crucial to the continuous operations of the Robotarium is the ability of the robots to charge autonomously without direct human supervision. The GRITSBot X is equipped with a 1A Qi wireless charger attached to its backplate. By autonomously navigating to the charging transmitters fitted on the walls of the arena, the team of GRITSBot X robots can recharge their batteries in between experiments.

Over-the-air Interfacing The GRITSBot X's computing hardware was chosen specifically keeping in mind easy interfacing and rapid prototyping of software. The main micro controller on the robot is a Teensy 3.2 and the main micro processor is a Raspberry Pi Zero W. The latter enables over the air interfacing and a Linux operating system for the robot while having small form factor. It is connected via USB to the Teensy 3.2 which provides real-time commands to the motors and receives data from sensors located across the robot.

7.1.2 Safely Handling Intermittent Communication Failures

Wireless communication plays an integral role in the daily operations of the Robotarium. Indeed, the robots wirelessly receive velocity commands from the central server in real-time. However, with an increasing number of robots operating on the testbed, there is an increase in the traffic flowing through the communication network and occasional failures

are expected. These may be caused due to transmitted packets getting lost [153] or corrupted [154], saturation of the communication channels [155] or hardware failures on the robots or the host. This raises the following question: *What should a robot do in case a communication failure prevents it from receiving critical motion commands from a central decision maker?*

Many different techniques have been explored to handle communication uncertainties in multi-robot teams [156, 157, 158]. In many cases, the robots are assumed to have significant decision making capability, have sensors to maneuver around obstacles, or have knowledge about the positions of other robots. Furthermore, some developed communication recovery techniques do not provide formal collision-avoidance guarantees in case of unforeseen communication failures.

An existing technique used to handle communication failures, mentioned in [158], is to stop the robot when critical data is not received. While this behavior preserves safety, it could cause the robot to behave erratically. For example, if only intermittent velocity commands are received, the robot could move in a jerky "start-stop" fashion. This behavior was observed on the Robotarium, where failures in the communication channels prevented robots from receiving velocity or position commands which caused them to abruptly stop moving. This lead to a disruption in the coordination algorithm being executed, and affected the ability of the Robotarium to faithfully reproduce the behavior specified by the user.

Motivated by the need to alleviate such problems in general, and resolve issues with the Robotarium in particular, this section proposes a strategy that allows differential-drive robots without sensory or decision-making capabilities, to continue moving safely for a specific amount of time even when velocity commands from a central decision maker are not received. For each robot, the central decision maker computes a time horizon over which collisions with other robots are guaranteed not to occur. This is called the safe time horizon. During normal operations, the desired velocity and the safe time horizon are trans-

mitted to the robots periodically. If a robot stops receiving data due to a communication failure, it executes the last received velocity command for the duration of the last received safe time horizon. This allows the robot to continue moving in a provably collision-free manner despite having no updated information about the environment. The robot can follow this open-loop trajectory for the entire duration of the safe time horizon, beyond which it stops moving [159].

In order to calculate the safe time horizon, we first compute the set of all possible locations that can be reached by a robot within a given time (i.e., the reachable set [160, 161]). This is followed by computing the time horizon for which each robot lies outside the reachable set of other robots. But, for the differential-drive robots considered here, performing such set-membership tests is computationally expensive owing to the non-convexity of the reachable set. Consequently, the reachable set is over-approximated by enclosing it within an ellipse whose convex structure allows for simpler set-membership tests and finite representation [162]. By minimizing the area of the ellipse enclosing the convex hull of the reachable set, we obtain the best ellipsoidal over-approximation of the reachable set in terms of the accuracy and effectiveness of set-membership tests.

Reachability Analysis

Consider a dynamical system whose state evolves according to the following differential equation,

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}), \quad \mathbf{y} = h(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x_0}, \quad \mathbf{u} \in \mathcal{U},$$

where $\mathbf{x}$ is the state of the system, $\mathbf{y}$ represents the output of the system, $\mathbf{u}$ is the control input, and $\mathcal{U}$ is the set of admissible control inputs. The reachable set of outputs at time t can be defined as

$$\mathcal{R}(t; \mathbf{x_0}) = \bigcup_{\mathbf{u}(\cdot) \in \mathcal{U}} \mathcal{Y}(t, \mathbf{x_0}, \mathbf{u}(\cdot)),$$

where $\mathcal{Y}(t, \mathbf{x_0}, \mathbf{u}(\cdot))$ represents the output of the dynamical system at time t under control action $\mathbf{u}(\cdot)$ with an initial condition $\mathbf{x_0}$.

Since the Robotarium, in its current form, is populated with differential-drive mobile robots, we investigate the reachability of two-wheeled differential-drive robots with non-holonomic dynamics. For such systems, let $z = (x, y) \in \mathbb{R}^2$ denote the position of the robot in the 2D plane, and let $\phi \in [-\pi, \pi]$ denote its orientation with respect to the horizontal axis. As such, the robot is described as a point (z, ϕ) in the configuration space $\mathbb{R}^2 \times S^1$. The motion of the robots can be captured using the unicycle dynamics model:

$$\begin{aligned} &\dot{x} = v\cos(\phi),\ \dot{y} = v\sin(\phi),\ \dot{\phi} = \omega, \\ &|v| \leq 1,\ |\omega| \leq 1. \end{aligned} \tag{7.1}$$

The bounds on the linear velocity v and the angular velocity ω represent the physical limitations of the robots, and are normalized to 1 without loss of generality.

Since collisions are ultimately defined by positions rather than orientations, we consider the output of the system to be z. Relative to this output, the reachable set $\mathcal{R}(t; z_0, \phi_0)$ is the set of all positions in the 2D plane that can be reached at time t by a robot starting in the configuration (z_0, ϕ_0) at $t = 0$. For $z_0 = (0, 0), \phi_0 = 0$, denote the reachable set as $\mathcal{R}(t)$. Since the dynamics in (7.1) is drift-free, the structure of the reachable set does not depend on z_0 and ϕ_0, i.e.,

$$\mathcal{R}(t; z_0, \phi_0) = z_0 + \Pi_{\phi_0}\mathcal{R}(t), \tag{7.2}$$

where Π_{ϕ_0} is a rotation by ϕ_0. Therefore, the study of reachable sets can be restricted to the case when $z_0 = (0, 0)$ and $\phi_0 = 0$.

In order to reach the boundary of the reachable set, a robot must travel in a time-optimal manner [163]. If this was not true, a point even farther away would be reachable in the same amount of time. A closely related motion model for which the structure of time-optimal paths has been extensively studied is the *Reeds-Shepp car* [164].

The motion model of a Reeds-Shepp car is similar to (7.1), except that the set of admissible inputs is $|v| = 1, |\omega| \leq 1$. In [165], the authors compute a family of trajectories rich enough to contain a time-optimal path between any two configurations for a robot with dynamics given in (7.1). It is shown that every trajectory in this family is also time-optimal for a Reeds-Shepp car, and furthermore, this family of trajectories is sufficient for time-optimality of a Reeds-Shepp car. Another way to view this result is that, when moving in a time-optimal manner, the robot behaves like a Reeds-Shepp car. As mentioned earlier, any path reaching the boundary of the reachable set has to be time-optimal. Therefore, the reachable set for the Reeds-Shepp car and for the differential-drive robot considered here are identical. By utilizing expressions for the reachable set of a Reeds-Shepp car given in [161] and summarized in [166], Fig. 7.3 portrays the reachable set for a robot with dynamics given in (7.1).

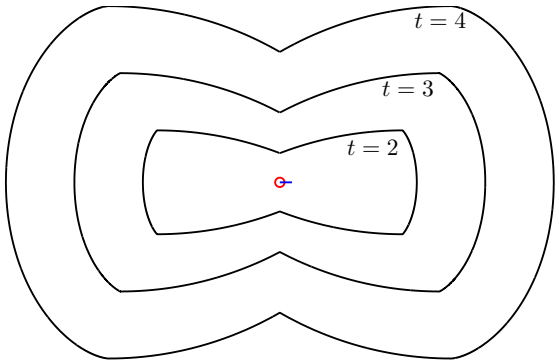


Figure 7.3: The reachable set $\mathcal{R}(t)$ of a robot with dynamics given in (7.1), is depicted for varying time horizons. The robot is represented as a circle at the center.

The complexity of performing set-membership tests with respect to the non-convex set $\mathcal{R}(t)$ increases the computational burden associated with the safe time horizon algorithm. Consequently, we next derive an ellipsoidal approximation of the reachable set.

Given any convex set $K \subset \mathbb{R}^n$, there exists a unique ellipsoid of minimum volume circumscribing it [167]. This ellipsoid is denoted as $\xi(K)$. One way to derive an ellipsoidal approximation of the reachable set would be to compute the minimum area ellipse $\xi(conv(\mathcal{R}(t)))$, where $conv(\mathcal{R}(t))$ denotes the convex hull of $\mathcal{R}(t)$. Additionally, the

derivation of analytical expressions for the ellipse $\xi(conv(\mathcal{R}(t)))$, if at all possible, will enable its efficient and fast computation in the safe time horizon algorithm.

The feasibility of the derivation of analytical expressions for the ellipses, is closely linked to the symmetry properties of the underlying set as well as the equations describing it [168]. Consequently, in order to allow for analytical solutions, we introduce a new set $\mathcal{K}(t)$ enclosing $conv(\mathcal{R}(t))$, which allows for an easier computation of $\xi(\mathcal{K}(t))$. Essentially, this enables us to swap the problem of computing $\xi(conv(\mathcal{R}(t)))$ with the simpler problem of computing $\xi(\mathcal{K}(t))$.

Furthermore, this approximation is justified by showing that the dissimilarity between $conv(\mathcal{R}(t))$ and $\mathcal{K}(t)$, as measured by the Jaccard distance metric [169], asymptotically goes to zero over time. For the sets $X, Y \subset \mathbb{R}^2$, the Jaccard distance based on the area measure is given by,

$$d_J(X,Y) = 1 - \frac{\mathbf{A}(X \cap Y)}{\mathbf{A}(X \cup Y)}, \tag{7.3}$$

where $\mathbf{A}(\cdot)$ denotes the area of the set. The following proposition formally introduces the set $\mathcal{K}(t)$.

Proposition 1. *Let $\mathcal{K}(t)$ be given by,*

$$\mathcal{K}(t) = \left\{ p = \begin{bmatrix} p_x \\ p_y \end{bmatrix} \in \mathbb{R}^2 : \|p\|_2 \leq t, |p_y| \leq \begin{cases} 1 - \cos(t), & \textit{if} \quad 0 < t \leq \pi/2 \\ t - \pi/2 + 1, & \textit{if} \quad t > \pi/2 \end{cases} \right\}. \tag{7.4}$$

Then, $conv(\mathcal{R}(t)) \subset \mathcal{K}(t)$ and

$$\lim_{t \to \infty} d_J(conv(\mathcal{R}(t)), \mathcal{K}(t)) = 0, \tag{7.5}$$

where d_J is the Jaccard distance.

Proof. Denote $\mathcal{C}^{++}(t)$ as the curved outer boundary of $conv(\mathcal{R}(t))$ in the first quadrant

(see Fig. 7.4). From [161], we know that, $\mathcal{C}^{++}(t)$ can be expressed as,

$$\mathcal{C}^{++}(t) = \left\{ p \in \mathbb{R}^2_{++} : \begin{cases} p_x = \sin\psi + \gamma\cos\psi \\ p_y = -\cos\psi + \gamma\sin\psi + 1 \end{cases} \right\}, \tag{7.6}$$

$$\text{where } \ \psi \in [0, \min(t, \pi/2)], \text{ and } \gamma = t - \psi.$$

In order to prove that $conv(\mathcal{R}(t)) \subset \mathcal{K}(t)$, we first show that all points on $\mathcal{C}^{++}(t)$ are closer to the origin than points on the outer boundary of $\mathcal{K}(t)$, which is a circular arc of radius t. Let $p(t,\psi) = (p_x(t,\psi), p_y(t,\psi))$ denote a point on the curve $\mathcal{C}^{++}(t)$. The squared l_2-norm of $p(t,\psi)$ is given as,

$$\|p(t,\psi)\|_2^2 = (t-\psi)^2 + 2 - 2\cos(\psi) + 2(t-\psi)\sin(\psi),$$

where $\psi \in [0, \min(t, \pi/2)]$. The derivative of $\|p(t,\psi)\|_2^2$ with respect to the parameter ψ, is always non-positive:

$$\frac{\partial \|p(t,\psi)\|_2^2}{\partial \psi} = 2(t-\psi)(\cos(\psi) - 1) \leq 0, \forall\ \psi \in [0, \min(t, \pi/2)]. \tag{7.7}$$

Since $p(t,0) = (t,0)$, this point lies on the circular arc of radius t. Then, the fact that $\frac{\partial \|p(t,\psi)\|_2^2}{\partial \psi} \leq 0$, implies that, as ψ increases, the curve either overlaps with the circular arc of radius t, or gets closer to the origin. This proves that, a part of the outer boundary of $conv(\mathcal{R}(t))$, represented by the curve $\mathcal{C}^{++}(t)$, can be over-approximated by a circular arc of radius t (see Fig. 7.4). Also, the definition of $p_y(t,\psi)$ gives us the y-axis bounds of $conv(\mathcal{R}(t))$:

$$\max_{\psi \in [0, \min(t,\pi/2)]} p_y(t,\psi) = \begin{cases} 1 - \cos(t), & \text{if } \ 0 < t \leq \pi/2 \\ t - \pi/2 + 1, & \text{if } \ t > \pi/2 \end{cases}. \tag{7.8}$$

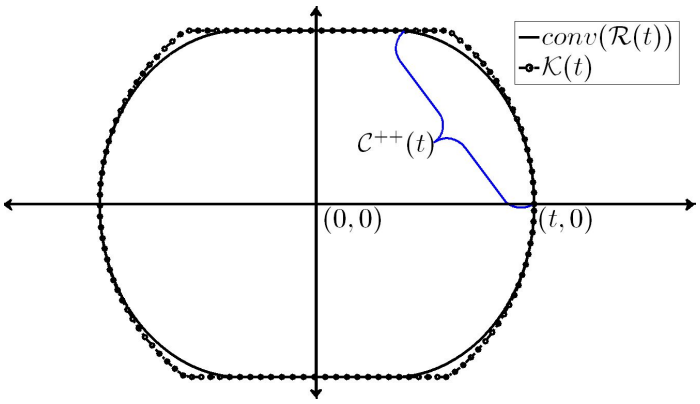


Figure 7.4: The set $conv(\mathcal{R}(t))$ is shown enclosed within $\mathcal{K}(t)$. In particular, it is illustrated how the curve $\mathcal{C}^{++}(t)$, which represents the curved boundary of $conv(\mathcal{R}(t))$ in the first quadrant, can be over-approximated by a circular arc of radius t, which forms the boundary of $\mathcal{K}(t)$ in the first quadrant.

By symmetry, these results hold in all the 4 quadrants and it follows that $conv(\mathcal{R}(t)) \subset \mathcal{K}(t)$.

Next, it is shown that, as t grows larger, the dissimilarity between the two sets asymptotically goes to zero. Due to the asymptotic nature of the result, only values of time greater than $\pi/2$ are considered. Since $conv(\mathcal{R}(t)) \subset \mathcal{K}(t)$, $conv(\mathcal{R}(t)) \cap \mathcal{K}(t) = conv(\mathcal{R}(t))$ and $conv(\mathcal{R}(t)) \cup \mathcal{K}(t) = \mathcal{K}(t)$. So, the Jaccard distance can be computed as,

$$d_J\big(conv(\mathcal{R}(t)), \mathcal{K}(t)\big) = 1 - \frac{\mathbf{A}(conv(\mathcal{R}(t)))}{\mathbf{A}(\mathcal{K}(t))}. \tag{7.9}$$

Since both the sets are symmetric with respect to the x and y axes, the areas are computed in the first quadrant alone. After integrating over the boundary of $conv(\mathcal{R}(t))$, $\mathbf{A}(conv(\mathcal{R}(t)))$ is given by,

$$\mathbf{A}(conv(\mathcal{R}(t))) = \frac{1}{48}(12\pi(t^2-1) + t(48 - 6\pi^2) + \pi^3). \tag{7.10}$$

Similarly, computing the area of $\mathcal{K}(t)$ in the first quadrant,

$$\mathbf{A}(\mathcal{K}(t)) = \frac{t^2}{2}\left(\sin^{-1}(1-\delta(t)) + \frac{1}{2}\sin(2\sin^{-1}(1-\delta(t)))\right), \tag{7.11}$$

where $\delta(t) = (\pi/2 - 1)/t$. As $t \to \infty$, the term t^2 in (7.10) dominates and $\delta(t) \to 0$ in (7.11). So, for large values of t,

$$\mathbf{A}(conv(\mathcal{R}(t))) \approx \frac{\pi}{4}t^2, \;\; \mathbf{A}(\mathcal{K}(t)) \approx \frac{\pi}{4}t^2. \tag{7.12}$$

Evaluating $\lim_{t\to\infty} d_J(conv(\mathcal{R}(t)), \mathcal{K}(t))$ using (7.12), the desired result is obtained. □

The set $\mathcal{K}(t)$ not only allows us to derive analytical expressions for the minimum area ellipse enclosing it, the asymptotic reduction in the dissimilarity between $\mathcal{K}(t)$ and $conv(\mathcal{R}(t))$ implies that, the impact of using $\xi(\mathcal{K}(t))$ instead of $\xi(conv(\mathcal{R}(t)))$ on the accuracy of set-membership tests in the safe time horizon algorithm, is minimal.

In $\mathbb{R}^n$, an ellipsoid can be represented uniquely by its center c and a positive-definite matrix H: $E(c, H) = \{x \in \mathbb{R}^n : (x-c)^T H(x-c) \leq 1\}$. According to results presented in [167], at any given time t, the minimum area ellipse circumscribing $\mathcal{K}(t)$ can be obtained by solving the following semi-infinite programming problem:

$$\begin{aligned} &\min_{c,H} - \log\ det(H) \\ &s.t.\ \ (z-c)^T H(z-c) \leq 1, \forall z \in \mathcal{K}(t). \end{aligned} \tag{7.13}$$

The rest of this section formulates an analytical solution to this semi-infinite programming problem. In Lemma 3, we utilize the symmetry properties of $\mathcal{K}(t)$ to determine the center and orientation of the ellipse $\xi(\mathcal{K}(t))$. Following this, Lemma 4 and Lemma 5 pose the semi-infinite programming problem as a convex optimization problem. By solving this, we present analytical expressions for the ellipse $\xi(\mathcal{K}(t))$ in Theorem 7.

Lemma 3. *The ellipse $\xi(\mathcal{K}(t))$ has the form $E(c, H(t))$, where $c = (0, 0)$ and $H(t) = diag(A(t), B(t))$ for some $A(t), B(t) \in \mathbb{R}$, such that $A(t) > 0, B(t) > 0, \forall t > 0$.*

Proof. For a convex set $\mathbf{K} \subset \mathbb{R}^n$, denote $\mathcal{O}(\mathbf{K})$ as a set of affine transformations which

leave the set $\mathbf{K}$ invariant:

$$\mathcal{O}(\mathbf{K}) = \{\mathbf{T}(x) = a + Px : \mathbf{T}(\mathbf{K}) = \mathbf{K}\}. \tag{7.14}$$

This set is called the automorphism group of $\mathbf{K}$. Applying results from [168], we know that, $\mathcal{O}(\mathbf{K}) \subseteq \mathcal{O}(\xi(\mathbf{K}))$. Since $\mathcal{K}(t)$ is symmetric about both the x and y axes, the transformation $\mathbf{T}(x) = -I_2 x$, where I_2 is the identity matrix, lies in $\mathcal{O}(\mathcal{K}(t))$, and hence in $\mathcal{O}(\xi(\mathcal{K}(t)))$. Furthermore, if $\mathbf{T} \in \mathcal{O}(\xi(\mathcal{K}(t)))$ then $\mathbf{T}(c) = c$. Applying this result with the transformation $\mathbf{T}(x) = -I_2 x$, we get $c = (0, 0)$.

We know from [168], that $\mathbf{T} \in \mathcal{O}(\xi(\mathcal{K}(t))) \implies P^T H(t) P = H(t)$. The structure of $H(t)$ appears by applying

$$P = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{7.15}$$

and the fact that $H(t)$ is always positive definite. □

Applying results from [167] for the case of $\mathcal{K}(t) \subset \mathbb{R}^2$, we know that, there exist contact points $\{q_i\}_1^h$, $0 < h \leq 5$ satisfying $q_i \in \partial\mathcal{K}(t) \cap \partial\xi(\mathcal{K}(t)), i = 1, \ldots, h$. Furthermore, these contact points cannot all lie in any closed halfspace whose bounding hyperplane passes through the center of $\xi(\mathcal{K}(t))$. Using these facts, the following lemma can be stated.

Lemma 4. *There are 4 contact points between $\mathcal{K}(t)$ and $\xi(\mathcal{K}(t))$.*

Proof. We seek to find the number of contact points h. Since $\mathcal{K}(t)$ and $\xi(\mathcal{K}(t))$ are symmetric about the x and y axes (see Lemma 3), and $0 < h \leq 5$, h can only take values 2 or 4. If $h = 2$, both contact points must lie on the x or y axes (otherwise symmetry in all the 4 quadrants is not possible). But, the contact points cannot lie in any closed halfspace whose bounding hyperplane passes through the center of $\xi(\mathcal{K}(t))$. Hence $h = 4$. □

Given the structure of $H(t)$ from Lemma 3, and the number of contact points from Lemma 4, the semi-infinite programming problem (7.13) will now be re-formulated as a

convex optimization problem.

Lemma 5. *The matrix $H(t) = diag(A(t), B(t))$ can be expressed as the solution of a convex optimization problem. At any given time $t > 0$, $A(t)$ is given as,*

$$A(t) = \underset{\mathbf{X}_t}{\operatorname{argmin}} \; -\log \; \mathbf{X}_t - \log \; \frac{1 - \mathbf{X}_t(t^2 - \alpha(t)^2)}{\alpha(t)^2} \qquad (7.16)$$
$$s.t. \;\; 0 < \mathbf{X}_t \leq \frac{1}{t^2}.$$

Furthermore,

$$B(t) = \frac{1 - A(t)(t^2 - \alpha(t)^2)}{\alpha(t)^2} \qquad (7.17)$$

where $\alpha(t) = \begin{cases} 1 - \cos(t), & \textit{if } \; 0 < t \leq \pi/2 \\ t - \pi/2 + 1, & \textit{if } \; t > \pi/2 \end{cases}$.

Proof. As outlined in the semi-infinite programming problem given by (7.13), we aim to derive expressions for $A(t)$ and $B(t)$ which minimize the cost function,

$$-\log \; det(H(t)) = -\log(A(t)) - \log(B(t)), \qquad (7.18)$$

subject to the constraint that $\mathcal{K}(t) \subset E((0,0), H(t))$. This constraint can be translated into the condition that the quadratic function,

$$g(y) = A(t)(t^2 - y^2) + B(t)(y^2) - 1, \qquad (7.19)$$

is non-positive for $|y| \leq \alpha(t)$.

As outlined in Lemma 4, there are a total of four contact points, one in each quadrant. Thus, there must be at least two distinct points at which $g(y)$ is zero. The roots of the quadratic function $g(y)$ must lie at $y = \pm\alpha(t)$ (in no other situation can $g(y)$ take non-

positive values in the given interval). So $g(y)$ can be alternatively expressed as,

$$g(y) = \lambda(y - \alpha(t))(y + \alpha(t)), \text{ for some } \lambda \geq 0. \tag{7.20}$$

By equating the coefficients in (7.19) and (7.20), the following constraints emerge,

$$B(t) - A(t) = \lambda \geq 0, \qquad A(t)t^2 - 1 + \lambda\alpha(t)^2 = 0.$$

Eliminating λ from the equations, the two constraints are reduced to $B(t) \geq A(t)$ and $A(t)t^2 - 1 + (B(t) - A(t))\alpha(t)^2 = 0$. Expressing $B(t)$ in terms of $A(t)$ using the second constraint, we get,

$$A(t) \leq \frac{1}{t^2},\ B(t) = \frac{1 - A(t)(t^2 - \alpha(t)^2)}{\alpha(t)^2}. \tag{7.21}$$

Substituting $B(t)$ from equation (7.21) into the cost function (7.18), and denoting $\mathbf{X}_t$ as the value taken by the function $A(\cdot)$ at time t, we obtain the optimization problem given in (7.16), whose point-wise minimizer in time gives the value of $A(t)$. □

The next theorem solves the convex optimization problem outlined above to obtain analytical expressions for the minimum area ellipses enclosing $\mathcal{K}(t)$ (see Fig. 7.5).

Theorem 7. *The minimum area ellipse $\xi(\mathcal{K}(t))$ has the form $E(c, H(t))$, where $c = (0, 0)$ and $H(t) = diag(A(t), B(t))$. $A(t)$ and $B(t)$ are given by the following expressions:*

1. *If $0 < t \leq \pi/2$, then*

$$A(t) = \frac{1}{2(t^2 - \alpha(t)^2)} \text{ and } B(t) = \frac{1}{2\alpha(t)^2}, \tag{7.22}$$

where $\alpha(t) = 1 - \cos(t)$.

2. *If* $\pi/2 < t \leq (1 + \frac{1}{\sqrt{2}})(\pi - 2)$, *then*

$$A(t) = \frac{1}{2(t^2 - \alpha(t)^2)} \text{ and } B(t) = \frac{1}{2\alpha(t)^2}, \tag{7.23}$$

where $\alpha(t) = t - \pi/2 + 1$.

3. *If* $t > (1 + \frac{1}{\sqrt{2}})(\pi - 2)$,

$$A(t) = \frac{1}{t^2} \text{ and } B(t) = \frac{1}{t^2} \tag{7.24}$$

Proof. We seek to calculate the minimizer to the convex cost function $f(\mathbf{X}_t)$ given in (7.16). In the interior of the constraint set, i.e., when $\mathbf{X}_t < \frac{1}{t^2}$, the minimizer can be found by setting the gradient to zero and solving for $\mathbf{X}_t$:

$$\begin{aligned} \nabla f(\mathbf{X}_t) &= -\frac{1}{\mathbf{X}_t} + \frac{t^2 - \alpha(t)^2}{1 - \mathbf{X}_t(t^2 - \alpha^2)} = 0, \\ \mathbf{X}_t &= \frac{1}{2(t^2 - \alpha(t)^2)} \end{aligned}$$

For this expression to satisfy the constraint, the inequality $t^2 > 2\alpha(t)^2$ must be satisfied. This is true for all values of t in the interval $(0, \pi/2]$. For $t > \pi/2$, this holds whenever the function $t^2 + 4(1 - \pi/2)t + 2(1 - \pi/2)^2$ takes non-positive values. This is true for the time interval $0 < t \leq (1 + \frac{1}{\sqrt{2}})(\pi - 2)$. For all values of $t > (1 + \frac{1}{\sqrt{2}})(\pi - 2)$, the minimum in the feasible set is achieved when $\mathbf{X}_t = 1/t^2$. Evaluating $A(t)$ from (7.16) and $B(t)$ from (7.21), we get the desired result. □

The following theorem justifies the ellipsoidal approximation by showing that the dissimilarity between $conv(\mathcal{R}(t))$ and $\xi(\mathcal{K}(t))$ asymptotically goes to zero as time grows larger.

Theorem 8. *Let* $\mathcal{R}(t)$ *denote the reachable set of a differential-drive robot with dynamics given in* (7.1)*,* $conv(\mathcal{R}(t))$ *denote its convex hull,* $\mathcal{K}(t)$ *denote an approximation of the con-*

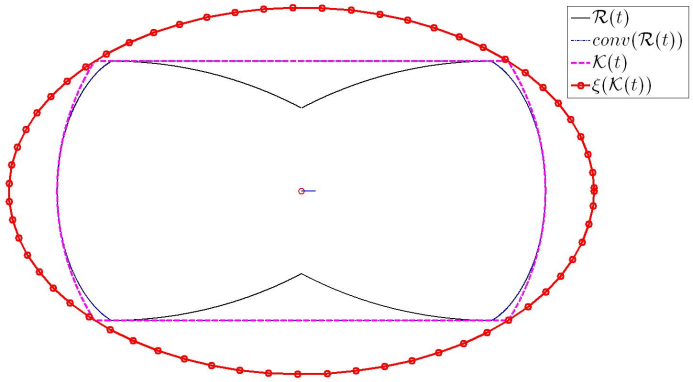


Figure 7.5: Minimum area ellipse enclosing $\mathcal{K}(t)$

vex hull as defined in Proposition 1 and let $\xi(t)$ be the minimum area ellipse circumscribing $\mathcal{K}(t)$. Then,

$$\lim_{t\to\infty} d_J(conv(\mathcal{R}(t)), \xi(t)) = 0,$$

where d_J is the Jaccard distance.

Proof. We first compute $d_J(\mathcal{K}(t), \xi(t))$. Since $\mathcal{K}(t) \subseteq \xi(t)$, $\mathcal{K}(t) \cup \xi(t) = \xi(t)$ and $\mathcal{K}(t) \cap \xi(t) = \mathcal{K}(t)$. So,

$$d_J(\mathcal{K}(t), \xi(t)) = 1 - \frac{\mathbf{A}(\mathcal{K}(t))}{\mathbf{A}(\xi(t))}. \tag{7.25}$$

Due to the asymptotic nature of the result, only values of time greater than $(1+\frac{1}{\sqrt{2}})(\pi-2)$ are considered. For these values of time, $\xi(t)$ is a circle of radius t (see Theorem 7). Also, since both $\xi(t)$ and $\mathcal{K}(t)$ are symmetric about the x and y axes, it suffices to compute areas in the first quadrant. So, $\mathbf{A}(\xi(t)) = \pi t^2/4$. From (7.12), we know that, for large values of t, $\mathbf{A}(\mathcal{K}(t)) \approx \pi t^2/4$ which is equal to the area of $\xi(t)$ in the first quadrant. Thus,

$$\lim_{t\to\infty} d_J(\mathcal{K}(t), \xi(t)) = 0. \tag{7.26}$$

Since the Jaccard distance is a metric distance, it obeys the triangle inequality,

$$d_J(conv(\mathcal{R}(t)), \xi(t)) \leq d_J(conv(\mathcal{R}(t)), \mathcal{K}(t)) + d_J(\mathcal{K}(t), \xi(t)). \tag{7.27}$$

Applying Proposition 1 and (7.26), the desired result is obtained. □

Figure 7.6 shows the evolution of $\mathcal{R}(t), conv(\mathcal{R}(t)), \mathcal{K}(t)$ and $\xi(\mathcal{K}(t))$ for different values of time. As predicted by the result, the dissimilarity between $conv(\mathcal{R}(t))$, $\mathcal{K}(t)$ and $\xi(\mathcal{K}(t))$ asymptotically goes to zero. Using the ellipsoidal approximation of the reachable

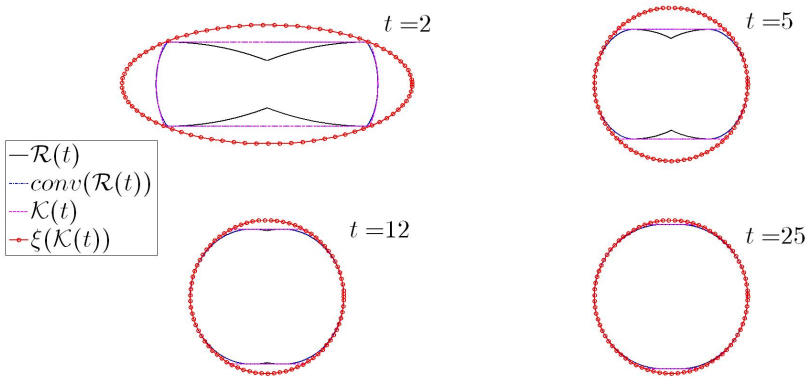


Figure 7.6: Evolution of $\mathcal{R}(t), conv(\mathcal{R}(t)), \mathcal{K}(t)$ and $\xi(\mathcal{K}(t))$ for $t = 2, 5, 12, 25$. As predicted by Theorem 8, the dissimilarity between $conv(\mathcal{R}(t)$ and $\xi(\mathcal{K}(t))$ asymptotically goes to zero. The scale for each figure is different.

set developed in this section, the next section outlines a safe time horizon based open-loop motion strategy for the robots.

A Safe Open-Loop Motion Strategy

Let $\mathcal{M} = \{1, \ldots, N\}$ be a set of N differential-drive robots, where each robot moves according to the dynamics specified in (7.1). Let $(z_i(t), \phi_i(t))$ denote the configuration of robot $i \in \mathcal{M}$ at time t. At regular intervals of time $t_k = k\delta, k \in \mathbb{N}$, the central decision maker transmits the desired velocities $u_i(t_k) = (v_i(t_k), \omega_i(t_k))$ and the corresponding safe time horizon $s_i(t_k)$ to each robot $i \in \mathcal{M}$ via a wireless communication channel. $1/\delta$ is called the update frequency.

When a robot experiences communication failure, it executes the last received velocity command repeatedly for the duration of the corresponding safe time horizon. This causes the robot to follow a circular trajectory. Let $Z_i(\mu, t_k)$ denote the position of robot i along this circular trajectory, where t_k is the time of the last received command and μ is the time elapsed since the communication failure. The expressions for $Z_i(\mu, t_k)$ are obtained by integrating (7.1) for constant velocity inputs. If $\omega_i(t_k) \neq 0$,

$$Z_i(\mu, t_k) = z_i(t_k) + \frac{v_i(t_k)}{\omega_i(t_k)} \begin{pmatrix} \sin(\omega_i(t_k)\mu + \phi_i(t_k)) - \sin(\phi_i(t_k)) \\ \cos(\phi_i(t_k)) - \cos(\omega_i(t_k)\mu + \phi_i(t_k)) \end{pmatrix} \tag{7.28}$$

and if $\omega_i(t_k) = 0$,

$$Z_i(\mu, t_k) = z_i(t_k) + \mu v_i(t_k) \begin{pmatrix} \cos(\phi_i(t_k)) \\ \sin(\phi_i(t_k)) \end{pmatrix}. \tag{7.29}$$

In order to ensure the scalability and computational tractability of the safe time horizon algorithm, we introduce the notion of a neighborhood set for each robot. To do this, the safe time horizon for each robot is upper-bounded by a pre-specified value L. This allows us to introduce the neighborhood set of robot i at time t as:

$$N_i(t) = \big\{ j \in \mathcal{M}, j \neq i : \|z_i(t) - z_j(t)\| < 2L \big\}, \tag{7.30}$$

where $\|.\|$ denotes the l_2 norm. If robot i and robot j are not neighbors, they cannot collide within the maximum safe time horizon L.

The safe time horizon $s_i(t_k)$ can be defined as,

$$s_i(t_k) = \min_{j \in N_i} s_{ij}(t_k), \tag{7.31}$$

where $s_{ij}(t_k)$ is called the pair-wise safe time and is defined as,

$$s_{ij}(t_k) = \max_{\lambda} \int_0^{\lambda} 1 \; d\lambda \tag{7.32}$$

$$\text{s.t. } Z_i(\mu, t_k) \notin \mathcal{R}(\mu; z_j(t_k), \phi_j(t_k)), \forall \mu \in [0, \lambda]$$

$$\text{and } \lambda \leq L.$$

Thus, the safe time horizon is the longest amount of time for which the trajectory of the robot after communication failure, does not intersect the reachable sets of its neighbors. But, as discussed earlier in this section, an ellipsoidal approximation of the reachable set can be used to simplify set-membership tests. Thus, the definition of $s_{ij}(t_k)$ can be modified by replacing $\mathcal{R}(\mu; z_j(t_k), \phi_j(t_k))$ with $\xi_j(\mu, t_k)$ in (7.32), where $\xi_j(\mu, t_k)$ denotes the ellipsoidal approximation corresponding to $\mathcal{R}(\mu; z_j(t_k), \phi_j(t_k))$. Next, we discuss how the safe time horizon is incorporated into the motion strategy of the robots.

As discussed earlier, the central decision maker transmits $u_i(t_k)$ and $s_i(t_k)$ to all the robots $i \in \mathcal{M}$ at regular time intervals $t_k,\ k \in \mathbb{N}$. Let c_i represent the status of the communication link of robot i:

$$c_i(t_k) = \begin{cases} 1, & \text{if } (u_i(t_k), s_i(t_k)) \text{ was received} \\ 0, & \text{if } (u_i(t_k), s_i(t_k)) \text{ was not received.} \end{cases} \tag{7.33}$$

Algorithm 6 outlines the motion strategy that robot i employs.

Figure 7.7 illustrates the rationale behind the safe time horizon algorithm. As long as the robot experiencing communication failure is outside the ellipsoidal reachable sets of its neighbors, it can safely move. The end of the safe time horizon corresponds to the time when the robot reaches the boundary of one of the ellipses. At this point, the robot stops moving.

In order to state formal safety guarantees regarding Algorithm 6, we make mild assump-

Algorithm 6 Safe Time Horizon based Open-Loop Motion Strategy

$k = 1, l = 1; u_i(0) = 0, s_i(0) = 0$
while `true` **do**
  if $c_i(t_k) = 1$ **then**
    `Execute` $u_i(t_k)$
    $l = k$
  else if $t_k - t_l < s_i(t_l)$ **then**
    `Execute` $u_i(t_l)$
  else
    `Stop Moving`
  end if
  $k = k + 1$
end while

tions on the capability of the control algorithm executing on the central decision maker. We assume that, the control algorithm ensures collision avoidance between communicating robots as well as between communicating robots and stationary obstacles. In particular, if $\exists i, j \in \mathcal{M}$ such that $c_i(t_k) = 1$ and $c_j(t_k) = 1$, then $u_i(t_k)$ and $u_j(t_k)$ guarantee that,

$$\|z_i(t_k) - z_j(t_k)\| > 0 \implies \|z_i(t_{k+1}) - z_j(t_{k+1})\| > 0. \tag{7.34}$$

Let z_O denote the position of a stationary obstacle. If $c_i(t_k) = 1$ for any $i \in \mathcal{M}$,

$$\|z_i(t_k) - z_O\| > 0 \implies \|z_i(t_{k+1}) - z_O\| > 0. \tag{7.35}$$

Utilizing these assumptions, the following theorem outlines the safety guarantees provided by Algorithm 6.

Theorem 9. *If robot i does not receive any commands from the central decision maker after time t_k, i.e., $c_i(t_k) = 1$ and $c_i(t_m) = 0\ \forall m > k$, then Algorithm 6 ensures that,*

$$\|z_i(t_k) - z_j(t_k)\| > 0 \implies \|z_i(t_k + \mu) - z_j(t_k + \mu)\| > 0,$$
$$\forall \mu \in [0, s_i(t_k)],\ \forall j \in \mathcal{M}, j \neq i.$$

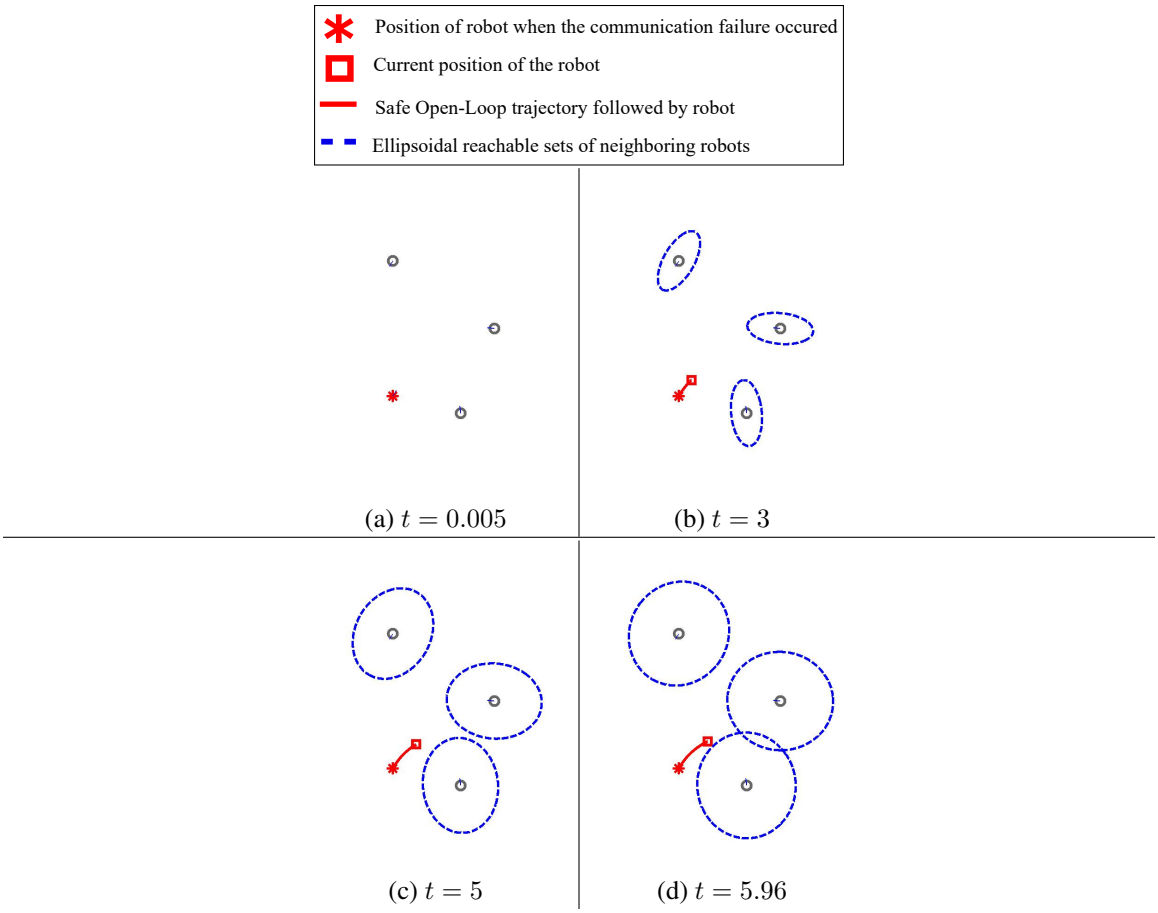


Figure 7.7: The safe time horizon represents the longest time duration for which the robot lies outside the ellipsoidal reachable sets of other robots. Thus, the robot experiencing communication failure can execute its last received velocity command for the corresponding safe time horizon and remain safe. Beyond this, the robot stops moving.

Proof. Since $\|z_i(t_k) - z_j(t_k)\| > 0 \ \forall j \in \mathcal{M}$, $s_i(t_k) > 0$. As seen in (7.30), if $j \notin N_i$, then a collision is not possible between robot i and j within the safe time horizon. Next, we consider the case when $j \in N_i$. From the definition of $s_i(t_k)$, we know that, $z_i(t_k + \mu) \notin \xi_j(\mu, t_k), \ \forall \mu \in [0, s_i(t_k)]$. Furthermore, from the definition of reachable sets, $z_j(t_k + \mu) \in \xi_j(\mu, t_k), \ \forall \mu \in [0, s_i(t_k)]$. From the previous two statements, it is clear that $z_i(t_k + \mu) \neq z_j(t_k + \mu), \ \ \forall \mu \in [0, s_i(t_k)]$. Hence,

$$\|z_i(t_k + \mu) - z_j(t_k + \mu)\| > 0, \quad \forall \mu \in [0, s_i(t_k)], \ \forall j \in N_i. \tag{7.36}$$

This completes the proof. □

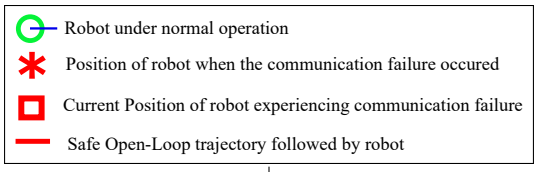
Beyond the safe time horizon, the robot stops moving, and (7.35) ensures that no collisions occur with the stationary robot. Thus, the original safety guarantee of the control algorithm is extended to situations where the robot is moving without commands from the central decision maker within the safe time horizon.

Simulations

We now present simulation results of the safe time horizon algorithm implemented on a team of 6 robots. Figure 7.8 compares the motion of the robots during a communication failure with and without the safe time horizon algorithm. A communication failure is simulated lasting from $t = 3.1s$ to $t = 8.3s$. In the case where safe time horizons are not utilized, shown by Fig. 7.8a and Fig. 7.8b, the robots experiencing communication failure abruptly stop moving, thus exhibiting a jerky motion pattern. When safe time horizons are utilized, the robots experiencing communication failure execute their last received velocity command for the duration of the safe time horizon (Fig. 7.8c and Fig. 7.8d). This allows them to keep moving during the communication failure, thereby avoiding jerky "start-stop" motion behaviors and reducing the disruption caused to the multi-robot system.

7.2 Brushbots

The *brushbots* are vibration-driven robots which employ elastic elements, referred to as brushes, to convert the energy of a vibration source into directed locomotion (see Fig. 7.9). In Chapter 6, we demonstrated how a swarm of colliding differential-drive–like brushbots equipped with no sensors can spontaneously form regions of non-uniform density. These experiments were, in part, enabled by the ability of brushbots to withstand collisions even at relatively high speeds, and in part by their ability to resolve collision-induced deadlocks due to the random nature of their movement. The use of vibration-driven robots as a platform for swarm robotics experiments has also been explored in [170], where the authors present the Kilobot, a small scale brushbot equipped with an infrared and a light sensor that enable



(a) No Safe Times: $t = 3.1s$

(b) No Safe Times: $t = 8.3s$

(c) With Safe Times: $t = 3.1s$

(d) With Safe Times: $t = 8.3s$

Figure 7.8: Comparison of the motion of robots with and without safe time horizons. Two robots experience communication failure from $t = 3.1s$ to $t = 8.3s$. In the case when safe time horizons are not used (Fig. 7.8a and Fig. 7.8b), the robots exhibit jerky motion behavior, since they abruptly stop during the communication failure. When safe time horizons are used (Fig. 7.8c and Fig. 7.8d), the robots continue moving by executing their last received velocity command for the corresponding safe time horizon, thus demonstrating the ability of the safe time horizon algorithm to effectively handle communication failures.

the execution of decentralized swarming algorithms. Collective behaviors of brushbots are investigated in [171], where the authors analyze the parameters governing the transition from a disordered motion to an organized collective motion.

With a specific focus on brushbots operating on planar smooth surfaces, the first part of this section briefly discusses some details on the locomotion modality of the brushbots. For a more in-depth and mathematical treatment, the reader is referred to [45]. Following this, we present the design of the differential-drive–like brushbot whose design simplicity and resulting robustness makes it ideal for swarm robotics applications.

7.2.1 Modeling of Vibration-based Locomotion

As in many types of locomotion, brushbots move by exploiting friction [172]. The source of energy for the system is given by vibration motors: these can be in the form of piezoelectric actuators as well as eccentric rotating mass motors. In the latter, a mass is mounted with

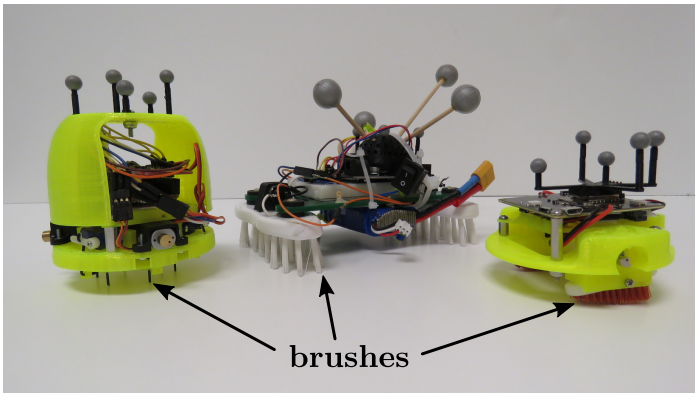
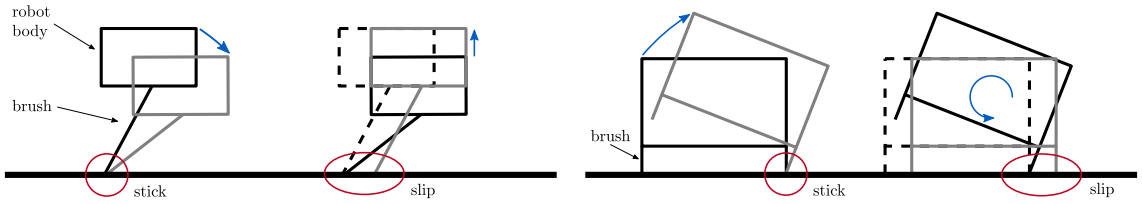


Figure 7.9: Examples of brushbots: metallic rods, brushes and toothbrushes are employed to convert energy of vibrations into directed locomotion.

an eccentricity with respect to the axle of a DC motor; when rotating, the mass produces a rotating centrifugal force which induces vibration in the robot body on which the motor is mounted. This kind of vibration motors are the ones considered here. The produced vibrations are transformed into net motion by alternating a *stick* phase and a *slip* phase.

Figure 7.10 shows the sequence of stick-slip phases for two regimes in which brushbots can operate. Figure 7.10a depicts *regime I*: on the left (stick phase), a schematic representation of the brushbot moves from the position depicted in black to the one depicted in gray. This is obtained by deforming the long flexible brush. On the right of the figure, the slip phase is shown: here the brush slides on the ground until the robot reaches the position depicted in gray. At this point, the robot has experienced a net displacement towards the right compared to the initial position (dashed contour). During these two steps, the robot body always remains parallel to the ground and the motion is achieved thanks to the deformation of the brush.

In Fig. 7.10b, *regime II* is depicted. This regime is characterized by the fact that stiff short brushes do not deform, but rather act as pivot points for the robot to rotate. During the stick phase (on the left of the figure), the robot body rotates about a pivot point, whereas, in the slip phase (on the right), the robot body rotates back to its initial orientation while sliding towards the right. The two regimes can be more or less predominant depending on the physical characteristics of the robot. For a detailed dynamic model of the brushbots



(a) *Regime I*: robots operating in this regime are characterized by a high flexibility of the brushes. The vibrations are modeled as alternating vertical forces which deform the brushes during the stick phase and pull the robot up during the slip phase. At this moment, the friction reduces proportionally to the reduction of normal force, allowing the brush to slide and the robot to step forward.

(b) *Regime II*: light robots with stiff brushes can operate in this regime. As the flexibility of the brushes cannot be exploited, locomotion is achieved by the sequence of two rigid body rotations happening in sequence. The first one in the stick phase and the second in the slip phase, during which the brushbot experiences a net displacement.

Figure 7.10: Two regimes of operation of the brushbot: locomotion is achieved by exploiting vibrations in two different ways, depending on the physical characteristics of the robot.

operating in the two described regimes, see [45].

7.2.2 Brushbots in Swarm Robotics

Leveraging the knowledge of regimes in which the brushbots can operate, this section presents the design of a simple and robust brushbot. The time to build the brushbot that is presented in this section is, in fact, less than three hours, which include 3D printing, soldering and preparation of the brushes. The unit cost is kept below 30$, which can be significantly reduced if the number of robots to produce increases. The design of simple, easy and fast-to-build, robust brushbots makes it very appealing and suitable for swarm robotics applications, which deals with the coordination and interactions of a large number of robots.

Figure 7.11 shows the schematic design of the brushbot presented in this section: it is a differential-drive-like brushbot, which consists of two sets of brushes mounted parallel to each other on two opposite sides of a rigid platform. Two motors (shown in Fig. 7.12) are mounted on top of each of the brushes. The motion of the differential-drive brushbot can be described as follows:

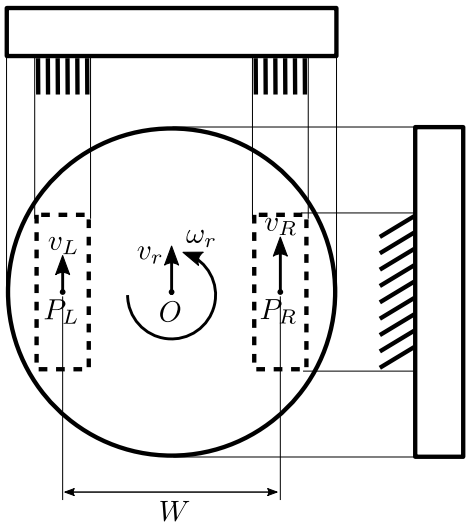
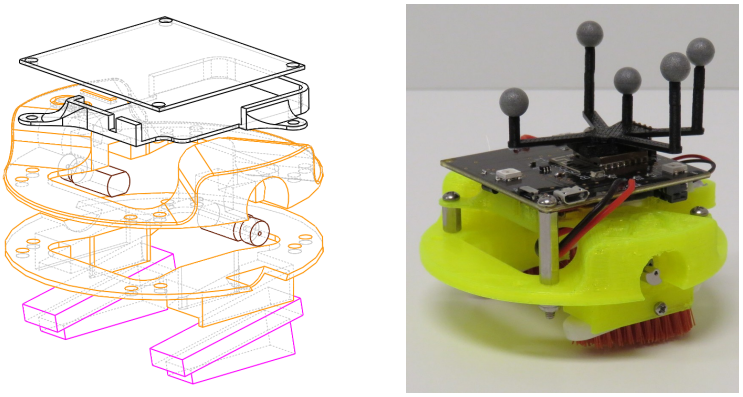


Figure 7.11: Differential-drive-like brushbot: two sets of brushes are mounted on the opposite sides of the robot body. Desired linear and angular velocities of the robot body can be achieved by varying the speed of vibration motors mounted on top of each set of brushes.



(a) (b)

Figure 7.12: Differential-drive-like brushbot. In Fig. 7.12a, the exploded view of the CAD model shows, from top to bottom: PCB and battery support (black), top body (orange), vibration motors (brown), bottom body (orange), brushes (purple). Figure 7.12b shows a 3D printed prototype of the brushbot: infrared-reflective balls are mounted on top for tracking its pose.

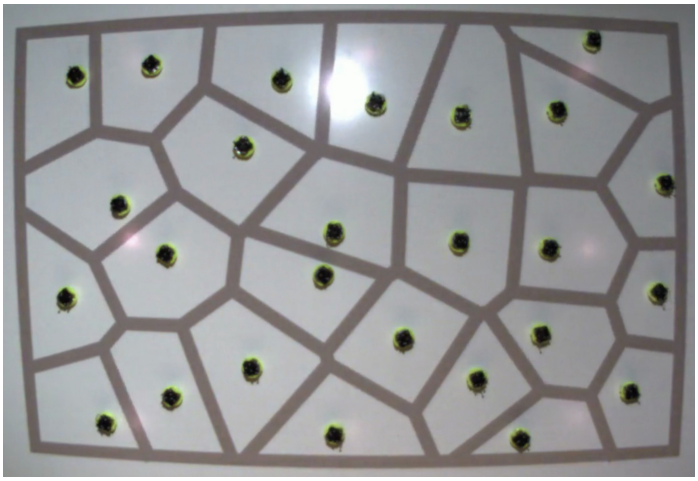


Figure 7.13: A swarm of 26 differential-drive-like brushbots (like the one shown in Fig. 7.12) performing coverage control [7]. The boundaries of the Voronoi cells corresponding to each robot are shown in grey.

- actuating the left motor produces a velocity, indicated as v_L in Fig. 7.11, at the left set of brushes (as described by regime I)
- at the same time, due to the actuation of the left motor, the robot pivots about the right set of brushes, which induces a net angular velocity, ω_r, of the robot (as predicted by regime II)

The rigid body dynamics of the robot are then:

$$v_{L,R} = v_r - \omega_r \times (P_{L,R} - O) \tag{7.37}$$

From (7.37), the expressions of linear and angular velocities of the differential-drive-like brushbot can be obtained:

$$v_r = \frac{v_L + v_R}{2}, \quad \omega_r = \frac{v_R - v_L}{W}, \tag{7.38}$$

where, with abuse of notation, all symbols have been used to denote the signed magnitudes of the vector quantities used in (7.37), their directions being given in Fig. 7.11.

Considering the brushbot as a unicycle, one can use controllers such as the one devel-

oped in [173] to implement swarm-robotics algorithms. As an example, here we consider the coverage control algorithm developed in [7]. Figure 7.13 shows 26 differential-drive-like brushbots running the coverage-control algorithm in order to evenly spread out over the shown rectangular domain. The boundaries of the Voronoi cells of the brushbots are depicted as grey lines.

Observation 6. *As pointed out above, the advantages related to design simplicity and ease of assembly of the brushbots presented in this section, lead to robustness properties which are desirable for swarm robotics applications. In particular, the fact that the vibration motors do not have to be directly coupled with the brushes lets us design the robots in such a way that all the moving parts are contained in the convex hull of the robot main body (as can be seen in Fig. 7.12). This allows brushbots to tolerate collisions, even of significant magnitude, with other robots and obstacles present in the environment.*

7.3 Conclusions

In this chapter, we outlined the design and development of two swarm robotics platforms which were used throughout this thesis to validate the developed encounter-based behaviors. The Robotarium has been designed to reliably and reproducably execute experiments submitted from users all across the world. The testbed is populated by a team of wheeled differential drive robots, which can autonomously recharge their batteries and have been specifically designed for reliability and ease of maintenance.

A swarm of brushbots, whose simple design makes them robust to collisions even at high speeds, were used to validate the inhomogeneous spatial density behaviors developed in Chapter 6. The design and locomotion principles of these robots, which achieve directed motion by vibrating flexible bristles, was highlighted in Section 7.2.

CHAPTER 8
CONCLUSIONS AND FUTURE DIRECTIONS

As robot swarms move from curated laboratory environments to the real world, deployment scenarios can present significant challenges to the coordinated actions of large swarms, e.g., by limiting their capabilities to sense and react to robots around them, or precluding the presence of a central coordinator. One need not look far to be convinced that leaderless and short range interactions among individuals can lead to useful behaviors—nature presents a plethora of evidence to this effect. Local interactions, often manifested as proximity encounters among individuals (or particles) in a living (or non-living) collective, play a crucial role in the emergence of phenomena like directed motion, predator avoidance, density regulation, and phase separation.

This thesis demonstrates that inter-robot encounters can act as a useful modality via which minimally equipped robot swarms can achieve desired objectives. Chapters 3, 4, and 5 envision a situation where individual robots can explicitly measure the proximity inter-robot encounters occurring as a by-product of their motion in the environment. In order to allow such measurements to be useful for the task at hand, we develop mathematical models to characterize the encounters as a function of swarm motion and density parameters. We use ideas from stochastic geometry and the molecular physics literature to develop a series of models which increase in their fidelity to systematically analyze real-world inter-robot encounters. The efficacy of these models is demonstrated by considering a different motivating application in each chapter.

In Chapter 3, we demonstrated how the information obtained from physical inter-robot collisions can allow robots in a swarm to localize themselves under certain conditions. This highlighted the ability of collisions to facilitate useful behaviors in certain classes of robot swarms where the physical impacts of such collisions can be tolerated. In Chapters

4 and 5, proximity encounters replaced physical collisions as the detected event—robots only needed to sense the presence of other robots around them, for e.g., during collision avoidance maneuvers. Inspired by the ability of ants to regulate densities and avoid clog formations in narrow tunnels, Chapter 4 developed a decentralized mechanism by which robots could regulate the density in a region to achieve a desired trade-off between the deployed team size and the productivity of the swarm at a distributed collection task. In particular, we provided an analytical characterization of the time effectiveness of the robots at performing the tasks as opposed to navigating around other robots.

One of the key features of the developed leaderless encounter-based algorithms is the robustness to failures among the individual robots—since no single robot or entity is in charge, the functioning of the swarm is not affected by failures among a few robots. Chapter 5 demonstrated this via a motivating application, where a swarm of robots autonomously distribute themselves among a set of tasks using inter-robot encounters as the primary interaction modality. To this end, we developed a *decentralized closed-loop task allocation mechanism*, where the swarm regulates to the desired allocation while also regulating the rate of task switching, with the ultimate aim of converging faster to the desired allocation and facilitating uninterrupted task execution by the robots. Chapter 5 also leverages ideas from the Enskog theory of high density gases to develop an improved encounter model; one which explicitly accounts for the non-zero footprint area occupied by each robot.

While the developed task allocation framework is decentralized, it relies on the uniformly ergodic motion of the robots in the environment irrespective of the task they are performing. One possible direction of future work would be to explore situations where individual tasks generate varying (and known) spatial distributions of robots, which could be accounted for in the developed encounter model. This would result in the applicability of the algorithm for a wider range of tasks. In such a scenario, the encounter model would likely have to explicitly account for gradients in the density distribution of robots—enabling further novel applications. For instance, how could the density regulation tech-

nique developed in Chapter 4 be generalized to account for local crowding of robots in an environment, e.g., when precipitated by the presence of obstacles or corners?

In Chapter 6, we switched gears and demonstrated how the persistent collisional interactions occurring among robots can *implicitly* lead to the predictable formation of non-uniform density regions in the environment. By viewing the swarm as a far-from-equilibrium physical system composed of self-propelled particles, we analytically formulated conditions on the motion parameters of the robots to achieve such behaviors. Results from *motility-induced phase separation* illustrated how the density-dependent speed profile of the robots ultimately resulted in the formation of high-density aggregates which coexist with regions of lower density. This algorithm was deployed on a swarm of vibration-driven robots called brushbots, which presented an ideal platform to validate collision-based interactions among robots.

This phase separation-based framework was formulated for scenarios where robots with extremely limited sensing and communication capabilities need to spread out in the environment, e.g., to perform a surveillance task, while requiring physical proximity to exchange information. A possible extension of the ideas developed in Chapter 6 could involve analyzing the spread of information across a swarm with and without phase separating behaviors—to study its efficacy while balancing spatially conflicting objectives. Deployment of such algorithms on micro-robotics platforms can help us better understand how statistical mechanics abstractions can enable novel swarm robotics applications.

To conclude, this thesis presented bio-physics inspired approaches for the decentralized coordination of robot swarms. Mean-field approximations allowed for a systematic analysis of inter-robot encounters, which were leveraged in illustrative applications presented throughout the thesis.

REFERENCES

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[2] Y. Tan and Z.-y. Zheng, "Research advance in swarm robotics," *Defence Technology*, vol. 9, no. 1, pp. 18–39, 2013.

[3] L. Iocchi, D. Nardi, and M. Salerno, "Reactivity and Deliberation: A Survey on Multi-Robot Systems," in *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, Springer Berlin Heidelberg, 2001, pp. 9–32, ISBN: 978-3-540-44568-5.

[4] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.

[5] W. Savoie, T. A. Berrueta, Z. Jackson, A. Pervan, R. Warkentin, S. Li, T. D. Murphey, K. Wiesenfeld, and D. I. Goldman, "A robot made of robots: Emergent transport and control of a smarticle ensemble," *Science Robotics*, vol. 4, no. 34, 2019. eprint: `https://robotics.sciencemag.org/content/4/34/eaax4316.full.pdf`.

[6] 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.

[7] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Transactions on robotics and Automation*, vol. 20, no. 2, pp. 243–255, 2004.

[8] E. Şahin, "Swarm Robotics: From Sources of Inspiration to Domains of Application," in *Swarm Robotics*, E. Şahin and W. M. Spears, Eds., Springer Berlin Heidelberg, 2005, pp. 10–20, ISBN: 978-3-540-30552-1.

[9] I. D. Couzin, "Collective cognition in animal groups," *Trends in cognitive sciences*, vol. 13, no. 1, pp. 36–43, 2009.

[10] J. Buhl, D. J. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, "From disorder to order in marching locusts," *Science*, vol. 312, no. 5778, pp. 1402–1406, 2006.

[11] J Aguilar, D Monaenkova, V Linevich, W Savoie, B Dutta, H.-S. Kuan, M. Betterton, M. Goodisman, and D. Goldman, "Collective clog control: Optimizing traffic flow in confined biological and robophysical excavation," *Science*, vol. 361, no. 6403, pp. 672–677, 2018.

[12] D. M. Gordon, "Behavioral flexibility and the foraging ecology of seed-eating ants," *The American Naturalist*, vol. 138, no. 2, pp. 379–411, 1991.

[13] N. R. Franks and J.-L. Deneubourg, "Self-organizing nest construction in ants: individual worker behaviour and the nest's dynamics," *Animal Behaviour*, vol. 54, no. 4, pp. 779–796, 1997.

[14] D. Chandler, *Introduction to modern statistical mechanics*. New York: Oxford University Press, 1987.

[15] M. C. Marchetti, J.-F. Joanny, S. Ramaswamy, T. B. Liverpool, J. Prost, M. Rao, and R. A. Simha, "Hydrodynamics of soft active matter," *Reviews of Modern Physics*, vol. 85, no. 3, p. 1143, 2013.

[16] N.S.F, *Condensed-matter and materials physics: The science of world around us*, 2010.

[17] J. M. Rieser, P. E. Schiebel, A. Pazouki, F. Qian, Z. Goddard, K. Wiesenfeld, A. Zangwill, D. Negrut, and D. I. Goldman, "Dynamics of scattering in undulatory active collisions," *Physical Review E*, vol. 99, no. 2, p. 022 606, 2019.

[18] C. Bechinger, R. Di Leonardo, H. Löwen, C. Reichhardt, G. Volpe, and G. Volpe, "Active particles in complex and crowded environments," *Reviews of Modern Physics*, vol. 88, no. 4, p. 045 006, 2016.

[19] C. Wong, E. Yang, X.-T. Yan, and D. Gu, "An overview of robotics and autonomous systems for harsh environments," in *2017 23rd International Conference on Automation and Computing (ICAC)*, IEEE, 2017, pp. 1–6.

[20] J. J. Abbott, Z. Nagy, F. Beyeler, and B. J. Nelson, "Robotics in the small, part i: Microbotics," *IEEE Robotics & Automation Magazine*, vol. 14, no. 2, pp. 92–103, 2007.

[21] D. Goldberg and M. J. Mataric, "Interference as a tool for designing and evaluating multi-robot controllers," in *Aaai/iaai*, 1997, pp. 637–642.

[22] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Collisions as Information Sources in Densely Packed Multi-Robot Systems Under Mean-Field Approximations," in *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, 2017.

[23] ——, "Localization in Densely Packed Swarms Using Interrobot Collisions as a Sensing Modality," *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 21–34, 2019.

[24] S. Mayya, S. Wilson, and M. Egerstedt, "Closed-loop task allocation in robot swarms using inter-robot encounters," *Swarm Intelligence*, pp. 1–29, 2019.

[25] S. Mayya, P. Pierpaoli, and M. Egerstedt, "Voluntary retreat for decentralized interference reduction in robot swarms," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 9667–9673.

[26] D. M. Gordon, *Ant encounters: interaction networks and colony behavior*. Princeton University Press, 2010.

[27] N. Gravish, G. Gold, A. Zangwill, M. A. Goodisman, and D. I. Goldman, "Glass-like dynamics in confined and congested ant traffic," *Soft matter*, vol. 11, no. 33, pp. 6552–6561, 2015.

[28] S. C. Pratt, "Quorum sensing by encounter rates in the ant Temnothorax albipennis," *Behavioral Ecology*, vol. 16, no. 2, pp. 488–496, 2005.

[29] S. Kernbach, R. Thenius, O. Kernbach, and T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system," *Adaptive Behavior*, vol. 17, no. 3, pp. 237–259, 2009.

[30] T. Schmickl, R. Thenius, C. Moeslinger, G. Radspieler, S. Kernbach, M. Szymanski, and K. Crailsheim, "Get in touch: cooperative decision making based on robot-to-robot collisions," *Autonomous Agents and Multi-Agent Systems*, vol. 18, no. 1, pp. 133–155, 2009.

[31] M. Mote, J. P. Afman, and E. Feron, "Robotic trajectory planning through collisional interaction," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 1144–1149.

[32] J. M. O'Kane and S. M. LaValle, "Localization with limited sensing," *IEEE Transactions on Robotics*, vol. 23, no. 4, p. 704, 2007.

[33] S. N. Chiu, D. Stoyan, W. S. Kendall, and J. Mecke, *Stochastic geometry and its applications*. John Wiley & Sons, 2013.

[34] F. Reif, ser. McGraw-Hill series in fundamentals of physics. Waveland Press, 2009, ISBN: 9781577666127.

[35] E. Castello, T. Yamamoto, F. Dalla Libera, W. Liu, A. F. Winfield, Y. Nakamura, and H. Ishiguro, "Adaptive foraging for simulated and real robotic swarms: The

dynamical response threshold approach," *Swarm Intelligence*, vol. 10, no. 1, pp. 1–31, 2016.

[36] R. Beckers, O. E. Holland, and J.-L. Deneubourg, "Fom local actions to global tasks: Stigmergy and collective robotics," in *Prerational Intelligence: Adaptive Behavior and Intelligent Systems Without Symbols and Logic*, Springer, 2000, pp. 1008–1022.

[37] A. Martinoli, A. J. Ijspeert, and L. M. Gambardella, "A probabilistic model for understanding and comparing collective aggregation mechanisms," in *European Conference on Artificial Life*, Springer, 1999, pp. 575–584.

[38] A. T. Hayes, "How many robots? group size and efficiency in collective search tasks," in *Distributed Autonomous Robotic Systems 5*, Springer, 2002, pp. 289–298.

[39] S. Chapman and T. G. Cowling, *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press, 1970.

[40] P. Cutchis, H Van Beijeren, J. Dorfman, and E. Mason, "Enskog and van der Waals play hockey," *American Journal of Physics*, vol. 45, no. 10, pp. 970–977, 1977.

[41] D. M. Gordon and N. J. Mehdiabadi, "Encounter rate and task allocation in harvester ants," *Behavioral Ecology and Sociobiology*, vol. 45, no. 5, pp. 370–377, 1999.

[42] J. Elston, M. Stachura, E. Frew, and U. Herzfeld, "Toward Model Free Atmospheric Sensing by Aerial Robot Networks in Strong Wind Fields," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, May 2009, pp. 369–374.

[43] Y. Diaz-Mercado and M. Egerstedt, "Multi-robot mixing using braids," in *52nd IEEE Conference on Decision and Control*, 2013, pp. 2001–2005.

[44] S. Mayya, G. Notomista, D. Shell, S. Hutchinson, and M. Egerstedt, "Non-uniform robot densities in vibration driven swarms using phase separation theory," *arXiv e-prints*, arXiv:1902.10662, Accepted for Publication, IROS 2019.

[45] G. Notomista, S. Mayya, A. Mazumdar, S. Hutchinson, and M. Egerstedt, "A Study of a Class of Vibration-Driven Robots: Modeling, Analysis, Control and Design of the Brushbot," *arXiv e-prints*, arXiv:1902.10830, Accepted for Publication, IROS 2019.

[46] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: A remotely accessible swarm robotics research testbed," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 1699–1706.

[47] J. Van Den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *IEEE International Conference on Robotics and Automation (ICRA), 2011*, IEEE, 2011, pp. 3475–3482.

[48] R. C. Arkin, *Behavior-based robotics*. MIT press, 1998.

[49] M. Schneider-Fontán and M. J. Mataric, "A study of territoriality: The role of critical mass in adaptive task division," in *From animals to animats IV*, Citeseer, 1996.

[50] K. Sugawara and M. Sano, "Cooperative acceleration of task performance: Foraging behavior of interacting multi-robots system," *Physica D: Nonlinear Phenomena*, vol. 100, no. 3-4, pp. 343–354, 1997.

[51] R. T. Vaughan, G. S. Sukhatme, and M. J. Mataric, "Go ahead, make my day: Robot conflict resolution by aggressive competition," in *In Proc. of the Intl. Conf. on Simulation of Adaptive Behavior (SAB)*, Citeseer, 2000.

[52] Y. Zhang and R. Vaughan, "Ganging up: Team-based aggression expands the population/performance envelope in a multi-robot system," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006.*, IEEE, 2006, pp. 589–594.

[53] R. C. Arkin, T. Balch, and E. Nitz, "Communication of behavorial state in multi-agent retrieval tasks," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, IEEE, 1993, pp. 588–594.

[54] D. Goldberg, "Evaluating the dynamics of agent-environment interaction," University of Southern California Los Angeles United States, Tech. Rep., 2001.

[55] D. A. Shell and M. J. Mataric, "On foraging strategies for large-scale multi-robot systems," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2006, pp. 2717–2723.

[56] E. Ostergaard, G. Sukhatme, and M Mataric, "Emergent bucket brigading," in *Autonomous Agents*, vol. 37, 2001, pp. 2219–2223.

[57] G. Pini, A. Brutschy, C. Pinciroli, M. Dorigo, and M. Birattari, "Autonomous task partitioning in robot foraging: An approach based on cost estimation," *Adaptive behavior*, vol. 21, no. 2, pp. 118–136, 2013.

[58] A. L.R. T. Vaughan, "Adaptive multi-robot bucket brigade foraging," *Artificial Life*, vol. 11, p. 337, 2008.

[59] T. H. Labella, M. Dorigo, and J.-L. Deneubourg, "Division of labor in a group of robots inspired by ants' foraging behavior," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, vol. 1, no. 1, pp. 4–25, 2006.

[60] S. Bullock, R. Crowder, and L. Pitonakova, "Task allocation in foraging robot swarms: The role of information sharing," in *Artificial Life Conference Proceedings 13*, MIT Press, 2016, pp. 306–313.

[61] W. Liu, A. F. Winfield, J. Sa, J. Chen, and L. Dou, "Towards energy optimization: Emergent task allocation in a swarm of foraging robots," *Adaptive behavior*, vol. 15, no. 3, pp. 289–305, 2007.

[62] T. S. Dahl, M. Matarić, and G. S. Sukhatme, "Multi-robot task allocation through vacancy chain scheduling," *Robotics and Autonomous Systems*, vol. 57, no. 6-7, pp. 674–687, 2009.

[63] M. J. Krieger, J.-B. Billeter, and L. Keller, "Ant-like task allocation and recruitment in cooperative robots," *Nature*, vol. 406, no. 6799, p. 992, 2000.

[64] A. Rosenfeld, G. A. Kaminka, and S. Kraus, "A study of scalability properties in robotic teams," in *Coordination of large-scale multiagent systems*, Springer, 2006, pp. 27–51.

[65] K. Lerman and A. Galstyan, "Mathematical model of foraging in a group of robots: Effect of interference," *Autonomous Robots*, vol. 13, no. 2, pp. 127–141, 2002.

[66] Y. Khaluf, C. Pinciroli, G. Valentini, and H. Hamann, "The impact of agent density on scalability in collective systems: Noise-induced versus majority-based bistability," *Swarm Intelligence*, vol. 11, no. 2, pp. 155–179, 2017.

[67] H. Hamann, "Superlinear scalability in parallel computing and multi-robot systems: Shared resources, collaboration, and network topology," in *International Conference on Architecture of Computing Systems*, Springer, 2018, pp. 31–42.

[68] M. Feinberg, "The existence and uniqueness of steady states for a class of chemical reaction networks," *Archive for Rational Mechanics and Analysis*, vol. 132, no. 4, pp. 311–370, 1995.

[69] N. Correll and A. Martinoli, "Modeling self-organized aggregation in a swarm of miniature robots," in *IEEE 2007 International Conference on Robotics and Automation Workshop on Collective Behaviors inspired by Biological and Biochemical Systems*, 2007.

[70] M. A. Hsieh, Á. Halász, S. Berman, and V. Kumar, "Biologically inspired redistribution of a swarm of robots among multiple sites," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 121–141, 2008.

[71] A. Martinoli, K. Easton, and W. Agassounon, "Modeling swarm robotic systems: A case study in collaborative distributed manipulation," *The International Journal of Robotics Research*, vol. 23, no. 4-5, pp. 415–436, 2004.

[72] L. Matthey, S. Berman, and V. Kumar, "Stochastic strategies for a swarm robotic assembly system," in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 1953–1958.

[73] T. P. Pavlic, S. Wilson, G. P. Kumar, and S. Berman, "Control of stochastic boundary coverage by multirobot systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, p. 034 504, 2015.

[74] S. Wilson, T. P. Pavlic, G. P. Kumar, A. Buffin, S. C. Pratt, and S. Berman, "Design of ant-inspired stochastic control policies for collective transport by robotic swarms," *Swarm Intelligence*, vol. 8, no. 4, pp. 303–327, 2014.

[75] H. Hamann and H. Wörn, "A framework of space–time continuous models for algorithm design in swarm robotics," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 209–239, 2008.

[76] K. Elamvazhuthi, "Controllability and stabilization of kolmogorov forward equations for robotic swarms," PhD thesis, Arizona State University, 2019.

[77] K. Elamvazhuthi, M. Kawski, S. Biswal, V. Deshmukh, and S. Berman, "Mean-field controllability and decentralized stabilization of markov chains (accepted)," in *IEEE Conference on Decision and Control (CDC)*, 2017.

[78] K. Elamvazhuthi, C. Adams, and S. Berman, "Coverage and field estimation on bounded domains by diffusive swarms," in *2016 IEEE 55th Conference on Decision and Control (CDC)*, IEEE, 2016, pp. 2867–2874.

[79] S. Berman, Á. Halász, M. A. Hsieh, and V. Kumar, "Optimized stochastic policies for task allocation in swarms of robots," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 927–937, 2009.

[80] N. Shlyakhov, I. Vatamaniuk, and A. Ronzhin, "Survey of methods and algorithms of robot swarm aggregation," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 803, 2017, p. 012 146.

[81] M. Kumar, D. P. Garg, and V. Kumar, "Segregation of heterogeneous units in a swarm of robotic agents," *IEEE transactions on automatic control*, vol. 55, no. 3, pp. 743–748, 2010.

[82] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Gross, "Clustering objects with robots that do not compute," in *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, International Foundation for Autonomous Agents and Multiagent Systems, 2014, pp. 421–428.

[83] R. Groß and M. Dorigo, "Self-assembly at the macroscopic scale," *Proceedings of the IEEE*, vol. 96, no. ARTICLE, pp. 1490–1508, 2008.

[84] A. Martinoli, A. J. Ijspeert, and F. Mondada, "Understanding collective aggregation mechanisms: From probabilistic modelling to experiments with real robots," *Robotics and Autonomous Systems*, vol. 29, no. 1, pp. 51–63, 1999.

[85] M. Gauci, J. Chen, W. Li, T. J. Dodd, and R. Groß, "Self-organized aggregation without computation," *International Journal of Robotics Research*, vol. 33, no. 8, pp. 1145–1161, Jul. 2014.

[86] J. Chen, M. Gauci, M. J. Price, and R. Groß, "Segregation in swarms of e-puck robots based on the Brazil nut effect," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Jun. 2012, pp. 163–170.

[87] P. Mitrano, J. Burklund, M. Giancola, and C. Pinciroli, "A minimalistic approach to segregation in robot swarms," *arXiv preprint arXiv:1901.10423*, 2019.

[88] D. S. Brown, R. Turner, O. Hennigh, and S. Loscalzo, "Discovery and exploration of novel swarm behaviors given limited robot capabilities," in *Distributed Autonomous Robotic Systems*, Springer, 2018, pp. 447–460.

[89] W. Savoie, S. Cannon, J. J. Daymude, R. Warkentin, S. Li, A. W. Richa, D. Randall, and D. I. Goldman, "Phototactic supersmarticles," *Artificial Life and Robotics*, vol. 23, no. 4, pp. 459–468, 2018.

[90] Y. Mulgaonkar, G. Cross, and V. Kumar, "Design of small, safe and robust quadrotor swarms," in *IEEE International Conference on Robotics and Automation (ICRA), 2015*, IEEE, 2015, pp. 2208–2215.

[91] T. Alam, L. Bobadilla, and D. A. Shell, "Space-efficient filters for mobile robot localization from discrete limit cycles," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 257–264, 2018.

[92] P. E. Schiebel, J. M. Rieser, A. M. Hubbard, L. Chen, D. Z. Rocklin, and D. I. Goldman, "Mechanical diffraction reveals the role of passive dynamics in a slith-

ering snake," *Proceedings of the National Academy of Sciences*, vol. 116, no. 11, pp. 4798–4803, 2019.

[93] J. C. Spagna, D. I. Goldman, P.-C. Lin, D. E. Koditschek, and R. J. Full, "Distributed mechanical feedback in arthropods and robots simplifies control of rapid running on challenging terrain," *Bioinspiration & biomimetics*, vol. 2, no. 1, p. 9, 2007.

[94] D. M. Gordon, "The organization of work in social insect colonies," *Nature*, vol. 380, p. 14, 1996.

[95] J. K. Parrish and W. M. Hamner, *Animal groups in three dimensions: how species aggregate*. Cambridge University Press, 1997.

[96] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical review letters*, vol. 75, no. 6, p. 1226, 1995.

[97] M. E. Cates and J. Tailleur, "Motility-induced phase separation," *Annu. Rev. Condens. Matter Phys.*, vol. 6, no. 1, pp. 219–244, 2015.

[98] P. Atkins and J. De Paula, *Physical chemistry for the life sciences*. Oxford University Press, USA, 2011.

[99] J. Stenhammar, A. Tiribocchi, R. J. Allen, D. Marenduzzo, and M. E. Cates, "Continuum theory of phase separation kinetics for active brownian particles," *Physical review letters*, vol. 111, no. 14, p. 145 702, 2013.

[100] Y. Fily, S. Henkes, and M. C. Marchetti, "Freezing and phase separation of self-propelled disks," *Soft matter*, vol. 10, no. 13, pp. 2132–2140, 2014.

[101] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition.," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[102] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on information theory*, vol. 20, no. 2, pp. 284–287, 1974.

[103] J. A. Ludwig and J. F. Reynolds, *Statistical ecology: a primer in methods and computing*. John Wiley & Sons, 1988, vol. 1.

[104] P. Elliot, J. C. Wakefield, N. G. Best, D. Briggs, *et al.*, *Spatial epidemiology: methods and applications.* Oxford University Press, 2000.

[105] G. J. Babu and E. D. Feigelson, *Astrostatistics*. CRC Press, 1996, vol. 3.

[106] H. Jaleel and M. Egerstedt, "Sleep scheduling of wireless sensor networks using hard-core point processes," in *American Control Conference (ACC), 2013*, IEEE, 2013, pp. 788–793.

[107] A. Baddeley, I. Bárány, and R. Schneider, "Spatial point processes and their applications," *Lecture Notes In Mathematics-Springer-Verlag*, vol. 1892, p. 1, 2007.

[108] P. J. Diggle, J. Besag, and J. T. Gleaves, "Statistical analysis of spatial point patterns by means of distance methods," *Biometrics*, pp. 659–667, 1976.

[109] J. F. C. Kingman, *Poisson processes*. Wiley Online Library, 1993.

[110] D. J. Daley and D. Vere-Jones, *An introduction to the theory of point processes: volume II: general theory and structure*. Springer Science & Business Media, 2007.

[111] B. Matérn, *Spatial variation*. Springer Science & Business Media, 2013, vol. 36.

[112] N. E. Du Toit and J. W. Burdick, "Probabilistic collision checking with chance constraints," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 809–815, 2011.

[113] S. Patil, J. Van Den Berg, and R. Alterovitz, "Estimating probability of collision for safe motion planning under Gaussian motion and sensing uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA), 2012*, 2012, pp. 3238–3244.

[114] A. Lambert, D. Gruyer, and G. Saint Pierre, "A fast Monte Carlo algorithm for collision probability estimation," in *10th International Conference on Control, Automation, Robotics and Vision, 2008*, IEEE, 2008, pp. 406–411.

[115] E. Kruse, R. Gutsche, and F. M. Wahl, "Estimation of collision probabilities in dynamic environments for path planning with minimum collision probability," in *Proceedings of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems' 96*, IEEE, vol. 3, 1996, pp. 1288–1295.

[116] J. Van Den Berg, P. Abbeel, and K. Goldberg, "LQG-MP: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.

[117] J.-Y. Le Boudec, D. McDonald, and J. Mundinger, "A generic mean field convergence result for systems of interacting objects," in *Fourth International Conference on the Quantitative Evaluation of Systems, 2007*, IEEE, 2007, pp. 3–18.

[118] A. Al-Hourani, R. J. Evans, and S. Kandeepan, "Nearest neighbor distance distribution in hard-core point processes," *IEEE Communications Letters*, vol. 20, no. 9, pp. 1872–1875, 2016.

[119] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, pp. 404–413, 1934.

[120] R. Simmons and S. Koenig, "Probabilistic robot navigation in partially observable environments," in *IJCAI*, vol. 95, 1995, pp. 1080–1087.

[121] F. Zanichelli, "Topological maps and robust localization for autonomous navigation," in *IJCAI Workshop on Adaptive spatial representations of dynamic environments*, Citeseer, 1999.

[122] I. Holmes and R. Durbin, "Dynamic programming alignment accuracy," *Journal of computational biology*, vol. 5, no. 3, pp. 493–504, 1998.

[123] F. Lavancier and J. Møller, "Modelling aggregation on the large scale and regularity on the small scale in spatial point pattern datasets," *Scandinavian Journal of Statistics*, vol. 43, no. 2, pp. 587–609, 2016.

[124] T. W. Anderson and L. A. Goodman, "Statistical inference about Markov chains," *The Annals of Mathematical Statistics*, pp. 89–110, 1957.

[125] J. Jeans, *An Introduction to the Kinetic Theory of Gases, Cambridge, UK: Cambridge University Press, 2009*. Jul. 2009.

[126] M. P. Allen, G. T. Evans, D. Frenkel, and B. Mulder, "Hard convex body fluids," *Advances in chemical physics*, vol. 86, no. 1, p. 166, 1993.

[127] V. Fourcassié, A. Dussutour, and J.-L. Deneubourg, "Ant traffic rules," *Journal of Experimental Biology*, vol. 213, no. 14, pp. 2357–2363, 2010.

[128] A. Dussutour, V. Fourcassie, D. Helbing, and J.-L. Deneubourg, "Optimal traffic organization in ants under crowded conditions," *Nature*, vol. 428, no. 6978, p. 70, 2004.

[129] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[130] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[131] R. Russell and T. Urban, "Vehicle routing with soft time windows and erlang travel times," *Journal of the Operational Research Society*, vol. 59, no. 9, pp. 1220–1228, 2008.

[132] S. Karlin, *A first course in stochastic processes*. Academic press, 2014.

[133] L. Le Cam, "Maximum likelihood: An introduction," *International Statistical Review/Revue Internationale de Statistique*, pp. 153–171, 1990.

[134] H. Van Beijeren and M. H. Ernst, "The modified enskog equation," *Physica*, vol. 68, no. 3, pp. 437–456, 1973.

[135] S. T. Paik, "Is the mean free path the mean of a distribution?" *American Journal of Physics*, vol. 82, no. 6, pp. 602–608, 2014.

[136] T Einwohner and B. Alder, "Molecular Dynamics. VI. Free-Path Distributions and Collision Rates for Hard-Sphere and Square-Well Molecules," *The Journal of Chemical Physics*, vol. 49, no. 4, pp. 1458–1473, 1968.

[137] G. F. Oster and E. O. Wilson, *Caste and ecology in the social insects*. Princeton University Press, 1979.

[138] D. Charbonneau, T. Sasaki, and A. Dornhaus, "Who needs lazyworkers? Inactive workers act as a reservelabor force replacing active workers, but inactive workers are not replaced when they are removed," *PloS one*, vol. 12, no. 9, e0184074, 2017.

[139] M. Jordan, "Why the logistic function? A tutorial discussion on probabilities and neural networks," Massachusetts Institute of Technology, Tech. Rep., 1995.

[140] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, "Particle filters for mobile robot localization," in *Sequential Monte Carlo methods in practice. Statistics for Engineering and Information Science*, Springer, New York, NY, 2001, pp. 401–428.

[141] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots," *Artificial intelligence*, vol. 128, no. 1-2, pp. 99–141, 2001.

[142] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Transactions on signal processing*, vol. 50, no. 2, pp. 174–188, 2002.

[143] K. H. Low, J. M. Dolan, and P. Khosla, "Active Markov Information-Theoretic Path Planning for Robotic Environmental Sensing," in *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, Taipei, Taiwan, May 2011.

[144] B. Tovar, L. Freda, and S. M. LaValle, "Using a robot to learn geometric information from permutations of landmarks," *Contemporary Mathematics*, vol. 438, pp. 33–46, 2007.

[145] W. M. Spears, D. F. Spears, R. Heil, W. Kerr, and S. Hettiarachchi, "An Overview of Physicomimetics," in *Swarm Robotics*, Springer Berlin Heidelberg, 2005, pp. 84–97.

[146] M. J. Schnitzer, "Theory of continuum random walks and application to chemotaxis," *Physical Review E*, vol. 48, no. 4, p. 2553, 1993.

[147] C. Dimidov, G. Oriolo, and V. Trianni, "Random walks in swarm robotics: An experiment with kilobots," in *International Conference on Swarm Intelligence*, Springer, 2016, pp. 185–196.

[148] M. Cates and J Tailleur, "When are active brownian particles and run-and-tumble particles equivalent? consequences for motility-induced phase separation," *EPL (Europhysics Letters)*, vol. 101, no. 2, p. 20 010, 2013.

[149] J Tailleur and M. Cates, "Statistical mechanics of interacting run-and-tumble bacteria," *Physical review letters*, vol. 100, no. 21, p. 218 103, 2008.

[150] R. Marino, "Dynamics and thermodynamics of translational and rotational diffusion processes driven out of equilibrium," PhD thesis, KTH Royal Institute of Technology, 2016.

[151] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multi-robot systems.," *IEEE Control Systems Magazine*, Accepted for Publication in 2020.

[152] D. Pickem, M. Lee, and M. Egerstedt, "The gritsbot in its natural habitat-a multi-robot testbed," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 4062–4067.

[153] Y. Lu, Y. Zhong, and B. Bhargava, "Packet loss in mobile ada hoc networks," 2003.

[154] V. Paxson, "End-to-end internet packet dynamics," in *ACM SIGCOMM Computer Communication Review*, ACM, vol. 27, 1997, pp. 139–152.

[155] G. H. Hagn and T. I. Dayharsh, "Land-mobile radio communication channel occupancy, waiting time, and spectrum saturation," *Electromagnetic Compatibility, IEEE Transactions on*, no. 3, pp. 281–284, 1977.

[156] R. C. Arkin and J Diaz, "Line-of-sight constrained exploration for reactive multiagent robotic teams," in *Advanced Motion Control, 2002. 7th International Workshop on*, IEEE, 2002, pp. 455–461.

[157] H Nguyen, N Pezeshkian, M Raymond, A Gupta, and J Spector, "Autonomous communication relays for tactical robots," DTIC Document, Tech. Rep., 2003.

[158] P. Ulam and R. C. Arkin, "When good communication go bad: Communications recovery for multi-robot teams," in *IEEE International Conference on Robotics and Automation, 2004.*, IEEE, vol. 4, 2004, pp. 3727–3734.

[159] S. Mayya and M. Egerstedt, "Safe open-loop strategies for handling intermittent communications in multi-robot systems," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2017, pp. 5818–5823.

[160] A Fedotov, V Patsko, and V Turova, *Reachable Sets for Simple Models of Car Motion*. INTECH Open Access Publisher, 2011.

[161] P. Souères, J.-Y. Fourquet, and J.-P. Laumond, "Set of reachable positions for a car," *Automatic Control, IEEE Transactions on*, vol. 39, no. 8, pp. 1626–1630, 1994.

[162] O. Maler, "Computing reachable sets: An introduction," Tech. Rep., 2008.

[163] J.-D. Boissonnat and X.-N. Bui, *Accessibility region for a car that only moves forwards along optimal paths*. INRIA France, 1994.

[164] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[165] H. J. Sussmann and G. Tang, "Shortest paths for the reeds-shepp car: A worked out example of the use of geometric techniques in nonlinear optimal control," *Rutgers Center for Systems and Control Technical Report*, vol. 10, pp. 1–71, 1991.

[166] S. Mayya, "Safe open-loop strategies for handling intermittent communications in multi-robot systems," Master's thesis, Georgia Institute of Technology, 2016.

[167] F John, "Extremum problems with inequalities as subsidiary conditions," in *Traces and Emergence of Nonlinear Programming*, Springer, 2014, pp. 197–215.

[168] O Güler and F Gürtuna, "Symmetry of convex sets and its applications to the extremal ellipsoids of convex bodies," *Optimization Methods and Software*, vol. 27, no. 4-5, pp. 735–759, 2012.

[169] P. Jaccard, *Distribution de la Flore Alpine: dans le Bassin des dranses et dans quelques régions voisines*. Rouge, 1901.

[170] M. Rubenstein, C. Ahler, and R. Nagpal, "Kilobot: A low cost scalable robot system for collective behaviors," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 3293–3298.

[171] L Giomi, N Hawley-Weld, and L Mahadevan, "Swarming, swirling and stasis in sequestered bristle-bots," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 469, no. 2151, p. 20 120 637, 2013.

[172] V Radhakrishnan, "Locomotion: Dealing with friction," *Proceedings of the National Academy of Sciences*, vol. 95, no. 10, pp. 5448–5455, 1998.

[173] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential/spl epsi/-tracking and/spl epsi/-stabilization of first-order nonholonomic se (2) vehicles," in *Proceedings of the 2002 American Control Conference*, IEEE, vol. 6, 2002, pp. 4690–4695.